UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Levičnik

Priporočilni sistem za izbiro turističnih svetovalcev

DIPLOMSKO DELO

NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: izr. Prof. Marko Robnik Šikonja

Ljubljana, 2013

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00423/2013
Datum: 08.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANDREJ LEVIČNIK**

Naslov: **PRIPOROČILNI SISTEM ZA IZBIRO TURISTIČNIH SVETOVALCEV**

RECOMMENDER SYSTEM FOR SELECTION OF TOURISTIC ADVISORS

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Priporočilni sistemi skušajo zajeti uporabnikove preference in mu svetovati pri izbiri. Tipično skušajo profilirati uporabnike na podlagi njihovega preteklega delovanja in jim ponuditi s tem skladno rangirano množico izbir.
Proučite področje priporočilnih sistemov in izdelajte prototip spletnega sistema za izbiro turističnega svetovalca. Profile uporabnikov izdelajte glede na njihove označene preference in glede na aktivnosti na spletnem omrežju Facebook, profile ponudnikov turističnega svetovanja pa izdelajte glede na ponujene storitve, dosedanjo zadovoljstvo uporabnikov in aktivnosti na spletnem omrežju. Spletni sistem naj uporabnikom ponudi rangirano izbiro svetovalcev. Sistem ovrednotite na simuliranih podatkih tipičnih uporabnikov in svetovalcev.

Mentor:
izr. prof. dr. Marko Robnik Šikonja

Dekan:
prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani Andrej Levičnik,

z vpisno številko 63060186,

sem avtor diplomskega dela z naslovom:

Priporočilni sistem za izbiro turističnih svetovalcev

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom

  izr. prof. Marka Robnika Šikonje,

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela

- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 25.9.2013 Podpis avtorja: _____

Zahvala

Hvala mentorju za pomoč in vodenje pri izdelavi diplomskega dela.

Hvala družini za neizmerno podporo med tekom študija.

Hvala vsem, ki ste mi pri izdelavi diplomske naloge kakorkoli pomagali.

Kazalo vsebine

Kazalo slik

Kazalo grafov

Kazalo tabel

Seznam uporabljenih kratic in simbolov

API – Application Programming Interface

ORM – Object-relational Mapper

HTML – Hyper Text Markup Language

SDK – Software Development Kit

MAE – Mean Absolute Error

TF – Term Frequency

IDF – Inverse Document Frequency

POVZETEK

Izdelali smo prototip priporočilnega sistema, ki bo turistom svetoval pri izbiri turističnega vodiča. V ta namen smo preučili področje priporočilnih sistemov in predstavili tipična pristopa, na podlagi katerih se priporočilni sistemi odločajo.

Priporočilni sistem uporabnike profilira na podlagi njihove aktivnosti na družbenem omrežju Facebook. Razvili smo algoritem za zajemanje tekstovnih vsebin iz Facebooka ter algoritem, ki uporablja jezikovne tehnologije. Razvili smo tudi funkcijo za pretvorbo besedila v vektorsko obliko po TF-IDF shemi in primerjalno funkcijo, ki uporablja kosinusno podobnost.

Sistem smo evalvirali s pomočjo ročno določene referenčne ocene. Rezultati kažejo močno korelacijo ocen razvitega priporočilnega sistema z referenčnimi.

Ključne besede: priporočilni sistemi, Facebook, jezikovne tehnologije, TF-IDF, kosinusna podobnost

ABSTRACT

We created a prototype recommender system, advising tourists on tourist guide choice. We examined the area of recommender systems and described two typical approaches to recommender systems design.

Our recommender system profiles users based on their activity on social network Facebook. We developed an algorithm for retreival of textual data from Facebook and an algorithm for its exploitation via language technologies. We also used a function for translation of text into vector form using TF-IDF schema and cosine similarity function.

The system was evaluated using manual recommendations. The results show a strong correlation between recommended guides and manually picked choices.

Key words: recommender systems, Facebook, natural language processing tools, TF-IDF, cosine similarity

1. Uvod

Turistom želimo ponuditi alternativne in pristne aktivnosti v obiskanem kraju. To so aktivnosti, ki jih turistične agencije ne zajemajo, na primer rolkanje po zanimivih predelih Ljubljane. Ponudba je namenjena je turistom, ki bi radi izkusili obiskani kraj skozi oči domačina.

Razvita spletna platforma domačinom omogoča objavljanje aktivnosti, na katere se lahko turisti prijavijo, kar lahko poveča prihodke skozi večje število izvedenih aktivnosti. Eden od načinov za povečanje rezervacij aktivnosti je priporočilni sistem, ki turistom predlaga vsebine, ki bi jih utegnile najbolj zanimati. Turisti tako lažje najdejo zanje zanimive aktivnosti.

Predmet diplomske naloge je izdelava prototipa priporočilnega sistema, ki turistom pomaga pri izbiri aktivnosti. V ta namen bomo preučili tipične priporočilne sisteme.

Določili bomo zasnovo svojega priporočilnega sistema. Definirali bomo glavno nalogo sistema, na podlagi katere bomo izbrali primerno metodo profiliranja. Uporabnikove preference, s katerimi sistem predlaga vsebine, bomo zajeli s pomočjo spletnega omrežja Facebook v obliki tekstovnih vsebin. Odločili se bomo kako tekstovne vsebine primerjati. Prav tako bomo definirali diagram poteka priporočilnega sistema.

Prototip bo zasnovan s pomočjo odprtokodnih programskih orodij, katera temeljijo na programskem jeziku Python. Pogledali bomo s katerimi knjižnicami, katere so v Pythonu precej priljubljene, si lahko pomagamo.

Pri razvoju prototipa bomo implementirali:

- zajemanje tekstovnih vsebin, ki so na uporabnikovem Facebook profilu,
- jezikovne tehnologije, ki se ukvarjajo s procesiranjem naravnega jezika,
- predstavitev tekstovnih vsebin v priporočilnem sistemu,
- primerjalno funkcijo, na podlagi katere bo priporočilni sistem turistu pomagal pri izbiri aktivnosti.

Učinkovitost priporočilnih sistemov je treba vseskozi preverjati, zato bomo prototip uporabili na testni množici tipičnih uporabnikov in ocenili njegovo učinkovitost.

2. Priporočilni sistemi

Priporočilni sistemi skušajo napovedati oceno elementa, kakor bi ga ocenil uporabnik. Na podlagi ocen uporabniku priporočijo stvari, ki bi ga utegnile zanimati.

V zadnjih letih so priporočilni sistemi postali zelo pogosti. Pomembno vlogo imajo na spletnih straneh kot so:

- eBay(http://www.ebay.com),
- Amazon (http://www.amazon.com),
- Netflix (http://www.netflix.com),
- Youtube (http://www.youtube.com).

Priporočilni sistem je tipično zasnovan na dva načina. Prvi pristop uporablja metodo izbiranja s sodelovanjem, drugi pa vsebinsko osnovano metodo.

2.1. Metoda izbiranja s sodelovanjem

Metoda izbiranja s sodelovanjem predvidevaprihodnje izbire na podlagi preteklih. Napoved je tako odvisna od uporabnikove podobnosti z drugimi uporabniki.

Za primer vzemimo osebi A in B, ki kupujeta v spletni trgovini, ki ima implementiran priporočilni sistem z metodo izbiranja s sodelovanjem. Obe osebi sta v preteklosti kupovali oblačila znamke Z. Oseba A je pred kratkim kupila športne copate Y. Naslednjič, ko oseba B obišče spletno trgovino, ji bo priporočilni sistem predlagal športne copate, ki jih je kupila oseba A.

Metoda torej temelji na zbiranju informacij o uporabnikih. Nabiranje informacij ločimo na dva tipa, eksplicitno in implicitno nabiranje.

Pri eksplicitnem nabiranju informacij zahtevamo od uporabnika, da oceni vsebino, razvrsti vsebine po preferenci, itd. Pri implicitnem nabiranju informacij opazujemo, kolikokrat je uporabnik pogledal določeno vsebino, beležimo pretekle nakupe uporabnika itd.

Glavna prednost te metode je, da je neodvisna od vsebine, ki jo napoveduje.

Slabe lastnosti metode izbiranja s sodelovanjem so naslednje [1]:

- počasen začetek (angl. cold start): metoda potrebuje veliko informacij o uporabniku za dobro priporočilo,
- draga širitev: v mnogih okoljih, kjer je implementiran tak sistem, je ogromno uporabnikov in izdelkov, zato je potrebna velika računska moč za izračun priporočila,
- pičlost: Veliko število izdelkov pomeni, da bodo uporabniki ocenili le majhno podmnožico izdelkov. Priporočilo tako ne bo optimalno.

2.2. Vsebinsko osnovana metoda

Vsebinsko osnovana metoda temelji na informacijah o vsebinah, ki jih ponuja. Metoda torej napove vsebine, ki so podobne uporabniku všečnim vsebinam.

Za primer vzemimo osebo A, ki nakupuje v spletni trgovini s priporočilnim sistemom z vsebinsko osnovano metodo. Oseba A je ljubiteljica igranja športnih iger na konzoli in tako opravi nakup simulacije s formulo 1. Pri naslednjem obisku v spletni trgovini, ji bo priporočilni sistem predlagal igre podobne simulaciji formule 1.

Glavni problem metode nastopi je napovedovanje različnih tipov vsebine [2]. Sistem je treba naučiti, da uporabnikove preference, ki se jih nauči iz enega tipa vsebine, uporabi pri priporočilih vsebin drugih tipov.

3. Zasnova priporočilnega sistema

Cilj našega priporočilnega sistema je ponuditi turistu njemu prilagojeno ponudbo.

Glavna naloga sistema je napoved ocen turističnih vodičev glede na profil turista, in jih pretvoriti v seznam najbolj ustreznih vodičev.

Odločili smo se za vsebinsko osnovano metodo. Ker ponujamo samo en tip vsebine, pomanjkljivost te metode ne pride do izraza. Na spletni platformi ShowMeAround [3] je premalo uporabnikov, zato bi sodelovalno izbiranje imelo težavo s počasnim začetkom.

Sistem bo profiliral vodiče in turiste glede na preference, ki jih preberemo iz njihovih Facebook profilov. Profiliranje bo temeljilo na tekstovnih vsebinah.

Shemo podatkovne baze bomo nastavili tako, da enega uporabnika predstavljajo osebni podatki, kot sta ime ter priimek, besedilo, sestavljeno iz tekstovnih vsebin in spletne povezave do teh vsebin.

Besedila, ki pripadajo uporabnikom, lahko primerjamo na več nivojih. Odločili smo se, da bomo besedila primerjali na tematskem nivoju. Za tako primerjavo je treba besedila, preden jih shranimo v podatkovno bazo, obdelati.

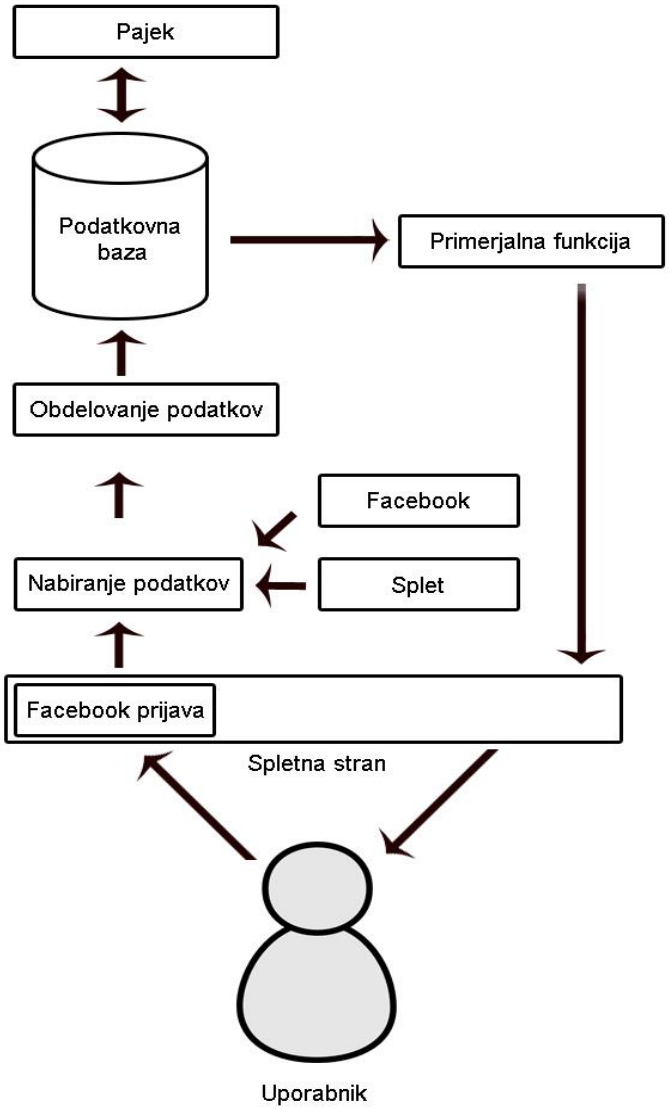
Za napoved ocen, ki jih generira priporočilni sistem, uporabimo primerjalno funkcijo. Primerjamo torej profila turista in vsakega vodiča posebej.

Za primerjalno funkcijo smo izbrali kosinusno podobnost, ta primerja dva vektorja in kot rezultat vrne kosinus kota, ki ga vektorja oklepata. Zato je treba besedila uporabnikov, ki jih primerjamo, spremeniti v vektorsko obliko. Za kakovostnejšo primerjavo, besedila spremenimo v vektorje po TF-IDF shemi.

Rezultat primerjalne funkcije uporabimo kot predvideno oceno vodiča. Ocene na koncu razvrstimo in vrnemo seznam najbolj ustreznih vodičev.

Vsebine na spletnih straneh se mnogokrat osvežujejo, zato bomo implementirali spletnega pajka, ki bo skrbel za ažurnost podatkov v podatkovni bazi.

Arhitektura priporočilnega sistema je prikazana na sliki 1.



Slika 1. Arhitektura priporočilnega sistema.

4. Programska orodja

V tem poglavju bomo podrobneje opisali programska orodja, ki smo jih uporabili za izdelavo prototipa priporočilnega sistema.

4.1. Python

Priporočilni sistem je zasnovan v programskem jeziku Python. Python je razširljiv odprtokodni programski jezik. Pri namestitvi programer privzeto dobi veliko knjižnic, ki so mu v pomoč pri razvoju. Poleg tega obstaja še veliko število drugih, ki jih lahko pridobi s spleta. Zaradi teh razlogov smo se odločili za ta programski jezik.

Python trenutno ločimo na dve verziji. Starejša verzija 2, na kateri temelji velika večina knjižnic in verzija 3, ki predstavlja prihodnost Pythona. Zaradi podpore knjižnic oz. lažjega razvoja smo izbrali verzijo 2.

V naslednjih poglavjih so podrobno opisane Python knjižnice in orodja, ki so bile uporabljena pri razvoju.

4.1.1. Pyfaceb

Pyfaceb [4] je odprtokodna Python knjižnica, namenjena komuniciranju s Facebook Graph in Facebook Query Language (FQL) vmesnikom. Razvijalcu omogoča preprosto izvajanje poizvedb na Facebookovih strežnikih.

4.1.2 NLTK

NLTK (Natural language tool kit) [5] je platforma, ki Python programerjem olajša delo z naravnim jezikom. Vsebuje zbirko knjižnic, ki poenostavijo obdelavo naravnega jezika.

NLTK je podporno orodje pri raziskavah ter pri poučevanju te tematike.

Priporočilni sistem izkorišča naslednje funkcionalnosti NLTK-ja:

- Krnjenje besed (angl. stemming): NLTK vsebuje »stem« paket namenjen krnjenju besed. Iz širokega nabora modulov smo izbrali PorterStemmer, ki je v veliko virih omenjen kot dobra privzeta izbira.
- Seznam nepomembnih besed: NLTK corpus module vsebuje funkcijo »stopword«, ki kot rezultat vrne seznam takšnih besed.
- Zajemanje vsebine iz spletnih strani: Funkcija nltk.html_clean() nam omogoča zajemanje besedila iz spletnih strani. Kot argument podamo spletno stran, kot rezultat pa dobimo naravno vsebino.

4.1.3. Microsoft Translator API

Microsoft Translator je spletni portal [6], ki uporabnikom omogoča prevanje besedil ali spletnih strani v različne jezike. Trenutno prevaja 40 jezikov. Iz nabora dodatnih funkcij, ki jih ponuja, omenimo samodejno zaznavanje jezika.

Funkcionalnost portala ni dostopna samo prek spletnega brskalnika. Razvijalcu je na voljo več vmesnikov(AJAX, SOAP, HTTP, ...). Odločili smo se za vmesnik HTTP, za lažjo izvedbo pa smo uporabili Microsoft Translator API za Python.

4.1.4. Langid

Langid [7] je python orodje za zaznavanje jezika besedila. Kot prednosti orodja avtor navaja:

- hitrost,
- odkrivanje mnogih jezikov (trenutno 97),
- minimalizem (samo ena python datoteka).

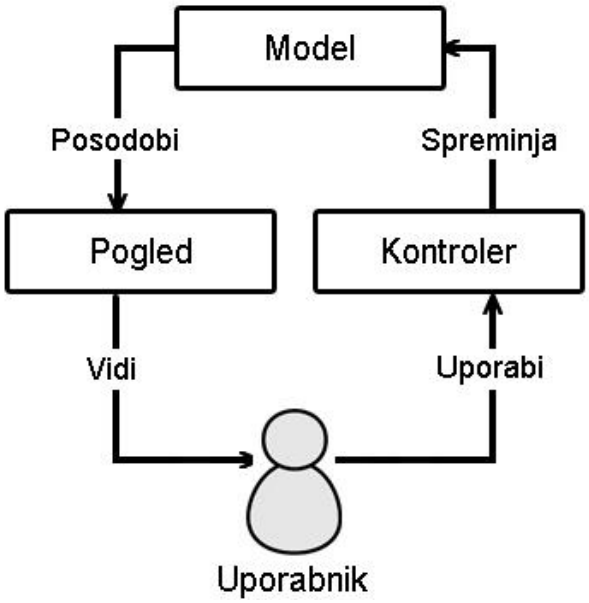
Langid se lahko uporablja tudi kot Python knjižnico, tega načina smo se poslužili tudi sami.

4.1.5. Django web framework

Django [8] je odprtokodno okolje za hiter razvoj spletnih aplikacij. Glavni cilj Djanga je olajšan razvoj kompleksnih in od podatkovnih baz odvisnih spletnih strani.

Okolje je napisano v Pythonu in sledi »Model-Pogled-Kontroler« arhitekturni logiki. Taka arhitektura narekuje, da projekt razdelimo na tri plasti, na katerih se izvajajo zanje značilne operacije.

V modelu so podatki, poslovna logika in funkcije. Pogled je namenjen predstavitvi podatkov. Kontroler pretvarja vhodne podatke v ukaze, ki jih nato posreduje modelu ali pogledu.



Slika 2. Primer koleboracije med posameznimi plastmi.

Ta zasnova prinaša prilagodljivost pri razvoju in implementaciji.

Glavne funkcionalnosti Djanga so naslednje:

- ORM (povezuje Python razrede in relacijsko podatkovno bazo),
- sistem za procesiranje zahtev s spletnimi predlogami,
- sistem za razpošiljanje, ki plastem omogoča medsebojno komunikacijo.

Django skupnost je velika, v primeru, da razvijalec naleti na napako, ki je ni zmožen rešiti sam, je odgovor možno hitro najti na spletu. Poleg tega Django krasi bogata dokumentacija. Zaradi teh razlogov smo se odločili za uporabo tega okolja.

4.2. SQLite3

SQLite3 [9] je sistem za upravljanje relacijskih podatkovnih baz. Sistem je skladen z ACID (Atomicity, Consistency, Isolation, Durability) lastnostmi. V nasprotju z drugimi sistemi za upravljanje podatkovnih baz, SQLite ni ločen proces, prek katerega bi uporabnik dostopal do podatkov, ampak sestavni del le-teh.

Zaradi majhne velikosti in preproste namestitve smo se odločili za SQLite.

Medtem, ko je za sam razvoj in testiranje SQLite zadosten, bi bilo za produkcijsko verzijo vredno preučiti druge rešitve.

Za branje in pisanje podatkov uporabljamo Django ORM, kar pomeni, da lahko brez težav zamenjamo tip podatkovne baze.

5. Prototip priporočilnega sistema

V naslednjih poglavjih so opisane ključne komponente našega priporočilnega sistema.

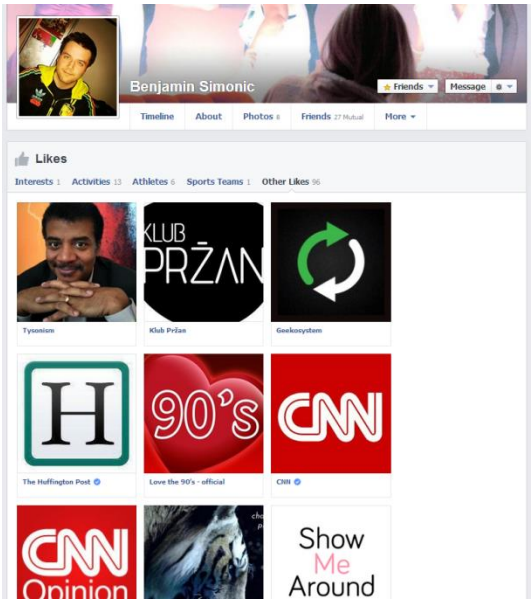
5.1. Facebook

5.1.1. O Facebooku

Facebook je družbena spletna stran. Uporabnikom omogoča, da se povežejo s stiki iz svojih omrežij (kot so npr. šola, mesto zaposlitve ali geografsko območje) in tako lažje komunicirajo z ostalimi ljudmi iz istega omrežja. Spletna stran je globalna, po celem svetu ima več kot 1,15 milijarde aktivnih uporabnikov. [10]

Uporabniki Facebooka si ustvarijo svoj profil. Na svoj profil dodajo prijatelje in s tem vidijo njihove profile. Profil lahko vsebuje tudi fotografije in seznam interesov.

Seznam interesov veliko pove o uporabniku. Npr. če je na seznamu veliko glasbenih skupin ter športnih aktivnosti, lahko predpostavimo, da je ta uporabnik ljubitelj glasbe in športa. Primer seznama interesov vidimo na sliki 3.



Slika 3. Primer seznama interesov uporabnika na Facebooku.

Vsak interes ima na Facebooku svojo stran, na kateri je njen daljši opis. Prisotne so lahko tudi povezave na relevantne strani na spletu. Slika 4 prikazuje primer take strani.

Slika 4. Primer strani interesa na Facebooku.

Naš priporočilni sistem predlaga vsebino uporabniku na podlagi podatkov, ki jih dobi na Facebook straneh njegovih zanimanj.

Podatki na Facebook strani so načeloma generični, sama stran pa lahko vsebuje tudi povezave na ostale strani. Priporočilni sistem bo uporabil podatke s Facebook strani in z vseh povezav na njej.

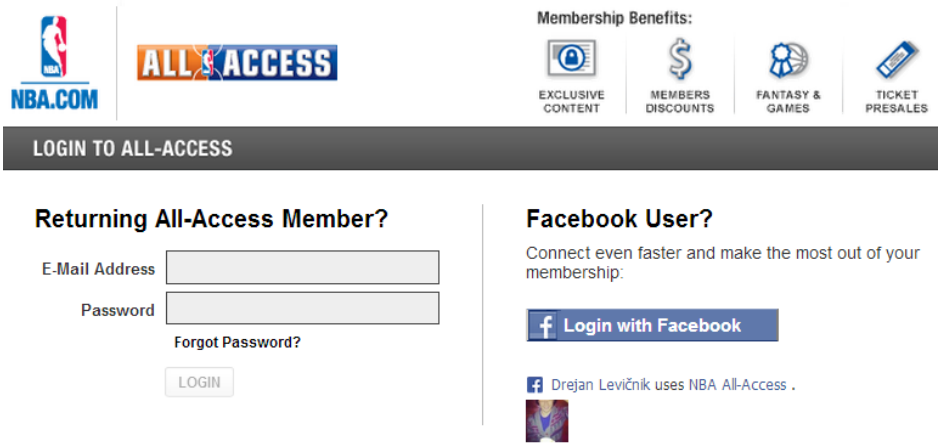
Predvidevamo, da povezave vodijo na relevantne spletne strani. Te so napisane v HTML jeziku. Vsaka spletna stran je sestavljena iz HTML elementov. Vsak element je predstavljen v obliki značk, zapisanih v oklepajih. Nov odstavek na primer predstavlja značka <p>.

Zanima nas samo vsebina, zato je potrebno izvzeti HTML označbe in upoštevati samo besedilo. To nam omogoča NLTK.

5.1.2. Facebook prijava

Facebook razvijalcem ponuja veliko orodij. Razvijalec lahko lastno spletno stran integrira s Facebookom. Tako lahko obiskovalcem ponudi gumb »Všeč mi je« neposredno, na svojem spletišču. Za nas zanimivo orodje je Facebook Login. To je način prijave v katerokoli spletno platformo. Od uporabnika ne zahteva kreiranja novega uporabniškega imena, gesla in ostalih osebnih podatkov. Vsi podatki se prenesejo s Facebook strežnika. Razvijalcu tako ni treba skrbeti za administracijo uporabnikov, saj vse poteka preko Facebooka.

Večini uporabnikov je izpolnjevanje obrazcev za registracijo odveč, zato se raje odločijo za tak način prijave, saj je hitrejša in ne zahteva pretirane interakcije.



Slika 5. Primer integracije Facebook prijave na spletni strani http://www.nba.com/.

Veliko razvijalcev spletnih strani se odloča za takšen način prijave, saj je varen, lahek za integracijo in zaobide programiranje administracije.

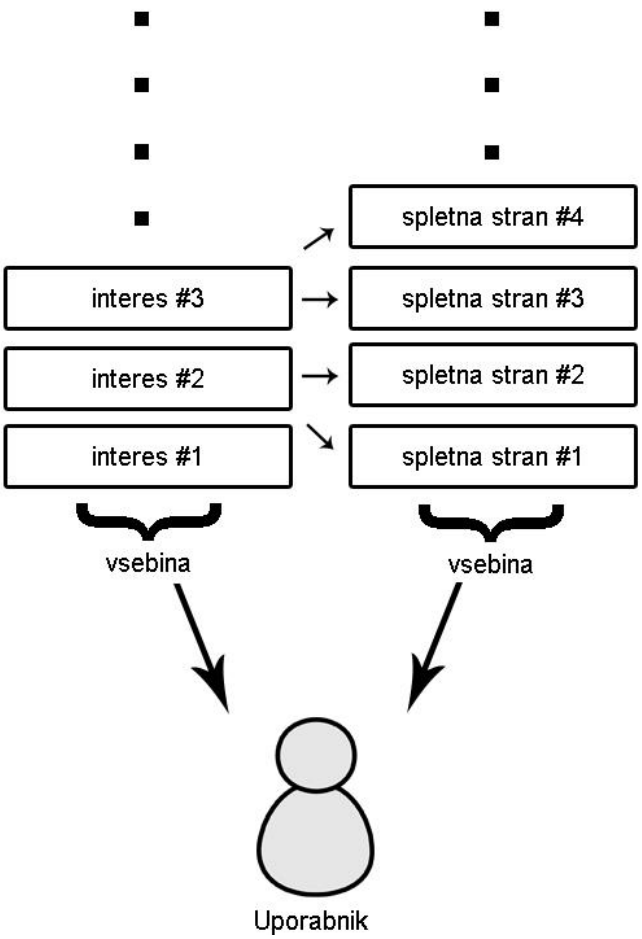
Integracija takšne prijave je preprosta. Potrebni so trije koraki, ki so opisani na Facebookovi strani za razvijalce:

- izpolniti formular, v katerem navedemo, katere informacije bomo zahtevali od uporabnika,
- pripeti namenski JavaScript SDK v HTML datoteko,
- po meri sprogramirati JavaScript datoteko, ki bo skrbela za prijavo.

Pri prijavi lahko od uporabnika zahtevamo precej podatkov, npr.: kraj bivanja, datum rojstva, delovno mesto, seznam prijateljev, seznam interesov, itd.

V našem primeru nas zanima uporabnikov seznam zanimanj. Facebook Login je tako osrednja točka našega priporočilnega sistema. Vsa priporočila, ki jih vračamo temeljijo na podatkih, pridobljenih na ta način.

5.2. Prevajanje besedila



Slika 6. Prikaz množice podatkov, ki pripadajo uporabniku.

Zgornja slika predstavlja uporabnika in njegove podatke, ki jih dobimo, ko se prijavi v našo spletno stran s Facebook Loginom.

Facebook je mednarodna spletna platforma. To pomeni, da so strani, ki opisujejo zanimanja, lahko napisane v kateremkoli jeziku.

Besedila v različnih jezikih je nemogoče neposredno primerjati, zato je treba poskrbeti, da je jezik v vseh besedilih poenoten. Odločili smo se za angleški jezik, saj so algoritmi za procesiranje naravnega jezika za ta jezik najbolj dovršeni in ker je besedil v angleškem jeziku največ.

Vsa besedila, ki niso v angleškem jeziku, je treba prevesti.

Prevajanje smo izvedli z Langid.py in Microsoft Translator API-jem.

5.3. Primerjanje besedil

Priporočilni sistem deluje tako, da podatke, ki pripadajo enemu uporabniku združi v celoto, ki jo shrani v podatkovno bazo. To celoto primerja z ostalimi in kot rezultat vrne najbolj podobne uporabnike.

Za predstavitev besedila v algoritmu priporočilnega sistema smo se poslužili modela »vreča besed« (angl. »bag of words«). Pri uporabi takega modela je besedilo predstavljeno kot množica neurejenih besed, slovnica ter vrstni red besed nista upoštevana. Tak način interpretacije besedila je pogosto uporabljen pri sistemih, ki se ukvarjajo s klasifikacijo dokumentov.

Z vidika takšnega modela sta naslednja stavka enaka:

Benjamin ima raje brokoli kot sladico.

Benjamin ima raje sladico kot brokoli.

Izvedbo primerjave lahko razbijemo na naslednje komponente:

- normalizacija besedila,
- pretvarjanje besedila v vektorsko obliko,
- primerjalna funkcija.

5.3.1. Normalizacija besedila

Za kakovostnejšo primerjavo besedil je posamezno besedilo treba normalizirati. Postopek normalizacije izvedemo v naslednjih korakih:

- iz besedila odstranimo vse nepotrebne besede in ločila,
- velike črke pretvorimo v male,
- krnimo besedilo.

5.3.1.1. Odstranjevanje nepotrebnih besed in ločil

Treba je izločiti vse nepotrebne besede. Nepotrebne besede pogosto nimajo leksikalnega pomena in služijo za izražanje slovničnih odnosov z drugimi besedami znotraj stavka. Pod nepotrebne besede štejemo predloge, zaimke, pomožne glagole, veznike in členke.

V našem algoritmu je besedilo predstavljeno kot množica besed. Množico nepomembnih besed pa pridobimo s pomočjo NLTK-ja. V to množica dodamo ločila, ki tako kot nepomembne besede ne prinašajo mnogo k tematiki besedila.

Razlika teh množic je besedilo brez nepomembnih besed in ločil.

5.3.1.2. Pretvarjanje velikih črk v male

Slovnična pravila zahtevajo, da je besedilo razčlenjeno na stavke, vsak stavek pa se začne z veliko začetnico.

Pri modelu »vreča besed« slovnice besedila in vrstnega reda besed ne upoštevamo, dobimo pa besedilo, kjer so nekatere besede zapisane z veliko začetnico.

Za boljšo primerjavo je tako treba vse besede predstaviti na enak način. Odločili smo se, da velike začetnice spremenimo v male.

5.3.1.3. Krnjenje besedila

Da bi izboljšali primerjavo, je posamezne besede v besedilu treba krniti. Krnjenje besede je postopek, pri katerem besedo preoblikujemo v njen koren. Na primer beseda koreniti se na ta način preoblikuje v besedo koren. S krnjenjem občutno zmanjšamo nabor različnih besed.

Podoben postopek je lematizacija besed. Lematizacija je tesno povezana s krnjenjem. Razlika med postopkoma je v tem, da lematizacija besedo preoblikuje v njeno slovarsko obliko, medtem ko krnjenje besedi odreže končnico.

Ker je besedilo v angleškem jeziku, smo se odločili, za krnjenje. Pri morfološko bogatejših jezikih, kot je na primer slovenščina, se namesto krnjenja priporoča uporaba lematizacije.

5.3.2. Pretvarjanje besedila v vektorsko obliko

Za primerjanje besedil smo izbrali algoritem, ki računa podobnost na podlagi vektorjev, zato moramo pred primerjavo vsa besedila primerno strukturirati.

Besedilo predstavimo kot vektor. Vsaka beseda predstavlja eno dimenzijo vektorskega prostora. Za primer vzemimo naslednja dva stavka:

Benjamin ima rad čebulo. Ravno tako ima rad Mašo.

Benjamin ima prav tako rad ananas.

Na podlagi omenjenih stavkov zgradimo slovar besed.

{

»benjamin« : 1,

»ima« : 2,

»rad« : 3,

»čebulo« : 4,

»ravno« : 5,

»tako«: 6,

»Mašo«: 7,

»prav«: 8,

»ananas«: 9

}

Slovar vsebuje vse različne besede obeh stavkov, v tem primeru jih je 9. Z uporabo slovarja lahko stavka predstavimo kot 9 mestni vektor.

[1,2,2,1,1,1,1,0,0]

[1,1,1,0,0,1,0,1,1]

Zaporedna številka polja predstavlja besedo v slovarju. Vrednost v polju predstavlja število pojavitev te besede.

V praksi posamezne besede utežujemo, kot najbolj popularno shemo uteževanja, je treba omeniti TF-IDF (angl. term frequency–inverse document frequency). Te metode smo se poslužili tudi mi.

5.3.2.1. TF-IDF

TF-IDF predstavlja frekvenco, ki kaže, kako pomembna je beseda, ki se pojavi v besedilu, gledano na celoten korpus besedil. [11]

TF (angl. term frequency) je merilo pojavitve besede v besedilu. Obstajajo različni načini za določitev TF vrednosti. Izbrali smo najbolj osnovnega, preprosto štetje pojavitve besed v dokumentu, ta je opisan v prejšnjem poglavju. Za naš primer torej velja naslednja enačba, kjer t predstavlja besedo, d pa besedilo.

$$tf(t,d) = f(t,d)$$

IDF (angl. inverse document frequency) je merilo prisotnosti besede v vseh besedilih. IDF je logaritem kvocienta vseh besedil in števila besedil, ki vsebujejo besedo. V spodnji enačbi t predstavlja besedo, D pa število vseh besedil.

$$idf(t,D) = log\frac{|D|}{|\{d \in D : t \in d\}|}$$

TF-IDF je produkt obeh faktorjev:

$$tfidf(t,d,D) = f(t,d) \times log\frac{|D|}{|\{d \in D : t \in d\}|}$$

5.3.3. Primerjalna funkcija: kosinusna podobnost

Besedili v vektorski obliki primerjamo s primerjalno funkcijo. Izbrali smo kosinusno podobnost dveh vektorjev. Rezultat kosinusne podobnosti je kosinus kota med dvema vektorjema.

Vektorja z enako smerjo imata kosinusno podobnost 1, vektorja, ki ležita diametralno nasprotno, imata podobnost -1.

Pri pretvarjanju besedil v vektorje smo vsak vektor definirali v pozitivnem prostoru (pojavitev besed v besedilu ne more biti negativna), to pomeni, da kot med vektorjema ne more biti večji kot 90° in tako je rezultat kosinusne podobnosti vedno na intervalu od 0 do 1.

Enačbo za podobnost dveh vektorjev izpeljemo iz enačbe za skalarni produkt (vsi vektorji izhajajo iz iste točke).

$$A \cdot B = \left||A|\right| \left||B|\right| cos\,\theta$$

Če vzamemo vektorja besedili A in B ter število vseh besed n, je enačba za izračun kosinusne podobnosti naslednja:

$$podobnost = cos(\theta) = \frac{A \cdot B}{\left||A|\right| \left||B|\right|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

5.4. Pajek

Da bi bili podatki, s katerimi sistem priporoča vsebino, ažurni, uporabljamo spletnega pajka. Spletni pajek je robot, ki opravlja avtomatizirane naloge na spletu.

Shema podatkovne baze je definirana tako, da vsakemu uporabniku poleg besedila, pripada tudi polje, v katerem so našteti vsi spletni naslovi, na podlagi katerih je sestavljeno besedilo.

Spletni pajek se sprehodi čez vse povezave uporabnikov in prebere vsebino. Vsebino procesira in osveži podatkovno bazo.

Ta proces se ne zgodi v trenutku, zato se pajek zaganja periodično, v našem primeru enkrat tedensko. To dosežemo z razporejevalcem nalog, ki je del vsakega operacijskega sistema.

6. Integracija priporočilnega sistema v spletno platformo

Priporočilni sistem bomo s časoma integrirali v spletno platformo ShowMeAround [1].

Pri razvoju spletne platforme, se sledi metodologiji »vitkega podjetja« [12].

Glavna značilnost te metodologije so kratke iteracije pri razvoju produkta. Ena iteracija je sestavljena iz treh faz: spreminjanje produkta, merjenje učinkov in učenja na podlagi učinkov. Pri razvoju produkta se tako ognemo programiranju funkcionalnosti produkta, ki jih uporabniki ne bi uporabljali.

Na začetku razvoja produkta je potrebno določiti najbolj nujne funkcionalnosti in jih zatem sproti nadgrajevati. Začetna faza produkta se imenuje MVP (angl. minimum viable product) in ShowMeAround je trenutno v tej fazi. Slika 7 prikazuje vstopno stran spletne platforme ShowMeAround.

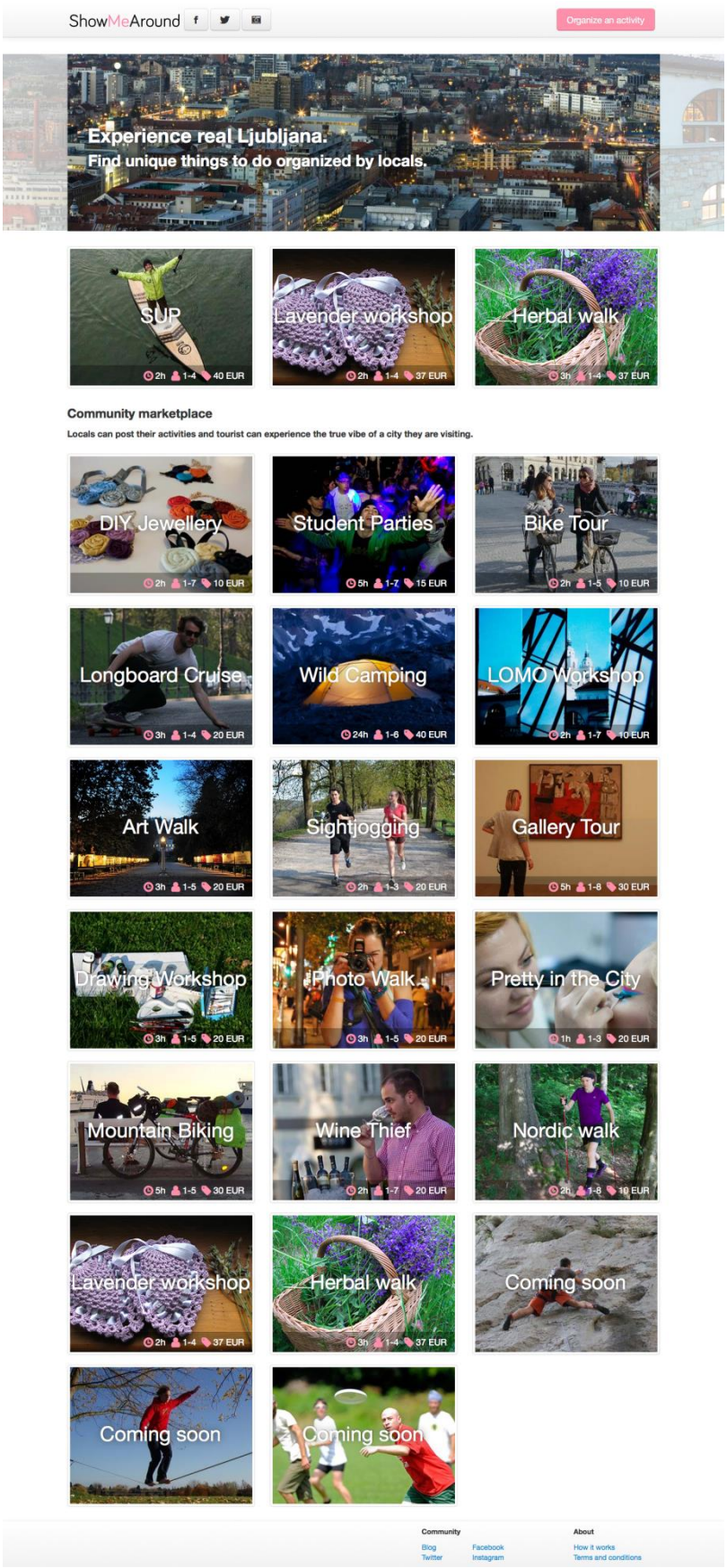
Veliko uporabnikov je izrazilo željo po uporabniških računih, zato bo potrebna nadgradnja sistema. Poleg običajnega kreiranja uporabniškega imena in gesla bomo omogočili tudi Facebook prijavo, ki je osrednji del priporočilnega sistema.

Sočasno z nadgradnjo bomo integrirali priporočilni sistem.

Pri razvoju priporočilnega sistema smo načrtno izbrali programska orodja, s katerimi bo integracija v spletno platformo najlažja. Tako sta priporočilni sistem in spletna platforma razvita, v programskem jeziku Python. Pri obeh smo uporabili razvojno okolje Django.

Učinke integracije priporočilnega sistema bomo merili s pomočjo storitve Google Analytics [13]. Storitev prikazuje statistiko obiska na spletni platformi. Integracija te storitve je preprosta, v spletno stran je potrebno vključiti namensko JavaScript datoteko. Za vmesnik, ki omogoča vpogled v statistiko, poskrbi Google.

Merili bomo korelacijo med številom obiska in številom rezervacij aktivnosti, pred in po integraciji priporočilnega sistema.



Slika 7. Spletna platforma ShowMeAround.

7. Evalvacija

Kakovosten priporočilni sistem odlikujejo dobre napovedi. To je posebej pomembno v našem primeru, saj priporočilni sistem neposredno vpliva na odločitve uporabnikov in posledično na prihodke podjetja. Za čim boljše napovedi je potrebno priporočilni sistem optimizirati. To storimo na podlagi evalvacije.

7.1. Evalvacijska metoda

Evalvacijske meritve za priporočilne sisteme lahko razdelimo na štiri glavne razrede [14]:

1. meritev natančnosti napovedi (angl. predictive accuracy metrics),
2. meritev natančnosti klasifikacije (angl. classification accuracy metrics),
3. meritev natančnosti razvrstitve (angl. rank accuracy metrics),
4. meritve, ki se ne ukvarjajo z natančnostjo (angl. non-accuracy metrics).

Meritev natančnosti napovedi nam pove, kako blizu so ocene, ki jih napovesta priporočilni sistem in uporabnik. Ta metoda je najbolj pogosta pri evalvaciji priporočilnih sistemov, saj je lahka za izračun in razumevanje.

Meritev natančnosti klasifikacije meri število pravilnih in nepravilnih klasifikacij kot relevantne oz. nerelevantne elemente, ki jih uporabniku posreduje priporočilni sistem.

Meritev natančnosti razvrstitve meri zmožnost priporočilnega sistema pri razvrščanju elementov v seznamu glede na uporabnika. Taka meritev upošteva samo vrstni red elementov neodvisno od ocene podobnosti.

V določenih primerih rezultati priporočilnega sistema niso najboljši, kljub temu, da so optimalno izračunani. Včasih uporabniku bolj pomaga priporočilo, ki je neodvisna od njene natančnosti. [15]

Odločili smo se za meritev natančnosti napovedi. Merili smo, kako blizu so si ocene, ki jih da priporočilni sistem in prave ocene. Oceno smo definirali kot realno število z eno decimalko, ki je na intervalu od 0 do 10.

Ocena, ki jo poda sistem, predstavlja kosinusno podobnost interesov dveh oseb, pomnoženo s številom deset. Pravo oceno določimo gleda na lastno poznavanje oseb.

7.2. Testna množica

Za testno množico smo izbrali dvajset ljudi. Za vsakega izmed njih smo sestavili dva seznama, v njuju so bile ocene podobnosti ostalim ljudem iz množice.

V prvem seznamu so ocene, ki jih izračuna priporočilni sistem, v drugem pa ročno določene vrednosti.

Testno množico sestavljajo naslednje osebe:

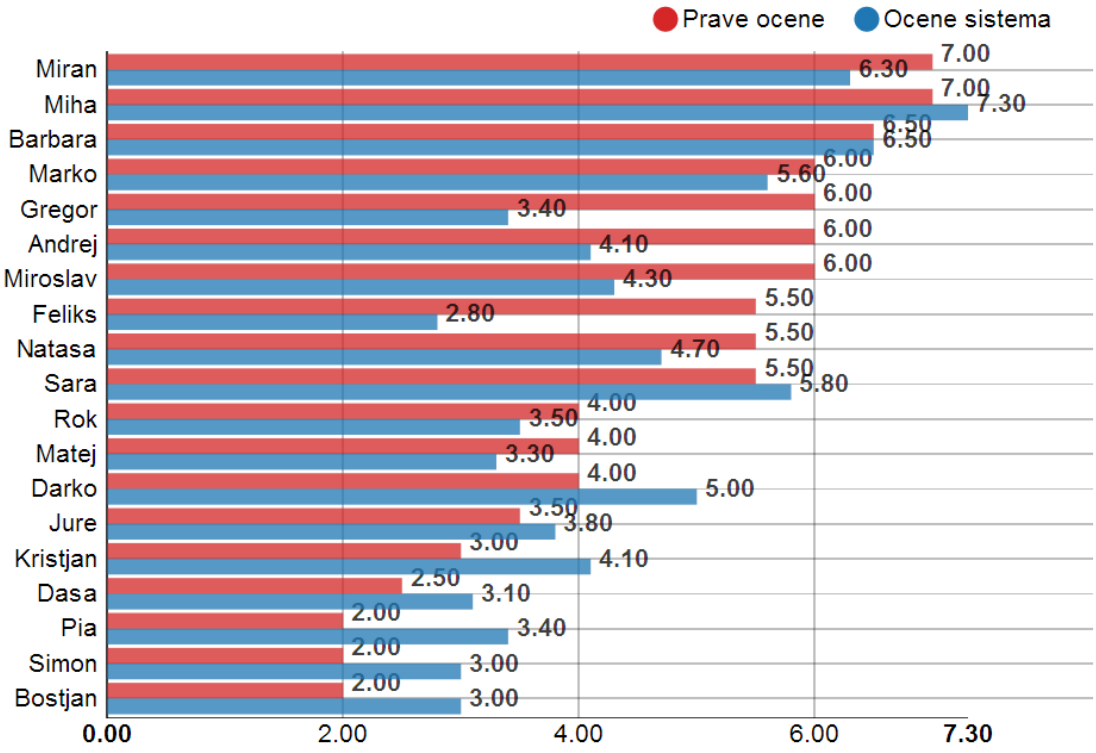
{ Nejc, Kristjan, Sara, Miroslav, Darko, Andrej, Barbara, Boštjan, Matej, Simon, Daša, Pia, Nataša, Rok, Feliks, Gregor, Miha, Miran, Jure, Marko }

Kot primer sta v spodnji tabeli prikazana seznama, ki pripadata Nejcu iz testne množice. Tabela je sortirana po pravih ocenah, v padajočem vrstnem redu.

	Prave ocene	Ocene sistema
Miran	7.0	6.3
Miha	7.0	7.3
Barbara	6.5	6.5
Marko	6.0	5.6
Gregor	6.0	3.4
Andrej	6.0	4.1
Miroslav	6.0	4.3
Feliks	5.5	2.8
Natasa	5.5	4.7
Sara	5.5	5.8
Rok	4.0	3.5
Matej	4.0	3.3
Darko	4.0	5.0
Jure	3.5	3.8
Kristjan	3.0	4.1
Dasa	2.5	3.1
Pia	2.0	3.4
Simon	2.0	3.5
Bostjan	2.0	3.0

Tabela 1. Tabelirani seznami Nejca.

Za lažje razumevanje je na grafu 1, prikazana primerjava vrednosti iz tabele 1.



Graf 1. Primerjava ocen sistema in lastnih ocen za osebo Nejc.

7.3. Korelacija

Zanimala nas je moč linearne povezanosti med pravimi in sistemsko določenimi ocenami. Korelacijo smo izmerili s Pearsonovim koeficientom [16], ki temelji na podlagi kovariance in standardnih odklonov serij obeh spremenljivk. Enačba za izračun je naslednja:

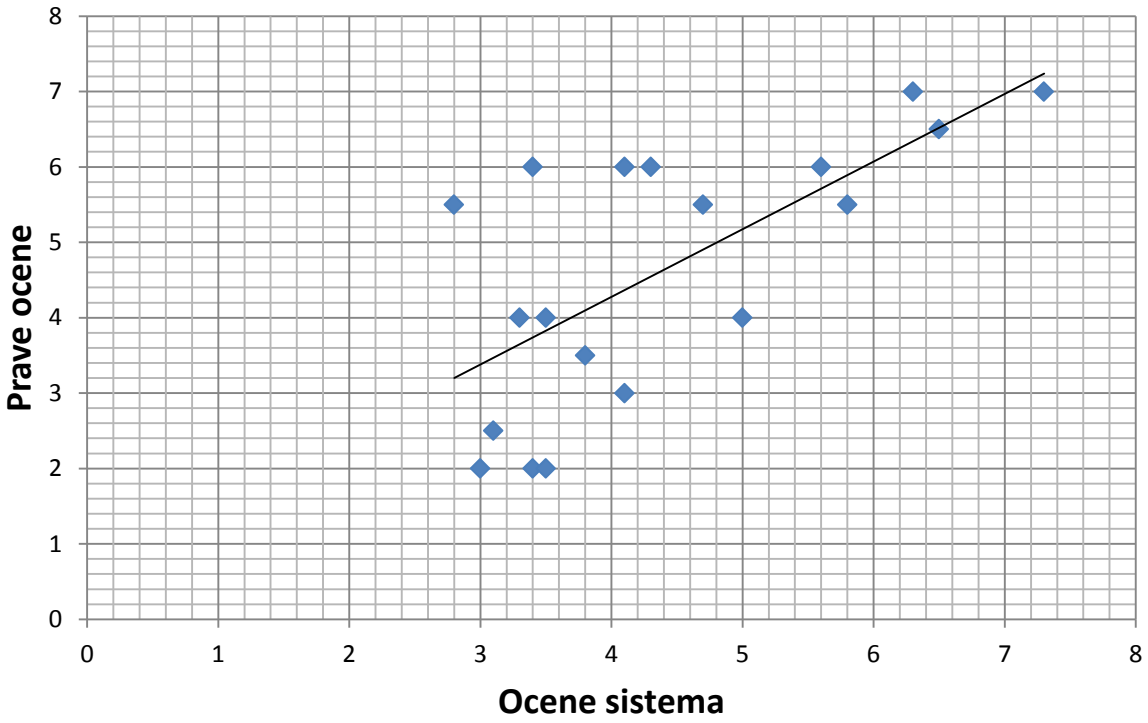
$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

kjer X in Y predstavljata pravo in sistemsko določeno oceno.

Pearsonov koeficient je omejen na intervalu od -1 do 1. V praksi se stopnja korelacije deli na tri razrede:

- visoka korelacija (koeficient med 0.5 in 1.0 oz. -0.5 in -1),
- srednja korelacija (koeficient med 0.3 in 0.5 oz. -0.3 in -0.5),
- nizka korelacija (koeficient med 0.1 in 0.3 oz. -0.1 in -0.3).

Za predstavitev povezanosti lastnih in sistemsko določenih ocen je spodaj graf, ki temelji na primeru Nejca, iz prejšnjega poglavja.



Graf 2. Prikaz korelacije med pravimi in sistemsko določenimi ocenami.

Na grafu je devetnajst točk, ki predstavljajo ostale osebe iz testne množice. Koordinato točke definira ocena sistema (horizontalna os) in lastno določena oz. prava ocena (vertikalna os).

Korelacijski koeficient za ta primer znaša 0.68, kar pomeni visoko korelacijo.

Korelacijski koeficienti za ostale iz testne množice so v tabeli 2.

Nejc	0.68
Miran	0.53
Miha	0.52
Barbara	0.75
Marko	0.68
Gregor	0.62
Andrej	0.71
Miroslav	0.49
Feliks	0.72
Natasa	0.68
Sara	0.69
Rok	0.57
Matej	0.47
Darko	0.47
Jure	0.24
Kristjan	0.67
Dasa	0.25
Pia	0.28
Simon	0.82
Bostjan	0.45

Tabela 2. Tabela prikazuje vse osebe iz testne množice ter njim pripadajoče korelacijske koeficiente.

Povprečni korelacijski koeficient vseh iz testne množice znaša 0.58.

Preučili smo primer Jureta, ki ima v testni množici najnižji korelacijski koeficient. Ugotovili smo, da je taka oseba zelo selektivna pri grajenju seznama interesov. Tako so na Juretovem seznamu samo redke aktivnosti, ki se ne pojavljajo pri ostalih v množici.

7.4. Rezultati evalvacije

Rezultat natančnosti napovedi smo ponazorili s pomočjo povprečne absolutne napake (angl. MAE – mean absolute error) [17]. MAE meri povprečno odstopanje napovedanih vrednosti od dejanskih.

Matematično jo izrazimo kot:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i|$$

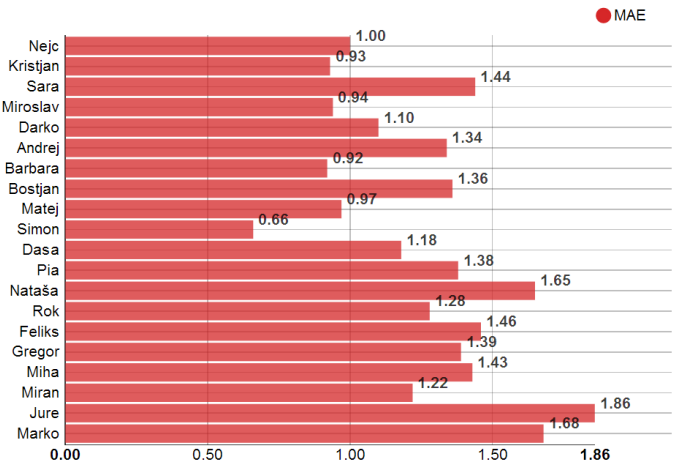
kjer n predstavlja velikost testne množice, f_i so sistemsko napovedane ocene, y_i pa človeško napovedane ocene.

Vzemimo prejšnji primer Nejca, ter njegovih seznamov:

- [6.3, 7.3, 6.5, 5.6, 3.4, 4.1, 4.3, 2.8, 4.7, 5.8, 3.5, 3.3, 5.0, 3.8, 4.1, 3.1, 3.4, 3.5, 3.0]
- [7.0, 7.0, 6.5, 6.0, 6.0, 6.0, 6.0, 5.5, 5.5, 5.5, 4.0, 4.0, 4.0, 3.5, 3.0, 2.5, 2.0, 2.0, 2.0]

Prvi element prvega seznama odštejemo od prvega elementa drugega seznama, drugi element prvega seznama odštejemo drugega elementa drugega seznama in tako do konca obeh seznamov. Absolutne razlike elementov seštejemo in na koncu vsoto delimo z dolžino enega od seznamov. V tem primeru povprečna absolutna napaka znaša 1,0.

Na konkretnem primeru testne množice iz prejšnjega poglavja, smo dobili naslednje rezultate:

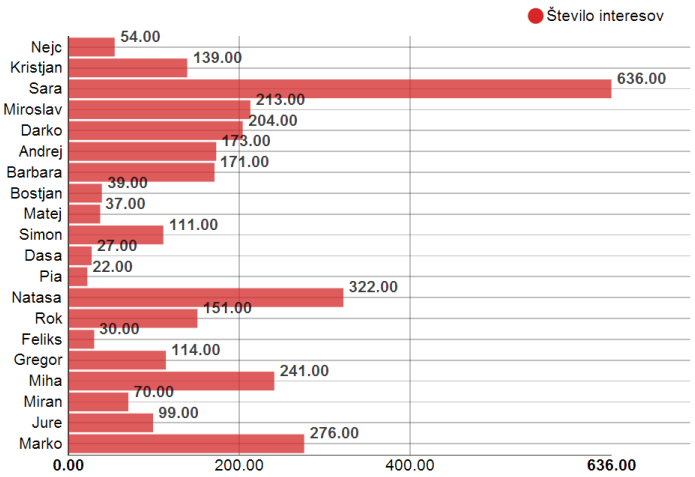


Graf 3. Povprečna absolutna napaka oseb iz testne množice.

Vse povprečne absolutne napake seštejemo in delimo z dvajset, tako dobimo povprečno absolutno napako vseh v množici, ta znaša 1,32.

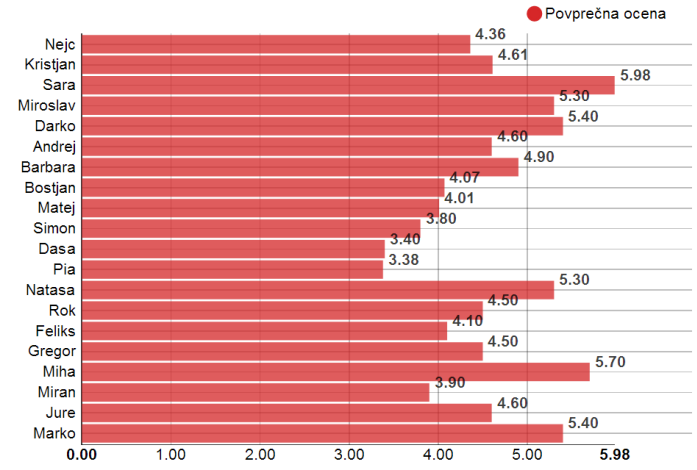
7.5. Opažanja

Opazili smo, da se največja absolutna napaka pojavlja pri uporabnikih, ki imajo na svojem Facebook profilu nadpovprečno velik seznam interesov. Spodnja slika prikazuje število interesov na posameznika v testni množici.



Graf 4. Število interesov na posameznika v testni množici.

Za boljše razumevanje smo pogledali povprečje ocen, ki jih generira sistem, napram drugim iz testne množice.

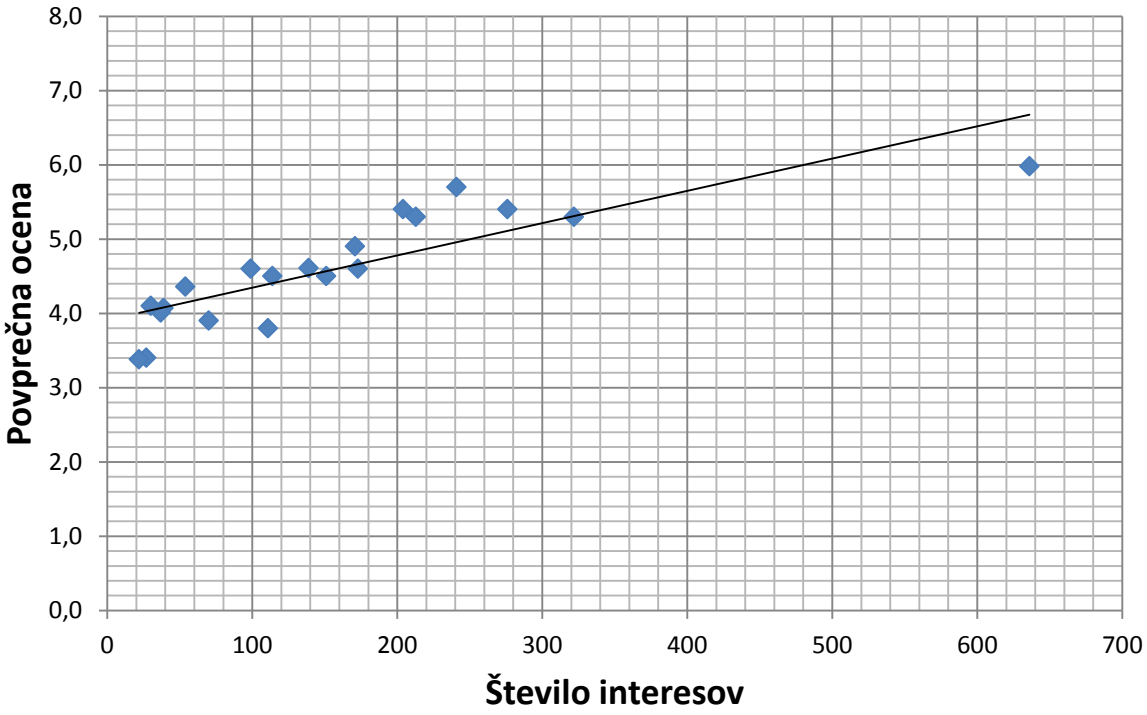


Graf 5. Povprečne ocene.

Iz zgornjih grafov lahko opazimo povezanost med številom interesov in povprečno oceno posameznika.

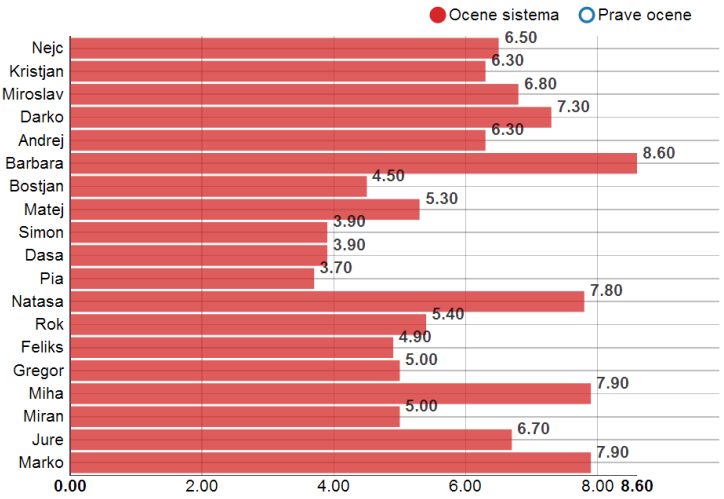
Izračunali smo Pearsonov korelacijski koeficient, ta znaša 0.84, ter potrjuje tesno povezanost spremenljivk.

Pari ocen in št. interesov so predstavljeni na naslednjem grafu.



Graf 6. Tesna povezanost med številom interesov in povprečno oceno posameznika.

Na primeru osebe Sara, ki ima nadpovprečno velik seznam interesov, je vidna pristranskost sistema do oseb, ki imajo prav tako nadpovprečno dolg seznam interesov.



Graf 7. Ocene Sare, ki jih je določil sistem.

Večje število interesov osebe pomeni večje besedilo, ki tej osebi pripada. Večje besedilo pa privede do pristranskosti sistema pri ocenjevanju.

Eden od možnih vzrokov je sledeč. Pri pretvarjanju besedila v vektorsko obliko, definiramo TF (angl. term frequency) kot število pojavitev besede v besedilu, taka definicija pri daljših dokumentih privede do pristranskosti.

Problem rešimo tako, da TF definiramo kot število pojavitev besede v besedilu deljeno z najvišjo pojavitvijo katerekoli besede v besedilu:

$$tf(t,d) = 0{,}5 + \frac{0{,}5 \times f(t,d)}{max\{ f(t,d) \}}$$

kjer, t predstavlja besedo, d pa besedilo.

Tako je TF vedno na intervalu od 0.5 do 1.

8. Sklepne ugotovitve

V diplomski nalogi smo izdelali prototip priporočilnega sistema, ki turistom svetuje pri izbiri turističnega vodiča. Priporočilni sistem uporabnikom priporoča vsebino na podlagi njihovih preferenc na Facebooku.

Za izvedbo prototipa priporočilnega sistema smo preučili priporočilne sisteme ter njihove tipične metode profiliranja, jezikovne tehnologije in metode evalvacije priporočilnih sistemov. Seznanili smo se s shemami uteževanja besedila in primerjalnimi funkcijami.

Ugotovili smo, da je priporočilni sistem, ki temelji na Facebookovih podatkih uspešen. Izjema so uporabniki, ki imajo prevelik seznam interesov na Facebooku. Problem takih, bi bilo treba podrobneje raziskati.

Ostalo je mnogo stvari, ki jih še nismo preizkusili:

- dodatno profiliranje vodičev, glede na ocene, katere bi prispevali turisti,
- upoštevanje cen turističnih vodičev pri svetovanju turistu, če je stopnja podobnosti dveh vodičev enaka, je bolj smiselno najprej priporočiti vodiča z višjo ceno,
- svetovanje turistu, glede na aktualno stanje vremena. Pri razvoju spletne platforme ShowMeAround smo ugotovili, da turisti povečini rezervirajo aktivnost na isti dan obiska spletne platforme.

9. Literatura in viri

[1] (2013) Collaborative filtering. Dostopno na: http://en.wikipedia.org/wiki/Collaborative_filtering/.

[2] (2013) Recommender system, Content-based filtering. Dostopno na http://en.wikipedia.org/wiki/Recommender_system/.

[3] (2013) Spletno mesto ShowMeAround. Dostopno na http://www.showmearound.net/.

[4] (2013) Spletno mesto Pyfaceb. Dostopno na https://bitbucket.org/sproutsocial/pyfaceb/.

[5] (2013) Spletno mesto NLTK. Dostopno na http://nltk.org/.

[6] (2013) Spletno mesto Microsoft Translator. Dostopno na http://www.bing.com/translator/.

[7] (2013) Spletno mesto Langid.py. Dostopno na https://github.com/saffsd/langid.py/.

[8] (2013) Spletno mesto Django. Dostopno na https://www.djangoproject.com/.

[9] (2013) Spletno mesto SQLite. Dostopno na http://www.sqlite.org/.

[10] (2013) By the numbers: 51 amazing facebook stats. Dostopno na http://expandedramblings.com/index.php/by-the-numbers-17-amazing-facebook-stats/.

[11] (2013) TF-IDF. Dostopno na http://en.wikipedia.org/wiki/Tf-idf/.

[12] (2013) Lean startup. Dostopno na http://en.wikipedia.org/wiki/Lean_Startup/.

[13] (2013) Spletno mesto Google Analytics. Dostopno na http://www.google.com/analytics/.

[14] G. Schröder, M. Thiele in W. Lehner, »Setting Goals and Choosing Metrics for Recommender System Evaulations«, v zborniku *Joint proceedings of the RecSys 2011 Workshop on Human Decision Making in Recommender Systems*, ZDA, 2011, str. 78-85.

[15] S. M. McNee, J. Riedl in J. A. Konstan, »Accurate is not always good: How Accuracy Metrics have hurt Recommender Systems«, v zborniku *Extended Abstracts of the 2006 ACM Confrence on Human Factors in Computing Systems*, Kanada, 2006, str. 1097-1101.

[16] (2013) Pearson product-moment correlation coefficient. Dostopno na http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient/.

[17] (2013) Mean absolute error. Dostopno na http://en.wikipedia.org/wiki/Mean_absolute_error/.