Laura Tolvanen

Suosittelualgoritmien käyttö sosiaalisen median yhteisöpalvelualustoilla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Ohjelmistotuotanto

Insinöörityö

1.12.2022

Tiivistelmä

Tekijä:	Laura Tolvanen
Otsikko:	Suosittelualgoritmien käyttö sosiaalisen median yhteisöpalvelualustoilla
Sivumäärä:	70 sivua
Aika:	1.12.2022
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Ohjelmistotuotanto
Ohjaajat:	Ohjaaja Vesa Ollikainen

Tämän insinöörityön tavoitteena oli tutkia tunnettuja suosittelualgoritmimalleja ja vertailla niiden soveltuvuutta sosiaalisen median yhteisöpalvelun verkkoalustalle. Vertailu toteutettiin luomalla valikoiduista suosittelualgoritmeista prototyypit, joiden avulla luotiin lista suosituksia. Suosituksien pohjalta arvioitiin, mikä suosittelualgoritmi sopisi parhaiten sosiaalisen median yhteisöpalvelulle.

Suosittelualgoritmit ovat koneoppimiseen pohjautuvia järjestelmiä. Ne seulovat suuria datamääriä, minkä pohjalta ne pyrkivät luomaan persoonallisia suosituksia eri käyttäjille. Tunnetuimpia ja käytetyimpiä suosittelualgoritmimalleja ovat yhteistoiminnalliset sekä sisältöpohjaiset suosittelualgoritmit, joiden toimintaa käytiin insinöörityössä tarkemmin läpi. Työssä tutustuttiin myös kolmeen samankaltaisuuden mittaan, joita suosittelualgoritmit hyödyntävät suosituksen tuottamisessa.

Monet verkkopohjaiset sosiaalisen median yhteisöpalvelut, kuten Facebook, Youtube ja Instagram, käyttävät suosittelualgoritmeja. Insinöörityössä tarkasteltiin lähemmin, mitä yhteisöpalvelut ovat ja miten käytetyimmät sosiaalisen median yhteisöpalvelut käyttävät suosittelualgoritmeja omilla alustoillaan. Samalla pohdittiin yleisesti sosiaalisen median vaikutusta suosittelualgoritmien toimintaan.

Lopuksi tehtiin kolme suosittelualgoritmiprototyyppiä, joilla mallinnettiin läpikäytyjen suosittelualgoritmimallien toimintaa. Työssä käytettiin kirja-datasettiä, jota hyödynnettiin luomaan valitulle käyttäjälle kirjasuosituksia. Saatujen tulosten pohjalta vertailtiin näiden kolmen suosittelualgoritmin toimintaa ja tehtiin havaintoja, kuinka ne soveltuisivat sosiaalisen median yhteisöpalvelun alustalle.

Avainsanat: suosittelualgoritmi, sosiaalinen media, yhteisöpalvelu

Abstract

Author:	Laura Tolvanen
Title:	Utilization of Recommender Systems in Social Media Networking Services
Number of Pages:	70 pages
Date:	1 December 2022
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communication Technology
Professional Major:	Software Engineering
Supervisors:	Vesa Ollikainen, Principal Lecturer

The goal of the study was to research and examine the most popular recommender algorithms and make comparisons to find out which one would work best in a social media networking service. The comparison was done with three different recommender algorithm prototypes that were used to create a list of recommendations. An evaluation was made based on the recommendations, as to which of them would suit the best for a social media networking service.

Recommender algorithms are systems based on machine learning. They sift through large amounts of data from which they aim to create personalized recommendations to different users. The most known and used recommender algorithms are collaborative filtering and content-based filtering. Here, a closer look is taken on how the algorithms work and also on three different similarity measures that these kinds of algorithms use to create recommendations.

Many web-based social media networking services such as Facebook, Youtube and Instagram use recommender algorithms. In the thesis a further look is taken into what networking services are and how the most used social media networking services utilize recommender algorithms in their platforms. Also, the thesis goes over effects of social media on how the recommender algorithms work.

Lastly, three recommender algorithm prototypes were made that replicate the functions of three different recommender algorithms. A book dataset was used to create recommendations to a selected user. Based on the recommendations, a comparison was made between the three recommender algorithms and how they would work in a social media networking service.

Keywords: recommender algorithm, social media, social networking service

Sisällys

Käsitteet

Suosittelualgoritmi:

> Koneoppimisen muoto, jolla dataa seulomalla pyritään luomaan persoonallisia suosituksia.

Käyttäjäprofiili:

> Yksilöityjä profiileja, johon kerätään käyttäjän antamia syötteitä sekä informaatiota.

Yhteistoiminnallinen suosittelualgoritmi:

> Algoritmi, joka tekee suosituksia perustuen käyttäjien aikaisemmin tehtyihin syötteisiin.

Sisältöpohjainen suosittelualgoritmi:

> Algoritmi, joka tekee suosituksia käyttäjälle ja tuotteille annettujen määritelmien pohjalta.

Samankaltaisuuden mitta:

> Suosittelualgoritmien käyttämä funktio, jota se käyttää samankaltaisten käyttäjien ja tuotteiden seulomisessa.

Yhteisöpalvelu:

> Verkkopohjainen alusta, jossa käyttäjät voivat verkostoitua muiden samankaltaisten käyttäjien kanssa.

Sosiaalinen media:

> Erilaiset verkkosivut ja sovellukset, joiden avulla voidaan jakaa tietoa ja kommunikoida muiden käyttäjien kanssa.

1 Johdanto

Teknologian saavutettavuuden lisääntyessä ihmisten jokapäiväisessä elämässä myös altistuminen erilaisille suosittelualgoritmeille lisääntyy. Monet modernit sosiaalisen median yhteisöpalvelut, kuten Instagram, Facebook ja Youtube, ovat kehittäneet alustoilleen omat yksityiset suosittelualgoritmit. Näiden algoritmien tarkoituksena on kerätä ja analysoida käyttäjän antamia syötteitä. Sen pohjalta algoritmi tuottaa parhaimman mahdollisen suosituksen tuotteesta tai palvelusta, joka kaikista todennäköisimmin kiinnostaisi käyttäjää. Normaalin käyttäjän on kuitenkin usein vaikea tiedostaa näiden algoritmien toimintaa, sillä ne on säädetty toimimaan alustalla lähes huomaamattomasti.

Tämän insinöörityön tavoitteena oli perehtyä käytetyimpiin suosittelualgoritmimalleihin ja tutkia niiden toimintaa sekä soveltuvuutta sosiaaliselle medialle tarkoitetulla yhteisöpalvelualustalla. Käsittelimme myös työssä esiteltävien suosittelualgoritmien vahvuuksia ja heikkouksia.

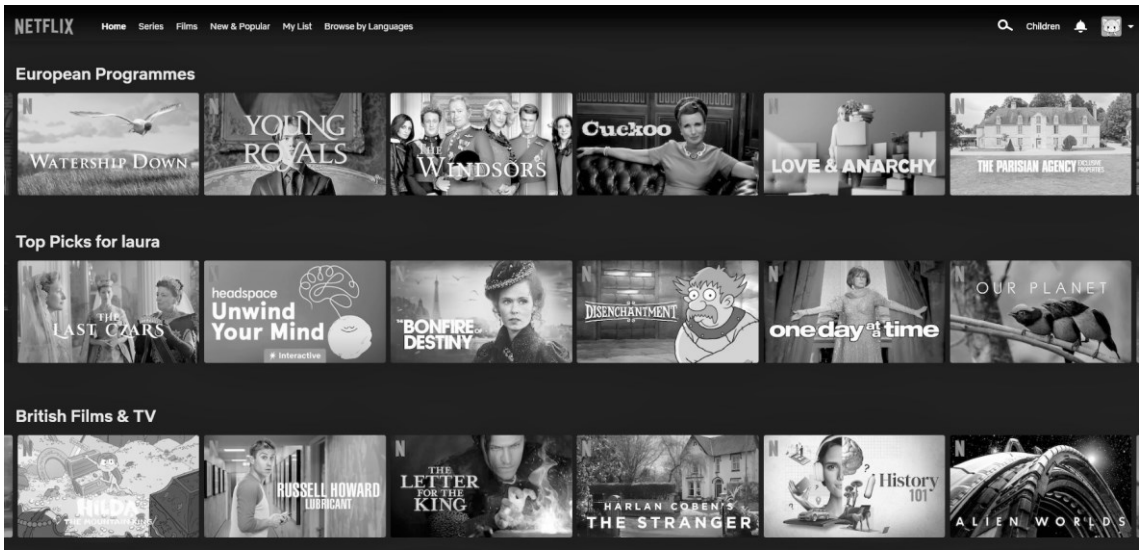
Työn tavoitteiden saavuttamiseksi toteutettiin kolme suosittelualgoritmiprototyyppiä. Näillä prototyypeillä havainnollistettiin tarkemmin käsiteltyjen algoritmien toimintaa. Arvioimme ja vertailemme eri suosittelualgoritmien toimintaa ja tehokkuutta prototyypeillä. Tämän pohjalta saatiin käsitystä siitä, miten sosiaalisen median yhteisöpalvelut saattavat hyödyntää suosittelualgoritmeja omilla alustoillaan. Tuloksena oli tarkoitus selvittää, hyödyllisiä havaintoja keräämällä ja vertailemalla, minkälainen suosittelualgoritmi toimii parhaiten sosiaalisen median yhteisöpalvelulla.

Tämän opinnäytetyön tarkoitus on antaa lukijalle parempi ymmärrys suosittelualgoritmien toiminnasta sekä niiden käytöstä sosiaalisen median yhteisöpalveluissa. Tiettävästi monet tunnetut sosiaalisen median sovellukset sekä sivustot käyttävät algoritmeja. Niiden tarkka toiminta on kuitenkin yrityssalaisuuksien takana. Tämän takia normaalin käyttäjän on vaikea ymmärtää, miten nämä algoritmit oikeasti toimivat ja kuinka niiden avulla käyttäjistä kerättyä tietoa käyte-

tään. Sosiaalisen median jatkaessa kehittymistä ja laajenemista on tärkeää tietää, millä tavoilla omaan käyttökokemukseen sekä kulutusvalintoihin näillä algoritmeilla voidaan pyrkiä vaikuttamaan.

2 Suosittelualgoritmi

Suosittelualgoritmit (recommender algorithm) ovat koneoppimisen (machine learning) muoto. Niitä käytetään työkaluina isojen ja monimutkaisten informaatiomäärien tarkastamiseen ja suodattamiseen (ks. kuva 1). Algoritmilla pyritään luomaan prosessoidun informaation pohjalta personalisoitu näkymä, jossa priorisoidaan käyttäjälle kiinnostavia tuotteita, kuten vaatteita, palveluita tai elokuvia [1, s. 13].



Kuva 1. Netflix luo käyttäjälle Top Picks -listan, johon suosittelualgoritmi on valikoinut käyttäjälle elokuvia, ohjelmia ja sarjoja.

Suosittelualgoritmien käyttötarkoitusta voidaan tulkita monella eri tavalla, jotka voidaan liittää joko algoritmin käyttämän palveluntarjoajan motiiveihin tai palvelun käyttäjän mahdollisiin tarpeisiin. Yhden tulkinnan mukaan suosittelualgoritmeja käytetään kannustamaan käyttäjää suorittamaan haluttuja valintoja, esimerkiksi mainoksen klikkaamista tai tuotteen ostamista. Toisen mukaan taas

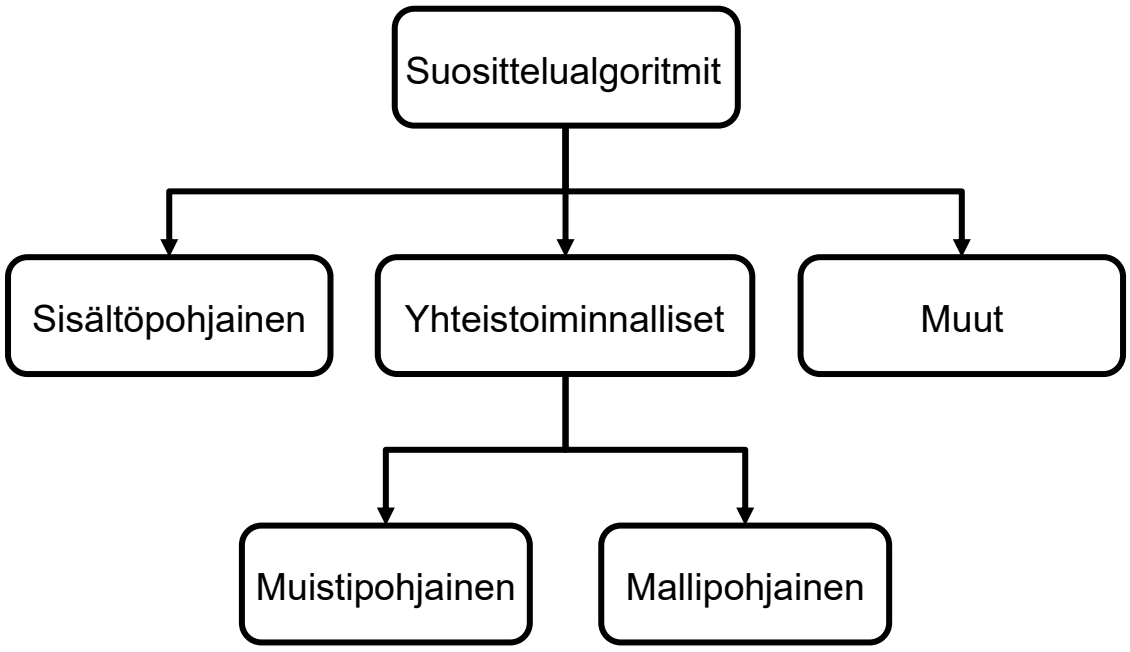
niiden käyttäminen nähdään enemmän datan ylikuormituksen (information overload) vähentämisen työkaluna. [2, s. 3.] Nämä näkökulmat kuitenkin viittaavat siihen, mitä suosittelualgoritmien käyttäjät mahdollisesti etsivät omien nettipohjaisten sivustojensa parantamiseen. Onkin hyvä pohtia myös suosittelualgoritmien käyttötarkoituksia käyttäjien näkökulmasta. Suosittelualgoritmit tarjoavat laajempia hakumahdollisuuksia sekä personalisoituja ehdotuksia ja näkymiä, esimerkiksi suodatettujen ehdotusten esittäminen luettelona, jossa suositukset on järjestetty osuvuuden tai käyttäjäprofiilin personalisoinnin mukaan [3, s. 6–7]. Tämä lisää käyttäjän todennäköisyyttä löytää tuote tai palvelu, josta hän pitää.

Monet nettipohjaiset palvelut kuten Youtube ja Instagram hyödyntävät suosittelualgoritmeja, joilla pyritään parantamaan käyttäjäkokemusta. Ne auttavat käyttäjää löytämään sopivimman vaihtoehdon, esimerkiksi videon tai kuvan. Suosittelualgoritmi tunnistaa tuotteiden välisiä eroja sekä ennustaa arvioitavan tuotteen hyötyä käyttäjälle ja vertaa saatua tulosta muiden tuotteiden kanssa. Tämän pohjalta algoritmi valitsee tuotteita, jotka ovat käyttäjälle suosittelun arvoisia. [3, s. 10.]

Nettipalvelut ja nettisivustot, jotka käyttävät suosittelualgoritmeja, hyödyntävät seulonnassaan käyttäjäprofiileja. Käyttäjäprofiilit ovat yksilöityjä profiileja, joihin kerätään käyttäjän antamia syötteitä sekä informaatiota, kuten klikkauksia, hakuja, ikää, sukupuolta ja arvosteluja. Profiileja voidaan myös luoda ilman käyttäjän antamia syötteitä [4]. Niiden data voi olla eksplisiittistä tai implisiittistä. Tällä jaolla on tarkoitus erotella käyttäjän antamia syötteitä, joiden avulla määritellään kyseisen käyttäjän mahdollisia preferenssejä. Eksplisiittinen data koostuu käyttäjän tekemistä syötteistä, joissa käyttäjän mielipide tuotteelle näkyy selvästi. Tämän kaltaisia syötteitä ovat esimerkiksi tuotteelle annettu numeerinen arvosana tai videolle annettu tykkäys. Implisiittinen data taas kuvastaa epäselvästi käyttäjän mieltymyksiä, joissa käyttäjän preferenssiä on vaikeampi määritellä. Esimerkkejä implisiittisestä datasta ovat muun muassa sivustolla käytetty selailuaika ja mainoksen katselukerrat. Käyttäjäprofiilin avulla tarkennetaan algorit-

min luoman ehdotuksen osuvuutta käyttäjälle sopivammaksi. Algoritmin tuottamille tuloksille on monesti palvelun sivuilla omistettu kohta, jossa käyttäjä pystyy selaamaan hänelle kohdistettuja tuloksia. Monissa tapauksissa algoritmi toimii huomaamattomasti sovelluksen tai sivuston taustalla, jolloin käyttäjän on vaikeampi hahmottaa, kuinka se mahdollisesti vaikuttaa palvelun käyttökokemukseen.

Käytämme päivittäin sovelluksia sekä nettisivustoja, jotka hyödyntävät toiminnassaan suosittelualgoritmeja. Nämä palvelut tuovat harvoin tarkemmin esille, kuinka heidän käyttämänsä algoritmit toimivat. Sosiaalisen median alustojen käyttämien suosittelualgoritmien todellinen toiminta on suojattu tarkoin, mutta voimme silti tutkia ja pohtia asiaa julkisesti tunnettujen algoritmien rakennemallien avulla. Suosittelualgoritmeja voidaan jakaa niiden rakenteen pohjalta erilaisiin malleihin (ks. kuva 2). Tunnetuimpia ja käytetyimpiä malleja ovat yhteistoiminnalliset ja sisältöpohjaiset suosittelualgoritmit. Tarkastellaan seuraavaksi tarkemmin, kuinka nämä algoritmit tuottavat suosituksia yksinkertaisten esimerkkien avulla.



Kuva 2. Hahmotelma suosittelualgoritmien luokittelusta.

2.1 Yhteistoiminnallinen suosittelualgoritmi

Collaborative filtering (CF) eli yhteistoiminnallinen seulonta on suosittelualgoritmi, joka luo suosituksia perustuen muiden käyttäjien mielipiteisiin [5, s. 291]. Algoritmi etsii yhtäläisyyksiä eri käyttäjäprofiilien välillä vertaamalla käyttäjien antamia syötteitä tuotteille, kuten tykkäyksiä tai arvosteluja. Vertailun pohjalta algoritmi jakaa käyttäjät ryhmiin, joissa käyttäjien mieltymykset ovat samankaltaisia. Se ehdottaa kerätyn datan perusteella käyttäjälle vaihtoehtoja, joista samankaltaiset käyttäjät ovat pitäneet ja jotka kaikista todennäköisimmin kiinnostaisivat myös käyttäjää.

Yksi tunnetuimmista yhteistoiminnallisen seulonnan esimerkeistä, jossa kyseistä suosittelualgoritmia on sovellettu, on yksittäisen tuotteen kohdennettu suosittelu, kuten esimerkiksi kirjan, elokuvan tai videopelin. Tarkastellaan seuraavaksi yhteistoiminnallisen seulonnan toimintaa yksinkertaisella kirjan suosittelu esimerkillä, jossa on listattu käyttäjiä ja kirjoja (ks. taulukko 1).

Taulukko 1. Käyttäjien kirja-arvostelut.

Käyttäjä	Kirja 1	Kirja 2	Kirja 3
Ida	2	5	2
Sara	5	1	
Eve		4	3

Tästä taulukosta näemme, miten kolme käyttäjää ovat arvioineet eri kirjoja. Tyhjät kohdat tarkoittavat, ettei käyttäjä ole lukenut kyseistä kirjaa. Ida on lukenut kaikki kolme, mutta Sara ja Eve vain kaksi. Annetusta käyttäjäsyötteistä, eli tässä esimerkissä eksplisiittisistä numeerisista arvosteluista, voimme seuloa eri käyttäjien yhtäläisyyksiä ja luoda mahdollisia suosituksia. Arvosteluista näemme, että käyttäjä Sara on antanut varsin erilaiset arvostelut muihin käyttäjiin verrattuna. Tämän perusteella yhteistoiminnallisessa seulonnassa algoritmi ei

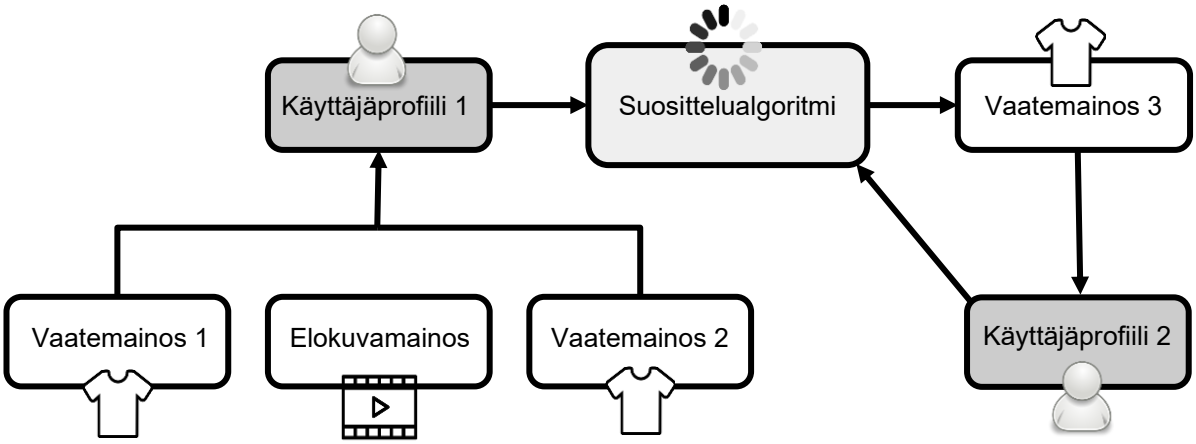
laittaisi Saraa samaan ryhmään käyttäjien Ida ja Eve kanssa. Algoritmi ei myöskään ehdottaisi Saralle kirjoja, joista Ida tai Eve on pitänyt. Toisaalta käyttäjät Ida ja Eve ovat arvostelleet kirjoja samanlaisesti, mistä voimme päätellä, että Eve ei todennäköisesti tule pitämään Kirja 1:stä. Algoritmi laittaisi Idan ja Even samaan ryhmään ja ehdottaisi kummallekin kirjoja, joista toinen on pitänyt.

Usein yhteistoiminnallisia suosittelualgoritmeja jaetaan kahteen eri ryhmään: muisti- ja mallipohjaisiin suosittelualgoritmeihin. Nämä ryhmät erottavat algoritmit sen mukaan, miten ne prosessoivat saamansa dataa suosituksen tuottamiseksi. Muistipohjaiset algoritmit hyödyntävät koko käyttäjäprofiilien tietokantaa, johon sisältyy erilaista käyttäjien dataa esimerkiksi tykkäyksiä ja klikkauksia. Nämä suosittelualgoritmit voidaan jakaa vielä tuote-tuote- tai käyttäjä-käyttäjäpohjaisiin algoritmeihin sen perusteella, miten algoritmi vertailee samankaltaisuuksia. Mallipohjaiset algoritmit taas käyttävät saamaansa käyttäjäprofiilitietokantaa oppiakseen tai määrittääkseen suosittelumallin, jota ne hyödyntävät suositusten tekemiseksi [6, s. 44]. Yksi tunnettu mallipohjainen suosittelualgoritmi on pääakselihajotelma (Singular Value Decomposition). Tässä opinnäytetyössä keskitymme tarkastelemaan muistipohjaisia yhteistoiminnallisia suosittelualgoritmeja sisältöpohjaisten lisäksi.

Käyttäjien, tuotteiden sekä käyttäjän ja tuotteiden välisiä samankaltaisuuksia voidaan arvioida ja laskea erilaisilla samankaltaisuuden mitoilla. Tämä on yksinkertainen tapa määritellä käyttäjien sekä tuotteiden välisiä samankaltaisuuksia. [7, s. 2.] Suosituimpia mittoja, joita suosittelualgoritmeissa on monesti käytetty, ovat kosinin samankaltaisuus (cosine similarity), Pearsonin korrelaatiokerroin (Pearson correlation coefficient) ja Euklidinen etäisyys (Euclidean distance). Hyödynnetään aikaisemmin selostettua kirja-arvostelu esimerkkiä näiden yhteisen seulonnan tyyppien, sisältöpohjaisen seulonnan sekä samankaltaisuuksien mittojen havainnollistamiseen. Käytetään jokaisen seulontatyypin esittämisen yhteydessä erilaista samankaltaisuuden mittaa, jotta saadaan parempi käsitys eri mittojen toiminnasta.

2.1.1 Tuote - tuotepohjainen

Tuote-tuotepohjaisessa (item-item) yhteistoiminnallisessa seulonnassa vertaillaan eri tuotteiden samankaltaisuuksia käyttäjän antamien syötteiden pohjalta. (ks. kuva 3). Algoritmi seuloo käyttäjäprofiilin dataa, esimerkiksi arvioita, klikkauksia tai ostoksia, ja ehdottaa niiden perusteella käyttäjälle samankaltaisia tuotteita [8, s. 4]. Tuotteiden vertailun lisäksi algoritmi ottaa huomioon muiden käyttäjäprofiilien syötteitä tuotteille, minkä pohjalta se valikoi ehdotuksiin tuotteita, joista muut käyttäjät ovat pitäneet. Suuryritys Amazon keksi tuote-tuoteseulonnan ja käytti sitä ensimmäisenä omassa verkkokaupassaan vuonna 1998.



Kuva 3. Tuote-tuotepohjainen yhteistoiminnallinen suosittelualgoritmi, jossa algoritmi luo suosituksen käyttäjäprofiili 1 käyttäjälle hänen tekemien mainosklikkausten sekä toisen käyttäjän tekemän syötteen pohjalta.

Algoritmi voidaan laittaa löytämään tuotteiden A ja B välisiä yhtäläisyyksiä usealla eri samankaltaisuuden mitalla. Yksi tunnetuimmista on mitata kosinikulma kahden vektorin välillä (cosine similarity). Sillä lasketaan kahden vektorin välisen kulman kosini asettamalla vektorit samaan sisätuloavaruuteen eli vektoriavaruuteen. Vektorien välinen samankaltaisuus määrittyy niiden välisen kulman suuruudesta. Jos vektorien välinen kulma on pieni eli lähellä 0 astetta, ne

ovat samankaltaisia keskenään. Päinvastoin jos vektorien välinen kulma on lähempänä 90 astetta, ne ovat enemmän erilaisia kuin samanlaisia.

$$samankaltaisuus\ (A,B)\ =\ cos\ (A,B)\ =\ \frac{A\ \cdot\ B}{\left||A|\right|\ \left||B|\right|}$$

Tarkastellaan tuote-tuote suosittelua aikaisemman kirja-arvostelu taulukon pohjalta (ks. taulukko 1). Tässä esimerkissä algoritmi hyödyntää käyttäjien syötteinä annettuja arvosteluja suosituksen perustana. Eve on lukenut kaksi kolmesta taulukon kirjasta. Tutkitaan, suosittelisiko yhteistoiminnallinen suosittelualgoritmi Evelle Kirja 1:stä, kun käytämme tuote-tuotevertailua. Kahdesta Even antamasta arvostelusta Eve piti eniten Kirja 2:sta. Tuote-tuotepohjaisen seulonnan kosinin samankaltaisuuden mittauksessa, vektorit A ja B ovat tuotteita. Tuote-tuotevertailussa hahmotellaan eri tuotteiden välisiä yhtäläisyyksiä eli katsotaan, kuinka samanlainen Kirja 1 on Kirja 2:n kanssa. Lasketaan kirjojen samankaltaisuus kosinin samankaltaisuuden mitalla, jossa vektori A on tuotteen Kirja 1 arvostelut ja vektori B on tuotteen Kirja 2 arvostelut.

$$A\ \cdot\ B\ =\ Kirja\ 1\ \cdot\ Kirja\ 2\ =2\times5+5\times1+0\times4\ =15$$

$$\left||A|\right|\ =\ \left||Kirja\ 1|\right|\ =\ \sqrt{2^2+5^2\ +0^2}=\sqrt{29}\ \approx5{,}39$$

$$\left||B|\right|\ =\ \left||Kirja\ 2|\right|\ =\ \sqrt{5^2+1^2\ +4^2}=\sqrt{42}\ \approx6{,}48$$

$$samankaltaisuus\ (Kirja\ 1,Kirja\ 2)\ =\ \frac{15}{5{,}39\ \times\ 6{,}48}\ \approx\ 0{,}43$$

$$\theta=\cos^{-1}(0{,}43)\ \approx65°$$

$$erilaisuus\ (Kirja\ 1,Kirja\ 2)\ =\ 1\ -\ cos\ (Kirja\ 1,Kirja\ 2)\ =\ 1\ -\ 0{,}43\ =\ 0{,}57$$

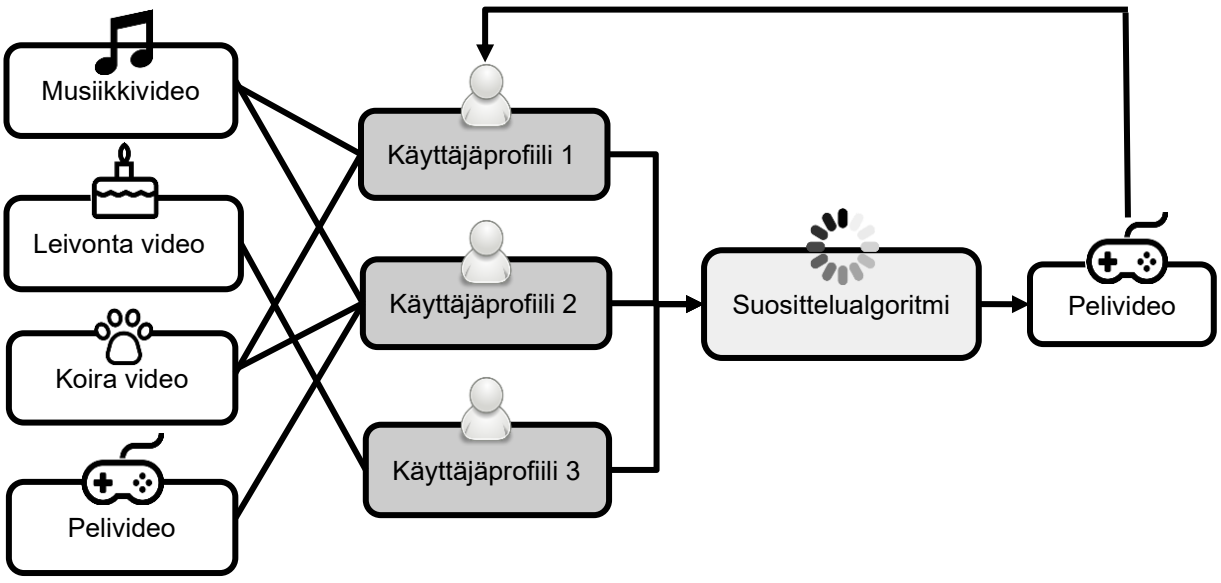
$$\theta=\cos^{-1}(0{,}57)\ \approx55°$$

Kosini samankaltaisuuden kaavalla saadaan selville, että Kirja 1 ja Kirja 2 ovat 0,43 samanlaiset ja 0,57 erilaiset. Kirjojen samankaltaisuuden välisen kosinikulman aste on noin 65°, mikä on lähempänä 90 astetta kuin 0 astetta eli kirjat

ovat enemmän erilaisia kuin samanlaisia. Tämä voidaan varmistaa katsomalla kirjojen erilaisuuden astetta, joka on noin 55°. Kirjojen erilaisuuden aste on lähempänä 0 astetta kuin samankaltaisuuden aste. Tämän menetelmän pohjalta algoritmi ei suosittelisi Evelle Kirja 1:stä, koska se ei ole tarpeeksi samanlainen hänen parhaiten arvostellun kirjan kanssa.

2.1.2 Käyttäjä - käyttäjäpohjainen

Käyttäjä-käyttäjäpohjaisessa (user-user) yhteistoiminnallisessa suosittelualgoritmissa luodaan ehdotus muiden käyttäjien tekemien syötteiden pohjalta (ks. kuva 4). Idea tämän ja tuote-tuotepohjaisen algoritmin takana on, että käyttäjät, jotka ovat olleet aikaisemmin samaa mieltä, ovat todennäköisesti tulevaisuudessakin [9, s. 6]. Algoritmissa verrataan käyttäjän antamia syötteitä muiden käyttäjäprofiilien syötteiden kanssa. Seulonnassa etsitään käyttäjien välisiä samankaltaisuuksia, joita voivat olla ikä tai sukupuoli sekä suoritetut tykkäykset tai videon katselukerrat. Algoritmi ehdottaa kerätyn datan pohjalta suosituksia, joista samanlaiset käyttäjät ovat pitäneet.



Kuva 4. Käyttäjä-käyttäjäpohjainen yhteistoiminnallinen suosittelualgoritmi, jossa algoritmi luo käyttäjäprofiilin 1 käyttäjälle suosituksen vertaamalla hänen ja muiden käyttäjien katsomia videoita.

Toinen tunnettu samankaltaisuuden mitta, jolla voidaan etsiä tuotteiden ja tässä tapauksessa käyttäjien A ja B välisiä samanlaisuuksia, on Pearsonin korrelaatiokerroin (Pearson correlation coefficient). Siinä mitataan, kuinka korreloituja eli vastaavanlaisia kaksi muuttujaa ovat. Kaava palauttaa arvon 1 ja -1 välillä. Mitä lähempänä tulos on arvoa 1, sitä enemmän muuttujat ovat samankaltaisia keskenään, koska niillä on positiivinen korrelaatio. Jos tulos on lähempänä -1, muuttujat ovat enemmän erilaisia kuin samanlaisia keskenään, koska niillä on negatiivinen korrelaatio. Nollatulos tarkoittaa, että muuttujilla ei ole lainkaan korrelaatiota keskenään.

$$samankaltaisuus\ (A,B)\ =\ cor(A,B)\ =\ \frac{\frac{1}{n}\Sigma((A_i\ -\ \bar{A})\ \times\ (B_i\ -\ \bar{B}))}{\frac{1}{n}\sqrt{\Sigma(A_i\ -\ \bar{A})^2}\ \times\ \sqrt{\Sigma\ (B_i\ -\ \bar{B})^2}}$$

Käytetään Pearsonin korrelaatiokerrointa tarkastamaan, kuinka samanlaisia aikaisemman kirja-arvostelutaulukon (ks. taulukko 1) kaksi käyttäjää ovat. Metodissa käytetään vain vertailtavien käyttäjien antamia arvosanoja kirjoille. Tarkastellaan käyttäjien Ida ja Sara välistä samankaltaisuutta. Käytämme tässä tapauksessa Idan ja Saran antamia arvosteluja Kirja 1:lle ja Kirja 2:lle. Sara ei ole lukenut Kirja 3:sta, joten emme voi arvioida hänen ja Idan välistä samanlaisuutta Kirja 3:een liittyen. Luodaan uusi taulukko, johon keräämme tarvittavat tiedot. Lasketaan aluksi käyttäjien arvostelujen tuottamat arvot erikseen ja lisätään ne taulukkoon (ks. taulukko 2).

Taulukko 2. Käyttäjä-käyttäjäesimerkki, käyttäjien Ida ja Sara välillä.

Ida	$Sara$	$I\ \times\ S$	I^2	S^2
2	5	10	4	25
5	1	5	25	1
2	-			

Seuraavaksi laskemme taulukon 2 sarakkeiden arvot yhteen, jotta saamme tarvittavat arvot Pearsonin korrelaatiokertoimen laskemiseen (ks. taulukko 3).

Taulukko 3. Lasketaan sarakkeiden arvot yhteen.

	Ida	$Sara$	$I \times S$	I^2	S^2
Σ	7	6	15	29	26

Sijoitetaan kerätyt arvot Pearsonin korrelaatiokerroin kaavaan ja selvitetään Idan ja Saran välisen korrelaation arvo.

$$cor(I,S) = \frac{n(\Sigma(I \times S) - \Sigma(I) \times \Sigma(S)}{\sqrt{(n\Sigma I^2 - (\Sigma I)^2)\,(n\Sigma S^2 - (\Sigma S)^2)}}$$

$$= \frac{2 \times 15 - 7 \times 6}{\sqrt{(2 \times 29^2 - 7^2)(2 \times 26^2 - 6^2)}}$$

$$= \frac{-12}{\sqrt{2\,149\,028}}$$

$$= -\frac{13}{1466}$$

$$\approx -0{,}002$$

Saimme vastaukseksi $-0{,}002$, mikä tarkoittaa, että käyttäjillä Ida ja Sara on erittäin pieni negatiivinen korrelaatio. Eli toisin sanoen he ovat enemmän erilaisia kuin samanlaisia käyttäjiä. Tämän perusteella algoritmi ehdottaa Saralle Kirja 3:sta, josta Ida ei antanut hyvää arvosanaa. Periaatteena on siis, että Sara todennäköisesti pitää kirjoista, joista Ida ei ole pitänyt – eli kirjoista, joille Ida on antanut huonon arvostelun. Toisaalta saatu korrelaatio on niin pieni, lähes olematon, että algoritmi voisi myös päätellä, että Idalla ja Saralla ei ole lainkaan korrelaatiota. Kun korrelaatio on nolla, käyttäjillä ei ole mitään samankaltaisuuksia, eli algoritmi ei voisi myöskään ehdottaa käyttäjälle Sara mitään käyttäjän Ida arvostelemia kirjoja.

2.1.3 Käyttäjä - tuotepohjainen

Käyttäjä-tuotepohjaiset (user-item) yhteistoiminnalliset suosittelualgoritmit kuuluvat mallipohjaiseen yhteistoiminnalliseen seulontaan, jossa yhdistyvät molemmat muistipohjaiset algoritmimenetelmät. Toisin kuin muistipohjaiset suosittelualgoritmit, jotka etsivät ja vertailevat tuotteiden tai käyttäjien välisiä samankaltaisuuksia kaksiulotteisessa taulukossa (ks. taulukko 1), mallipohjaiset suosittelualgoritmit pyrkivät arvaamaan ja täyttämään taulukon tyhjiä kohtia.

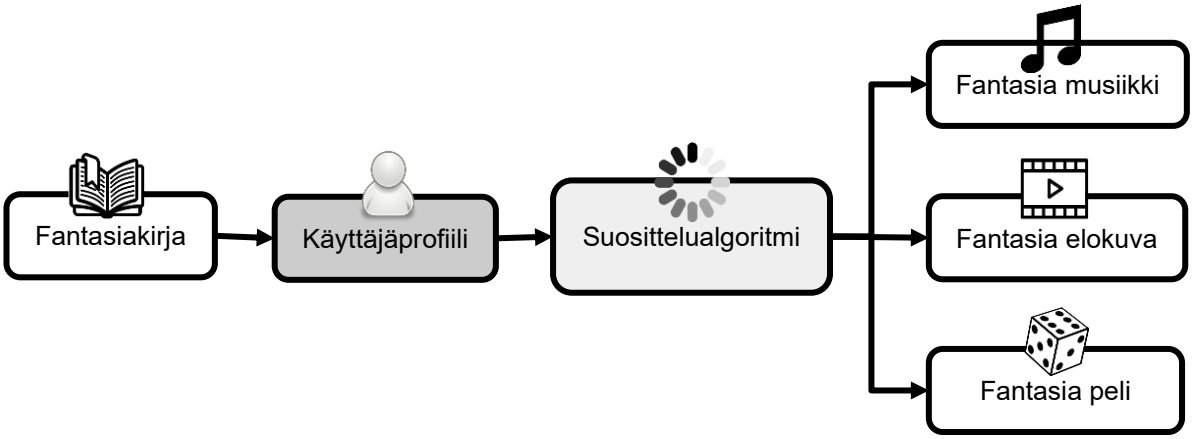
Yksinkertaisimmat ja suosituimmat esimerkit mallipohjaisesta yhteistoiminnallisen seulonnan suosittelualgoritmista pohjautuvat matriisin tekijöihin jakamisesta (matrix factorization) [10]. Sen ideana on jakaa käyttäjä-tuotematriisi omiin pienempiin käyttäjä- ja tuotematriisiin. Tässä menetelmässä algoritmi harjoittelee ja oppii vapaasti suomennettuna matalan ulottuvuuden vektoreita kaikille käyttäjille ja tuotteille. Nämä vektorit auttavat algoritmia ymmärtämään tarkemmin jokaisen käyttäjän tuotemieltymyksiä sekä tuotteiden ominaisuuksia. Harjoittelun pohjalta algoritmi luo yleisen mallin, jolla se ennustaa käyttäjän ja tuotteen välistä suhdetta ja luo sen pohjalta tarkempia suosituksia.

Muutamia tunnetuimpia käyttäjä-tuotepohjaisia yhteistoiminnallisia suosittelualgoritmeja ovat pääakselihajotelma (Singular Value Decomposition), Bayesverkko (Bayesian Networks) sekä vaihteleva pienemmän neliön summa (Alternating Least Square). Vaikka emme käy käyttäjä-tuotepohjaista yhteistoiminnallista suosittelualgoritmia tämän tarkemmin läpi, se on myös hyvin tunnettu ja käytetty suosittelualgoritmimalli, josta on hyvä olla tietoinen.

2.2 Sisältöpohjainen suosittelualgoritmi

Content-based filtering (CBF) eli sisältöpohjainen suosittelualgoritmi luo suosituksia käyttäjän aikaisempien syötteiden ja valmiiksi annetun tuotetietokannan pohjalta. Algoritmi valitsee käyttäjäprofiilin perusteella tuotteita, joista käyttäjä

on pitänyt ja etsii näiden tuotteiden määritelmien ja ominaisuuksien avulla samankaltaisia tuotteita [11, s. 8]. Seulonnassa etsitään käyttäjän ja tuotteen välisiä samankaltaisuuksia, kuten tuotetyyppiä tai siihen liitettyä lajityyppiä, joiden pohjalta algoritmi tuottaa suosituksia (ks. kuva 5). Algoritmi vertailee käyttäjän antamia syötteitä, kuten hakutuloksia ja ostotapahtumia, tuotteiden kanssa, joiden ominaisuudet ja määritelmät ovat samankaltaisia. Tässä tyypissä korostuu tuotteiden määrittelyn tarkkuus. Mitä tarkemmin tietokantaan syötetyt tuotteet on määritelty avainsanoilla, sitä todennäköisemmin algoritmi löytää sopivia tuotteita käyttäjälle suositeltavaksi.



Kuva 5. Esimerkki sisältöpohjaisesta suosittelualgoritmista, jossa algoritmi ehdottaa samaan tyylilajiin kuuluvia tuotteita käyttäjälle.

Kolmas paljon käytetty samankaltaisuuden mitta on Euklidinen etäisyys (Euclidean distance). Sitä käytetään mittaamaan kahden eri vektorin etäisyyttä. Tämä toteutuu asettamalla verrattavat vektorit n-ulottuvuuteen, jossa voimme laskea niiden välisen etäisyyden. Mitä lyhyempi matka A ja B vektorien välillä on, sitä samankaltaisempia ne ovat keskenään.

$$samankaltaisuus(A,B) = euclDist(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

Sovelletaan seuraavaksi Euklidista etäisyyttä aikaisemman kirja-arvostelu esimerkin kanssa. Pyrimme sen avulla määrittelemään, kuinka samankaltaisia jotkin muut tuotteet ovat käyttäjän korkeimmin arvostellun kirjan kanssa. Valitaan vertailtavaksi kirjaksi Saran parhaiten arvosteltu kirja eli Kirja 1. Tätä varten tehdään taulukko tuotteista, joihin Kirja 1:stä vertaillaan. Oletetaan, että Kirja 1 on fantasiakirja. Asetetaan kirjan tuotemäärittelylle yleisesti fantasiakirjoille kuuluvia avainsanoja ja merkitään muut tuotteet arvolla 1, jos niillä on sama määritelmä, tai arvo 0, jos ei ole (ks. taulukko 4).

Taulukko 4. Annetaan arvo 1 tuotemäärittelyille, joita Kirja 1:llä on.

Kirja 1	fantasia	J.R.R. Tolkien	seikkailu	klassikko	fiktio	kirja
Peli	1	0	1	0	1	0
Vaate	1	0	1	0	0	0
Elokuva	1	1	1	1	1	0

Luodun taulukon pohjalta voimme katsoa, kuinka algoritmin seuloma tietokanta löytää tuotteita, joiden ominaisuudet ovat yhteneviä Kirja 1:n kanssa. Tämän vertailun pohjalta voimme laskea kunkin tuotteen ja Kirja 1:n välisen Euklidisen etäisyyden.

$$euclDist(Kirja\ 1, Peli)$$

$$= \sqrt{(1-1)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2}$$

$$\approx 1{,}73$$

$$euclDist(Kirja\ 1, Vaate)$$

$$= \sqrt{(1-1)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2}$$

$$= 2$$

$$euclDist(Kirja\ 1, Elokuva)$$

$$= \sqrt{(1-1)^2 + (1-1)^2 + (1-1)^2 + (1-1)^2 + (1-1)^2 + (1-0)^2}$$

$$= 1$$

Tuloksien perusteella vertailluista tuotteista elokuva on etäisyydeltään lähimpänä Kirja 1:stä. Jos suosittelualgoritmin pitäisi valita yksi tuotteista Saralle ehdotettavaksi, se suosittelisi Euklidisen etäisyyden pohjalta elokuvaa.

2.3 Yleiset vahvuudet ja heikkoudet

Yksi yhteistoiminnallisten suosittelualgoritmien suurimmista vahvuuksista on, että ne eivät tarvitse tarkempaa tietoa käyttäjistä tai tuotteista [12]. Algoritmit saavat tarvitsemansa datan käyttäjien aikaisemmin tehdyistä syötetapahtumista. Tämä tekee yhteistoiminnallisista suosittelualgoritmeista monipuolisesti sovellettavia.

Koska yhteistoiminnalliset suosittelualgoritmit tarvitsevat aikaisempia tapahtumia suosituksien tuottamiseen, yksi yhteistoiminnallisen suosittelijan heikkouksista on kylmäkäynnistys (cold start) [13, s. 33]. Kylmäkäynnistysongelma viittaa siihen, että algoritmin on vaikea luoda suosituksia uusille käyttäjille, joilla ei ole aikaisempaa syötehistoriaa. Myös kun tuotteella ei ole vielä käyttäjien välisiä syötteitä, uusien tuotteiden ehdottaminen hankaloituu, koska algoritmi ei pysty arvioimaan minkälaiset käyttäjät pitävät mistäkin tuotteesta. Tätä ongelmaa on pyritty vähentämään luomalla alkuun tietynlaisia valmiita ehdotuksia, kunnes algoritmi saa tarkempaa tietoa esimerkiksi suosittelemalla uusille käyttäjille yleisesti suosittuja tai satunnaisia tuotteita, tai käyttämällä jotain muuta suosittelualgoritmia uusien käyttäjien ja tuotteiden kanssa.

Yksi syy, miksi sisältöpohjainen suosittelualgoritmi on suosittu, on sen yksinkertaisuus. Sitä voi soveltaa erilaisiin sovelluksiin sekä alustoihin, koska se vaatii vain käyttäjän sekä ennalta määritellyn tietokannan. Koska algoritmi ei tarvitse

muiden käyttäjien syötteitä suosituksen tuottamiseen, sisältöpohjaisten suosittelualgoritmien tuottamat suositukset käyttäjälle ovat usein tarkkoja ja osuvia. Sisältöpohjaiset suosittelualgoritmit eivät myöskään kärsi kylmäkäynnistysongelmasta niin kuin yhteistoiminnalliset suosittelualgoritmit. Algoritmi pystyy luomaan uusille käyttäjille suosituksia käyttäjän antamien tietojen, kuten sukupuolen, iän ja asuinmaan, mukaan sekä ehdottamaan uusia lisättyjä tuotteita niille annettujen määritysten ja ominaisuuksien pohjalta. Eli jos esimerkiksi juuri julkaistulla alustalla on vähän käyttäjiä, algoritmin ehdotukset pysyvät silti suhteellisen hyvinä.

Toisinaan sisältöpohjaisilta suosittelualgoritmeilta puuttuu ehdotusten monipuolisuus, koska niiden on huomattu ehdottavan samantyyppisiä tuotteita [13, s. 34]. Tämä voi vähentää isojen, eri tuotteita sisältävien, alustojen käytön tehokkuutta, mikä taas vaikuttaa käyttäjien käyttökokemukseen negatiivisesti. Algoritmin antamien suositusten osuvuus riippuu myös merkittävästi tietokannassa olevien tuotteiden avainsanojen määrittelyn täsmällisyydessä. Määritelmien tulee olla tarpeeksi yksityiskohtaisia, jotta algoritmi pystyy luomaan toimivia käyttäjämieltymyksiä [1, s. 16]. Jos määrittelystä puuttuu jokin olennainen termi tai sana, algoritmi ei pysty ehdottamaan tuotetta joillekin käyttäjille, mikä vähentää algoritmin tehokkuutta. Myös käyttäjien antamat tiedot voivat vaikuttaa suositusten osuvuuteen.

Kummallakin suosittelualgoritmilla on selviä vahvuuksia ja heikkouksia, jotka ilmenevät riippuen siitä, miten niitä hyödynnetään. Suosittelualgoritmista on vaikeaa tehdä täydellistä sekä jokaiseen käyttötarkoitukseen sopivaa, minkä takia on tehty useita erilaisia suosittelualgoritmeja näiden lisäksi.

2.4 Hybridialgoritmit sekä tietopohjaiset ja muut suosittelualgoritmit

Suosittelualgoritmit kehittyvät jatkuvasti, mikä antaa tilaa uusille suosittelualgoritmimalleille. Yhteistoiminnallisen ja sisältöpohjaisen seulonnan suosittelualgo-

ritmien lisäksi löytyy myös muita algoritmeja. Yksi näistä on hybridisuosittelualgoritmi, jossa sekoitetaan kahta tai useampaa yhteistoiminnallisen sekä sisältöpohjaisen seulonnan algoritmimallia. Hybridialgoritmeilla pyritään hyödyntämään muiden suosittelualgoritmien vahvuuksia ja välttämään niissä esiintyviä rajoituksia sekä ongelmia, kuten kylmäkäynnistystä ja datan puutteellisuutta [14, s. 2–3]. Vaikka hybridisuosittelualgoritmit ovat uudempia muihin suosittelualgoritmeihin verrattuna, niiden käyttö on viime vuosina yleistynyt. Yksi syy tälle voi olla, että hybridimallit tarjoavat tarkempia suosittelutuloksia kuin mitä aikaisemmat suosittelualgoritmit.

Toinen algoritmi on tietopohjainen suosittelualgoritmi. Tämä algoritmi hyödyntää käyttäjistä ja tuotteista saamaansa tietoa luomaan perusteluita tuotteiden sopivuudesta käyttäjälle. [15, s. 69.] Algoritmi ei käytä käyttäjän antamaa eksplisiittistä tai implisiittistä dataa, vaan se pyrkii saamaan tarkempaa tietoa hyödyntämällä käyttäjälle annettuja kyselyjä tai ominaisuuksia, joita käyttäjältä pyydetään ennen suosituksen tuottamista. Näitä voivat olla esimerkiksi erilaiset hakuominaisuudet, joilla hakua voidaan rajata.

Hybridi- ja tietopohjaisten algoritmien lisäksi on suosittelualgoritmeja, joiden toiminta perustuu syväoppimiseen (deep learning), väestöryhmittelyyn (demographic based), hyödyllisyyteen (utility based) sekä moneen muuhun. Näistä algoritmeista monien käyttö on usein hyvin rajattu tietyille käyttötarkoituksille, mikä tekee niiden laajamittaisemmasta soveltamisesta vaikeampaa. Suosittelualgoritmien kehittyessä entistä tarkemmiksi ja tehokkaammiksi, on mahdollista, että näiden erityisen tarkasti suunniteltujen, rajattuun käyttötarkoitukseen valmistettujen algoritmien yleisyys kasvaisi tulevaisuudessa.

3 Yhteisöpalvelu

Yhteisöpalvelulla (social networking service) viitataan verkkopohjaiseen alustaan, jonka käyttäjät pääsevät verkostoitumaan muiden käyttäjien kanssa, joilla on samoja mielenkiinnon kohteita, mielipiteitä ja muita samankaltaisuuksia [16,

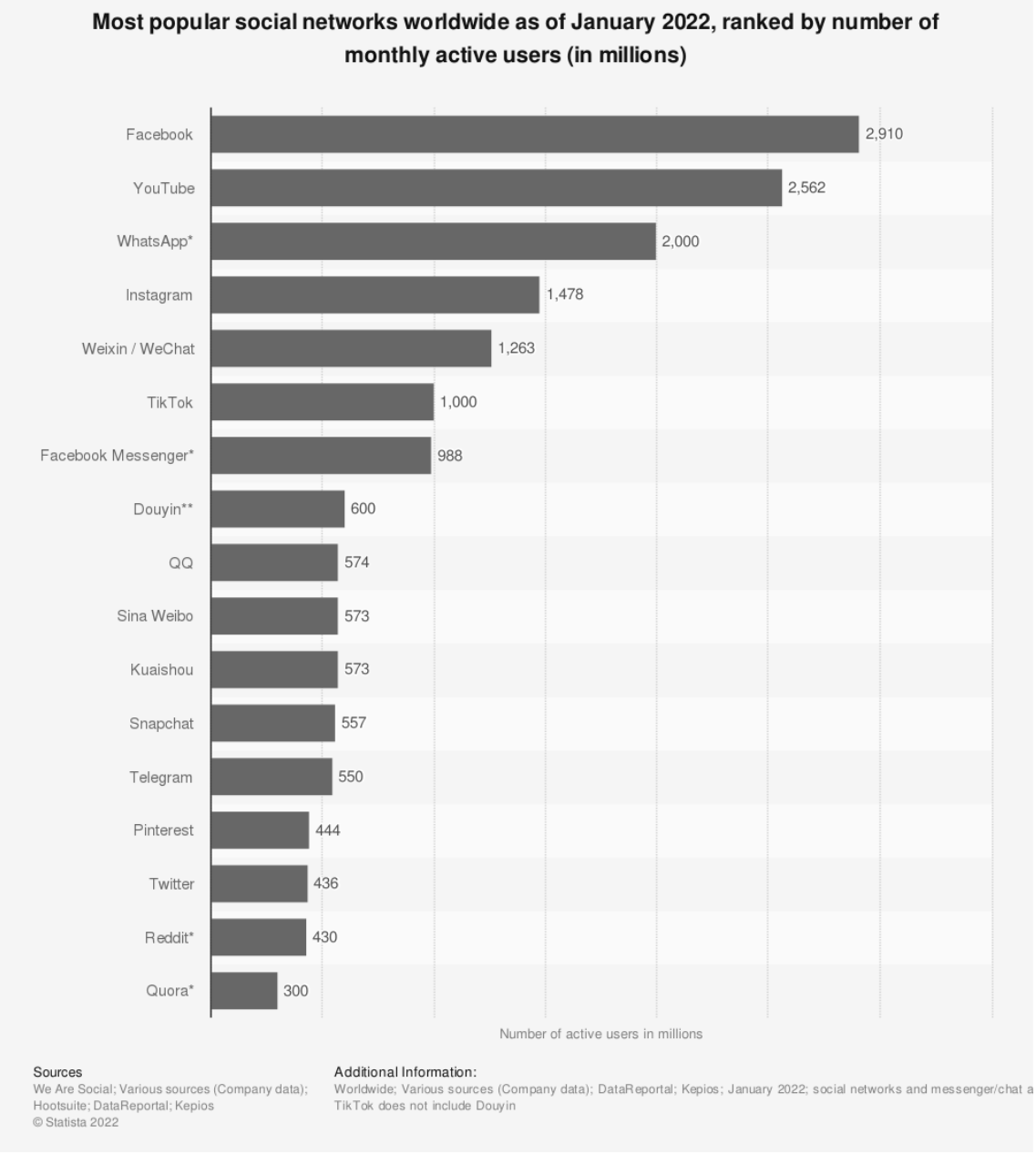
s. 126]. Näiden palveluiden käyttäjänä voi toimia lähes kuka vain, esimerkiksi yksityishenkilö, yritys tai työntekijä. Monet yhteisöpalvelut ovat ottaneet eri käyttäjäkuntia huomioon lisäämällä heitä hyödyttäviä ominaisuuksia: esimerkiksi Instagramissa pystyy vaihtamaan henkilökohtaisen käyttäjätilin yritystiliksi. Yhteisöpalveluita on monenlaisia. LinkedIn on ammatillinen verkkoalusta, jonka kautta käyttäjät voivat jakaa ansioluettelonsa ja verkostoitua muiden alansa ammattilaisten kanssa. TikTok taas on mobiilisovellus, jossa käyttäjät voivat luoda, julkaista, kommentoida ja katsella lyhyitä videopätkiä. Vaikka LinkedIn ja TikTok ovat käyttötarkoituksiltaan hyvin erilaisia, molemmat voidaan silti luokitella yhteisöpalveluiksi.

Jotta käyttäjä pääsisi käyttämään haluamaansa palvelua, hänen täytyy usein ensin luoda ilmainen käyttäjäprofiili palveluun lisäämällä henkilökohtaisia tietoja itsestään ja hyväksymällä palvelun käyttöehdot. Käyttäjällä on monesti vapaus määritellä, mitä henkilökohtaisia tietoja sekä palvelussa suoritettuja toimintoja hän haluaa muiden käyttäjien näkevän. Profiilin luotuaan käyttäjä pääsee hyödyntämään kyseisen yhteisöpalvelun ominaisuuksia valitsemalla häntä kiinnostavia sisältöjä. Internetin saatavuuden lisääntyessä yhä useammalla henkilöllä on pääsy yhteisöpalveluihin.

Yksi tapa, miten nämä yhteisöpalvelut hyödyntävät palveluissaan suosittelualgoritmeja on kerätä käyttäjän antamia syötteitä, minkä pohjalta palveluun lisätty suosittelualgoritmi ehdottaa käyttäjälle häntä mahdollisesti kiinnostavia aiheita, muita käyttäjiä ja sisältöä. Näiden suosittelualgoritmien tarkka tekninen toiminta on hyvin salattu, mutta voimme yrittää arvioida niiden mahdollista perusrakennetta joidenkin yhteisöpalveluiden antamien julkisten tietojen avulla.

Yhteisöpalveluita on tällä hetkellä tuhansia ja uusia julkaistaan jatkuvasti. Nykyisten joukosta löytyy selviä maailmanlaajuisia suosikkeja (ks. kuva 6). Otetaan käytetyimmistä yhteisöpalveluista kolme lähempään tarkasteluun, jotta saisimme paremman käsityksen yleisten yhteisöpalvelujen toiminnasta. Nämä kolme yhteisöpalvelua ovat Facebook, Youtube ja Instagram. Katsotaan myös,

miten kyseiset yhteisöpalvelut ovat itse selittäneet suosittelualgoritmiensa toimintaa. Päättelemme sen pohjalta, minkälaista suosittelualgoritmimallia niiden suosittelijat saattaisivat käyttää. Vilkaistaan myös lyhyesti näiden suosittelualgoritmien ongelmia. Käytämme myöhemmin tätä kerättyä tietoa perustana suosittelualgoritmien vertailussa.



Kuva 6. Käytetyimmät yhteisöpalvelut tammikuussa vuonna 2022 [17].

3.1 Facebookin suosittelualgoritmin toiminta

Facebook on maailmanlaajuisesti käytetyin sosiaalisen median yhteisöpalvelu. Se on Meta Platforms Inc. yhtiön alaisuuteen kuuluva verkkopalvelu, joka auttaa käyttäjää pitämään yhteyttä ystäviin, perheeseen ja yhteisöihin, jotka jakavat käyttäjän mielenkiinnon kohteita [18]. Facebook on vuonna 2004 luotu ja nyt maailman suurin yhteisöpalvelu, jolla on kuukausittain melkein kolme miljardia aktiivista käyttäjää. Facebookissa käyttäjä voi jakaa monenlaista sisältöä, kuten viestejä, video- ja äänitiedostoja. Sen käyttötarkoitus kohdistuu erilaisten yhteisöjen, kuten perheenjäsenistä, ystävistä, kollegoista, jonkun henkilön tai aiheen faneista koostuvien ryhmien, ylläpidosta. Vuosien saatossa Facebook on kehittynyt yksinkertaisesta yhteydenpitopalvelusta yhdeksi verkkoympäristön keskeisimmäksi ja monipuolisimmaksi yhteisöpalveluksi. Tehokkaan ylläpidon lisäksi se on pysynyt kilpailukykyisenä päivittämällä palvelua uusilla ominaisuuksilla, esimerkiksi Facebook Marketplace, Facebook Live ja Facebook Portal. Näiden sekä muiden lisättyjen toimintojen ansiosta Facebook on pysynyt tähän asti ajankohtaisena yhteisöpalveluna. Jatkuneen suosionsa takia monet muut verkkopalvelut ovat yhdistäneet Facebookin omiin verkkosivuihinsa ja sovelluksiinsa antamalla käyttäjän kirjautua sisään Facebook-tunnusten avulla.

Facebookin sisällön suosittelualgoritmin tekninen toiminta on hyvin salattu tieto. Voimme kuitenkin tarkastella, miten sen algoritmi toimii Facebookin mukaan ja arvioida, minkälaista seulontatapaa sen suosittelualgoritmi mahdollisesti käyttää. Facebookin algoritmi suorittaa aina neljä vaihetta, minkä avulla se seuloo käyttäjälle häntä kiinnostavaa sisältöä [19].

Nämä neljä vaihetta ovat:

1. Valikoima: Mitä sisältöä kaverit ja julkaisijat ovat julkaisseet?

2. Signaalit: Kuka julkaisi tarinan?

3. Ennusteet: Kuinka todennäköisesti sitoudut julkaisuun?

4. Pisteytys: Kuinka kiinnostuneita käyttäjät ovat julkaisusta?

Näiden vaiheiden läpikäymisen aikana suosittelualgoritmi kerää tietoa viimeisimmistä julkaisuista, joita käyttäjän ystävät ja seuratut sivut ovat julkaisseet. Facebook ottaa huomioon monta eri vaikuttavaa tekijää suosituksen seulomisessa, kuten selausalustan, julkaisujen tekijän, julkaisuajan, käyttäjän aikaisemmat syötteet saman julkaisijan aikaisemmin tekemiin julkaisuihin ja käyttäjän internetin nopeuden. Eli algoritmi käsittelee niin käyttäjäprofiileista saamaansa eksplisiittistä dataa kuin myös alustan sisällölle määriteltyä dataa. Tämä viittaisi yhteistoiminnalliseen suosittelualgoritmiin tai jonkinlaiseen hybridi suosittelualgoritmiin.

Facebookin algoritmi käyttää myös ennusteita, joilla se arvioi sisällön merkityksellisyyttä käyttäjälle. Se pyrkii ennakoimaan todennäköisyyttä sille, että käyttäjä suorittaisi jotain jatkotoimia, kuten suositellun sisällön kommentointia tai jakamista. Voisiko kyseessä olla mallipohjainen käyttäjä-tuoteyhteistoiminnallinen suosittelualgoritmi? Kuinka algoritmi tämän ennakoinnin toteuttaa ja mitä informaatiota se käyttää näiden ennustusten saamiseksi ei ole kuitenkaan julkisesti tiedossa. Algoritmi luo tämä vaiheen aikana monia ennakoituja todennäköisyyksiä esimerkiksi klikkausten, ajan käytön, informatiivisen kokemuksen sekä klikkien kalastelun todennäköisyyttä.

Lopuksi suosittelualgoritmi antaa ennusteiden pohjalta suositukselle osuvuuspistemäärän. Tämä pisteytys kuvaa sitä, miten paljon algoritmi uskoo käyttäjän olevan kiinnostunut kyseisestä sisällöstä. Pisteytyksen perusteella algoritmi sijoittaa julkaisuehdotukset käyttäjälle niiden osuvuuden mukaan. Algoritmi suorittaa tämän prosessin joka kerta, kun käyttäjä avaa Facebookin. Facebook ei kuitenkaan selvennä, kuinka paljon esimerkiksi kohdennetut mainokset ja muut palvelut vaikuttavat algoritmin tekemiin suosituksiin. Käyttäjillä ei ole tietoa siitä, kuinka paljon suosittelualgoritmi antaa arvoa eri todennäköisyyksille. Suosiiko se esimerkiksi jotain todennäköisyyttä toisia enemmän?

Facebookin suosittelualgoritmia on kritisoitu viime vuosina siitä, kuinka algoritmi kannustaa antisosiaalisia tapoja, kuten vihapuhetta sekä äärimmäisiä poliittisia asenteita [20]. Vuonna 2021 yhdysvaltalainen Frances Haugen, joka työskenteli aiemmin Facebookille, syytti entisen työnantajansa tietoisesti salaavan olennaista tietoa sen käyttäjiltä ja maailman hallituksilta [21]. Tämä salailu on Haugenin mukaan johtanut valheisiin, ihmisten jakautumiseen ja jopa väkivaltaan. Kaikesta kritiikistä huolimatta Facebookin johto ei ole jakanut julkisuuteen sen tarkempaa tietoa algoritmiensa toiminnasta.

Facebookin suosittelualgoritmi on nykyään hyvin tehokas ja edistyksellinen, mikä auttaa lukuisia käyttäjiä saamaan heitä kiinnostavaa sisältöä. Näitä ansioita on kuitenkin vaikea arvostaa tietäen, miten sen toiminta on mahdollisesti vaikuttanut muihin käyttäjiin ja saattaa edelleenkin vaikuttaa. Tämän takia on tärkeää ymmärtää suosittelualgoritmien toimintaa ja olla jatkuvasti tietoinen sisällön mahdollisista asiayhteyksistä. Vaikka suosittelualgoritmimallit eivät itsessään tuota haitallisia suosituksia ilman ulkopuolisia vaikutteita, voivatko kuitenkin jotkut algoritmimallit olla alttiimpia tai taipuvaisempia tämänkaltaisten haitallisten suositusten tekemiselle?

3.2 Youtuben suosittelualgoritmin toiminta

Youtube on tunnettu videonjakopalvelu, jonka kautta voi etsiä, katsoa ja jakaa videoita ja muuta sisältöä. Se toimii yhteisöpalveluna sisällöntuottajille sekä mainostajille ja tarjoaa foorumin, jossa ihmiset voivat inspiroida ja luoda yhteyksiä kaikkialle [22]. Youtube luotiin vuonna 2005, minkä jälkeen miljoonat käyttäjät ovat ladanneet sinne erilaisia videoita. Googlen ostettua Youtuben omistusoikeudet vuonna 2006, Youtube alkoi kasvattamaan suosiotaan käyttäjien keskuudessa. Samanlaiset yhteisöpalvelut kuten Vimeo ja Dailymotion eivät pysyneet Youtuben käyttäjämäärän kasvun mukana, minkä takia Youtube on tällä hetkellä maailman suosituin videonjakelupalvelu. Youtuben vuosia kestänyt suosio osoittaa, kuinka suosittua laadukkaiden videoiden julkaiseminen ja nii-

den jakaminen muille käyttäjille on. Tällä on voinut olla vaikutus uudempien yhteisöpalvelujen, kuten TikTokin ja Instagramin sisältämiin videonjakeluominaisuuksiin.

Vuonna 2021, Youtuben tekniikan varapääjohtaja Cristos Goodrow julkaisi Youtuben viralliseen blogiin kirjoituksen, jossa hän avasi tarkemmin Youtuben algoritmin toimintaa [23]. Julkaisussa Goodrow selitti, kuinka Youtuben suosittelualgoritmi alun perin suositteli käyttäjille videoita niiden katsomiskertojen pohjalta, minkä takia käyttäjät saivat monesti tylsiä suosituksia. Eli todennäköisesti kyseessä oli jonkinlainen sisältöpohjainen algoritmi, mutta se ei käyttänyt käyttäjäprofiiliin dataa, minkä avulla se olisi voinut tuottaa kiinnostavampia suosituksia.

Youtuben kehittäjät huomasivat tämän ongelman ja ymmärsivät käyttäjillä olevan henkilökohtaisia katselumieltymyksiä. Nykyään Youtuben suosittelualgoritmi vertailee käyttäjän katselutottumuksia, kuten haku- ja katseluhistoriaa, muihin käyttäjiin, joilla on samanlaisia katselutottumuksia ja ehdottaa sen pohjalta käyttäjälle sisältöä. Tästä voimme päätellä, että nykyisen Youtuben suosittelualgoritmin rakenne vaikuttaa paljon yhteistoiminnallisen suodattamisen käyttäjä-käyttäjäpohjaiselta suosittelualgoritmilta. Goodrow huomauttaa, että toisin kuin muut yhteisöpalvelualustat, he eivät yhdistä katsojia sisältöihin heidän sosiaalisten verkostojensa kautta. Eli algoritmi ei ehdota sisältöä esimerkiksi seurattujen käyttäjien tai käyttäjää seuraavien käyttäjien katselutottumusten perusteella niin kuin Facebook. Sen sijaan algoritmi mukauttaa ehdotuksensa käyttäjän omien henkilökohtaisten syötteiden, kuten klikkausten, katseluajan, jakojen sekä tykkäysten perusteella.

Viime vuosien aikana Youtuben käyttäjien keskuudessa on ollut paljon puhetta Youtuben algoritmin toiminnasta. Goodrow pyrkii selittämään, että Youtube ei pysty olemaan avoin suosittelualgoritminsa toiminnasta, koska sen rakenne ei ole niin yksinkertaisesti selitettävissä. Algoritmi ei toimi tietyn mallin tai kaavan pohjalta, vaan se kehittyy jatkuvasti ja oppii sitä mukaan, kun käyttäjät lataavat palveluun sisältöä ja muuttavat katselutottumuksiaan. Me tiedämme, miten You-

tuben suosittelualgoritmi hyödyntää käyttäjäsyötteitä ehdotuksien suodattamiseksi, mutta emme tiedä, mitä kaikkia muita, käyttäjäprofiilin ulkopuolisia syötteitä, se seulonnassaan mahdollisesti käyttää. Algoritmin ollessa jatkuvassa dynaamisessa opetustilassa, oppien jokaisen yksittäisen käyttäjän katselumieltymyksiä, se saattaa myös helpommin altistua suositteluongelmille.

Youtube myönsi vuonna 2019 julkaistussa blogikirjoituksessa, että heidän suosittelualgoritminsa on ehdottanut käyttäjilleen haitallista sisältöä, kuten virheellistä tietoa [24]. He kuitenkin vakuuttavat, että he ovat kehittäneet suosittelualgoritminsa toimintaa ongelmallisen sisällön suosittelun vähentämiseksi, esimerkiksi alentamalla kyseenalaisen sisällön näkyvyyttä Youtubessa. Yleishyödyllinen organisaatio Mozilla Foundation on tuonut esille, että Youtube ei anna näille väitteille minkäänlaista tukevaa dataa tai todisteita niiden todenmukaisuudesta. Organisaatio on myös kritisoinut, kuinka Youtube ei ota kantaa journalistien ja tutkijoiden tekemiin löytöihin Youtuben algoritmin suosituksiin liittyen. [25.] Eli vaikka Youtube on tietoinen algoritminsa ongelmista, he eivät jaa tietoa siitä, miten algoritmia muutetaan niiden korjaamiseksi. Käyttäjien on tämän takia vaikea hahmottaa, miten nämä muutokset tulevat vaikuttamaan heidän käyttökokemuksiin.

Youtuben toimintamalli antaa sen suosittelualgoritmille mahdollisuuden toimia monipuolisesti usealle käyttäjälle samaan aikaan, mutta palvelun avoimuus altistaa sen myös suositteluongelmille, jotka voivat olla haitaksi palvelun käyttäjille. Suosittelualgoritmeille tulisi Youtuben kaltaisissa yhteisöpalveluissa kehittää tällaisten ongelmien välttämiseksi rajoituksia, jotka algoritmi ottaa suosituksissa huomioon, esimerkiksi ajan milloin käyttäjä on luonut profiilinsa. Saattaa olla, että niin on jo tehty. On kuitenkin vaikea arvioida, miten pienet muutokset saattavat vaikuttaa laajemmin algoritmin suosituksiin ja niiden laatuun.

3.3 Instagramin suosittelualgoritmien toiminta

Instagram on myös yksi, vuodesta 2012 lähtien, Meta Platforms Inc. yhtiön omistama kuvien ja videoiden jakamiseen tarkoitettu sovellus. Sen käyttäjät voivat ladata kuvia ja videoita, joita he voivat jakaa seuraajiensa tai valitsemansa ryhmän kanssa [26]. Tämä vuonna 2010 julkaistu yhteisöpalvelu on nopeasti noussut suosituksi niin yksityishenkilöiden ja kuuluisuuksien kuin myös yritysten ja brändien keskuudessa. Instagramin käyttäjät voivat Facebookin tavoin seurata muita käyttäjiä ja viestitellä yksityisesti heidän kanssaan. Käyttäjät voivat myös tykätä, kommentoida sekä jakaa muiden julkaisemaa sisältöä. Mikä tekee Instagramista erikoisen, on sen visuaalisen sisällön priorisointi. Käyttäjät voivat julkaista ja jakaa vain kuvia ja videoita, joiden selostukseen ja kommentteihin he voivat lisätä tekstiä, hashtageja sekä linkkejä. Instagramin tarjoamien editointi- ja filtteriominaisuuksien ansiosta käyttäjät voivat muokata kuvistaan ja videoistaan heille kauniimpia. Keskittymällä tämän yhden perusominaisuuden parantamiseen kuin myös lukuisien uusien toimintojen julkaisemiseen, Instagram kuuluu nykyään suosituimpien yhteisöpalvelujen joukkoon.

Vuonna 2021 Instagram julkaisi blogissaan sarjan kirjoituksia, joissa selitettiin tarkemmin Instagramin käyttämää teknologiaa [27]. Nämä julkaisut olivat mahdollisesti seurausta Facebookin samana vuonna saamaan, aikaisemmin mainittuun, kritiikkiin. Kirjoituksissa nostetaan heti esille, kuinka Instagramin sisällä toimii useita eri tehtäviin tarkoitettuja algoritmeja, lajitteluja sekä prosesseja. Alussa Instagramin algoritmin seulontatapa oli hyvin yksinkertainen; käyttäjälle näytettiin julkaisuja niiden ilmestymisen perusteella. Eli alun perin Instagramin suosittelualgoritmi saattoi hyödyntää sisältöpohjaista suosittelua, jossa käyttäjälle suositeltiin sisältöä sen julkaisuajan mukaan. Mutta pian käyttäjien lisääntyessä myös julkaisujen määrä lisääntyi, eikä käyttäjillä ollut mahdollisuuksia nähdä heitä kiinnostavia julkaisuja. Kyseisen ongelman huomattuaan Instagram otti käyttöön Syöte-osion, joka sijoittaa julkaisut sen mukaan, mistä käyttäjä välittää eniten. Kaikki Instagramin nykyiset ominaisuudet, kuten Tutki-hakuominaisuus sekä Kelat-videon luonti- ja jako-ominaisuus, käyttävät niille suunniteltuja

erillisiä algoritmeja. Näiden algoritmien toiminta perustuu sitä käyttävän ominaisuuden käyttötarkoitukseen. Instagramin suosittelualgoritmien toiminta vastaa paljon Facebookin algoritmia siten, että sen toiminta koostuu eri vaiheista, jotka se toteuttaa suositusten saamiseksi.

Muutamia poikkeuksia, kuten mainoksia, lukuun ottamatta Syöte- ja Tarinat-ominaisuuksien suosittelualgoritmit käyttävät seulonnassaan kaikkia viimeaikaisia julkaisuja, joita käyttäjän seuraamat käyttäjät ovat jakaneet. Tämä metodi vaikuttaa paljon yhteistoiminnallisen suodattamisen käyttäjä-käyttäjäpohjaisen ja sisältöpohjaisen suosittelualgoritmin hybridiltä, jossa algoritmi näyttää julkaisuja, joista käyttäjän seuraamat profiilit ovat tykänneet. Instagram ei avaudu tarkemmin siitä, miten algoritmi seuloo ja valitsee eri poikkeuksia, kuten mainoksia, käyttäjän katseltavaksi. Facebookin algoritmin tavoin Instagramin Syöte- ja Tarinat-ominaisuuksien algoritmit käyttävät signaaleja, jotka sisältävät lisätietoa kerätyistä julkaisuista ja käyttäjän antamista syötteistä, esimerkiksi tykkäyksien määrän, tietoja julkaisijasta sekä käyttäjän aikaisemmista reaktioista samankaltaisiin tai saman julkaisijan luomiin julkaisuihin.

Tärkeimmät Syöte- ja Tarinat-ominaisuuksien signaalit järjestyksessä:

- julkaisuun liitetty tieto
- tietoja julkaisijasta
- käyttäjän aikaisemmat syötteet
- käyttäjän vuorovaikutushistoria muiden käyttäjien kanssa.

Näiden signaalien pohjalta algoritmi luo tusinan verran ennustuksia eri todennäköisyyksistä, kuten käyttäjän todennäköisyydestä viettää muutama sekunti julkaisun parissa, kommentoida, tykätä tai jakaa julkaisua sekä klikata julkaisijan profiilikuvaa. Algoritmi arvioi eri julkaisuille tuotettuja ennustuksia ottamalla huomioon myös muita tekijöitä, joita suosittelija pyrkii välttämään, esimerkiksi useamman saman käyttäjän tekemien julkaisujen näyttämistä kerrallaan. Mitä suurempi todennäköisyys, että käyttäjä ryhtyy jatkotoimiin julkaisun nähtyään, sitä ylemmäksi se sijoittuu käyttäjän näkymälle.

Instagramin Tutki-ominaisuuden suosittelualgoritmi seuraa samanlaisia vaiheita. Se poikkeaa Syöte- ja Tarinat-ominaisuuksien algoritmeista, koska se käyttää suositusten tuottamiseen kuvia ja videoita profiileista, joita käyttäjä ei seuraa. Eli mahdollisesti kyseessä on enemmän yhteistoiminnallisen suodattamisen tuote-tuotepohjainen algoritmimalli, jossa suosituksissa otetaan huomioon sisältö, jota käyttäjä ei ole todennäköisesti aikaisemmin nähnyt. Suosittelualgoritmi etsii käyttäjän aikaisemmin pitämien julkaisujen ja muiden julkaisujen välisiä samankaltaisuuksia. Algoritmi toimii samalla tavalla käymällä kerättyjen julkaisujen signaalit läpi priorisoiden käyttäjän syötteisiin liittyviä signaaleja, kuten julkaisuja, joista käyttäjä on tykännyt tai kommentoinut aikaisemmin.

Tärkeimmät Tutki-ominaisuuden signaalit järjestyksessä:

- julkaisun tiedot
- käyttäjän ja julkaisijan välinen vuorovaikutushistoria
- käyttäjän aikaisemmat syötteet
- tietoja julkaisijasta.

Tästä jatketaan julkaisujen todennäköisyyksien ennustamiseen, minkä jälkeen se sijoittaa ne Tutki-ominaisuuden näkymään käyttäjän arvioidun kiinnostuksen mukaan käyttäjälle selailtavaksi.

Kelat-ominaisuuden suosittelualgoritmin halutaan löytävän käyttäjälle pääasiallisesti häntä viihdyttävää sisältöä. Algoritmi suorittaa samankaltaisen prosessin, jota Syöte-, Tarinat- ja Tutki-ominaisuuksien algoritmit suorittavat ja valikoivat Kelat-videoita, joista käyttäjä voisi pitää. Tutki-ominaisuuden suosittelualgoritmin tavoin suurin osa valikoiduista ehdotuksista on käyttäjiltä, joita käyttäjä ei itse seuraa. Suosittelualgoritmi saattaa siis olla taas yhteistoiminnallisen suodattamisen tuote-tuotepohjaisen mallin kaltainen. Tässä tapauksessa algoritmi korostaa vertailussaan julkaisujen välisiä viihdytettävyyden samankaltaisuuksia. Videon viihdytettävyyden mittaamiseksi algoritmi käyttää saatuja palautteita, joissa käyttäjiltä on kysytty arvioita tiettyjen Kelat-julkaisujen viihdearvosta tai

hauskuudesta. Eli suosittelualgoritmissa on voitu hyödyntää tietopohjaista suosittelumallia. Palautteiden pohjalta algoritmi oppii seulomaan paremmin sisältöä ja valikoimaan mahdollisia ehdotuksia, jotka todennäköisemmin viihdyttävät käyttäjää.

Tärkeimmät Kelat-ominaisuuden signaalit järjestyksessä:

- käyttäjän aikaisemmat syötteet
- käyttäjän ja julkaisijan välinen vuorovaikutushistoria
- kelat-videojulkaisun tiedot
- tietoja julkaisijasta.

Ennustusten arvioinnissa Kelat-ominaisuuden algoritmi priorisoi kuinka todennäköisesti käyttäjä katsoo Kelat-videon kokonaan, tykkää videosta ja kertoo sen olevan viihdyttävä. Algoritmi ottaa huomioon myös käyttäjän aikaisempia syötteitä ja katseluhistoriaa suositusten seulonnassa. Kelat-ominaisuuden suosittelualgoritmissa on myös joitakin poikkeavia syitä, joita Instagram välttää suosituksissaan, kuten videon laatua, mahdollista vesileimaa, hiljennystä tai sisällön keskittymistä poliittisiin ongelmiin.

Instagramin taktiikka jakaa yhteisöpalvelunsa suosittelualgoritmit tarkempiin käyttötarkoituksiin näyttää toimineen tähän asti hyvin. Kun algoritmin ei tarvitse ottaa huomioon usean ominaisuuden erilaisia käyttötarkoituksia, joiden pohjalta se suosituksia tekee, myös riski huonoille, sekalaiselle ja tylsille suosituksille vähenee. Tällä Instagram todennäköisesti pyrkii hallinnoimaan paremmin eri ominaisuuksia ja parantamaan palvelunsa käyttökokemusta sekä todennäköisyyttä, että käyttäjä jatkaa palvelun käyttöä.

Instagramin algoritmit eivät kuitenkaan vaikuta kokonaan täydellisiltä. Viime vuosien aikana Instagramin suosittelualgoritmien on väitetty olevan rasistisia, seksistisiä, LGBTQIA+ ja alkuperäiskansavähemmistöjä sekä muita ryhmiä kuten plus-kokoisia ja taiteellisia käyttäjiä syrjiviä [28; 29; 30]. Tarkemmin ottaen algoritmien suodattaessa sisältöä ne samalla alentavat näihin ryhmiin liittyvien

ja kuuluvien henkilöiden tekemien julkaisujen näkyvyyttä sekä sensuroivat julkaisuja ja jopa poistavat käyttäjiä [31]. Tätä väitettä tukevat tapaukset, joissa algoritmit eivät kohtele samankaltaisia julkaisuja samalla tavalla [32].

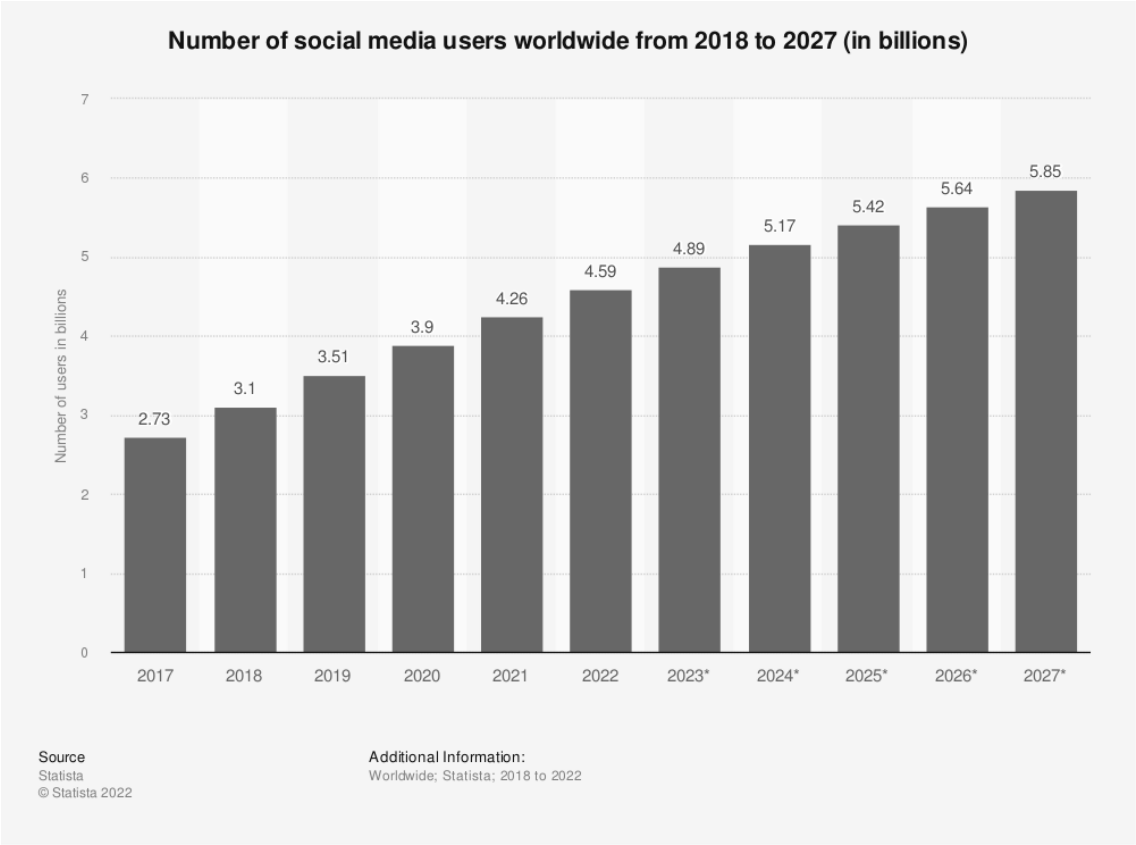
Instagram on julkisesti myöntänyt, että he eivät ole tehneet tarpeeksi töitä selittääkseen, miksi tiettyjä julkaisuja poistetaan tai millainen sisältö on suositeltavaa ja millainen ei. Instagram selittää työskentelevänsä ahkerasti algoritmiensa parantamiseksi ja niiden aiheuttamien virheiden vähentämiseksi. Instagram on myös sanonut pyrkivänsä estämään tuotteidensa alitajuntaisia ennakkoluuloja [33; 34]. He myös korostavat, miten käyttäjien tekemien julkaisujen sisältö saattaa vaikuttaa niiden kelpoisuuteen olla suositeltavia [35]. Samanaikaisesti he eivät ole kuitenkaan antaneet todisteita siitä, ettei heidän algoritmit ole ennakkoasenteisia, esimerkiksi he eivät ole julkaisseet signaaleja, joita algoritmit käyttävät ennustuksissa. Käyttäjät eivät tiedä, ovatko jotkin näistä signaaleista mahdollisesti sellaisia, jotka aiheuttaisivat algoritmin asenteellisuuden.

Instagramin suuren suosion takia he eivät ymmärrettävästi myöskään halua jakaa heidän kilpailijoita mahdollisesti hyödyttäviä tietoja. Tämän takia suurien yhteisöpalvelujen, kuten Instagramin, käyttämiä suosittelualgoritmeja on vaikea tutkia ja analysoida. Instagramin suosittelualgoritmit ovat hyvin monimutkaisia, ja ne todennäköisesti koostuvat erilaisista suosittelumalleista. Vaikka nämä algoritmit suosittelevat onnistuneesti sisältöä, jotkin niiden osista aiheuttavat edelleen ongelmia, joita Instagram pyrkii yhä korjaamaan. Kuinka paljon suosittelualgoritmeja täytyy suunnitella ja muuttaa, että samanlaisilta ongelmilta vältyttäisiin? Jatkuvien päivitysten ja uusien ominaisuuksien ohella on haasteellista ennakoida sekä seurata, miten ne vaikuttavat suosittelualgoritmien toimintaan.

3.4 Sosiaalinen media osana yhteisöpalvelua

Yhteisöpalvelut koostuvat kahden tai useamman käyttäjän välisestä vuorovaikutuksesta ja verkostoitumisesta. Sosiaalinen media koostuu verkkosivuista ja tietokoneohjelmista, joiden kautta voidaan kommunikoida ja jakaa tietoa internetiin

[36]. Tämä kommunikointi toteutetaan yleensä verkkosivuston tai puhelinsovelluksen kautta, jossa käyttäjät voivat jakaa tietoa muille [37, s. 10]. Sosiaalinen media vaatii käytettäväkseen yhteisöpalveluja, joiden kautta sisältöä voidaan jakaa. Yhteisöpalvelut kuten Instagram ja Facebook sisältävät monta erilaista tapaa, joilla niin yksityishenkilöt kuin yrityksetkin voivat julkaista ja jakaa sisältöä. Yhä useammalla henkilöllä on mahdollisuus käyttää yhteisöpalveluja kehittyneen teknologian ansiosta. Tämä näkyy jatkuvana sosiaalista mediaa kuluttavien käyttäjien kasvuna (ks. kuva 7). Monilla suosituimmilla sosiaalisen median alustoilla kuten Facebookilla, Youtubella ja Instagramilla on tiettävästi käytössä jonkinlainen suosittelualgoritmi. Katsotaan niiden avulla, mitä asioita kannattaa ottaa huomioon sosiaalisen median alustalle tarkoitetun suosittelualgoritmin suunnittelussa.



Kuva 7. Sosiaalisen median käyttäjien määrän arvioitu kehittyminen maailmanlaajuisesti vuosien 2018 ja 2027 välillä [38].

Yhteisöpalvelujen käyttämät suosittelualgoritmit määrittelevät, miten käyttäjien julkaisema media tulee muille käyttäjille näkyviin. Nykyään algoritmit mukautuvat automaattisesti käyttäjän antamien syötteiden pohjalta, millä ne pyrkivät parantamaan käyttökokemusta ja antamaan jokaiselle käyttäjälle mahdollisimman osuvia suosituksia. Sosiaalinen media voi sisältää erilaista mediaa kuten videoita, kuvia, tekstiä, sosiaalisia verkostoja ja tuotearvosteluja riippuen yhteisöpalvelusta. Tämä on hyvä pitää mielessä suosittelualgoritmia suunnitellessa.

Julkaistun median määrä voi myös kasvaa nopeasti ongelmaksi, minkä takia esimerkiksi Instagram otti käyttöön useamman suosittelualgoritmin. Täytyy siis miettiä, miten suosittelualgoritmi toimii alustan julkaisuvaiheessa sekä sen kehittyessä. Monet yhteisöpalvelut päivittävät algoritmejaan jatkuvasti pysyäkseen kilpailussa mukana ja parantaakseen käyttäjäkokemusta. Suosituimmat yhteisöpalvelut lisäävät jatkuvasti uusia ominaisuuksia, jotka voivat vaikuttaa suosittelualgoritmin toimintaan. Eli täytyy ottaa huomioon, kuinka usein suunniteltua sosiaalisen median alustaa tullaan päivittämään tai muuttamaan sen julkaisun jälkeen ja mitä mahdollisia vaikutuksia sillä saattaa olla suosittelualgoritmiin. Ihanteellista olisi, että suosittelualgoritmi on helposti mukautettavissa mahdollisiin muutoksiin.

Monet suosituimmista yhteisöpalveluista koostuvat käyttäjistä ja heidän tuottamistaan mediajulkaisuista. He ovat ottaneet tämän huomioon tekemällä algoritmeistaan sellaisia, että ne tuottavat käyttäjille personalisoituja suosituksia perustuen käyttäjäprofiileihin. Kun suosittelualgoritmi ottaa huomioon jokaisen käyttäjän henkilökohtaiset käyttötottumukset ja sisältömieltymykset, käyttäjillä on suurempi todennäköisyys käyttää alustaa jatkossakin.

Sosiaalisen median alustalle tarkoitettu suosittelualgoritmi

- pystyy analysoimaan erilaista sisältöä (kuvia, videoita ym.)
- pystyy mukautumaan muutoksiin
- pystyy tekemään personalisoituja suosituksia.

Pidetään nämä ominaisuudet mielessä, kun tutkimme seuraavaksi, mikä suosittelualgoritmi sopisi parhaiten sosiaalisen median alustalle.

4 Koeasetelma

4.1 Kysymyksenasettelu

Seuraavaksi suoritetaan suosittelualgoritmikoe, jossa vertaamme yksinkertaisten prototyyppien avulla muistipohjaisten yhteistoiminnallisien ja sisältöpohjaisen suosittelualgoritmien toimintaa. Näillä testeillä pyritään tutkimaan työssä läpikäytyjen suosittelualgoritmien soveltuvuutta sosiaalisen median yhteisöpalvelualustalle. Tavoitteena on saada parempi käsitys näiden suosittelualgoritmien toiminnasta ja luoda johtopäätöksiä siitä, kumpi algoritmi sopii parhaiten koeasetelman osana esittelemällemme sosiaalisen median yhteisöpalvelulle.

Luvuissa 3.1–3.3 käsittelimme, kuinka suosittelualgoritmit toimivat käytetyimmissä yhteisöpalveluissa. Tarkastelimme myös sosiaalisen median käyttöä yhteisöpalveluissa sekä miten suosittelualgoritmit tämänlaisessa alustassa pystyvät toimimaan. Käytetään tätä kerättyä tietoa perustana koeasetelmassa ja hyödynnetään sitä vertailussa. Toteutettu vertailu perustuu laadulliseen eli kvalitatiiviseen arviointiin. Tarkastellaan datasetistä valitun käyttäjän saamia suosituksia eri suosittelualgoritmimenetelmien avulla ja arvioidaan havaintojen pohjalta kunkin algoritmin soveltuvuutta sosiaalisen median alustojen suosittelutarpeisiin.

4.2 Työssä käytetty datasetti

Vertailussa käytettiin kirja-datasettiä [39]. Se koostui käyttäjien antamien arvostelujen ratings.csv-tiedostosta ja kirjojen books.csv-tiedostosta. Kirjoja oli yhteensä 10 000 ja käyttäjiä 53 424. Jokaiselle kirjalle oli annettu keskiarvolta 98

arvostelua ja käyttäjät ovat antaneet keskimäärin 18 kirja-arvostelua. Arvostelu-datassa oli listattu käyttäjien antamia arvosanoja kirjoille (ks. kuva 8) ja kirjada-tassa oli listattu kaikki kirjat ja niille annetut lisätiedot, kuten nimi, kirjailija ja jul-kaisuvuosi (ks. kuva 9).

	book_id	user_id	rating
0	1	314	5
1	1	439	3
2	1	588	5
3	1	1169	4
4	1	1185	4

Kuva 8. Datasetin ratings.csv-tiedoston sisältö.

	id	book_id	best_book_id	work_id	books_count	isbn	isbn13	authors
0	1	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins
1	2	3	3	4640799	491	439554934	9.780440e+12	J.K. Rowling, Mary GrandPré
2	3	41865	41865	3212258	226	316015849	9.780316e+12	Stephenie Meyer
3	4	2657	2657	3275794	487	61120081	9.780061e+12	Harper Lee
4	5	4671	4671	245494	1356	743273567	9.780743e+12	F. Scott Fitzgerald

Kuva 9. Datasetin books.csv-tiedoston sisältö.

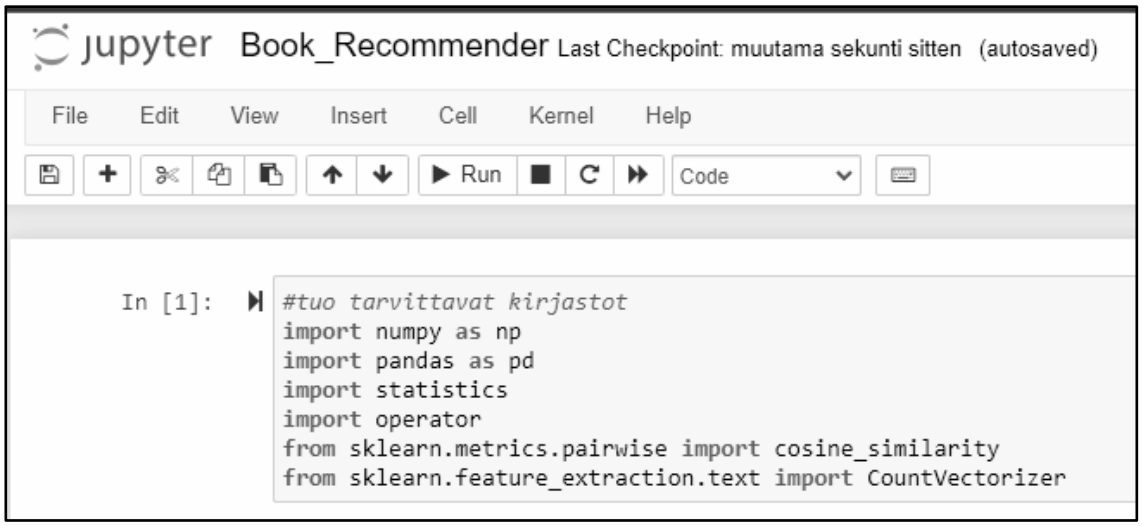
Työn alussa tehdyissä harjoitustesteissä sovellettiin peli-datasettiä, joka koostui peleistä ja niihin liitetyistä tiedoista. Tällä datasetillä pystyttiin kuitenkin testaamaan vain sisältöpohjaista suosittelua. Jotta datasetin kanssa pystyi testaamaan myös yhteistoiminnallista seulontaa, tarvittiin käyttäjiä. Tämän takia työssä päädyttiin lopulta käyttämään kirja-datasettiä, joka sisälsi sekä tuote- että käyttäjätietoja.

Työssä toteutettuja kirjasuositteluesimerkkejä voidaan kuitenkin jossain määrin soveltaa sosiaalisen median yhteisöpalvelussa tapahtuvaan suositteluun. Datasetissä olevat kirjat voitaisiin helposti korvata jollakin muulla tuotedatalla, jota yhteisöpalvelun suosittelualgoritmi ehdottaa käyttäjille, esimerkiksi mainoksia tai erilaisia mediajulkaisuja (video, kuva, teksti jne.). Yhteisöpalvelu voi suositella monenlaista mediaa käyttäjilleen. Parhaiten tähän kokeeseen olisi sopinut jonkinlainen multimedia-datasetti, koska monet sosiaalisen median yhteisöpalvelut käyttävät useampaa mediatyyppiä. Se olisi parantanut suosittelualgoritmien soveltuvuuden vertailua ja tehnyt havainnoista hyödyllisempiä. Nämä prototyypit ovat esimerkkejä yksittäisen tuotedatan suosittelusta.

4.3 Työvälineet ja prototyyppien toteutus

Prototyypit, joilla suosittelualgoritmeja testattiin, luotiin Jupyter Notebook -ohjelman avulla. Jupyter Notebook on palvelin-asiakasohjelma, jolla voidaan kirjoittaa, editoida ja suorittaa notebook-dokumentteja [40]. Dokumentit sisältävät muun muassa koodia, yhtälöitä sekä visualisointeja, joita Jupyter Notebook voi suorittaa selaimessa. Koodikielenä käytettiin Pythonia ja koodissa hyödynnettiin NumPy, Pandas ja Scikit-learn (Sklearn) Python-kirjastoja sekä statistics ja operator Python-moduuleja (ks. kuva 10). Suosittelualgoritmien rakentamisessa käytettiin sekä sovellettiin käyttäjien GreekDataGuy ja Mahnoor Javed tekemiä suosittelualgoritmiesimerkkejä [41; 42]. Jupyter Notebookin käynnistämiseksi ja tiedostojen hallitsemiseksi käytettiin myös Anaconda Navigator -pöytäko-

nesovellusta, jolla parannettiin työskentelyä. Testauksissa käytetty datasetti löydettiin Kaggle-sivustolta. Kaggle on Googlen omistama yhteisöpalvelu, jossa käyttäjät voivat löytää, ladata ja julkaista erilaisia datasettejä.



Kuva 10. Tuodaan tarvittavat kirjastot ja moduulit työympäristöön.

4.3.1 Yhteistoiminnalliset suosittelualgoritmiprototyypit

Aloitettiin suosittelualgoritmiprototyyppien tekeminen käyttäjäpohjaisesta yhteistoiminnallisesta suosittelualgoritmista. Koska kyseessä oli yhteistoiminnallinen suosittelualgoritmi, tarvittiin datasetti, joka sisälsi tuotteita ja käyttäjiä. Aluksi tuotiin käyttäjien arvosteluista koostuva RatingData-datasetti Jupyter Notebook -työympäristöön (ks. kuva 11).

```
In [2]: #YHTEISTOIMINNALLINEN käyttäjä-käyttäjä

        #tuo kirja-arvostelu datasetti
        book_ratings = pd.read_csv("RatingData.csv")
        book_ratings.head()
```

Out[2]:

	book_id	user_id	rating
0	1	314	5
1	1	439	3
2	1	588	5
3	1	1169	4
4	1	1185	4

Kuva 11. Tuodaan arvostelu-datasetti työympäristöön.

Seuraavaksi tutkittiin arvostelu-datasetin sisältöä. Datasetin pohjalta laskettiin käyttäjien arvostelemien kirjojen keskiarvo sekä kirjoille annettujen arvostelujen keskiarvo (ks. kuva 12).

```
In [3]: #arvosteltujen kirjojen määrän keskiarvo per käyttäjä
        ratings_per_user = book_ratings.groupby('user_id')['rating'].count()
        statistics.mean(ratings_per_user.tolist())

Out[3]: 18.376684636118597

In [4]: #annettujen arvostelujen määrän keskiarvo per kirja
        ratings_per_book = book_ratings.groupby('book_id')['rating'].count()
        statistics.mean(ratings_per_book.tolist())

Out[4]: 98.1756
```

Kuva 12. Lasketaan käyttäjien antamien arvostelujen ja kirjoille annettujen arvostelujen keskiarvollinen määrä.

Keskiarvojen pohjalta pienennettiin datasettiä poistamalla käyttäjiä ja kirjoja. Kirjat, joilla oli alle 50 arvostelua, ja käyttäjät, jotka ovat antaneet alle 10 arvostelua poistettiin (ks. kuva 13). Näin saatiin kohdennettua hyödyllisen datan määrää ja samalla pienennettiin seuraavissa vaiheissa tarvittavan muistin määrää, mikä myös vähentää Memory Error -virheen riskiä.

```
In [5]:  #arvostelujen määrä per kirja dataframe:nä
         ratings_per_book_df = pd.DataFrame(ratings_per_book)

         #poista kirja jos alle 50 arvostelua
         filtered_ratings_per_book_df = ratings_per_book_df[ratings_per_book_df.rating >= 50]

         #luo lista kirjoista, joilla on enemmän kuin 50 arvostelua
         popular_book = filtered_ratings_per_book_df.index.tolist()

In [6]:  #arvostelujen määrä per käyttäjä dataframe:nä
         ratings_per_user_df = pd.DataFrame(ratings_per_user)

         #poista  käyttäjä jos alle 10 annettua arvostelua
         filtered_ratings_per_user_df = ratings_per_user_df[ratings_per_user_df.rating >= 10]

         #luo lista käyttäjistä, joilla on enemmän kuin 10 annettua arvostelua
         prolific_users = filtered_ratings_per_user_df.index.tolist()

In [9]:  #seulo valitut kirjat ja käyttäjät
         filtered_ratings = book_ratings[book_ratings.book_id.isin(popular_book)]
         filtered_ratings = book_ratings[book_ratings.user_id.isin(prolific_users)]
         len(filtered_ratings)

Out[9]: 857538
```

Kuva 13. Poistetaan kirjat, joilla on alle 50 arvostelua ja käyttäjät, joilla on alle 10 annettua arvostelua.

Seuraavaksi luotiin käyttäjä-kirjamatriisi järjestämällä BookData-datasetin arvot uudelleen (ks. kuva 14). Täytettiin samalla tyhjät NaN-kohdat arvolla 0, koska kosinin samankaltaisuus ei toimi NaN-arvoilla. Matriisissa nolla-arvot tarkoittavat, että käyttäjät eivät ole antaneet kyseiselle kirjalle arvostelua.

```
In [8]:  #luo käyttäjä-kirja matriisi
         user_book_matrix = filtered_ratings.pivot_table(index=['user_id'], columns=['book_id'], values='rating')
         user_book_matrix = user_book_matrix.fillna(0)
         user_book_matrix
```

Out[8]:

book_id	1	2	3	4	5	6	7	8	9	10	...	9991	9992	9993	9994	9995	9996	9997	9998	9999	10000
user_id																					
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
53409	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53411	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53413	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53422	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53424	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

24405 rows × 10000 columns

Kuva 14. Luodaan käyttäjä-kirjamatriisi ja korvataan mahdolliset NaN-arvot nolla-arvoilla.

Sitten tarkastettiin, että kirjoille annetut arvostelut näkyvät matriisissa (ks. kuva 15).

```
In [13]:  #katsotaan ketkä käyttäjät ovat arvostelleet kirjaa id:llä 1
          numbers = user_book_matrix.loc[user_book_matrix[1] != 0.0]
          numbers.head()
```

Out[13]:

book_id	1	2	3	4	5	6	7	8	9	10	...	9991	9992	9993	9994	9995	9996	9997	9998	9999	10000
user_id																					
314	5.0	0.0	3.0	0.0	4.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
439	3.0	0.0	0.0	5.0	0.0	0.0	3.0	0.0	3.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
588	5.0	0.0	1.0	0.0	0.0	0.0	0.0	3.0	3.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1169	4.0	3.0	0.0	5.0	5.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1185	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 10000 columns

Kuva 15. Katsotaan matriisista arvosteluja kirjalle, jonka id-arvo on 1.

Käyttäjä-käyttäjäpohjaisessa suosittelussa suositukset luodaan muiden samankaltaisten käyttäjien pohjalta. Kirjojen suosittelua varten luotiin metodi, joka etsii 10 samankaltaista käyttäjää hyödyntämällä kosinin samankaltaisuuden mittaa (ks. kuva 16).

```
In [14]:
#luo metodi samankaltaisten käyttäjien löytämiseksi
def similar_users(user_id, matrix, k=10):
    #luo uusi dataframe valitulle käyttäjälle
    user = matrix[matrix.index == user_id]

    #luo toinen dataframe muille käyttäjälle
    other_users = matrix[matrix.index != user_id]

    #laske kosinin samankaltaisuus käyttäjän ja muiden käyttäjien välillä
    user_similarities = cosine_similarity(user,other_users)[0].tolist()

    #luo lista muiden käyttäjien indekseistä
    indices = other_users.index.tolist()

    #luo avain/arvo parit muiden käyttäjien indekseistä ja niiden samankaltaisuudesta
    index_similarity = dict(zip(indices, user_similarities))

    #järjestä käyttäjät samankaltaisuuden mukaan
    index_similarity_sorted = sorted(index_similarity.items(), key=operator.itemgetter(1))
    index_similarity_sorted.reverse()

    #valitse 10 samankaltaisinta käyttäjää listan kärjestä
    top_users_similarities = index_similarity_sorted[:k]
    users = [u[0] for u in top_users_similarities]

    return users
```

Kuva 16. Luodaan metodi samankaltaisten käyttäjien löytämiseksi.

Valittiin seuraavaksi yksi käyttäjä, jolle teemme kaikilla prototyypeillä suosituksia. Tähän käyttäjään viitattiin hänelle määritellyllä user_id-arvolla, mikä oli 314. Sen jälkeen käytettiin aikaisemmin tehtyä metodia ja luotiin käyttäjälle 314 samankaltaisten käyttäjien lista (ks. kuva 17).

```
In [15]:
#valitse käyttäjä, jolle suosituksia tehdään
user = 314

#etsitään samankaltaisia käyttäjiä, kuin käyttäjä 314
found_similar_users = similar_users(user, user_book_matrix)

#katsotaan ketkä käyttäjät ovat kaikista samanlaisia
similar_users_to_random_user = pd.DataFrame(found_similar_users, columns=['similar_users'])
print(similar_users_to_random_user)
```

```
   similar_users
0          21228
1          24339
2          11927
3           2077
4          48482
5          24499
6          19171
7          49022
8          45269
9          51692
```

Kuva 17. Valitaan käyttäjä 314 ja luodaan hänen pohjalta 10 samankaltaisen käyttäjän lista.

Verrattiin saatuja samankaltaisia käyttäjiä ja heidän antamia arvosteluja (ks. kuva 18). Huomattiin, että käyttäjät ovat antaneet samoille kirjoille samanlaisia arvosteluja keskenään sekä käyttäjän 314 kanssa (vrt. kuvat 15 ja 18).

```
In [16]:  #tarkastetaan, että käyttäjillä on samanlaisia arvosteluja
          user_similarity_check = user_book_matrix.loc[[21228, 24339, 11927, 2077, 48482, 24499, 19171, 49022, 45269, 51692]]
          user_similarity_check
```

Out[16]:

book_id	1	2	3	4	5	6	7	8	9	10	...	9991	9992	9993	9994	9995	9996	9997	9998	9999	10000
user_id																					
21228	5.0	4.0	3.0	3.0	4.0	4.0	5.0	4.0	5.0	4.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24339	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11927	4.0	5.0	4.0	0.0	5.0	5.0	5.0	0.0	0.0	5.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2077	4.0	0.0	2.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
48482	3.0	5.0	3.0	3.0	0.0	0.0	3.0	0.0	3.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24499	5.0	5.0	0.0	0.0	3.0	0.0	2.0	3.0	2.0	4.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19171	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
49022	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
45269	4.0	4.0	0.0	5.0	4.0	0.0	4.0	5.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51692	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

10 rows × 10000 columns

Kuva 18. Katsotaan kymmenen samankaltaisen käyttäjän antamia arvosteluja ja tarkastetaan, ovatko ne samanlaisia keskenään.

Kirjojen suosittelemiseksi tehtiin metodi, joka loi 20 kirjasuosituksen listan (ks. kuva 19). Metodi vertaili käyttäjän 314 ja samankaltaisten käyttäjien tekemiä kirja-arvosteluja. Koska yhteistoiminnallinen suosittelualgoritmi käyttää seulonnassa käyttäjän aikaisempia syötteitä, metodi poistaa myös vertailusta kirjat, jotka käyttäjä 314 on jo lukenut.

```
In [17]:
#luo metodi kirja suosituksille (käyttäjä-käyttäjä)
def recommend_books_to_user_user_based(user_index, similar_user_indices, matrix, items=20):

    #lataa vektorit samanlaisille käyttäjille
    similar_users = matrix[matrix.index.isin(similar_user_indices)]

    #laske 3 samanlaisen käyttäjän arvostelujen keskiarvo
    similar_users = similar_users.mean(axis=0)

    #muuta dataframe:ksi
    similar_users_df = pd.DataFrame(similar_users, columns=['mean'])

    #lataa vektori käyttäjälle, jolle suositukset tehdään
    user_df = matrix[matrix.index == user_index]

    #transponoi, jotta dataframe on helpompi muokata
    user_df_transposed = user_df.transpose()

    #nimeä kolumni uudelleen 'rating'
    user_df_transposed.columns = ['rating']

    #poista rivit, joilla ei ole 0 arvoa niin saadaan kirjat, joita käyttäjä ei ole lukenut
    user_df_transposed = user_df_transposed[user_df_transposed['rating']==0]

    #luo lista kirjoista, joita käyttäjä ei ole lukenut
    books_unseen = user_df_transposed.index.tolist()

    #valitse samanlaisten käyttäjien arvostelujen keskiarvot kirjoille, joita käyttäjä ei ole lukenut
    similar_users_df_filtered = similar_users_df[similar_users_df.index.isin(books_unseen)]

    #järjestä dataframe
    similar_users_df_ordered = similar_users_df.sort_values(by=['mean'], ascending=False)

    #valitse 20 samankaltaisinta kirjaa listan kärjestä
    top_n_book = similar_users_df_ordered.head(items)
    top_recommended_books = top_n_book.index.tolist()

    return top_recommended_books
```

Kuva 19. Metodi käyttäjäpohjaiselle yhteistoiminnalliselle kirjojen suosittelulle.

Käytettiin käyttäjäpohjaista kirjasuosittelumetodia luomaan käyttäjälle 314 kirjasuosituksia (ks. kuva 20).

```
In [146]:
#luo kirja suosituksia käyttäjälle 314
created_book_recommendations = recommend_books_to_user_user_based(314, found_similar_users, user_book_matrix)
```

Kuva 20. Käyttäjälle 314 luodaan kirjasuosituksia.

Lopuksi tuotiin BookData-datasetti, josta saatiin suositeltujen kirjojen nimet. Tätä varten luotiin metodi, joka hakee kirjojen nimet kirja-datasetistä sille annettujen kirjojen id-arvojen perusteella. Viimeiseksi tulostettiin kaikkien kirjasuositusten nimet (ks. kuva 21).

In [19]:
```
#tuo kirja datasetti
books = pd.read_csv("BookData.csv")
books = books.filter(['id', 'book_id', 'original_title'])
books.head()
```

Out[19]:

	id	book_id	original_title
0	1	2767052	The Hunger Games
1	2	3	Harry Potter and the Philosopher's Stone
2	3	41865	Twilight
3	4	2657	To Kill a Mockingbird
4	5	4671	The Great Gatsby

In [21]:
```
#luo metodi kirjojen nimien noutamiseen
def fetch_book_titles(ids):
    titles = []
    for i in ids:
        titles.append(books['original_title'].loc[books.index[i]])
    return titles

#hae suositeltujen kirjojen nimet ja tulosta ne
get_recommendations = fetch_book_titles(created_book_recommendations)
print(f'Käyttäjä {user} saattaa pitää näistä kirjoista:', *get_recommendations, sep = "\n")
```

```
Käyttäjä 314 saattaa pitää näistä kirjoista:
Lord of the Flies
The Da Vinci Code
Harry Potter and the Order of the Phoenix
Dear John
nan
Harry Potter and the Prisoner of Azkaban
Harry Potter and the Chamber of Secrets
The Lovely Bones
Old Man's War
It
Eat, pray, love: one woman's search for everything across Italy, India and Indonesia
 The Fellowship of the Ring
Gone
Nineteen Eighty-Four
Great Expectations
Twilight
11/22/63
The Lord of the Rings
Stranger in a Strange Land
Frankenstein; or, The Modern Prometheus
```

Kuva 21. Tuodaan kirja-datasetti ja tehdään metodi, joka hakee kirjojen nimet, jotka lopuksi tulostetaan.

Nyt oli tehty käyttäjäpohjainen yhteistoiminnallinen kirjasuosittelija ja saatiin sillä jo muutamia kirjasuosituksia. Tehtiin seuraavaksi tuotepohjainen yhteistoiminnallinen kirjasuosittelija. Tässä suosittelijassa sovellettiin aikaisempaa käyttäjäpohjaisen suosittelijan rakennetta. Seurattiin samoja vaiheita kuin käyttäjäpohjaisessa versiossa (ks. kuvat 11, 12 ja 13), mutta tällä kertaa käyttäjä-kirjamatriisin sijaan tehtiin kirja-käyttäjämatriisi (ks. kuva 22).

In [22]:
```
#YHTEISTOIMINNALLINEN tuote-tuote

#luo kirja-käyttäjä matriisi
book_user_matrix = filtered_ratings.pivot_table(index=['book_id'], columns=['user_id'], values='rating')
book_user_matrix = book_user_matrix.fillna(0)
book_user_matrix
```

Out[22]:

user_id	7	10	19	22	23	24	25	27	35	36	...	53388	53389	53400	53401	53403	53409	53411	53413	53422	53424
book_id																					
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9996	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9997	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9998	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9999	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

10000 rows × 24405 columns

Kuva 22. Luodaan kirja-käyttäjämatriisi ja korvataan NaN-arvot nolla-arvoilla.

Käytettiin seuraavaksi aikaisemmin samankaltaisten käyttäjien etsimiseen tehtyä metodia (ks. kuva 16), mutta nyt muutamme metodin etsimään datasetistä käyttäjien sijaan samankaltaisia kirjoja kosinin samankaltaisuuden avulla (ks. kuva 23).

In [23]:
```
#sovelletaan aiemmin luotua metodia samankaltaisten kirjojen löytämiseen
def similar_books(book_id, matrix, k=20):
    #luo uusi dataframe valitulle kirjalle
    book = matrix[matrix.index == book_id]

    #luo toinen dataframe muille kirjoille
    other_books = matrix[matrix.index != book_id]

    #laske kosinin samankaltaisuus kirjan ja muiden kirjojen välillä
    book_similarities = cosine_similarity(book,other_books)[0].tolist()

    #luo lista muiden kirjojen indekseistä
    indices = other_books.index.tolist()

    #luo avain/arvo parit muiden kirjojen indekseistä ja niiden samankaltaisuudesta
    index_similarity = dict(zip(indices, book_similarities))

    #järjestä kirjat samankaltaisuuden mukaan
    index_similarity_sorted = sorted(index_similarity.items(), key=operator.itemgetter(1))
    index_similarity_sorted.reverse()

    #valitse 20 samankaltaisinta kirjaa listan kärjestä
    top_books_similarities = index_similarity_sorted[:k]
    most_similar_books_user_based = [u[0] for u in top_books_similarities]

    return most_similar_books_user_based
```

Kuva 23. Tehdään metodi samankaltaisten kirjojen hakemiseen.

Valittiin sama käyttäjä kuin käyttäjäpohjaisessa versiossa eli käyttäjä 314. Katsottiin kyseisen käyttäjän parhaiten arvostelemia kirjoja eli kirjoja, joista käyttäjä on pitänyt (ks. kuva 24). Näistä kirjoista valittiin yksi, jonka pohjalta haluttiin etsiä samankaltaisia kirjoja käyttäjälle suositeltavaksi.

```
In [25]:  #valitse käyttäjä, jolle suosituksia tehdään
          user = 314

          #valitse yksi käyttäjän parhaiten arvostelemista kirjoista
          users_favourites = pd.DataFrame(book_user_matrix[user].dropna(axis=0, how='all')\
                                                  .sort_values(ascending=False)\
                                                  .reset_index()\
                                                  .rename(columns={1:'rating'}))

          users_favourites.head(10)
```

Out[25]:

	book_id	314
0	1	5.0
1	2673	5.0
2	141	5.0
3	6069	5.0
4	190	5.0
5	215	5.0
6	267	5.0
7	279	5.0
8	31	5.0
9	3220	5.0

Kuva 24. Valitaan sama käyttäjä, jolle suosituksia halutaan tehdä ja katsotaan käyttäjän parhaiten arvostelemia kirjoja.

Otettiin käyttäjän 314 pitämistä kirjoista kirja, jonka book_id-arvo on 1. Käyttäjä oli antanut tälle kirjalle arvosteluksi 5 eli käyttäjä on pitänyt siitä todennäköisesti paljon. Käytettiin aikaisemmin luotua metodia (ks. kuva 23), jolla tuotettiin kirjan 1 pohjalta 20 samankaltaisen kirjan lista (ks. kuva 25). Huomioidaan, että suosittelualgoritmit voivat käyttää useampaa kirjaa suosituksen tuottamisessa, mutta tässä testissä keskitymme yksittäisten kirjojen pohjalta tuotettuihin samankaltaisten kirjojen suositteluihin.

```
In [13]:  #valitaan kirja, jolle etsitään samanlaisia kirjoja
          book = 1
          found_similar_books = similar_books(book, book_user_matrix)
          similar_books_to_random_book = pd.DataFrame(found_similar_books, columns=['similar_books'])
          print(similar_books_to_random_book)

              similar_books
          0              17
          1              31
          2               2
          3              20
          4               3
          5              93
          6               5
          7              16
          8               9
          9              37
          10             36
          11              4
          12             45
          13             46
          14             27
          15             11
          16             22
          17             33
          18             57
          19             15
```

Kuva 25. Luodaan lista 20 kirjasta, jotka ovat samankaltaisia kirja 1:n kanssa.

Jotta pystyttiin tarkastelemaan saatuja samankaltaisia kirjoja, käytettiin uudelleen BookData-datasettiä ja luotiin uusi metodi yksittäisten kirjan nimien hakemiseen kirja-datasetistä (ks. kuva 26). Useamman kirjan nimen hakemiseen käytettiin samaa metodia, jota käytettiin käyttäjäpohjaisessa versiossa.

```
In [29]:  #tuo kirja datasetti
          books = pd.read_csv("BookData.csv")
          books = books.filter(['id', 'book_id', 'original_title'])
          books.head()
```

Out[29]:

	id	book_id	original_title
0	1	2767052	The Hunger Games
1	2	3	Harry Potter and the Philosopher's Stone
2	3	41865	Twilight
3	4	2657	To Kill a Mockingbird
4	5	4671	The Great Gatsby

```
In [30]:  #Luo metodit yksittäisen kirjan ja usean kirjan nimikkeiden hakemiseen
          def fetch_book_title(book_id):
              title = books['original_title'].loc[books.index[book_id]]
              return title

          def fetch_book_titles(ids):
              titles = []
              for i in ids:
                  titles.append(books['original_title'].loc[books.index[i]])
              return titles
```

Kuva 26. Tuodaan kirja-datasetti ja tehdään metodit yksittäisten sekä usean kirjan nimen hakemiseen datasetistä.

Käytettiin kirjojen nimien hakemiseen tehtyjä metodeja ja tulostettiin lista samankaltaisista kirjoista kuin kirja 1 (ks. kuva 27). Huomattiin, että kirja 1 on Harry Potter and the Philosopher’s Stone (Harry Potter ja viisasten kivi). Algoritmin nähtiin valitsevan ehdotuksiin muiden Harry Potter -kirjojen lisäksi muita seikkailu- ja fantasiakirjoja, kuten The Lion the Witch and the Wardrobe sekä The Adventures of Huckleberry Finn. Se ehdotti myös toisia nuortenkirjoja, esimerkiksi Twilight, The Fault in Our Stars, Catching Fire ja Divergent. Mutta myös erilaisempia kirjoja suositeltiin kirjaan 1 verrattuna, kuten Of Mice and Men, To Kill a Mockingbird sekä The Book Thief.

```
In [17]:  #katsotaan, mitkä kirjat ovat samankaltaisia
          book_title = fetch_book_title(book)
          get_similar_books = fetch_book_titles(found_similar_books)
          print(f'Samankaltaisia kirjoja kuin {book_title} :', *get_similar_books, sep = "\n")

          Samankaltaisia kirjoja kuin Harry Potter and the Philosopher's Stone :
          Harry Potter and the Prisoner of Azkaban
          Of Mice and Men
          Twilight
          Harry Potter and the Order of the Phoenix
          To Kill a Mockingbird
          Cien años de soledad
          The Fault in Our Stars
          Catching Fire
          Pride and Prejudice
          The Time Traveler's Wife
          The Lion, the Witch and the Wardrobe
          The Great Gatsby
          Water for Elephants
          The Book Thief
          Lord of the Flies
          Divergent
          Harry Potter and the Chamber of Secrets
          Fifty Shades of Grey
          The Adventures of Huckleberry Finn
          Män som hatar kvinnor
```

Kuva 27. Tulostetaan lista kirjoista, jotka ovat kosinin samankaltaisuuden mukaan samankaltaisia kirjoja Harry Potter and the Philosopher’s Stone -kirjan kanssa.

Seuraavaksi luotiin uusi metodi tuotepohjaiselle kirjojen suosittelijalle (ks. kuva 28). Metodissa hyödynnettiin aikaisemmin tehtyä käyttäjäpohjaista kirjasuosittelijametodia, jolla tehtiin 20 kirjan suosittelu lista.

```
In [35]:
#luo metodi kirja suosituksille (tuote-tuote)
def recommend_books_to_user_item_based(user_index, similar_book_indices, matrix, items=20):

     #lataa vektorit samanlaisille kirjoille
    similar_books = matrix[matrix.index.isin(similar_book_indices)]

    #muuta dataframe:ksi
    similar_books_df = pd.DataFrame(similar_books, columns=['mean'])

    #lataa vektori käyttäjälle, jolle suositukset tehdään
    user_df = matrix[matrix.index == user_index]

    #transponoi, jotta dataframe on helpompi muokata
    user_df_transposed = user_df.transpose()

    #nimeä kolumni uudelleen 'rating'
    user_df_transposed.columns = ['rating']

    #poista rivit, joilla ei ole 0 arvoa niin saadaan kirjat, joita käyttäjä ei ole lukenut
    user_df_transposed = user_df_transposed[user_df_transposed['rating']==0]

    #luo lista kirjoista, joita käyttäjä ei ole lukenut
    books_unread = user_df_transposed.index.tolist()

    #valitse samanlaisten kirjojen arvostelujen keskiarvot kirjoille, joita käyttäjä ei ole lukenut
    similar_books_df_filtered = similar_books_df[similar_books_df.index.isin(books_unread)]

    #järjestä dataframe
    similar_books_df_ordered = similar_books_df.sort_values(by=['mean'], ascending=False)

    #valitse 20 samankaltaisinta kirjaa listan kärjestä
    top_n_book = similar_books_df_ordered.head(items)
    top_similar_books_item_based = top_n_book.index.tolist()

    return top_similar_books_item_based
```

Kuva 28. Tehdään tuotepohjainen kirjasuosittelijametodi.

Käytettiin uutta tuotepohjaista kirjasuosittelumetodia ja tehtiin käyttäjälle 314 kirjasuosituksia sen avulla (ks. kuva 29).

```
In [37]:
#luo kirja suosituksia, käyttäjälle 314
created_book_recommendations_item_based = recommend_books_to_user_item_based(314, found_similar_books, book_user_matrix)
```

Kuva 29. Tehdään tuotepohjaisia kirjasuosituksia käyttäjälle 314.

Lopuksi saadut kirjasuositukset tulostettiin käyttämällä aikaisemmin tehtyä metodia, jolla haettiin kirjojen nimet kirja-datasetistä (ks. kuva 30).

```
In [24]:  #hae suositeltujen kirjojen nimet ja tulosta ne
          get_recommendations_item_based = fetch_book_titles(created_book_recommendations_item_based)
          print(f'Käyttäjä {user} saattaa pitää näistä kirjoista:', *get_recommendations_item_based, sep = "\n")

          Käyttäjä 314 saattaa pitää näistä kirjoista:
          Twilight
          To Kill a Mockingbird
          The Great Gatsby
          The Fault in Our Stars
          Pride and Prejudice
          Divergent
          Män som hatar kvinnor
          Catching Fire
          Harry Potter and the Prisoner of Azkaban
          Harry Potter and the Order of the Phoenix
          Harry Potter and the Chamber of Secrets
          Lord of the Flies
          Of Mice and Men
          Fifty Shades of Grey
          The Lion, the Witch and the Wardrobe
          The Time Traveler's Wife
          Water for Elephants
          The Book Thief
          The Adventures of Huckleberry Finn
          Cien años de soledad
```

Kuva 30. Tulostetaan tuote-tuotepohjaisen yhteistoiminnallisen suosittelijan antamat suositukset käyttäjälle 314.

4.3.2 Sisältöpohjainen suosittelualgoritmiprototyyppi

Sisältöpohjaisessa suosittelualgoritmissa suositukset tehdään kirjoille ja käyttäjälle annettujen määritelmien mukaan. Käytetyssä datasetissä käyttäjille ei ollut määritelty tarkempia tietoja, kuten ikää tai sukupuolta, joten käytettiin tässä suosittelussa vain kirjoille annettuja määritelmiä. Aloitettiin tuomalla BookData-datasetti työympäristöön ja seulottiin taulukosta sarakkeet, jotka haluamme suosittelijan ottavan huomioon suosituksia tehdessä (ks. kuva 31). Valittiin kirjoille määritellyt kirjailijat, julkaisuvuosi ja nimi. Näitä ominaisuuksia voi olla vähemmän tai enemmän riippuen siitä, mitä halutaan suosittelijan ottavan suosituksissaan huomioon.

```
In [40]:  #SISÄLTÖPOHJAINEN

          #avaa kirja datasetti
          all_books = pd.read_csv("BookData.csv")
          all_books = all_books.filter(['book_id', 'authors', 'original_publication_year', 'original_title'])
          all_books
```

Out[40]:

	book_id	authors	original_publication_year	original_title
0	2767052	Suzanne Collins	2008.0	The Hunger Games
1	3	J.K. Rowling, Mary GrandPré	1997.0	Harry Potter and the Philosopher's Stone
2	41865	Stephenie Meyer	2005.0	Twilight
3	2657	Harper Lee	1960.0	To Kill a Mockingbird
4	4671	F. Scott Fitzgerald	1925.0	The Great Gatsby
...	...	...	...	...
9995	7130616	Ilona Andrews	2010.0	Bayou Moon
9996	208324	Robert A. Caro	1990.0	Means of Ascent
9997	77431	Patrick O'Brian	1977.0	The Mauritius Command
9998	8565083	Peggy Orenstein	2011.0	Cinderella Ate My Daughter: Dispatches from th...
9999	8914	John Keegan	1998.0	The First World War

10000 rows × 4 columns

Kuva 31. Tuodaan kirjadata työympäristöön ja pidetään kirjailija-, julkaisuvuosi- ja nimi-sarakkeet.

Kirjailija, julkaisuvuosi ja nimi ovat ominaisuuksia, joita haluamme algoritmin käyttävän. Nämä ominaisuudet valittiin datasetistä ja täytettiin mahdolliset tyhjät NaN-kohdat, jotta emme saisi mitään virhearvoja (ks. kuva 32).

```
In [41]:  #valitse ominaisuudet ja täytetään tyhjät kohdat
          features = ['authors', 'original_publication_year', 'original_title']

          for feature in features:
              all_books[feature] = all_books[feature].fillna('')
```

Kuva 32. Valitaan halutut ominaisuudet ja täytetään mahdolliset NaN-arvot välilyönnillä.

Tehtiin seuraavaksi valituista ominaisuuksista oma sarake, jossa rivien sisältämä ominaisuusarvo on yhdistetty yhdeksi sanajanaksi (ks. kuva 33). Samalla muutettiin jokaisen ominaisuuden arvo float-tyypistä string-tyypiksi. Liitettiin uusi sarake vielä datasetin loppuun.

```
In [42]:
#yhdistä kaikki valitut ominaisuudet omaan kolumniinsa
def combined_features(row):
    #muutetaan samalla float arvot string arvoiksi
    return str(row['authors'])+" "+str(row['original_publication_year'])+" "+str(row['original_title'])

all_books["combined_features"] = all_books.apply(combined_features, axis =1)
all_books.head()
```

Out[42]:

	book_id	authors	original_publication_year	original_title	combined_features
0	2767052	Suzanne Collins	2008.0	The Hunger Games	Suzanne Collins 2008.0 The Hunger Games
1	3	J.K. Rowling, Mary GrandPré	1997.0	Harry Potter and the Philosopher's Stone	J.K. Rowling, Mary GrandPré 1997.0 Harry Potte...
2	41865	Stephenie Meyer	2005.0	Twilight	Stephenie Meyer 2005.0 Twilight
3	2657	Harper Lee	1960.0	To Kill a Mockingbird	Harper Lee 1960.0 To Kill a Mockingbird
4	4671	F. Scott Fitzgerald	1925.0	The Great Gatsby	F. Scott Fitzgerald 1925.0 The Great Gatsby

Kuva 33. Liitetään valitut ominaisuudet uudeksi kolumniksi datasetin loppuun.

Käytettiin Scikit-kirjaston CountVectorizer-moduulia, jolla pystyttiin muuttamaan uudessa ominaisuussarakkeessa oleva data lukumatriisiksi (ks. kuva 34). Tämä muuttaa datan muotoon, jota samankaltaisuuden mittoja käyttävät algoritmit pystyvät ymmärtämään ja käyttämään.

```
In [43]:
#muuta yhdistetyt ominaisuudet tekstit luku-matriisiksi
count_vectorizer = CountVectorizer()
count_matrix = count_vectorizer.fit_transform(all_books["combined_features"])
print("Luku matriisi:", count_matrix.toarray())

Luku matriisi: [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

Kuva 34. Tuodaan työympäristöön CountVectorizer-moduuli, jonka avulla luodaan ominaisuus-sarakkeessa olevasta datasta lukumatriisi.

Nyt meillä oli matriisi, jota kosinin samankaltaisuuden algoritmi pystyi lukemaan. Seuraavaksi käytettiin lukumatriisia kirjojen välisten kosinin samankaltaisuuksien laskemiseen. Sen jälkeen valittiin sama kirja kuin tuotepohjaisessa yhteistoiminnallisessa suosittelijassa (ks. kuva 35). Käytettiin kosinin samankaltaisuuden laskussa saatua tulosta samankaltaisten kirjojen kuin Harry Potter and the Philosopher’s Stone -kirjan seulomiseen. Mitä lähempänä numero on nolla-arvoa eli 90°, sitä pienempi kirjojen samankaltaisuus on, ja jos se on lähellä arvoa 1 eli 0°, kirjat ovat enemmän samankaltaisia keskenään.

```
In [44]:  #laske kosinin samankaltaisuus
          book_similarity = cosine_similarity(count_matrix)
          book_similarity

Out[44]: array([[1.        , 0.12909944, 0.        , ..., 0.16666667, 0.18731716,
                 0.15430335],
                [0.12909944, 1.        , 0.        , ..., 0.12909944, 0.14509525,
                 0.11952286],
                [0.        , 0.        , 1.        , ..., 0.        , 0.        ,
                 0.        ],
                ...,
                [0.16666667, 0.12909944, 0.        , ..., 1.        , 0.18731716,
                 0.15430335],
                [0.18731716, 0.14509525, 0.        , ..., 0.18731716, 1.        ,
                 0.17342199],
                [0.15430335, 0.11952286, 0.        , ..., 0.15430335, 0.17342199,
                 1.        ]])

In [45]:  #etsi samanlaisia kirjoja kuin kirja 1
          book_id = 1
          similar_books = list(enumerate(book_similarity[book_id]))
```

Kuva 35. Lasketaan lukumatriisin kosinin samankaltaisuus ja käytetään sitä luomaan kirja 1:n pohjalta suosituksia.

Valittiin saaduista samankaltaisista kirjoista 20 kaikista samankaltaisinta kirjaa ja tulostetaan ne listana (ks. kuva 36).

```
In [46]:  #järjestä löydetyt 20 samankaltaisinta kirjaa
          sorted_similar_books = sorted(similar_books, key=lambda x:x[1], reverse=True)
          top_similar_books = sorted_similar_books[:20]
          most_similar_books_content_based = [u[0] for u in top_similar_books]
          most_similar_books_content_based

Out[46]: [1,
          24,
          20,
          22,
          23,
          26,
          17,
          2100,
          3752,
          3274,
          278,
          8368,
          421,
          4296,
          8664,
          36,
          3053,
          8622,
          963,
          1465]
```

Kuva 36. Laitetaan samankaltaiset kirjat listaksi ja valitaan ensimmäiset 20 kirjasuositusta tulostettavaksi.

Lopuksi käytettiin aikaisemmissa prototyypeissä tehtyä, kirjojen nimien hakemiseen tarkoitettua, metodia. Haettiin suositeltujen kirjojen nimet samasta BookData-kirjadatasta ja tulostettiin kirjojen nimet (ks. kuva 37).

```
In [47]:  #hae suositeltujen kirjojen nimet ja tulosta ne
          get_book_title = fetch_book_title(book_id)
          get_similar_books_content = fetch_book_titles(most_similar_books_content_based)
          print(f'Samankaltaisia kirjoja kuin {get_book_title} :', *get_similar_books_content , sep = "\n")

          Samankaltaisia kirjoja kuin Harry Potter and the Philosopher's Stone :
          Harry Potter and the Philosopher's Stone
          Harry Potter and the Deathly Hallows
          Harry Potter and the Order of the Phoenix
          Harry Potter and the Chamber of Secrets
          Harry Potter and the Goblet of Fire
          Harry Potter and the Half-Blood Prince
          Harry Potter and the Prisoner of Azkaban
          Harry Potter Boxed Set Books 1-4
          Harry Potter Collection (Harry Potter, #1-6)
          nan
          Harry Potter and the Cursed Child, Parts One and Two
          Harry, A History: The True Story of a Boy Wizard, His Fans, and Life Inside the Harry Potter Phenomenon
          Complete Harry Potter Boxed Set
          The Chronicles of Narnia: The Lion, the Witch and The Wardrobe (Sheet Music)
          The Country Mouse and the City Mouse, The Dog and His Bone, The Fox and the Crow (A Little Golden Book)
          The Lion, the Witch and the Wardrobe
          Harry Potter and the Chamber of Secrets: Sheet Music for Flute with C.D
          Billions and Billions: Thoughts on Life and Death at the Brink of the Millennium
          The Hobbit and The Lord of the Rings
          The Agony and the Ecstasy: A Biographical Novel of Michelangelo
```

Kuva 37. Tulostetaan sisältöpohjaisen suosittelijan luomat kirjasuositukset.

Tulostettiin saadut kirjasuositukset vielä BookData-datasetin taulukossa, jotta voidaan tarkastella niiden ominaisuuksien samankaltaisuuksia (ks. kuva 38). Huomattiin, että suosittelija painotti seulonnassa kirjailijoiden ja kirjan nimiä. Tämä johtui todennäköisesti siitä, että näiden suositeltujen kirjojen sisältämät määritelmät ovat eniten samankaltaisia kirjalle 1 määriteltyjen ominaisuuksien kanssa.

```
In [48]:  #tulosta suositukset dataframe:ssä
          recommendations_content_based = all_books.loc[[1, 24, 20, 22, 23, 26, 17, 2100, 3752, 3274, 278, 8368, 421, 4296, 8664, 36,
          recommendations_content_based
```

Out[48]:

	book_id	authors	original_publication_year	original_title	combined_features
1	3	J.K. Rowling, Mary GrandPré	1997.0	Harry Potter and the Philosopher's Stone	J.K. Rowling, Mary GrandPré 1997.0 Harry Potte...
24	136251	J.K. Rowling, Mary GrandPré	2007.0	Harry Potter and the Deathly Hallows	J.K. Rowling, Mary GrandPré 2007.0 Harry Potte...
20	2	J.K. Rowling, Mary GrandPré	2003.0	Harry Potter and the Order of the Phoenix	J.K. Rowling, Mary GrandPré 2003.0 Harry Potte...
22	15881	J.K. Rowling, Mary GrandPré	1998.0	Harry Potter and the Chamber of Secrets	J.K. Rowling, Mary GrandPré 1998.0 Harry Potte...
23	6	J.K. Rowling, Mary GrandPré	2000.0	Harry Potter and the Goblet of Fire	J.K. Rowling, Mary GrandPré 2000.0 Harry Potte...
26	1	J.K. Rowling, Mary GrandPré	2005.0	Harry Potter and the Half-Blood Prince	J.K. Rowling, Mary GrandPré 2005.0 Harry Potte...
17	5	J.K. Rowling, Mary GrandPré, Rufus Beck	1999.0	Harry Potter and the Prisoner of Azkaban	J.K. Rowling, Mary GrandPré, Rufus Beck 1999.0...
2100	99298	J.K. Rowling, Mary GrandPré	1999.0	Harry Potter Boxed Set Books 1-4	J.K. Rowling, Mary GrandPré 1999.0 Harry Potte...
3752	10	J.K. Rowling	2005.0	Harry Potter Collection (Harry Potter, #1-6)	J.K. Rowling 2005.0 Harry Potter Collection (H...
3274	8	J.K. Rowling, Mary GrandPré	2003.0		J.K. Rowling, Mary GrandPré 2003.0
278	29056083	John Tiffany, Jack Thorne, J.K. Rowling	2016.0	Harry Potter and the Cursed Child, Parts One a...	John Tiffany, Jack Thorne, J.K. Rowling 2016.0...
8368	3130430	Melissa Anelli, J.K. Rowling	2008.0	Harry, A History: The True Story of a Boy Wiza...	Melissa Anelli, J.K. Rowling 2008.0 Harry, A H...
421	862041	J.K. Rowling	1998.0	Complete Harry Potter Boxed Set	J.K. Rowling 1998.0 Complete Harry Potter Boxe...
4296	93000	Harry Gregson-Williams	2006.0	The Chronicles of Narnia: The Lion, the Witch ...	Harry Gregson-Williams 2006.0 The Chronicles o...
8664	1034605	Patricia M. Scarry, Richard Scarry	1961.0	The Country Mouse and the City Mouse, The Dog ...	Patricia M. Scarry, Richard Scarry 1961.0 The ...
36	100915	C.S. Lewis	1950.0	The Lion, the Witch and the Wardrobe	C.S. Lewis 1950.0 The Lion, the Witch and the ...
3053	2223324	John Williams	2003.0	Harry Potter and the Chamber of Secrets: Sheet	John Williams 2003.0 Harry Potter and the Ch...

Kuva 38. Tulostetaan saadut kirjasuositukset kirjataulukkona.

5 Suosittelualgoritmien arviointi

5.1 Vaatimukset

Vaatimuksena oli luoda kolme prototyyppiä muistipohjaisista yhteistoiminnallisista suosittelualgoritmeista ja sisältöpohjaisesta suosittelualgoritmista. Muistipohjaisissa suosittelijoissa käytettiin samaa käyttäjää, jonka user_id-arvo on 314. Tuotepohjaisessa suosittelijassa ja sisältöpohjaisessa suosittelijassa käytettiin samaa Harry Potter and the Philosopher’s Stone -kirjaa, jonka pohjalta suosituksia tehtiin. Tämä kirja oli yksi valitun käyttäjän parhaiten arvostelemista kirjoista. Käyttäjä-käyttäjäpohjaisessa yhteistoiminnallisessa suosittelussa katselтiin käyttäjien samankaltaisuuksia ja tuote-tuotepohjaisessa taas tuotteiden välisiä samankaltaisuuksia. Molemmissa otettiin huomioon muiden käyttäjien antamia aikaisempia syötteitä eli kirjoille annettuja arvosteluja. Sisältöpohjaisessa suosittelussa suosittelija käytti samaa kirjaa kuin tuote-tuotepohjaisessa

yhteistoiminnallisessa suosittelijassa ja teki sille kirjoille määriteltyjen kirjailija-, julkaisuvuosi- ja nimi-arvojen pohjalta suosituksia. Kaikki prototyypit käyttivät samaa samankaltaisuuden mittaa eli kosinin samankaltaisuutta, jolla luotiin jokaisella prototyypillä 20 kirjasuosituksen lista.

5.2 Havainnot ja prototyyppien antamat suositukset

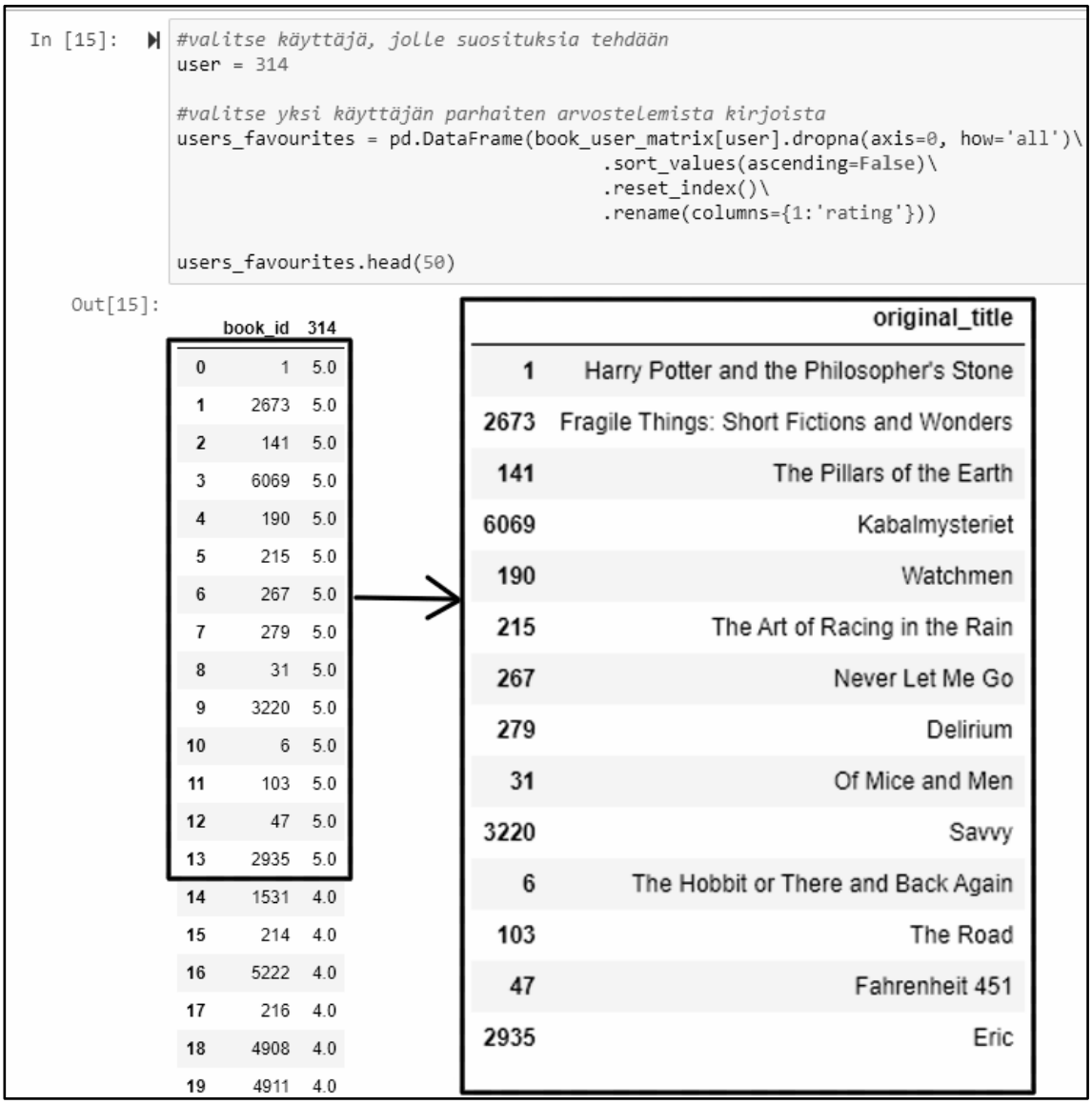
Tehdyistä prototyypeistä saimme siis kolme erilaista 20 kirjan suositusta. Tarkastellaan ensiksi muistipohjaisen käyttäjä-käyttäjäyhteistoiminnallisen suosittelijaprototyypin luomia suosituksia (ks. kuva 39).

```
Käyttäjä 314 saattaa pitää näistä kirjoista:
Lord of the Flies
The Da Vinci Code
Harry Potter and the Order of the Phoenix
Dear John
nan
Harry Potter and the Prisoner of Azkaban
Harry Potter and the Chamber of Secrets
The Lovely Bones
Old Man's War
It
Eat, pray, love: one woman's search for everything across Italy, India and Indonesia
 The Fellowship of the Ring
Gone
Nineteen Eighty-Four
Great Expectations
Twilight
11/22/63
The Lord of the Rings
Stranger in a Strange Land
Frankenstein; or, The Modern Prometheus
```

Kuva 39. Käyttäjä-käyttäjäpohjaisen yhteistoiminnallisen seulonnan tuottamat suositukset.

Käyttäjä-käyttäjäpohjaisen suosittelijan tekemät suositukset pohjautuvat muiden, käyttäjän 314 kanssa samankaltaisten käyttäjien antamiin kirja-arvosteluihin. Nämä kirjasuositukset ovat myös kirjoja, joita käyttäjä 314 ei ole lukenut. Jos katsotaan valitun käyttäjän 314 parhaiten arvosteltuja kirjoja (ks. kuva 40), voimme päätellä käyttäjän pitävän erityisesti fantasia ja tieteisfiktionaalisista

nuortenkirjoista. Suurin osa käyttäjäpohjaisista kirjasuosituksista ovat kirjallisuuslajiltaan fiktionaalisia kuten Lord of the Flies, The Da Vinci Code, The Lovely Bones ja Old Man's War. Tämä osoittaa, miten samankaltaiset käyttäjät, jotka pitävät samankaltaisesta kirjallisuudesta auttavat suosittelualgoritmia ehdottamaan kirjoja, joiden olemassaolosta käyttäjä ei ole mahdollisesti edes tiennyt ennen niiden suositusta.



Kuva 40. Tulostetaan käyttäjän 314 parhaiten arvostelemia kirjoja.

Huomataan, että käyttäjä 314 on pitänyt Harry Potter sekä J.R.R. Tolkienin The Hobbit or There and Back Again -kirjoista. Näemme, miten samankaltaiset käyttäjät ovat pitäneet näiden kirjailijoiden muista kirjoista, minkä takia suosituksissa näkyy muita Harry Potter -sarjan sekä Tolkienin kirjoja. Valittu käyttäjä on pitänyt joistakin klassikkokirjoista, kuten Of Mice and Men sekä Fahrenheit 451:stä. Samankaltaiset käyttäjät, jotka ovat arvostelleet nämä klassikot samalla tavalla ovat myös pitäneet Nineteen Eighty-Four, Great Expectations ja Frankenstein, or the Modern Prometheus -klassikkokirjoista, minkä takia ne ovat nousseet suosituksien listalle. Tämä ominaisuus tekee suosituksista osittain persoonallisia, mikä on ominaista sosiaalisen median yhteisöpalvelujen suosittelijoille.

Saaduista suosituksista voidaan myös havaita, miten erilaisia kirjallisuuden lajityyppejä on suositeltu. Sosiaalisen median yhteisöpalveluissa on monenlaisia kategorioita, jotka on jaettu aiheittain. Instagramissa sisältöä voidaan kategorisoida hashtagien avulla ja Youtubessa videoita on jaettu aihealueittain, esimerkiksi peleihin ja uutisiin. Voidaan myös huomata, kuinka käyttäjäpohjainen suosittelija altistaa käyttäjän mahdollisesti uusille aiheille ja sisällölle. Tässä tapauksessa käyttäjälle on ehdotettu It kauhu- ja Dear John -romanssikirjaa. Tämä voi luoda vaihtelua ja näyttää käyttäjälle jotain häntä kiinnostavia uusia asioita, joita hän ei välttämättä olisi muuten nähnyt. Sosiaalisen median päivittyessä jatkuvasti samankaltaisen sisällön selaaminen saattaa nopeasti uuvuttaa tai tylsistyttää käyttäjän, mikä ei ole toivottavaa. Käyttäjäpohjainen suosittelija pystyy tekemään suosituksista vaihtelevia, koska se saa jatkuvasti käyttäjiltä uusia syötteitä, joiden pohjalta se seuloo suosituksia. Tämän avulla käyttäjää voidaan pitää kiinnostuneena sosiaalisen median yhteisöpalvelun käyttämisestä.

Toisaalta nämä suositukset ovat muiden käyttäjien mieltymyksistä osittain riippuvaisia. Vaikka kaksi käyttäjää saattaa pitää yhdestä asiasta, saattavat he olla eri mieltä toisesta. Käyttäjäpohjainen suosittelija ehdotti kirjaa Eat Pray Love: One Woman’s Search for Everything Across Italy, India and Indonesia, joka on tietokirjallisuuteen kuuluva elämäkerta. Todennäköisyys, että valitsemamme

käyttäjä kiinnostuisi lukemaan kyseisen kirjan on matala verrattuna muihin suosituksiin. Emme kuitenkaan osaa sanoa, onko kyseinen suositus kokonaan huono tai hyvä, koska meillä ei ole mahdollisuutta saada valitun käyttäjän mielipidettä asiaan liittyen. Mutta nämä erikoisemmat suositukset voivat lisätä huonojen suosituksien riskiä, mikä voi vaikuttaa käyttäjäkokemukseen negatiivisesti. Ne tekevät suosituksista myös vähemmän persoonallisia.

Jos käyttäjä saa jatkuvasti huonoja tai tylsiä suosituksia sosiaalisen median alustalla, hän todennäköisemmin lopettaa yhteisöpalvelun käytön. Siksi olisi mahdollisesti hyvä laittaa suosittelijalle jonkinlaiset rajat samankaltaisten käyttäjien etsimiseen. Käyttäjän täytyy pitää tietyn verran samoista asioista suosituksia saavan käyttäjän kanssa ennen kuin algoritmi ottaa käyttäjän samankaltaisten käyttäjien joukkoon. Tämä voisi tehdä suosituksista osuvampia ja vähentäisi huonojen suositusten riskiä. Tämä voi olla yksi syy, miksi esimerkiksi Facebook ja Instagram käyttävät useita vaiheita omissa suosittelualgoritmiprosesseissa. On myös mahdollista, että nämä hieman erikoisemmat ehdotukset ovat osuvia joillekin käyttäjille ja auttavat suosittelijaa selvittämään käyttäjän mieltymyksiä. Sosiaalisen median yhteisöpalvelussa on kannattavaa pitää käyttäjäkokemus yhtenäisenä ja varoa jatkuvia suuria vaihteluita esimerkiksi suosituksien kannalta, jotta vältytään käyttäjää häiritseviltä tekijöiltä.

Listalla on myös viidennelle riville saatu NaN-arvo eli kyseiselle kirjalle ei ollut määritelty nimeä, mikä tekee suosittelusta heikomman. Käytettävän tuotedatan täytyy siis olla hyvin määritelty, jotta tämänkaltaisia huonoja suosituksia ei tapahtuisi.

Tarkastellaan seuraavaksi tuote-tuotepohjaisen yhteistoiminnallisen suosittelijan tekemiä suosituksia käyttäjälle 314 (ks. kuva 41). Tälläkin prototyypillä käyttäjälle suositeltiin kirjoja, joita hän ei ole tiettävästi lukenut. Nähdään, että suositukset eroavat jonkin verran käyttäjäpohjaisen suosittelijan suosituksista. Tuotepohjaisen suosittelijan parhaimmat suositukset näyttävät painottuvan käyttäjän 314 pitämään fiktiiviseen nuortenkirjallisuuteen, kuten Twilight, The Fault in Our

Stars ja Divergent. Muutamia klassikoita on myös ehdotettu, esimerkiksi To Kill a Mockingbird, The Great Gatsby ja Pride and Prejudice. Koska suositukset pohjautuvat käyttäjän antamiin aikaisempiin arvosteluihin, suositukset ovat entistä persoonallisempia, mikä on hyvä ominaisuus sosiaalisen median alustalla.

```
Käyttäjä 314 saattaa pitää näistä kirjoista:
Twilight
To Kill a Mockingbird
The Great Gatsby
The Fault in Our Stars
Pride and Prejudice
Divergent
Män som hatar kvinnor
Catching Fire
Harry Potter and the Prisoner of Azkaban
Harry Potter and the Order of the Phoenix
Harry Potter and the Chamber of Secrets
Lord of the Flies
Of Mice and Men
Fifty Shades of Grey
The Lion, the Witch and the Wardrobe
The Time Traveler's Wife
Water for Elephants
The Book Thief
The Adventures of Huckleberry Finn
Cien años de soledad
```

Kuva 41. Tuote-tuotepohjaisen yhteistoiminnallisen seulonnan tuottamat suositukset.

Ehdotuksissa löytyy myös joitakin samoja suosituksia, kuten muita Harry Potter -kirjoja, Lord of the Flies ja Of Mice and Men, mutta ei kuitenkaan J.R.R. Tolkienin kirjoja, mitä käyttäjäpohjainen suosittelualgoritmi taas ehdotti. Tämä voi mahdollisesti johtua kirjoille annettujen määritelmien vajaudesta tai puutteesta, mikä ei vaikuta niin suuresti käyttäjäpohjaiseen suosittelualgoritmiin verrattuna tuotepohjaiseen suosittelualgoritmiin. Jos tuotteille annetut määritelmät ovat vajaat tai niissä on jokin virhe, ne vaikuttavat suoraan tuotepohjaisen yhteistoiminnallisen suosittelualgoritmin tekemien suosituksien laatuun. Sosiaalisen median yhteisöpalvelussa, missä käyttäjät ovat usein itse vastuussa omien julkaisujen määrittelemisestä, kuten videon tai kuvan nimeämisestä, heidän tekemät virheet vaikuttavat suoraan tuote-tuotepohjaiseen suosittelijaan. Tämän ongelman

välttämiseksi voisi jälleen luoda rajoituksia, esimerkiksi tuotteesta pitää löytyä tietty määrä ominaisuuksia, jotka ovat samankaltaisia käyttäjän mieltymysten kanssa.

Kaiken kaikkiaan tuotepohjaisen yhteistoiminnallisen suosittelijan tekemät suositukset pysyvät käyttäjän 314 pitämien kirjallisuuslajien sisällä, mikä pitää ehdotukset enemmän persoonallisina kuin käyttäjäpohjainen yhteistoiminnallinen suosittelualgoritmi. Mutta suosituksista nousi yksi hieman erikoisempi ehdotus, mikä eroaa muista kirjoista ainakin kirjallisuuslajiltaan, ja se on Fifty Shades of Grey. Eli tuotepohjainen suosittelija saattaa ehdottaa myös erilaisia tuotteita, mutta ei kuitenkaan niin paljon, mitä käyttäjäpohjainen suosittelija. Täytyy muistaa, että tämä suositus perustuu käyttäjän aikaisempiin arvosteluihin. Käyttäjä on saattanut lukea pari erilaisen kirjallisuustyypin kirjaa, minkä takia algoritmi tuotti hieman erikoisemman ehdotuksen. Eli suositus on edelleen persoonallinen, mutta sen osuvuutta on silti vaikea arvioida. Toisin kuin käyttäjäpohjaisessa suosittelualgoritmissa, tuotepohjaisen suosittelualgoritmin suositukset ovat todennäköisemmin sellaisia, joista käyttäjä pitää, koska ne ovat persoonallisempia. Sosiaalisen median yhteisöpalvelussa tämä on suositeltavaa, koska se lisää käyttäjän positiivista käyttökokemusta, kun käyttäjä pitää saamistaan suosituksista. Toisaalta riskinä voi olla, että suositukset muuttuvat tylsiksi, kun käyttäjälle ehdotetaan samoja aiheita koskevaa sisältöä.

Lopuksi katsellaan sisältöpohjaisen suosittelualgoritmiprototyypin tuottamia tuloksia (ks. kuva 42). Tämän suosittelijan tekemät suositukset eroavat kaikista eniten muista tuloksista niiden suoraviivaisuuden takia. Suosittelija ei ottanut huomioon käyttäjän aikaisemmin lukemia kirjoja, koska suosituksissa haluttiin painottaa sisältöpohjaisen suosittelijan keskittymistä tuotteisiin liitettyihin määritelmiin. Algoritmille annettiin tehtäväksi etsiä samanlaisia kirjoja kuin käyttäjän 314 pitämä Harry Potter and the Philosopher’s Stone -kirja. Samaa kirjaa käytettiin tuotepohjaisessa yhteistoiminnallisessa suosittelualgoritmissa. Sisältöpohjainen suosittelualgoritmi otti suosituksissa huomioon kirjan nimen, julkaisuvuoden ja kirjailijan.

```
Samankaltaisia kirjoja kuin Harry Potter and the Philosopher's Stone :
Harry Potter and the Philosopher's Stone
Harry Potter and the Deathly Hallows
Harry Potter and the Order of the Phoenix
Harry Potter and the Chamber of Secrets
Harry Potter and the Goblet of Fire
Harry Potter and the Half-Blood Prince
Harry Potter and the Prisoner of Azkaban
Harry Potter Boxed Set Books 1-4
Harry Potter Collection (Harry Potter, #1-6)
nan
Harry Potter and the Cursed Child, Parts One and Two
Harry, A History: The True Story of a Boy Wizard, His Fans, and Life Inside the Harry Potter Phenomenon
Complete Harry Potter Boxed Set
The Chronicles of Narnia: The Lion, the Witch and The Wardrobe (Sheet Music)
The Country Mouse and the City Mouse, The Dog and His Bone, The Fox and the Crow (A Little Golden Book)
The Lion, the Witch and the Wardrobe
Harry Potter and the Chamber of Secrets: Sheet Music for Flute with C.D
Billions and Billions: Thoughts on Life and Death at the Brink of the Millennium
The Hobbit and The Lord of the Rings
The Agony and the Ecstasy: A Biographical Novel of Michelangelo
```

Kuva 42. Sisältöpohjaisen suosittelualgoritmin tuottamat suositukset.

Heti alkuun voidaan huomata, miten suosittelija ehdotti ensimmäiseksi todennäköisesti kaikki kirjadatassa olevat kirjat, jotka liittyvät Harry Potteriin. Tämä on hyvä esimerkki sisältöpohjaisen suosittelualgoritmin yksinkertaisuudesta. Sen tuottamat suositukset ovat osuvia siinä, että ne kuuluvat käyttäjän pitämään kirjallisuuteen, mikä tekee suosituksista hyvin persoonallisia. Sisältöpohjainen suosittelualgoritmi antoi myös kaikki mahdolliset kirjasuositukset liittyen Harry Potteriin. Sosiaalisen median yhteisöpalvelussa tällainen suosittelu olisi hyvin tehokas löytämään käyttäjälle nopeasti tarkennettua sisältöä, josta hän pitäisi. Yhteisöpalvelut sisältävät tuhansittain erilaisia julkaisuja, joten nopea ja tarkka suosittelu voisi olla joissain tapauksissa varsin kätevää.

Mutta voivatko suositukset olla liian persoonalliset? Yksi kirjasuosituksista, The Country Mouse and the City Mouse; The Fox and the Crow; The Dog and His Bone on lastenkirja, mikä taas ei kuulu käyttäjän 314 lempikirjallisuuteen eli se on todennäköisesti huono suositus. Vaikka suositukset ovat tarkkoja, ne kärsivät monipuolisuuden puutteesta. Yhteisöpalvelussa on tärkeää pystyä pitämään käyttäjän mielenkiinto yllä ja suositella tarpeeksi monipuolista sisältöä, minkä suhteen sisältöpohjainen suosittelija näyttää takkuilevan.

Tässä suosittelussa korostuu entistä enemmän tuotteilla sekä käyttäjillä olevat määrittelyt. Jos käyttäjä rajoittaa itsestään annettujen tietojen jakamista suosittelualgoritmin käsiteltäväksi, sillä on suora vaikutus suosituksiin. Sosiaalisen median yhteisöpalvelun käyttäjillä voi olla tämän takia varsin erilaiset käyttökokemukset. Myös tuotteille annettujen määrittelyjen painoarvo nousee sisältöpohjaisessa suosittelijassa. Jos käyttäjä laittaa omaan julkaisuun kirjoitusvirheen tai muun väärän määrittelyn, algoritmi ei välttämättä ota julkaisua suosittelussaan huomioon tai saattaa vahingossa suositella sitä vääränlaiselle käyttäjälle. Yhteisöpalvelussa tämänlainen suosittelujen arvaamattomuus ei ole toivottua, koska se voi todennäköisesti vaikuttaa taas käyttäjien käyttökokemukseen negatiivisesti. Ongelmaa voisi mahdollisesti pienentää jälleen jonkinlaisella rajoituksella, esimerkiksi käyttämällä sisältöpohjaista algoritmia vain tiettyyn yhteisöpalvelun sisäiseen ominaisuuteen, kuten sisällön tarkennettuun hakemiseen.

6 Yhteenveto

Insinöörityön tavoitteena oli tutkia, miten käytetyimmät suosittelualgoritmimallit toimivat ja miten niitä käytetään sosiaalisen median yhteisöpalveluissa. Tavoitteena oli tämän pohjalta selvittää, minkälainen suosittelualgoritmi sopisi parhaiten sosiaalisen median yhteisöpalvelualustalle. Työssä tutkittiin tarkemmin käytetyimpien suosittelualgoritmien toimintaa samankaltaisuuden mittojen avulla ja vertailtiin niiden heikkouksia ja vahvuuksia. Samalla tarkasteltiin suosituimpia yhteisöpalveluja ja sitä, miten ne käyttävät suosittelualgoritmeja omilla alustoillaan. Lisäksi pohdittiin sosiaalisen median vaikutusta suosittelualgoritmin toimintaan. Läpikäytyjen suosittelualgoritmien vertailemiseksi luotiin kolme suosittelualgoritmiprototyyppiä, joiden avulla tehtiin kirja-datasetin pohjalta valitulle käyttäjälle suosituksia. Saatuja suosituksia vertailtiin keskenään ja pohdittiin niiden avulla algoritmien soveltuvuutta sosiaalisen median yhteisöpalvelun alustalle.

Prototyypeissä käytettiin muistipohjaisia eli käyttäjä- ja tuotepohjaista yhteistoiminnallisia suosittelualgoritmeja sekä sisältöpohjaista suosittelualgoritmia. Jokainen prototyyppi saatiin toimimaan ja tuottamaan haluttu määrä suosituksia

onnistuneesti. Tuloksien pohjalta saatiin selville, mitä suosittelualgoritmi ottaa huomioon suosituksia tehdessään sekä miten erilaiset tekijät, kuten käsiteltävän datan määritykset voivat vaikuttaa saatuihin suosituksiin.

Käyttäjäpohjainen yhteistoiminnallinen suosittelualgoritmi tuotti valitulle käyttäjälle monipuolisia suosituksia. Osa oli persoonallisia, mutta ehdotuksissa huomattiin olevan eniten käyttäjän mieltymyksistä poikkeavia suosituksia. Sosiaalisen median yhteisöpalvelussa suositaan persoonallisia suosituksia sekä vaihtelua, millä vältetään käyttäjän tylsistymistä. Mutta liian vaihtelevilla ja erikoisilla suosituksilla voi olla riski aiheuttaa huonoja ehdotuksia, mikä taas voi vaikuttaa käyttäjän käyttökokemukseen negatiivisesti. Tuotepohjaisen yhteistoiminnallisen suosittelualgoritmin tulokset olivat käyttäjäpohjaiseen verrattuna persoonallisempia eli suositukset vastasivat käyttäjän henkilökohtaisia mieltymyksiä. Suosituksista nousi yksi poikkeava suositus, mikä tuo käyttäjälle hieman vaihtelua altistamalla hänet uudelle sisällölle, mikä vaikuttaa juuri sopivalta määrältä sosiaalisen median yhteisöpalvelussa. Toisaalta huomattiin, kuinka yhteisöpalvelun käyttäjien tekemät määrittelyvirheet voivat heikentää tuotepohjaisen suosittelualgoritmin tehokkuutta. Sisältöpohjaisen suosittelualgoritmin tulokset olivat kaikista yksinkertaisimmat. Sen suositukset olivat tarkkoja ja osuvia, mikä on toivottua sosiaalisen median yhteisöpalvelussa. Kuitenkin suositukset osoittivat puutetta monipuolisuudessa. Sosiaalisen median yhteisöpalvelussa pyritään pitämään käyttäjän mielenkiintoa yllä tarjoamalla erilaista sisältöä, ettei käyttäjä tylsisty ja lopeta yhteisöpalvelun käyttöä.

Saatujen tuloksien ja niiden vertailun pohjalta ajattelen, että kolmesta algoritmista tuote-tuotepohjainen yhteistoiminnallinen suosittelualgoritmi toimisi parhaiten koeasetelman sosiaalisen median yhteisöpalvelulle. Sen pääpaino pysyy persoonallisissa suosituksissa, kuten sisältöpohjaisessa suosittelualgoritmissa, mutta sen lisäksi se altistaa käyttäjän aina silloin tällöin uudelle, vaihtelevalle sisällölle, kuten käyttäjäpohjaisessa suosittelualgoritmissa. Pitää kuitenkin muistaa, että suosittelualgoritmeja sekä sosiaalisen median yhteisöpalveluja on monenlaisia. Suosittelualgoritmin soveltuvuus ja hyödyllisyys riippuvat paljolti pal-

velun tarjoajan asettamista käyttötarkoitustavoitteista, mikä tekee suosittelualgoritmin soveltuvuudesta enemmän tapauskohtaisen. Tämänkaltaiset testit sekä erilaisten suosittelualgoritmien vertailut voivat kuitenkin tarjota tienviittaa kehittäjille ja palveluntarjoajille suosittelualgoritmin valitsemisessa.

Täytyy myös ottaa huomioon, että vertailussa käytettiin 20 suositusta. Sosiaalisen median yhteisöpalvelut sisältävät valtavia määriä dataa, joita suosittelija käy jatkuvasti läpi. Koeasetelma oli tähän verrattuna varsin yksinkertainen, joten on mahdollista, että tässä vertailussa tehdyt havainnot voivat olla sitä erilaisempia, mitä isompia ja monimutkaisempia datasettejä algoritmit seulonnassaan käyttävät. Pitää myös huomioida, että tässä työssä käsiteltiin kolmea käytetyintä suosittelualgoritmimallia. Näiden lisäksi on monia muitakin malleja, joista prototyyppiä ei ehditty tekemään. Voi olla, että joku käsittelemättömistä suosittelualgoritmimalleista soveltuisi vielä paremmin sosiaalisen median yhteisöpalvelulle. Työtä voisi jatkossa laajentaa skaalaamalla prototyyppien käyttämää datasettiä vastaamaan paremmin sosiaalisen median yhteisöpalvelun dataa sekä testaamalla muita suosittelualgoritmimalleja ja samankaltaisuuden mittoja. Tätä insinöörityötä voidaan käyttää myös apuna muille samankaltaisille tutkimuksille sekä suosittelualgoritmien toiminnan perehdyttämisessä.

Lähteet

1 Burke Robin, Felfernig Alexander, Göker Mehmet H.. 2011. Recommender Systems: An Overview. AI Magazine Vol. 32 No. 3. Verkkoaineisto. Saatavissa: https://ojs.aaai.org/index.php/aimagazine/issue/view/195 Luettu: 23.7.2022.

2 Jannach Dietmar, Zanker Markus, Felfernig Alexander Felfernig, Friedrich Gerhard. 2011. Recommender Systems: An Introduction. Cambridge: Cambridge University Press.

3 Ricci Francesco, Rokach Lior, Shapira Bracha, Kantor Paul B.. 2010. Recommender Systems Handbook. New York: Springer.

4 Nerge Elsa. 2015. Information and Recommender Systems. London: ISTE Ltd and John Wiley Sons Inc.

5 Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg. Verkkoaineisto. Saatavissa: https://doi.org/10.1007/978-3-540-72079-9_9 Luettu: 19.9.2022.

6 John S. Breese, David Heckerman, Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence. Verkkoaineisto. Saatavissa: https://arxiv.org/ftp/arxiv/papers/1301/1301.7363.pdf Luettu:17.9.2022.

7 Fiasconaro A., Latora V., Mantegna R. N., Nicosia V., Tumminello M.. 2021. Hybrid recommendation methods in complex networks. Verkkoaineisto. Saatavissa: https://core.ac.uk/works/18031740 Luettu: 22.9.2022.

8 Sarwar Badrul, Karypis George, Konstan Joseph, Riedl John. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. Verkkoaineisto. Saatavissa: https://www.researchgate.net/publication/2369002_Item-based_Collaborative_Filtering_Recommendation_Algorithms Luettu: 22.9.2022.

9 Arias Jose J. Pazos, Vilas Ana Fernandez, Redondo Rebeca P. Diaz. 2014. Recommender Systems for the Social Web. Berlin: Springer-Verlag Berlin and Heidelberg GmbH & Co. KG.

10 Crossing Minds. 2020. What are today's top recommendation engine algorithms? Verkkoaineisto. Saatavissa: https://itnext.io/what-are-the-top-recommendation-engine-algorithms-used-nowadays-646f588ce639 Luettu: 30.10.2022.

11 Mohanty Sachi Nandan, Chatterjee Jyotir Moy, Jain Sarika, Elngar Ahmed A., Gupta Priya. 2020. Recommender System with Machine Learning and Artificial Intelligence. Yhdysvallat: John Wiley & Sons, Inc 2020.

12 Rocca Baptiste. 2019. Introduction to recommender systems. Verkkoaineisto. Saatavissa: https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada Luettu: 5.10.2022.

13 Throat Poonam B., Goudar R. M., Barve Sunita. 2015. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. International Journal of Computer Applications (0975 – 8887) Volume 110 No. 4. Verkkoaineisto. Saatavissa: https://www.academia.edu/39606345/Survey_on_Collaborative_Filtering_Content_based_Filtering_and_Hybrid_Recommendation_System?auto=citations&from=cover_page Luettu: 13.10.2022.

14 Çano Erion, Morisio Maurizio. 2019. Hybrid Recommender Systems: A Systematic Literature Review. Verkkoaineisto. Saatavissa: https://arxiv.org/abs/1901.03888 Luettu: 1.10.2022.

15 R. Burke. 1999. Integrating knowledge-based and collaborative filtering recommender systems. AAAI Workshop on Artificial Intelligence for Electronic Commerce WS-99-01. AAAI Press. Menlo Park. California. Verkkoaineisto. Saatavissa: https://www.aaai.org/Papers/Workshops/1999/WS-99-01/WS99-01-011.pdf Luettu: 1.10.2022.

16 Sadiku Matthew N. O., Omotoso Adedamola A., Musa Sarhan M.. 2019. Social Networking. Verkkoaineisto. Saatavissa: https://www.academia.edu/39402250/Social_Networking?auto=citations&from=cover_page Luettu: 17.9.2022.

17 We Are Social, Hootsuite, DataReportal. 2022. Most popular social networks worldwide as of January 2022, ranked by number of monthly active users (in millions). Statista. Statista Inc.. Verkkoaineisto. Saatavissa: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/ Luettu: 2.10.2022.

18 Meta. 2022. About Facebook App. Verkkoaineisto. Saatavissa: https://about.meta.com/technologies/facebook-app/ Luettu: 2.10.2022.

19 Meta. 2022. Sisällön sijoittelu Facebookissa. Metan ohje- ja tukikeskus yrityksille. Verkkoaineisto. Saatavissa: https://www.facebook.com/business/help/718033381901819 Luettu: 14.10.2022.

20 Horwitz Jeff, Seetharman Deepa. 2020. Facebook Executives Shut Down Efforts to Make the Site Less Divisive. The Wall Street Journal. Verkkoaineisto. Saatavissa: https://www.wsj.com/articles/facebook-knows-it-encourages-division-top-executives-nixed-solutions-11590507499 Luettu: 14.10.2022.

21 Perrigo Billy. 2021. Here's How to Fix Facebook, According to Former Employees and Leading Critics. TIME. Verkkoaineisto. Saatavissa: https://time.com/6103793/how-to-fix-facebook/ Luettu: 14.10.2022.

22 Youtube. 2022. Terms of Service. Verkkoaineisto. Saatavissa: https://www.youtube.com/static?template=terms Luettu: 21.9.2022.

23 Goodrow Cristos. 2021. On YouTube's recommendation system. Inside Youtube. Youtube Official Blog. Verkkoaineisto. Saatavissa: https://blog.youtube/inside-youtube/on-youtubes-recommendation-system/ Luettu: 21.9.2022.

24 The Youtube Team. 2019. The Four Rs of Responsibility, Part 2: Raising authoritative content and reducing borderline content and harmful misinformation. Inside Youtube. Youtube Official Blog. Verkkoaineisto. Saatavissa: https://blog.youtube/inside-youtube/the-four-rs-of-responsibility-raise-and-reduce/ Luettu: 22.9.2022.

25 Geurkink Brandi, McDonald Helena. 2020. Congratulations, YouTube... Now Show Your Work. Mozilla Foundation. Verkkoaineisto. Saatavissa: https://foundation.mozilla.org/en/blog/congratulations-youtube-now-show-your-work/ Luettu: 22.9.2022.

26 Instagram Ohje- ja tukikeskus. 2022. Mikä on Instagram? Verkkoaineisto. Saatavissa: https://help.instagram.com/182492381886913/?helpref=hc_fnav Luettu: 25.9.2022.

27 Mosseri Adam. 2021. Shedding More Light on How Instagram Works. About Instagram. Instagram. Verkkoaineisto. Saatavissa: https://about.instagram.com/blog/announcements/shedding-more-light-on-how-instagram-works Luettu: 25.9.2022.

28 Stienstra Flávia. 2021. Racial bias and #IWantToSeeNyome on Instagram. Diggit Magazine. Verkkoaineisto. Saatavissa: https://www.diggitmagazine.com/articles/racial-bias-iwanttoseenyome-instagram Luettu: 28.9.2022.

29 Cook Jesselyn. 2019. Women Are Pretending To Be Men On Instagram To Avoid Sexist Censorship. Huffpost. Verkkoaineisto. Saatavissa: https://www.huffingtonpost.co.uk/entry/women-are-pretending-to-be-men-on-instagram-to-avoid-sexist-censorship_n_5dd30f2be4b0263fbc99421e Luettu: 28.9.2022.

30 Dickson EJ. 2019. Why Did Instagram Confuse These Ads Featuring LGBTQ People for Escort Ads? Rolling Stone. Verkkoaineisto. Saatavissa: https://www.rollingstone.com/culture/culture-features/instagram-transgender-sex-workers-857667/ Luettu: 27.9.2022.

31 Amin Faiza. 2021. The growing criticism over Instagram's algorithm bias. CityNews Toronto. Verkkoaineisto. Saatavissa: https://toronto.citynews.ca/2021/04/05/the-growing-criticism-over-instagrams-algorithm-bias/ Luettu: 29.9.2022.

32 Parsons Vic. 2021. Instagram accused of 'sexualising, policing and censoring trans bodies'. PinkNews. Verkkoaineisto. Saatavissa: https://www.pinknews.co.uk/2021/04/22/instagram-trans-bodies-censorhip-we-deserve-to-be-here/ Luettu: 29.9.2022.

33 Mosseri Adam. 2020. Ensuring Black Voices are Heard. Instagram Blog. Verkkoaineisto. Saatavissa: https://about.instagram.com/blog/announcements/ensuring-black-voices-are-heard Luettu: 29.9.2022.

34 Instagram Ohje- ja tukikeskus. Mitä Instagram-suositukset ovat? Meta. Verkkoaineisto. Saatavissa: https://help.instagram.com/313829416281232 Luettu: 8.10.2022.

35 BBC News Technology. 2020. Facebook and Instagram to examine racist algorithms. BBC. Verkkoaineisto. Saatavissa: https://www.bbc.com/news/technology-53498685 Luettu: 8.10.2022.

36 Cambridge Dictionary. 2022. Meaning on social media in English. Verkkoaineisto. Saatavissa: https://dictionary.cambridge.org/dictionary/english/social-media Luettu: 30.10.2022.

37 Lightning Guides. 2015. Social media: Facebook, Twitter and the Modern Revolution. Berkeley: Sonoma Press.

38 Statista. 2022. Number of social media users worldwide from 2018 to 2027 (in billions). Statista. Statista Inc.. Verkkoaineisto. Saatavissa: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/ Luettu: 2.10.2022.

39 Suosittelualgoritmiprototyypeissä käytetty “goodbooks-10k” -kirjadatasetti. Verkkoaineisto. Saatavissa: https://www.kaggle.com/datasets/zygmunt/goodbooks-10k?select=ratings.csv.

40 About us. Project Jupyter’s origins and governance. Verkkoaineisto. Saatavissa: https://jupyter.org/about Luettu: 16.10.2022.

41 GreekDataGuy. 2019. Build a user-based collaborative filtering recommendation engine for Anime. Verkkoaineisto. Saatavissa: https://towardsdatascience.com/build-a-user-based-collaborative-filtering-recommendation-engine-for-anime-92d35921f304 Luettu: 2.10.2022.

42 Javed Mahnoor. 2020. Using Cosine Similarity to Build a Movie Recommendation System. Verkkoaineisto. Saatavissa: https://towardsdatascience.com/using-cosine-similarity-to-build-a-movie-recommendation-system-ae7f20842599 Luettu: 10.10.2022.