



Univerzitet „Džemal Bijedić“ u Mostaru  
Fakultet informacijskih tehnologija

Završni rad nakon II ciklusa studija

## Mogućnosti primjene sistema preporuke u kolaborativnom web okruženju za e-učenje

*Mentor:*

doc. dr. Nina Bijedić

*Kandidat:*

Elvis Kadić

Mostar, Avgust 2014. godine

---

*Zahvaljujem mentoru, doc. dr. Nini Bijedić,  
profesoru na Fakultetu informacijskih tehnologija u Mostaru,  
na podršci, savjetima i primjedbama tokom izrade ovog rada.*

*Takođe, želio bih da se zahvalim cijelom osoblju Fakulteta informacijskih tehnologija u  
Mostaru, na maksimalnoj podršci tokom studiranja.*

*Veliko hvala mojoj porodici na podršci i strpljenju.*

## Sažetak

*Inspiracija za ovaj rad nastala je na osnovu činjenice, da moderno društvo doživljava virtualizaciju u punom smislu. Iako se ne može sa sigurnošću odgovoriti na pitanje da li je to zaista pozitivan smjer kretanja društva, realnost je takva kakva jeste i mora se prihvatiti. U ovom nezaustavljivom procesu, pojavljuju su se mnogobrojna virtualna okruženja, koja nude različite oblike kolaboracije među korisnicima ili grupama korisnika. Koncepti iz „stvarnog svijeta“ dobili su drugu dimenziju i trenutno doživljavaju svoj procvat. Između ostalog, pojavio se i revolucionarni koncept elektronskog učenja, koji je u akademskoj zajednici predmet sadašnjih, ali i budućih istraživanja. Ovaj rad će pokušati napraviti određene pomake u oblasti sistema preporuke, kako bi se korisnicima omogućilo ugodnije virtualno okruženje. U cilju implementacije takve ideje, neophodno je istražiti oblasti, kao što su: kolaboracija i komunikacija, upravljanje znanjem, rudarenje podataka, umjetna inteligencija, sistemi preporuke i sl. Postoji nada da će rezultati ovog istraživanja ostaviti dovoljno prostora za ispitivanje i primjenu novih pristupa.*

**Ključne riječi:** virtualizacija, kolaboracija, elektronsko učenje, rudarenje podataka, umjetna inteligencija

## Summary

*The inspiration for this work arose from the fact, that modern society is experiencing virtualization in the fullest sense. Although we can not answer with certainty the question of whether this is indeed a positive direction of the society, the reality is the way it is and must be accepted. In this unstoppable process, many virtual environments were being developed, offering various forms of collaboration among users or groups of users. The concepts of "real world" were given another dimension and is currently experiencing its peak. Among other things, there is also the revolutionary concept of e-learning, that is, the subject of current and future researches in the academic community. This paper will try to make some progress in the area of system recommendations, in order to provide users more comfortable virtual environment. To implement such ideas, it is necessary to explore the area, such as collaboration and communication, knowledge management, data mining, artificial intelligence, systems recommendations, etc. There is hope that the results of this study leave enough space for the study and application of new approaches.*

**Keywords:** virtualization, collaboration, e-learning, data mining, artificial intelligence

## SADRŽAJ:

<b>1</b>	<b>Uvod .....</b>	<b>1</b>
1.1	<i>Hipotetički okvir istraživanja .....</i>	3
1.1.1.	Glavna hipoteza .....	4
1.1.2.	Pomoćne hipoteze .....	4
1.2	<i>Predmet i ciljevi istraživanja .....</i>	4
1.3	<i>Metode i tok istraživanja .....</i>	5
<b>2</b>	<b>Pregled relevantnih istraživanja iz oblasti .....</b>	<b>7</b>
2.1	<i>Kolaborativni sistemi za upravljanje znanjem .....</i>	7
2.1.1	Kolaboracija .....	8
2.1.2	Komunikacija i upravljanje sadržajem .....	11
2.1.3	Kolaborativno upravljanje znanjem .....	12
2.1.4	Wiki modul .....	15
2.1.4.1	Primjena wiki-ja u kolaborativnom učenju .....	16
2.1.4.2	Primjena wiki-ja u proučavanju i istraživanju .....	20
2.1.4.3	Wiki alati .....	20
2.1.5	Forum .....	26
2.2	<i>Inteligentni programski agenti .....</i>	27
2.2.1	Struktura inteligentnog agenta .....	28
2.2.2	Vrste inteligentnih agenata .....	30
2.2.2.1	Jednostavni refleksivni agenti .....	31
2.2.2.2	Refleksivni agenti bazirani na modelu .....	31
2.2.2.3	Agenti bazirani na cilju .....	32
2.2.2.4	Agenti bazirani na korisnosti .....	33
2.2.2.5	Učeći agenti .....	33
2.2.2.6	Ostale klase inteligentnih agenata .....	34
2.3	<i>Sistemi preporuke .....</i>	35
2.3.1	Formalna definicija problema generisanja preporuke .....	36
2.3.2	Klasifikacija tradicionalnih sistema preporuke .....	38
2.3.2.1	Sistemi preporuke zasnovane na sadržaju .....	38
2.3.2.2	Kolaborativno filtriranje .....	40
2.3.2.3	Hibridni (mješoviti) sistemi preporuke .....	43
2.3.3	Ograničenja tradicionalnih sistema preporuke .....	44
2.3.4	Savremeni pristupi u razvoju sistema preporuke .....	45
2.3.4.1	Teorija i dosadašnja istraživanja .....	46
2.3.4.2	Adaptivni sistem preporuke(ASP) .....	47
2.3.4.3	Adaptivna agregacija .....	50
2.3.5	Prikupljanje podataka za model korisnika .....	56
2.3.5.1	Eksplisitno prikupljanje podataka .....	56
2.3.5.2	Implicitno prikupljanje podataka .....	57
<b>3</b>	<b>Praktični dio .....</b>	<b>60</b>
3.1	<i>FITCKMS .....</i>	60
3.1.1	FITCKMS - Opis sistema .....	61

3.1.1.1	Moduli sistema.....	61
3.1.2	FITCKMS – Baza podataka.....	62
3.1.3	FITCKMS - Pregled mogućnosti aplikacije .....	64
3.1.4	FITCKMS - Korisnički interfejs.....	64
3.2	<i>Eksperimentalno okruženje</i> .....	65
<b>4</b>	<b>Metode</b> .....	<b>67</b>
4.1	<i>Uloga inteligentnog programskog agenta</i> .....	67
4.2	<i>Priprema podataka</i> .....	69
4.3	<i>Pregled sistema preporuke</i> .....	72
4.3.1	Sistemi preporuke .....	73
4.3.1.1	Standardni (tradicionalni) sistemi preporuke .....	73
4.3.1.2	Adaptivni sistemi preporuke .....	74
4.3.2	Procjena greške.....	75
<b>5</b>	<b>Eksperimenti i rezultati</b> .....	<b>77</b>
5.1	<i>Eksperiment 1</i> .....	77
5.2	<i>Eksperiment 2</i> .....	79
5.3	<i>Eksperiment 3</i> .....	81
5.4	<i>Eksperiment 4</i> .....	85
<b>6</b>	<b>Diskusija i zaključak</b> .....	<b>89</b>
6.1	<i>Kolaboracija i upravljanje znanjem</i> .....	89
6.2	<i>Wiki stranica</i> .....	90
6.3	<i>Implikacije i ograničenja sistema preporuke</i> .....	90
6.4	<i>Kompleksnost adaptivnih sistema preporuke</i> .....	92
6.5	<i>Vremenska ograničenja</i> .....	93
6.6	<i>Mjere evaluacije</i> .....	93
6.7	<i>Inteligentni programski agenti</i> .....	94
6.8	<i>Pravci budućih istraživanja</i> .....	95
6.9	<i>Zaključak</i> .....	96
<b>7</b>	<b>Literatura</b> .....	<b>98</b>
<b>8</b>	<b>Prilozi</b> .....	<b>106</b>

# 1 UVOD

Informacione i komunikacijske tehnologije (ICT) su danas, direktno ili indirektno, integrisane u skoro sve aspekte društva. Vrijeme protoka i obrade informacija je svedeno na minimum, te je potreba za razvojem i implementacijom novih i savremenih softverskih rješenja u konstantnom porastu. Između ostalog, uvođenje ICT-a u obrazovne sisteme pokazalo se kao obavezujuće, te je u velikoj mjeri je uticalo na poboljšanje kvaliteta obrazovanja. Dva su osnovna koncepta ovakvog obrazovanja: *učenje na daljinu* (eng. *distance-learning*) i *elektronsko obrazovanje* (eng. *e-learning*). Spomenuti koncepti su vrlo slični i često se miješaju. Elektronsko obrazovanje se može definisati kao učenje uz pomoć računara, odnosno ICT-a, i podrazumjevano je u konceptu učenja na daljinu. Kombinacija e-obrazovanja sa tradicionalnim konceptom se naziva *miješano obrazovanje* (eng. *blended-learning*).

Pomenuti koncepti su, prije svega, omogućili brži protok informacija, podršku za multimedijalne nastavne materijale, kao i ogromne finansijske uštede obrazovnim ustanovama, osoblju i studentima. Može sa sigurnošću konstatovati da su svojevremeno predstavljali i još uvijek predstavljaju „revoluciju“ u obrazovanju.

Međutim, testiranjem u različitim okruženjima, otkriveni su mnogobrojni problemi koje je potrebno prevazići. Jedan od osnovnih problema ovakvih sistema je što se svi korisnici tretiraju na isti način. Oni imaju pristup istim resursima, kao i jednaku sistemsku podršku, bez obzira na njihovo znanje, ciljeve, komunikacijske vještine, interesovanja i slične karakteristike. Ovaj problem je vrlo česta pojava u sistemima današnjice i naziva se *problem latentne subjektivnosti* (eng. *latent subjectivity problem*).

Razmišljanje u smjeru pružanja individualnog pristupa korisnicima, bi predstavljalo rješenje za većinu navedenih problema, što bi ovakvim sistemima dalo potpuno novu dimenziju. Slijedeći tu ideju, razvoj modernih sistema je fokusiran na mnogobrojne modifikacije, odnosno različite module, koji omogućavaju prilagodljivu hipermedijsku podršku baziranu na korisničkoj individualnosti. Prije svega, ovaj koncept se odnosi na razvoj metoda i tehnika za adaptivnu navigaciju i prezentaciju sadržaja.

Cilj primjene adaptivne navigacije, odnosno adaptacije hiperveza<sup>1</sup> je navođenje korisnika prema relevantnim i zanimljivim informacijama. Tehnikama poput direktnog navođenja, sortiranja, označavanja, skrivanja i brisanja hiperveza pokušava se pojednostaviti njihova struktura, kako bi se umanjila dezorijentacija u hiperprostoru, a istovremeno sačuvala slobodna navigacija kao glavna karakteristika hipermedijske paradigme<sup>[1]</sup>.

Na osnovu primjera iz stvarnosti, početkom XXI vijeka počinje razvoj sistema preporuke u kontekstu e-učenja, gdje se različitim tehnikama studentima preporučuju online resursi, aktivnosti ili staze pretraživanja na osnovu njihovih karakteristika, kao i na osnovu aktivnosti njima sličnih studenata, čime se uvažava socijalna komponenta procesa učenja.

Iako vrlo slični adaptivnim hipermedijskim aplikacijama, kod kojih se prilikom adaptacije hiperveza u obzir uzimaju različiti aspekti korisnika (npr. predznanje, interesi, ciljevi i sl.), sistemi preporuke se često ograničavaju na jedan aspekt koji se naziva *interes korisnika*<sup>[1]</sup>. Istraživanja u ovoj oblasti usmjerena su na prevladavanje tog nedostatka koji je vrlo značajan za kontekst obrazovanja, u smislu dostupnosti i usvajanja novog znanja.

Korištenje sistema za preporuke u kontekstu hipermedijskih obrazovnih aplikacija se može direktno povezati i s novijim pristupima u razvoju obrazovnih okruženja na web-u koji ističu važnost *kolaborativnog*<sup>2</sup> učenja (*eng. collaborative learning*). Kolaborativno učenje se bazira na konstruktivističkoj teoriji, koja studenta stavlja u središte obrazovnog procesa i pretpostavlja da studenti uče jedni od drugih, dok učestvuju u različitim aktivnostima. Jedan od osnovnih ciljeva mnogobrojnih adaptivnih hipermedijskih sistema je i dostavljanje nastavnih materijala, tako da podrška kolaborativnom učenju predstavlja novu priliku za implementaciju adaptacije<sup>[2]</sup>.

U ovom radu će se razmatrati mogućnosti primjene različitih modela preporuke u kolaborativnim sistemima za upravljanje znanjem, odnosno u svim sistemima koji imaju

---

<sup>1</sup> Hiperveza ili poveznica (*eng. Hyperlink*) je osobina pojedinih, posebno označenih riječi, slika ili dijelova slika u tekstovima da predstavljaju posrednu vezu ka dodatnim informacijama. One predstavljaju referencu ili navigacioni element ka drugom odjeljku, drugom dokumentu ili odjeljku iz drugog dokumenta.

<sup>2</sup> Kolaboracija (*lat. co-* = „sa-“, *laborare* = „raditi“) označava saradnju pojedinaca ili grupa radi postizanja određenog cilja.

potrebu za predviđanje nepoznatih ocjena za određene sadržaje. Osim standardnih sistema preporuke, koji su već nekoliko godina u praktičnoj upotrebi, istraživanje će biti fokusirano na adaptivno-agregativne pristupe. Detaljno će biti istraženi i objašnjeni postojeći koncepti, problemi i napredak u ovoj oblasti. Takođe, posebna pažnja će biti posvećena problemima kao što su latentna subjektivnost i preopterećenje informacijama.

Kao dodatna tehnika za rješavanje spomenutih problema, simuliraće se upotreba inteligentnog programskog agenta, koja bi trebala napraviti određene pomake u smislu preciznosti predviđanja.

U praktičnom dijelu će biti razvijen *kolaborativni sistem za upravljanje znanjem (FITCKMS)*, kao i eksperimentalno okruženje, specijalizovano za testiranje ideja ovog rada. *FITCKM sistem* će se sastojati iz dva modula: *wiki stranica* (eng. *wiki page*) i *pitanja i odgovori* (eng. *questions and answers, QA*).

Za potrebe istraživanja teorija i hipoteza rada, u eksperimentalnom okruženju koristiće se stvarni podaci iz FITCKM sistema. Kao eksterni izvor podataka u eksperimentima, izabrana je besplatna kolekcija [MovieLens](#), koja po svim karakteristikama odgovara planiranim testovima i najčešće koristi za istraživanja u ovoj oblasti.

Iako je tema rada zaista široka i zahtijeva istraživanje iz različitih oblasti ICT-a, navedene kolekcije podataka će biti sasvim dovoljne za izvođenje planiranih eksperimenata i predstavljanje rezultata. Na osnovu rezultata praktičnog testiranja, bit će izvedeni zaključci i predstavljeni pravci budućih istraživanja u ovoj oblasti.

### **1.1 Hipotetički okvir istraživanja**

U skladu sa temom ovog rada i preliminarnim istraživanjima informatičkih oblasti kao što su: kolaboracija i komunikacija, upravljanje znanjem, rudarenje podataka, umjetna inteligencija, sistemi preporuke i sl., stvoreni su uslovi za razmišljanje o potpuno novim konceptima rješavanja problema personalizacije korisnika. Kombinovanjem novih ideja sa modelima, tehnikama i algoritmima, koji su trenutno u upotrebi, postavljene su slijedeće hipoteze:



### **1.1.1. Glavna hipoteza**

- *H1: Razvojem inteligentnih agenata, u kontekstu razotkrivanja nedavnih/trenutnih korisničkih kretanja, interesovanja i potreba, stvara se pretpostavka da je moguće razviti nove oblike znanja (metapodatke), i pomoću njih unaprijediti modele preporuke usmjerene ka konkretnom entitetu, odnosno postići viši nivo umjetne inteligencije sistema.*

### **1.1.2. Pomoćne hipoteze**

- *H1.1: Praćenjem korisnika kroz sistem moguće je otkriti nove odnose među podacima i stvoriti nove metapodatke, te iskoristiti njihovo značenje*
- *H1.2: Na osnovu novostvorenih metapodataka moguće je preciznije preporučiti resurse kombinovanjem adaptivnih pondera<sup>3</sup> određenih sistema preporuke.*

## **1.2 Predmet i ciljevi istraživanja**

Predmet istraživanja ovog rada je mogućnost primjene različitih sistema preporuke u kolaborativnim okruženjima, s akcentom na adaptivno-agregativni pristup. Kolaborativna okruženja, između ostalog, mogu poslužiti kao podrška samostalnim virtualnim zajednicama za e-učenje ili tradicionalnim LMS ili VLE sistemima.

Za realizaciju ovakvog koncepta je neophodno je postići slijedeće ciljeve:

- Istražiti koncepte, module i sisteme kolaborativnog upravljanja znanjem
- Istražiti mogućnosti primjene inteligentnog programskog agenta
- Analizirati i definisati postojeće koncepte sistema preporuke, koji će se koristiti u radu
- Definirati metode i testove koji će se koristiti u radu
- Razviti eksperimentalno okruženje za testiranje ideja i hipoteza rada
- Izvršiti eksperimentalna testiranja

---

<sup>3</sup> Relativni ponder pokazuje udio pojedine vrijednosti u strukturi cjeline

- Predstaviti rezultate i zaključke, te definisati pravce budućih istraživanja

Kroz detaljne analize prednosti i nedostataka pristupa korištenih u ovom radu, jedan od osnovnih ciljeva je omogućiti korisniku zanimljivo, korisno i ugodno kolaborativno okruženje.

Postoji iskrena nada da će rezultat, odnosno svrha ovog rada, biti postizanje određenih poboljšanja u oblastima kolaborativnog upravljanja znanjem i sistemima preporuke, te da će predstavljanje novih ideja možda odrediti pravac budućih istraživanja. To se ne odnosi isključivo domenu e-učenja, već bi se eventualno poboljšani koncepti mogli primjeniti u različitim scenarijima.

### **1.3 Metode i tok istraživanja**

Za testiranje hipoteza rada neophodno je napraviti detaljnu analizu dosadašnjih dostignuća iz oblasti koja se bavi sistemima preporuke. Posebna pažnja bit će usmjerena prije svega na trenutne koncepte korištenja takvih sistema. Osnovna ideja je adaptacija dosadašnjih dostignuća u ovoj oblasti, odnosno unaprijeđeno korištenje provjerenih metoda i algoritama sistema preporuke, koje bi bilo usmjereno ka individualnom korisniku.

Pokretačka ideja je veoma jednostavna i bazirana na tome da kvalitet preporuke ne zavisi od preciznosti korištenih metoda, već od različitosti skupova, obrade i aspekata podataka. Ovaj pristup bi mogao omogućiti pronalaženje skrivenih, potencijalno vrijednih, objekata čija bi se relevantnost dodatno provjeravala.

Za realizaciju ove ideje koristit će se standardne statističke metode koje na različite načine istražuju nepoznate veze među podacima. To svakako podrazumijeva korištenje kvalitetnih skupova podataka koji po svojoj strukturi i veličini odgovaraju podacima u stvarnim sistemima. U smislu poboljšanja kvaliteta podataka i predviđanja, glavnu ulogu će imati inteligentni programski agent opisan u poglavlju 4.1.

Svim standardnim metodama, odnosno algoritmima koje one koriste, bit će izmjereno vrijeme izvršavanja nad konkretnim podacima.

Nakon definisanja metoda i skupova podataka, pristupiće se eksperimentalnoj fazi. Na prvom sloju modelovanja izvršiće se procjena preciznosti svih standardnih metoda i upotrebom greška-modela, rezultati biti predstavljeni kao odstupanje od stvarnog predviđanja. Dobijene vrijednosti će se iskoristiti na drugom sloju modelovanja pri određivanju težina (pondera) svih korištenih standardnih metoda. Ponderi će, uz normalizaciju grešaka, određivati koliko će svaki od definisanih algoritama biti angažovan u adaptivnoj preporuci. Takođe i ovom adaptivnom pristupu će se procijeniti preciznost. Istraživanje ovog pristupa i njegova primjena detaljno je objašnjena u poglavljima 2.3.4 i 4.

Komparacijom rezultata dobijenih sa ili bez korištenja inteligentnog programskog agenta izvršiće se dodatna procjena preciznosti predviđanja.

## 2 PREGLED RELEVANTNIH ISTRAŽIVANJA IZ OBLASTI

U ovom poglavlju će se napraviti presjek pristupa i trenutnih dostignuća iz oblasti koje su zanimljive za ovaj rad. Posebno interesantne teme su: kolaboracija, komunikacija, upravljanje znanjem, inteligentni programski agenti i sistemi preporuke. Uz interpretaciju definicija, osnovnih i savremenih pristupa, razmotrit će se mogućnost poboljšanja u ovim oblastima.

Kolaborativna virtualna okruženja su dostigla zavidan nivo upotrebe, koji se teško mogao zamisliti prije samo par godina. Slične karakteristike se mogu pronaći u oblastima upravljanja znanjem i umjetne inteligencije. Iako su se u bliskoj prošlosti ovi koncepti smatrali nedokučivim, napredak je danas očigledan. S druge strane, trenutna dostignuća se mogu posmatrati samo kao početnički pokušaji, u odnosu na širok spektar ideja koji se pojavio u akademskoj zajednici.

Modeli preporuke, na određen način objedinjuju spomenute oblasti i u fokusu su interesovanja IT stručnjaka koji se zanimaju ili praktično na toj temi. Može se sa sigurnošću reći, da razvoj bilo kakvog informativnog sistema, bez modela preporuke, ne može nositi epitet „*savremenog sistema današnjice*“.

Trenutno su upotrebi različiti napredni sistemi preporuke koji koriste veoma komplikovane algoritme. Najbolji primjeri za to su popularne socijalne mreže, a u posljednje vrijeme primjena ovakvih sistema je sve češća u poslovnim aplikacijama, kao i u aplikacijama koje za osnovu koriste kolaboraciju među korisnicima.

U svim navedenim primjerima može se prepoznati jedinstveni cilj a to je omogućiti korisniku što ugodnije okruženje, sa što bržom isporukom interesantnih i značajnih sadržaja prilagođenim individualnom korisniku ili grupi.

### 2.1 Kolaborativni sistemi za upravljanje znanjem

Globalizacija, eksplozivan razvoj interneta kao i drugih komunikacijskih sistema omogućili su visoku dinamiku razmjene informacija i novu vrstu zajednica. Dvije osnovne karakteristike ovakvih zajednica su kolaboracija i upravljanje znanjem. Ove

karakteristike imaju binomni odnos, što znači, kolaboracija se ne može ostvariti bez kolaboracije, i obratno, kolaboracije u novom globalnom društvu, se ne može graditi bez kolaboracije.

Problem je dosta kompleksan, jer u kolaboracijskim sistemima, znanje je heterogeno i *sistem upravljanja znanjem (KMS)* treba prevesti ogromne količine *prešutnog znanja*<sup>4</sup>, u *eksplicitno znanje* i obrnuto. Shodno tome, nastala je nova oblast pod nazivom *kolaborativni sistemi upravljanja znanjem (CKMS)*<sup>[3]</sup>.

Paradigme kolaboracije i upravljanja znanjem su u međusobnoj interakciji i stvaraju nove vrijednosti društva kao što su kolaborativne zajednice ili znanje izgrađeno na osnovu društva. Jedan od glavnih izazova u akademskom istraživanju, ali i u praksi, je proučavati kako ovaj mehanizam funkcioniše u stvarnom svijetu. U tom smislu, postoji veliki broj studija koje se uglavnom baziraju na istraživanje ekonomije i marketinga, ali su vrlo rijetke u oblasti obrazovanja. Ovaj problem je još izraženiji u zemljama u razvoju, gdje je iskustvo sa primjenom ovakvih sistema na vrlo niskom nivou.

U skladu sa temom rada, u ovom poglavlju bit će istražena oblast *kolaborativnih sistema za upravljanje znanjem (CKMS)*, virtualne zajednice (*VC*) i osnovni koncepti društvenih mreža (*SN*).

### **2.1.1 Kolaboracija**

Kolaboracija se može posmatrati iz različitih perspektiva. Opšta definicija kolaboracije može podrazumjevati slijedeće<sup>[4]</sup>:

*“Kolaborativni sistemi predstavljaju interdisciplinarno polje, i mogu se posmatrati kao presjek informatike, kibernetike, kognitivnih znanosti, psihologije, lingvistike, sistema za podršku u odlučivanju, upravljanja i sl. “*

Iz navedene definicije, mogu se izdvojiti određene specifičnosti:

---

<sup>4</sup> Prešutno znanje - znanje koje može steći iskustva i komunikacije

- Kolaboracija podrazumijeva vrlo sofisticirane odnose, odnosno zahtijeva velike napore za ujedinjavanje ljudi i organizacija, radi ostvarivanja zajedničkih ciljeva, koji se ne mogu postići samostalnim djelovanjem istih<sup>[5]</sup>.
- Kolaboracija je proces u kojem učesnici dijele podatke, resurse i odgovornosti, te zajednički planiraju, implementiraju, i evaluaciju programa aktivnosti za postizanje zajedničkog cilja<sup>[6]</sup>.

Autori<sup>[7]</sup> su koristili sljedeću definiciju, poznatu kao 3C paradigma:

„Kolaboracija je kooperacija, komunikacija (umrežavanje) i koordinacija<sup>[8][9]</sup>“.

- *Kooperacija* se posmatra kao zajednički napor za postizanje zajedničkih ciljeva;
- *Komunikacija* je aktivnost razmjene informacija između učesnika;
- *Koordinacija* s aktivnostima osigurava da različiti učesnici zajedno rade na postizanju zajedničkih ciljeva.

Drugi autori<sup>[10]</sup> insistiraju na kolaborativnoj kreativnosti, kao i na značajnoj ulozi upravljanja sadržajem. Prema njima, upravljanje sadržajem je proces stvaranja, ažuriranja i objavljivanja sadržaja na internetu, te može se tumačiti na različite načine:

- *S aspekta poslovanja*, upravljanje sadržajem predstavlja sredstvo za isporuku poslovnih vrijednosti, koristeći procese i alate koji se nalaze u pozadini
- *Sa tehničkog stanovišta*, upravljanje sadržajem predstavlja kombinaciju hardvera i softvera koji čine sistem za upravljanje sadržajem

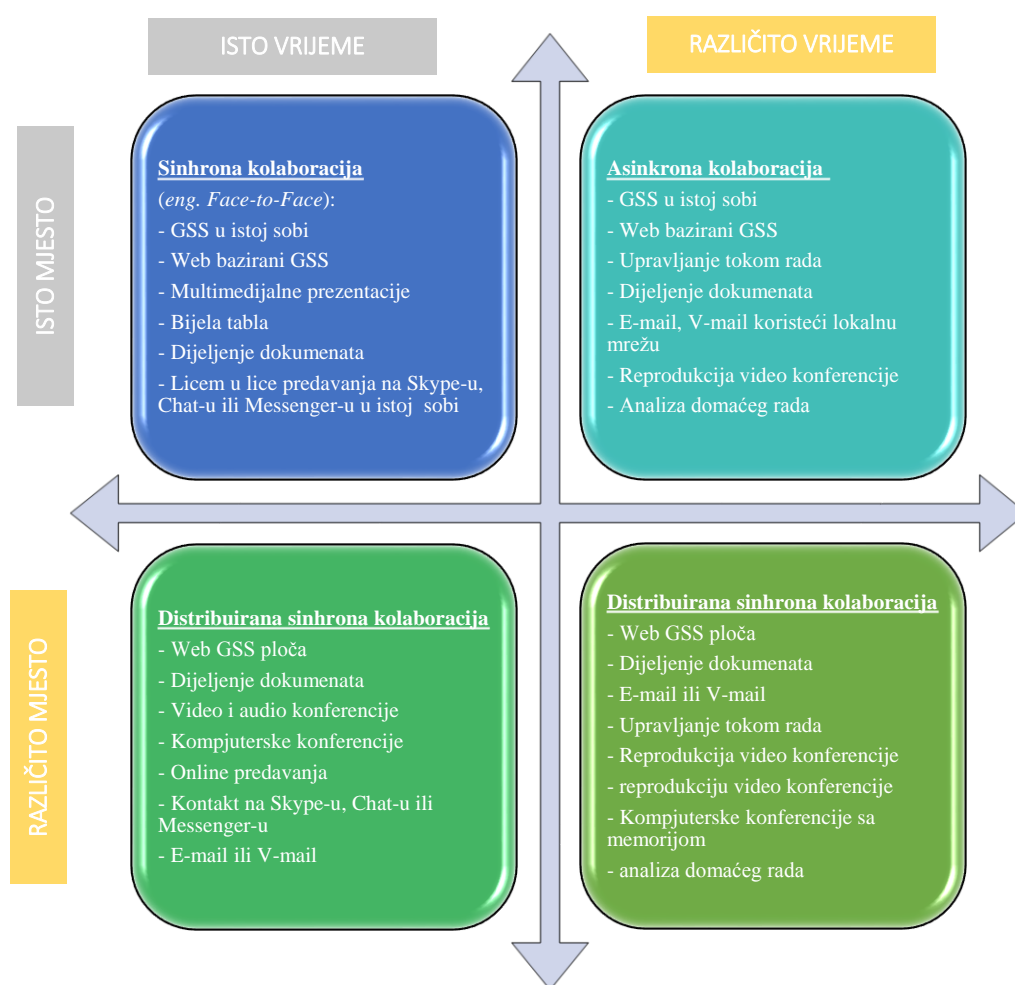
U navedenom kontekstu, kolaboracija se može posmatrati kao presjek četiri osnovne domene: *kooperacije, komunikacije, koordinacije i upravljanja sadržajem*. Proizvod ovakve teorije se može protumačiti kao oplemenjivanje 3C paradigme, koja prvenstveno podrazumijeva mogućnost integrisanja metoda upravljanja sadržajem u kolaboraciji. U modernim sistemima, upravljanje sadržajem ima jednu od najvažnijih uloga.

Takođe, kolaboracija se može posmatrati sa teoretskog, ali i sa praktičnog stanovišta, tako da je i bez detaljnih studija, moguće prepoznati dvije taksonomije:

a) Pozadina i uloge učesnika u saradnji sa epistemičkog<sup>5</sup> stanovišta<sup>[11]</sup>:

- Poslodavac / Radnik
- Učitelj / Učenik
- Slični entiteti
- Različiti entiteti

b) Drugo stanovište je praktično i predstavlja kombinaciju komunikacijskih modela za vrijeme i mjesto<sup>[12]</sup> s nekim od kolaborativnih tehnologija<sup>[11]</sup> (slika 1).



Slika 1 Kombinacija vrijeme/mjesto komunikacijskih modela i kolaborativnih tehnologija

<sup>5</sup> Epistemologija (iz grčkog jezika επιστήμη - episteme, "znanje" + λόγος, "logos") ili teorija znanja je grana filozofije koja se bavi prirodom i dosegom znanja. Izraz je prvi uveo škotski filozof James Frederick Ferrier (1808. – 1864.).

Kako bi se iskoristio puni potencijal sticanja, upravljanja, dijeljenja i širenja znanja u dinamičkom okruženju, na polju saradnje moraju biti urađene određene modifikacije sistema <sup>[13]</sup>.

### **2.1.2 Komunikacija i upravljanje sadržajem**

*"Bez komunikacije, nema kolaboracije" <sup>[11]</sup>.*

Komunikacija se može ostvariti na različite načine, putem: direktne komunikacije (*eng. face-to-face*), e-mail poruka, razgovorom (*eng. chat*), e-mail listi (*eng. listservs*), grupa, oglasnih ploča, video konferencija i sistema za elektronske sastanke, portala i sl.

*„Osnovni zadatak komunikacije je da prenosi informacije od pošiljaoca do primaoca, ali kolaboracija je mnogo dublji pojam. Kolaboracija prenosi značenje ili znanje među članovima grupe“ <sup>[11]</sup>.*

U velikim grupama, informacija je heterogena i neophodna za generiranje rezultata za upravljanje sadržajem. Sistem za upravljanje sadržajem (CMS) osigurava nabavku, upravljanje i objavljivanje sadržaja. Ovi sistemi mogu biti sastavljeni od različitih komponenti, ali u okvirima ovog rada, najvažniji dijelovi će predstavljati upravljanje dokumentima i upravljanje Web sadržajem.

Tehnički aspekti CMS-a su vrlo zanimljivi, kao na primjer, različiti formati podataka i hiperveza (HTML, RDF ili XML). Jedan od najinteresantnijih primjera CMS-a je Wiki povezivanje (*eng. Wiki linkage*). To je klasični Wiki sistem prikupljanja Web stranica putem hiperveza <sup>[14]</sup>. Prednosti ovakvih sistema su:

- Ne zahtijevaju tehničke sposobnosti kao što su HTML
- Posjeduju povratni mehanizam s verzijama
- Osiguravaju različite načine povezivanja
- Osiguravaju učitavanje različitih sadržaja
- Posjeduju funkcije za pretraživanje
- Predstavljaju vrlo jaku bazu za CKMS, tj. sadrži heterogene informacije



U posljednje vrijeme, klasični Wiki je proširen i podržava različite semantičke oblike <sup>[14]</sup>.

### 2.1.3 Kolaborativno upravljanje znanjem

Kolaborativno upravljanje znanjem se bazira na dvije, vrlo važne, nove IT tehnologije <sup>[15]</sup>:

- sistemi za upravljanje znanjem (KMS)
- kolaborativni informacijski sistemi (CIS)

Osnovni cilj upravljanja znanjem (KM) je prikupljanje značajnih informacija iz znanja, kojeg organizacije posjeduju kroz svoje adekvatno i eksplicitno upravljanje <sup>[16]</sup>. KM uključuje alate, strategije i praktična iskustva za identifikaciju, razvoj, predstavljanje, distribuciju i dostupnost različitih vrsta znanja <sup>[17]</sup>. Znanje je prihvaćeno kao najvažnija imovina današnjih organizacija <sup>[18]</sup>. KM također ima različita značenja. Dva gotovo ekvivalentna značenja su :

- Upravljanje znanjem je sastavljeno od prikupljanja, kategorizacije i širenja znanja u organizaciji <sup>[16]</sup>.
- Upravljanje znanjem je sastavljeno od praktičnog sticanja, snimanja, gradnje, dijeljenja i korištenja znanja <sup>[19]</sup>.

Neki autori <sup>[20]</sup> sisteme za upravljanje znanjem opisuju kao specijalne, tekst ili hipertekst orijentirane sisteme, jer uglavnom upravljaju dokumentima različitih formata.

Sistemi za upravljanje znanjem su fokusirani na neprestano poboljšanje performansi, prednošću nad konkurencijom i inovacijama na dva glavna polja: *organizacijskog učenja* (OL) i *organizacijske memorije* (OM), kao *spremište znanja* (KR) koje sadrži znanje i najbolje prakse. *Spremište znanja* se sastoji od strateške imovine i ima zadatak podsticanje i dijeljenja znanja u organizacijama.

Upravljanje znanjem se teorijski i praktično počelo proučavati krajem prošlog vijeka i trenutno se postepeno kreće do akademske zrelosti. *Organizacijskom učenju* je posvećena posebna pažnja, prvenstveno zbog pozitivnog efekta u skladištima organizacijskog znanja

kod promjenjivih i dinamičkih okruženja<sup>[21]</sup>. To osigurava razvoj novih znanja, koja mogu biti jako važna u ponašanju organizacije. U virtualnim okruženjima i društvenim mrežama, povećava se potreba za *organizacijskim učenjem* prvenstveno radi dijeljenja i širenja informacija. *Organizacijsko učenje, u pravilu, uvijek započinje s pojedinačnim učenjem, te prerasta u kolektivno učenje, pri čemu se misli na grupe, timove ili nivo cijele organizacije. Sposobnosti učenja uključuju:*

- otvaranje novih perspektiva
- definisanje ličnih sklonosti
- rad sa sirovim podacima
- realno razumijevanje

Organizacijsko učenje može ponuditi organizaciona rješenja kod slijedećih pitanja<sup>[22]</sup>:

- sistemskih rješavanja problema
- kreativnih eksperimenata
- učenja na osnovu ranijeg iskustva
- učenja na osnovu najboljih praksi sličnih sistema
- brze i efikasne transformacija znanja u organizaciji

*Organizacijska memorija* osigurava prikupljanje, prezentaciju, pohranjivanje i distribuciju znanja u organizaciji. Formirana je<sup>[23]</sup> od četiri tipa memorije:

- *radna memorija* sadrži eksplicitno referentno znanje, dokumente, korisne programe koji se koriste za poslovanje
- *organizacija memorija* je vezana za aktivnosti organizacije i njenih partnera
- *individualna memorija* sadrži sposobnosti i kompetencije pojedinaca
- *projektna memorija* pohranjuje definicije projekta, aktivnosti, resurse, povijest projekta i rezultate

Iz perspektive domene sistema organizacijska memorija se dijeli na :

- internu memoriju , sadrži znanje i informacije unutar organizacije
- vanjsku memoriju , sadrži znanje i informacije iz okoline

Može se zaključiti da je OL i OM zavise manje o tehnologiji, a više o ljudskim resursima. Stvaranje novog znanja podrazumijeva<sup>[24]</sup>:

- generisanje novih značenja, ideja i ponašanja
- spajanje eksplicitnog i prešutnog znanja
- realizacija spiralnog znanja pojedinca, grupe ili organizacije

Postoje četiri modela stvaranja znanja<sup>[24]</sup> :

- *socijalizacija* - pretvaranje prešutnog znanja u novo prešutno znanje pomoću socijalne interakcije i razmjene znanja između članova virtualne organizacije (npr. pomoću instrukcija)
- *eksternalizacija* - prevođenje prešutnog znanja u novo eksplicitno znanje (proizvodnju novog nastavnog materijala u školi )
- *internalizacija* - stvaranje prešutnog od eksplicitnog znanja (stvaranje značenja iz pisanih dokumenata ili upravljanje sadržajem)
- *kombinacija* - stvaranje novih eksplicitnih znanja prema klasifikaciji, kategorizaciji i sistematizaciji (npr. upotreba rudarenja podataka (DM) ili statističkih analiza nad različitim skladištima podataka koristeći upite za otkrivanje znanja (KDQL)<sup>[25]</sup>).

Prikupljanje znanja podrazumijeva otkrivanje izvora znanja i korištenje internog znanja, te se može organizovati različitim metodama i bez informacionih tehnologija. Jedan od najčešćih je verbalni diskurs<sup>6</sup>.

Znanje je moguće prikupiti od pojedinaca ili grupa. Osnovne tehnike pojedinačnog prikupljanja znanja su: intervjui, neposredno opažanje, upitnici, analiza prethodnih slučajeva, introspekcije , simulacija i sl<sup>[28]</sup>.

Vrlo interesantno pitanje je<sup>[26]</sup>: "*Mogu li se radne grupe posmatrati kao distributivni sistem za upravljanje znanjem?*". Taj problem nije dovoljno istražen i zastupljen u

---

<sup>6</sup> Diskurs (lat. discursus) je imenica koja znači govor, razgovor, odnosno raščlanjivanje, analizu. Danas se često rabi u semantici, jezikoslovnoj disciplini gdje označava lingvističku jedinicu sastavljenu od nekoliko rečenica.

literaturi. Upravljanje znanjem je više koncentrisano na integraciju znanja u grupama samo kao aditiv. Takođe, neki autori smatraju<sup>[10]</sup> da su distributivni sistemi za upravljanje znanjem jako komplikovani u virtualnim organizacijama. Uopšteno<sup>[15]</sup>, mreže znanja koriste informacione tehnologije za dijeljenje znanja i mudrosti među pojedincima, grupama i jedinicama širom organizacije, pa čak i sa vanjskim organizacijama.

U praktičnom dijelu ovog rada, tačnije u FITCKM Sistemu (*Poglavlje 3.1.1.1*), će se pristupiti integraciji modula *Wiki stranica* i *Pitanja i odgovori*, koji predstavljaju asinhrono kolaborativno okruženje. U narednim poglavljima predstavljene su mogućnosti ovih modula.

#### **2.1.4 Wiki modul**

U bliskoj prošlosti je bilo potrebno ići u biblioteku i satima čitati ili prepisivati tekstove neophodne za školovanje. Poseban značaj pridavao se enciklopedijama, kao vjerodostojnim i sistematizovanim skupovima objašnjenja i tumačenja. Porastom upotrebe ICT-a, porasle su i potrebe korisnika, te su zbog jednostavnosti i lakoće upotrebe, elektronski formati dokumenata zasjenili one štampane. Iz ovog ugla gledano, internet je danas jedna velika biblioteka bogatog sadržaja, i skoro sve informacije su udaljene na samo par „klikova“.

Na web stranici američkog programera *Varda Kaningema*, 1995. godine je prvi put predstavljena revolucionarna ideja, da bilo ko može uređivati članke online enciklopedije. Vrlo brzo je stekla ogromnu popularnost. Upravo ovaj autor je i tvorac naziva “wiki”, što znači brzo<sup>[12]</sup>. Spomenuta web stranica je ustvari preteča, sada već popularne Wikipedije, enciklopedije slobodnog sadržaja, koju razvijaju dobrovoljci uz pomoć wiki softvera. Danas, wiki predstavlja dio serverskog softvera koji dozvoljava korisnicima da kreiraju i vrše izmjene nad člancima, uz pomoć bilo kog pretraživača<sup>[34]</sup>.

Primjena wiki-ja u učenju ogleda se u obezbjeđivanju relevantnih tekstova iz velikog broja oblasti. Jedinstvena baza tekstova različitih korisnika, omogućava efikasno pretraživanje, kao i izmjenu članaka<sup>[34]</sup>.

U kontekstu kolaborativne podrške korisnicima, wiki nudi različite alate, kao što su: PBWiki, Drupal Wiki, Twiki, MediaWiki, FosWiki i mnogi drugi. Spomenuti alati omogućavaju saradničko formiranje baza znanja. Akcent je na aktivnoj participaciji korisnika, koji, uvidom u učinak svojih kolega, stiču nova znanja, istražuju, rješavaju probleme i izrađuju projekte. Svaki od autora može predlagati izmjene rada, koje su vidljive i ostalim autorima pa se mogu dodatno mijenjati.

Wiki predstavlja idealan izbor za naučnike koji žele da objave svoje istraživačke članke na Internetu, a nisu upućeni u osnove HTML kodiranja, odnosno web programiranja. Takođe, oni žele podijeliti svoja iskustva komentarišući članke koji im se učine zanimljivim. Wiki alati povezuju istraživače sa geografski udaljenih lokacija, te je njihova ekspanzivna upotreba u praksi opravdana i logična.

U ovom poglavlju, prikazan je osvrt na primjenu wiki alata u kolaborativnom učenju, poučavanju i istraživanju.

#### *2.1.4.1 Primjena wiki-ja u kolaborativnom učenju*

Wiki je, kao što je već naglašeno, vrlo moćan sistem koji nudi različite mogućnosti saradnje i razmjene znanja. U kolaborativnom učenju, korisnici mogu da rade u heterogenim grupama za podršku učenju. Kolaborativno učenje omogućava pozitivne međuzavisnosti između članova grupa, pojedinačne odgovornosti, interakcije licem u lice, što rezultira razvojem individualnih kolaborativnih vještina.

Rad u grupi omogućava postizanje boljeg rezultata i članovi imaju pristup velikoj količini različitih informacija, za razliku od korisnika koji individualno obavljaju neki zadatak. Velika podrška za kolaborativno učenje je mogućnost saradnje korisnika iz različitih sredina. Učenje podržano ICT-om, je znatno olakšano uz pomoć wiki-ja, što obrazovanju daje potpuno novu dimenziju. Takođe prilikom kolaborativnog rada dolazi do poboljšanja korisničke interakcije, distribucije znanja i razmjene stručnosti u zajednici.

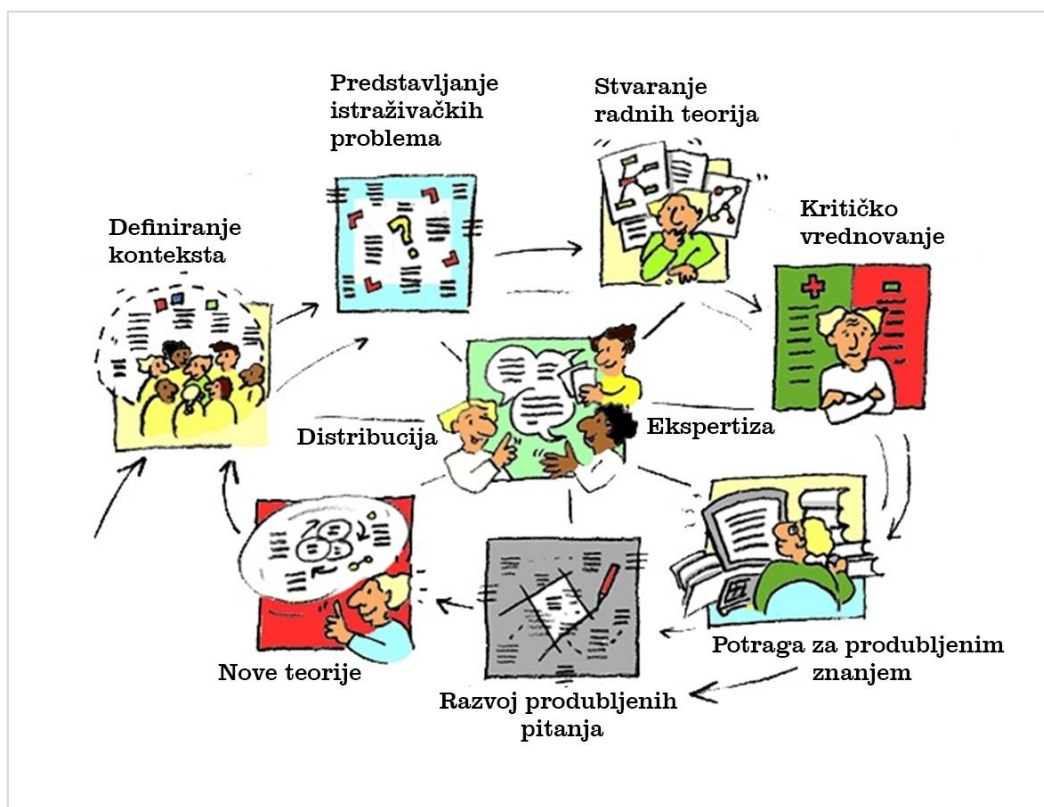
Wiki u procesu kolaborativnog učenja takođe poboljšava asinhronu komunikaciju i kooperativno učenje između korisnika i promovira saradnju umesto konkurencije. Kolaborativno učenje postaje još snažnije kada se odvija u kontekstu zajedničke prakse.,

koja se sastoji od članova koji rade u kolektivnom učenju određenom domenu i učenje postaje zajednički proces u okviru radne grupe. Wiki karakteriziraju i neki elementi od fundamentalnog značaja za uspješnu grupnu realizaciju zadataka, uključujući i virtualno prisustvo, lako i jednostavno učešće u izradi zadataka, vezu sa širokim krugom korisnika, lični i zajednički identitet i interakcija.

Kolaborativno učenje treba da uključi korisnike u učenju smisleno i kritično tako da :

- Aktivno i manipulacijski angažira učenike u interakcijama i istraživanju sa nastavnim materijalima
- Konstruktivno i reflektirajuće omogući studentima/učenicima da integiraju nove ideje uz prethodna znanja i da omogući učenje kroz refleksiju
- Pruže mogućnost studentima da manipuliraju svojim ciljevima učenja i prate napredak dostizanja ciljeva
- Autentično i izazovno u realnom svijetu omoguće razumjevanja i transfer učenja u novim situacijama
- Pruže studentima mogućnost da komuniciraju jedni sa drugima, da razjasne i razmjenjuju ideje, traže pomoć i diskutuju rješenja.

U posljednjih nekoliko godina, oni koji su uključeni u online učenje imaju veću svijest o prednostima kolaborativnog učenja u virtualnom okruženju korištenjem wiki-ja. Istraživanja pokazuju da veliki broj korisnika, koji su koristili wiki-je za kolaborativno učenje, imaju pozitivno mišljenje o ovakvom načinu učenja. Online razvoj saradničkih tekstova pruža nove mogućnosti, ne samo kada je reč o usvajanju znanja, već podstiče i, takozvano, saradničko razmišljanje. Takođe, sve je više nastavnog osoblja koji koriste ovaj način saradnje i dijeljenja znanja. Kreiranje nastavnog sadržaja postaje jedan kreativan proces, a i dovodi do primjene principa demokratičnosti u obrazovanju. Na ovaj način, savremene tehnologije se uključuju u nastavni proces. Često se uspostavlja kulturna i prijateljska saradnja između korisnika<sup>[34]</sup>.



Slika 2 Kolaborativno učenje

Karakteristike kolaborativnog učenja:

- Članovi u grupi, razmjenjuju ideje i misli, zajednički uče i istražuju,
- Akumuliranje znanja o datoj temi, razvijanje jednostavnih vještina, rješavanje problema,
- Kolaborativno učenje se smatra distribuiranim procesuiranjem ili inteligencijom, što daje veći spektar mogućih kombinacija ideja i stavova, sa manjim posljedicama i puno većom vjerovatnoćom dolaska do najboljeg mogućeg rešenja postavljenog problema
- Početna ideja pojedinca se u toku kolaboracije nadograđuje (oplemenjuje) zahvaljujući razmišljanjima ostalih članova. Takvim pristupom pojedinac se lakše oslobađa predrasuda, odnosno ograničavajućih stavova koji su prepreka konstruktivnoj promjeni mišljenja kao rezultatu efektivnog učenja.
- Svaka početna ideja nosi potencijal generiranja novih ideja u kolaborativnoj zajednici i veliku je vjerovatnoću aktiviranja naprednih ideja i rješenja, zbog čega se stvara

- Uz pomoć kolaborativne grupe lakše je uvidjeti pogrešan stav i razmišljanje, te je moguće je pravovremeno djelovati
- Korisnici preuzimaju odgovornost za samostalno učenje, kao i za učenje u grupi
- U kolaborativnoj zajednici događaju se lične i društvene preobrazbe. Korisnici se ohrabruju da se suprotstave autoritetima, pa kolaborativno učenje vodi razvijanju stvarne “demokratije” u kojoj svako ima pravo glasa
- Alternativni način provjere znanja studenta/učenika,
- Razvijanje socijalnih vještina kolaborativnih partnera<sup>[35]</sup>.

The diagram illustrates the information management process with the following components and relationships:

- KOORDINACIJA** (Coordination) and **SARADNJA** (Cooperation) are represented by light blue rounded rectangles at the top.
- KOMUNIKACIJA** (Communication) is represented by a light blue rounded rectangle at the bottom.
- PERCEPCIJA (opažanje)** (Perception/Observation) is represented by a grey triangle in the center.
- SUKOBI** (Conflicts) is represented by a grey oval at the top right.
- PREZASIĆENOST INFORMACIJAMA** (Information Overload) is represented by a grey oval at the bottom right.
- KOLABORACIJA** (Collaboration) is represented by a purple rounded rectangle at the bottom left.

The relationships between these components are as follows:

- KOORDINACIJA** *organizira* (organizes) **SARADNJA**.
- KOORDINACIJA** *omogućava* (enables) **PERCEPCIJA**.
- SARADNJA** *omogućava* (enables) **PERCEPCIJA**.
- PERCEPCIJA** *opskrbljuje informacijama* (provides information) to **KOORDINACIJA** and **SARADNJA**.
- KOMUNIKACIJA** *opskrbljuje informacijama* (provides information) to **PERCEPCIJA**.
- PERCEPCIJA** *omogućava* (enables) **KOMUNIKACIJA**.
- KOLABORACIJA** *omogućava* (enables) **KOMUNIKACIJA**.
- KOMUNIKACIJA** *uzrokuje* (causes) **KOORDINACIJA** and **SARADNJA**.
- KOORDINACIJA** *razmatra* (considers) **SUKOBI**.
- SARADNJA** *ugrožavaju* (threatens) **SUKOBI**.
- SARADNJA** *prikazuje rezultate* (presents results) to **KOMUNIKACIJA**.
- PERCEPCIJA** *prikazuje rezultate* (presents results) to **PREZASIĆENOST INFORMACIJAMA**.
- PREZASIĆENOST INFORMACIJAMA** *omogućava* (enables) **KOMUNIKACIJA**.

19



#### *2.1.4.2 Primjena wiki-ja u proučavanju i istraživanju*

Velika prednost wiki-ja u poučavanju se ogleda u procesu učenja na daljinu, čime se obezbjeđuje olakšanje u komunikaciji, oblikovanju i dijeljenju znanja. Jedan od nedostataka je validnost objavljenih podataka, jer ukoliko je wiki otvoren podatke može upisivati bilo ko. Ukoliko se postavi neki sadržaj, bilo da je tekst ili prezentacija, proučavanjem ovih sadržaja i diskusijom na ovu temu učenici/studenti mogu imati direktan uticaj na preoblikovanje nastavnog materijala za buduće generacije. Wiki platforma za podučavanje donosi prednosti, odnosno pogodnosti predavačima da imaju direktan i kompletan uvid u pojedinačni učinak svakog učenika/studenta kao i kompletnu istoriju razvoja nekog rešenja, ali zahtjeva dosta vremena odnosno praćenja rada svojih učenika/studenata<sup>[36]</sup>.

Ako se posmatra istraživački rad u okvirima procesa učenja, najveća vrijednost wiki-ja, leži u ogromnoj bazi podataka, koja sadrži kvalitativne i kvantitativne podatke. Na primjer, kvantitativni podaci mogu biti: broj pristupa svakog korisnika, vrijeme provedeno u određenoj aktivnosti i sl., dok nastavno osoblje može dobiti kvalitativne podatke, recimo posmatranjem i analiziranjem diskusija koje vode korisnici. Međutim, ako se izađe iz ovih okvira može se doći do veoma velikog broja raznih podataka koji se mogu koristiti u sveobuhvatnim istraživanjima<sup>[36]</sup>.

#### *2.1.4.3 Wiki alati*

Wiki alati predstavljaju web stranice čiji sadržaj može lako da se kreira i da bude mijenjan od strane zajednice, koje najčešće uključuju i posjetioce. Postoje desetine wiki softvera koje imaju svoje prednosti i mane ali suština svakog od tih softvera je ista. Radi se o web platformama koje kreiraju web stranice i napravljene su tako da ih bilo ko može mijenjati i uređivati i koje služe kao informacije drugim ljudima koji žele da vide sadržaj te stranice. Još jedna od bitnih karakteristika jeste povezivanje sa drugim wiki stranicama.

U narednom dijelu rada ukratko će biti predstavljeni najpopularniji wiki alati: MediaWiki, DokuWiki, TermWiki, WikiEducator, Chiq Chaq, PmWiki, FosWiki, i MathWiki.

#### 2.1.4.3.1 MediaWiki

MediaWiki predstavlja softver izdat po GNU<sup>7</sup> licencom. Ovaj softver je dizajniran za pokretanje na velikom serveru, tako da ga istovremeno može koristiti veliki broj korisnika i može podržati veliki broj različitih stranica. Koristi MediaWiki wiki-tekst format, tako da korisnici ne moraju posjedovati programerske vještine, prije svega XHTML i CSS, kako bi uređivali stranicu. Korisnik kada izmjeni određenu stranicu i sačuva promjene, MediaWiki automatski tu promjenu upisuje u bazu podataka ali ne briše prethodnu verziju u slučaju da dođe do zloupotreba. Ekstenzije za MediaWiki uglavnom određuje sistemski administrator, ali to mogu uraditi i programeri iz cijelog svijeta. Pošto se radi o kompleksnom softveru, dosta su česte greške, pa su kreatori MediaWiki organizovali web stranicu [Bugzilla](#) na kojoj korisnici mogu da prijavljuju greške<sup>[37]</sup>.



Slika 4 MediaWiki - logo

#### 2.1.4.3.2 DokuWiki

Ovaj alat je namijenjen razvojnim timovima i radnim grupama, razvijen je pod GPL2 (*General Public Licence ver.2*) licencom i kreiran je u PHP programskom jeziku. Tvorac ovog softvera je Andreas Gohr (Jun 2004. godine). Prednosti u odnosu na druge slične alate, su laka instalacija i upotreba, niski hardverski zahtjevi, sigurnija kontrola pristupa (Lista pristupa), veliki izbor dodataka, više od 55 podržanih jezika, softver otvorenog koda i sl. Ovaj softver ima jednostavnu ali ujedno i kompletnu sintaksu (sličnu onoj koju koristi MediaWiki), tako da su fajlovi rađeni u DokuWiki-ju kompatibilni sa čitačima izvan Wiki-ja. Takođe, je prilagođen kolaborativnom radu sa opcijom čuvanja cjelokupne istorije promjena i nije omogućeno istovremeno paralelno uređenje stranice. Kontrola

---

<sup>7</sup> GNU je operativni sistem sastavljen isključivo od slobodnog softvera. Njegovo ime je rekurzivna skraćenica za "GNU Nije Unix" ("GNU's Not Unix").

pristupa se može definisati od strane administratora putem kontrolne liste. Podaci se čuvaju kao obični tekstualni fajlovi tako da nisu potrebne baze podataka<sup>[38]</sup>.



*Slika 5 DokuWiki - logo*

#### *2.1.4.3.3 TermWiki*

TermWiki je glavna socijalna mreža koja omogućava korisnicima da uče, otkrivaju, dijele i čuvaju lične termine i rječnike u 1764 kategorija, na 102 jezika sa ukupno 6 964 453 termina tj. riječi. Sajt naglašava saradnju preko foruma, pitanje/odgovor modula, funkcije razmjene poruka za interakciju korisnika, diskusija na stranicama termina. Strana ličnog profila omogućava korisnicima pa postanu pratioci ostalih korisnika, dodaju fotografije, poveznice i postavljaju komentare na aktivnosti ostalih korisnika. TermWiki takođe omogućava kompanijama da sprovedu međunarodne kampanje o uslovnim ključnim riječima, za bolje sopstvene performanse.

Alat je razvijen od strane kompanije [CSOFT International](#). Objavljen je u maju 2010., posjeduje otvorenu, besplatnu verziju al i privatnu, profesionalnu verziju za internu upotrebu unutar organizacija<sup>[39]</sup>.



*Slika 6 TermWiki – logo*

#### *2.1.4.3.4 WikiEducator*

WikiEducator je slobodni rječnik internacionalnih online zajednica za kolaborativni razvoj učenja, odnosno materijala koji su edukatori voljni da daju kao slobodan za upotrebu, adaptaciju i dijeljenje bez ograničenja. WikiEducator je osnovan 2006. godine i podržan je od strane neprofitne organizacije *Open Education Resource (OER)*

*Foundation*. Nekoliko resursa su dostupni na WikiEducator-u: *direktna nastavna sredstva* kao što su nastavni planovi i cjelokupni kursevi kao i *sredstva za podršku učenju*, kao što su pojedinačni školski portal i prijedlozi finansiranja. WikiEducator je počeo sa ciljem da se uspostavi uspješna i održiva globalna zajednica posvećena dizajnu, razvoju i isporuci besplatnih sadržaja za učenje u realizaciji slobodne verzije obrazovnog plana i programa do 2015. Prijedlog projekta je fokusiranje na izgradnju kapaciteta za korišćenje MediWiki-ja i srodnih besplatnih softverskih tehnologija, razvoj slobodnog sadržaja za korištenje u obrazovnim institucijama ili bilo kom mjestu, olakšavanje uspostavljanja mreže zajednice i saradnje za podsticanje novih tehnologija<sup>[40]</sup>.



*Slika 7 WikiEdukator - logo*

#### *2.1.4.3.5 Chiq Chaq*

Chiq Chaq je alat za prikupljanje i organizovanje kolektivnog znanja baziranog na grupnom radu. To je veoma moćan wiki alat za objavljivanje u kombinaciji sa prihvatljivim grafičkim interfejsom. Komplikovan je za instalaciju i zahtjeva Perl<sup>8</sup> i CGI kompatibilnost za UNIX i WINDOWS web servere. Jednostavan za učenje krajnjim korisnicima i upotrebljava širom svijeta u kancelarijskom intranetu, zajednicama za oglase, sportskim fan stranicama. Besplatan softver se može preuzeti direktno sa stranice [Source Forge](#).

---

<sup>8</sup> Perl je programski jezik opće namjene. Originalni autor Perla je Larry Wall, a prva inačica pojavila se 18. prosinca 1987. godine. Perl vuče svoje korijene iz drugih jezika kao što su primjerice C, sed, awk i Unix shell. Perl je danas ne samo programski jezik već i vrlo aktivna zajednica programera i korisnika. Odlikuje ga kvalitetan repozitorij gotovih programskih rješenja, što mu je ujedno i glavna prednost u odnosu na konkurentne jezike.



*Slika 8 Chiq Chaq - logo*

#### *2.1.4.3.6 PmWiki*

PmWiki je alat baziran na kolaborativnom sistemu za stvaranje i održavanje sajtova, razvije je pod GPL (General Public Licence) licencom i kreiran je u PHP programskom jeziku. Dizajniran je da se lako instalira, podešava i održava za nekoliko različitih aplikacija. Trenutno je u upotrebi verzija [PmWiki 2.2.64](#). PmWiki je zaštićeno licencom od strane Patrick R. Michaud<sup>[41]</sup>.



*Slika 9 PmWiki - logo*

#### *2.1.4.3.7 FosWiki*

Foswiki je potpuno besplatan i slobodan za upotrebu, on je jedan od svega nekoliko wiki-ja otvorenog koda za preduzeća, koji ima zvanično udruženje za podršku pri radu i korištenju. Softver je vrlo zabavan pomoću kojeg možete raditi nevjerovatne stvari na veoma lagan način. Posjeduje veliku zajednicu programera, korisnika i profesionalnih konsultanata koji brinu za njegov razvoj i podršku.

Koristi se u mnogim velikim kompanijama već godinama. Posjeduje odličnu bezbjednost-obostranu kontrolu pristupa, kao i zaštitu od hakera. Strukturiran je, i može funkcionirati kao baza podataka, ali takođe nudi mogućnost fleksibilnog pisanja Wiki-ja. Ima preko stotinu ekstenzija i konfiguracija je moguća na mnogim nivoima, od najprostijih stvari kao što je promjena logotipa, do najkompleksnijih stvari, kao što su teme ili cijeli izgled i stil stranice. Najbitnija činjenica je da se konstantno razvija, uz veliku posvećenost

kompatibilnosti. Treba napomenuti da je Foswiki nastao iz projekta koji se nekada zvao TWiki. Njegova misija je: “Nova generacija online razvoja aplikacija i integracijskih platformi“(eng. *The Next Generation Online Application Development and Integration Platform*)<sup>[42]</sup>.



Slika 10 FosWiki - logo

#### 2.1.4.3.8 MathWiki

MathWiki je kolaborativni pristup usmjeren ka matematičkom obrazovanju. Okruženje je bazirano na principu „otvorenih udžbenika“, u kojem svi korisnici mogu pisati ili ispravljati sadržaj.

MathWiki je komponenta [StemWiki HyperLibrary](#), koja je multi-institucionalni kolaborativni poduhvat za razvoj budućih generacija otvorenog pristupa e-tekstovima za poboljšanje NTIM - Nauka, Tehnologija, Inženjerstvo i Matematika (eng. Science, Technology, Engineering, and Mathematics), na svim nivoima visokog obrazovanja<sup>[43]</sup>.



Slika 11 Math Wiki

U prethodnom poglavlju su ukratko predstavljene mogućnosti upotrebe Wiki ideje u kolaborativnom okruženju, kao i mogućnosti njene primjene u proučavanju i istraživanju. Takođe su predstavljeni osnovni alati. U ovom radu je razvijen modul koji je po svojim karakteristikama najbliži alatu *MediaWiki*. Osnovne mogućnosti ovog modula su opisane u poglavlju 3.1.1.1.

### **2.1.5 Forum**

Internet forum je usluga na Internetu koja omogućava razmjenu mišljenja među sudionicima upotrebom web preglednika. Sve poruke koje korisnik napiše i pošalje na forum vidljive su svim ostalim sudionicima foruma. To nalikuje na oglasnu ploču (*eng. message board*) na kojoj sudionici ostavljaju poruke. U načelu poruke na forumu mogu ostavljati i čitati sudionici interneta bez ograničenja. Sudionici su najčešće anonimni jer se pri slanju poruke na forumu ne moraju navesti pravi identitet. Zbog lakšeg snalaženja forum je obično podijeljen u nekoliko skupina prema temama razgovora. Jednostavnost uporabe i mogućnost anonimne rasprave o različitim temama jedan je od glavnih razloga popularnosti foruma. Poruka poslana na jedan forum vidljiva je samo na tom forumu. Forum se uglavnom sastoji od mnogobrojnih kategorija unutar kojih se nalaze teme koje otvaraju i započinju korisnici. Svaka tema ima svoju diskusiju u kojoj se sudjeluje pisanjem, odnosno objavljivanjem post-ova. Svi razgovori/diskusije ostaju sačuvani na forumu dok ih ne obriše moderator.

#### *Istorija*

Prvi internet forumi su pokrenuti još 1996. godine. Internet forumi su uglavnom strogo tematski. Prema nekom nepisanom pravilu, popularne teme foruma su: tehnologija i tehnika, informatika, kompjuterske igre, politika i sl.

#### *Članovi i korisnici*

Članovi foruma mogu objavljevati post-ove, mijenjati iste, otvarati nove teme te mijenjati svoje postavke. Na većini foruma, korisnici uz svoje ime obično imaju malu sliku koja se naziva avatar. Takođe mogu primati i slati privatne poruke ostalim korisnicima. U određenim slučajevima daje im se mogućnost brisanja vlastite teme ili post-a. Korisnici često koriste i potpise, najčešće riječi ili slike koje se pojavljuju na kraju svakog posta koji objave.

## *Organizacija foruma*

Forumi mogu biti različitih stilova. Uobičajeni forumi sadrže hiperveze koje vode do podforumu. U svakom podforumu se nalazi mjesto za objavljivanje post-ova, a nove se teme nalaze na vrhu, sa posebnim upozorenjem o njihovoj aktivaciji. Isto tako, mnogi forumi na vrhu liste tema sadrže važne obavijesti. One teme u kojima se često odgovara, smješteni su pri vrhu, dok se oni u kojima se nalazi tek pokoji odgovor nalaze na dnu.

Drugi modul *Pitanja i Odgovori*, implementiran u praktičnom dijelu FITCKM sistema, je razvijen na principu foruma u nešto izmijenjenom obliku. Logika je vrlo slična, s tim da je za potrebe teme rada, omogućeno ocjenjivanje postavljenih pitanja i odgovora. Prilikom kreiranja pitanja, logirani korisnik je dužan da izabere kategoriju i podkategoriju kojoj pitanje pripada, kao i da unese odgovarajuće oznake. Sve ocjene, oznake i korisnici su uključeni u kalkulaciju preporuke, te se logiranom korisniku u vidu hiperveza preporučuju značajni i zanimljivi sadržaji, kao i slični korisnici. Takođe, korisnicima je omogućeno i napredno pretraživanje cijelog teksta (*eng. full-text searches*). Ove implementacije su integrisane u cilju poboljšanja udobnosti kolaborativnog okruženja.

## **2.2 Inteligentni programski agenti**

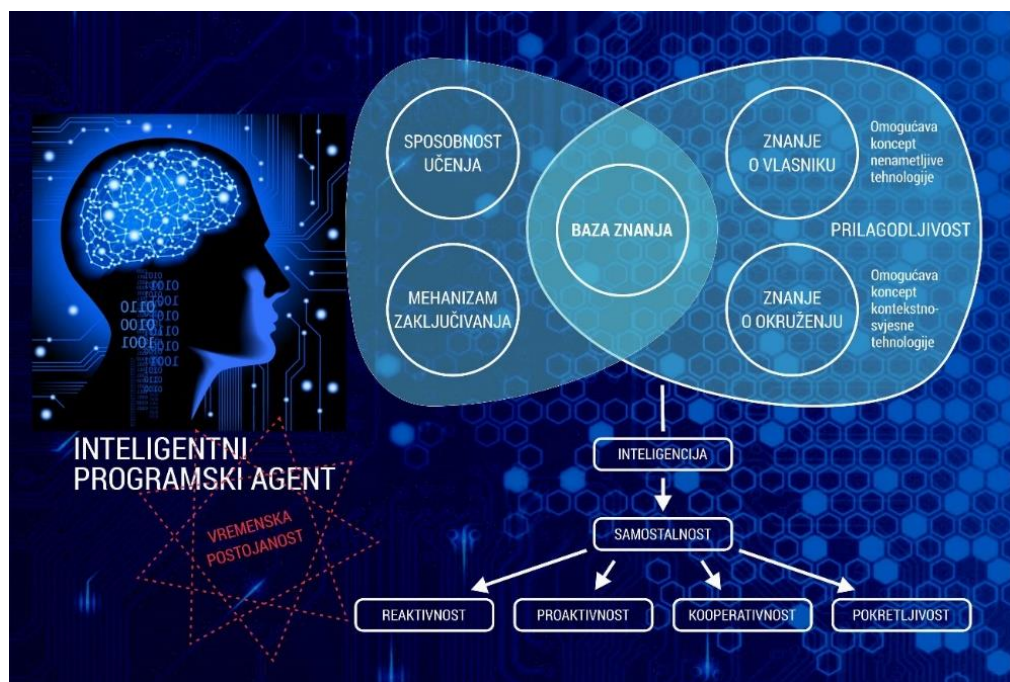
Evolucija IT industrije zahtijeva od modernih aplikacija ne samo da odgovaraju na zahtjeve, nego da i inteligentno učestvuju u aktivnostima u svom okruženju, te se neprestano prilagođavaju trenutnom stanju svog okruženja i na taj način aktivno traže način kako da što bolje ostvare interese entiteta kojeg zastupaju<sup>[44],[45],[46],[47],[48]</sup>. Kompjuterski program koji ima opisane karakteristike i koji u interakciji sa okruženjem, ima sposobnost da fleksibilno i samostalno reaguje u skladu sa ciljevima koji su mu postavljeni naziva se programski agent. On je ustvari program koji djeluje u ime svog vlasnika, obavljajući pri om složene informacijske i komunikacijske poslove u okruženju<sup>[49],[50]</sup>. Programski agenti omogućavaju autonomno izvršavanje i koordiniranje različitih procesa<sup>[51],[52],[53],[54]</sup>, te na taj način stvaraju dodanu vrijednost za svog vlasnika<sup>[55],[56]</sup>. Posjeduju sposobnost da fleksibilno, samostalno i bez intervencije korisnika izvršavaju postavljeni zadatak, te krajnjeg korisnika izvještavaju o završetku



zadatka ili samoj pojavi događaja koji se očekuje. Agent je u interakciji s okolinom kako bi na što precizniji način izvršio postavljeni zadatak.

### 2.2.1 Struktura inteligentnog agenta

Jednostavni agent može biti matematički definisan kao funkcija agenta, koji mapira svaku moguću sekvencu percepcije prema mogućoj akciji, koju agent može izvršiti ili prema koeficijentu, povratnoj vezi, funkciji ili konstanti koja utiče na eventualnu akciju. Funkcija agenta je apstraktni koncept, koji uključuje različite principe donošenja odluka kao što je proračun individualnih mogućnosti, dedukcije iznad logičkih pravila i slično. Na *Slika 12* Inteligentni programski agent predstavljene su osnovne karakteristike inteligentnih programskih agenata i njihovi odnosi<sup>[44],[57],[58],[59],[60],[61]</sup>.



*Slika 12 Inteligentni programski agent*

*Inteligencija:* Agent mora posjedovati inteligenciju<sup>[62]</sup> baziranu na vlastitoj bazi znanja, mehanizmima zaključivanja i sposobnosti učenja. Inteligencija agenta je preduslov za sva ostala njegova obilježja. Zavisno od namjene agenta, razlikuje se i znanje koje sadrži njegova baza znanja, ali uopšteno se to znanje može podijeliti na dva dijela – znanje o vlasniku i znanje o okruženju.

*Prilagodljivost (adaptivnost):* Veoma bitno je da agent neprestano osvježava bazu znanja, odnosno da ne sadrži statično znanje<sup>[63]</sup>. Na taj se način agent pokušava, na najbolji mogući način, prilagoditi potrebama svoga vlasnika, što mu omogućava da ga efikasno predstavlja u okruženju nove generacije i na taj način ostvari koncept nenametljive tehnologije<sup>[64],[65],[66]</sup>. Agent takođe, na osnovu aktuelnih događanja u njegovom okruženju, neprestano osvježava znanje o svom okruženju i na taj način ostvaruje koncept kontekstno-svjesne tehnologije<sup>[67],[68],[69],[70],[71]</sup>. Svjesnost o kontekstu (*eng. Context-awareness*)<sup>[72],[73]</sup> je sposobnost agenta da prilagođava akcije trenutnom stanju u kojem se nalazi njegov vlasnik i trenutnoj situaciji u kojoj se nalazi njegovo okruženje<sup>[74],[75],[76]</sup>. Agent koji nema izražene osobine adaptivnosti pretražuje unaprijed definisane uzorke ponašanja u svrhu pronalaska optimalnog odgovora na neki poticaj iz okruženja. Za razliku od toga adaptivni agenti, kroz svoje mehanizme zaključivanja, koristeći sposobnost učenja, mogu kreirati nove uzorke ponašanja koji im osiguravaju bolje prilagođavanje najnovijim događajima u svom okruženju ili novim potrebama svoga vlasnika<sup>[77]</sup>.

*Samostalnost (autonomnost):* Agent izvršava poslove u potpunosti samostalno, bez bilo kakvih intervencija svog vlasnika, što ga čini „nevidljivim slugom“<sup>[66]</sup>. Autonomnost agenta pretpostavlja da on ima sposobnost kontrolisati svoje aktivnosti i osigurati resurse za njihovo izvođenje. Samostalnost agenta je omogućena njegovom inteligencijom i najbolji primjer za to su njegove osobine: *reaktivnost, proaktivnost i kooperativnost*.

*Reaktivnost:* Agent reaguje na uticaje iz okruženja u kojem djeluje, zasnivajući svoje djelovanje na interakciji s okruženjem, ili se umjesto toga oslanja na vlastiti model okruženja u kojem djeluje.

*Proaktivnost:* Agent ne reaguje samo na poticaje iz svog okruženja, već poduzima inicijative u skladu sa preuzetim zadacima, uz što je moguće bolje definisan cilj.

*Kooperativnost:* Agent sarađuje s ostalim agentima iz svog okruženja i na temelju te saradnje poduzima akcije koje omogućavaju efikasnije rješavanje postavljenih zadataka<sup>[78],[79],[80]</sup>. Upotreba tehnologija semantičkog web-a za definisanje sadržaja u agentskoj komunikaciji omogućava ne samo da se agenti, koji su razvijani potpuno

nezavisno jedan od drugog, međusobno razumiju, već i da komuniciraju na semantičkom nivou, slično kao i ljudi<sup>[81],[82]</sup>.

*Pokretljivost:* Agent se može kretati između različitih korisničkih uređaja i drugih čvorova u mreži, te na taj način fizički migrirati unutar mrežne infrastrukture<sup>[83]</sup>. Da bi agent posjedovao osobinu pokretljivosti moraju se osigurati dodatni sistemski uslovi (postojanje agentske platforme na svakom čvoru domaćinu), jer konvencionalni programski sistemi ne dopuštaju kretanje programa. Osobina pokretljivosti najčešće nije neophodna da bi agent mogao obavljati svoje zadaće, ali može biti korisna u smislu oslobađanja ograničenih performansi na korisničkim uređajima, kao i zbog činjenice da se na taj način oslobađaju ukupni mrežni resursi. Ipak, ako je pokretljivost i implementirana u agentu, ona može biti više ili manje „inteligentna“, zavisno od toga da li je agentu unaprijed zadano koje čvorove u mreži mora posjetiti, ili on samostalno određuje gdje i kako će se kretati kroz mrežu.

*Vremenska postojanost:* Agent posjeduje osobinu perzistentnosti identiteta i stanja kroz duže vremenske periode. Ako agent posjeduje i osobinu pokretljivosti, to znači da, prilikom premještanja s jednog čvora na drugi, mora imati sposobnost očuvanja svog programskog koda kao i svih svojih stanja i parametara.

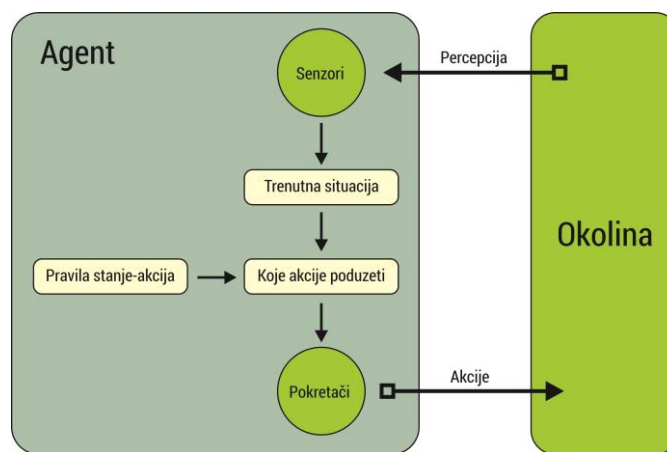
### **2.2.2 Vrste inteligentnih agenata**

Autori (*Russell i Norvig, 2003*) su grupirali agente u pet klasa na osnovu stepena percepcijske inteligencije i mogućnosti<sup>[84]</sup>:

- *jednostavni refleksivni agenti*
- *refleksivni agenti temeljeni na modelu*
- *agenti temeljeni na cilju*
- *agenti temeljeni na korisnosti*
- *učeći agenti*

### 2.2.2.1 Jednostavni refleksivni agenti

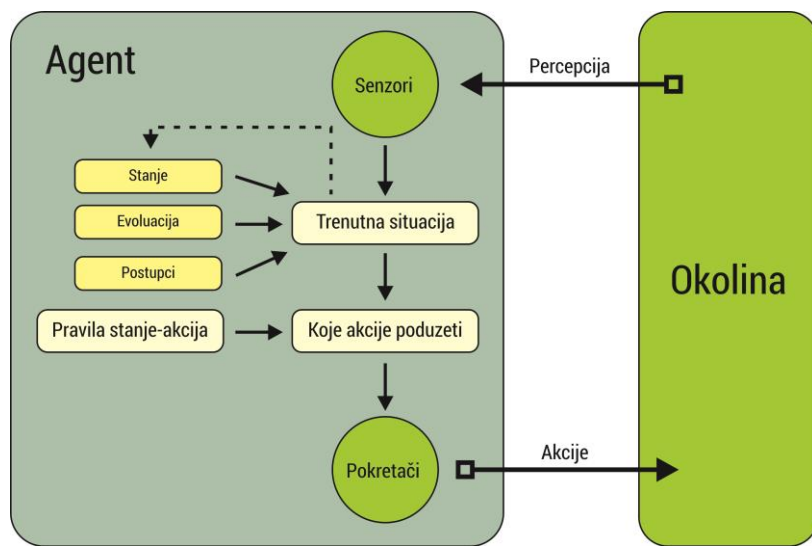
Jednostavni refleksivni agenti djeluju samo na osnovu trenutne percepcije, ignorišući ostatak percepcijske istorije. Funkcija agenta je bazirana na pravilu *stanje-akcija*: ako se zadovoljava stanje (uslov), tada slijedi akcija. Ta funkcija agenta uspijeva samo kada je okruženje u potpunosti primjetljivo. Neki refleksivni agenti mogu sadržavati informacije o njihovom trenutnom stanju što im omogućava zanemarivanje uslova čiji su pokretači već pokrenuti. Beskonačne petlje su često neizbježne za jednostavne refleksivne agente koji djeluju u djelimično primjetljivim okruženjima (*Slika 13* Jednostavni refleksivni agenti).



*Slika 13 Jednostavni refleksivni agenti*

### 2.2.2.2 Refleksivni agenti bazirani na modelu

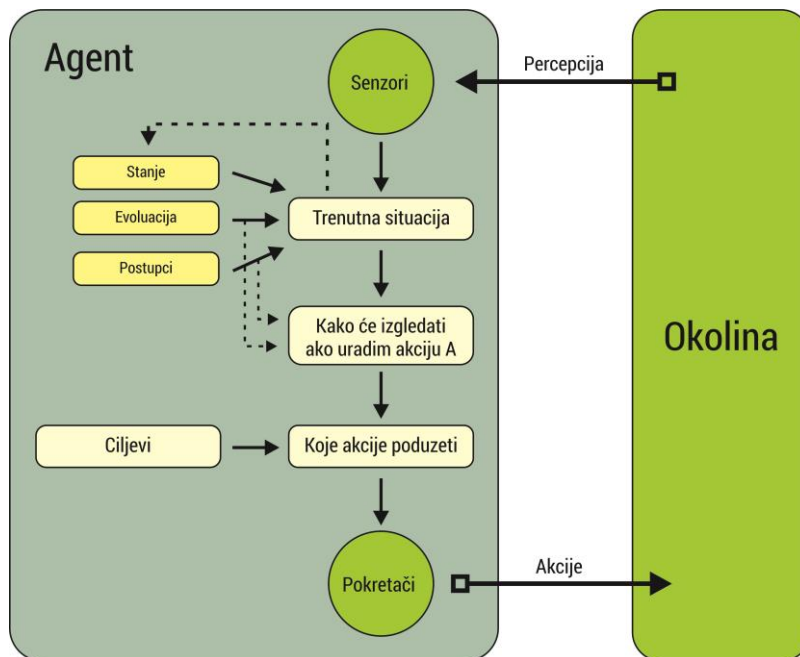
Agenti bazirani na modelu se mogu nositi sa djelimično primjetljivim okruženjem. Njihovo trenutno stanje je pohranjeno unutar agenta, održavajući neku vrstu strukture koja opisuje dio svijeta koji ne može biti viđen. To znanje o trenutnoj situaciji "kako svijet radi" je nazvano model svijeta, što i sugerise naziv "agent baziran na modelu". Refleksivni agent baziran na modelu treba održavati neku vrstu internog modela koji zavisi o istoriji percepcije te odražavati bar dio neprimjetnog aspekta trenutnog stanja. Tada bira akciju na isti način kao i jednostavni refleksivni agent (*Slika 14*).



Slika 14 Refleksivni agenti bazirani na modelu

### 2.2.2.3 Agenti bazirani na cilju

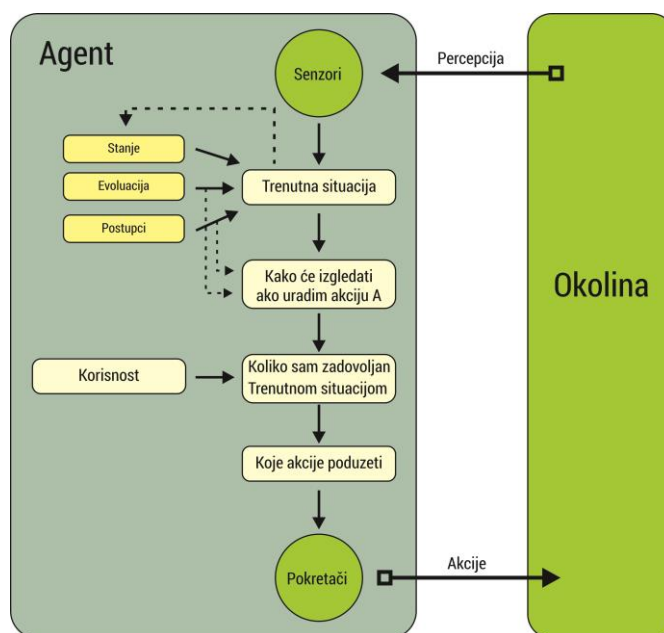
Agenti bazirani na cilju proširuju mogućnosti agenata baziranih na modelu, korištenjem informacije o cilju. Informacija o cilju opisuje situaciju koja je poželjna. To omogućava agentu način izbora jedne, od mnogobrojnih mogućnosti, birajući onu koja ispunjava ciljno stanje (Slika 15).



Slika 15 Agenti bazirani na cilju

#### 2.2.2.4 Agenti bazirani na korisnosti

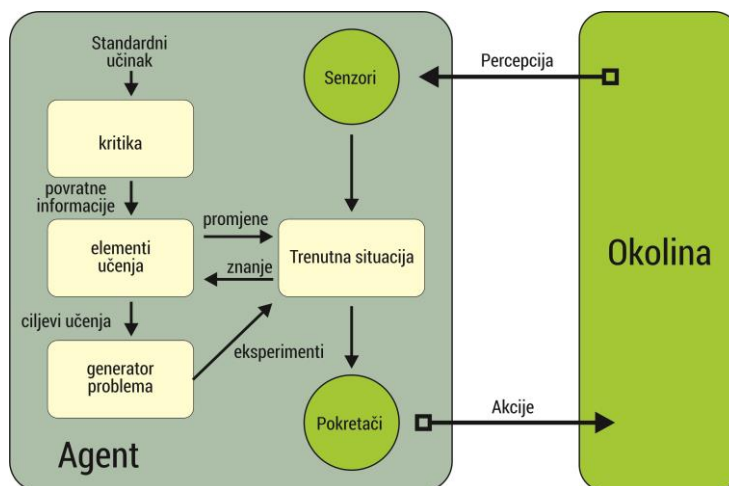
Agenti bazirani na cilju samo poznaju razliku između ciljnog i „ne-ciljnog“ stanja. Moguće je definisati mjeru kojom se mjeri poželjnost određenog stanja. Ta mjera se može dobiti kroz uporabu funkcije korisnosti. Mjere opšti performansi je omogućeno kroz poređenje različitih stanja okruženja, na način da se mjeri „usrećenost“ agenta. Ustvari, pojam "korisnost" opisuje koliko je agent "sretan". Racionalan agent baziran na korisnosti bira akciju koja maksimizira očekivanu korisnost ishoda akcije (Slika 16).



Slika 16 Agenti bazirani na korisnosti

#### 2.2.2.5 Učeći agenti

Učenje ima prednost koja omogućava agentu da inicijalno djeluje u nepoznatoj okolini i da postane kompetentan u odnosu na svoje inicijalno znanje. Najvažnije je znati razliku između "elementa učenja", koji je odgovoran za poboljšanje, i "elementa performanse" koji je odgovoran za izbor eksternih akcija. Takođe, postoji komponenta učećeg agenta koja se naziva "generator problema". Ona je odgovorna za predlaganje akcija koje će voditi prema novim i informativnim iskustvima (Slika 17).



Slika 17 Učeći agenti

#### 2.2.2.6 Ostale klase inteligentnih agenata

Postoje još neke vrste inteligentnih agenata:

- *Odlučujući agenti* (pomažu u donošenju odluka)
- *Ulazni agenti* (procesiraju ulaze, senzori)
- *Procesirajući agenti* (rješavaju probleme kao što su prepoznavanje govora)
- *Prostorni agenti* (vezani su za fizički, realni svijet)
- *Svjetski agenti* (kombinacija ostalih klasa agenata omogućujući autonomno ponašanje)
- *Agenti vjerovatnoće*
- *Fizički agenti* (Entitet koji percipira kroz senzore i djeluje kroz pokretače)
- *Vremenski agenti*

U kolaborativnim okruženjima za upravljanje znanjem, inteligentni programski agenti se mogu koristiti u interakciji sistema i krajnjih korisnika. U FITCKM sistemu programski agenti imaju ulogu da na osnovu događaja, vezanih za korisničke akcije, izvršavaju kompleksne kalkulacije. Rezultat kalkulacija je dijelom prikazan korisniku a dijelom zapisan u bazu podataka u ulozi informacija koje pomažu agentu da „razumije“ novonastalo stanje okruženja, odnosno da usvoji novo znanje, koje će opet predstavljati podlogu za buduća znanja. Više o konkretnoj ulozi programskog agenta u FITCKM sistemu se nalazi u poglavlju 3.1.3.

### 2.3 Sistemi preporuke

Jedan od standardnih problema koji prati razvoj ICT-a, je rapidno povećanje količine informacija, odnosno problem preopterećenja informacijama (*eng. information overload*). Okruženja bogata informacijama, koja ne koriste metode za adekvatno upravljanje njima, mogu biti isto neefikasna kao i okruženja sa nedostatkom informacija. Kako manjak informacija predstavlja očigledan problem, višak dovodi do preopterećenja, tako da se relevantne informacije utapaju u irelevantnoj buci. Najčešći razlog tzv. konfuzije, u okruženjima koja obiluju informacijama, je činjenica da su informacije često vrlo slične, bez opisanih specifičnosti. Optimalnu količinu informacija, u smislu fizičkog brisanja ili dodavanja, je nemoguće postići i kontrolisati, jer takvo nešto zahtijeva kompleksne metode koje bi negativno uticale na performanse sistema. Takođe, nemogućom se čini i odluka za brisanje određene informacije, jer svaka informacija može biti potencijalno relevantna u određenom kontekstu.

S druge strane, ako se velika količina informacija posmatra kao bogatstvo koje treba adekvatno iskoristiti, javlja se potreba za izdvajanjem, filtriranjem, rangiranjem i sl. Osnovni cilj je preporuka korisnih, značajnih i interesantnih sadržaja koji mogu biti različiti: članci, knjige, muzički albumi, igre, filmovi, vijesti i sl. U radu će se za različite vrste sadržaja koristiti termin „resurs“, što je u skladu sa terminologijom koja se koristi u akademskoj zajednici. Osnovni zadatak sistema za preporuku je da identifikuje resurse potencijalno „važne“ korisniku. U procesu identifikacije sistem bi trebao predvidjeti „važnosti“ resursa, uporediti ih, te korisniku preporučiti „najvažnije“ rezultate. Za takve resurse u radu će se koristiti termin „relevantan“.

Pretpostavka od koje treba krenuti pri razvoju sistema preporuke je ljudska sposobnost u donošenju promišljenih odluka na osnovu dostupnih informacija<sup>[85]</sup>. Slijedeći tu ideju, kao pomoć u neprestanoj borbi čovjeka sa suvišnim informacijama, počeo je razvoj različitih algoritama u području umjetne inteligencije (*eng. artificial intelligence, AI*). U okvirima sistema preporuke, oni imaju veoma važnu ulogu u poboljšavaju performansi pristupa informacijama. Neki autori ovakve koncepte nazivaju: „nerazumna efikasnost podataka“<sup>[86]</sup>. Iako kontradiktorno zvuči, upravo višak informacija, često je glavni adut



za efikasnije izvršavanje pojedinih algoritama. Drugi autori pokazuju kako se uobičajeni algoritmi u AI mogu znatno poboljšati, dajući im što više podataka za rad<sup>[87]</sup>.

Sistemi za preporuku predstavljaju tehnike za modeliranje objekta korisnika, na osnovu kojeg se vrši procjena relevantnosti resursa. Ovi sistemi koriste podatke kao što su log-ovi pretrage, ocjene sličnih korisnika, društvene veze i sve značajne podatke koji se mogu iskoristiti u predviđanju nepoznatih relevantnosti. Gdje god postoji potreba za individualnim preporukama, prvenstveno u web okruženjima, sistemi preporuke obavljaju vrlo naporan posao iza scene.

Moderni pristupi modeliranja sistema preporuke često koriste „*nerazumnu efikasnost podataka*“, tako što kombinuju više sistema za preporuku, gdje svaki od njih predviđa relevantnost na različit način. Obzirom na različite afinitete korisnika i različitost resursa, ovi sistemi prilikom predviđanja koriste veoma složene metode, koje se oslanjaju na mnoštvo dokaza.

Ako sistem može precizno predvidjeti kako će korisnici reagovati na određene resurse, učinjen je ogroman korak prema rješavanju problema preopterećenosti informacijama.

### **2.3.1 Formalna definicija problema generisanja preporuke**

Problem generisanja preporuka može se formalno zapisati<sup>[88],[89]</sup>:

Neka je  $U = (u_1, u_2, \dots, u_m)$  skup korisnika sistema,  $I = (i_1, i_2, \dots, i_n)$  skup svih resursa koji se mogu preporučiti, a  $R$  potpuno uređeni skup.

Neka je  $g: U \times I \rightarrow R$  funkcija korisnosti (*eng. utility function*).

Vrijednost funkcije  $g(u_j, i_k)$  predstavlja korisnost resursa  $i_k$  za korisnika  $u_j$ .

Cilj sistema je da za svakog korisnika  $u_j \in U$  odrediti resurs  $i^{max, u_j} \in I$  za koji je vrijednost funkcije  $g$  maksimalna:

$$\forall u_j \in U, i^{max, u_j} = \arg \max_{i_k \in I} g(u_j, i_k) \quad (1)$$

Korisnost nekog resursa se obično predstavlja brojčanom ocjenom koju određuje direktno korisnik, ili se ona izračunava na osnovu prethodno definisanih pravila sistema. Drugim riječima, ocjena ne mora isključivo predstavljati vrijednost koju su korisnici dodijelili resursu, već se još može izraziti i brojem pregleda svidanja, komentara, kupljenih primjeraka i sl.

Ocjena predstavlja mjeru, odnosno procjenu koliko je određeni resurs relevantan za korisnika. Za provođenje što preciznijeg ocjenjivanja potrebno je prikupiti što više značajnih informacija o sudionicima. Zato je neophodno skupom karakteristika opisati svakog korisnika  $u_j \in U$  i svaki resurs  $i_k \in I$ . Prva faza u procesu generiranja preporuke je izgradnja modela korisnika, koji služi za čuvanje prikupljenih informacija o korisniku. Podaci se mogu prikupljati na dva načina: *eksplicitno* (npr. pomoću upitnika) ili *implicitno*, odnosno automatski (npr. na osnovu interakcije korisnika sa sistemom). U narednoj fazi podaci iz tog modela upoređuju se s karakteristikama resursa i/ili s podacima ostalih korisnika, kako bi se generisale odgovarajuće preporuke[89]. Osnovni problem sistema za preporuke je što u većini slučajeva funkcija  $g$  nije definisana na cijelom skupu  $U \times I$ , već samo na njegovom podskupu<sup>[90]</sup>. Inicijalno ocjena je poznata samo za one resurse koje su korisnici prethodno ocijenili, dok se za nepoznate vrijednosti ona utvrđuje na osnovu predviđanja. Prema tome, jedan od ciljeva sistema preporuke je što preciznije predvidjeti ocjene za neocijenjene resurse na osnovu poznatih ocjena. Nakon predviđanja vrijednosti za sve uređene parove skupa  $K \times S$ , preporuka za određenog korisnika se određuje prema formuli (1). U zavisnosti od načina prezentacije preporučenih resursa, korisniku se može ponuditi jedan ili više rezultata (npr. lista od  $n$  preporuka sortiranih prema visini ocjene)<sup>[91]</sup>.

Većina postojećih sistema preporuke obično koristi samo jedan kriterij za predstavljanje relevantnosti nekog resursa za korisnika u skupu  $U \times I$ , dok se u novijim istraživanjima preporučuje korištenje više kriterija kako bi se odredile preciznije preporuke. U tom slučaju preporuka se generiše na osnovu proširenog skupa podataka o korisniku i resursima koji se preporučuju. Funkcija  $g$  se tada može definisati na sljedeći način:  $g: U \times I \rightarrow R_0 \times R_1 \times \dots \times R_l$ , gdje je  $R_0$  skup mogućih ukupnih ocjena (relevantnosti) resursa, a skupovi  $R_c$ , skupovi mogućih ocjena prema kriterijima  $c = 1, \dots, l$ . Za razliku

od ukupne ocjene iz skupa  $R_0$ , ocjene po kriterijima predstavljaju podatak o tome zašto se određeni resurs korisniku sviđa, a ne samo koliko mu se sviđa. Takođe, na temelju ocjena po kriterijima moguće je preciznije odrediti sličnosti među korisnicima sistema<sup>[90]</sup>.

### 2.3.2 Klasifikacija tradicionalnih sistema preporuke

Postoji mnogo načina za predviđanje relevantnosti resursa korisniku. Istraživači u ovoj oblasti otkrili su različite obrasce za preporuku, koji se u posljednje vrijeme i praktično primjenjuju, te se u ovom radu posmatraju kao tradicionalni.

Obzirom na poznate tehnike i algoritme koji su se do sada našli u upotrebi pri procjeni relevantnosti resursa, tradicionalni sistemi preporuke mogu biti:

- *sistemi preporuke zasnovane na sadržaju (eng. content-based, CB)*
- *kolaborativno filtriranje (eng. collaborative filtering, CF)*
- *hibridni (mješoviti) sistemi preporuke*

#### 2.3.2.1 Sistemi preporuke zasnovane na sadržaju

CB metode su bazirane na pretpostavci da će se korisniku svidjeti resursi sa sličnim karakteristikama onim resursima koji su mu se svidjeli u prošlosti. Sistem na osnovu prethodno ocijenjenih ili odabranih resursa pokušava "razumjeti" šta korisnik preferira. Vrijednost funkcije  $g(u_j, i_k)$  kojom se predstavlja relevantnost resursa  $i_k \in I$  za korisnika  $u_j \in U$ , u tom slučaju se procjenjuje na temelju poznatih vrijednosti  $g(u_j, i_s)$  za korisnika  $u_i$ , pri čemu je resurs  $i_s \in I$  "sličan" resursu  $i_k \in I$ . U literaturi se za ovu vrstu sistema koristi i naziv *individualni sistemi preporuke*<sup>[92]</sup>.

Prilikom generisanja preporuka, preferencije korisnika se utvrđuju na osnovu karakteristika resursa koje je odabrao, odnosno ocijenio. Ti se podaci upoređuju s karakteristikama svih elemenata skupa  $I$  te se izdvajaju oni resursi koji se po karakteristikama u najvećoj mjeri podudaraju s preferencijama korisnika. Na taj se način značajno smanjuje broj potencijalnih resursa za preporuku određenom korisniku.

Opis resursa  $i_n \in I$ , u oznaci  $Content(i_n)$  se obično predstavlja vektorom s realnim vrijednostima, odnosno težinama, pri čemu se svakom komponentom vektora izražava mjera "važnosti" (*eng. importance*) neke karakteristike u opisu<sup>[89]</sup>:

$$Content(i_n) = i_n = (i_{n,1}, i_{n,2}, \dots, i_{n,K}) \in R^k \quad (2)$$

Na primjer, opis može sadržavati ključne riječi (*eng. keywords*), odnosno oznake (*eng. tags*), kojima se opisuje određeni resurs. Analogno se predstavljaju preferencije korisnika  $u_m \in U$  vezane za resurs (*eng. content-based preferences*), u oznaci  $ContentBasedUserProfile(u_m)$ , s težinskim vrijednostima za resurse koje opisuju sklonosti, interese i potrebe korisnika:

$$ContentBasedUserProfile(u_m) = u_m = (i_{m,1}, i_{m,2}, \dots, i_{m,k}) \in R^K \quad (3)$$

Vrijednost korisnosti resursa  $i_n \in I$  za korisnika  $u_m \in U$  određuje se funkcijom *score* čija se vrijednost određuje na temelju kombinacije komponenti opisa resursa  $i_n$  i profila korisnika  $u_m$ :

$$g(u_m, i_n) = score(ContentBasedUserProfile(u_m), Content(i_n)) \in R \quad (4)$$

Tehnike za određivanje vrijednosti funkcije *score* navedenog izraza klasifikuju se u heurističke (*eng. heuristic-based*) i tehnike zasnovane na modelu (*eng. model-based*). Kod heurističkih tehnika se predviđanje vrijednosti određuje pomoću heurističkih formula, baziranih na metodama za pretraživanje informacija (*eng. information retrieval*). Primjeri takvih metoda su kosinusova sličnost, euklidska udaljenost, *Pearson-Spearmanova* korelacija. Tehnike je bazirane na modelu koriste znanje iz podataka, koristeći mašinsko učenje i statističke modele poput *Bayesovih* i neuronskih mreža, klaster algoritama i stabala odluke.

Postoje ograničenja koja predstavljaju problem za obje skupine tehnika<sup>[88][89]</sup>. Jedno od njih je ograničena mogućnost automatske analize sadržaja. Uspješnost rada sistema preporuke zasnovanih na sadržaju zavisi o dostupnim podacima kojima se opisuju resursi. Iz tog razloga je potrebno omogućiti automatsko opisivanje resursa ili karakteristike resursima treba dodijeliti ručno. U većini okruženja, automatsko opisivanje resursa je vrlo

složeno (na primjer kod opisa multimedijalnih sadržaja), dok je ručni opis vremenski zahtjevan.

Ograničenje CB metoda, za nove korisnike, predstavlja i takozvani problem „hladnog starta“ (*eng. cold-start problem*). Korisnik mora ocijeniti dovoljan broj resursa, kako bi mu sistem mogao predložiti odgovarajuće preporuke. Zbog načina određivanja vrijednosti funkcije korisnosti (gdje se upoređuju karakteristike pojedinog resursa s profilom korisnika) i preporuke samo onih sadržaja za koje je ta vrijednost maksimalna ili vrlo visoka, događa se da se korisniku nikada ne preporučuju resursi sadržajno različiti od resursa koje je korisnik pregledavao ranije (*eng. content overspecialization problem*), nego uglavnom međusobno vrlo slični resursi (*eng. non diversity problem*)<sup>[89]</sup>.

#### 2.3.2.2 Kolaborativno filtriranje

CF metode imaju zadatak predviđanja korisnosti pojedinog resursa, za nekog korisnika, na osnovu vrednovanja resursa od strane ostalih korisnika sistema.

Vrijednost funkcije  $g(u_j, i_k)$  kojom se predstavlja korisnost resursa  $i_k \in I$  za korisnika  $u_j \in U$  se u tom slučaju procjenjuje na osnovu poznatih vrijednosti  $g(u_s, i_k)$  za resurs  $i_k \in I$  pri čemu je korisnik  $u_s \in U$  "sličan" korisniku  $u_k \in U$ . Kako se pri generisanju preporuka u obzir uzimaju podaci o ostalim korisnicima, za ovu vrstu sistema u literaturi se koristi i naziv *kolaborativni sistemi za preporuke*, bez obzira što se i u ovom slučaju preporuke generišu za individualne korisnike<sup>[92]</sup>.

Kod CF metoda, ocjene koje su korisnici dodijelili resursima koriste se kao aproksimativna reprezentacija njihovih interesa i potreba.

Za razliku od CB metoda, CF metode ne manipulišu podacima iz resursa<sup>[93]</sup>, već se ocjene dodijeljene od strane ciljnog korisnika upoređuju s ocjenama koje su dodijelili preostali korisnici sistema, te se određuje skup njemu "najbližih komšija". U skladu s tim, uz model ciljnog korisnika važna je baza podataka s modelima ostalih korisnika sistema. Ciljnom korisniku će među resursima koje još nije ocijenio biti preporučeni oni koje su visoko ocijenili njemu "slični" korisnici iz skupa "najbližih komšija". Pristupi CF metoda se razlikuju upravo po načinu na koji se određuje sličnost među korisnicima.

U nastavku je prikazana definicija modela (profila) korisnika, sa oznakom  $CollaborativeUserProfile(u_m)$ , kao vektor čije su komponente ocjene koje je korisnik dodijelio dostupnim resursima<sup>[89]</sup> :

$$CollaborativeUserProfile(u_m) = r_m = (r_{m,1}, r_{m,2}, \dots, r_{m,k}) \in R^N \quad (5)$$

Ocjene koje su određenom resursu dodijelili ostali korisnici predstavljaju se vektorom  $Ratings(i_n)$ :

$$Ratings(i_n) = r_n = (r_{1,n}, r_{2,n}, \dots, r_{M,n}) \in R^M \quad (6)$$

U slučaju da korisnik  $u_i$  nije ocijenio sadržaj  $i_j$  za komponentu  $r_{i,j}$  u oba gore navedena vektora vrijedi  $r_{i,j} = \emptyset$ . Vrijednost relevantnosti resursa  $i_n \in I$  za korisnika  $u_m \in U$  određuje se funkcijom  $score$  koja u proračunu kombinuje komponente profila korisnika  $u_m$  i ocjene za  $i_n$ :

$$g(u_m, i_n) = score(CollaborativeUserProfile(u_m), Ratings(i_n)) \in R \quad (7)$$

Tehnike za kolaborativno filtriranje dijele se na osnovu različitih formulacija za navedeni izraz u dva osnovna pristupa<sup>[89]</sup> :

- 1) pristup baziran na filtriranju po korisnicima (eng. user-based)
- 2) pristup baziran na filtriranju po resursima (eng. item-based)

Kod pristupa baziranom na filtriranju po korisnicima, vrednovanje resursa ciljnog korisnika se upoređuje s vrednovanjem ostalih korisnika kako bi se odredila grupa sličnih korisnika. Preporučuju se resursi koje su slični korisnici ocijenili najvišim ocjenama. Na taj način se postiže da se ciljnom korisniku preporuče resursi koje preferiraju njemu slični korisnici.

S druge strane, kod pristupa baziranog na filtriranju po resursima, za resurse koje je ciljnik korisnik pregledavao ili ocijenio pronalaze se i preporučuju slični sadržaji. Prilikom određivanja sličnosti među resursima upoređuje se vrednovanje ciljnog korisnika s vrednovanjem ostalih korisnika. Dakle, ciljnom korisniku se preporučuju sadržaji koji su ostali korisnici slično vrednovali.

U oba navedena slučaja, skup sadržaja za preporuku ograničen je na one resurse koje su ocijenili ostali korisnici<sup>[89]</sup>. Preporuke su nezavisne o sadržaju, jer se prilikom generisanja preporuka koriste ocjene ostalih korisnika, a ne karakteristike sadržaja koji se preporučuju. Takav način rada omogućuje da se korisniku preporuče resursi koji se sadržajno razlikuju od onih koje je korisnik pregledavao ranije. Sistemi za kolaborativno filtriranje rješavaju samo neke od problema koji se javljaju kod sistema za preporuke zasnovanih na sadržaju. Ipak, isključiva primjena tehnika kolaborativnog filtriranja ima određene nedostatke.

Uspješnost tehnika CF zavisi od poređenja ocijenjenih resursa, te se, obično u okruženjima sa bogatim resursima koje sistem može preporučiti, javlja problem malog broja ocjena (*eng. sparse rating problem*). Resursi koji su vrednovani od strane malog broja korisnika tako neće gotovo nikada biti preporučeni, nezavisno od visine ocjena. Uz to, ukoliko se skup sadržaja za preporuke često mijenja, ranije dodijeljene ocjene neće koristiti novim korisnicima.

I kod CF metoda se javlja problem „hladnog starta“ za nove korisnike. Kako na početku rada sa sistemom nema ocijenjenih resursa od strane korisnika, nije moguće generisati odgovarajuće preporuke koristeći kolaborativno filtriranje. Isti se problem javlja i u slučaju dodavanja novih resursa u sistem. Sve dok nove resurse ne ocijeni dovoljan broj korisnika, sistem ga neće preporučivati. Taj problem je posebno izražen u sistemima u kojima postoji potreba za konstantno i često dodavanje resursa (npr., na portalima s vijestima).

CF metoda su se pokazale uspješne za one korisnike koji se mogu svrstati u skupove "najbližih komšija" (klastere). Korisnici čije se preferencije razlikuju u poređenju s ostatkom korisničke mreže i za koje je teško naći slične korisnike (*eng. gray sheep problem*) ne mogu od ovakvih sistema očekivati dobre rezultate. Uz navedeno, treba spomenuti i problem raznovrsnosti sadržaja (*eng. non diversity problem*). Kako je kod CF znanje sistema o resursima bazirano na tome šta korisnici preferiraju, preporuke su gotovo u potpunosti zavisne o tome koje sadržaje korisnici ocjenjuju, pa se u tom slučaju ne preporučuju najrelevantniji, nego samo najpopularniji resursi<sup>[89]</sup>.

### 2.3.2.3 Hibridni (mješoviti) sistemi preporuke

Mješoviti ili hibridni sistemi preporuke kombinuju CB i CF modele i nastoje prevazići ograničenja pojedinih pristupa uzimajući u obzir karakteristike i vrednovanje resursa od strane korisnika. Pristupi implementaciji mogu se klasifikovati zavisno od načina kombinovanja različitih tehnika za generisanje preporuka. Neke od mogućnosti su da se na osnovu određenog kriterija vrši izmjena korištenih tehnika u nekoliko faza, da se primjenom jedne tehnike dobiva model koji služi kao ulaz za sljedeću tehniku ili da se zajedno prezentiraju preporuke dobivene na osnovu različitih tehnika <sup>[89]</sup>.

Iako se mješovitim pristupima pokušavaju otkloniti nedostaci CB i CF modela, kod postojećih sistema za preporuke se mogu identifikovati i neki opšti nedostaci. Tako većina sistema je bazirana na ograničenom skupu podataka o korisnicima i resursima koje preporučuje, ne uzimajući pritom informacije koje se mogu dobiti na osnovu praćenja korisnika, tehnikama kao što je dubinska analiza podataka (*eng. data mining*). Modeli korisnika su vrlo jednostavni i ne koriste naprednije tehnike modeliranja korisnika. Sistemi za kolaborativno filtriranje se često prilikom generiranja preporuka oslanjaju isključivo na vrednovanje resursa od strane korisnika. Ne uzimaju u obzir dodatne informacije zavisne o kontekstu, koje kod nekih primjena ne treba zanemariti, jer korisnost nekog resursa može varirati u odnosu na okolnosti u kojima će biti korišten.

Još jedan od nedostataka je i nefleksibilnost. Većina postojećih sistema podržava generisanje preporuka po jednom kriteriju, ali ne i preporuka na osnovu više kriterija što može biti važno za određene primjene. Takođe, u većini slučajeva resursi se preporučuju pojedincima, ali ne i grupama. Treba spomenuti i problem skalabilnosti. Prilikom korištenja algoritama poput nalaženja "najbližih komšija", sistem se ne prilagođava dobro povećanju broja korisnika i sadržaja. Za funkcionalnost sistema u slučaju velikog broja korisnika i sadržaja, implementiraju se tehnike *klasterovanja* (*npr. co-clustering*) ili tehnike za  *smanjivanje dimenzije problema* (*npr. singular value decomposition*).

Većina sistema traži od korisnika eksplicitne povratne informacije, što se uglavnom tretira kao uznemiravanje korisnika, pa se ovakve metode smatraju *nametljivim* (*eng. intrusive*). Ipak u nekim istraživanjima se spominju metode za smanjivanje nametljivosti



uz zadržavanje određenog nivoa preciznosti prilikom generisanja preporuka, jer se pokazalo kako ocjene dobivene bez nametljivih tehnika ne mogu u potpunosti zamijeniti eksplicitne povratne informacije dobivene od korisnika <sup>[89]</sup>.

### **2.3.3 Ograničenja tradicionalnih sistema preporuke**

Uprkos prividnoj snazi, tradicionalni sistemi preporuke su često ograničeni na jednostavne zadatke kao što su stvaranje kratke liste preporučenih sadržaja ili izračunavanje sličnih sadržaja zanimljivih korisniku. Uobičajeni primjeri su liste preporučenih sadržaja na osnovu gledanosti, predlaganje novih društvenih veza ili sugeriranje novinskih članaka na osnovu prethodnih čitanja. Vrlo rijetka su okruženja koja iskorištavaju puni potencijal sistema za preporuku i postižu stvaranje potpuno novih, adaptivnih sistema, koji naporno rade kako bi ublažili znakove preopterećenja informacijama.

Gledano iz perspektive adaptacije, spomenuti tradicionalni sistemi preporuke imaju značajne nedostatke. Jedan od glavnih je manjak subjektivnosti u predviđanju relevantnosti, gdje postoji nesklad između načina izvođenja predviđanja i poželjnog predviđanja, baziranog na korisničkim potrebama ili željama.

Svjesno donošenje odluke o izboru odgovarajućeg obrasca preporuke, u fazi dizajniranja sistema, često se smatra kao ograničavajući faktor<sup>[88]</sup>. Prije bilo kakvih akcija modeliranja korisnika ili resursa, istraživač ili programer bira jednu ili više metoda, koja najviše odgovara u stvaranju univerzalnog modela. Dakle, modeliranje korisnika ili resursa je bazirano na pretpostavci, što u osnovi subjektivnosti nije poželjno. Ova pojava je klasičan *problem latentne subjektivnosti* Greška! Nije pronađen izvor reference.

Postoji mnogo primjera u kojima su prisutni ovakvi problemi. Recimo, nekim korisnicima je bitan društveni uticaj pri preporuci, dok kod ostalih to nije slučaj. Neki korisnici će biti zadovoljni preporukom baziranoj na sličnosti filmova prema, naslovu, žanru ili akterima, dok će drugi biti zadovoljniji ako im se preporuči film iz istog vremenskog perioda. Nekom korisnicima mogu odgovarati rezultati koji su globalno visoko ocijenjeni, dok drugima više odgovara iskazano mišljenje sličnih korisnika. Slični problemi se javljaju i

kod rezultata, odnosno objekata preporuke. Dok su pojedini resursi možda najrelevantniji po svom sadržaju, drugi imaju svoju vrijednost na osnovu prethodne ocjene od drugih korisnika. Relevantnost nekog resursa može biti usko vezana za period kada je nastala, dok kod ostalih resursa vremenska dimenzija uopšte nije bitna. Precizno definisanje ovih razlika nije toliko bitno, koliko je bitno saznanje da one postoje.

Agregacijske metode modeliranja suočavaju s istim problemom, nedostatkom subjektivnosti. Agregacija se vrši na generaliziranom, globalnom nivou, gdje se u različitim metodama modeliranja pretpostavlja ista važnost individualnih korisnika i resursa [85],[94], [96],[97],[98]. Ona može poslužiti za minimiziranje grešaka nad skupom podataka, dok problem subjektiviteta ostaje. Kao što je spomenuto, sastavni dio agregacije je generalizacija u kojoj se svi korisnici tretiraju na isti način, što predstavlja veliki problem u pronalasku osnovne ideje izgradnji modela korisnika.

#### **2.3.4 Savremeni pristupi u razvoju sistema preporuke**

Prioritet algoritama treba da bude implicitno i automatski baziran na uspješnosti prijašnjeg učinka za individualne korisnike i resurse. Skup korisnika i resursa je često prevelik, te za jednu ili generalizovanu kombinaciju metoda ne bi bilo moguće akumulirati sve nijanse značajnih predviđanja<sup>[99]</sup>.

Za rješavanje ovog problema, neki autori predlažu novu metodu baziranu na prilagodljivoj preporuci (*eng. adaptive recommenders, AR*)<sup>[99]</sup>. U odnosu na tradicionalne sisteme preporuke, AR predstavlja novi nivo apstrakcije i personalizacije, a odluke se donose implicitno, bez dodatne eksplicitne interakcije. Drugim riječima, u pitanju je novi sloj u sistemima za preporuku, koji je zadužen za preporuku algoritma koji je najučinkovitiji u konkretnom slučaju. To se, prije svega, odnosi na izbor poznatih algoritama. Istraživanje će biti fokusirano na rješavanje problema latentne subjektivnosti, odnosno bit će razmotreno kako bi dodatni sloj apstrakcije mogao pomoći u rješavanju ovog problema.

#### 2.3.4.1 Teorija i dosadašnja istraživanja

U nastavku rada bit će istraženi obrasci kojim se pokušavaju riješiti ili umanjiti navedeni problemi tradicionalnih sistema preporuke. Posebna pažnja će biti usmjerena na savremene pristupe agregativnog predviđanja preporuke, odnosno na kombinaciji više metoda predviđanja, koje koriste različite algoritme.

[4] opisuje niz jednostavnih pristupa. Min, Max i suma modeli kombinuju individualna predviđanja na neki način, ili vrše izbor jednog ili više rezultata kao konačno predviđanje. Drugi modeli koriste prosjek, ili log-prosjek različitih metoda. Modeli linearne kombinacije provjeravaju težine svakog prediktora i tretiraju predviđanja skladu s tim. Nešto složeniji pristup ima model logističke regresije, koji preko provjere skupa podataka pokušavaju pronaći kombinaciju sa minimalnom greškom<sup>[94]</sup>.

U ranijim pristupima eksperimentisano je sa kombinovanjem tradicionalnih CB i CF metoda. Kao što je ranije spomenuto, CB metode baziraju predviđanja na sličnosti resursa i prethodnih akcija određenog korisnika, dok CF pristupi predviđaju rezultate na osnovu prethodnih akcija sličnih korisnika u sistemu. Kombinacijom ova dva pristupa pokušavaju se minimizirati poznati problemi u svakoj od metoda<sup>[97]</sup>. U CF metodama se obično javljaju problemi rejtinga resursa za nove korisnike, radikalno različite korisnike ili kada se raspolaže vrlo oskudnim podacima. CB metode nemaju takve probleme, ali su manje efikasne nego CF metode na duge staze, jer se CB ne oslanjaju na znanje drugih korisnika u sistemu, odnosno na znanje izvan okvira jednostavne analiza sadržaja. U obje metode, sistemi preporuke se koriste za stvaranje *jednostavno ponderisanog rezultata*<sup>[97]</sup>.

Autor(Burke, 2007) opisuje hibridni sistem preporuke kroz niz pristupa u kombinovanju CB i CF modela, odnosno njihovih rezultata<sup>[101]</sup>. Drugim riječima, hibridni sistemi preporuke su već odavno predmeti istraživanja u ovoj oblasti i ne predstavljaju ništa novo. S druge strane *adaptivni sistemi preporuke (ASP)* imaju zadatak da uklone subjektivno-bazirani izbor u kombinaciji metoda.

Generalno, metode za agregaciju predviđanja u oblasti mašinskog učenja nazivaju se *asemblirane metode* (eng. *ensemble methods, EM*)<sup>[102]</sup>. Iako se najčešće koriste za

kombinovanje klasifikatora za klasificiranje resursa diskretnim oznakama (*eng. tags*), ove metode se takođe koriste za agregaciju numeričkih vrijednosti<sup>[100]</sup>. Ovi pristupi uključuju *bagging* i *boosting* metode i stvaraju kombinovane softverske agente za kombinovanje prediktora<sup>[103]</sup>. Autor <sup>[104]</sup> objašnjava srodne radove i teoriju ovakvih pristupa.

#### 2.3.4.2 Adaptivni sistem preporuke(ASP)

Adaptivni sistemi preporuke (*eng. adaptive recommenders, AR*) predstavljaju tehnike za kombinovanje različitih sistema preporuke, u cilju optimizacije sistema preporuke za svakog korisnika i svaki resurs. U predviđanju relevantnosti određenog resursa korisniku, koriste se različite metode koje razmatraju disjunktne<sup>9</sup> obrasce podataka. Kako bi se uspješno provela analiza adaptivnih sistema preporuke (ASP) neophodno je odgovoriti na dva važna pitanja:

- 1) Kakav rejting svaka od metoda predviđa?
- 2) Koliko je precizno svako od tih predviđanja?

Prve generacije sistema preporuke pokušavaju odgovoriti uglavnom na prvo pitanje koristeći samo jednu metodu za predviđanje nepoznatog rejtinga. Moderne agregativne tehnike pokušavaju odgovoriti na oba pitanja kombinacijom različitih sistema preporuke, koristeći optimalno-generalizovanu šemu ponderisanja. Na primjer, svaki sistem preporuke korišten u kombinaciji može biti ponderisan, na način da rezultira najnižom prosječnom greškom u odnosima između svih posmatranih korisnika i resursa. Prema tome, adaptacija agregativnih sistema preporuke zavisi isključivo od konkretnih objekata, korisnika ili resursa, koji se trenutno razmatraju.

ASP bi se mogli formalno definisati kao *adaptacija skupa sistema preporuke pomoću drugog komplementarnog skupa sistema preporuke* (slika X.). Ovo se može posmatrati kao određeni oblik meta-modeliranja, gdje je jedan skup metoda modeliranja prilagođen pomoću drugog skupa metoda modeliranja. Prvi skup stvara standardne rezultate predviđanja i odgovara na prvo pitanje, dok drugi skup predviđa preciznost svake od

---

<sup>9</sup> Disjunktne skupove - Presjek skupova A i B, označen sa  $A \cap B$ , je skup svih elemenata koji su članovi i skupa A i skupa B. Ako je  $A \cap B = \emptyset$ , tada za A i B kažemo da su disjunktne.

metoda za trenutnog korisnika i resurs, i odgovara na drugo pitanje. Zanimljivo je da ASP može koristiti skup sistema preporuke za oba navedena zadatka, kao što će biti prikazano u nastavku.

ASP je određen takozvanom 6-torkom  $\{ASP = (I; U; R; F; M; A; )\}$ , gdje su :

- Resursi ( $I - Items$ ),
- Korisnici ( $U - Users$ ),
- Rejting ( $R - Ratings$ ),
- Okruženje ( $F - Framework$ ),
- Metode ( $M - Methods$ ),
- Adapteri ( $A - Adapters$ ).

Svaki korisnik  $u \in U$  može ocijeniti, odnosno proizvesti rejting  $r \in R$  za resurs  $i \in I$ . Tada  $r_{u,i}$  predstavlja rejting (skup ocjena) resursa  $i$  koju je proizveo korisnik  $u$ . Kao što je već spomenuto, resurs može predstavljati različite objekte, a u kontekstu kolaboracije, resursi mogu biti i sami korisnici, koji implicitno ili eksplicitno proizvode ocjene<sup>[105],[106],[107],[108]</sup>.

Korištenje termina „rejting“ treba uzeti sa rezervom, jer u kontekstu ASP -a predstavlja mnogo šire značenje. Ekvivalentni termini su: relevantnost, korisnost, skor, ocjena, jačina konekcije i sl. Drugim riječima, rejting predstavlja mjeru korisničkog mišljenja ili odnosa prema određenom resursu i na razumljivom jeziku u posmatranom okruženju. U okviru ovog rada termin „rejting“ će predstavljati akumulaciju svih relevantnih informacija koje će biti korištene u različitim prediktabilnim modelima.

*Okruženje* ( $F$ ) varijabla određuje način predstavljanja podataka u sistemu. Grafikoni i matrice predstavljaju dva kanonska<sup>10</sup> načina predstavljanja korisnika, resursa i rejtinga<sup>[109]</sup>. U nastavku je prikazana matrica ukupnog rejtinga, u kojoj prva dimenzija

---

<sup>10</sup> Kanon (grč. preko lat.) - pravilo, obrazac, norma; normativni kriterij za bilo koje područje znanja ili djelovanja.

predstavlja korisnike, druga resurse, a svaka prikazana ćelija eksplicitni ili implicitni rejting :

$$R_{u,i} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,i} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ r_{u,1} & r_{u,2} & \cdots & r_{u,i} \end{bmatrix}$$

Za iskorištavanje disjunktih obrazaca podataka, koristi se skup *Metoda* ( $M$ ) modeliranja, sa svojim vlastitim načinima procjene nepoznatih rejtinga. Svaki model  $m \in M$  se koristi za izračunavanje nezavisnih predviđanja, a svaka metoda predstavlja poseban sistem preporuke. Za pronalaženje, predviđanje i prezentaciju relevantnih informacija koriste se različiti algoritmi preporuke koji, podatke posmatraju iz različitih perspektiva i sa različitih aspekata. Osim konkretno pohranjenih podataka o korisnicima, resursima i ocjenama, vrlo često se analiziraju „skriveni“ podaci, kao što su vrste i jačina veza između poznatih podataka i sl.<sup>[90],[110],[111],[112]</sup>.

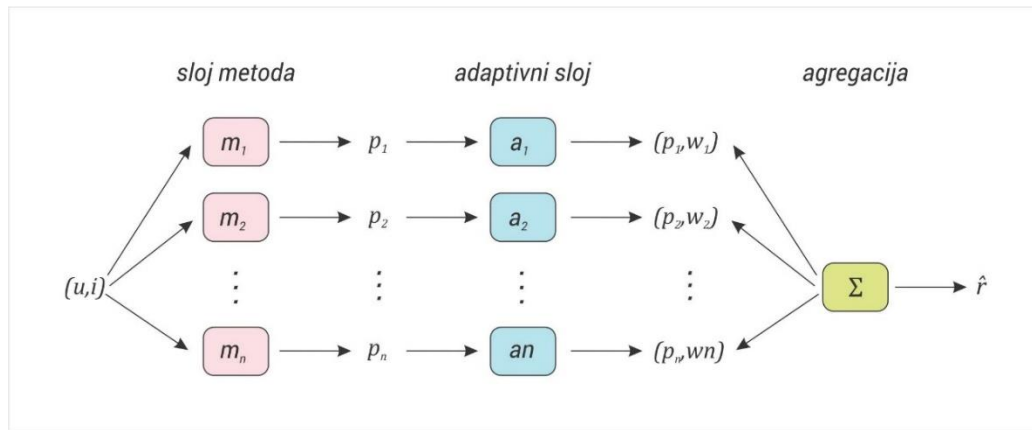
Primjeri takvih sistema preporuke su uključuju „*Slope one*“ familiju algoritama<sup>[113]</sup>, SVD faktORIZACIJU(eng. *singular value decomposition factorization*)<sup>[85],[114],[115],[34]</sup> i ponderisano predviđanje najbližih komšija<sup>[116]</sup>. Ove metode predviđaju nepoznate veze između korisnika i resursa na osnovu određenog obrasca nad podacima, na primjer korisnički profil sličnosti, rejting korelacije ili društvenih veza. Kao što je već napomenuto, u postizanju najboljeg mogućeg zajedničkog rezultata, potrebno je koristiti metode koje koriste disjunktne obrasce, odnosno komplementarno-prediktivne dijelove podataka.

Adapteri (A), u pomenutoj 6-torki, se odnose na drugi nivo metoda modeliranja korisnika. U tradicionalnim agregacijskim metodama, adapteri predstavljaju jednostavnu linearnu funkciju za kombinovanje različitih predviđanja (npr. proračun sume težinskih vrijednosti, po jedan za svaku metodu).

Preciznost kombinovanih prediktora zavisi više od sposobnosti različitih prediktora u otkrivanju različitih aspekata podataka, nego od pojedinačne preciznosti svakog prediktora<sup>[85]</sup>. Višestruki rezultati predviđanja se obično kombinuju u konačni,

jedinstveni rezultat, baziran na generalizovanoj kombinaciji koja je izvedena minimiziranjem određenih grešaka nad svim korisnicima.

Adapteri u ASP ustvari, predstavljaju metode modelovanja korisnika, ali umjesto toga, potrebno je modelovati ponašanje, odnosno preciznost sistema preporuke. Metode na drugom sloju koriste se za predviđanje preciznosti svakog od odgovarajućih osnovnih predviđanja. Dakle, ove metode omogućavaju adaptivnu agregaciju za određenog korisnika i resurs. Drugim riječima, postoje dva različita sloja modeliranja korisnika (Slika 18).



Slika 18 Slojevi sistema preporuke

#### 2.3.4.3 Adaptivna agregacija

Za izvođenje adaptivne agregacije, potrebno je da adapteri predstavljaju stvarne sisteme preporuke. Najjednostavniji generalizovani način predviđanja agregacije je izračunavanje prosjeka svih predviđanja od strane različitih metoda (npr. [4, p.3]). Većina agregacionih tehnika pokušava izvršiti procjenu svake metode na drugačiji način (npr. [16]):

$$\hat{r}_{u,i} = \sum_{m \in M} w_m \times p(m, u, i) , \quad \text{gdje je } 0 \leq w_m \leq 1, \sum_{m \in M} (w_m) = 1. \quad (8)$$

U Formuli(2),  $w_m$  je težina  $w$  primjenjena na metodu modeliranja  $m$ . Težina  $w$  može imati vrijednost u rasponu od 0 do 1. Težine se mogu procijeniti kroz različite metode mašinskog učenja. Međutim, to je još uvijek generalizirani rezultat, odnosno prosjek kroz svako predviđanje. Sistem pretpostavlja da najbolji prosjek predstavlja najbolji rezultat

za svakog pojedinačnog korisnika i resurs. Čak i uz pomoć ovakvih metoda za specifičnu težinu, problem latentne subjektivnosti nije riješen.

U cilju iskorištavanja što većeg broja obrazaca, ostavljajući po strani latentnu subjektivnost, potrebno je raspolagati adaptivnim težinama koje se izračunavaju specijalno za svaku kombinaciju korisnika i resursa. Ako bi svaka težina bila specifična kombinacija, proračunavanje težine za svaku metodu bi praktično postalo teško izvodljivo, odnosno bilo bi potrebno izračunati sve moguće ocjene za svaku korištenu metodu. Drugim riječima, ove adaptivne težine takođe moraju biti procjenjene, isto kao i same ocjene:

$$\hat{r}_{u,i} = \sum_{m \in M} p_w(m, u, i) \times p_r(m, u, i), \quad \text{gdje je } \sum_{m \in M} (p_w(m, u, i)) = 1. \quad (9)$$

U ovom primjeru,  $p_w(m, u, i)$  je predviđena optimalna težina za metodu  $m$  kada se primjenjuje na korisnika  $u$  i resurs  $i$ . Adaptivna preporuka je jedan od načina za procjenu ovih težina, odnosno jedan od načina da se provede  $p_w$ .

Osnovna ideja je da se koriste standardni sistemi preporuke za predviđanje optimalne adaptivne težine. Za sprovođenje ovakve ideje, potrebno je kreirati matricu (grafikon), u kojoj će se čuvati poznate vrijednosti o preciznosti svakog predviđanja rejtinga.

To se može učiniti modelovanjem grešaka za svaku od metoda. Modelovanjem grešaka u standardnim sistemima preporuke, mogu se predvidjeti greške za neprovjerene kombinacije. Predviđanjem grešaka sistema preporuke za korisnika i resurse, takođe se predviđa i preciznost, što je moguće izvesti kreiranjem *greška-matrice* :

$$E_{u,i} = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,i} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ e_{u,1} & e_{u,2} & \cdots & e_{u,i} \end{bmatrix}$$

Kreiranje *greška-matrice* za metode modeliranja se vrši razdvajanjem skupa podataka o rejtingu u dva podskupa. Prvi podskup, tzv. *trening skup*, koriste standardne metode modelovanja, dok se drugi, *podskup za procjenu grešaka*, koristi za popunjavanje *greška-matrice*. Svaka standardna metoda modeliranja može proizvesti greška-matricu u kojoj



neke ćelije imaju vrijednosti. Vrijednost ćelija u toj matrici odgovara predviđanju greške za kombinaciju korisnika i resursa. Svaka metoda modeliranja koristi dio podataka o rejtingu. Greška-matrica je popunjena od ostatka podataka, računanjem greške svih poznatih ocjena iz standardnih metoda:

$$\forall (u, i, r) \in (d_e - d_m): E(m)_{u,i} = |r - p(m, u, i)| \quad \text{gdje je } d_e, d_m \subset D. \quad (10)$$

U formuli 10,  $d_m$  i  $d_e$  su podskupovi posmatranog skupa podataka  $D$ .  $m$  je metoda modeliranja posmatrani podskup  $d_m$ . Za popunjavanje greška-matrice za određenu metodu, uzima se svaki rejting koji se ne koristi za stvarne rezultate i izračunava se greška metode u ovoj kombinaciji. Oskudan rezultat greška-matrice može se koristiti za predviđanje nepoznate greške. Takođe, može se primijetiti velika sličnost između ove dvije matrice, što će na kraju omogućiti standardnim sistemima preporuke obavljanje adaptivne agregacije.

#### 2.3.4.3.1 Faza modelovanja

Za implementaciju novih metoda modelovanja, primjenjuju se sljedeći koraci:

- 1) Podijeliti skup podataka o rejtingu u dva podskupa, trening skup i skup za procjenu grešaka
- 2) Iz trening skupa omogućiti dostupnost specifičnih podataka za svaku metodu modeliranja
- 3) Kreirati greška-matricu koristeći podatke iz skupa za procjenu grešaka
- 4) Podatke iz greška matrice učiniti dostupnim za model greške

Kreiranje podskupova dostupnih podataka je osnovni zadatak u *asembliranom učenju*<sup>[103]</sup>. Za ovu ideju se koriste „*Bootstrap*“ agregacije, poznate kao *pakovanja* (eng. *bagging*)<sup>[100],[104]</sup>. Izvorno, *pakovanja* su korištena za klasifikacijske metode asembliranog učenja, gdje su višestruki klasifikatori koristili podatke iz uniformno uzorkovanog podskupa dostupnog trening skupa. Svaki model je koristio podatke iz odgovarajućih podskupova, a agregacija modela je bazirana na prosjeku njihovih pojedinačnih predviđanja.

Pakovanja odgovaraju i potrebama adaptivne agregacije u slijedećem:

- pomažu pri stvaranju disjunktih prediktora, jer svaki prediktor koristi, ili je specijaliziran za, samo podskup dostupnih podataka
- moguće je testiranje osnovne metode modeliranja bez kompleksnog raščlanjivanja podataka.

Pojava nepovezanih prediktora je objašnjena slijedećom izjavom:

*"Čvrsto vjerujemo da uspjeh pristupa asembliranog učenja zavisi od sposobnosti različitih prediktora da razotkrije različite, dopunjavajuće aspekte podataka. Iskustvo pokazuje da se takav pristup dosta razlikuje od pristupa koji optimizuje tačnost svakog pojedinog prediktora. Analizirani rezultati većinom ukazuju na to, da je kombinovanje preciznih prediktora manje korisno."*

Drugim riječima, prediktori bi u ovom slučaju trebali biti usmjereni na različite aspekte podataka, što se može postići kroz pakovanja slučajnog podskupa dostupnih podataka. Performanse kombinovanih sistema preporuke često ne zavisi od njihovih pojedinačnih preciznosti, nego od njihove vještine predviđanja ocjena za različite dijelove cijelog skupa podataka<sup>[17]</sup>.

Modeli greške su prilagođeni za korištenje standardnih sistema preporuke. Očekivani ulazi i izlazi su isti. Postoje dvije dimenzije, s oskudnim skupom poznatih veza, na osnovu kojih treba predvidjeti nepoznate veze između podataka. Rezultat je skup metoda modeliranja koje mogu predvidjeti greške sistema preporuke za kombinaciju određenog korisnika i resurs.

Primjenom greška-matrice u sistemima preporuke, kao izlaz se generišu greške čija se vrijednost nalazi na istoj skali kao što je i rejting. Rejting koji se posmatra kao buka u matrici rejtinga, odnosno rejting čija vrijednost ne ulazi u skor nekog od osnovnih obrazaca, je pogodan za popunjavanje greška-matrice.

Na primjer, neka metoda može imati veliku vrijednost greške za kombinaciju određenog korisnika i resursa, što može biti korisno za oba elementa, jer predstavlja prepoznatljivu vrstu buke koju sistemi preporuke s lakoćom procesuju, odnosno najčešće odbacuju.

Posebno su interesantne situacije u kojima određena metoda ima stabilne i značajne greške za veliki broj ocjena od korisnika, odnosno veliki broj rejtinga za resurse. U ovom slučaju primjena obrasca greška-matrice nije od velike koristi, jer u sistemima preporuke, ovakvi obrasci su najefikasniji u ulozi identifikacije.

Drugim riječima, iste mogućnosti koje sisteme preporuke na bazi *rejting-matrice* čine efikasnim, će takođe omogućiti efikasnost i pri upotrebi *greška-matrice*. Osobine koje su potrebne za predviđanje rejtinga, takođe se koriste i za predviđanje preciznosti. Međutim, i slabosti sadržane u *rejting-matrici* se prenose na *greška-matricu*. To se prije svega odnosi na: pristrasnost korisnika i resursa, koncepte otklona i varirajuće količine buke. Osim navedenih prisutni su i klasični problemi kao što su hladni start i problem preporuke novim korisnicima, čije preference još uvijek nisu zabilježene u sistemu.

Adaptivni sistemi preporuke bi se trebali suprotstaviti navedenim slabostima, jer u protivnom bi bilo teško pronaći svrhu njihovog korištenja. Korištenjem greška-matrice, će se djelimično pokušati odgovoriti na ovakve izazove, koji su već odavno tema razmatranja u akademskoj zajednici.

Tražeći globalne obrasce među podacima, različiti sistemi preporuke će različito raditi za adaptivni sloj. Fokus interesovanja je na identifikovanju grupe korisnika ili resursa, ili njihovoj kombinaciji, nad kojima će sistemi preporuke raditi vrlo efikasno ili ne. Sistemi preporuke na bazi SVD algoritama su jedna vrsta sistema koji se mogu koristiti za tu svrhu. Smanjivanjem dimenzije za model-greške u posebno kategorizovanu dimenziju greške, može se testirati efikasnost svih dostupnih metoda nad različitim grupama skupova. O ovoj problematici će više biti riječi u diskusiji i zaključcima ovog dokumenta.

#### 2.3.4.3.2 Faza predviđanja

Kada je izgrađen model greške za svaku metodu modeliranja, ove greške se mogu koristiti za procjenu kontekstualne težine. Za adaptivno-agregativno predviđanje primjenjuje se sljedeći algoritam:

$$\hat{r}_{u,i} = \sum_{(m_e, m_r) \in M} \left(1 - \frac{p(m_e, u, i)}{error(u, i)}\right) \times p(m_r, u, i), \text{ gdje je } error(u, i) = \sum_{m_e \in M} p(m_e, u, i) \quad (11)$$

U nastavku su ukratko opisani koraci korišteni u Formuli 11:

- 1) prikupiti predviđanja iz svaku od metoda modeliranja za (u, i).
- 2) prikupiti procijenjene greške za svaku metodu za (u, i).
- 3) Izračunati težine za svaku metodu bazirane na njihovom relativnom predviđanju grešaka.
- 4) Sumiranjem ponderisanih predviđanja izračunati adaptivno predviđanje rejtinga.

Svako predviđanje rejtinga, odnosno relevantnosti, je ponderisano sopstvenim predviđanjem preciznosti koje je uslovljeno trenutno posmatranim korisnikom i resursom.

U jednačini 6 je prikazano svaka metoda sistema preporuke ima dva odgovarajuća modela:

- $m_r$  je *rejting-model*, koristi za predviđanje ocjena,
- $m_e$  je *greška-model*, koristiti za predviđanje greške.

$p(m_e, u, i)$  je predviđanje za model  $m$  (sistem preporuke) za relevantnost između korisnika  $u$  i resursa  $i$ . Svaka metoda iz skupa  $M$  je ponderisana svojim predviđanjem preciznosti. Ponderi su izračunati uzimajući suprotnu vrijednost od predviđenih grešaka metoda. Greške su normalizovane za sve korisnike i resurse pomoću vrijednosti  $error(u, i)$ , koja predstavlja sumu grešaka svih metoda za trenutnu kombinaciju korisnika  $u$  i resursa  $i$ . Rezultat normalizacije su težine u rasponu od 0 do 1, čime se osigurava da konačna prognoza rejtinga bude na istoj skali kao što je povratna vrijednost standardnih sistema preporuke.

U ovom dijelu se treba istaći da jedina razlika između  $m_r$  i  $m_e$  je njihov način kreiranja. Dok  $m_r$  koristi podatke iz *rejting-matrice*,  $m_e$  koristi podatke iz *greška-matrice*. Iz navedenog slijedi logičan zaključak, da se bilo koji standardni sistem preporuke može koristiti u izvođenju adaptivne agregacije. Takvi sistemi se nazivaju *adaptivni sistemi preporuke*. Skup sekundarnih sistema preporuke na adaptivnom sloju se koristi za prilagođavanje (adaptaciju) skupa standardnih (primarnih) sistema preporuke za određene korisnike i resurse.

Važno je napomenuti da su vrste sistema preporuke koje se koriste na adaptivnom sloju apsolutno nezavisne od standardnih sistema preporuke. Adaptivnom sistemu preporuke su potrebni samo predviđeni rejtinzi od standardnih sistema preporuke, i ne zanima ga koji je algoritam upotrijebljen. Prilikom predviđanja, proračuni u metoda-sloju i adaptivnom sloju su potpuno nezavisni, kao jer oba koriste prethodno izračunate modela. Metoda-sloj koristi rejting matricu, ili sopstvene modele stvorene tokom izvođenja, dok adaptivni sloj koristi greška-matrice za svaku standardnu metodu.

Rezultat toga je sistem koji ne samo da prikuplja niz predviđanja za svaku nepoznatu kombinaciju korisnika i resursa, već kombinuje spomenute metode na osnovu vjerovatnoće preciznosti svakog predviđanja.

### ***2.3.5 Prikupljanje podataka za model korisnika***

U kontekstu sistema za generiranje preporuka podaci za model korisnika mogu se prikupljati eksplicitno, koristeći povratne informacije od korisnika ili implicitno odnosno automatski, prikupljanjem podataka o interakciji korisnika sa sistemom [4]. Navedeni pristupi su nezavisni o tehnikama, odnosno algoritmima koje se koriste za generisanje preporuka.

#### ***2.3.5.1 Eksplicitno prikupljanje podataka***

Većina pristupa modelovanju korisnika uglavnom zavise od povratnih informacija od strane korisnika. Kod kolaborativnog pristupa, potrebni podaci se prikupljaju najčešće putem raznih upitnika ili anketa u kojima se navode podaci o ciljevima, preferiranom

načinu učenja, sadržajima koji im se sviđaju i sl. Takvo eksplicitno prikupljanje informacija može se vršiti kroz interakciju sa sistemom preporuke na nekoliko načina<sup>[91]</sup>:

- korisnik direktno navodi podatke ili zahtjeve
- ukoliko preporučeni sadržaj ne odgovara njegovim interesima, korisnik može zatražiti druge dostupne sadržaje
- korisnik vrednuje (ocjenjuje) sadržaje, a kako bi uticao na preporuke koje prima, korisnik može ispraviti ili promijeniti ocjene koje je predvidio sistem
- korisnik bira među ponuđenim opcijama i ukazuje sistemu da mu se određeni sadržaj sviđa, da želi vidjeti još sličnih sadržaja, da je već vidio neki sadržaj i sl.

Interakcija se pritom može odvijati nezavisno o prethodnim povratnim informacijama ili u obliku dijaloga. Opisni načini pretpostavljaju da je korisnik voljan pružiti sistemu potrebne informacije i da je spreman za to odvojiti određeni dio svoga vremena, što predstavlja nedostatak ovoj pristupa<sup>[118]</sup>.

#### 2.3.5.2 *Implicitno prikupljanje podataka*

Kod implicitnog ili automatskog prikupljanja podataka model se izgrađuje i nadograđuje dinamički, bez povratnih informacija od strane korisnika. Na taj način se korisnici ne opterećuju raznim upitnicima i anketama, već se prikupljaju podaci iz njihovih akcija u sistemu.

Potrebni podaci mogu se automatski prikupljati na strani servera ili klijenta. Izvor podataka u tom slučaju mogu biti keš memorija pretraživača (*eng. browser cache*), proksi server (*eng. proxy servers*), agenti za pretraživanje (*eng. browser agents*) ili interfejs (*eng. desktop agents*) te web logovi (*eng. web logs*) i logovi pretraživanja (*eng. search logs*)<sup>[118]</sup>.

Na strani klijenta dostupni su samo podaci o individualnom korisniku što znači da je moguće ostvariti samo individualnu personalizaciju, dok su na strani servera dostupni podaci o svim korisnicima sistema te je moguća i personalizacija grupama korisnika. Ipak, podaci na strani poslužitelja odnose se samo na interakciju korisnika s resursima dostupnima u okviru sistema. Kod pristupa prikupljanja podataka na strani klijenta

dostupni su i podaci o interakciji s ostalim resursima na webu, pri čemu se najčešće automatski prikupljaju podaci o toku klikova (*eng. click stream*) koje je izvršio korisnik za vrijeme pregledavanja resursa te istorije navigacije (*eng. browsing history*)<sup>[119]</sup>.

Obzirom na količinu tako prikupljenih podataka, za otkrivanje znanja u podacima u svrhu personalizacije koriste se postupci dubinske analize podataka na web-u (*eng. Web data mining*). Procesu generiranja preporuka koji se bazira na otkrivanju znanja u podacima koristeći tehnike dubinske analize podataka prethode faze pripreme i transformacije podataka (*eng. data preparation and transformation*) te otkrivanje uzoraka (*eng. Pattern discovery*). U fazi pripreme podaci sakupljaju se podaci iz različitih izvora i transformišu se u oblik pogodan za dalju obradu. U fazi otkrivanja uzoraka koriste se različite tehnike dubinske analize podataka poput klasteriranja (*eng. clustering*), otkrivanja pravila povezivanja atributa (*eng. association rule mining*), analize sekvencijalnih uzoraka (*eng. sequential pattern mining*) i probabilističkog modeliranja (*eng. Probabilistic modeling*). Dobiveni rezultat se tada uključuje u model korisnika u obliku pogodnom za fazu generiranja preporuka [13].

Uspoređujući eksplicitno i implicitno prikupljanje povratnih informacija, (Gauch i ostali, 2007) na osnovu evaluacije primjera primjene ovih pristupa zaključuju kako nema jasnog odgovora na pitanje pruža li implicitno prikupljanje podataka osnovu za izgradnju pouzdanijih modela korisnika<sup>[118]</sup>. Oni navode kako su ranije studije u kojima se pokušavaju uporediti navedeni pristupi, davale prednost eksplicitnom prikupljanju podataka, dok one vršene u posljednje vrijeme predlažu da je pristup implicitnog prikupljanja povratnih informacija bolji. Razlog za to se može potražiti u napretku istraživanja o tome kako automatski prikupiti podatke te kako ih iskoristiti u svrhu izgradnje modela korisnika.

Nedostatak pristupa implicitnog prikupljanja povratnih informacija jeste što ne pruža mogućnost negativnih povratnih informacija. Naime, prilikom automatskog prikupljanja povratnih informacija akcije poput klika na hipervezu koja usmjerava do određenog resursa, pregledavanje neke web stranice i sl., tumače se kao pozitivna povratna informacija - resurs se korisniku "svida". Zato je vrlo upitno može li se zaključiti da za

neki sadržaj korisnik nije zainteresovan samo zato što ga nije odabrao, odnosno pregledao.

U ovom radu, tačnije u FITCKM sistemu, će se koristiti kombinacija eksplicitnog i implicitnog prikupljanja podataka. Sistem će na osnovu eksplicitnih ocjena kojima korisnik ocjenjuje resurse, izvršiti adaptaciju njihovih vrijednosti i kao takve ih pohraniti u bazu podataka. Ovaj koncept će se pokušati realizovati uz pomoć inteligentnog programskog agenta (4.1).



### 3 PRAKTIČNI DIO

Praktični dio rada se sastoji iz dva dijela. U prvom dijelu ovog poglavlja će biti objašnjeni najvažniji detalji razvoja web aplikacije, odnosno virtualnog kolaborativnog web okruženja namijenjenog za razmjenu informacija među korisnicima. U ovom slučaju korisnici bi trebali biti studenti, a objekti kolaboracije bi trebali obezbijediti što jednostavnije i ugodnije okruženje za e-učenje. Prikupljeni podaci će biti iskorišteni u eksperimentalnom okruženju za testiranje različitih sistema preporuke, koji je predstavljen u drugom dijelu ovog poglavlja. Osim spomenutih podataka, u eksperimentalnom dijelu, koristiće se i eksterna kolekcija podataka kako bi se što vjerodostojnije testirala mogućnost primjene sistema preporuke.

#### 3.1 FITCKMS

U ovom poglavlju će ukratko biti predstavljen razvoj kolaborativnog sistema za upravljanje znanjem (FITCKMS), odnosno web okruženja koje će poslužiti za testiranje i istraživanje ideja, teorija i hipoteza navedenih u radu. Kako bi se ostalo u navedenim okvirima, akcent će biti na modelu korisnika i resursa, odnosno na značajnim podacima koje sistem bilježi o njima. Inteligentni programski agent će biti zadužen za posmatranje događaja u sistemu i na osnovu njih će vršiti određene proračune, čije će vrijednosti biti pohranjene u bazu podataka. U bazi podataka sistem će pronalaziti i one „skrivenne“, odnosno metapodatke, te ih koristiti kao ulaz za algoritme preporuke. Krajnja namjena aplikacije je prije svega podrška korisnicima u procesu e-učenja, u smislu kolaboracije i adaptivne navigacije usmjerene ka individualnim interesima. Za potrebe rada, odnosno eksperimentalnog okruženja, koristit će se prikupljeni podaci iz baze podataka sistema.

Za izradu FITCKM sistema korištene su slijedeće aplikacije:

- *Microsoft Visual Studio 2013 – Student Version 12.0.21005.1.REL*
- *MicrosoftSQLServer 2014 Management Studio*

### 3.1.1 FITCKMS - Opis sistema

Fit\_CKMS je sastavljen iz dva modula: *Wiki stranica* i *Pitanja i Odgovori*. Iako pojam kolaboracije ima mnogo šire značenje, kao što je navedeno u prethodnim poglavljima, navedeni moduli će biti sasvim dovoljni za testiranje ideja i hipoteza ovog rada. Pri razvoju sistema će postojati velika doza fleksibilnosti, te će u budućnosti biti moguće unaprijediti postojeće ili dodati nove module i funkcionalnosti.

#### 3.1.1.1 Moduli sistema

Wiki stranica je modul koji omogućava kreiranje, uređivanje, brisanje korisničkih članaka. Osim manipulacije tekстом, ovaj modul će omogućiti i postavljanje različitih multimedijalnih sadržaja. Svi korisnici će biti u mogućnosti da kreiraju novi ili urede svaki članak, nakon čega će promjena biti predložena autoru članka. Autor članka odlučuje da li je promjena primjerena ili ne, te na osnovu toga ima mogućnost da istu odobri. Ako je autor članka odobrio promjenu, promijenjeni članak postaje finalni, a sve prethodne ili buduće verzije ostaju pohranjene u bazi podataka. Takođe će biti omogućeni razgovori o članku, kao i komentari na članak. Svi korisnici će moći eksplicitno ocijeniti članak putem rejtinga (*eng. Rating*) ili izraziti se o tome da li im se članak sviđa ili ne (*eng. Like, UnLike*).

Prilikom čitanja određenog članka sistem bi trebao preporučiti listu nepoznatih, korisnih ili zanimljivih članaka logiranom korisniku. Osim preporuke članaka, sistem predlaže i listu potencijalnih korisnika koji su prema određenom kriterijumu slični logiranom korisniku. Izborom korisnika sa preporučene liste predlaže se lista zanimljivih članaka izabranog korisnika. Paralelno sa spomenutim preporukama sistem preporučuje i listu zanimljivih pitanja i odgovora, odnosno predlože hiperveze za modul *Pitanja i Odgovori*.

Modul Pitanja i odgovori je vrlo sličan modulu Wiki stranica i zbog veličine ovog rada neće biti dodatno analiziran. Treba spomenuti samo da je njegova osnovna funkcionalnost omogućavanje akcija za postavljanje pitanja i davanje odgovora korisnicima, a dodatno je omogućeno i ocjenjivanje odgovora.



Sve predstavljene tabele imaju specifičnu ulogu u sistemu i u nastavku slijedi kratko objašnjenje:

- Tabela *Post* čuva podatke o svim člancima, pitanjima i odgovorima.

*Značajne kolone:*

- *PostVrsta* (tip: int )

vrijednosti: 1 - Članak; 2 - Članak izmjena; 5 - pitanje; 6 - odgovor

- *BrojPoena* (tip: double)

*Kompleksni proračun o relevantnosti objekta objekta. U kalkulacije je uključen rejting, svidanje, nesvidanje, kao i korisnički status koji je učestvovao u navedenim akcijama. Ova kolona ima ulogu u glavnom proračunu za sistem preporuke.*

- *VlasnikID* (tip: int)

*Predstavlja vrijednost jedinstvenog identifikatora vlasnika odnosno kreatora objekta. Ova kolona ima ulogu da obezbijedi propisane akcije i vlasnička prava nad objektom. Dokazuje vlasništvo kako bi rejting objekta bio uključen u proračun statusa i ranga korisnika (Bedž) i učestvuje u preporuci sličnih korisnika.*

- Tabela *Korisnik* čuva podatke o svim korisnicima u sistemu.
- Tabela *Tag* čuva podatke o svim oznakama u sistemu.
- Tabela *Posts\_Tags* čuva podatke o svim oznakama svih objekata iz tabele *Post*.
- Tabela *User\_Likes* povezuje tabele *Korisnik* i *Post*. Čuva podatke korisničkim akcijama rangiranja objekata iz tabele *Post* (rejting, svidanje, nesvidanje). Ima važnu ulogu u proračunu sistema preporuke.
- Tabela *Komentari* povezuje tabele *Korisnik* i *Post*. Čuva podatke o svim komentarima i razgovorima.
- Tabela *Bedž* čuva informacije o svim bedževima u sistemu. Proračunom se korisniku automatski dodjeljuje određeni bedž u zavisnosti od njegovog rejtinga u sistemu.
- Tabela *Potkategorija* čuva podatke o svim otvorenim temama u sistemu. Svi objekti tabele *Post* pripadaju jednom od objekata tabele *Potkategorija*.

- Tabela Kategorija čuva podatke o svim otvorenim oblastima u sistemu. Svi objekti tabele Potkategorija pripadaju jednom od objekata tabele Kategorija.

### **3.1.3 FITCKMS - Pregled mogućnosti aplikacije**

U ovom poglavlju će biti navedeni najvažniji dijelovi FITCKM sistema koji su potrebni za istraživanje i testiranje hipoteza rada. Akcent je na kolaboraciji i dijeljenju znanja, a osnovna uloga sistema je da osigura validne podatke, koji će se koristiti kao ulazi u algoritme sistema preporuke. Osnovne funkcionalnosti FITCKM sistema su:

- Registracija korisnika
- Prijava korisnika
- Kreiranje/uređivanje članaka
- Komentarisanje članaka
- Ocjenjivanje članaka (rejting, sviđanje, nesviđanje)
- Razgovor o člancima
- Kreiranje pitanja /odgovora
- Ocjenjivanje pitanja ili odgovora (rejting, sviđanje, nesviđanje)
- Pretraživanje članaka, pitanja ili odgovora
- Preporuka članaka
- Preporuka pitanja ili odgovora
- Preporuka sličnih korisnika
- Preporuka iz eksternog izvora

### **3.1.4 FITCKMS - Korisnički interfejs**

Komunikacija sa korisnikom je jedan od najvažnijih zadataka aplikacije nezavisno od korištenog programskog jezika. GUI, grafički korisnički interfejs ili grafički interfejs (*eng. graphical user interface*) je metoda interakcije sa računarom kroz manipulaciju grafičkim elementima i dodacima uz pomoć tekstualnih poruka i obavještenja. Kod Web aplikacija korisnički interfejs je fokusiran na ugodno okruženje, laganu navigaciju i brzo pronalaženje željenih informacija.

U prilogu dokumenta (Slika 23, Slika 24 i Slika 25) nalaze se prikazi FITCKMS korisničkog interfejsa.

U narednom poglavlju bit će objašnjen razvoj eksperimentalnog okruženja, koje će između ostalog, koristiti spomenutu kolekciju podataka i testirati različite sisteme preporuke.

### 3.2 Eksperimentalno okruženje

Za potrebe istraživanja i testiranja hipoteza rada pristupljeno je razvoju eksperimentalnog okruženja. Izvorni kod je rađen u programskom jeziku Java i kao osnovna biblioteka korištena je Apache Mahout<sup>TM</sup>.

- [Eclipse Standard/SDK, Version: Luna Release \(4.4.0\)](#)
- [Biblioteka, Apache Mahout<sup>TM</sup>, Version 0.9](#)

U biblioteci Apache Mahout<sup>TM</sup> integrisani su osnovni algoritmi za grupisanje, klasifikaciju i kolaborativno filtriranje. Distribuirana pod komercijalno prijateljskom Apache Software licencom. Trenutno Apache Mahout<sup>TM</sup> podržava tri slučaja korištenja:

- *Preporuka (eng. Recommendation) koja na osnovu korisničkog ponašanja u sistemu koja pokušava pronaći resurse koji bi se potencijalno svidjeli korisniku.*
- *Grupisanje (eng. Clustering) tekstualnih dokumenta u grupe lokalno vezanih dokumenata.*
- *Klasifikacija (eng. Classification) koja na osnovu prethodnog učenja, odnosno učenja iz postojećih kategorizovanih dokumenata pokušava odrediti ispravnu kategoriju za neoznačene dokumente.*

Više detalja o ovoj biblioteci se nalazi na slijedećem linku: <http://mahout.apache.org/>.

Za ovaj rad posebno je zanimljiv prvi slučaj korištenja, koji se odnosi na preporuke. Pakovanja i klase spomenute biblioteke, koristit će se za testiranje različitih sistema

preporuke. Pored standardnih u praksi već korištenih metoda, posebna pažnja će biti usmjerena na adaptivno-agregacione metode.

Za testiranje korištena je kolekcija podataka [MovieLens](#), koja se najčešće koristi za testiranje performansi sistema preporuke <sup>[88],[113],[120],[121]</sup>. Sastoji se od skupa korisnika, filmova i ocjena korisnika filmove, na skali od 1 do 5. *MovieLens* kolekcija je dostupna u više formata i veličina, te je za ovaj eksperiment izabran podskup cijele *MovieLens* kolekcije, sa 100.000 ocjena od 943 korisnika i 1.682 filma.

Postoji nekoliko razloga zašto je za testiranje izabrana baš ova kolekcija. Korištenjem ogromnog skupa podataka, kao što cijela *MovieLens* kolekcija, se ne postiže značajna razlika u performansama<sup>[99]</sup>. Izabrana kolekcija je mnogo manjeg obima što odgovara složenosti izvođenja eksperimenata u ovom radu. Struktura kolekcije je slična podacima FITCKM sistema koji će se takođe koristiti u testiranju.

U prilogu dokumenta se nalazi prikaz programskog koda urađenog u programskom jeziku Java (*Slika 26*).

## 4 METODE

U ovom poglavlju je objašnjena uloga inteligentnog programskog agenta u FITCKM sistemu i metode korištene u eksperimentalnom okruženju. FITCKM sistem je razvijen u okruženju Visual Studio 2013, i za te potrebe korišten je programski jezik *.net*. Kompletan programski kod za eksperimente napisan je u programskom jeziku Java, koristeći Eclipse okruženje. Biblioteka Apache Mahout™, koja je prethodno integrisana u okruženje, sadrži vrlo korisne klase koje su korištene u eksperimentima.

### 4.1 Uloga inteligentnog programskog agenta

U ovom dijelu rada je ukratko objašnjena konkretna upotreba inteligentnog programskog agenta u FITCKM sistemu. Metodologija kompletne implementacije agenta izlazi iz okvira ovog rada, te su u nastavku objašnjen dio koji je interesantan i koristan za sisteme preporuke, odnosno za planirane eksperimente.

Na osnovu istorije kretanja korisnika kroz sistem, agenti FITCKM sistema su zaduženi za procjenu rejtinga korisnika, odnosno procjenu vrijednosti njihovog mišljenja (ocjenjivanja članaka ostalih korisnika) u sistemu, koji opet može da utiče na procjenu rejtinga ostalih korisnika. Iako prethodna rečenica zvuči komplikovano, ideja je vrlo jednostavna.

Pri proračunu rejtinga (Bedž-a) korisnika, ne koristi se samo broj objavljenih post-ova (članaka, razgovora, pitanja ili odgovora), već i njihov kvalitet. Kvalitet članaka je određen brojem poena koje je korisnik dobio od drugih korisnika, putem ocjenjivanja (rangiranje, svidanja, nesvidanja) post-a. Kada post osvoji ili izgubi određeni broj poena, to predstavlja događaj koji će pokrenuti agenta da izvrši potrebne proračune za procjenu rejtinga vlasnika post-a. Na osnovu broja poena, post može dobiti određene medalje. U programskom kodu inteligentnog agenta, medalje su simbolično nazvane imenima: *bronza*, *srebro* i *zlato* i određene brojevima poena na slijedeći način:

- *bronza*, 10 poena
- *srebro*, 20 poena
- *zlato*, 40 poena .



Na osnovu događaja promjene medalje post-a, agent pokreće proračun rejtinga vlasnika post-a i dodjeljuje mu odgovarajući bedž. Bedž predstavlja ponderisanu vrijednost koja se koristi u proračunu prilikom ocjenjivanja. U FITCKM sistemu korisnici mogu imati 10 vrsta bedževa, kojima su dodijeljeni odgovarajuće težine  $wb$  u rasponu od 0,6 do 1,0 (*Greška! Nije pronađen izvor reference.*).

Bedž	Težina
King	1
Akademik	0,92
Doktor	0,85
Magistar	0,79
Profesor	0,74
Student	0,69
Senior	0,65
Junior	0,63
Pionir	0,61
Početak	0,6

Tabela 1 Težine bedževa

Kalkulacija finalne ocjene može se predstaviti jednostavnom formulom:

$$fri = wb * r, (12)$$

U *Formuli 12* finalna ocjena post-a  $fri$  predstavlja produkt ocjene  $r$  na skali od 1 do 5 i težine bedža korisnika  $wb$ , koji ocjenjuje post. Na primjer, ako korisnik koji ocjenjuje post „nosi bedž profesora“ i ocijeni određeni post sa ocjenom 4, koristeći prethodnu formulu, finalna ocjena će imati vrijednost 2,96.

Ocjene, proizvedene upotrebom ovog programskog agenta, se pohranjuju u bazu podataka i kao takve koriste za preporuku. Potrebno je napomenuti da promjena bedža korisnika utiče samo na buduća ocjenjivanja post-ova, odnosno, ocjene koje je korisnik u prošlosti dodijelio post-ovima, noseći neki drugi bedž, ostaju nepromijenjene.

Kakvi se rezultati očekuju primjenom inteligentnih programskih agenata u FITCKM sistemu?

Većina dosadašnjih rješenja, koja se bave problematikom rejtinga korisnika, bazirana su uglavnom na broju post-ova, a ne na njihovom kvalitetu. Implementacijom opisanih inteligentnih programskih agenata mogu se desiti neke vrlo logične situacije. Na primjer, moguća je situacija da jedan post ima veliki broj pozitivnih ocjena od nisko rangiranih korisnika, a da drugi, ima mali broj pozitivnih ocjena ali od visoko rangiranih korisnika. To znači da je moguće da drugi post ima višu vrijednost rejtinga u odnosu na prvi, iako ima manji broj pozitivnih ocjena. O ukupnom rejtingu korisnika u ovom slučaju odlučuje okolina, odnosno ostali korisnici, čije mišljenje nije na isti način vrijednovano. Dakle, rejting korisnika predstavlja varijablu, koja ne zavisi od toga da li je taj isti korisnik trenutno prisutan u sistemu (*eng. online*).

Slično tome, neocijenjeni postovi bi se mogli posmatrati kao punopravni resursi sistema preporuke, jer se ovom slučaju mogu rangirati o odnosu na rejting vlasnika, što opet otvara nove mogućnosti i ideje u budućnosti razvoja ovakvih sistema.

Gledano iz perspektive sistema preporuke primjena ovakvih agenata može poboljšati kvalitet ulaznih podataka u algoritme, odnosno unaprijediti kvalitet sistema preporuke.

Težine bedževa mogu se razlikovati od sistema do sistema, zavisno od vrste, namjene, kapaciteta, zajednice koja ga koristi i sl. Na primjer, ako je riječ o akademskom okruženju, ove težine mogu biti blago kategorisane, u suprotnom mogu biti vrlo rigorozne, ali to bi bio predmet nekog šireg istraživanja.

## 4.2 Priprema podataka

Kao što je u prethodnim poglavljima spomenuto, u eksperimentima će se koristiti dvije kolekcije podataka, kolekcija [MovieLens](#), i kolekcija eksportovana iz baze podataka FITCKM sistema. Za potrebe eksperimenata, podaci su prilagođeni u formatu .csv (*eng. comma-separated values*), i to određenim redoslijedom: korisnik, resurs, ocjena.

Kolekcija [MovieLens](#) je prilagođena pomoću klase *DataConvert*:

```

1 package com.prediction.RecommenderApp;
2 import java.io.BufferedReader;
3 import java.io.BufferedWriter;
4
5 import java.io.FileReader;
6 import java.io.FileWriter;
7 import java.io.IOException;
8
9 public class DataConvert {
10
11     public static void main(String[] args) throws IOException {
12
13         BufferedReader br = new BufferedReader(new FileReader("data/u1.base"));
14         BufferedWriter bw = new BufferedWriter(new FileWriter("data/m1_base.csv"));
15
16         String line;
17         while((line = br.readLine()) != null)
18         {
19             String[] values = line.split("\\t", -1);
20             bw.write(values[0] + "," + values[1] + "," + values[2] + ".0" + "\n");
21         }
22         br.close();
23         bw.close();
24     }
25 }

```

Slika 20 Klasa DataConvert

Kao što je prethodno spomenuto, skup je podijeljen na pet disjunktnih podskupova, te je klasa *DataConvert* (Slika 20 Klasa DataConvert) korištena za svaki od njih pojedinačno. Prilagođeni podaci su spremljeni u datoteku „data“, kao dokumenti: „m1\_base.csv“, „m2\_base.csv“, „m3\_base.csv“, „m4\_base.csv“ i „m5\_base.csv“.

Kolekcija iz baze podataka FITCKM sistema kreirana je jednostavnim upitom (Slika 21 SQL Upit - Spremanje podataka FITCKM) i snimljena u istu datoteku pod nazivom „Fit.csv“.

The screenshot shows the SQL Server Enterprise Manager interface. On the left, a query window displays the following SQL query:

```

SELECT [UserId]
, [PostId]
, [Rate]
FROM [Triglav].[dbo].[User_Likes]
WHERE rate != 0 or Rate != NULL

```

Below the query, the 'Results' tab shows a table with 9 rows and 3 columns: UserId, PostId, and Rate.

	UserId	PostId	Rate
1	5	116	3
2	5	113	3
3	12	113	1
4	5	118	1
5	5	1125	3
6	5	2127	1
7	5	2128	5
8	5	1122	5
9	5	2129	3

On the right, the 'Save Grid Results' dialog box is open. It shows the file path 'ItemRecommenders > data' and a list of files including 'movies.csv'. The 'File name' field is set to 'Fit' and the 'Save as type' is set to 'CSV (Comma delimited) (\*.csv)'. The 'Save' button is highlighted.

Slika 21 SQL Upit - Spremanje podataka FITCKM

FITCKM sistem je testnog karaktera, tako da kreirana kolekcija podataka predstavlja mnogo manji skup nego što je to slučaj u sistemima koji su u praktičnoj upotrebi. Kolekcija se sastoji od 230 ocjena, 66 post-ova i 21 korisnika.

Radi testiranja uloge inteligentnog programskog agenta, simulacija programskog agenta, koji ima zadatak da na osnovu rejtinga post-ova u FITCKM sistemu, proračuna bedž vlasnika, te da na osnovu njega napravi pondere budućih ocjenjivanja to istog vlasnika. Simulacija agenta se zasniva na pokretanju programskog koda koji na osnovu korisničkog bedža i postojećih ocjena, kreira nove ponderisane ocjene i zapisuje ih u bazu podataka. Programski kod je rađen u izvornoj aplikaciji korištenjem programskog jezika .net (*Slika 22*).

```
//funkcija prema formuli  $fri = wb * r$ , racuna ponderisane ocjene i zapisuje ih u bazu podataka
public void AgentCalculationRatings()
{
    List<User_Likes> UL = new List<User_Likes>();
    UL = getUserLikesCorrectRatings();
    foreach (var item in UL) {
        item.RatePonder = item.Rate * getBedzPonder(item.UserId.Value);
        context.SaveChanges();
    }
}

// funkcija vraća listu objekata validnih ocijenjenih postova
public List<User_Likes> getUserLikesCorrectRatings()
{
    var query = (
        from ul in context.User_Likes
        join p in context.Post on ul.PostId equals p.id
        where (p.PostVrsta == 1 || p.PostVrsta == 1) && (ul.Rate != null && ul.Rate != 0)
        select ul).ToList();
    return( List<User_Likes>)query;
}

// funkcija vraća vrijednost pondera na osnovu korisnickog bedza
public float getBedzPonder(int UID)
{
    Korisnik k = getKorisnikByID(UID);
    switch (k.BedzID)
    {
        case 1: return 1.0f;
        case 2: return 0.92f;
        case 3: return 0.85f;
        case 4: return 0.79f;
        case 5: return 0.74f;
        case 6: return 0.69f;
        case 7: return 0.65f;
        case 8: return 0.63f;
        case 9: return 0.61f;
        case 10: return 0.6f;
        default: return 0f;
    }
}
```

*Slika 22 Simulacija programskog agenta*

Nakon izvršavanja programskog koda (*Slika 22*) podaci su uspješno spremljeni u bazu podataka i eksportovani na sličan način kao što je prikazano na *Slika 21*, pod nazivom „FitPonder.csv“. Ovaj postupak je omogućio testiranje sistema preporuke koristeći dva skupa podataka FITCKM sistema, sa istim kardinalitetom. Svi elementi, spomenutih skupova, su ostali identični, osim ocijena koje u drugom skupu imaju ponderisanu vrijednost u odnosu na bedževe korisnika, koji su ocjenjivali post-ove.

Na ovaj način svi podaci su pripremljeni za izvođenje planiranih eksperimenata.

### 4.3 Pregled sistema preporuke

Nakon pripreme podataka, potrebno je definisati skup sistema preporuke koji će se koristiti u eksperimentima. Standardni sistemi preporuke će se koristiti za standardna predviđanja i preciznost procjene, kao što je opisano u *poglavlju 2.3.4*. U cilju pronalaženja disjunktin obrazaca u podacima, potrebno je izabrati metode koje sa različitih aspekata obrađuju podatke. *Tabela 2* Sistemi preporuke daje kratak pregled izabranih sistema preporuke :

	Metoda	Algoritam	Opis
<b>S</b>	PCC	Pearson correlation	Slični korisnici
<b>S</b>	cosine	Cosine similarity	Slični resursi
<b>S</b>	baseline	Baseline	Prosjeci korisnika i resursa
<b>S</b>	item_avg	Baseline	Standardni prosjeci resursa
<b>S</b>	slope_one	Slope One	Delta proračun rejtinga
<b>S</b>	svd1	SVD	ALSWR factorizer, 10 funkcija
<b>S</b>	svd2	SVD	ALSWR factorizer, 20 funkcija
<b>S</b>	svd3	SVD	SVDPlusPlusFactorizer, 10 funkcija
<b>S</b>	svd4	SVD	SVDPlusPlusFactorizer, 20 funkcija.
<b>A</b>	median	Aggregation	Medijana agregovanih rejtinga
<b>A</b>	average	Aggregation	Prosjek agregovanih rejtinga
<b>A</b>	adaptive	Adaptive Aggregation	Prilagođeni agregovani rejtinzi

*Tabela 2 Sistemi preporuke*

Prva kolona u *Tabela 2* se odnosi na vrstu sistema preporuke, gdje oznaka S predstavlja standardne sisteme, a oznaka A agregacione. Za sve standardne sisteme izvršiće se

vremenska procjena izvođenja, kroz konkretne primjere preporuke. Svi agregacioni sistemi će koristiti svaki standardni sistem preporuke.

U sljedećem poglavlju su ukratko opisani sistemi preporuke navedeni u *Tabela 2*.

#### **4.3.1 Sistemi preporuke**

Ovo poglavlje daje kratak opis korištenih metoda u eksperimentima. Detaljne informacije o sistemima preporuke se nalaze u *poglavlju 2.3*. a dio vezan za implementaciju u prilogu (Slika 27, Slika 28).

##### **4.3.1.1 Standardni (tradicionalni) sistemi preporuke**

Kao što je predstavljeno u *Tabela 2*, koristiće se dvije vrste sistema preporuke. Standardni sistemi preporuke (*oznaka S*), koriste različite obrasce među podacima za predviđanje ocjena, što predstavlja osnovni razlog izbora ovako širokog spektra sistema.

Kao što je već objašnjeno, performanse agregativnih metoda (*oznaka A*) više zavise od različitosti standardnih sistema preporuke, nego od njihovog individualnog izvođenja (Bell et al., 2007, str.6). Radi boljeg razumijevanja različitosti ovih obrazaca, u nastavku su ukratko predstavljeni osnovni principi izvođenja standardnih sistema preporuke, koji su definisani u prethodnom poglavlju.

*SVD metode* traže globalne obrasce u podacima smanjivanjem rejting-prostora u koncept-prostora. Smanjivanjem ovog prostora, algoritam je u mogućnosti pronaći latentne odnose, kao što su grupe resursa koji imaju isti ili sličan obrazac rejtinga, ili grupe korisnika koji često ocjenjuju resurse na sličan način. U SVD metodama, faktorizacija se odnosi na algoritme koji se koriste za razlaganje podataka (na faktore), te njihovo pohranjivanje u rejting matricu.

*Slope One* i *Baseline* algoritmi koriste prosječne ocjene resursa i korisnika, te na taj način predviđaju ocjenu. To su jednostavni algoritmi koji se često koriste u kombinaciji u okvirima složenijih pristupa.

*Kosinus sličnosti* je algoritam koji koristi resurse koji su na sličan način ocijenjeni od strane određenog korisnika, i na osnovu njih predviđa relevantnost za nepoznate resurse. Kosinusova mjera je mjera sličnosti između dva vektora koji se nalaze unutar  $n$  dimenzija pronalazeći kosinus ugla između njih. Kosinusova mjera je često korištena za upoređivanje dokumenata u pretraživanju i obradi teksta.

Pearsonov koeficijent korelacije (PCC) algoritam koristi još jedan pristup. Ovaj algoritam, ima sličnu strategiju kao i kosinus sličnosti, traži korisnike sa sličnim obrascima rejtinga. Sličnost se mjeri Pearsonovim koeficijentom korelacije. Predviđanje se stvara prikupljanjem ocjena od sličnih korisnika, za posmatrani resurs, ponderisanjem njihovih sličnosti.

Osnovna razlika između agregativnih i standardnih sistema preporuke su obim obrazaca koje koriste. SVD i standardne metode su usmjerene na globalne efekte, kao što su latentne kategorije i ukupne prosječne ocjene. Kosinus sličnosti i PCC algoritmi posmatraju manje klastere sličnih korisnika i resursa, i računaju prosječnu ocjenu ponderisanu na osnovu sličnosti elemenata.

Ovaj široki spektar sistema preporuke bi trebao olakšati posao adaptivno-agregacionim metodama, koje će u ovom radu biti bazirane na disjunktnim obrascima za predviđanje rejtinga.

#### *4.3.1.2 Adaptivni sistemi preporuke*

Drugi tip sistema preporuke, koji će se koristiti u eksperimentima, su agregativne metode, predstavljene oznakom A u *Tabela 2*. Ove metode kombinuju rezultate svakog od standardnih sistema preporuke. Prve dvije metode imaju vrlo jednostavan pristup agregaciji. *Medijan metoda* agregacije izračunava vrijednost medijane predviđanja, proizvedenu pomoću standardnih sistemima preporuke. Slično, *agregativna metoda prosjeka* (eng. average) uzima srednju vrijednost standardnih predviđanja. Navedene agregativne metode nisu kompleksne po svojoj prirodi, te će biti od pomoći u poređenju adaptivno-agregativnih metoda sa jednostavnim tehnikama agregacije.

Posljednji red u *Tabela 2* se odnosi na adaptivni pristup, koji je detaljno predstavljen u poglavlju 2.3.4.3. Ovaj pristup se nalazi na drugom sloju i izvršava sekundarnu procjenu preciznosti standardnih sistema preporuke. Procjena preciznosti podrazumijeva proračun adaptivnih težina (pondera) standardnih metoda, koje na kraju određuju koliko će koji algoritam biti angažovan u adaptivnoj preporuci.

Prvi korak je procjena greške u predviđanjima za svaku od standardnih metoda. Procijenjene vrijednosti greške se nalaze na istoj skali kao i predviđanje rejtinga (od 1 do 5), tako da niža vrijednost greške predstavlja preciznije predviđanje rejtinga. Drugi korak je normalizacija grešaka u odnosu na ukupnu grešku svih predviđanja. Nakon normalizacije grešaka moguće je proračunati srednju vrijednost greške za agregativne metode uključujući i adaptivni pristup.

Vrijednosti adaptivnih težina (pondera) predstavljaju ustvari opozitne vrijednosti normalizovanih grešaka za svaki od standardnih sistema preporuke.

#### 4.3.2 Procjena greške

Za ocjenu modela tokom predviđanja, potrebno je odrediti mjeru za računanje ukupne greške u velikom broju predviđanja. Uobičajena mjera za procjenu greške sistema preporuke je RMSE (eng. *root mean squared error*) mjera<sup>[85],[88],[121]</sup>. RMSE se obično koristi za predstavljanje devijacija između predviđanja i trenutnog rejtinga i može se predstaviti slijedećom formulom:

$$RMSE(\hat{R}, R) = \sqrt{\frac{\sum_{i=1}^n (\hat{R}_i - R_i)^2}{n}}, \quad (13)$$

U *Formuli 13*  $n$  je ukupan broj predviđanja. Prethodna formula kombinuje skup grešaka u jednu kombinovanu grešku na način da skup procijenjenih predviđanja  $\hat{R}$  poredi sa skupom trenutnih vrijednosti rejtinga  $R$ .

Korisna karakteristika RMSE mjere je da će rezultat greške biti na istoj skali kao i procjene. Na primjer, ako se predviđaju vrijednosti na skali od 1 do 5, izračunate greške



će biti na istoj skali. Konkretno, ako greška ima vrijednost 1 onda se može konstatovati da udaljenost od ispravne ocjene u prosjeku iznosi 1.

RMSE predstavlja nelinearni način procjenjivanja greške, što znači da su veće greške oštrije sankcionisane. Zbog kvadratnih razlika u formuli, nekolicina velikih grešaka ima mnogo veći značaj nego mnoštvo manjih. Dakle, u RMSE pristupu prednost imaju metode koje daju stabilnija predviđanja, u odnosu na precizne metode koje, zbog nekoliko korisnika ili resursa, ne daju željeni rezultat.

RMSE će se koristiti za procjenu svih standardnih i agregativnih metoda korištenih u eksperimentima. Rezultati sa dodatnim objašnjenjima predstavljeni su u slijedećem poglavlju.

## 5 EKSPERIMENTI I REZULTATI

U ovom poglavlju su predstavljani provedeni eksperimenti i njihovi rezultati, na osnovu kojih bi se trebale potvrditi ili odbaciti hipoteze ovog rada (*poglavlje 1.1.1 i poglavlje 1.1.2*). U skladu sa temom ovog rada, eksperimenti su uglavnom usmjereni na testiranje adaptivno-agregativnog pristupa i na ulozi inteligentnih programskih agenata u sistemima preporuke.

### 5.1 Eksperiment 1

U ovom eksperimentu testirane su standardne metode (*S*) nad skupom podataka iz baze FITCKM sistema. Top 6 rezultata preporučenih resursa svih standardnih metoda za proizvoljno izabranog konkretnog korisnika (*jedinstveni identifikator 5*) prikazani su u *Tabela 3*:

<i>tip</i>	<i>metoda</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>S</i>	<i>pearson</i>	4205 4,5359	4180 4,4641	4179 4,3489	4181 4,3022	3154 4,2933	4174 4,0000
<i>S</i>	<i>cosine</i>	4204 4,0000	4193 3,6667	4189 3,6643	4187 3,5729	2131 3,5678	4199 3,5508
<i>S</i>	<i>baseline</i>	4178 4,3913	4196 4,3913	4171 4,3913	3142 4,3913	3154 4,2663	4204 4,1413
<i>S</i>	<i>item_avg</i>	4189 3,6651	4193 3,6645	4169 3,5082	4183 3,5069	4179 3,5062	2130 3,5062
<i>S</i>	<i>slope_one</i>	4189 5,5000	4196 4,5000	4171 4,5000	3142 4,2500	4204 4,2500	4190 4,0000
<i>S</i>	<i>SVD_1</i>	4176 4,0709	4208 2,9383	3154 2,4073	4203 2,3584	4199 1,8057	4202 1,7706
<i>S</i>	<i>SVD_2</i>	4196 3,6613	3142 3,6263	4171 3,5420	4178 3,4967	4192 3,4256	3154 3,3725
<i>S</i>	<i>SVD_3</i>	3154 4,0881	4180 3,6764	4174 3,6318	4173 3,6095	4181 3,5928	3145 3,5754
<i>S</i>	<i>SVD_4</i>	3154 4,1458	4180 3,7382	4174 3,6984	4173 3,6414	3145 3,6295	4181 3,6280

*Tabela 3 Preporučeni resursi (FITCKM baza podataka)*

U *Tabela 3*, redovi predstavljaju standardne metode, a kolone top 6 preporučenih resursa. U vrhovima ćelija nalaze se jedinstveni identifikacioni brojevi, a u dnu vrijednost ocjene

za pripadajući resurs. Ne razmatrajući relevantnost preporučenih resursa, različitost svih rezultata ukazuje na to, da korištene standardne metode koriste različite obrasce među podacima, odnosno veze između podataka se u većini metoda posmatraju sa različitih *aspekata*. U koloni 1, koja predstavlja prvi preporučeni resurs, može se primijetiti da su metode *item\_avg* i *slope\_one* preporučile isti resurs (*id: 4189*) kao i metode *SVD\_3* i *SVD\_4*, koje su preporučile resurs(*id: 3154*). U obzir treba uzeti i činjenicu da je korišten veoma mali skup podataka za testiranje (230 ocjena), te da bi u slučaju korištenja većeg skupa ova razlika bila mnogo izraženija. Kako bi prethodna izjava bila i dokazana, standardne metode su testirane na mnogo većem skupu (*MovieLens*, *podskup d1*), koji se sastoji od oko 20.000 ocjena. Korisnik je takođe prozvoljno izabran (*jedinstveni identifikator 5*), a rezultati predviđanja su prikazani u *Tabela 4*.

<i>tip</i>	<i>metoda</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>S</i>	<i>pearson</i>	212 5,0000	515 5,0000	639 5,0000	640 4,5000	86 4,5000	170 4,5000
<i>S</i>	<i>cosine</i>	1654 4,3750	1500 4,2857	1477 4,2500	1417 4,2500	1390 4,2500	1243 4,2500
<i>S</i>	<i>baseline</i>	1599 4,4277	1653 4,4277	1467 4,4277	1122 4,4277	1500 4,4277	1189 4,4277
<i>S</i>	<i>item_avg</i>	1354 5,0000	114 4,4945	515 4,4006	206 4,3866	1654 4,3705	654 4,3049
<i>S</i>	<i>slope_one</i>	1080 5,5000	1368 5,4074	1367 5,1667	1293 5,0000	1233 4,9167	1344 4,6296
<i>S</i>	<i>SVD_1</i>	1367 5,1324	1643 5,1263	1167 4,7713	1385 4,7000	1160 4,6487	320 4,6337
<i>S</i>	<i>SVD_2</i>	851 4,4689	1467 4,3565	1347 4,3392	1642 4,3189	867 4,3018	1524 4,2039
<i>S</i>	<i>SVD_3</i>	318 4,6901	515 4,6313	64 4,6185	483 4,5959	302 4,5897	127 4,5610
<i>S</i>	<i>SVD_4</i>	483 4,7578	64 4,7318	318 4,7071	12 4,7008	603 4,6661	127 4,6085

*Tabela 4 Preporučeni resursi (MovieLens podskup d1)*

Rezultati iz *Tabela 4* potvrđuju različitost preporučenih resursa, odnosno da izabrane standardne metode koriste različite obrasce pri generisanju preporuke. Standardne metode i njihov način rada su već opisane u poglavljima 2.3.2 i 4.3.1 i ovaj dokaz ne predstavlja ništa novo, ali je vrlo bitan kao podloga za izvođenje narednih eksperimenata, pogotovo za testiranje adaptivno-agregativnog pristupa.

## 5.2 Eksperiment 2

Drugi eksperiment ima zadatak da procjeni greške za sisteme preporuke RMSE metodom u kojoj se skupovi podataka dijele na trenig i testne skupove. U tom procesu kreira se novi trening model podataka i sistem preporuke za testiranje, te se njihove procijenjene karakteristike upoređuju sa aktuelnim testnim podacima.

U eksperimentu je korištena kolekcija podataka MovieLens podijeljena u 5 disjunktih skupova u kolonama *d1*, *d2*, *d3*, *d4* i *d5*. Na ovaj način je osigurana pouzdanost rezultata, jer bi se korištenjem jednog skupa rezultati mogli protumačiti kao slučajni. Kolona *d6* predstavlja kolekciju podataka FITCKM sistema, koja je, kao što je spomenuto, mnogo manjeg obima. Rezultati standardnih sistema (*S*) su korišteni u procjeni greške za agregativne sisteme (*A*).

<i>tip</i>	<i>metoda</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d4</i>	<i>d5</i>	<i>d6</i>
<i>S</i>	<i>pearson</i>	1,2477	1,1683	1,2381	1,2204	1,2375	1,1660
<i>S</i>	<i>cosine</i>	1,0393	1,0483	1,1054	1,0483	1,0520	1,2238
<i>S</i>	<i>baseline</i>	0,9724	0,9740	0,9771	0,9835	0,9822	1,5887
<i>S</i>	<i>item_avg</i>	0,9717	0,9605	0,9833	0,9921	0,9783	1,7255
<i>S</i>	<i>slope_one</i>	0,9471	0,9568	0,9651	0,9624	0,9572	1,7135
<i>S</i>	<i>SVD_1</i>	1,0057	1,0062	1,0040	0,9976	1,0064	2,0414
<i>S</i>	<i>SVD_2</i>	1,0083	1,0189	1,0234	1,0170	1,0282	1,6844
<i>S</i>	<i>SVD_3</i>	1,0903	1,0928	1,0892	1,1075	1,1028	1,2567
<i>S</i>	<i>SVD_4</i>	1,0730	1,0770	1,0759	1,1228	1,1211	1,2599
<i>A</i>	<i>median</i>	0,7969	0,8038	0,8055	0,8016	0,8079	1,3040
<i>A</i>	<i>average</i>	0,8126	0,8077	0,8151	0,8193	0,8232	1,2506
<i>A</i>	<i>adaptive</i>	<b>0,7218</b>	<b>0,7177</b>	<b>0,7241</b>	<b>0,7278</b>	<b>0,7311</b>	<b>1,0951</b>

*Tabela 5. RMSE vrijednosti sistema preporuke*

U ćelijama *Tabela 5.* se nalaze RMSE vrijednosti za sve sisteme preporuke. Niže RMSE vrijednosti predstavljaju bolje rezultate. U ovom eksperimentu, rezultati dokazuju da agregativne metode imaju mnogo bolje rezultate u odnosu na standardne metode. Adaptivno-agregativna metoda je ostvarila najbolji rezultat, što bi moglo opravdati njenu primjenu u praksi.

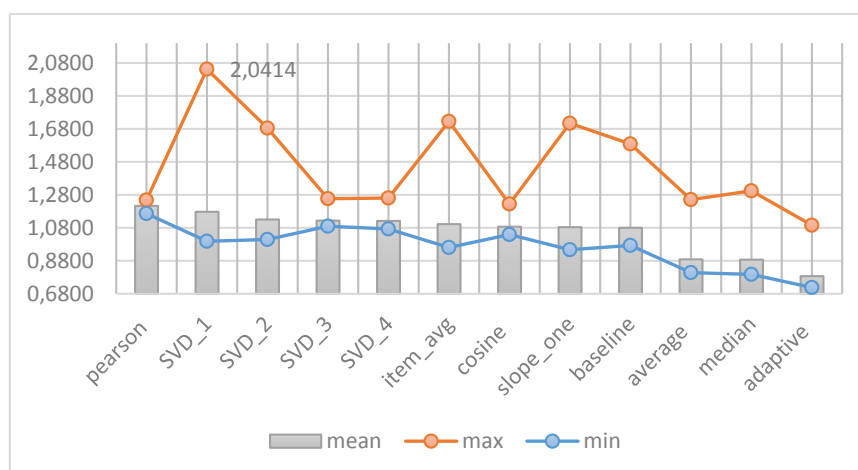
<i>tip</i>	<i>metoda</i>	<i>min</i>	<i>max</i>	<i>mean</i>	<i>stdev</i>
<i>S</i>	pearson	1,1660	1,2477	1,2130	<b>0,0366</b>
<i>S</i>	cosine	1,0393	1,2238	1,0862	0,0715
<i>S</i>	baseline	0,9724	1,5887	1,0797	0,2494
<i>S</i>	item_avg	0,9605	1,7255	1,1019	0,3057
<i>S</i>	slope_one	0,9471	1,7135	1,0837	0,3086
<i>S</i>	SVD_1	0,9976	2,0414	1,1769	0,4235
<i>S</i>	SVD_2	1,0083	1,6844	1,1300	0,2717
<i>S</i>	SVD_3	1,0892	1,2567	1,1232	0,0658
<i>S</i>	SVD_4	1,0730	1,2599	1,1216	0,0715
<i>A</i>	median	0,7969	1,3040	0,8866	0,2045
<i>A</i>	average	0,8077	1,2506	0,8881	0,1777
<i>A</i>	adaptive	<b>0,7177</b>	<b>1,0951</b>	<b>0,7863</b>	0,1514

*Tabela 6 Statističke vrijednosti sistema preporuke*

U *Tabela 6.* prikazana je statistika sistema preporuke, koristeći rezultate iz *Tabela 5.* Korištene su statističke vrijednosti: minimum (*min*), maksimum (*max*), srednja vrijednost (*mean*) i standardno odstupanje (*stdev*). Rezultati iz ove tabele potvrđuju rezultate iz *Tabela 5,* u kojoj je adaptivno-agregativni pristup ostvario najbolje rezultate.

Standardno odstupanje od sredine ima visoke vrijednosti za svaku metodu, što bi predstavljalo veliki problem, ako je ono rezultat nestabilnog predviđanja nad istim skupom podataka. Međutim, u ovom eksperimentu odstupanje je uzrokovano različitim rezultatima nad različitim skupovima podataka, što predstavlja opravdanje za ovako relativno visoke vrijednosti. Maksimalna odstupanja se uglavnom mogu prepisati skupu *d6,* koji se po svim karakteristikama razlikuje od ostalih podskupova koji su uglavnom zaslužni za vrijednosti minimalnih odstupanja. Minimalne vrijednosti su većinom ujednačene, što ide u prilog metodologiji izvođenja eksperimenta.

Na ***Greška! Nije pronađen izvor reference.*** su predstavljena minimalna i maksimalna odstupanja od srednje RMSE vrijednosti za svaku metodu.



Grafikon 1 Minimalna i maksimalna odstupanja od srednje RMSE vrijednosti

Iako rezultati prikazani na ***Greška! Nije pronađen izvor reference.*** ukazuju na to da je RMSE metoda uspješnija ako se koriste veće kolekcije podataka, također je dokazano da je istu moguće primijeniti i na manjim kolekcijama, sa nešto višim RMSE vrijednostima.

### 5.3 Eksperiment 3

Treći eksperiment bi trebao procijeniti opravdanost upotrebe i uloge inteligentnog programskog agenta, opisanog u poglavlju 4.1. U eksperimentu [1](#) i [2](#) korišteni su podaci iz „Fit.csv“ skupa, FITKMS baze podataka bez implementacije inteligentnog programskog agenta. U ovom eksperimentu je korišten spomenuti skup, preuzetih iz Tabela 3, ali i prilagođeni skup „FitPonder.csv“, sa ponderisanim vrijednostima ocjena. Za preporuku su korištene standardne metode, kao i u prethodnim eksperimentima.

Cilj ovog eksperimenta je dokazivanje različitosti preporučenih resursa korištenjem istih standardnih sistema preporuke, nad kardinalnim skupovima podataka. Razlika između tih skupova se ogleda samo u ocjenama, odnosno načinu pohranjivanja ocjena, dok su ostali elementi ostali nepromijenjeni. U prvom skupu su pohranjene standardne vrijednosti ocjena, a u drugom ponderisane, upotrebom inteligentnog agenta, što je objašnjeno u poglavlju 4.1.

Postizanje bilo kakve razlike u predviđanjima na ovakav način bi otvorilo razne teme za diskusiju, pogotovo one koje se odnose na upotrebu ovakvih ili sličnih inteligentnih programskih agenata u adaptivno-agregativnim sistemima preporuke.

U *Tabela 7.* prikazani su rezultati *eksperimenta 3*, odnosno komparacija top 6 rezultata preporuke, za proizvoljno izabranog konkretnog korisnika (*jedinstveni identifikator 5*), koristeći standardne metode nad spomenutim skupovima.

<i>metoda</i>	<i>rejting</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>pearson</i>	<i>r</i>	4205	4180	4179	4181	3154	4174
	<i>pr</i>	4205	4181	4179	3154	2130	4180
<i>cosine</i>	<i>r</i>	4204	4193	4189	4187	2131	4199
	<i>pr</i>	4204	4193	4189	4187	2131	4199
<i>baseline</i>	<i>r</i>	4178	4196	4171	3142	3154	4204
	<i>pr</i>	3142	4196	3154	4183	4178	4204
<i>item_avg</i>	<i>r</i>	4189	4193	4169	4183	4179	2130
	<i>pr</i>	4189	4193	4169	4183	2130	3158
<i>slope_one</i>	<i>r</i>	4189	4196	4171	3142	4204	4190
	<i>pr</i>	4189	4196	4204	4190	4171	3142
<i>SVD_1</i>	<i>r</i>	4176	4208	3154	4203	4199	4202
	<i>pr</i>	4187	4199	3160	4183	4179	3144
<i>SVD_2</i>	<i>r</i>	4196	3142	4171	4178	4192	3154
	<i>pr</i>	4180	3145	4195	4169	3160	4189
<i>SVD_3</i>	<i>r</i>	3154	4180	4174	4173	4181	3145
	<i>pr</i>	3154	4205	4180	4181	2130	4186
<i>SVD_4</i>	<i>r</i>	3154	4180	4174	4173	3145	4181
	<i>pr</i>	3154	4205	4180	4181	2130	4186

*Tabela 7. Komparacija rezultata preporuke standardnih metoda nad različitim skupovima iz baze podataka FITCKMS*

U *Tabela 7* prva kolona predstavlja standardne metode preporuke testirane na skupu sa standardnim ocjenama sa oznakom *r*, i skupu sa ponderisanim ocjenama sa oznakom *pr* (*kolona 2*). Ostale kolone, sa oznakama od 1 do 6, predstavljaju top 6 preporučenih resursa za oba skupa. Čelije predstavljaju jedinstveni identifikator resursa u bazi podataka.

Plavom i zelenom bojom su označene razlike u predviđanjima koristeći iste standardne metode nad spomenutim skupovima. Kao što je već spomenuto, broj elemenata oba skupa je veoma mali (*kardinalitet* 230), i postizanje bilo kakve razlike u predviđanjima predstavlja odličan rezultat. Realna su očekivanja da bi u većim skupovima ova razlika bi bila još izraženija, što će biti predmet diskusije u narednom poglavlju, ali ta teorija se zbog veličine ovog rada neće provjeravati.

Može se primijetiti da sve metode, osim kosinusa sličnosti, imaju dosta različita predviđanja. Uzrok istih predviđanja za metodu kosinusa sličnosti može da bude metodologija simulacije inteligentnog programskog agenta, u kojem su postojeće ocjene, ponderisane korisničim rejtinzima, tako da se odnos između resursa nije drastično mijenjao. Ovu teoriju bi takođe bilo potrebno provjeriti upotrebom većih skupova, ali ona nije toliko bitna za dokazivanje hipoteza ovog rada, te može postati predmet budućih istraživanja. Jedan od mogućih zaključaka može biti da upotreba inteligentnog programskog agenta u kombinaciji sa metodom kosinus sličnosti nad malim skupovima podataka nije opravdana, što bi odbacilo glavnu hipotezu ([H1](#)) ovog rada. Međutim, pošto je u pitanju samo jedna standardna metoda, koja zavisi od dodatnih istraživanja, hipoteza ne može biti kompletno odbačena.

Pod pretpostavkom da upotreba inteligentnog programskog agenta za metodu kosinus sličnosti nije opravdana, treba istražiti pozitivne rezultate ostalih standardnih metoda. Razlike u predviđanjima, dokazuju da je upotreba inteligentnog programskog agenta u kombinaciji sa ovim metodama sasvim opravdana. U ovom slučaju, rezultati predviđanja su precizniji u odnosu na rezultate iz *Tabela 3*, jer su dodatno balansirana i filtrirana prema ocjenama koje su upotrebom ovog agenta ponderisane prema rejtinzima korisnika sistema.

Ako se izuzme neistražena metoda kosinusa sličnosti, standardne metode u kombinaciji sa programskim agentom koje su korištene u ovom radu su unaprijeđene kvalitetom ulaznih podataka i kao rezultat imaju preciznija predviđanja, čime je potvrđena pomoćna hipoteza ovog rada ([H1.1](#)):



- *H1.1: Praćenjem korisnika kroz sistem moguće je otkriti nove odnose među podacima i stvoriti nove metapodatke, te iskoristiti njihovo značenje*

Kako je razvijeni inteligentni programski agent FITCKM sistema obučen da djeluje samostalno i stvara nova znanja bez ljudske pomoći, može se zaključiti da je postignut određeni vid umjetne inteligencije, te se i glavna hipoteza rada [H1](#) može smatrati potvrđenom:

- *H1: Razvojem inteligentnih agenata, u kontekstu razotkrivanja nedavnih/trenutnih korisničkih kretanja, interesovanja i potreba, stvara se pretpostavka da je moguće razviti nove oblike znanja (metapodatke), i pomoću njih unaprijediti modele preporuke usmjerene ka konkretnom entitetu, odnosno postići viši nivo umjetne inteligencije sistema.*

Na kraju je potrebno provjeriti šta bi rezultati iz *Tabela 7.* mogli značiti za agregativne metode korištene u ovom radu. Kao što je više puta do sada spomenuto, ove tehnike koriste rezultate standardnih sistema preporuke. Dokazivanjem hipoteza [H1](#) i [H1.1](#) potvrđeno je da su rezultati predviđanja standardnih sistema preporuke postigli dodatnu preciznost, koja se automatski reflektuje na agregativne metode. U [eksperimentu 2](#) je takođe potvrđeno da je adaptivno-agregativni pristup ostvario najbolje rezultate u odnosu na sve standardne i agregativne metode korištene u ovom radu, u kojem su na osnovu [pondera](#) (preciznosti) angažuju i kombinuju standardni modeli, te se i pomoćna hipoteza *H1.2* može smatrati potvrđenom :

- *H1.2: Na osnovu novostvorenih metapodataka moguće je preciznije preporučiti resurse kombinovanjem adaptivnih pondera<sup>11</sup> određenih sistema preporuke.*

Iako su sve hipoteze u ovom radu teoretski i praktično dokazane u radu su se koristila testna okruženja i subjektivno osmišljeni eksperimenti, tako da je ove konstatacije potrebno provjeriti u realnim okruženjima sa mnogo strožijim testovima. Primjena

---

<sup>11</sup> Relativni ponder pokazuje udio pojedine vrijednosti u strukturi cjeline

tehnika korištenih u ovom radu bi mogla imati vrlo široke razmjere ako se dodatno provjere mogućnosti i opravdanost istih, što će biti jedan od predmeta diskusije .

#### 5.4 Eksperiment 4

U eksperimentu 4 je testirana brzina izvođenja algoritama standardnih metoda, koristeći skup podataka „*FitPonder.csv*“ i podskup MovieLens kolekcije „*ml\_base.csv*“, odnosno podskup sa oznakom *dl*.

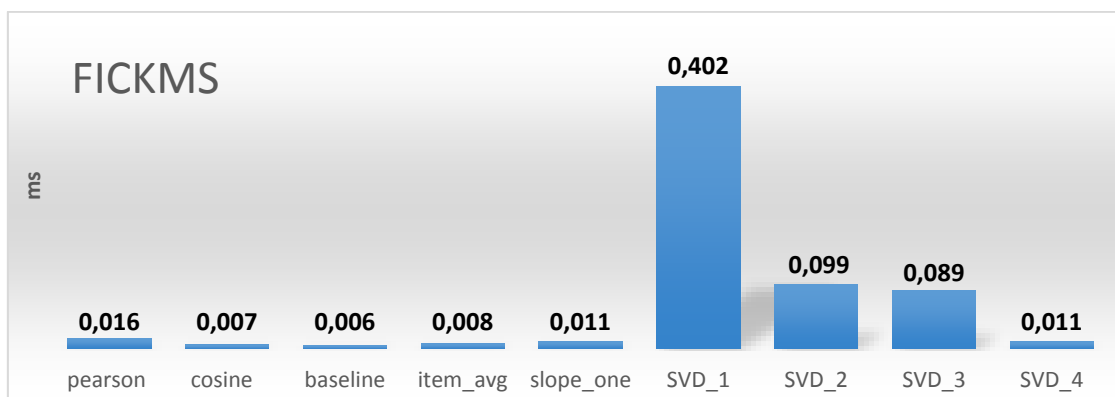
Iako nije planirano istraživanje ove problematike, rezultati ovog eksperimenta mogu postati predmet diskusije i dovesti do određenih zaključaka, jer su usko povezani sa temom ovog rada.

U prvom dijelu ovog eksperimenta testirana je brzina izvođenja standardnih metoda, koristeći skup podataka „*FitPonder.csv*“. Radi pouzdanosti rezultata test je izvršen pet puta, a finalni rezultat predstavlja srednja vrijednost rezultata testova.

<i>tip</i>	<i>metoda</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>srednja vrijednost</i>
<i>S</i>	pearson	0,015	0,019	0,014	0,017	0,015	0,016
<i>S</i>	cosine	0,007	0,006	0,007	0,006	0,007	0,007
<i>S</i>	baseline	0,006	0,005	0,006	0,006	0,006	0,006
<i>S</i>	item_avg	0,010	0,006	0,007	0,009	0,009	0,008
<i>S</i>	slope_one	0,011	0,010	0,012	0,012	0,010	0,011
<i>S</i>	SVD_1	0,363	0,348	0,411	0,414	0,475	0,402
<i>S</i>	SVD_2	0,104	0,091	0,089	0,100	0,110	0,099
<i>S</i>	SVD_3	0,091	0,089	0,094	0,084	0,088	0,089
<i>S</i>	SVD_4	0,009	0,010	0,009	0,012	0,013	0,011

*Tabela 8 Brzina izvođenja standardnih metoda (skup "FitPonder.csv")*

U *Tabela 8* kolone od 1 do 5 predstavljaju broj testa a redovi standardne metode. Čelije predstavljaju rezultate brzina izvođenja algoritama, s jedinicom mjere milisekunda (*ms*). Posljednja kolona predstavlja srednju vrijednost rezultata testova, što je vizuelno prikazano na *Grafikon 2*.



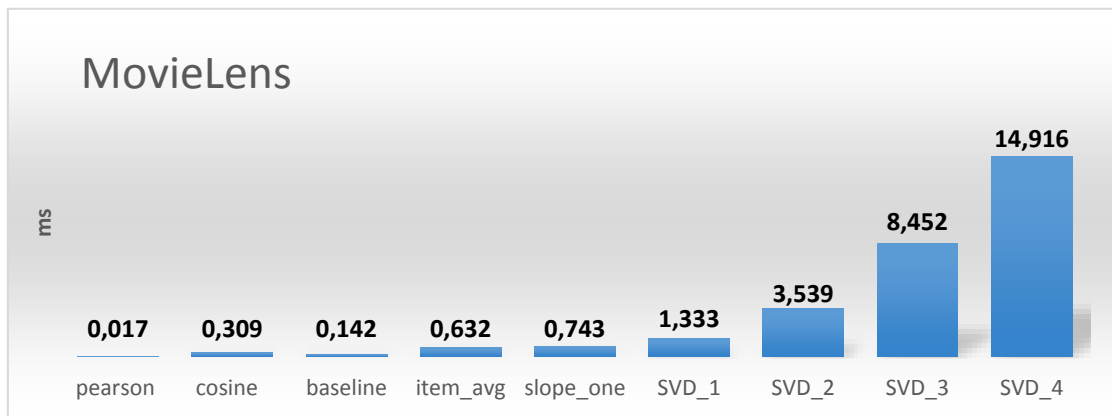
Grafikon 2 Brzina izvođenja standardnih metoda (skup "FitPonder.csv")

Na *Grafikon 2* se primijeti da najduže prosječno vrijeme izvođenja algoritma imaju SVD metode, s naglaskom na SVD\_1 metodu, za čije izvođenje je potrebno 0,402 milisekunde. Rezultati iz *Tabela 5*. RMSE vrijednosti sistema preporuke pokazali da ova metoda ima najvišu RMSE vrijednost, koristeći sličan skup podataka, što ujedno predstavlja i najvišu maksimalnu vrijednost standardnog odstupanja od srednje RMSE vrijednosti, koja je prikazana na ***Greška! Nije pronađen izvor reference.*** Procijenjena brzina izvođenja možda i ne predstavlja veliki problem, jer ipak se radi o 0,4 sekunde, koliko je upitna procijenjena preciznost ove metode u eksperimentu [2](#). Na kraju može se zaključiti da se ova metoda može izostaviti u agregativnim modelima, zbog najviše brzine izvođenja i najniže preciznosti. Ipak, kako su ove metode testirane na malom skupu, ovu teoriju bi valjalo provjeriti na većem skupu, koji je sličniji stvarnim sistemima koji su danas u upotrebi, što je zadatak drugog dijela ovog eksperimenta (*Tabela 9*).

tip	metoda	d1	d2	d3	d4	d5	srednja vrijednost
S	pearson	0,046	0,012	0,008	0,008	0,012	0,017
S	cosine	0,346	0,293	0,307	0,308	0,291	0,309
S	baseline	0,189	0,134	0,133	0,123	0,133	0,142
S	item_avg	0,800	0,662	0,580	0,547	0,573	0,632
S	slope_one	0,879	0,656	0,584	1,032	0,562	0,743
S	SVD_1	2,139	1,052	1,188	1,173	1,112	1,333
S	SVD_2	4,163	3,272	3,419	3,475	3,366	3,539
S	SVD_3	8,833	8,059	8,707	8,471	8,191	8,452
S	SVD_4	15,730	14,758	14,498	14,679	14,915	14,916

Tabela 9 Brzina izvođenja standardnih metoda (MovieLens)

U drugom dijelu eksperimenta testirane su iste standardne metode ali na mnogo većem skupu. U ovom slučaju izvršen je samo jedan test nad disjunktivnim skupovima MovieLens kolekcije ( $dx$ ). Čelije u tabeli predstavljaju takođe brzine izvođenja standardnih metoda u milisekundama. Posljednja kolona, koja predstavlja srednju vrijednost predstavljena je na ***Greška! Nije pronađen izvor reference.***



*Grafikon 3 Brzina izvođenja standardnih (MovieLens)*

Kao i u prvom dijelu ovog testa, rezultati srednje vrijednosti brzine izvođenja pokazuju da su najzahtjevnije SVD metode. U ovom slučaju najviše vremena za predviđanje je potrebno SVD\_4 metodi, čak oko 15 sekundi. Takvo vrijeme je apsolutno neprihvatljivo pri izvođenju ove metode u realnom vremenu. U eksperimentu [2](#), ako se posmatraju samo disjunktivni podskupovi MovieLens kolekcije, ova metoda je imala sasvim solidne RMSE vrijednosti, sa prihvatljivim standardnim odstupanjima od sredine. Slične karakteristike imaju i SVD\_2 i SVD\_3 metode, sa nešto nižim vrijednostima brzine izvođenja. Metoda SVD\_1 koja se u prvom dijelu eksperimenta pokazala kao neupotrebljiva, u ovom slučaju je dala relativno prihvatljive rezultate u brzini izvođenja i preciznosti, te je moguća njena upotreba u agregativnim modelima.

Eksperiment 4 je prikazao brzine izvođenja standardnih metoda korištenih u ovom radu. Korištenje SVD\_Metoda u agregativnim modelima se pokazalo kao vrlo upitno. Ove metode su vrlo korisne u adaptivno-agregativnim pristupima, zbog načina izvođenja i različitosti obrazaca u odnosu na ostale standardne modele, ali zbog svojih loših rezultata u ovom eksperimentu, mogu se smatrati neupotrebljivim. Ipak u praktičnoj upotrebi bilo

bi neminovno provesti detaljna istraživanja, koristeći stvarne podatke konkretnih sistema i eventualno pronaći alternativna rješenja za njihovo korištenje.

Na kraju ovog poglavlja bitno je istaći da su provedeni eksperimenti potvrdili sve hipoteze ovog rada, ali isto tako i da postoje mnoga otvorena i neodgovorena pitanja na ovom polju. O prednostima i nedostacima tematike ovog rada i pristupa predstavljениh u radu bit će više riječi u narednom poglavlju.

## 6 DIKUSIJA I ZAKLJUČAK

U ovom poglavlju će se raspravljati o interesantnim dijelovima ovog rada. Tema je veoma široka, što je dodatno otežalo istraživanje usmjereno na ciljnu problematiku. Bez dodatnog istraživanja komplementarnih i opštih tema, bilo bi skoro nemoguće donijeti valjane zaključke. Iz tog razloga, ovo poglavlje je segmentirano na razmatranje samo značajnih dijelova kompletnog rada.

Fokus će svakako biti na implikacijama i ograničenjima dobijenih rezultata. Kroz provedene eksperimente u prethodnom poglavlju, hipoteze rada su potvrđene u teoriji i praksi, ali možda je važnije razjasniti određene zaključke, nova saznanja i njihova ograničenja.

Na kraju će biti sumirani svi doprinosi vezani za temu, koji ukazuju na buduća istraživanja.

### 6.1 Kolaboracija i upravljanje znanjem

U naslovu rada korišten je izraz „*kolaborativno web okruženje za e-učenje*“. Simbolički gledano, to može biti bilo koja web aplikacija, koja omogućava određeni vid kolaboracije a koristi se u svrhu e-učenja. U naslovu je takođe, naveden i izraz „*mogućnosti primjene sistema preporuke*“, što je i bila osnovna tema istraživanja. Iz perspektive praktične primjene metoda i ideja istraženih u radu, može se zaključiti da je adaptivni pristup preporuci, kao i uloga inteligentnog agenta, sasvim primjenjiva u različitim kolaborativnim okruženjima, koja sadrže komponentu upravljanja znanjem. Upravljanje znanjem je takođe vrlo široka tematika, vrlo aktuelna i u posljednje vrijeme često istraživana, u akademskoj zajednici.

Iako kvalitet komunikacije i dalje prvenstveno zavisi od socijalno-emocionalnih vještina koje čovjek posjeduje, era informacija i naprednih tehnoloških otkrića dala je komunikaciji jednu potpuno novu i, pokazalo se, izuzetno značajnu dimenziju. Upotreba informacionih tehnologija u kolaboraciji donijela je kao nusproizvod određen broj potpuno novih problema, ali što je daleko značajnije, usloвила je i daleko veći napredak.

Očigledna je činjenica da su ovakvi načini komunikacije višestruko premostili i smanjili udaljenosti, skratili vrijeme, povezali milione ljudi koji sarađuju i na taj način pojačali fokus na rješavanje problema, koji su sve kompleksniji.

Razvojem FITCKM sistema, uloga kolaborativnog web okruženje za e učenje u ovom radu je minimizirana, jer su korištena samo dva osnovna modula: *Wiki stranica* i *Pitanja i odgovori*. Za testiranje ideja i hipoteza nije bilo potrebe pristupiti razvoju kompleksnijeg sistema, jer su u eksperimentima korišteni samo podaci iz baze podataka.

Kolaboracija je danas mnogo više od toga, jer je u posljednje vrijeme razvijeno mnoštvo korisnih modula, koji praktično gledano povećavaju jednostavnost razmjene multimedijalnih informacija.

## **6.2 Wiki stranica**

Primjena wiki stranice u kolaborativnom učenju, podučavanju i istraživanju ima svoje prednosti ali i nedostatke. Realnim razmišljanjem i analizama može se doći do permanentnih zaključaka koji idu u korist primjene wiki stranica, kako danas tako i u budućnosti.

Primjenom wiki alata moguće je uspješno organizovati zajednički rad, čime se pozitivno utiče na kolaborativno učenje. Prije svega postiže se pozitivan uticaj na korisnika koji se nalazi u središtu procesa učenja.

Jednostavna upotreba, brzo dodavanje novih stranica, kao i izmjena postojećih sadržaja je ono što utiče na izbor wiki-ja kao jednog od alata koji se često koriste u procesu kolaborativnog učenja, podučavanja i istraživanja. U ovom radu korišten je samo jedan od mnogobrojnih aspekata primjene predstavljenih wiki alata, koji danas imaju veliki doprinos u procesu kolaborativnog e-učenja.

## **6.3 Implikacije i ograničenja sistema preporuke**

Na početku je bitno razjasniti opšte implikacije i ograničenja u adaptivnom pristupu, kao jednom od mogućih rješenja za realizaciju sistema preporuke.

Osnovna pretpostavka je da se moderni sistemi preporuke součavaju sa problemom latentne subjektivnosti. U većini takvih sistema, načini za modelovanje korisnika su predefinisani, na osnovu istraživanja i procjene razvojnog tima. Kako se korisnici razlikuju po svojim karakteristikama, željama, afinitetima i ponašanju pristup modelovanju korisnika ne bi trebao da bude univerzalna, već individualna stvar. Slijedeći tu ideju, modelovanje korisnika je potrebno prepustiti upravo korisnicima putem eksplicitnog ili implicitnog prikupljanja podataka. Iz perspektive sistema preporuke navedeno se može protumačiti kao korisnički izbor sistema preporuke. Kako je u ovom radu izabrano implicitno prikupljanje podataka, po istom modelu je osmišljena i ideja implementacije sistema preporuke. Takvu ideju je moguće sprovesti samo putem adaptivnih metoda. Definisana je kolekcija različitih algoritama za predviđanje, te na osnovu individualnih karakteristika korisnika omogućena je adaptacija. Adaptacija je bazirana na individualnom ponderisanju od kojeg zavisi koliko će se neki od algoritam angažovati za konkretnog korisnika. Provedeni eksperimenti su pokazali da tehnika adaptacije djeluje vrlo obećavajuće, te da se može koristiti kao čvrsta podloga za buduća istraživanja. U isto vrijeme treba se napomenuti da ova tehnika nije testirana u svim mogućim slučajevima korištenja.

Istraživanja provedena u ovom radu, navode na zaključak da bi adaptivni modeli preporuke ostvariti svoj puni potencijal, ako se nalaze u okruženju raznovrsnih algoritama koji koriste različite obrasce u otkrivanju veza između dostupnih podataka, kako bi zaključili koliko je određeni resurs relevantan za individualnog korisnika.

Relevantnost resursa, način i područje pretraživanja informacija je još jedno polje u kojem se javlja problem individualnosti. Postoji mnogo informacija koje određeni resurs čine potencijalno relevantnim. Te informacije, koje se mogu nazvati atributima, predstavljaju ulazne podatke za algoritme, koji mogu biti bazirani na broju atributa, kao što su informacija o vremenu, geografska lokacija, tema, ključne riječi, oznake, rezultati statističkih analiza i sl. Nije teško pretpostaviti da će različiti algoritmi imati različitu preciznost u određivanju relevantnosti za određeni resurs. Ovaj problem bi takođe bilo moguće riješiti upotrebom adaptivnog pristupa baziranog na karakteristikama individualnog resursa.



Korištenje greška-modela ima veoma važnu ulogu u adaptivnom sistemu preporuke, koji je sastavljen iz dva sloja. Prvi sloj je zadužen za standardna predviđanja rejtinga, dok drugi, adaptivni sloj ima zadatak da na osnovu greška-modela predvidi adaptivne težine uposlenosti pojedenih standardnih modela preporuke. Adaptivni sloj, za ovo predviđanje koristi takođe standardne modele preporuke, što se može smatrati kao prednost ovog pristupa, jer su ovi modeli već odavno u upotrebi, testirani su na različitim sistemima i njihov način rada je vrlo dobro poznat u akademskoj zajednici.

Međutim, agregativni pristup preporuci osim spomenutih potencijalnih prednosti ima nekoliko vrlo važnih ograničenja kao što su kompleksnost, izbor podataka, mjere preciznosti i generalni problemi sistema preporuke.

#### **6.4 Kompleksnost adaptivnih sistema preporuke**

Činjenica da su adaptivni sistemi preporuke mnogo kompleksniji od standardnih, dovodi do pitanja da li su prednosti adaptacije vrijedne u odnosu na dodatnu složenost. To zavisi od više faktora. Jedan su sigurno potrebe okruženja u kojem se ovi sistemi koriste. Ako je riječ o kolaborativnim okruženjima za upravljanje znanjem, razmatranim u ovom radu, implementacija adaptivnog pristupa bi itekako imala smisla, jer tu postoji potreba za posmatranjem korisnika kao centralne tačke u sistemu. Korisnici bi imali različite važnosti i adaptivni sloj bi omogućio potrebnu individualnost i preciznost. Ako bi se razmatrala upotreba ovog pristupa u jednostavnim i zatvorenim sistemima, ona apsolutno ne bi imala smisla, ali na sreću takvi sistemi su sve manje u upotrebi i nemaju svijetlu budućnost.

S druge strane, u kontekstu kompleksnosti, korištenje jednostavnih sistema preporuke koji posmatraju slične obrasce u podacima, standardne agregativne metode bi bile mnogo prihvatljivije. Potvrđivanje hipoteza i eksperimentalno postignuta poboljšanja u ovom radu sigurno nisu dovoljna da se ovakav pristup proglasi nadmoćnijim u odnosu na druge bez rezerve. Iako su eksperimentalni rezultati prikazali veliki potencijal adaptivnog pristupa, bez uspjeha u „stvarnom svijetu“ neosnovano bi bilo zaključiti da je ova tehnika posebna ili bolja od bilo kojeg običnog standardnog sistema preporuke.

Pojam kompleksnosti obično prate negativne konotacije i prirodan izbor pri razvoju sistema bi bili standardni pristupi sistemu preporuke. Ideja ovog rada se ne slaže sa tom činjenicom. Preopterećenje informacijama je vrlo nijansiran problem, i jedino rješenje koje ima perspektivu, je inteligentni adaptivni pristup. Ipak realnost je malo drugačija, većina današnjih okruženja često teži ka jednostavnosti, tako da standardni sistemi preporuke još uvijek predstavljaju optimalno rješenje. Ova previranja bi se mogla opisati kao vječna borba između složenosti i potrebne preciznosti.

Realno se nadati da će u skorije vrijeme razvoj virtualnih kolaboracijskih sistema doživjeti određena poboljšanja i da će u moru različitih scenarija, adaptivni pristupi opisani u ovom radu, imati svoju ulogu.

## **6.5 Vremenska ograničenja**

U eksperimentu [4](#) je procijenjeno vrijeme izvođenja standardnih modela preporuke i rezultati upućuju na određene probleme. Ti problemi se takođe prenose i na agregativne metode koje za svoje izvođenje koriste standardne sisteme preporuke. Kod pojedinih standardnih modela primjećen je problem neprihvatljivog trajanja izvođenja u realnom vremenu.

Iz ove perspektive postoji nekoliko mogućih rješenja. Jedno od njih je vrlo rigorozno i podrazumijeva isključivanje ovih metoda iz agregativnih pristupa, a drugo je optimizacija ovih metoda u kontekstu vremenskog izvođenja. Takođe treba ispitati mogućnost i „*offline*“ izvođenja.

Ipak, za rješavanje ovog problema potrebno je provesti dodatna istraživanja i na osnovu njih donijeti odgovarajuće zaključke.

## **6.6 Mjere evaluacije**

Za ocjenjivanje adaptivnog pristupa izabrani su skupovi podataka i evaluacijske mjere koji se najčešće koriste za testiranje sistema preporuke. Izborom tradicionalnih skupova i mjera omogućena je komparacija sa drugim istraživanjima na ovom polju. U testiranje

su dodatno uvršteni lokalni skupovi podataka iz testne aplikacije FITCKMS, čime su omogućena dodatna istraživanja i otvorena nova pitanja.

## 6.7 Inteligentni programski agenti

Iako su rezultati adaptivnog pristupa u početnim eksperimentima su bili vrlo obećavajući, potrebno je naglasiti činjenicu da su testovi obavljeni korištenjem svega nekoliko testnih skupova podataka. Testiranje u skupovima iz realnih okruženja bi sigurno dovelo do novih zaključaka. Ipak, ovaj rad se može još uvijek smatrati kao pionirsko istraživanje novih tehnika, koje su dokazano korisne samo u nekolicini scenarija.

Nakon pozitivnih rezultata primjenom adaptivnog pristupa istraživanje u ovom radu je dobilo novu dimenziju. Osim preopterećenja informacijama, jedan od najvećih problema u današnjim sistemima predstavlja kvalitet podataka. Konkretno, ako je riječ o eksplicitnim ocjenama koje često nastaju „u prolazu“ korisnika kroz sistem, dovodi se u pitanje bilo kakva upotreba sistema preporuke. S druge strane, trenutni pristupi se baziraju uglavnom na statičnim vrijednostima ocjena na određenoj skali. Za prvi problem rješenja su uglavnom nemoguća, bez dodatnog uznemiravanja korisnika, pa čak ni tada se ne bi moglo sa sigurnošću reći da je postignut bilo kakav napredak. Za drugi problem određena poboljšanja su vrlo izvjesna.

Kao što je poznato iz oblasti statistike, na ordinalnoj skali značaj svakog koraka nije nužno jednak. Odnosno, razlika je nepoznata između pojedinih jedinica skale, ali ni ona nije jednaka. Na primjer, ako se koristi skala od 1 do 5, razlika između 2 i 3 možda nije značajna kao razlika između 4 i 5. Ova opaska predstavlja ograničenje većine današnjih sistema u smislu kvaliteta ulaznih podataka u izabrane algoritme sistema preporuke. Upravo iz tog razloga osmišljen je eksperiment [3](#) koji je na tragu poboljšanja ovog problema.

U cilju podizanja kvaliteta ulaznih podataka osmišljen je i integrisan inteligentni programski agent u testni FITCKM sistem. Njegova uloga je vrlo jednostavna i odnosi se na individualnu ulogu svih korisnika, resursa i veza između njih u sistemu. Drugim riječima, ako sistem posmatramo kao prostor, pomijeranje jedne čestice će uzrokovati

odgovor prostora pomjeranjem druge ili grupe čestica. Konkretno, ako je riječ o korisnicima, resursima i njihovim vezama, odnosno ocjenama, bilo kakva promjena će uzrokovati neku drugu promjenu.

Rezultati ovog pristupa su vrlo obećavajući, uzimajući u obzir i jednostavnu implementaciju inteligentnog programskog agenta. Ovim je postignut značajan korak u načinu tretiranja korisnika u sistemu, povećavanju nivoa korisničke individualnosti kao i adaptacije u ocjenjivanju resursa. Ipak, kao što je slučaj i u prethodnim razmatranjima, za osnovano zaključivanje neminovno je svaku ideju provjeriti praktično sa različitim aspektima korištenja.

## **6.8 Pravci budućih istraživanja**

Kroz ovaj rad prikazane su osnovne održivosti adaptivno-agregativnih sistema preporuke i upotrebe inteligentnog agenta u ulozi povećanja kvaliteta ulaznih podataka u algoritme tih sistema. Bilo kakav pomak u ovoj oblasti (na bolje ili na gore) bi predstavljao veliki uspjeh za buduća istraživanja. Ipak, primjene i mogućnosti razmatranih pristupa, mogu imati mnogo šire okvire nego što je u ovom radu prikazano.

U toku osmišljavanja teme i postavljanja hipoteza ovog rada, ideja za implementaciju kolaborativnog web okruženja, sa komponentom upravljanja znanjem, je bila mnogo šira, ali i konkretnija. Planiran je razvoj potpuno nove virtualne akademske zajednice, u kojoj bi bile integrisane sve značajne informacije i podaci sistema najpopularnijih visokoškolskih ustanova u regionu. Zanimljivo bi bilo ispitati mogućnost upotrebe adaptivnog sistema preporuke a pogotovo upotrebu inteligentnih programskih agenata i njihovu ulogu u skupu različitih sistema. Ideja se ne čini nemogućom, ali zahtijeva dosta istraživanja i angažman velikog broja stručnjaka iz različitih IT domena. Zbog svoje kompleksnosti, i ova ideja je ostavljena za buduća istraživanja.

Još jedno vrlo interesantno polje za ovu tematiku je kvantitativno izvođenje prilagođenog pretraživanja. Vrlo izvjesna ideja je da bi se adaptivni pristupi preporuci mogli dati određena poboljšanja u adaptivnom pretraživanju, koje se izvodi putem ključnih riječi, oznaka, log-ova, broja klikova, fraza i sl. To zahtijeva veoma široka istraživanja iz ove

oblasti, u smislu preciznosti, trajanja i rada sa ogromnim brojem podataka. Takvi eksperimenti izlaze iz okvira ovog rada, te su ostavljeni za buduća istraživanja.

Za adaptivni pristup preporuci korištene su standardne metode preporuke. Naravno, ni na tom planu nisu iskorištene sve mogućnosti te bi valjalo istražiti kombinaciju sa drugim metodama. Uopšteno, ideja adaptivne preporuke sigurno predstavlja pravce budućeg istraživanja, bez obzira na pristupe koji se koriste. U ovom radu je istražen samo jedan od njih. Takođe, trenutno postoji mnogo raznovrsnih domena gdje bi se ovakvi pristupi mogli testirati.

U pravcu poboljšanja ulaznih podataka u algoritme, predstavljena je uloga inteligentnog programskog agenta, usmjerena samo na određene jednostavne podatke. Ova uloga bi se mogla dodatno proširiti na ostale značajne podatke i time podići nivo umjetne inteligencije sistema. U ovom radu, inteligentni programski agent je dokazano stvorio novo znanje i omogućio sistemu korištenje drugačije dimenzije podataka. Kroz različite uloge i reakcije na različite događaje, ovo znanje bi se dodatno moglo proširiti i kombinovati u podizanju „svjesnosti“ sistema. Takođe, i ove ideje su ostavljene čitaocima na razmišljanje.

## **6.9 Zaključak**

Problem latentne subjektivnosti je često zanemaren i sprečava standardnim sistemima preporuke postizanje svog punog potencijala. Osnovna ideja ovih sistema je predviđanje nepoznate ocjene za resurse, na osnovu obrazaca koji koriste ocjene ostalih resursa. Njihov individualni nastup je različit od sistema do sistema i zavisi od korisnika i resursa tog sistema. Moderni pristupi agregativnim sistemima preporuke koriste različite obrasce, većinom na generalizovanom nivou, jer se korisnici i resursi tretiraju na isti način, što narušava osnove subjektivnosti. Pristupi koji koriste prosječne ili predefinisane težine (pondere) modela, pretpostavljaju da je najbolji prosjek ujedno i najbolji rezultat predviđanja za sve pojedinačne korisnike ili resurse. Latentna subjektivnost predstavlja sveobuhvatan problem koji često može biti otkriven u različitim tehnikama mašinskog učenja.

Adaptivni sistemi preporuke mogu predstavljati određene pomake u rješavanju ovog problema. Rezultati eksperimentima dokazuju da adaptivno-agregativni pristup, koji kombinuje predviđanja na osnovu procijenjene tačnosti (pondera), nadmašuje standardne sisteme i jednostavne agregativne pristupe. Dodatna vrijednost ovog pristupa je korištenje standardnih sistema preporuke u procjeni preciznosti, odnosno određivanju pondera. Problem latentne subjektivnosti se u adaptivnom pristupu rješava se stvaranjem boljih rezultata, odnosno boljih kombinacija različitih prediktora za individualnog korisnika. Ako postoji potreba za kombinacijom više algoritama čiji je rezultat skup predviđanja baziran na resursima, adaptivni pristup može procijeniti greške i adaptirati preciznost predviđanja za individualni resurs. To dovodi do zaključka da ovakav pristup može biti primjenjiv i izvan oblasti sistema preporuke, te da postoji mogućnost postizanja mnogo boljih rezultata nego što je to bio slučaj u eksperimentima ovog rada. Zanimljivo bi bilo istraživanje potencijala adaptivnog pristupa u društvenim mrežama ili sistemima koji predviđaju relevantnost veoma različitih resursa.

Uloga inteligentnog programskog agenta je dala dodatni doprinos adaptivnom pristupu preporuke i rješavanju problema latentne subjektivnosti. U eksperimentu [3](#) je dokazano da kvalitet podataka pozitivno utiče na kvalitet preporuke. Dublja istraživanja na polju adaptacije ulaznih podataka ne samo da bi poboljšala rad postojećih algoritama i efikasnost predviđanja, nego bi donekle riješila i problem preopterećenja informacijama. Možda bi ovaj pristup nabolje objasnila izreka: „*kvalitet kreira kvantitet*“.

Na kraju ovog rada treba postaviti još samo jedno pitanje:

*„Koji sistem preporuke primijeniti u kolaborativnom web okruženju za e-učenje“*

?

## 7 LITERATURA

- [1] P. Brusilovsky, *Adaptive navigation support: From Adaptive Hypermedia to the Adaptive Web and Beyond*, *PsychNology Journal*, Vol. 3, 2004, pp. 7-23
- [2] A. Paramythis, *Adaptive Support for Collaborative Learning with IMS Learning Design: Are We There Yet? Proceedings of the Workshop on Adaptive Collaboration Support, held in conjunction with the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Hannover, Germany, 2008*, pp. 17-29
- [3] Mamcenko J., Beleviciute I. (2007), *Data Mining for Knowledge Management in Technology Enhanced Learning. In proceedings of the 6th WSEAS International Conference on Applications of Electrical Engineering, Istanbul, Turkey, May 27-29, 2007*
- [4] Giboin A., Dieng R., Karsenty L., De Michelis G., *Designing Cooperative Systems, The Use of Theories and Models, Proc. Of 5-th Int. Conf. on the Design of Coop. Syst. (COOP'2000)*, IOS Perss 2000.
- [5] Osher D.M., *Creating Comprehensive and Collaborative Systems, Journal of Child & Family Studies*, vol. 11, no. 1, p. 91-99, 2002.
- [6] Camarinha-Matos L.M. and H. Afsarmanesh, *Collaborative Networks: Reference Modeling*. Springer US, 2008, ch. Collaboration forms.
- [7] Nițchi S.I., A.Mihăilă, C. Mihăilă, *Some Remerks on Knowlegde Management in Virtual Organizations, 1st CEE Symposium on Business Informatics, Viena, februarie 2009, Oestrische Computer Gessellschaft, Austrian Computer Society, Wi09, ISBN 987-3, 85403, 242-7, p. 207-218.*
- [8] Fuks H., et al. *The 3C Collaboration Model*, 2008, pp. 637-644.
- [9] Grosz B., 1996, *AAAI-94 President Address Collaborative Systems*, *AAAI Magazine*, summer, 1996, p. 67-85.
- [10] Alavi, M. and A. Tiwana, A. *Knowledge integration in virtual teams: The potential role of KMS. Journal of the American Society for Information Science and Technology*, 53(12), 2002, 1029-1037.
- [11] Turban E., J.A. Aronson, L. Ting-Peng, *Decisions Support Systems and Intelligent Systems, Seventh Edition*, Prentice Hall, 2005.
- [12] Thagard P., 1997, *Collaborative Knowledge*, <http://cogsci.uwaterloo.ca/Articles/Pages/Collab.html>
- [13] Muntean, M., Târnăveanu, A. (2009), *Information Technology & Organizational Knowledge Management, Proceedings of the 13th WSEAS International Conference on COMPUTERS*, WSEAS Press, ISBN 979- 960-474-099-4, pp. 335-339
- [14] Scaffert S., *IkeWiki: A SemanticWiki for Collaborative Knowledge Management*, Salzburg Research Forschungsgesellschaft/Salzburg New Media Lab
- [15] Nunamaker J.F., Romano N.C., Bricks R.O., *A Framework for Collaboration and Knowledge Management, Proceedings of the 34th Hawaii International Conference on System Sciences – 2001*

- [16] Andrade J., Ares J., García R., Rodríguez S., Suárez S., *A Knowledge-Based System for Knowledge Management Capability Assessment Model Evaluation*, WSEAS Transactions on Computers, Issue 5, Volume 9, 2010, pp. 506 – 515.
- [17] Paladini E.P., de Carvalho F.G., *New Standards for Competitive Distinctions: A Practical Model*, WSEAS Transactions on Computers, Issue 2, Volume 10, 2011, pp. 21 – 50.
- [18] Tučková Z., Strouhal J., *Knowledge-Intensive Services: New Leader of Production Stages?*, WSEAS Transactions on Systems, Issue 4, Volume 9, 2010, pp. 432 -441.
- [19] Akoumiakakis D., *Distributed Knowledge Management in Virtual Organizations: the 'Social' Experience Factory*, The Electronic Journal of Knowledge Management, Volume 6, Issue 1, 2008, pp.13-32
- [20] Jones P.M., *Collaborative Knowledge Management, Social Networks and Organizational Learning*, NASA Ames Research Center, Human Factors Research and Technology Division
- [21] Tomblin M.S., *Group and Organizational Learning Effects from Multiparticipant DSS Usage*, in BURSTEIN F and C.W.HOLSAPPLE (Eds.), *Handbook on Decision Support Systems, 1, Basic Themes*, Springer, 2008, p. 813-133.
- [22] Garvin D.A., *Building a learning organization*, Harvard Business Review, 1993, July/August.
- [23] Bosch-Sijtsema P.M., *Knowledge Management in Virtual Organizations: Interorganizational and interproject knowledge transfer*, Organizational Knowledge, Learning and Capabilities conference 2002, <http://www2.warwick.ac.uk/fac/soc/wbs/conf/olkc/archive/oklc3/papers/id191.pdf>
- [24] Nonaka I., *A dynamic Theory of Organizational Knowledge Creation*, Organization Science, Vol.5, No.1, February.
- [25] Zakaria Suliman Zubi. 2008. *Knowledge discovery query language (KDQL)*. In *Proceedings of the 12th WSEAS international conference on Computers (ICCOMP'08)*, N. E. Mastorakis, V. Mladenov, Z. Bojkovic, D. Simian, S. Kartalopoulos, A. Varonides, C. Udriste, E. Kindler, S. Narayanan, J. Lloret Mauri, H. Parsiani, and Ka Lok Man (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 497-519
- [26] Berryman R., *Knowledge management in virtual organizations: A study of a best practices knowledge transfer model*, Dissertation Prepared for the Degree of PhD., University of North Texas, May 2005.
- [27] Massey A.P., *Collaborative Technologies*, in BURSTEIN F and C.W.HOLSAPPLE (Eds.), *Handbook on Decision Support Systems, 1, Basic Themes*, Springer, 2008, p. 341-354.
- [28] Dieng-Kuntz R. et al., *Methodes et outils pour la gestion des connaissances, Une approche pluridisciplinaire du Knowledge Management*, Dunod, Paris, 2-nd Edition, 2001.
- [29] P. Brusilovsky and N. Henze, *Open Corpus Adaptive Educational Hypermedia*, In *The Adaptive Web, ser. Lecture Notes in Computer Science*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4321, ch. 22, pp. 671-696
- [30] P. Anderson, *What is Web 2.0? Ideas, technologies and implications for education*, JISC Technology and Standards Watch, 2007



- [31] J. G. Recuenco and D. Bueno, *Balanced Recommenders: A hybrid approach to improve and extend the functionality of traditional Recommenders*. In *Proceedings of International Workshop on Adaptation and Personalization for Web 2.0 (AP-WEB 2.0 2009)*, Trento, Italy, 2009, pp. 88-97
- [32] E. García, C. Romero, S. Ventura and C. de Castro, *An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering*, *User Model User-Adap Inter* (2009) 19:99–132
- [33] P. Resnick and H. Varian, *Recommender Systems*. *Communications of the ACM (CACM)*, 40 (3), 56-58, 1997.
- [34] V. Radovic, M. Prodic, a. Trifunovic, A. Stojicic, (12.05.2013.), *Primena wikija u kolaborativnom učenju i istraživanju* [Online]. Dostupno na: <http://skolskizadatak.wordpress.com/>, [26.08.2014.].
- [35] Antil, L., J. Jenkins, S. Wayne, and P. Vadasy. "Cooperative Learning: Prevalence, Conceptualizations, and the Relationship between Research and Practice." *AMERICAN EDUCATIONAL RESEARCH JOURNAL* 35, no.3 (1997): 419-454.
- [36] Jane E. Brindley and Christine Walti, University of Oldenburg, Germany, Lisa M. Blaschke, University of Maryland University College, USA „*The International review of research in open and distance learning*“, Dostupno na: <http://www.irrodl.org/index.php/irrodl/article/view/675/1271>, [26.08.2014.].
- [37] MediaWiki, Slobodna enciklopedija Wikipedia, (26.06.2014.), Dostupno na: <http://sh.wikipedia.org/wiki/MediaWiki>, [26.08.2014.].
- [38] DokuWiki, (26.04.2014.), Dostupno na: <https://www.dokuwiki.org/dokuwiki>, [26.08.2014.].
- [39] TermWiki, Slobodna enciklopedija Wikipedia, (21.11.2013.), Dostupno na: <http://en.wikipedia.org/wiki/TermWiki>, [26.08.2014.].
- [40] WikiEducator, Slobodna enciklopedija Wikipedia, (29.07.2014.), Dostupno na: <http://en.wikipedia.org/wiki/WikiEducator>, [26.08.2014.].
- [41] PmWiki, (03.02.2013.), Dostupno na: <http://www.pmwiki.org/>, [26.08.2014.].
- [42] FosWiki, (22.05.2014.), Dostupno na: <http://foswiki.org/>, [26.08.2014.].
- [43] MathWiki, Dostupno na: <http://mathwiki.ucdavis.edu/>, [26.08.2014.].
- [44] Bradshaw, J.: *Software Agents*. MIT Press, Cambridge (USA) (1997)
- [45] Milicic, T., Podobnik, V., Petric, A., Jezic, G.: *The CrocodileAgent: A Software Agent for SCM Procurement Gaming*. *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science* 5027, 865-875 (2008)
- [46] Podobnik, V., Petric, A., Jezic, G.: *An Agent-Based Solution for Dynamic Supply Chain Management*. *Journal of Universal Computer Science* 14(7), 1080-1104 (2008)
- [47] Petric, A., Podobnik, V., Grguric, A., Zemljic, M.: *Designing an Effective E-Market: An Overview of the CAT Agent*. U: *Proceedings of the 2008 AAAI Workshop on Trading Agent Design and Analysis (TADA'08)*, Chicago (SAD), pp. 62-65 (2008)

- [48] Sardinha, A., Benisch, M., Sadeh, N., Ravichandran, R., Podobnik, V., Stan, M.: *The 2007 Procurement Challenge: A Competition to Evaluate Mixed Procurement Strategies*. *Electronic Commerce Research and Applications* 8(2), 106-114 (2009)
- [49] Ceric, V., Varga, M.: *Informacijska tehnologija u poslovanju*. Element, Zagreb (2004)
- [50] Dumić, G., Podobnik, V., Jezic, G., Tržec, K., Petric, A.: *An Agent-Based Optimization of Service Fulfillment in Next-Generation Telecommunication Systems*. U: *Proceedings of the 9th International Conference on Telecommunications (ConTEL'07)*, Zagreb (Hrvatska), pp. 57-63 (2007)
- [51] Petric, A., Podobnik, V., Jezic, G.: *The CrocodileAgent: Analysis and Comparison with Other TAC SCM 2005 Agents*. U: *Proceedings of the AAMAS Workshop on Trading Agent Design and Analysis and Agent Mediated Electronic Commerce (TADA/AMEC'06)*, Hakodate (Japan), pp. 202-205 (2006)
- [52] Petric, A., Podobnik, V., Jezic, G.: *The CrocodileAgent 2005: An Overview of TAC SCM Agent*. *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science* 4452, 219-233 (2007)
- [53] Petric, A., Podobnik, V., Jezic, G.: *The CrocodileAgent: Designing a Robust Trading Agent for Volatile E-Market Conditions*. *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science* 4496, 597-606 (2007)
- [54] Podobnik, V., Petric, A., Jezic, G.: *The CrocodileAgent: Research for Efficient Agent-Based Cross-Enterprise Processes*. *Lecture Notes in Computer Science* 4277, 752-762 (2006)
- [55] Podobnik, V.: *Programski agenti na elektroničkom tržištu*. Diplomski rad, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, Zagreb, Hrvatska (2006)
- [56] Albayrak, S.: *Agent-oriented technology for telecommunications: introduction*. *Communications of the ACM* 44(4), 30-33 (2001)
- [57] Chorafas, D.: *Agent Technology Handbook*. McGraw-Hill, New York (1998)
- [58] Cockayne, W., Zyda, M.: *Mobile Agents*. Manning Publications, Greenwich (SAD) (1998)
- [59] Brenner, W., Zarnekow, R., Wittig, H.: *Intelligent Software Agents Foundations and Applications*. Springer-Verlag, Berlin (1998)
- [60] Nwana, H.: *Software Agents: An Overview*. *Knowledge Engineering Review* 11(3), 205-244 (1996)
- [61] Fasli, M.: *Agent Technology For E-Commerce*. John Wiley & Sons, Chichester (2007)
- [62] Boudriga, N., Obaidat, M.: *Intelligent Agents on the Web: A Review*. *IEEE Computing in Science & Engineering* 6(4), 35-42 (2004)
- [63] van Bragt, D.D.B., La Poutre, J.A.: *Why agents for automated negotiations should be adaptive*. *Netnomics* 5, 101-118 (2003)
- [64] Weiser, M.: *The Computer for the 21st Century*. *Scientific American* 265(3), 94-104 (1991)
- [65] Weiser, M.: *The World is not a Desktop*. *ACM Interactions* 1(1), 7-8 (1994)
- [66] Weiser, M., Brown, J.: *The Coming Age of Calm Technology*. U: *Beyond Calculation: The Next Fifty Years of Computing*. Springer-Verlag, New York (1997) 75-86

- [67] Jung, J.: *Shared Context for Knowledge Distribution: A Case Study of Collaborative Taggings*. *Lecture Notes in Computer Science* 5027, 641-648 (2008)
- [68] Kraft, R., Chang, C., Maghoul, F., Kumar, R.: *Searching with context*. U: *Proceedings of the 15th International Conference on World Wide Web (WWW'06)*, New York (SAD), pp. 477-486 (2006)
- [69] Kwon, O.: *Multi-agent system approach to context-aware coordinated web services under general market mechanism*. *Decision Support Systems* 41(2), 380-399 (2006)
- [70] Ranganathan, A., Campbell, R.: *An infrastructure for context-awareness based on first order logic*. *Personal and Ubiquitous Computing* 7(6), 353-364 (2003)
- [71] Rashid, O., Coulton, P., Edwards, R.: *Implementing Location Based Information/Advertising for Existing Mobile Phone Users in Indoor/Urban Environments*. U: *Proceedings of the 4th IEEE International Conference on Mobile Business (ICMB'05)*, Sydney (Australija), pp. 377-383 (2005)
- [72] Dietze, S., Gugliotta, A., Domingue, J.: *Bridging the Gap between Mobile Application Contexts and Semantic Web Resources*. U: *Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications*. Information Science Publishing (IGI Global), London (2008) 217-234
- [73] Bentley, F., Metcalf, C. J.: *The Use of Mobile Social Presence*. *IEEE Pervasive Computing* 8(4), 35-41 (2009)
- [74] Bellavista, P., Corradi, A., Montanari, R., Tonin, A.: *Context-Aware Semantic Discovery for Next Generation Mobile Systems*. *IEEE Communications* 44(9), 62-71 (2006)
- [75] Panagiotakis, S., Alonistioti, A.: *Context-Aware Composition of Mobile Services*. *IEEE IT Pro* 8(4), 38-43 (2006)
- [76] Petric, A., Trzec, K., Jurasovic, K., Podobnik, V., Jezic, G., Kusek, M., Ljubi, I.: *Agent-based support for context-aware provisioning of IMS-enabled ubiquitous services*. *Lecture Notes in Computer Science* 5907, 71-82 (2009)
- [77] Amin, M., Ballard, D.: *Defining New Markets for Intelligent Agents*. *IEEE IT Professional* 2(4), 29-35 (2000)
- [78] Jezic, G., Kusek, M., Sinkovic, V.: *Teamwork Coordination in Large-Scale Mobile Agent Network*. *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science* 4251, 236-243 (2006)
- [79] Kusek, M., Lovrek, I., Sinkovic, V.: *Agent Team Coordination in the Mobile Agent Network*. *Lecture Notes in Artificial Intelligence, Subseries of Lecture Notes in Computer Science* 3053, 240-246 (2005)
- [80] Ljubic, I., Podobnik, V., Jezic, G.: *Cooperative Mobile Agents for Automation of Service Provisioning: A Telecom Innovation*. U: *Proceedings of the 2nd IEEE International Conference on Digital Information Management (ICDIM'07)*, Lyon (Francuska), pp. 817-822 (2007)
- [81] Hendler, J.: *Agents and the Semantic Web*. *IEEE Intelligent Systems* 16(2), 30-37 (2001)
- [82] Tamma, V., Payne, T.: *Is a Semantic Web Agent a Knowledge-Savvy Agent?* *IEEE Intelligent Systems* 23(4), 82-85 (2008)

- [83] Lange, D., Oshima, M.: *Seven Good Reasons for Mobile Agents*. *Communications of the ACM* 42(3), 88-89 (1999)
- [84] *Inteligentni agenti*, Slobodna enciklopedija Wikipedia, (14.02.2013.), Dostupno na: [http://hr.wikipedia.org/wiki/Inteligentni\\_agenti#CITEREFRussellNorvig2003](http://hr.wikipedia.org/wiki/Inteligentni_agenti#CITEREFRussellNorvig2003), [26.08.2014.].
- [85] R. Bell, Y. Koren, and C. Volinsky. *The BellKor solution to the Netflix prize*. *KorBell Team's Report to Netflix*, 2007.
- [86] A. Halevy and P. Norvig. *The unreasonable effectiveness of data*. *Intelligent Systems, IEEE*, 24(2):8–12, Mar. 2009.
- [87] M. Banko and E. Brill. *Mitigating the paucity-of-data problem: Exploring the effect of training corpus size on classifier performance for natural language processing*. In *Proceedings of the first international conference on Human language technology research*, pages 1–5. Association for Computational Linguistics, 2001.
- [88] G. Adomavicius and A. Tuzhilin, *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*. *IEEE TKDE: IEEE Transactions on Knowledge and Data Engineering*, 17, 2005. pp. 734-749
- [89] I. Cantador, A. Bellogín and P. Castells, *A Multilayer Ontology-based Hybrid Recommendation Model*, *AI Commun.* 21, 2-3, 2008, pp. 203-210 [10] G. Adomavicius, Y. Kwon, *New Recommendation Techniques for Multicriteria Rating Systems*, *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 48-55, May/June 2007
- [90] G. Adomavicius, Y. Kwon, *New Recommendation Techniques for Multicriteria Rating Systems*, *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 48-55, May/June 2007
- [91] N. Tintarev and J. Masthoff, *A Survey of Explanations in Recommender Systems*, In *Proceedings of the 2007 ACM Conference on Recommender Systems (Minneapolis, MN, USA, 2007)*. *RecSys '07*. ACM, New York, NY, pp. 203-206
- [92] W. Woerndl, A. Helming, and V. Prinz, *Experiences from Implementing Collaborative Filtering in a Web 2.0 Application*. In *Proceedings of International Workshop on Adaptation and Personalization for Web 2.0 (AP-WEB 2.0 2009)*, Trento, Italy, 2009, pp. 120-129
- [93] B. Mobasher, *Data mining for web personalization*. In *The Adaptive Web*, ser. *Lecture Notes in Computer Science*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4321, ch. 3, pp. 90-135
- [94] O. Bjørkøy. *User Modeling on The Web: An Exploratory Review*, 2010.
- [95] J. a. Aslam and M. Montague. *Models for metasearch*. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01*, pages 276–284, 2001.
- [96] D. Carmel, N. Zwerdling, I. Guy, S. Ofek-Koifman, N. Har'el, I. Ronen, E. Uziel, S. Yogev, and S. Chernov. *Personalized social search based on the user's social network*. *Proceeding of the 18th ACM conference on Information and knowledge management – CIKM '09*, page 1227, 2009.

- [97] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. *Combining Content-based and collaborative filters in an online newspaper*. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*, number June, pages 60–64. Citeseer, 1999.
- [98] B. Sergey and P. Lawrence. *The anatomy of a large-scale hypertextual web search engine*. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.
- [99] O. Bjørkøy. „Adaptive Aggregation of Recommender Systems“ Norwegian University of Science and Technology Trondheim, Norway, October 3rd, 2011.
- [100] L. Breiman. *Bagging predictors*. *Machine learning*, 24(2):123–140, 1996.
- [101] R. Burke. *Hybrid web recommender systems*. In *The adaptive web*, pages 377–408. Springer-Verlag, 2007.
- [102] T. Dietterich. *Ensemble methods in machine learning*. *Multiple classifier systems*, pages 1–15, 2000.
- [103] R. Polikar. *Ensemble based systems in decision making*. *Circuits and Systems Magazine, IEEE*, 6(3):21–45, 2006.
- [104] A. Jovic. „Postupci dubinske analize podataka“, Fakultet elektrotehnike i računarstva, Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave, Dostupno na: <http://www.fer.unizg.hr/download/repository/Jovic,KDI.pdf>, [26.08.2014.].
- [105] F. Liu, C. Yu, and W. Meng. *Personalized web search by mapping user queries to categories*. *Proceedings of the eleventh international conference on Information and knowledge management - CIKM '02*, page 558, 2002.
- [106] X. Shen, B. Tan, and C. Zhai. *Implicit user modeling for personalized search*. *Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM '05*, page 824, 2005.
- [107] M. Speretta and S. Gauch. *Personalized Search Based on User Search Histories*. *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 622–628, 2000.
- [108] K. Sugiyama, K. Hatano, and M. Yoshikawa. *Adaptive web search based on user profile constructed without any effort from users*. *Proceedings of the 13th conference on World Wide Web - WWW '04*, page 675, 2004.
- [109] B. Mirza and B. Keller. *Studying recommendation algorithms by graph analysis*. *Journal of Intelligent Information*, 2003.
- [110] M. Pazzani and D. Billsus. *Content-based recommendation systems*. In *The adaptive web*, pages 325–341. Springer-Verlag, 2007.
- [111] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen. *Collaborative filtering recommender systems*. *The adaptive web*, pages 291–324, 2007.
- [112] T. Segaran. *Programming collective intelligence*. O'Reilly Books, 1st edition, 2007.
- [113] D. Lemire and A. Maclachlan. *Slope one predictors for online rating-based collaborative filtering*. *Society for Industrial Mathematics*, 2005.
- [114] D. Billsus and M. Pazzani. *Learning collaborative information filters*. In *Proceedings of the Fifteenth International Conference on Machine Learning*, volume 54, page 48, 1998.

- [115] J. Sun, H. Zeng, H. Liu, and Y. Lu. *CubeSVD: a novel approach to personalized Web search. on World Wide Web*, pages 382–390, 2005.
- [116] T. Segaran. *Programming collective intelligence*. O'Reilly Books, 1st edition, 2007.
- [117] R. Bell, Y. Koren, and C. Volinsky. *Modeling relationships at multiple scales to improve accuracy of large recommender systems*. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, page 95, 2007.
- [118] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. *User profiles for personalized information access*. In *The Adaptive Web*, ser. *Lecture Notes in Computer Science*. P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4321, ch. 2, pp. 54-89.
- [119] H. Drachsler, H.G.K. Hummel, and R. Koper, *Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning*. In *Proceedings of the Workshop on Social Information Retrieval in Technology Enhanced Learning (SIRTEL 2007)*, Crete, Greece, 2007. pp. 18-26.
- [120] M. Alshamri and K. Bharadwaj. *Fuzzy-genetic approach to recommender systems based on a novel hybrid user model*. *Expert Systems with Applications*, 35(3):1386–1399, Oct. 2008.
- [121] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. *Evaluating collaborative filtering recommender systems*. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, Jan. 2004.

## 8 PRILOZI

RegistracijaLogiranje

FitWIKI

WikiQ&ANedavne promjeneSlučajna stranicaPomoćDonacije

Kreirajte članakPostavite pitanje

ČlanakRazgovorČitajUrediVidi izmjene

Korisnik: blocky | Reputacija: 19 | Članak kreiran: 30.3.2013 1:52:06 | Pregleda: 80 | Odgovora: 1 | Rejting: 0 | Komentara: 1



### Lansiran Google Keep

Osim tekstualnih zabilježki, Keep može čuvati i zvukovne snimke, te fotografije. Sučelje podsjeća na pločice s Windows Phonea, koje je moguće reorganizirati u skladu s prioritetom. Uključene su funkcionalnosti pretrage i arhiviranja zabilježki, a za svaku je moguće odabrati i boju. Za uređaje s Androidom 4.2 ili novijim dostupan je i widget koji omogućuje korištenje aplikacije sa zaključanog ekrana uređaja.

Verzija aplikacije na webu dostupna je na adresi [drive.google.com/keep](https://drive.google.com/keep), no korisnici zasad prijavljuju velik broj problema pri korištenju. U izdanju za Android aplikacija je dostupna s Play Storea.

UREDENO UREDENO

Osim tekstualnih zabilježki, Keep može čuvati i zvukovne snimke, te fotografije. Sučelje podsjeća na pločice s Windows Phonea, koje je moguće reorganizirati u skladu s prioritetom. Uključene su funkcionalnosti pretrage i arhiviranja zabilježki, a za svaku je moguće odabrati i boju. Za uređaje s Androidom 4.2 ili novijim dostupan je i widget koji omogućuje korištenje aplikacije sa zaključanog ekrana uređaja.

Verzija aplikacije na webu dostupna je na adresi [drive.google.com/keep](https://drive.google.com/keep), no korisnici zasad prijavljuju velik broj problema pri korištenju. U izdanju za Android aplikacija je dostupna s Play Storea.

0Bodova

W Like

2 osoba je glasovalo za!

W Unlike

2 osoba je glasovalo protiv!

1


2

3

4

5

Za ocjenjivanje članaka morate se logirati



blocky | Reputacija19 | Sviđanja0 | Nesviđanja0

as

Komentiraj

WIKI preporuke

Google će proizvoditi Google naočale u Kaliforniji  
Google Glass uskoro u proizvodnji  
Kako Google zarađuje na mobilnoj platformi?  
Miliarda korisnika na YouTubeu  
iOS donosi četiri puta veće prihode od Androida

Q&A preporuke

Uklanjanje iz Google tražilice  
Pozicioniranje web stranice na Google-u  
Google AdSense na web stranici

Korisnici

blocky

19

kor12

0

kor14

0

Made by Elvis. Copyright © 2014 by Fit. All rights reserved.

Slika 23 FITCKMS – Modul Wiki (Članak)

106






Registracija
Logiranje


FitQ&A

Wiki
Q&A
Tagovi
Korisnici
Bedževi
Neodgovoreno


Postavite pitanje
Kreirajte članak




Naziv: King  
Opis: King  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 0  
korisnika




Naziv: Akademik  
Opis: Akademik  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 0  
korisnika




Naziv: Doktor  
Opis: Doktor  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 0  
korisnika




Naziv: Magistar  
Opis: Magistar  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 0  
korisnika




Naziv: Profesor  
Opis: Profesor  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 0  
korisnika




Naziv: Student  
Opis: Student  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 0  
korisnika




Naziv: Pionir  
Opis: Pionir  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 0  
korisnika



Naziv: Početnik  
Opis: Početnik  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 13  
korisnika



Naziv: Senior  
Opis: Senior  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 1  
korisnika



Naziv: Junior  
Opis: Junior  
Kreiran: prije prije  
godinu dana  
Bedz posjeduje: 1  
korisnika

Slika 25 FITCKMS - Korisnički bedževi

```

public class EvaluateRecommender {

    public static void main(String[] args) throws Exception {

        DataModel model6 = new FileDataModel(new File("data/Fit.csv"));
        DataModel model1 = new FileDataModel(new File("data/m1_base.csv"));
        DataModel model2 = new FileDataModel(new File("data/m2_base.csv"));
        DataModel model3 = new FileDataModel(new File("data/m3_base.csv"));
        DataModel model4 = new FileDataModel(new File("data/m4_base.csv"));
        DataModel model5 = new FileDataModel(new File("data/m5_base.csv"));

        //RecommenderEvaluator//////////////////////////////////////
        RecommenderEvaluator evaluator = new RMSRecommenderEvaluator();
        RecommenderEvaluator evaluatara = new AverageAbsoluteDifferenceRecommenderEvaluator();

        //RecommenderBuilder//////////////////////////////////////
        RecommenderBuilder knn_Builder = new KnnPearson();
        Recommender r = knn_Builder.buildRecommender(model1);

        // method: Pearson, algorithm: Pearson Correlation, description: from similar items
        double knn1 = evaluator.evaluate(knn_Builder, null, model1, 0.7, 1.0);
        double knn2 = evaluator.evaluate(knn_Builder, null, model2, 0.7, 1.0);
        double knn3 = evaluator.evaluate(knn_Builder, null, model3, 0.7, 1.0);
        double knn4 = evaluator.evaluate(knn_Builder, null, model4, 0.7, 1.0);
        double knn5 = evaluator.evaluate(knn_Builder, null, model5, 0.7, 1.0);
        double knn6 = evaluator.evaluate(knn_Builder, null, model6, 0.8, 1.0);

        // method: Cosine, algorithm: Cosine similarity, description: from similar items
        RecommenderBuilder cos_Builder = new CosineSimilarity();
        double cos1 = evaluator.evaluate(cos_Builder, null, model1, 0.7, 1.0);
        double cos2 = evaluator.evaluate(cos_Builder, null, model2, 0.7, 1.0);
        double cos3 = evaluator.evaluate(cos_Builder, null, model3, 0.7, 1.0);
        double cos4 = evaluator.evaluate(cos_Builder, null, model4, 0.7, 1.0);
        double cos5 = evaluator.evaluate(cos_Builder, null, model5, 0.7, 1.0);
        double cos6 = evaluator.evaluate(cos_Builder, null, model6, 0.8, 1.0);

        // method: baseline, algorithm: Baseline, description: User and item averages
        RecommenderBuilder bas_Builder = new Baseline();
        double bas1 = evaluator.evaluate(bas_Builder, null, model1, 0.7, 1.0);
        double bas2 = evaluator.evaluate(bas_Builder, null, model2, 0.7, 1.0);
        double bas3 = evaluator.evaluate(bas_Builder, null, model3, 0.7, 1.0);
        double bas4 = evaluator.evaluate(bas_Builder, null, model4, 0.7, 1.0);
        double bas5 = evaluator.evaluate(bas_Builder, null, model5, 0.7, 1.0);
        double bas6 = evaluator.evaluate(bas_Builder, null, model6, 0.8, 1.0);
    }
}

```

Slika 26 Primjer Java koda

```

RecommenderBuilder knn_Builder = new KnnPearson();
Recommender r_knn = knn_Builder.buildRecommender(model7);
List<RecommendedItem> recommendations1 = r_knn.recommend(5, 6);
for (RecommendedItem recommendation : recommendations1) {
    System.out.println(recommendation.getItemID()+"\t"+recommendation.getValue());
}
    System.out.println("\n");

// method: Pearson, algorithm: Pearson Correlation, description: from similar items
double knn1 = evaluator.evaluate(knn_Builder, null, model1, 0.7, 1.0);
double knn2 = evaluator.evaluate(knn_Builder, null, model2, 0.7, 1.0);
double knn3 = evaluator.evaluate(knn_Builder, null, model3, 0.7, 1.0);
double knn4 = evaluator.evaluate(knn_Builder, null, model4, 0.7, 1.0);
double knn5 = evaluator.evaluate(knn_Builder, null, model5, 0.7, 1.0);
double knn6 = evaluator.evaluate(knn_Builder, null, model6, 0.7, 1.0);

// method: Cosine, algorithm: Cosine similarity, description: from similar items

RecommenderBuilder cos_Builder = new CosineSimilarity();
Recommender r_cos = cos_Builder.buildRecommender(model1);
List<RecommendedItem> recommendations2 = r_cos.recommend(5, 6);
for (RecommendedItem recommendation : recommendations2) {
    System.out.println(recommendation.getItemID()+"\t"+recommendation.getValue());
}
    System.out.println("\n");

double cos1 = evaluator.evaluate(cos_Builder, null, model1, 0.7, 1.0);
double cos2 = evaluator.evaluate(cos_Builder, null, model2, 0.7, 1.0);
double cos3 = evaluator.evaluate(cos_Builder, null, model3, 0.7, 1.0);
double cos4 = evaluator.evaluate(cos_Builder, null, model4, 0.7, 1.0);
double cos5 = evaluator.evaluate(cos_Builder, null, model5, 0.7, 1.0);
double cos6 = evaluator.evaluate(cos_Builder, null, model6, 0.7, 1.0);

// method: baseline, algorithm: Baseline, description: User and item averages

RecommenderBuilder bas_Builder = new Baseline();
Recommender r_bas = bas_Builder.buildRecommender(model1);
List<RecommendedItem> recommendations3 = r_bas.recommend(5, 1122);
for (RecommendedItem recommendation : recommendations3) {
    System.out.println(recommendation.getItemID()+"\t"+recommendation.getValue());
}
    System.out.println("\n");
double bas1 = evaluator.evaluate(bas_Builder, null, model1, 0.7, 1.0);
double bas2 = evaluator.evaluate(bas_Builder, null, model2, 0.7, 1.0);

```

*Slika 27 Preporuke i procjena (Java code)*

```

648
649 // SlopeOne////////////////////////////////////
650 class SlopeOne implements RecommenderBuilder {
651     @SuppressWarnings("deprecation")
652     public Recommender buildRecommender(DataModel dataModel)
653         throws TasteException {
654         DiffStorage diffStorage = new MemoryDiffStorage(dataModel, Weighting.UNWEIGHTED, Long.MAX_VALUE);
655         return new SlopeOneRecommender(dataModel, Weighting.UNWEIGHTED, Weighting.UNWEIGHTED, diffStorage);
656     }
657 }
658
659 // Singular Value Decomposition - ALSWRFactorizer - 10
660 // Features////////////////////////////////////
661 class SVD1 implements RecommenderBuilder {
662     public Recommender buildRecommender(DataModel dataModel)
663         throws TasteException {
664         return new SVDRecommender(dataModel, new ALSWRFactorizer(dataModel, 10,
665             0.05, 10));
666     }
667 }
668
669 // Singular Value Decomposition - ALSWRFactorizer - 20
670 // Features////////////////////////////////////
671 class SVD2 implements RecommenderBuilder {
672     public Recommender buildRecommender(DataModel dataModel)
673         throws TasteException {
674         return new SVDRecommender(dataModel, new ALSWRFactorizer(dataModel, 20,
675             0.08, 10));
676     }
677 }
678
679 // Singular Value Decomposition - SVDPlusPlusFactorizer - 10
680 // Features////////////////////////////////////
681 class SVD3 implements RecommenderBuilder {
682     public Recommender buildRecommender(DataModel dataModel)
683         throws TasteException {
684         return new SVDRecommender(dataModel, new SVDPlusPlusFactorizer(
685             dataModel, 10, 10));
686     }
687 }
688
689 // Singular Value Decomposition - SVDPlusPlusFactorizer - 10
690 // Features////////////////////////////////////

```

*Slika 28 Implementacija algoritama (Java code)*

## POPIS SLIKA:

Slika 1 Kombinacija vrijeme/mjesto komunikacijskih modela i kolaborativnih tehnologija .....	10
Slika 2 Kolaborativno učenje .....	18
Slika 3 Koncept kolaborativnog učenja .....	19
Slika 4 MediaWiki - logo .....	21
Slika 5 DokuWiki - logo .....	22
Slika 6 TermWiki – logo .....	22
Slika 7 WikiEduktor - logo.....	23
Slika 8 Chiq Chaq - logo .....	24
Slika 9 PmWiki - logo.....	24
Slika 10 FosWiki - logo .....	25
Slika 11 Math Wiki .....	25
Slika 12 Inteligentni programski agent .....	28
Slika 13 Jednostavni refleksivni agenti .....	31
Slika 14 Refleksivni agenti bazirani na modelu .....	32
Slika 15 Agenti bazirani na cilju .....	32
Slika 16 Agenti bazirani na korisnosti .....	33
Slika 17 Učeći agenti .....	34
Slika 18 Slojevi sistema preporuke .....	50
Slika 19 Baza podataka Fit_CKMS .....	62
Slika 20 Klasa DataConvert .....	70
Slika 21 SQL Upit - Spremanje podataka FITCKM .....	70
Slika 22 Simulacija programskog agenta .....	71
Slika 23 FITCKMS – Modul Wiki (Članak).....	106
Slika 24 FITCKMS - Modul Pitanja i odgovori.....	107
Slika 25 FITCKMS - Korisnički bedževi.....	108
Slika 26 Primjer Java koda.....	109
Slika 27 Preporuke i procjena (Java code) .....	110
Slika 28 Implementacija algoritama (Java code) .....	111

## POPIS TABELA:

Tabela 1 Težine bedževa .....	68
Tabela 2 Sistemi preporuke .....	72
Tabela 3 Preporučeni resursi (FITCKM baza podataka) .....	77
Tabela 4 Preporučeni resursi (MovieLens podskup d1) .....	78
Tabela 5. RMSE vrijednosti sistema preporuke .....	79
Tabela 6 Statističke vrijednosti sistema preporuke .....	80
Tabela 7. Komparacija rezultata preporuke standardnih metoda nad različitim skupovima iz baze podataka FITCKMS .....	82
Tabela 8 Brzina izvođenja standardnih metoda (skup "FitPonder.csv") .....	85
Tabela 9 Brzina izvođenja standardnih metoda (MovieLens) .....	86

## POPIS GRAFIKONA:

Grafikon 1 Minimalna i maksimalna odstupanja od srednje RMSE vrijednosti .....	81
Grafikon 2 Brzina izvođenja standardnih metoda (skup "FitPonder.csv") .....	86
Grafikon 3 Brzina izvođenja standardnih (MovieLens) .....	87

## POPIS SKRAĆENICA:

LMS – Learning Management System  
VLE – Virtual Learning Environment  
KMS - Knowledge Management System  
CIS - Collaborative Information Systems  
CKMS - Collaborative Knowledge Management System  
SN – Social Network  
VO – Virtual Community  
GSS - Group Support Systems  
HTML - Hypertext Markup Language  
RDF - Resource Description Framework  
XML - Extended Markup Language  
CIS - Collaborative Information Systems

OL - Organizational Learning

OM - Organizational Memory

KR - Knowledge Repository

KDQL - Knowledge Discovery Query Language

ICT - Information and Communication Technology

CGI - Computer-generated imagery

AR - adaptive recommenders

QA – Questions and Answers

KM - Knowledge Management

DM - Data Mining