

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

Robert Kullamaa 112945IAPB

VEEBIPOE TOODETE SOOVITUSSÜSTEEMI ARENDAMINE

Bakalaureusetöö

Juhendaja: Tarvo Treier

MSc

Tallinn 2018

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Robert Kullamaa

Annotatsioon

Antud bakalaureusetöö eesmärgiks on olemasolevale veebipoele toodete soovitusüsteemi loomine.

Töö käigus analüüsitakse eksisteerivat veebipoodi, antakse ülevaade kasutatud tehnoloogiast, mille abil rakendus loodi ning arhitektuurist, kuidas süsteem üles ehitati. Tutvustatakse seadmepoe müügi ja konfigureerimise eest vastutavate tootehaldurite nõudeid süsteemile ning andmeallikaid soovitude pakkumiseks- kasutaja ostude ajalugu, otsingute kriteeriumid, erinevate toodete detailvaadete külastamine.

Leitakse veebipoe arhitektuurile ja tootehalduri soovidele vastav soovitusüsteemi tüüp, valitakse algoritm, mille abil soovitusüsteem üles ehitada.

Realiseeritakse soovitusüsteem, mis integreeritakse olemasoleva veebipoega. Soovitusüsteem analüüsib kasutajate käitumisi ning vastavalt sellele pakub erinevaid tooteid. Pikemas perspektiivis on võimalik kasvatada veebipoe kaudu müüdavate toodete kogust. Koostatakse ka plaan, mille ülesanne on selgitada, kuidas edukust mõõta ning järgmisi sooritusi parandada, täiustada.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 7 peatükki ja 11 joonist.

Abstract

Development of web shop product recommendation system

The aim of this thesis is to introduce the field of recommendation systems and to develop a web shop recommendation system for a web shop.

The thesis will describe the current web shop, introduce the technology used in the web shop solution and give an overview of the architecture of the existing system. The rules and requirements of the product managers will be introduced. Product managers are specialists that are responsible for the configuration of products and manage sales. The thesis will also analyse existing data sources like client shopping history, product search history and keep track of which product pages the client has visited.

A suitable recommendation system will be chosen considering the rules given by product managers and the architecture of the web shop. A suitable algorithm will be found to implement the recommendation system.

The thesis will describe steps taken to create a recommendation system that will be able to create recommendations for users and by doing so increase sales. The steps needed to integrate the recommendation system into the web shop will be described.

A plan will be composed to measure the efficiency of the designed recommendation system and how to improve the system's performance.

The thesis is in Estonian and contains 29 pages of text, 7 chapters and 11 figures.

Lühendite ja mõistete sõnastik

AngularJS	JavaScripti raamistik dünaamiliste veebilehtede loomiseks
API	<i>Application Programming Interface</i> , rakendusliides, arvuti operatsioonisüsteemiga või rakendusprogrammiga määratud reeglistik, mille alusel rakendusprogramm kasutab operatsioonisüsteemi või teise rakendusprogrammi teenuseid
CSS	<i>Cascading Style Sheets</i> , veebilehtede kujunduseks kasutatav märgistuskeel
de facto	<i>de facto</i> (ld. tegelikult, faktiliselt). Laialdaselt aksepteeritud tõde, mis aga pole ametlikult seadusega kinnistatud või fikseeritud.
HTML	<i>HyperText Markup Language</i> , hüpertekst-märgistuskeel, enimlevinud kodeerimissüsteem (tekstivorming) veebidokumentide loomiseks
HTTP	<i>HyperText Transfer Protocol</i> , hüperteksti edastusprotokoll., TCP/IP klient-server protokoll HTML-dokumentide vahetamiseks veebis
Java	Objektorienteeritud programmikeel, mille abil saab luua võrguservereid ja klientrakendusi
JavaScript	Skriptikeel, mis võimaldab luua interaktiivseid veebilehti
knn	<i>k-nearest neighbours</i> , <i>k-lähima naabri algoritm</i>
REST	<i>Representational State Transfer</i> , arhitektuuriistiil, mida kasutatakse hajutatud süsteemides
Minu-elisa	Elisa veebipoe süsteemi sessioonihalduse rakenduskiht
Epd-api	Elisa veebipoe süsteemi äriloogika- ja andmebaasi suhtluskiht

Sisukord

1 Sissejuhatus	8
1.1 Ülevaade olemasolevast süsteemist	9
2 Soovitussüsteemi valik	11
2.1 Tootehaldurite reeglid	11
2.2 Soovitussüsteemide murekohad	12
2.3 Lisatarvikute soovitussüsteem	12
2.4 Põhitoote soovitussüsteem	13
3 Ülevaade soovitussüsteemide valdkonnast	14
3.1 Sisu järgi (Content-based)	14
3.2 Kollektiivne filtreerimine (Collaborative filtering)	15
3.3 Teadmistel baseeruv (Knowledge-based).....	16
3.4 Sotsiaalmeedia andmetel baseeruv (Community-based).....	17
3.5 Hübrid soovitussüsteem (Hybrid recommender system)	17
3.6 Kokkuvõte	18
4 Andmeallikad	19
4.1 Olemasolevad andmeallikad.....	19
4.2 Uute andmeallikate loomine	19
4.2.1 Kasutaja käest tootehinnangu küsimine.....	19
4.2.2 Otsingute ajalugu andmeallikana	20
4.3 Hinnangute andmise funktsionaalsuse realiseerimine.....	20
4.3.1 Kasutaja identifikaatori loomine.....	20
4.3.2 Hinnangu andmise funktsionaalsus	21
4.4 Kokkuvõte	22
5 Soovitussüsteemi realiseerimine	23
5.1 K-lähima naabri (knn) algoritm	23
5.2 Lähima naabri algoritmi tugevused.....	26
5.3 Lähima naabri algoritmi nõrkused	26
5.4 Nõrkuste leevendamine.....	27
5.4.1 Külma käivituse probleem	27
5.4.2 Andmete hõredus	28
5.4.3 Uute toodete leidmine	30
5.4.4 Algoritmi kiiremaks muutmine	30
5.5 Algoritmi realiseerimine rakenduses.....	31
5.5.1 Hinnangute kogumise funktsionaalsus	32
5.5.2 Soovituste juhtimine tootehaldurite poolt.....	32
5.5.3 Soovituste pakkumine	33
5.6 Soovituse kasutamine.....	34
6 Järeldused ja järgmised sammud	36
7 Kokkuvõte	37
Kasutatud kirjandus	38

Jooniste loetelu

Joonis 1. Veebipoe komponendid ning nendevaheline suhtlus.....	10
Joonis 2. Hinnangukomponent toote detailvaates.....	21
Joonis 3. Andmebaasi tabel tootehinnangute hoidmiseks.....	22
Joonis 4. Lähima naabri näide.....	24
Joonis 5. Kasutaja hinnangute tabel, $k=2$	26
Joonis 6. Kasutaja hinnangute tabel, puuduvad väärtused on nullid.....	28
Joonis 7. Kasutaja hinnangute tabel normaliseeritud koos algsete hinnangutega.....	29
Joonis 8. Veebipoe rakenduse kihiline arhitektuur.....	31
Joonis 9. Tootehaldus tööriistas koefitsiendi määramine.....	33
Joonis 10. Lisatarvikute kuvamise komponent.....	34
Joonis 11. Soovituste kasutamise statistika jaoks loodud andmebaasi tabel.....	35

1 Sissejuhatus

E-kaubandus (läbi interneti kaupade ja teenuste soetamine) muutub iga aastaga tähtsamaks müügikanaliks. Statistikaameti 2016. aasta andmete kohaselt ostis aasta jooksul internetist mõne toote või teenuse 65% (543 100) 16–74-aastastest internetikasutajatest[1]. Tugeva konkurentsi tõttu E-kaubandusega tegelevate ettevõtete seas, peab aina rohkem investeerima süsteemidesse, mis võimaldaks rohkem müüa. Uurides E-kaubanduse suurimaid ja edukamaid ettevõtteid nagu Amazon, on võimalik leida erinevaid tehnoloogilisi strateegiaid, mille abil müüki kasvatada.

Üheks Amazoni edu võtmeks loetakse nende soovitusüsteemi, mis võimaldab kliendil avastada ning kiiremini üles leida uusi ja huvi pakkuvaid tooteid. Soovitusüsteemide tähtsusest on näiteid veel teistestki valdkondadest - videoedastusteenust pakkuv ettevõtte Netflix kuulutas 2006. aastal välja konkursi, mille eesmärgiks oli leida uusi soovitusüsteemi algoritme ning parima süsteemi disaini esitanud meeskonnale lubati miljon dollarit[2].

Antud töö autor on veebipoe arendaja, kelle huviks on uurida personaalsete soovitusüsteemi erinevaid komponente ja algoritme. Bakalaureusetöö tulemusena realiseeritakse soovitusüsteem, mis annab võimalikult täpseid soovitusi veebipoe kasutajale, et panna ta huvi tundma ka teiste toodete vastu ning seeläbi suurendada veebipoes müüdavate toodete kogust. Valmib soovitusüsteem, mis sobib olemasoleva veebipoe arhitektuuri, toodete ning ettevõtte äristrateegiaga. Töö käigus uuritakse ka erinevat tüüpi laialt levinud soovitusüsteeme.

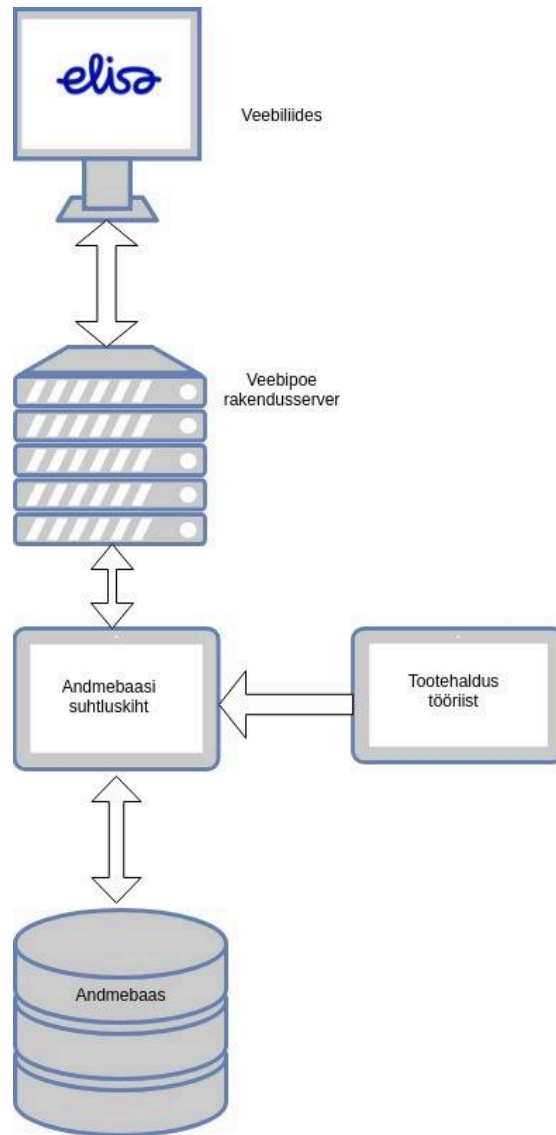
Teises peatükis käsitletakse ettevõtte toodete müügiga tegelevate spetsialistide nõudmisi ning reegleid, millega peab arvestama soovitusüsteemi arendades. Olemasolevate soovitusüsteemide analüüs toimub kolmandas peatükis, et leida sobivaim süsteem teises peatükis käsitletud nõuetele ja reeglitele ettevõtte poolt. Soovitusüsteemid töötavad erinevat tüüpi andmete põhjal- neljandas peatükis analüüsitakse veebipoe andmeid, mida saab kasutada soovitusüsteemi loomiseks ning leitakse ka võimalusi

uute andmeallikate loomiseks. Soovitussüsteemi disain, realisatsioon, algoritmide analüüs ning valik on kirjeldatud viiendas peatükis. Kuuendas peatükis määratakse mõõdikud, mille põhjal saaks loodud süsteemi edukust mõõta, kalibreerida süsteemi ning võrrelda, kas sooritus paranes või mitte.

1.1 Ülevaade olemasolevast süsteemist

Olemasolev süsteemiks on Elisa Eesti veebipood, mille komponente on võimalik näha 1. Jooniselt, koosneb järgnevatest osadest:

- Veebipood – Veebileht, kus on toodete nimekiri, erinevad parameetrid, hind. Veebipood võimaldab tooteid omavahel võrrelda, otsida parameetrite ja hinna järgi ning osta. Veebipoe serveripoolne rakendus on loodud Java programmeerimiskeeles ning veebibrauseris esitletud kasutajaliides on kirjutatud JavaScript-i raamistiku AngularJS-i kasutades.
- Tootehaldustööriist –Veebiliidesega tootehalduri tööriist, mille abil hallatakse veebipoes müüdavaid tooteid. Tööriist lubab sisestada ning muuta toodete parameetreid ja luua seoseid toodete vahel.
- Suhtluskiht – kolmandaks komponendiks on rakendus, mis pakub liidestust veebipoe ning tootehaldusetööriista vahel ning mille läbi pääseb ligi andmebaasis olevatele andmetele.



Joonis 1. Veebipoe komponendid ning nendevaheline suhtlus

Veebipoes müüdavad tooted on seadmed. Seadmed jagunevad kategooriatesse, näiteks: nutitelefonideks, tahvelarvutiteks, televiisoriteks ja sülearvutiteks. Lisaks on põhimõtteline jagunemine põhiseadmeteks ning lisatarvikuteks. Näiteks Samsung Galaxy S8 telefon on põhiseade ning temale sobiv ekraanikaitsekile või kaitseümbris on Elisa veebipoe mõistes lisatarvikud.

2 Soovitussüsteemi valik

2.1 Tootehaldurite reeglid

Töö kirjutamise käigus uuris bakalaureusetöö autor kolme erineva tootehalduseksperdi käest nende nõudeid soovitussüsteemile. Autor palus halduritel välja tuua erandid, mida nad tooteid configureerides arvestavad. Uurimise põhjal said kirja reeglid, millega soovitussüsteem peab arvestada:

- Soovitused peavad jääma sama kategooria raamesse. Kui soovitus kuvatakse nutitelefonis vaates, siis peaks halduri arvates soovitus kuuluma nutitelefonis kategooriasse. Tehnikapoodi ostma tulles on tal tavaliselt kindel eelistus kategooria suhtes- nutitelefonis ostma tulles ei soovi kasutaja tavaliselt kõlareid osta.
- Vähendamaks kaupade tagastamist klientide poolt, peab olema halduril võimalik defineerida kokku sobivad seadmed. Koos ostetavate toodete soovitusel aluseks peab olema tootehalduri poolt loodud seos põhiseadme ja temaga kokku sobiva lisatarviku vahel, näiteks nutitelefoni ning ekraani kaitsekile.
- Teatud toodete kategooriate puhul peab lähtuma haldurite kogemuste põhjal koostatud reeglitest, näiteks nutitelefonide puhul Apple iPhone toodetest huvi tundvad kasutajad tavaliselt ei huvitu Androidi operatsioonisüsteemiga toodetest.
- Soovitades kokku sobivaid lisatarvikuid, ei tohi soovitusite arv ületada kaheksat soovitusi.
- Vajadusel saab haldur soovitussüsteemi valikuid mõjutada nii, et määrab toote prioriteetsemaks, kui teised ning toode tuleb alati soovitatud tootena kliendile nähtavale.

2.2 Soovitussüsteemide murekohad

Soovitussüsteemide üheks levinud probleemiks on “külmakäivituse probleem” (Cold-start problem). Algselt, kui andmeid ei ole veel kogunenud, on süsteemil väga raske soovitusi luua, sest täpsete soovituste loomiseks on vaja andmeid kasutaja ja toote kohta. Selle probleemi populaarseimaks lahenduseks “juhuslikult valitud” toodete strateegia, “populaarseimate toodete” strateegia ning segu mõlemast. [5].

Kindlasti peab mõtlema süsteemi võimele võidelda pahatahtliku mõjutamise vastu. Kui süsteem on kergesti vastuvõtlik pahatahtlikule tegevusele, annab see võimaluse konkurentidel veebipoe müüki negatiivselt mõjutada - näiteks toote maaletoojal on soov oma toote populaarsust teiste ees kergitada. Mõlemal juhul ei annaks soovitussüsteem enam häid soovitusi ning kaotaks klientide silmis oma väärtuse.

2.3 Lisatarvikute soovitussüsteem

Tootehalduri soovitusi arvesse võttes ei sobi üks konkreetne soovitussüsteem nii põhitoodete kui ka lisatarvikute jaoks. Olemasolev veebipood kuvab põhitoote detailvaates osa kokku sobivatest lisatarvikutest. Kõiki lisatarvikuid ei ole mõistlik põhitoote juures kuvada, sest põhitootel võib olla mitukümmend kokku sobivat lisatarvikut. Seega on vaja soovitussüsteemi, millele kehtiksid järgnevad reeglid:

- Kõik soovitatud tooted peavad sobima kokku põhitootega.
- Maksimaalselt kaheksa soovitust.
- Haldur saab määrata millised tooted on alati soovitatud, olenemata andmetest, mille põhjal süsteem otsuseid teeb.

Kasutaja eelistused on teada ainult põhitoote kohta, sest tegu on mingi põhitoote detailvaatega, kus lisatarviku soovitusi kuvatakse. Uurides levinud veebipood nagu amazon.com või kaup24.ee, siis on levinud stiil, kus pakutakse osta tooteid, mida tihti koos ostetakse. Seega peaks looma teadmistel baseeruva süsteemi, mis soovitaks ainult kokku sobivaid tooteid, mida kõige tihedamini koos põhitootega ostetakse. Andmeid tihti koos ostetud toodete kohta saab veebipoe tellimuste ajalugu uurides.

Tugevuseks antud lahendusel on laialdane kasutus e-kaubanduse valdkonnas. Tuttav funktsionaalsus tekitab klientides kindlust ning nad oskavad soovitusi kasutada. Ei ole vaja lisaarendusi, et hakata koguma lahenduse jaoks vajalike andmeid. Juba on olemas pikaajalisel perioodil kogutud andmed, mille põhjal algusest peale soovitusi luua. Nõrkuseks on süsteemi suutmatust soovitada uusi tooteid.

2.4 Põhitootete soovitusüsteem

Toodete soovitusüsteemil on samuti määratud teatud piirangud tootehaldurite poolt.

- Soovitatud toode peab olema samast kategooriast nagu toode, mille juures soovitus näidatakse.
- Teatud parameetrid on klientidele väga tähtsad ning need peaks soovitusi mõjutama. Näitena Apple iPhone nutitelefoni puhul peaks ka soovitus olema Apple iPhone erinevad variatsioonid.

Haldurid peavad saama määrata parameetreid, mis peaksid pakkuma huvi toote kohta-see osa süsteemist on sisupõhine soovitus. Kuna soovitused peavad jääma ka samasse kategooriasse, siis ka seda võib võtta sisupõhise soovitusena või siis teadmispõhisena. Eelnimetatud reeglid piiravad kõikide toodete hulka väiksemaks alamhulgaks, kuid ka neid võib olla sadu. Seega on vaja veel alamhulgast parimate soovitusete jaoks oma süsteemiosa, mis võiks olla kollektiivsel filtreerimisel baseeruv. See annaks võimaluse kliendil avastada teistsuguseid tooteid, kui ainult sisupõhisel lähenemisel.

Olemasolevas süsteemis puudub integratsioon sotsiaalmeediaga ning iga lisapingutus kliendi poolt vähendab ostu sooritamiseni jõudnud klientide arvu. Tänu sellele võib bakalaureusetöös välistada sotsiaalmeedial baseeruva süsteemi.

Lõputöö raames realiseeritava soovitusüsteemi puhul on tegu hübriidsüsteemiga, kus algselt piirab toodete hulga sisupõhine süsteem ning seejärel allesjäänute seast valitakse teiste kasutajate ning kliendi enda käitumismustri järgi parimad soovitused.

3 Ülevaade soovitusüsteemide valdkonnast

Antud peatükis tehakse ülevaade soovitusüsteemide valdkonnast. Tuuakse välja enim kasutatud süsteemitüübid[3] ning nende realiseerimiseks kasutatud algoritmid. Süsteemitüübid erinevad üksteisest otsuste tegemiseks valitud objekti poolest. Analüüsitakse erinevate soovitusüsteemi tüüpide tugevaid ning nõrku külgi.

Soovitusüsteemide põhieesmärgid on järgnevad [4]:

- Aitab otsustada – annab arvulise hinnangu objekti sobivuse kohta.
- Aitab võrrelda – annab võimaluse hinnangute abil objekte võrrelda.
- Aitab avastada – võimaldab leida uusi objekte, mis võivad kasutajat huvitada.
- Aitab leida – esitleb huvi pakkuvad tooted kasutajale ilma otsimise vaevata.

3.1 Sisu järgi (Content-based)

Sisu järgi soovitusi tegevad süsteemid võtavad sisendiks kasutaja eelistused ning otsivad olemasolevate objektide seast sarnaseid objekte[3]. Näiteks saab võrrelda arvuteid tehniliste näitajate järgi: kõvaketta maht, operatiivmälu, protsessori kiirus, võib ka võrrelda mittetehniliste näitajate järgi: tootja, hind, garantii pikkus.

Sisu järgi soovitava süsteemi tugevused on:

- Suudab edukalt uusi objekte soovitada, ei vaja aega, et teised uusi objekte hindaks või populaarsust koguks.
- Väga läbinähtav, otsustusprotsess on kergesti mõistetav ja valideeritav.
- Tugev pahatahtliku mõjutamise vastu, sest soovitused on individuaalsed.

Nõrkusteks on:

- Klient ei avasta uut tüüpi tooteid läbi soovitude, sest soovitatud objektid on sarnased juba kliendi poolt valitud objektidele. (Vastuolus eesmärgiga “Aitab avastada”).
- Ei oska pakkuda täpseid soovitusi, kui on vähe andmeid kasutaja eelistuste kohta.

3.2 Kollektiivne filtreerimine (Collaborative filtering)

Kollektiivse filtreerimise süsteem on soovitusüsteem, mille eelduseks on idee, et sarnase maitsega kasutajatele pakuvad huvi samad objektid[3]. Tihti tuuakse näitena Netflix-i soovitusüsteemi: kui kasutajale A meeldivad sarjad X, Y ja Z ning Kasutajale B meeldivad sarjad X ja Z, siis kasutajale B võiks ka meeldida sari Y.

Süsteemi põhiline käitumismuster üritab kasutajaid kokku viia maitse poolest sarnaste kasutajatega ning pakkuda objekte, mis ühele on meeldinud.

Kollektiivse filtreerimise soovitusüsteemi tugevusteks on:

- Süsteem suudab pakkuda kasutajale uut tüüpi tooteid.
- Laialdasem kasutusala.

Nõrkusteks on:

- Vastuvõtlik rünnakute vastu. Kasutaja tegevus mõjutab teistele pakutavaid soovitusi.
- Väga halvasti saab hakkama uute objektidega- kui keegi pole veel uuele tootele hinnangut andnud, siis seda objekti ei eksisteeri süsteemi jaoks.
- Vajab palju andmeid kasutaja kohta, et oskaks leida sarnaseid kasutajaid.

3.3 Teadmistel baseeruv (Knowledge-based)

Teadmistel baseeruv süsteem talletab süsteemi haldaja poolt sisestatud reegleid, mis kirjeldavad, kuidas süsteem peaks seoseid looma ning soovitusi looma[3]. Näiteks soovitusi võib pakkuda ainult ühest kategooriast toodetele – mobiiltelefonile pakutakse ainult mobiiltelefone. Teine näide on see, et nutitefonide puhul on teada reegel, et inimesed, kes on huvitatud Apple-i toodetest ei huvitu tavaliselt Androidi operatsioonisüsteemiga nutitefonidest.

Teadmistel baseeruva soovitusüsteemi tugevusteks on:

- Süsteemi soovitused on alfaasis tavaliselt täpsemad, kui teistel süsteemidel. Reeglite täpsuse tõttu on paljud soovitused täpsed.
- Süsteem suudab hakkama saada uute objektidega.
- Süsteem on väga tugev rünnakutele – rünnakud ei saa süsteemi mõjutada, sest süsteem ei jälgi kasutaja ajalugu.
- Võimaldab müüjal reklaamida suure laoseisuga või kehtivusaja kaotavat kaupa.

Nõrkusteks on:

- Süsteem ei ole õpivõimeline.
- Nõuab haldurilt pidevat süsteemi konfigureerimist.

3.4 Sotsiaalmeedia andmetel baseeruv (Community-based)

Süsteem loodab selle peale, et kasutajad usaldavad tuttava soovitusi rohkem, kui anonüümseid soovitusi ja hinnanguid[3].

Sotsiaalmeedia andmetel baseeruva süsteemi tugevusteks on:

- Inimestes tekib usaldus soovituste vastu ning seega on soovitused efektiivsemad.
- Lihtne ja läbinähtav algoritm.
- Süsteem on tugev rünnakute vastu.

Nõrkusteks on:

- Süsteem saab ainult siis toimida, kui kasutajad on aktiivsed sotsiaalmeedia kasutajad.
- Tuttavad peavad hinnanguid andma, muidu ei saa soovitusi luua.

3.5 Hübrid soovitussüsteem (Hybrid recommender system)

Hübrid soovitussüsteem on süsteem, mis võtab parimad palad teistest soovitussüsteemidest, et saavutada parim tulemus. Enamik edukaid e-kaubandusega tegelevaid soovitussüsteeme on hübriidsüsteemid[3].

Süsteemi tugevused ja nõrkused ühtivad süsteemis kasutusele võetud komponentidega, siis neid eraldi välja loetleda ei ole mõtet. Eesmärk on võtta kõige paremini sobiv süsteem ning leevendada valitud süsteemi nõrkusi mõne teise süsteemi funktsionaalsusega.

3.6 Kokkuvõte

Kõige populaarsem soovitusüsteemi liik on hübriid soovitusüsteem ning hübriidis kasutatakse põhiliselt kollektiivset filtreerimist ja sisupõhist soovitusüsteemi. Teadmistele põhinevat süsteemi kasutatakse hübriidides erinevate valdkondade ekspertide püstitatud reeglite realiseerimiseks soovitusüsteemis.

4 Andmeallikad

Täpsete soovitude loomise eelduseks on andmed, mille põhjal otsuseid luua. Järgnevas peatükis kirjeldatakse olemasolevad andmeallikad ning analüüsitakse võimalike uusi andmeallikaid. Hinnatakse ka andmete kasulikkust ning arendusmahtu, mis kulub uute andmete kogumisele.

4.1 Olemasolevad andmeallikad

Olemasolevad andmed võib jagada kaheks - sisestatud andmed ning ajalugu. Sisestatud andmed on haldurite poolt määratud täiendavad tooteparameetrid. Ajaloo alla võib lugeda klientide tellimuste ajalugu. Lisaks on võimalik testimiseks hankida andmeid rakenduse logisid töödeldes. Näiteks rakendusserver talletab viimase 30 päeva kõik päringud serverisse logifaili. Logifailist saab filtreerida välja kõik toote detailvaate külastused, mille põhjal luua nimekirja kõige populaarsematest toodetest.

Toodete populaarsuse kohta leidub andmeid ka Google Analytics-is, mis salvestab lehekülgede vaatamisi.

4.2 Uute andmeallikate loomine

4.2.1 Kasutaja käest tootehindangu küsimine

Kasutaja käest hinnangu küsimine on suure tõenäosusega kõige täpsem viis, kuidas õige hinnang tootele määrata. Leidub mitmeid erinevaid mooduseid, kuidas kasutajalt arvamust küsida. Tihtipeale on toote lehel mingi kasutajaliidese komponent, mis võimaldab kasutajal hinnangu määrata, teksti kujul tagasisidet jätta ja oma hinnangut põhjendada.

Teine populaarne viis kasutaja käest hinnangu saamiseks on hüpikaknad. Kui kasutaja on viibinud mingi toote lehel üle poole minuti, siis avaneb talle hüpikaken, mis palub anda hinnangu sellele tootele.

Leidub näiteid sellistest süsteemidest, kus kasutajal endal palutakse täita oma profiil[6]. Antud viisi tugevuseks on konkreetsed eelistused otse kasutaja käest, mis omakorda võimaldab pakkuda näidistena soovitusi ning kasutaja saab soovitusi heaks kiita või

maha laita. Lähenemise nõrkuseks on privaatsuse küsimus. Paljud kasutajad ei ole valmis oma personaalseid andmeid avaldama suvalises veebipoes ja veebipoel on ka kohustus andmeid kaitsta. Euroopa Parlamendi andmekaitse direktiivis[7] sätestatud nõuete kohaselt võib kasutaja nõuda oma andmete väljavõtet ning kustutamist.

Kuna kasutaja profiili täitmine on välistatud, siis vaadatakse tehnilise lahenduse poolest kahte viimast- kasutaja käest hinnangu küsimine toote juures või hüpikaknas. Mõlemad lahendused on tehniliselt väga sarnased, erinevad vaid JavaScript-i arendusmahu poolest. Luuakse JavaScript-i ja HTML-i kasutades hinnangu andmise komponent ning uus päring, mille abil hinnang serverisse jõuaks. Serveris talletatakse hinnang andmebaasi, kus hinne soetakse hinnatud tootega ning hinnangu andnud kasutajaga.

4.2.2 Otsingute ajalugu andmeallikana

Võrreldes toote veebilehe külastusega annab otsing rohkem andmeid toote omadustest, mis on kasutajale olulised. Kui otsingus on määratud hind, siis annab see hinnavahemiku, mida kasutaja on valmis maksma toote eest. Seega on võimalik soovitada kasutajale sellesse vahemikku jäävaid tooteid.

Realiseeritud lahendus suurendab kasutaja hinnangut toodetele, mis vastasid otsingukriteeriumitele ning veel enam nende toodete hinnangut, mille kasutaja otsingutulemustest välja valis ehk avas toote detailvaate.

4.3 Hinnangute andmise funktsionaalsuse realisatsioon

4.3.1 Kasutaja identifikaatori loomine

Veebipood võimaldab toodetega tutvuda, otsinguid sooritada ning hinnanguid anda ilma, et kasutaja oleks sunnitud end tuvastama. Enne kui kasutaja end tuvastab, on talle loodud anonüümne sessioon. Sessiooni lühikese eluea ning hävimise tõttu kasutaja väljalogimisel, ei sobi sessiooni id pikaajaliseks kasutaja identifikaatoriks. Lahenduseks loodi JavaScript-i abil kasutajale juhuslik räsi, mida edaspidi käsitletakse kasutaja identifikaatorina (`user_id`).

JavaScript ei toeta vaikumisi räsi genereerimise funktsionaalsust. Kirjutati funktsioon, mis genereeriks kasutajale unikaalse 32 sümbolise räsi, kasutades JavaScript-i `Math.random()` funktsiooni.

4.3.2 Hinnangu andmise funktsionaalsus

Hinnangu andmiseks on vaja luua HTML-i ja JavaScript-i abil visuaalne komponent, mis võimaldaks kasutajal hinnat anda. Pärast kasutaja hinnangu andmist peab hinnangu salvestama andmebaasi.

Internetis on saadaval mitmeid veebipoe AngularJS raamistikuga loodud lahendusi, mis suudavad kasutajale kuvada hinnangukomponenti. Valik osutus “Angular1 Star Rating”¹ nimelise projekti kasuks, mis pakub graafilist komponenti joonisel 2 ning võimaldab kasutajal anda hinnang viie palli skaalas. Graafiline osa on kergelt muudetav erinevate parameetrite abil, saab muuta ikooni ja teksti asukohta. Antud lahendus sobib kokku veebipoe tehnoloogiatega, mistõttu on kerge kasutusele võtta.



Joonis 2. Hinnangukomponent toote detailvaates

¹ Angular1 Star Rating <https://github.com/BioPhoton/angular1-star-rating>

Kui kasutaja on hinnangu tootele andnud, saadetakse toote ID, hinnang ning kasutaja räsi serverrakendusele andmebaasi salvestamiseks. Selleks on vaja luua andmebaasi tabel, kuhu andmed salvestada. Andmebaasitabel on näha joonisel 3.

webshop_article_rating		
id	INT	PK
user_id	VARCHAR(32)	
date	TIMESTAMP	
rating	FLOAT(7,4)	
handsetmodel_id	INT	

Joonis 3. Andmebaasi tabel tootehinnangute hoidmiseks.

4.4 Kokkuvõte

Olemasolevad andmeallikad aitavad leida populaarseimad tooted erinevatest kategooriatest, mille abil saab tundmatule kasutajale midagi soovitada.

Juurde loodavatest andmeallikatest on mõistlik otsingute ajalugu logidesse salvestada. Süsteemile tehtav arendus on suhteliselt väike ning saadavad andmed peaks andma hea pildi kasutaja eelistustest.

Teiseks lisatavaks andmeallikaks on kasutaja hinnangu küsimine tootele. Lahenduseks valiti toote juures kuvatav hinnangu andmise komponent, mis võimaldab anda viie palli süsteemis hinnangu. Kasutajalt profiili täitmist ei ole esialgu mõistlik arendada, sest vajalik arendusmaht ning andmekaitse seaduse väljakutsed ületavad arendusega saadavate andmete väärtuse.

5 Soovitussüsteemi realiseerimine

Soovitussüsteemides saab kasutada mitmeid masinõppes levinud algoritme ja tehnikaid. Andmete kogumiseks on võimalik kasutada andmekaevandamiseks loodud funktsionaalsust ning soovituste jaoks on võimalik integreerida erinevaid masinõppe algoritme, et luua soovitamiseks mudeleid ning jagada erinevaid tooteid gruppidesse mingi parameetri alusel.

Eelnevalt jõuti järeldusele, et antud töös on mõistlik arendada kollektiivsel filtreerimisel baseeruvat süsteem. Järgnevalt uuritakse algoritme, mille abil saaks süsteemi valmis implementeerida ning leitakse algoritmi tugevad ning nõrgad küljed. Analüüsitakse, kuidas saaks soovitussüsteemi nõrkusi kas eemaldada või leevendada nende mõju rakenduse efektiivsusele.

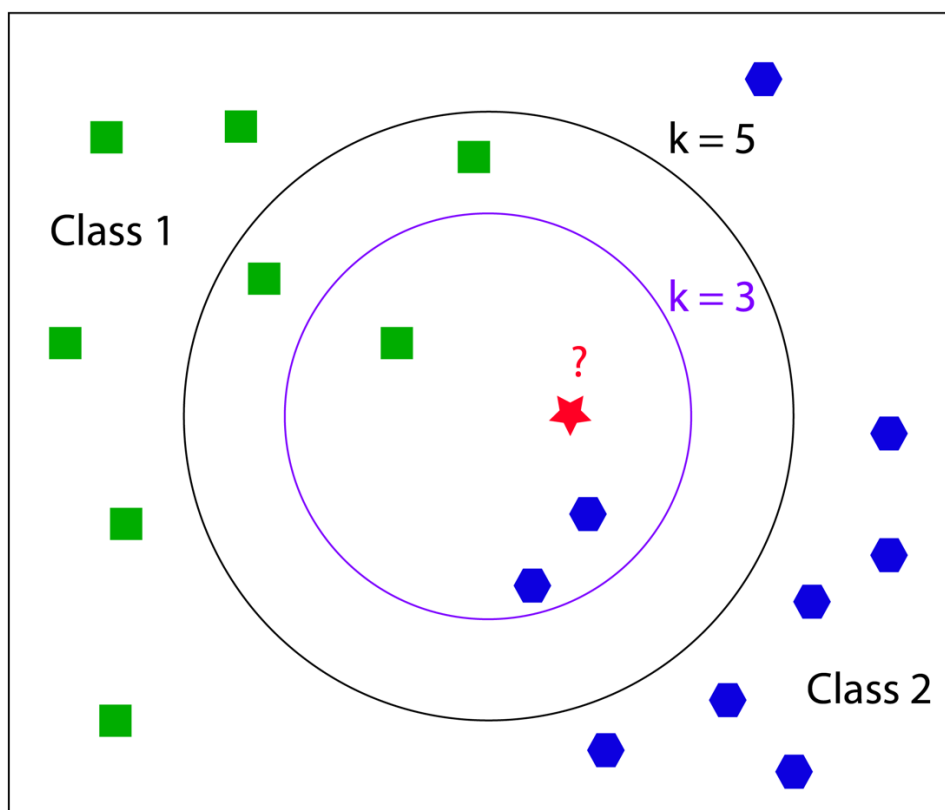
5.1 K-lähima naabri (knn) algoritm

Lähima naabri algoritmi loetakse kollektiivse filtreerimise meetodil töötava soovitussüsteemi *de facto* algoritmiks[3]. Lähima naabri lähenemine on saavutanud suure populaarsuse, sest tulemused on head, algoritm on väga kergesti mõistetav ning lihtsalt realiseeritav. Antud bakalaureusetöö on esimene katse luua veebipoele soovitussüsteem, tänu sellele sobib kergesti mõistetav algoritm, mida on lihtne muuta veebipoe ärioloogikat arvestades.

K-lähima naabri algoritm sobib väga hästi kollektiivse filtreerimise süsteemi arendamiseks, sest nii süsteemi kui ka algoritmi põhiidee on sarnased. Mõlema puhul on eesmärk sarnaseid objekte leida - antud töös on objektideks veebipoe kasutajad.

Algoritm võtab sisendiks mingi kasutaja või objekti ning otsib K lähimat naabrit, kus K on positiivne paaritu täisarv. Lähimad naabrid leitakse sarnasusefunktsiooni abil, milleks on tavaliselt kauguse arvutamine kahe kasutaja või objekti andmete järgi. Enim leidub näiteid, kus lähima naabri algoritmi kasutatakse objektide klassifitseerimise eesmärgil.

Vaadatakse näitena joonist 4. Joonisel on kujutatud 2-mõõtmelist süsteemi, kus on kaks klassi, A. rohelised ja B. sinised. Rohelised ja sinised kujundid on juba klassifitseeritud objektid. Punane täht on sisend, mille klass on veel määramata. Sisemine ring tähistab piirkonda $k=3$, kus leiti punasele tähele 3 lähimat naabrit väärtuste x_1 ja x_2 sarnasuse järgi. Klassifitseerimine käib populaarsuse järgi - kui leidub kaks naabrit, mis kuuluvad klassi B, siis ka uuritav objekt kuulub klassi B. Kui nüüd vaadata teist ringjoont, mis tähistab ala $k=5$, siis selles piirkonnas leidub rohkem objekte, mis kuuluvad klassi A, mistõttu klassifitseeritakse uuritav objekt klassi A. See näide illustreerib lähima naabri algoritmi kõige keerulisemat probleemi, mis k väärtus valida.



Joonis 4. Lähima naabri näide.¹

Sarnasus leitakse kasutades erinevaid vektori pikkuse valemeid, millest üks lihtsaim ning populaarseim on Eukleidilise vahemaa valem.

¹<http://www.coxdocs.org/lib/exe/fetch.php?media=perseus:user:activities:matrixprocessing:learning:knn.png>

Olgu meil punktid $P = (p_1, p_2, p_3, \dots, p_n)$ ja $Q = (q_1, q_2, q_3, \dots, q_n)$, siis vahemaa nende punktide vahel on leitav kasutades valemit: $\sqrt{\sum_{i=1}^n (p_i - q_i)^2}$. Mida lähemal punktid on üksteisele, seda sarnasemad objektid on.

Algoritmi saab kohandada nii, et iga kasutaja poolt hinnatud toode on üks väärtus, mis lisab ühe mõõtme süsteemi. Kui kõik kasutajad on andnud hinnangu viiele tootele, siis saab tekitada 5-mõõtmelise ruumi, kus igale kasutajale vastab üks punkt, millel on viis koordinaati. Nüüd saab leida kauguse erinevate punktide vahel ning võetakse k lähimat punkti uuritavale punktile. Kui naabrid on leitud, siis saab uurida naabreid ning leida tooted, millel on kõrgeim kombineeritud hinnang k naabri hinnanguid arvestades. Leitud tooted on soovitusel sobivad tooted. Peale algoritmi kohandamist ei ole tegu enam klassifitseerimisega, siis ei ole tähtis enam valida k väärtus, mis oleks paaritu arv.

Näitena vaadatakse joonist 5. Kasutaja poolt toodetele antud hinnangute tabelit. Tabelis on uuritav kasutaja A märgitud punasega, kellele soovitakse leida soovitusi. Tabelis on ära toodud tooted A-F, ning neile antud hinnangud 5-palli süsteemis. Olemas on andmeid uuritava kasutaja hinnangutest ning selle põhjal saab leida talle kõige sarnasemad kasutajad. Näites leitakse 2 kasutajat, kellel on sarnased hinnangud ning märgitakse nad rohelisega. Kasutaja A hinnangute erinevus kasutaja C hinnangutest erineb ainult 1 võrra ning kasutajaga D erineb 2 võrra.

Leitud sarnased kasutajate hinnanguid uurides leitakse, et toode F meeldis mõlemale kasutajale, seega on tõenäoline, et toode F meeldib ka kasutajale A.

Hinnang	A	B	C	D	E	F
Kasutaja						
A	3	5	-	2	-	-
B	4	4	2	1	1	5
C	3	4	-	2	1	5
D	4	5	-	3	2	5
E	2	2	1	-	1	3

Joonis 5. Kasutaja hinnangute tabel, $k=2$

5.2 Lähima naabri algoritmi tugevused

Üheks tähtsaks tugevuseks on algoritmi sobivus ükskõik milliste objektide puhul, võrreldes algoritmidega, mis nõuavad, et arendaja defineeriks mingid tähtsad parameetrid, mille järgi toimida. Näiteks on väga raske leida see atribuut, mis on tähtis filmi juures, mille põhjal soovitusi luua. Selleks võib olla filmižanrid, päritoluriik, filmi pikkus, aasta, keel või näitleja.

Lisaks on algoritm väga läbipaistev - see aitab kaasa arendamise protsessis vigade leidmisele, hinnangu andmisele algoritmi töös ning ka klientidele selgitamisel, kuidas järeldusele jõuti.

5.3 Lähima naabri algoritmi nõrkused

Esimeseks nõrkuseks on “külmakäivituse probleem” (Cold-start problem). Nagu eelnevalt juba mainitud, on tegu andmete puudumise probleemiga. Kui pole veel küllalt andmeid, siis ei suuda algoritm häid tulemusi saavutada. Kui treeningandmeid on ainult kahe kasutaja kohta ning algoritm peab leidma kaks lähimat naabrit, siis on ilmselge, et pakutavad soovitused on peaaegu juhuslikult valitud, sest kasutajad ei pruugi olla üldse sarnased.

Teiseks andmetega seotud probleemiks on andmete hõredus. Andmete hõredus kujutab endas seda, et isegi kui leidub tuhandeid kasutajaid, kes on oma hinnanguid andnud, ei pruugi leiduda ühtki, kellel oleks kõikidele samadele toodetele hinnangud antud nagu klient, kellele soovitusi luuakse.

Uute toodete leidmine on raske, sest tooteid soovitatakse teiste kasutajate soovitude (hinnangute) põhjal. Kui keegi ei ole veel toodet avastanud või hinnanud, siis toode võib sobida ideaalselt kliendile aga soovitusüsteem ei oska seda soovitada, sest puuduvad vajalikud andmed. Esineb ka vastupidist probleemi - mingid tooted võivad olla väga populaarsed turul ning väga hea mainega, mistõttu enamik kasutajatest annavad tootele kõrge hinnangu. Selle tagajärjeks on väga heade ja kõrgelt hinnatud toodete soovitamine kõigile. See vähendab aga kliendi lootust avastada uusi tooteid, sest klient on arvatavasti ka ise juba selle populaarse tootega tuttav.

Lisaks on võimalik, et leidub väga hea toode, mis ei ole tuntud - kuigi ta on saanud kõrgeid hinnanguid, on hinnangu andnud klientide arv küllalt väike, et toode jääb populaarsemate toodete varju.

Eelnevad olid kõik ärioloogikaga seotud probleemid. Lisaks sellele leidub probleeme ka tehnilise lahenduse poolest. Kuna lähima naabri algoritm on "laisk algoritm" (lazy algorithm), siis süsteem teostab kõik arvutused ja võrdlused sel hetkel, kui seda vaja ning nende andmete põhjal, mis hetkel saadaval on. See lähenemine ei ole väga skaleeruv ning suurte andmehulkadega muutub soovitusüsteem aeglaseks.

5.4 Nõrkuste leevendamine

Antud peatükis proovitakse leida lahendusi eelnevalt loetletud nõrkustele, mis esinevad lähima naabri algoritmi puhul.

5.4.1 Külmakäivituse probleem

Kuna ollakse juba eelnevalt neljandas peatükis rääkinud andmeallikatest ning leidnud, et andmed, mille põhjal luua populaarsete toodete nimekirja on esimesest hetkest peale olemas, siis saab algselt kasutada populaarseid tooteid soovitusena, kuni süsteem muutub küpsemaks ning kogub küllalt andmeid.

5.4.2 Andmete hõredus

Andmete hõreduse probleemiks on see, et omades sadu või tuhandeid tooteid, ei saa loota, et iga kasutaja on hinnanud igat toodet. Esimene ilmselge lahendus on puuduvad hinnangud asendada nullidega. Vaadeldakse eelnevalt uuritud näiteandmeid joonisel 6.

Selle lahendusega tekib olukord, kus ei saa eristada kasutaja hinnangut, kui kasutajale üldse ei meeldinud toode ning hinnati toode nulliga ning kasutajat, kes ei ole hinnangut andnud.

Hinnang	A	B	C	D	E	F
Kasutaja						
A	3	5	0	2	0	0
B	4	4	2	1	1	5
C	3	4	0	2	1	5
D	4	5	0	3	2	5
E	2	2	1	0	1	3

Joonis 6. Kasutaja hinnangute tabel, puuduvad väärtused on nullid.

Teine mure, mida antud lahendus ei leevenda on see, et mõned kasutajad on väga kriitilised ning ei anna kunagi paremat hinnangut, kui 3 palli ning mõni kasutaja on väga lahke ning ei anna ühelegi tootele halvemat hinnangut, kui 3 palli. Nende kahe kasutajatüübi hinne 3 tähendab väga erinevat asja.

Üks viis, kuidas selle probleemi mõju vähendada on hinnangute normaliseerimine. Üheks võimalikuks normaliseerimise valemiks on Gaussi normaliseerimise meetod:

$$\hat{R}_y(x) = \frac{R_y(x) - \bar{R}_y}{\sqrt{\sum_x (R_y(x) - \bar{R}_y)^2}}, \text{ kus}$$

$\hat{R}_y(x)$ – kasutaja y normaliseeritud hinnang tootele x

\bar{R}_y - kasutaja y keskmine hinnang üle kõigi toodete

$R_y(x)$ – kasutaja y hinnang tootele x.

Tekib probleem, et on puuduvaid hinnanguid ning tänu sellele, mõistlikum on kasutada keskmise väärtuse normaliseerimismeetodi.

Kasutaja A puhul keskmine hinnang $\bar{R}_A = (3+5+2) / 3 = 10/3$.

Keskmise väärtuse normaliseerimise käigus lahutatakse teadaolevatest hinnangutest keskmise väärtuse.

$$\hat{R}_y(x) = R_y(x) - \bar{R}_y$$

Määramata hinnangu puhul $\hat{R}_y(x) = 0$

Seega kasutaja A normaliseeritud hinnang tootele A on:

$$\hat{R}_A(A) = R_y(A) - \bar{R}_A = 3 - \frac{10}{3} = -\frac{1}{3}$$

Kasutades keskmise väärtuse normaliseerimist saadakse uus kasutajate hinnangute tabel joonisel 7.

Hinnang	A	B	C	D	E	F
Kasutaja						
A	$-\frac{1}{3}$ (3)	$\frac{5}{3}$ (5)	0	$-\frac{4}{3}$ (2)	0	0
B	$\frac{7}{6}$ (4)	$\frac{7}{6}$ (4)	$-\frac{5}{6}$ (2)	$-\frac{11}{6}$ (1)	$-\frac{11}{6}$ (1)	$\frac{13}{6}$ (5)
C	0 (3)	1 (4)	0	-1 (2)	-2 (1)	2 (5)
D	$\frac{1}{5}$ (4)	$\frac{6}{5}$ (5)	0	$-\frac{4}{5}$ (3)	$-\frac{9}{5}$ (2)	$\frac{6}{5}$ (5)
E	$\frac{1}{5}$ (2)	$\frac{1}{5}$ (2)	$-\frac{4}{5}$ (1)	0	$-\frac{4}{5}$ (1)	$\frac{6}{5}$ (3)

Joonis 7. Kasutaja hinnangute tabel normaliseeritud koos algsete hinnangutega sulgudes.

Vaadates kasutajate D ja E hinnanguid on näha, et kasutaja D annab keskmiselt kõrgemaid hindeid ning kasutaja E annab keskmiselt madalamaid. Peale normaliseerimist on kasutaja D hinnang tootele F $5 \Rightarrow \frac{6}{5}$ ning kasutaja E $3 \Rightarrow \frac{6}{5}$. On näha, et normaliseerimine ühtlustas hinnangute väärtuse keskmise hinnanguga võrreldes ning aitab täpsemalt hinnata kasutaja hinnangut.

5.4.3 Uute toodete leidmine

Uute toodete leidmine läbi soovitude on raske, sest alguses ei ole ühtki hinnangut uutele toodetele, mistõttu on toodete hinnang keskmine. Lahenduseks on tootehaldurite nõuetes mainitud funktsionaalsus, mis võimaldaks määrata tootele kõrgema prioriteedi ning kergitaks toote hinnangut. Toode tuleb tänu sellele soovitudes tihedamini välja.

5.4.4 Algoritmi kiiremaks muutmine

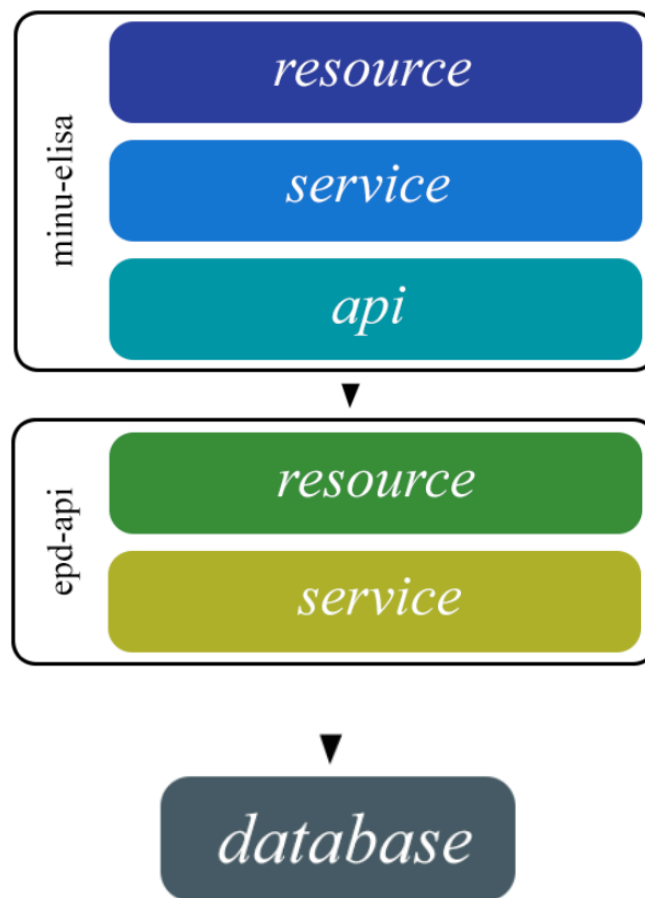
Lähima naabri algoritm on laisk algoritm, mis tähendab, et kõik arvutused ning võrdlused sooritatakse siis, kui süsteemi poole pöörduetakse. Üks levinud viis algoritmi töö vähendamiseks on teatud hulk arvutusi ette ära teha. Kuna arvutused sõltuvad väga palju sellest, mis hinnangutega kasutaja kohta andmeid soovitakse, siis on raske ette valmistada tulemusi, kui kasutaja kohta andmed puuduvad.

Teine võimalus on andmete grupeerimine väiksematesse alamhulkadesse ning seeläbi vähendada tehete arvu. Kuna algoritm eeldab, et iga uuritavat kasutajat võrreldakse kõigi teiste kasutajate andmetega, siis eemaldades eelevalt mingi osa võrreldavate kasutajate hulgast, annab võidu algoritmi kiiruses.

Tooteid saab grupeerida näiteks kategooria alusel. Kui tegu on televiisoriga, siis vaadataksegi ainult kasutajate hinnanguid, kes on hinnanud mõnda toodet, mille kategooriaks on televiisor.

5.5 Algoritmi realiseerimine rakenduses

Minu-elisa rakendus hoiab endas sessiooni haldust ning tõlgete süsteemi. Kõik päringud läbivad algselt minu-elisa kolm kihti, seejärel saadetakse HTTP REST päring epd-apile, kus paikneb äriloojika ning mis pakub andmebaasi suhtluskihi teenust nagu võib näha jooniselt 8. Andmebaasikihti saame käsitleda antud töös ühe andmebaasina. Andmebaasidega suhtlus toimub läbi sama rakenduse ning kõik on MySQL andmebaasid.



Joonis 8. Veebipoe rakenduse kihiline arhitektuur.

Soovitussüsteemi erinevad osad on realiseeritud epd-api teenuskihis teenustena. Tootehinnangute teenus pakub võimalust salvestada hinnanguid andmebaasi ning ligipääsu andmetele. Soovituste pakkumise äriloojika paikneb soovitusteenuses, mis

võtab sisendiks kasutajatunnuse ning toote identifikaatori, mille põhjal luuakse soovitud.

5.5.1 Hinnangute kogumise funktsionaalsus

Kasutaja hinnangu andmisel pöördatakse teenuse poole ning salvestatakse antud kasutaja, hinnang ning toote andmed.

Veebipoe toote detailvaate päringule lisati loogika, mis pöörduv hinnangute teenuse poole ning suurendatakse antud kasutaja hinnangut tootele x palli võrra, kus esialgu $x=1$. Arv x on vahend, millega saab lahendust kalibreerida ning suurendada või vähendada kasutaja käitumise mõju soovitudele. Antud loogika rakendub ainult juhul, kui kasutaja ei ole juba tootele hinnangut andnud. Idee seisneb selles, et kui kasutaja ei ole ise hinnangut andnud, siis vähemalt salvestatakse maha märgi sellest, et kasutaja on huvitatud antud tootest.

Tooteotsingu päring pöörduv samuti hinnangute teenuse poole. Idee on sarnane eelnevalt mainitud Toote detailvaate avamise loogikaga- kui tooted esinevad kasutaja otsingutulemuses, siis võib järeldada, et need tooted vastavad kasutaja eelistustele ning seega suurendatakse hinnangut y võrra, kus $y = 0,5$.

5.5.2 Soovitude juhtimine tootehaldurite poolt

Tootehaldus tööriista lisatud funktsionaalsus võimaldab tootehalduritel määrata tootele hinnangu koefitsiendi, millega saab juhtida, kui tihti toode esineb soovitudes. Kui määrata koefitsient, siis liidetakse koefitsient hinnangu andmisel kasutaja poolt valitud hinnangule. Positiivse koefitsiendi puhul suureneb toote hinnang ning rohkem inimesi tutvuvad tootega, negatiivne koefitsient vähendab toote esinemist soovitudes. Tootehaldustööriista lisatud võimekus määrata koefitsienti on näidatud joonisel 9.

Põhiparameetrid

Kategooriad	<input type="text" value="nutitelefonid"/>
Tootja	<input type="text" value="SAMSUNG"/>
Directo kood	<input type="text" value="SAMA600F"/>
Mudel	<input type="text" value="Samsung Galaxy A6"/>
Soovitus koefitsent	<input type="text" value="2"/>
SEO URL	<input type="text" value="samsung/galaxy-a6"/>
Olek	<input type="text" value="Müügis"/>

Joonis 9. Tootehaldus tööriistas koefitsiendi määramine.

5.5.3 Soovituste pakkumine

Soovituste pakkumise teenus leiab kõik antud tootega samasse kategooriasse kuuluvate toodete hinnangud. Edasi grupeeritakse hinnangud kasutajate kaupa ning normaliseeritakse hinnangud keskmise hinnangu järgi. Kui andmed on ette valmistatud, siis leitakse viis kasutajat, kelle hinnangud on kõige enam soovitusi saava kasutaja hinnangute moodi. Sarnasus leitakse Eukleidiline vahemaa valemi järgi. Iga kasutaja saab numbrilise väärtuse mis iseloomustab sarnasust ning neist valitakse viis kasutajat kelle numbriline väärtus on vähim.

Kokku sobivad tooted on käsitsi määratud tootehaldus tööriistas. Seos kahe seadme vahel luuakse läbi seadme kombinatsiooni tabeli, kus määratakse põhiseade ning talle sobiv lisatarvik. Veebipoes kuvatakse 8 kõige populaarsemat lisatarvikut, 10. joonisel on näha lisatarvikute kuvamise komponent.

Vali sobiv lisaseade

Kõik kokkusobivad lisatarvikud >



**Copter Exoglass ekraanikaitseklaas
Huawei P10 Lite**

24.90€

+ Lisa

**Toshiba M302 Micro-SD mälukaart,
16GB, U1 klass...**

14.90€

+ Lisa

Joonis 10. Lisatarvikute kuvamise komponent.

5.6 Soovituse kasutamine

Soovitusüsteemi edukuse mõõtmiseks on vaja teada, kui paljud kliendid leiavad soovitused üles ning kui paljudele soovitus oli asjakohane. Edukuse mõõtmiseks lisati olemasolevale toote detailvaate päringule kood, mis tuvastab, kas detailvaateni jõuti läbi soovitusel või mingil teisel viisil. Soovitus viitab soovitatud toote detailvaatele, kuhu on lisatud parameeter, mis edastatakse detailvaate päringusse ning seeläbi jõuab informatsioon rakendusserverini. Rakendusserver salvestab kasutaja identifikaatori, toote identifikaatori ning ajahetke, millal soovitus kasutati. Andmebaasi tabel on näha joonisel 11.

webshop_recommendation_reaction		
id	INT	PK
handsetmodel_id	INT	
user_id	VARCHAR(32)	
date	TIMESTAMP	

Joonis 11. Soovituste kasutamise statistika jaoks loodud andmebaasi tabel.

6 Järeldused ja järgmised sammud

Antud töös kirjeldatud sammude tulemusena valmis soovitussüsteem, pakub funktsionaalsust, mis oli esialgselt planeeritud. Sellegipoolest ei saa süsteemi edus kindel olla enne, kui ei ole mõõdetud süsteemi efektiivsust müügi kasvatamisele.

Edaspidi on planeeritud antud süsteemi funktsionaalsus teha avalikuks ettevõtte siseseks testimiseks. Testimise käigus leitud vead on planeeritud parandada ning võimalik, et süsteemi tuleb täiendada. Pärast testimist tehakse süsteem avalikuks veebipoes, kus kliendid saavad hakata hinnanguid andma ning soovitusi saama. Olenevalt hinnangute laekumise kiirusest, valitakse mingi periood, mille järel hakatakse võrdlema müügitulemusi ning soovituste statistikat.

Loodud süsteem võimaldab arendajal süsteemi kalibreerida. Süsteemi erinevaid parameetreid hakatakse perioodiliselt muutma ning analüüsitakse parameetri muutmise mõju müügile. Katsetades leitakse parim valim parameetreid, mis mõjutab müüki kõige positiivsemalt.

7 Kokkuvõte

Käesoleva bakalaureusetöö tulemusena valmis veebipoe toodete soovitusüsteem. Uuriti, millised soovitusüsteemid on kõige enam levinud ning mis on nende tugevused ja nõrkused. Analüüsi veebipoe rakenduse äriloogikat ning tootehaldurite vajaduste põhjal valiti välja kollektiivse filtreerimise põhimõttel töötav soovitusüsteem. Antud rakendus käitub vastavalt püstitatud eeldustele, et sarnastele kasutajatele pakuvad huvi samad tooted. Soovituste pakkumiseks valiti “lähima naabri” algoritm, mis olemasolevale süsteemile sobivaks muudeti.

Andmeallikate analüüsi käigus uuriti olemasolevaid andmeid ning otsustati, milliseid neist saab kasutada soovitude loomiseks. Olemasolevatest andmetest võeti kasutusele tellimuste ajalugu. Teiseks andmeallikaks on kasutaja toodete hinnangud. Loodi erinevaid viise kasutaja hinnangute kogumiseks.

Soovitusüsteem pakub kasutajale võimalust anda oma hinnangut toodete, lisaks jälgib kasutajate käitumist veebipoes ning annab selle põhjal erinevatele toodetele hinnanguid. Hinnangute alusel koostatakse kasutajatele soovitusel, mida kuvatakse klientidele konkreetse toote juures. Klientidel on võimalik soovitud toodetega lähemalt tutvuda-selleks saab klient valida soovitude nimekirjast toote, mis talle huvi pakkus ning klikkides tootele suunatakse klient toote detailvaatesse.

Valminud esimene versioon soovitusüsteemist täidab püstitatud nõuded süsteemile, seega võib lugeda, et antud bakalaureusetöö täitis oma eesmärgi. Edaspidi hakatakse koguma soovitude kohta statistikat, mille põhjal saab otsustada, kui palju soovitusel müüki mõjutavad. Katsetades süsteemi parameetritega otsitakse parimaid sätteid, mis suurendaks müüki.

Kasutatud kirjandus

- [1] Statistikaamet. “E-kaubanduse kasutamine on tõusuteel” <https://blog.stat.ee/2017/01/19/e-kaubanduse-kasutamine-on-tousuteel/> (15.05.2018)
- [2] Netflix prize <https://www.netflixprize.com/index.html> (15.05.2018)
- [3] F. Ricci, L. Rokach, B. Shapira, P.B. Kantor, 2011 “Recommender Systems Handbook“ (15.05.2018)
- [4] F. Meyer, (2006) “Recommender systems in industrial contexts” <https://arxiv.org/pdf/1203.4487.pdf> (15.05.2018)
- [5] M. Nadimi-Shahraki, M. Bahadorpour, (2014) “Cold-start Problem in Collaborative Recommender Systems: Efficient Methods Based on Ask-to-rate Technique” <https://hrcak.srce.hr/file/185686> (15.05.2018)
- [6] W. Nadee, (2016). “MODELLING USER PROFILES FOR RECOMMENDER SYSTEMS” https://eprints.qut.edu.au/93723/1/Wanvimol_Nadee_Thesis.pdf (15.05.2018)
- [7] Euroopa andmekaitse direktiiv. <https://www.eugdpr.org/> (15.05.2018)