TUTKIMUSRAPORTTI

VTT-R-00024-20

Tekoäly mediasisältöjen esittämisen tukena

Kirjoittajat: Asta Bäck, Timo Laakko

Luottamuksellisuus: Julkinen

Raportin nimi Tekoäly mediasisältöjen esittämisen tukena	
Asiakkaan nimi, yhteyshenkilö ja yhteystiedot MEDIA-ALAN TUTKIMUSSÄÄTIÖ	**Asiakkaan viite**
Projektin nimi Tekoälyä mediasisällön personointiin ja kohderyhmien tunnistamiseen	**Projektin numero/lyhytnimi** Älymepe
Raportin laatija(t) Asta Bäck, Timo Laakko	**Sivujen/liitesivujen lukumäärä** 31
Avainsanat Vahvistusoppiminen, tekoäly, suositukset, personointi	**Raportin numero** VTT-R-00024-20

Tiivistelmä

Tutkimuksen tavoitteena oli tarkastella kone- ja vahvistusoppimismenetelmien hyödyntämismahdollisuuksia mediatuotannossa, erityisesti liittyen mediasisältöjen esittämiseen etusivulla. Raportin teoriaosiossa on esitelty vahvistusoppimismenetelmiä: on käsitelty sekä yksivaiheiseen päätöksentekotilanteeseen sopivia, ns. monikätisen rosvon problematiikan ratkaisualgoritmeja, että täysmittaista vahvistusoppimista, jossa ennakoidaan vasta usean vaiheen jälkeen saavutettavissa olevaa tavoitetta.

Sisältöjen esittämistä tarkasteltiin kahden esimerkkikohteen avulla. Ensimmäisenä kohteena oli tutkia paikan merkitystä juttujen suosioon ja tässä käytettiin esimerkkilehden aineistoa, josta ilmeni jutun sijainti etusivulla yläreunasta katsottuna, jutun ominaisuudet kuten ikä, kategoria ja otsikko, sekä etusivun kautta tulleet klikkaukset viiden minuutin aikavälein. Tilastollinen tarkastelu osoitti, että mitä korkeammalla juttu on etusivulla, sitä enemmän se saa klikkauksia. Merkitys kuitenkin tasoittuu suhteellisen nopeasti mentäessä alemmaksi etusivulla. Kerätyn aineiston pohjalta tehtiin ennustava koneoppimismalli, joka edeltävän aikajakson klikkausten ja jutun ja etusivun dynaamisten ja staattisten ominaisuuksien pohjalta tekee ennusteen jutulle eri paikoissa kertyvästä klikkausmäärästä. Tätä voidaan käyttää määrittelemään etusivun artikkeleille hyvä järjestys.

Toisessa esimerkkikohteessa käytettiin simuloitua dataa, eli generoitiin annettujen parametrien perusteella käyttäjät, jutut ja käyttäjäsessiot. Aineiston avulla laadittiin malli, joka arvioi tarjolla olevien juttujen käyttäjäkohtaisen hyvyyden, kun hyvyys määriteltiin tilaukseen johtavana käyttäytymisenä. Tilausta indikoivina merkkeinä käytettiin maksullisten artikkelin klikkaamista. Malli opetettiin simuloidulla datalla, ja sitä voidaan käyttää pohjana todellisen datan kanssa tehtävälle jatkokehitystyölle.

Hanke tarkasteli jutturesurssien käyttöä ja avaa mahdollisuuksia hyvän esitysjärjestyksen ja etusivun määrittelemiseen koneoppimismenetelmiä hyödyntäen. Ensimmäinen esimerkki näytti, että jos käyttäjistä ei ole tausta- tai preferenssitietoa, esitysjärjestyksen määrittelyssä voidaan käyttää aiemmilta käyttäjiltä syntynyttä palautetietoa sekä myös juttujen joitakin ominaisuuksia. Toisessa esimerkissä oletettiin olevan tietoa käyttäjien preferensseistä ja istunnoista sekä tilaukseen johtaneista käyttäytymispoluista, minkä pohjalta vahvistusoppimismalli pystyy generoimaan käyttäjäkohtaisia suosituksia. Kehitetty ohjelmisto tarjoaa ympäristön, jossa voidaan tehdä ensimmäisen vaiheen kokeiluita ennen todellisen tiedonkeruun käynnistämistä ja menetelmien käyttöä todellisella aineistolla.

Luottamuksellisuus	Julkinen

Espoo 13.1.2020

Laatijat	Tarkastaja	Hyväksyjä
Asta Bäck, Johtava tutkija	Carl Södergård Tutkimusprofessori	Tuomo Tuikka Tutkimuspäällikkö
Timo Laakko Erikoistutkija		

VTT:n yhteystiedot
Asta Bäck, 050 5515187, asta.back@vtt.fi

Jakelu (asiakkaat ja VTT)
Vapaa

Sisällysluettelo

1. Johdanto

Mediatalojen uutispalvelujen tavoitteena on tarjota luotettava ja kattava uutislähde, jossa asiakkaat viihtyvät, johon he tulevat toistuvasti ja jonka käytöstä he ovat valmiita maksamaan. Avainasemassa on kyky tuottaa sisältöä, joka kiinnostaa ja aktivoi lukijoita. Uutistuotanto on luonteeltaan hyvin dynaamista: uutta sisältöä syntyy jatkuvasti ja uudesta tuotannosta pitäisi valita esitettäväksi käyttäjiä parhaiten palveleva kokonaisuus. Suosittelumenetelmien käytölle haasteena on juttujen lyhyt elinikä, joten juttujen kiinnostavuudesta saatavaa tietoa pitää hyödyntää hyvin nopeasti.

Oman näkökulmansa sisällön esittämiseen tuo lehden liiketoimintamali. Tilattavien lehtien kohdalla tyypillinen ratkaisu on, että osa lehden sisällöstä on tarjolla kaikille ja osa vain tilaajille. Vain tilaajille tarkoitettujen juttujen tarjoaminen ei-tilaajille on tarpeen, jotta ei-tilaajat tiedostavat, mitä tarjottavaa lehdellä on tilaajille. Voidaan kuitenkin myös olettaa, että mitä enemmän etusivulla on vain tilaajille tarkoitettuja artikkeleita, sitä vähemmän se kannustaa ei-tilaajia käymään palvelussa ja tutustumaan sen tarjontaan.

Vahvistusoppiminen on ohjatun (supervised) ja ohjaamattoman (unsupervised) opetuksen menetelmien kanssa yksi koneoppimismenetelmien pääryhmistä. Vahvistusoppimisen menetelmät ovat viime aikoina saaneet paljon huomiota dynaamisiin tilanteisiin liittyvän päätöksenteon automatisoinnissa niillä saatujen hyvien tulosten ansiosta. Tunnettu esimerkki on vahvistetun oppimisen soveltaminen voitokkaasti Go-peliin (Silver et al. 2016).

Vahvistusoppimismenetelmät voidaan jakaa kahteen ryhmään sen mukaan, onko kyse yksi- vai useampivaiheisesta päätöksentekotilanteesta. Yksivaiheisessa tilanteessa palaute päätöksen hyvyydestä saadaan välittömästi ja tällöin on kyse stokastisesta allokointiongelmasta. Näitä tilanteita kutsutaan monikätinen rosvo (Multi-armed bandit, MAB) -ongelmiksi. Jos palautetta päätöksen hyvyydestä ei saada välittömästi vaan palaute tulee vasta useasta vaiheesta koostuvan päätöksentekoketjun päätteeksi, on sovellettava täysmittaista vahvistusoppimista. Siinä keskeistä on kyetä arvioimaan, mitä tarjolla olevista monista vaihtoehtoisista reiteistä tulisi kulloinkin käyttää.

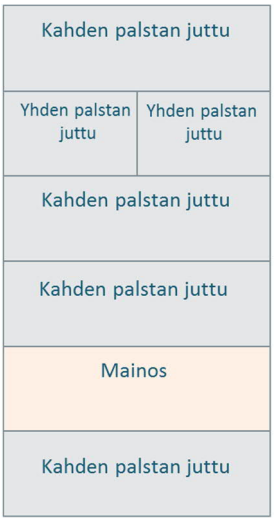
2. Tavoite, kohdealue ja rajaukset

Tutkimuksen tavoitteena oli tarkastella koneoppimismenetelmien ja erityisesti vahvistusoppimisen hyödyntämismahdollisuuksia mediatuotannossa ja antaa näin suuntaviivoja jatkotutkimukselle ja yritysten kehitystoimenpiteille. Tarkennetuksi kohdealueeksi valittiin hankkeessa mukana olleiden yritysten kanssa käytyjen keskustelujen perusteella mediasisältöjen esittäminen etusivulla.

Tutkimuksen toteuttamisessa oli neljä vaihetta:

1. vahvistusoppimisen menetelmiin ja soveltuvuuteen tutustuminen kirjallisuuden perusteella,
2. kohdealueiden valinta yhdessä esimerkkiyritysten kanssa,
3. kokeellinen työ,
4. tulosten arviointi ja raportointi

Uutissivustojen etusivuilla paljon käytetty rakenne on yhden tai kahden palstan uutisista muodostuva kokonaisuus (Kuva 1). Lehden liiketoimintamallista riippuen osa jutuista voi olla vain tilaajien luettavissa. Yleensä tämä merkitään näkyville jo etusivulla niin, että lukija tietää jo ennen jutun avaamista kyseessä olevan maksullinen juttu. Jos ei-tilaaja avaa tällaisen jutun, jutusta tyypillisesti näytetään vähän alkua ja linkki tilauslomakkeelle.



Kuva 1. Uutissivuston etusivu koostuu tyypillisesti kahden ja yhden palstan jutuista ja mahdollisista mainoksista.

Case-yritysten kanssa käydyissä keskusteluissa tarkennetuiksi tutkimuskohteiksi valittiin:

- Juttujen sijainnin vaikutus niiden suosioon, ja
- Etusivun koostaminen vahvistusoppimista hyödyntäen tavoitteena tilausten aikaansaaminen.

Etusivun koostamisessa otetaan huomioon monia tekijöitä, kuten lehden toimituspolitiikka, lukijoiden kiinnostusten kohteet ja ajankohtaisuus. Sähköisen etusivun rakennetta voidaan muokata nopeaan tahtiin, mikä antaa mahdollisuuden ottaa huomioon kävijöiden käyttäytyminen etusivun sisällön päivittämisessä; mikäli kävijöistä on taustatietoa, voidaan etusivun rakenteen valinnassa ottaa huomioon myös käyttäjä- tai käyttäjäryhmäkohtainen tieto.

Jokaisella mediatalolla on omat toimintaperiaatteensa ja tavoitteensa juttujen järjestämisessä etusivulle, mutta yhteistä on, että etusivun yläosaan pyritään sijoittamaan kulloinkin kiinnostavimmiksi arvioidut jutut, jotta lukijat löytävät nopeasti luettavaa ja kokevat näin palvelun hyödylliseksi ja toimivaksi.

Tilauksen tekemistä voidaan edistää kahdella tavalla. Ensinnäkin pyritään tietenkin tekemään ja esittämään käyttäjiä kiinnostavia juttuja, ja toiseksi tasapainottamaan ilmaista ja maksullista tarjontaa. Maksullisten artikkelien esiin nostaminen pyrkii houkuttelemaan tilaajaksi, mutta jos ilmaisia artikkeleita on kovin vähän, ei se kannusta vierailemaan palvelussa.

Tämän tutkimuksen tavoitteena on siis edistää algoritmisten menetelmien hyödyntämistä etusivun koostamisessa, sekä lisätä ymmärrystä algoritmisten menetelmien hyödyntämismahdollisuuksista mediasovelluksissa.

Tutkimus avaa mahdollisuuksia ensisijaisesti dynaamiseen mutta myös ennakoivaan hyödyntämiseen. Dynaaminen hyödyntäminen tarkoittaa algoritmien liittämistä osaksi julkaisujärjestelmää niin, että juttujen sijoituspäätöksen tukena käytetään algoritmisia menetelmiä. Ennakoiva hyödyntäminen tarkoittaa saadun ymmärryksen käyttämistä sisältöjen tuottamisen yhteydessä yksinkertaisimmillaan ohjeina ja suuntaviivoina, ja kehittyneemmässä muodossa toimittajaa tukevana järjestelmänä.

3. Koneoppimismenetelmistä

3.1 Koneoppimisen perusmenetelmät

Koneoppimismenetelmät luokitellaan yleensä kolmeen ryhmään: ohjaamaton, ohjattu ja vahvistusoppiminen. *Ohjaamattomassa oppimisessa* ei tarvita opetusaineistoa, vaan algoritmit tekevät päätelmiä aineistosta ilman opetusaineistoa. Ohjaamattoman oppimisen menetelmät ovat pääosin lajittelumenetelmiä, joissa algoritmi kykenee tunnistamaan aineistossa olevia ominaisuuksia ja rakenteita ja ryhmittelemään aineiston näiden perusteella.

Ohjatussa oppimisessa on käytettävissä opetusaineisto, jonka ominaisuuksien perusteella koneoppimisalgoritmi oppii toimintatavan, jota voi soveltaa vastaavissa, uusissa tilanteissa. Ohjatun oppimisen menetelmiä voidaan käyttää hyvin monenlaisissa tehtävissä, kunhan käytettävissä on riittävä opetusaineisto. Ohjatun oppimisen menetelmiä käytetään paljon päätöksenteon tukena, kuten päätettäessä, annetaanko henkilölle lainaa, tai arvioitaessa henkilön riskiä saada jonkin sairaus.

Koneoppimisen menetelmät voidaan jakaa perinteisiin ja *syväoppimismenetelmiin*. Syväoppimismenetelmät perustuvat neuroverkkojen käyttöön. Perinteisissä koneoppimismenetelmissä piirreirrotus ja piirteiden valinta ovat keskeisessä roolissa ja laajan ja monipuolisen aineiston yhteydessä tämä onkin vaativa vaihe.

Kun käytettävissä on riittävän laaja opetusaineisto, syväoppivat neuroverkot kykenevät erottamaan tehtävän kannalta merkitykselliset piirteet ilman, että ihmisen on määriteltävä, mitä piirteitä käytetään. Tämä on iso etu, vaikkakaan tällöin ei välttämättä selviä, minkä ominaisuuksien perusteella neuroverkko tekee päätelmänsä.

Viime vuosina on alettu enenevästi hyödyntää ns. *siirto-oppimista* (transfer learning). Tämä tarkoittaa, että otetaan käyttöön muulla kuin varsinaisen tehtävän aineistolla opetettu neuroverkko. Tällöin riittää, että neuroverkko vain viritetään haluttuun tehtävään tehtäväkohtaisella opetusaineistolla. Näin saadaan vähennettyä tapauskohtaisen opetusaineiston tarvetta. Esimerkiksi kuvien luokittelussa ja luonnollisen kielen käsittelyssä on otettu isoja edistysaskeleita siirto-oppimisen avulla (Gligic et al., 2020). Kuvien osalta tämä tarkoittaa, että pohjaksi otettava verkko on opetettu tunnistamaan kuvissa esiintyviä tyypillisiä muotoja, ja tapauskohtaisen opetusaineiston roolina on opettaa tehtävään liittyvät erityispiirteet. Luonnollisen kielen käsittelyn yhteydessä pohjamallille on opetettu kielen yleisrakenteet, minkä jälkeen neuroverkko voidaan virittää tapauskohtaisella opetusaineistolla käsillä olevaan tehtävään (Liu et al., 2019).

Vahvistusoppimisen erityispiirre on ratkaisun etsiminen kokeilemisen kautta. Kyse on siis yrityksen ja erehdyksen kautta oppimisesta. Palaute voi tulla välittömästi toimenpiteen tekemisen jälkeen tai tulla vasta hyvin monen toimenpiteen jälkeen, jolloin puhutaan viivästyneestä palautteesta. (Sutton & Barto, 2018)

Vahvistusoppimisen menetelmien kehittämisen ja soveltamisen vaativuus vaihtelee suuresti eri tapausten välillä. Vahvistusoppiminen edellyttää, että ratkaisijalle pystytään määrittelemään tavoite ja että ratkaisija pystyy keräämään kokemusta eri valintojen seurauksista. Eri vaihtoehtojen kokeilemisen kautta ratkaisija oppii vähitellen löytämään hyviä toimintatapoja.

Jos tarkasteltava tilanne koostuu vain yhdestä päätöksestä ja sen seurauksesta, voidaan hyödyntää ns. MAB-ratkaisijoita (Monikätisen rosvon ongelman ratkaisijoita). Jos vaiheita on useampia, niin yksittäisen päätöksen hyvyyttä ei voida päätellä välittömän palautteen perusteella, vaan sen perusteella, lisääkö päätös todennäköisyyttä saavuttaa haluttu lopputulos. (Sutton & Barto, 2018)

Vahvistusoppimisen pääkomponentit ja niiden roolit ovat seuraavat:

- Oppija tai ratkaisija (Agent) tarkoittaa vahvistusoppimisalgoritmia.
- Toimenpide (Action) on tarjolla oleva toimenpide. Vaihtoehtoisia toimenpiteitä on yleensä tarjolla useita, ja vahvistusoppimisella tavoitellaan kulloinkin parhaan toimenpidevaihtoehdon valintaa.
- Tila (State) kuvaa kulloisenkin päätöksentekotilanteen.
- Toimintapolitiikka (Policy) koostuu oppijan käyttäytymisen kunakin ajanhetkenä määrittelevistä säännöistä, eli logiikka, jolla oppija valitsee tiettynä hetkenä ja tietyissä olosuhteissa tehtävän toimenpiteen.
    - Toimintapolitiikka voi olla determinististä, yksinkertaisen funktion tai taulukon määrittelemää tai perustua laajaan laskentaan, jossa hyödynnetään esimerkiksi tilastollisia menetelmiä.
- Palkkio (signaali) (Reward) määrittelee vahvistusoppimisongelman tavoitteen. Joka toimenpiteen jälkeen ratkaisija saa ympäristöltä numeerisen palautteen, palkkion, ja ratkaisijan tehtävänä on maksimoida pitkällä tähtäimellä saatavien palkkioiden yhteenlaskettu määrä.
- Arvofunktio (Value function) antaa ennusteen siitä, miten hyviä kulloisessakin tilassa tarjolla olevat toimenpiteet ovat pitkällä tähtäimellä, eli miten suuren kumulatiivisen palkkion voidaan olettaa kertyvän kulloisestakin ajanhetkestä tulevaisuuteen. Arvon ennakoiminen on vaikea, mutta myös vahvistusoppimisen keskeisin tehtävä. Geneettiset algoritmit eivät käytä arvofunktiota vaan niissä kokeillaan useaa staattisista toimintapolitiikkaa, ja jatkoon valitaan kulloinkin parhaat vaihtoehdot ja parhaiden perusteella tehtävät muunnelmat. MAB-ratkaisijat, joita käytetään vain yhden päätöksen sisältävissä tehtävissä, eivät tarvitse arvofunktiota.
- Malli (Model) on ratkaisijan näkymä toimintaympäristöön; ratkaisija tekee mallin perusteella päätelmiä siitä, miten ympäristö tulee reagoimaan eri toimenpiteisiin. Mallia ei kaikissa tapauksissa määritellä, jolloin puhutaan mallittomista menetelmistä. Mallittomat menetelmät perustuvat kokonaan eri toimenpiteiden kokeilemiseen ja sen perusteella tehtävään toimenpiteiden keskinäisen paremmuuden vertailemiseen. On olemassa myös vahvistusoppimismenetelmiä, joissa samanaikaisesta sekä opitaan eri toimenpiteiden hyvyydestä että kerätään tietoa ympäristön mallintamiseksi.

Kaikkia yllä mainittuja osia ei siis kaikissa tapauksissa tarvita ja käytetä. Yksinkertaisimmillaan riittää, että määritellään toimintapolitiikka, jonka avulla valitaan tehtävä toimenpide. Vahvistusoppimisen keskeinen oletus on, että kaikki tavoitteet voidaan kuvata odotetun kumulatiivisen palkkion maksimoimisen muodossa.

3.2 Monikätisen rosvon ongelma ja sen ratkaisijat

3.2.1 Perusperiaatteet

Monikätisen rosvon ongelmalla (MAB = Multi-Armed Bandit problem) tarkoitetaan tilannetta, jossa on valittava kahden tai useamman vaihtoehtoisen toimenpiteen välillä epävarmuuden vallitessa ja palaute tehdyn toimenpiteen oikeellisuudesta saadaan välittömästi. (Sutton & Barto, 2018)

Ongelman nimi tulee tilanteesta, jossa on tarjolla monta, voittoprosentiltaan toisistaan poikkeavaa yksikätisistä rosvoa ja vain yksi vaihtoehto voidaan kerralla valita. On päätettävä, kannattaako vetää tähän mennessä parhaat voitot tuottaneesta vivusta, vai kokeilla, olisiko jokin muu vipu vielä parempi. Julkaisumaailmaan liittyviä esimerkkejä ovat esitettävien mainosten tai juttujen valitseminen: yleensä on käytettävissä enemmän juttuja tai mainoksia kuin mitä jokaiselle käyttäjälle voidaan esittää ja pitäisi pystyä valitsemaan ne, joihin käyttäjä todennäköisemmin reagoi toivotulla tavalla.

MAB-ratkaisijan käyttämän toimintapolitiikan määrittelyssä keskeistä on löytää hyvä tasapaino parhaan vaihtoehdon etsimisen ja jo kerätyn tiedon hyödyntämisen välillä. Mitä pitempään parasta ratkaisua etsitään (exploration), sitä varmemmin löydetään paras vaihtoehto, mutta sitä vähemmän aikaa jää parhaaksi tunnistetun ratkaisun hyödyntämiseen (exploitation) ja on tarjottu muita kuin parasta vaihtoehtoa. Tällöin ns. katumus (regret) on suuri, eli on menetetty hyötyjä käytettäessä muita kuin parasta toimenpidettä.

MAB-ratkaisualgoritmeja on useita ja niissä varioidaan mm. seuraavia asioita:

- Millaiset alkuarvot asetetaan?
- Miten iso osa kerroista etsitään parasta toimenpidettä eli mikä on exploration-toiminnan osuus. Päätettävä:
    - Tehdäänkö alussa jonkin aikaa vain etsintää?
    - Kuinka usein tehdään etsintää?
- Muutetaanko arvoja ajan kuluessa?
    - Oletetaanko, että olosuhteissa, esimerkiksi käyttäjissä, tapahtuu muutoksia?
        - Jos oletetaan, ettei muutoksia tapahdu, niin etsintään käytettävien kertojen (exploration-osuuden) osuutta voidaan ajan kuluessa pienentää.
        - Muutoksiin käyttäjissä voidaan reagoida myös painottamalla tuoreimpia havaintoja vanhoja havaintoja enemmän.
    - Epsilon greedy -algoritmi on tunnettu perusratkaisija, jossa alun tutkimisvaiheen (esim. 100 kertaa) jälkeen käytetään valitun kertoimen (epsilon) mukainen osuus kerroista tutkimiseen (esim. 5 %) ja muu aika (95 %) hyödynnetään kullakin hetkellä parasta tunnettua vaihtoehtoa.
- Millä perusteilla valitaan kokeiltava toimenpide:
    - Valitaanko toimenpide satunnaisesti niin, että kaikkia vaihtoehtoja tulee kokeiltua?
    - Painotetaanko parhaaksi oletetun toimenpidevaihtoehdon lähellä olevia vaihtoehtoja? Tällöin ajatuksena on, että tutkitaan vain lähellä parasta vaihtoehtoa olevia toimenpiteitä, eikä käytetä aikaa jo huonoksi todettujen vaihtoehtojen testaamiseen.
    - Otetaanko huomioon toimenpiteiden hyvyysarvioon liittyvä epävarmuus?
    - UCB, Upper-Confidence-Bound -menetelmä on esimerkki menetelmästä, joka laskee vaihtoehtojen hyvyydelle varmuuden ylärajan, jonka perusteella valitaan käytettävä vaihtoehto. Kun vaihtoehtojen hyvyys arvioidaan alkuvaiheessa yläkanttiin, eri vaihtoehdot tulevat varmimmin testattua.

Testauksen myötä kunkin vaihtoehdon hyvyysarvio lähentyy sen todellista arvoa eikä huonoimpia vaihtoehtoja enää myöhemmässä vaiheessa testata.

- Thompson sampling on bayesilaiseen heuristiikkaan perustuva menetelmä, jossa vaihtoehdon kokeilemisen todennäköisyys kasvaa sitä mukaan, mitä useammin se on antanut palkkion. Thompson sampling -menetelmä tuottaa usein hyviä tuloksia ja sitä suositellaankin käytettäväksi vertailutason antajana menetelmäkehityshankkeissa (Chapelle & Li, 2011).

Algoritmien toimintaeroja on havainnollistettu seuraavassa kuvassa (Kuva 2). Siinä on verrattu satunnaisvalinnan sekä UCB ja Thompson sampling -algoritmien antamaa tulosta simuloidussa aineistossa. UCB ja Thompson sampling -aloritmit ovat keskeiset perusmenetelmät ja niitä muunnoksineen on sovellettu monenlaisten ongelmien ratkaisemiseen.

Kuvan taustalla olevat ajot on tehty niin, että on katsottu kunkin algoritmin osalta, minkä vaihtoehdon algoritmi valitsee ja verrattu simuloituun aineistoon; jos aineistossa seuraavassa havainnossa on klikattu kyseistä vaihtoehtoa, niin palkkioksi tulee 1, muussa tapauksessa 0. Aineiston koko on kussakin tapauksessa 100 000 havaintoa. Vaihtoehtojen keskinäistä suosiota on varioitu seuraavasti: toisessa tapauksessa yksi kohteista on muita neljää suositumpi (24%, muut 19%) ja toisessa kolmen kohteen suosio on 22 % ja kahden 17 %.

Kuva 2. Satunnais-, UCB- ja Thompson sampling -algoritmien vertailu. Vasemmalla kohteen 4:n suosio on 24 % ja muiden 19 % klikkauksista, ja oikealla kohteiden 2,3 ja 4 suosio on 22% ja muiden 17%.

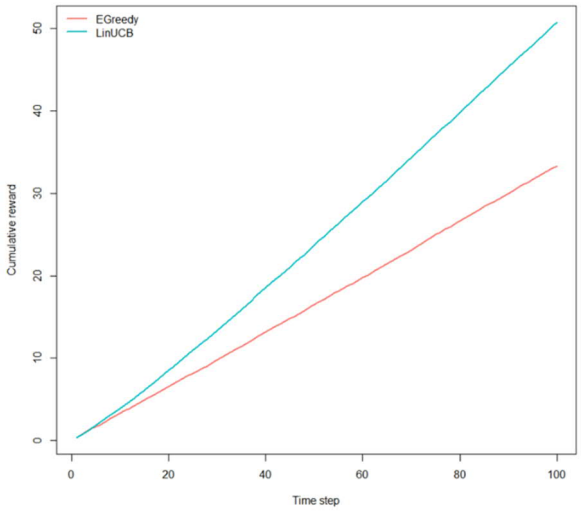
Nähdään, että algoritmien tuottama hyöty riippuu tilanteesta. Kun yksi vaihtoehdoista on selvästi muita suositumpi, kumpikin algoritmeista tuottaa selvästi satunnaisvalintaa paremman kokonaispalkkion (kuvan vasen puoli). Tilanteessa, jossa erot eivät ole niin selviä, algoritmien tuottaman hyöty jää pienemmäksi. Kummassakin tapauksessa Thompson sampling tuotti hieman paremman tuloksen kuin UCB. Thompson sampling -algoritmi päätyy nopeasti käyttämään parhaaksi tunnistettua vaihtoehtoa; jälkimmäisessä tapauksessa se on päätynyt suosimaan yhtä kolmesta, tasavahvasta vaihtoehdosta, kun UCB-algoritmi ohjaa käyttöä jakautumaan melko tasan kolmen suositumman vaihtoehdon kesken.

3.2.2 Kontekstuaaliset ratkaisijat

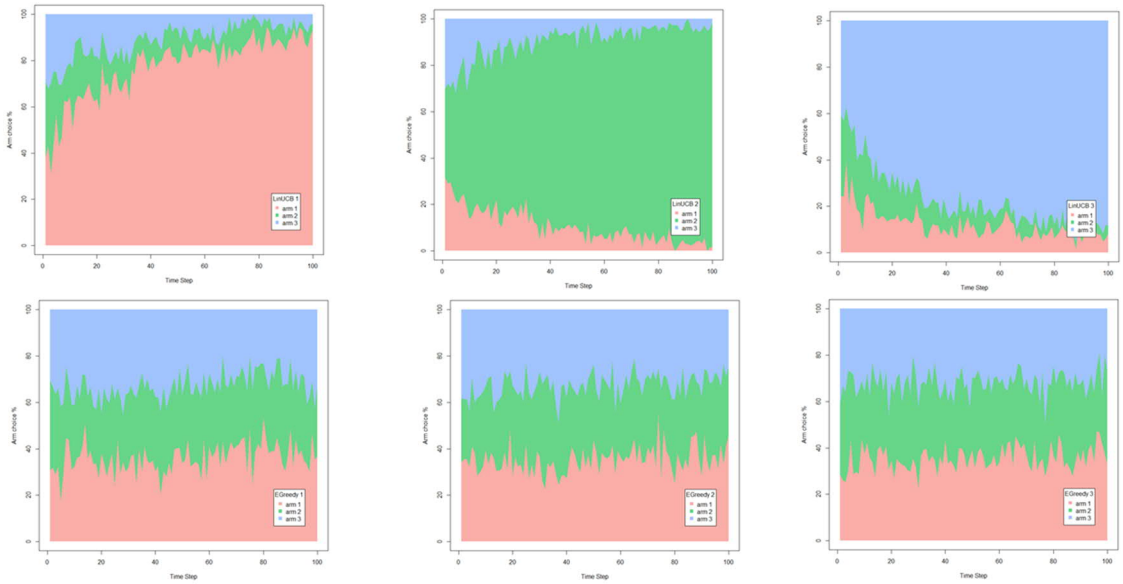
Edellä kuvatut menetelmät eivät ota lainkaan huomioon ympäristöstä saatavissa olevaa lisätietoa. Kun olosuhdetietoa on olemassa, voidaan käyttää ns. kontekstuaalisia MAB-ratkaisijoita (CMAB = Contextual Multi-Armed Bandits).

LinUCB (Li et al., 2010) on tunnettu esimerkki *kontekstuaalisesta* MAB-ratkaisijasta. Menetelmä kehitettiin Yahoo-palvelun etusivun ykkösartikkelin valitsemiseksi kulloinkin tarjolla olevista, 19 – 25 artikkelista. Artikkelista ja käyttäjistä oli kummistakin käytettävissä ominaisuustietoja, jotka ratkaisija ottaa huomioon. Ominaisuudet perustuivat juttukategorioihin: jutut oli luokiteltu viiteen, toisensa pois sulkevaan kategoriaan, ja käyttäjät kuvattiin suhteessa siihen, miten paljon he suosivat eri kategorioiden juttuja. Käyttäjät luokiteltiin ominaisuuksiensa perusteella erillisiksi ryhmiksi, joista kullakin on oma MAB-ratkaisija.

Kuvat 3 ja 4 havainnollistavat kontekstuaalisen LinUCB:n ja kontekstia huomioimattoman Epsion Greedy -algoritmien suorituskykyeroa tapauksessa, jossa on tarjolla vaihtoehtoihin liittyvää kontekstitietoa. Kuva 3 näyttää kummallakin menetelmällä saatavissa olevan kumulatiivisen palkkion, joka jää Epsion Greedy -algoritmia käytettäessä selvästi pienemmäksi kuin kontekstuaalista LinUCB:tä käytettäessä. Kuva 4 havainnollistaa, miten valinnat osuvat eri vaihtoehdoille. LinUCB-algoritmi oppii nopeasti tunnistamaan, mitä vaihtoehtoa kussakin kontekstissa kannattaa tarjota; Epsion Greedy toimii kaikissa tilanteissa samalla tavalla ja tarjoaa eri vaihtoehtoja tasaisin osuuksin. (van Emden & Kaptein, 2018)



Kuva 3. Simulointiesimerkki tilanteesta, jossa on kolme kontekstia ja kaksi ratkaisualgoritmia. Epsilon Greedy -algoritmi ei osaa hyödyntää kontekstitietoa toisin kuin LinUCB, minkä ansiosta LinUCB-algoritmilla saavutetaan korkeampi kumulatiivinen palkkio. (van Emden & Kaptein, 2018)



Kuva 4. Käsivarsien valinta eri algoritmien ja kontekstien yhteydessä. Ylärivin LinUCB-algoritmi ottaa kontekstin huomioon ja se oppii tarjoamaan kussakin kontekstissa parhaiden pärjäävää vaihtoehtoa (allekkain olevat kuvat kertovat samasta kontekstista). Epsilon Greedy-algoritmi valitsee käsivarret samalla tavalla kaikissa kolmessa eri kontekstissa (sama ajo kuin edellisessä kuvassa). (van Emden & Kaptein, 2018)

3.2.3 Sovellusalueita

Bouneffouf ja Rish (2019) ovat tuoreessa artikkelissaan tunnistaneet kiinnostavimmiksi MAB- ja CMAB -sovellusalueiksi seuraavat:

- Terveydenhoito: sovellettavan hoitomenetelmän valinta
- Rahoitus: sijoituskohteiden valinta
- Dynaaminen hinnoittelu
- Suosittelujen tekeminen
    - Tästä yksi esimerkki on edellä mainittu Yahoon ykkösartikkelin valinta (Li et al., 2010)
- Sosiaalisen median kampanjoiden vastaanottajien valinta
- Tiedonhaku
- Dialogisysteemien opettaminen
- Poikkeamien tunnistaminen
- Koneoppimismenetelmien valinta ja säätäminen.

Collier & Llorens (2018) käyttivät CMAB-menetelmää etsimään markkinointikirjeiden optimaalista lähettämisajankohtaa, jossa palkkio määräytyi sen mukaan, avasiko ja reagoiko vastaanottaja markkinointikirjeeseen.

CMAB-ratkaisijoiden soveltamisessa suositustehtävään keskeinen kysymys on, miten kohteet ja käyttäjät kannattaa ryhmitellä, jolloin CMAB-ratkaisijan mahdollisuudet suositella kiinnostavia kohteita paranevat (Li, Karatzoglu & Gentile, 2016).

Mattos, Bosch & Olsson (2019) keräsivät yrityshaastatteluilla kokemuksia ja haasteita, joihin MAB-ratkaisijoiden käytännön soveltamisessa oli törmätty. Heidän mukaansa nopeatahtisessa sisällön tai mainosten tarjoamisessa online-ympäristössä MAB-ratkaisijat ovat hyvä vaihtoehto, mutta pitempään voimassa olevia päätöksiä tehtäessä on vaarana, että MAB-ratkaisija ei valitse kokonaisuuden kannalta parasta vaihtoehtoa. AB-testauksella tutkitaan tilastollisesti luotettavalla tavalla eri vaihtoehtojen hyvyys, joten se on sopiva menetelmä pitkäaikaiseen käyttöön tulevien ratkaisujen valintatilanteeseen.

3.3 Täysmittainen vahvistusoppiminen

3.3.1 Menetelmiä

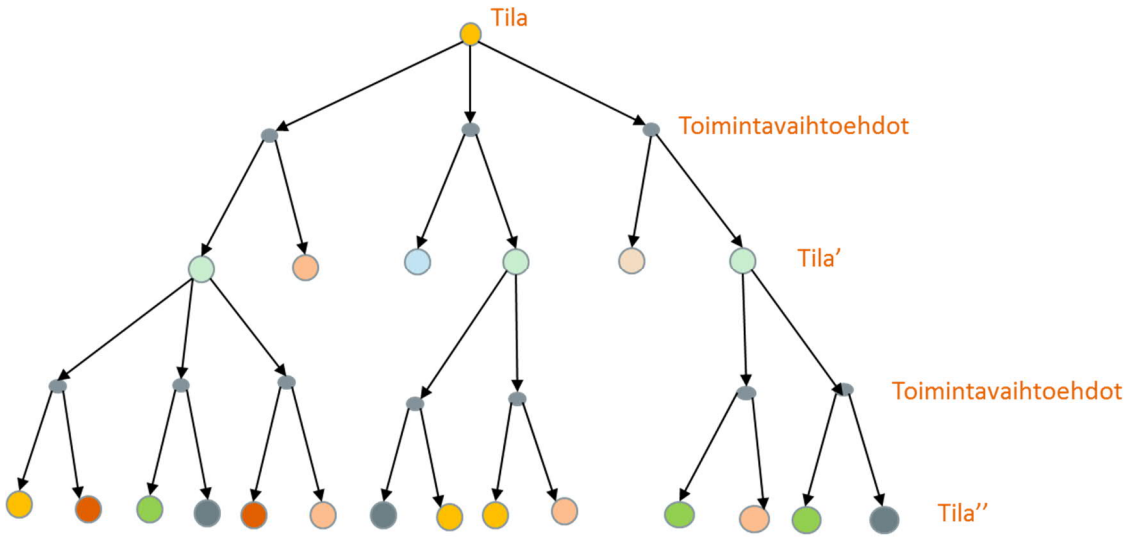
Edellä kuvattu monikätisen rosvon problematiikka on yksinkertaistettu tapaus vahvistusoppimisesta. Yksinkertaistus koskee palkkion saamista eli edellä kuvatuissa tapauksissa mahdollinen palkkio saadaan välittömästi valitun toimenpiteen tekemisen jälkeen. Usein tilanteet ovat kuitenkin sellaisia, että tavoite saavutetaan vasta usean päätöksentekovaiheen jälkeen. Tällaisen ongelman ratkaisemiseksi tarvitaan täysmittaista vahvistusoppimista. (Sutton & Barto, 2018)

Vahvistusoppimisessa palkkiot pyritään asettamaan niin, että ne ohjaisivat haluttuun tavoitteeseen pääsemistä mahdollisimman tehokkaasti. Esimerkiksi, jos tehtävänä on oppia reitti ulos labyrintista ja jokaisesta siirrosta saa negatiivisen palkkion, vaikkapa -0,5, ja poistumisesta 100, niin tämä ohjaa ratkaisijan hakemaan reitin, jota käyttäen labyrintista pääsee nopeasti ulos. Jokaisesta siirrosta tuleva negatiivinen palkkio ohjaa ratkaisijan minimoimaan siirtojen määrää eikä ratkaisija jää tarpeettomasti harhailemaan labyrinttiin.

Vahvistusoppimisen kannalta tehtävät jaotellaan episodisiin ja jatkuviin. Episodisilla tehtävillä on selkeä loppu, minkä jälkeen tehtävä aloitetaan alusta. Pelit ovat hyvä esimerkki tällaisesta tehtävästä. Jatkuvatoimisen prosessin ohjauksessa ei ole selkeää katkokohtaa ja sen jälkeen toistuvaa alkutilannetta. Vahvistusoppimista voidaan käyttää myös tällaisissa tapauksissa, mutta niin, että palkkioiden arvo diskontataan: kun arvofunktio arvioi tulevien palkkioiden kokonaismäärää, tulevaisuuden palkkiot muutetaan nykyarvoon kertomalla ne diskonttaustekijällä.

Arvofunktion määrittely ja sen arvon estimointi kussakin tilassa on vahvistusoppimisen keskeinen haaste. Yleensä ympäristöä ja toimenpiteiden seurauksia ei tunneta tarkkaan ja vaikka tunnettaisiinkin, vaatisi arvofunktioiden ratkaiseminen niin paljon laskentatehoa ja muistia, ettei sitä käytännössä pystytä tekemään. Käytännön ratkaisu on näytteisiin perustuvien menetelmien käyttö, puhutaan ns. Monte Carlo -menetelmistä.

Monte Carlo -menetelmillä arvioidaan arvofunktion arvoa tekemällä monia kokeiluita ja laskemalla niiden pohjalta keskiarvot. Jos käytössä on ympäristöä kuvaava malli, toimintapolitiikka voidaan määrittää tilojen arvojen perusteella, eli katsotaan vain yksi askel eteenpäin ja valitaan se toimenpide, joka tuottaa parhaan arvon, kun otetaan huomioon palkkio ja seuravan tilan arvo. Ilman mallia on arvioitava tila-toimenpide -parien arvoja.



Kuva 5. Kun tavoiteltu tila voidaan saavuttaa vasta usean päätöksentekovaiheen kautta, tarvitaan täysmittaisen vahvistusoppimisen menetelmiä. Rajatuissa tapauksissa voidaan tutkia kaikki tilat, mutta käytännössä tämä ei useinkaan ole mahdollista, jolloin pyritään tunnistamaan ja hyödyntämään lupaavimmat reitit.

Q Learning on vahvistusoppimisessa paljon käytetty algoritmiperhe, joka ei vaadi mallia ympäristöstä vaan joka oppii kokeilemalla eri vaihtoehtoja. Algoritmi käyttää Q- ja R-matriiseja. Q-matriisiin talletetaan ympäristöstä saatu tietämys. R-matriisin rivit edustavat eri tiloja ja sarakkeet palkkioita siirryttäessä tilasta toiseen. Q learningin yhteydessä voidaan käyttää jo aiemmin esillä ollutta Epsilon Greedy -algoritmia, jossa säädellään, miten paljon aikaa käytetään uusien vaihtoehtojen etsimiseen ja missä määrin hyödynnetään käytettävissä olevaa Q-matriisia. (Beysolow II, 2019)

Q Learning -menetelmän vahvuuksina voidaan pitää sitä, ettei se vaadi mallia ympäristöstä ja että siihen pohjautuvat päätökset ovat ymmärrettäviä ihmisille. Suurin heikkous on suuri laskentatarve, erityisesti jos ympäristö on laaja. Deep Q Learning hyödyntää syväoppivaa neuroverkkoa Q-taulukon arvojen approksimointiin, mikä tekee mahdolliseksi soveltaa Q Learning -menetelmiä myös moniulotteisiin tapauksiin. (Beysolow II, 2019)

Q Learning -menetelmillä on taipumus yliarvioida joidenkin toimenpiteiden arvoa, minkä välttämiseksi on kehitetty Double Deep Q Learning -menetelmä (van Hasselt et al., 2015). Double Deep Q Learning käyttää erillisiä neuroverkkoja toimenpiteen valintaan ja toimenpiteen hyvyyden arvioimiseen, kun perusmenetelmässä käytetään samaa neuroverkkoa molempiin tehtäviin.

3.3.2 Esimerkkejä

Vahvistusoppimista on sovellettu myös media-alan kannalta relevantteihin haasteisiin, mistä annetaan tässä kaksi esimerkkiä.

Zheng et al. (2018) ovat kehittäneet uutissivustolle suosittelijan, jonka tavoitteena on ottaa klikkausmäärien lisäksi huomioon palvelun käyttötiheys. Ratkaisijan saama palkkio siis riippuu kahdesta tekijästä: välittömästi tulevista klikkauksista ja viiveellä tulevasta palkkiosta sen mukaan, miten pian käyttäjä tulee uudelleen palveluun.

Suositusten antamiseksi ratkaisija saa käyttöön neljää piirrekokonaisuutta:

- Uutisten ominaisuudet kuten kategoria, uutislähde ja uutisen saamat klikkaukset eri aikajaksoilla, kuten 6 ja 24 tuntia (417 dimensiota).

- Käyttäjien ominaisuudet, jotka kertovat millaisiin uutisiin käyttäjien kiinnostus on kohdistunut (vastaava kuin uutisten ominaisuudet, mutta vain ne uutiset, joista käyttäjä on ollut kiinnostunut) (2065 dimensiota).

- Käyttäjän uutisinteraktio, joka kuvaa tarkemmin käyttäjän uutiskohtaisen interaktion (25 dimensiota), ja

- Konteksti, kuten viikonpäivä ja kellonaika, jolloin käyttäjä tulee palveluun (32 dimensiota).

Syvä Q-verkko mallintaa näiden piirteiden pohjalta, millä todennäköisyydellä käyttäjä tulee klikkaamaan mitäkin tarjolla olevaa uutista ja tämän pohjalta käyttäjälle laaditaan suosituslista. Tunnetun, hyvän ratkaisun hyödyntämisen ja paremman ratkaisun etsimisen keinona tässä käytetään kahta, toisiaan lähellä olevaa, mutta kuitenkin eri painoarvot sisältävää verkkoa. Verkkojen antamien ehdotusten hyvyyttä verrataan ja huonomman tuloksen antaneen verkon arvoja muutetaan kohti paremman tuloksen antaneen verkon arvoja.

Verkkojen arvoja päivitetään jokaisen käyttäjän jälkeen klikkaustapahtumien perusteella. Tämän lisäksi verkkoa päivitetään ajoittain, jolloin mukaan tulee myös käyttäjän aktiivisuus. Tämän ajoittaisen päivityksen tekemiseksi havainnot pidetään muistissa kyseisen ajan ja muistista otetaan näyte, jonka perusteella päivitys tehdään.

Chen et al. (2018) ovat soveltaneet vahvistusoppimista verkkokauppasivustolla tavoitteena antaa käyttäjille vinkkejä, jotka auttavat käyttäjää löytämään etsimänsä tuote mahdollisimman nopeasti. Kohteena olevalla sivustolla on olemassa 20 000 erilaista vinkkiä, joita voidaan tarjota käyttäjälle hakua tukemaan, ja tavoitteena on osata ehdottaa kullekin sivulle käyttäjälle hyödyllinen vinkki.

Tässä esimerkissä tila määräytyy käyttäjästä tiedossa olevien ominaisuuksien ja käyttäjän viimeaikaisen klikkaushistorian perusteella. Dimensioita on 156. Toimenpidevaihtoehtoja on, kuten edellä ilmeni, 20 000. Palkkiota tulee sitä enemmän mitä nopeammin käyttäjä klikkaa tarjottua vinkkiä ja luonnollisesti siitä, jos käyttäjä tekee tämän jälkeen ostoksen. Järjestelmä ottaa myös huomioon käyttäjän mieltymyksen vinkkien klikkaamiseen, eli jos käyttäjä klikkaa vinkkejä harvoin, niin silloin palkkio on suurempi kuin vinkkejä usein klikkaavan käyttäjän tapauksessa.

3.4 Vahvistusoppimisalgoritmien kehittäminen

Vahvistusoppimisen menetelmät on tarkoitettu hyödynnettäviksi dynaamisessa ympäristössä, mutta menetelmien kehittäminen suoraan tuotantoympäristössä on riskialtista ja epäkäytännöllistä. Tämä onkin tyypillinen vahvistusoppimismenetelmien kehittämiseen liittyvä haaste (Johansson & Mchome 2018). Tähän liittyy seuraavia kysymyksiä:

- Miten ottaa huomioon erot toimintapolitiikassa datan keruun yhteydessä ja uudessa tilanteessa?

- Paljonko dataa tarvitaan?

    - Riittääkö sama määrä dataa kaikkiin tilanteisiin, vai vaativatko eri algoritmit eri määrän dataa suorituskyvyn luotettavaan arvioitiin?

- Miten hyvin offline-kehityksen tulokset vastaavat todellista online-ympäristön suorituskykyä?

- Miten määriteltävissä olevat parametrit vaikuttavat menettelytapoihin ja tulosten siirrettävyyteen offline-ympäristöstä tuotantojärjestelmään?

Vaihtoehdot ovat simuloinnin käyttö ja muussa yhteydessä kerätyn aineiston hyödyntäminen. Simulointi voidaan tehdä eri tavoilla riippuen siitä, millaista ongelmaa ollaan ratkaisemassa. Yksi keskeinen ratkaistava asia on, miten määritellään seuraako toimenpiteestä palkkio vai ei. Yksi mahdollisuus on määritellä todennäköisyysjakauma, jonka perusteella päätetään, seuraako toimenpiteestä palkkio. Tällöin on tapana tehdä monia ajoja, joiden yhteistuloksen pohjalta arvioidaan menetelmän hyvyyttä. Kuvat 3 ja 4 ovat esimerkkejä tällaisesta simuloinnin käytöstä.

Yksinkertaisissa tapauksissa voidaan myös luoda kuvitteellista käyttödataa ja verrata, miten kehitettävä algoritmi olisi suoriutunut kyseisessä tehtävässä suhteessa käyttödataan. Kuva 2 on esimerkki tästä tapauksesta.

Jos riittävän hyvin todellisuutta kuvaava simulointimalli pystytään rakentamaan, simulointi on hyvä vaihtoehto eri algoritmien kehittämiseen ja testaamiseen ennen todelliseen käyttöympäristöön viemistä.

Myös todellisesta järjestelmästä kerättyä dataa voidaan käyttää. Koska tämä data on syntynyt olosuhteissa, joissa ei ole käytetty testattavana olevaa toimintapolitiikkaa, on päätettävä, miten aineistoa käytetään, jotta se kuvaisi tutkittavana olevan uuden toimintapolitiikan suorituskykyä. Tätä ongelmaa kutsutaan nimellä off-policy evaluation -ongelmaksi.

Li et al. (2010) ovat esitelleet LinUBC-menetelmän esittelevässä paperissa menettelytavan, jolla muuta tarkoitusta varten kerättyä aineistoa voi käyttää niin, että aineisto edustaa online-käyttötilannetta. Menettelytapa on seuraava: muuta tarkoitusta varten kerätystä tai simuloidusta data-aineistosta otetaan mukaan vain ne havainnot, joissa toteutunut toimenpide on sama kuin jonka arvioitavana oleva algoritmi olisi valinnut. Mukaan otetaan siis vain tapaukset, joissa testattava toimintapolitiikka valitsee saman käsivarren kuin mikä aineistossa on valittu. Muussa tapauksessa sekä näyte että politiikan mukainen valinta hylätään. Kerätyn lokidatan tulisi koostua datasta, jossa tapahtuvat ovat toisistaan riippumattomia ja eri vaihtoehtojen näyttäminen on sattumanvaraista. Jälkimmäinen vaatimus ei kuitenkaan ole täysin ehdoton. Tämä menettelytapa johtaa käytännössä usein siihen, että iso osa havainnoista ei täytä ehtoa ja joudutaan hylkäämään, joten aineistoa on oltava paljon, jotta kelpuutettavia havaintoja saadaan tarpeeksi.

4. Etusivun järjestyksen vaikutus juttujen suosioon

4.1 Aineisto ja sen kuvailu

Juttujen järjestyksen vaikutuksen tutkimiseksi koottiin aineisto, joka koostui kolmesta pääosasta:

1. Sivun rakennetta kuvaavat tiedot, eli missä järjestyksessä jutut ovat etusivulla,

2. Juttuihin etusivun kautta tulevien klikkausten määrät, ja

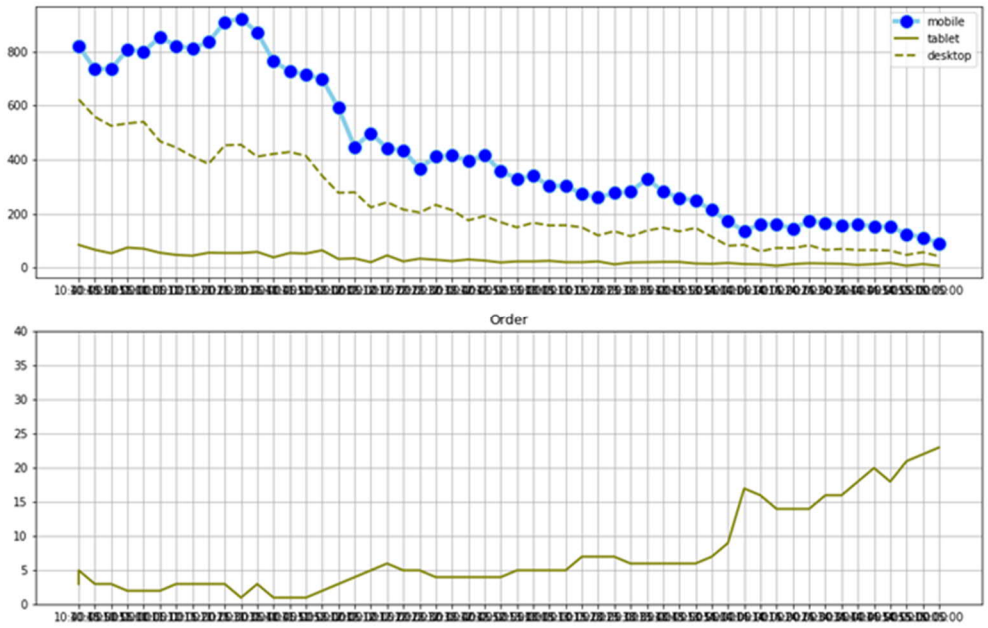
3. Juttujen etusivulla näkyvä otsikko ja mahdollinen lisäteksti sekä juttujen osastoluokitukset.

Aineisto kerättiin kahdelta viikon mittaiselta seurantajaksolta niin, että toinen jakso oli kesäkuussa ja toinen syyskuussa 2019. Tiedot etusivun rakenteesta ja klikkausmääristä kerättiin viiden minuutin välein.

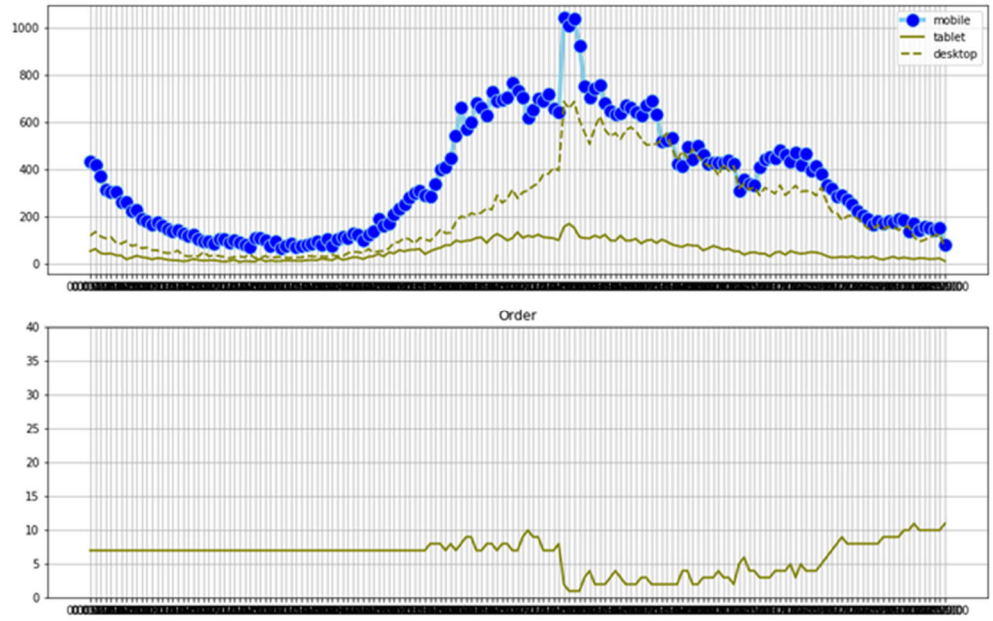
Yksittäisen jutun herättämän kiinnostuksen ja sen saamien klikkausten määrään vaikuttaa paitsi juttu itse ja sen sijainti, myös se, mitä muita juttuja on samaan aikaan tarjolla joko muista tai samasta aiheesta. Tämän kuvaamiseksi johdettiin käytettävissä olevien tietojen pohjalta etusivujen rakennetta kuvaavia muuttujia, kuten esimerkiksi, onko seuraava juttu samaa kategoriaa, paljonko etusivulla on saman kategorian juttuja, ja mikä on kunakin ajanhetkenä etusivulla olevien juttujen mediaani-ikä. Juttujen etusivulla näkyvistä otsikoista laskettiin tunnuslukuja kuten otsikossa sanojen ja isojen kirjaimien lukumäärät.

Seuraavassa annetaan joitakin esimerkkejä kerätystä datasta tehdyistä havainnoista. Kuva 6 näyttää esimerkin jutusta, joka ilmestyi 10:40 ja oli etusivulla iltapäivällä kello 15:05 asti. Jutun elinkaari vastaa ennakko-odotusta, eli juttu kiinnostaa noin ensimmäisten kahden tunnin ajan, mutta sitten kiinnostus vähenee ja juttu painuu etusivulla alemmas uusien juttujen ilmestymisen myötä.

Kuva 7 esittää tapausta, jossa juttu ilmestyy keskiyöllä ja saavuttaa suurimmat klikkausmäärät vasta usean tunnin kuluttua aamun tultua. Jutun nosto kärkeen tuottaa hetkellisen nousun myös klikkausmäärissä, mutta pian tämän jälkeen klikkausmäärät tasoittuvat ja kääntyvät laskuun.

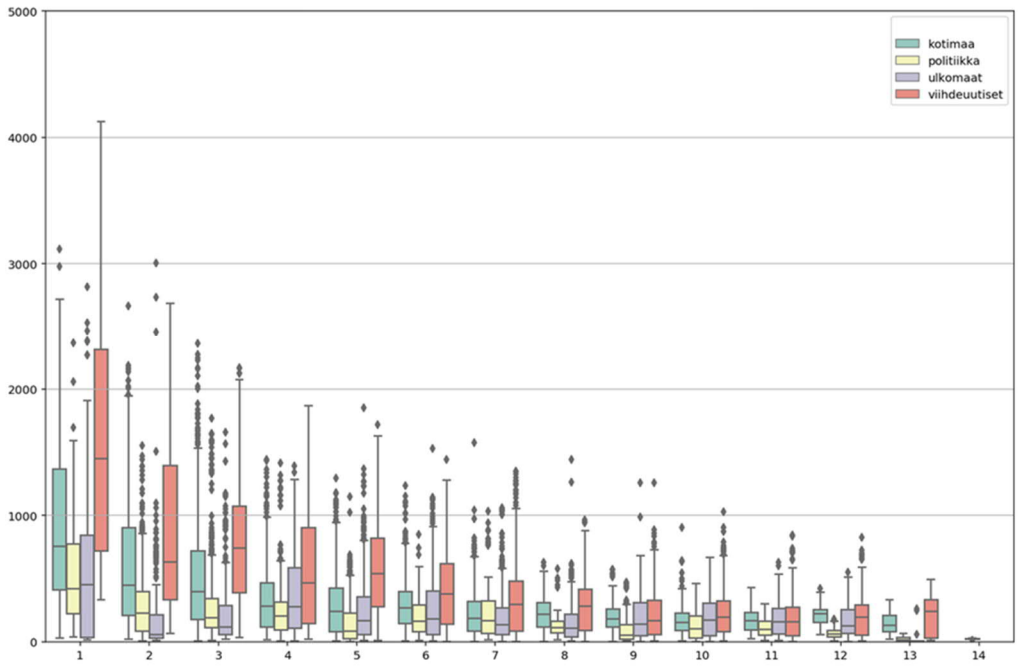


Kuva 6. Esimerkki jutun klikkausmääristä viittä minuuttia kohti (ylempi kaavio) ja jutun sijainti etusivulla (alempi kaavio). Juttu ilmaantui etusivulle aamupäivällä 10:40.



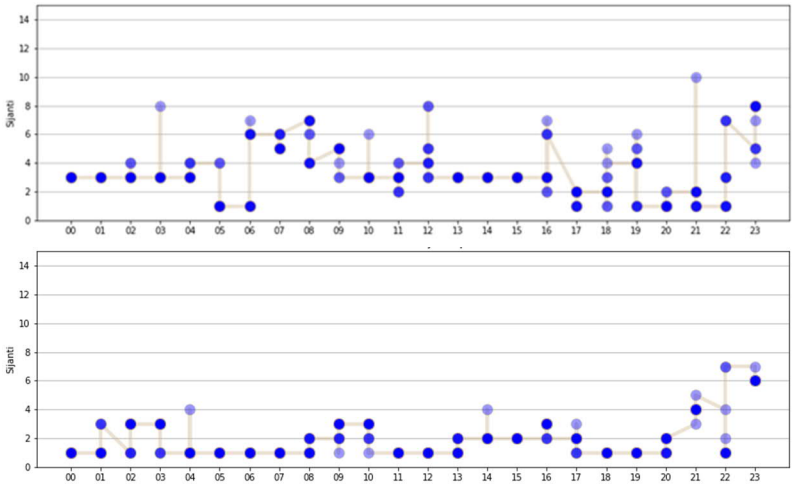
Kuva 7. Esimerkki jutun klikkausmääristä viittä minuuttia kohti ja sen sijainti etusivulla; juttu on julkaistu keskiyön vaiheilla, joten kiinnostus juttua kohtaan kasvaa vasta aamun tultua.

Edelliset kaksi kuvaa kertoivat tilanteesta yksittäisten juttujen osalta. Kuva 8 näyttää etusivun kautta tulleiden, viiden minuutin jaksojen juttukohtaisten klikkausmäärien jakaumat suhteessa juttujen sijaintiin etusivulla viikon mittaisessa aineistossa. Nähdään, että sivulla ensimmäiseksi olevat jutut saavat enemmän klikkauksia kuin alempana sivustolla olevat. Nähdään myös, että juttujen saamat klikkausmäärät riippuvat sijainnin lisäksi myös niiden osastosta. Viihde- ja kotimaan uutiset kiinnostavat keskimäärin selvästi enemmän kuin politiikan tai ulkomaan uutiset.



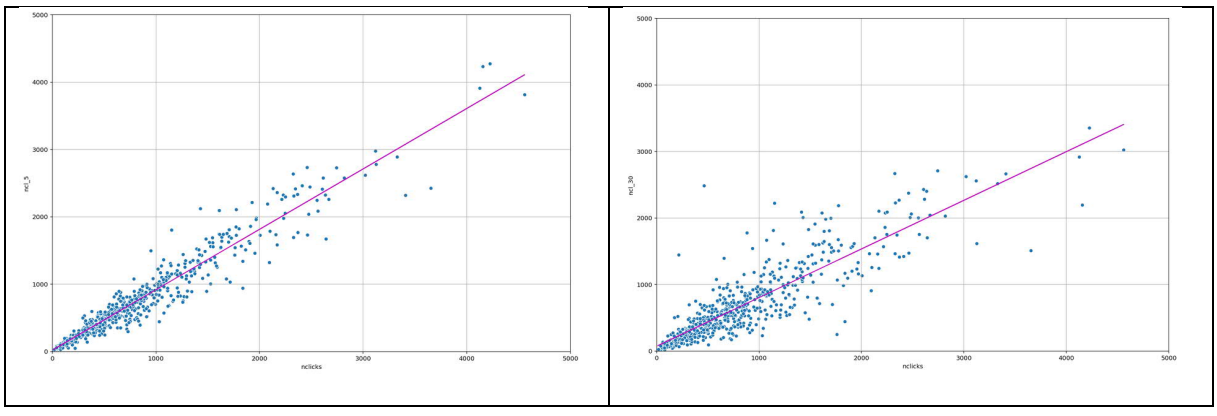
Kuva 8. Etusivun kautta viidessä minuutissa tulleiden juttukohtaisten klikkausmäärien jakaumat yhden viikon jaksolla neljän osaston osalta.

Etusivun alkuun pyritään sijoittamaan kulloinkin kiinnostavimmiksi arvioidut jutut. Kuva 9 tarjoaa kaksi esimerkkiä siitä, millä paikalla kunakin 5 minuutin jaksona eniten klikkauksia saanut juttu on ollut kahden esimerkkipäivän aikana. Alemmassa kuviossa suosituin juttu on lähes aina ollut sivulla ensimmäisenä tai toisena. Ylemmässä on sitä vastoin ollut enemmän hajontaa: useasti suosituin juttu on sijoilla neljännestä kahdeksanteen.



Kuva 9. Kaksi esimerkkiä kunkin viiden minuutin jakson aikana eniten klikkauksia saaneiden juttujen sijainnista. Tuntia kohti on 12 havaintoa ja mitä useampi havainto sijoittuu samaan kohtaan, sitä tummempi piste paikassa on.

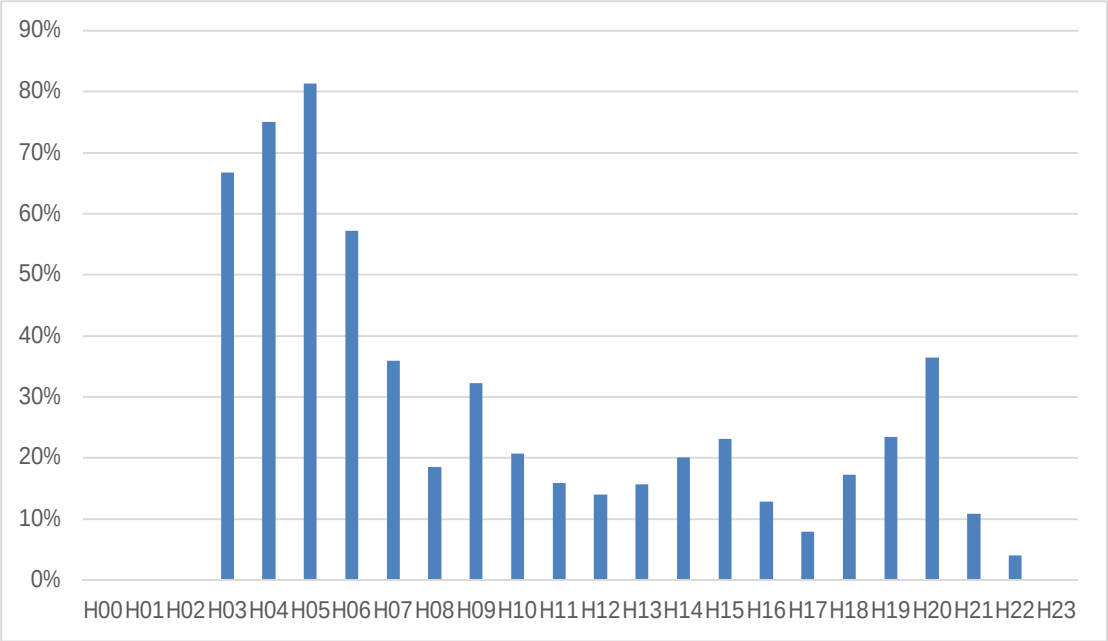
Kuva 10, vasen puoli esittää klikkausmäärät ajan hetkeä 0 edeltävien 5 minuutin aikana (x-akseli) ja sitä seuraavien viiden aikana (y-akseli). Oikealla puolella on sama x-akseli, mutta y-akseli kertoo klikkaukset ajan hetkenä 25-30 minuuttia. Nähdään, että havaintojen välillä on suhteellisen lineaarinen riippuvuus, mutta mitä pitempi havaintojen välinen aikaväli on, sitä enemmän havainnoissa on hajontaa. Nähdään myös, että klikkausmäärissä on laskeva tendenssi. Toteutuneet klikkausmäärät siis ennustavat selvästi tulevien klikkausten määrää, mutta ennustavuus heikkenee aikavälin pidentyessä.



Kuva 10. Toteutuneet klikkausmäärät ennustavat tulevien klikkausten määrää. Kummassakin kuvassa on x-akselina ovat klikkaukset ajan hetkeä 0 edeltäneiden 5 minuutin aikana; vasemmanpuoleisessa kuvassa y-akselina ovat klikkaukset seuraavien 5 minuutin kuluessa ja oikeanpuoleisessa klikkaukset 25 - 30 minuutin kuluttua nollapisteestä.

Koko aineistossa 37 % artikkeleista sai enemmän klikkauksia 5 minuutin kohdalla kuin ajanhetkellä 0; 30 minuutin kohdalla tällaisia artikkeleita oli 27 % prosenttia ja 60 minuutin kohdalla 22% prosenttia. Kuva 11 näyttää, miten suuri osa eri aikoina julkaistuista jutuista

sai 60 minuutin kuluttua enemmän klikkauksia kuin alussa. Nähdään, että jutun julkaisuajalla on suuri vaikutus: varhain aamulla julkaistujen juttujen huomio huipentuu vasta joidenkin tuntien kuluttua sivuston liikennemäärien kasvaessa. Päivällä julkaistuista jutuista vain 10 - 20% herättää pitempiaikaista kiinnostusta.



Kuva 11. Niiden, kunakin tuntina julkaistujen juttujen osuus, joiden klikkausmäärä aikavälillä 55 - 60 minuuttia oli suurempi kuin ensimmäisten viiden minuutin aikana.

4.2 Klikkausmäärän ennustaminen paikan suhteen

Artikkelin ja sen sijainnista etusivulla johdettujen ominaisuuksien perusteella kehitettiin koneoppimismalli, jolla voi ennustaa artikkeliin etusivun kautta tulevia klikkausmääriä eri sijainneissa ja eri ominaisuuksien vaikutusta klikkausmäärään.

Opetusaineistona mallille olivat täyden viikon jaksolta (kesäkuu 2019) kerätyt tiedot etusivun rakenteesta ja artikkeleiden toteutuneista klikkausmääristä. Klikkausmäärät kerättiin vakioaikavälein (viisi minuuttia). Aika-askeleella etusivun rakenteen ja artikkeleiden sijaintien oletetaan pysyvän suurin piirtein vakiona.

Koneoppimista varten datasta johdetut ominaisuudet (piirteet) liitettiin toteutuneeseen klikkausmäärään jokaisella aika-askeleella. Toiset ominaisuuksista olivat staattisia eli artikkelin pysyviä ominaisuuksia kuten kategoria ja toiset ajan suhteen muuttuvia kuten jutun ikä ja sijainti etusivulla. Kunkin artikkelin edellisinä aika-askeleina saamat klikkausmäärät tulivat mukaan opetusaineistoon. Artikkelin saama klikkausmäärä tietyllä aika-askeleella on suora vahvistus/palaute artikkelin kiinnostavuudesta ja sen sijainnin hyvyydestä.

Koneoppimismallin luonnissa käytettiin Gradient Boosting Ensemble -koneoppimismenetelmää (Hastie et al., 2009), joka soveltuu erittäin hyvin myös opetusdatan ominaisuuksien vaikutuksen arvioimiseen lopputulokseen.

Kehitetty koneoppimismalli ennustaa seuraavan aika-askeleen klikkausmääriä. Mallia voi käyttää myös iteratiivisesti ennustukseen useamman aika-askeleen päähän, jolloin ajanhetkeen ennustettu klikkausmäärä toimii syötteenä seuraavan ajanhetken ennusteelle.

Taulukko 1 kuvaa yhdessä esimerkkitapauksessa kahden palstan juttujen yhden aika-askeleen aikana toteutuneita ja ennustettuja klikkauksia. Juttujen sijainti esitetään suhteellisena etäisyyslukuna sivun yläreunasta (taulukossa luvut 16, 32, …, 222).

Taulukko 1. Ennustetut ja toteutuneet klikkausmäärät artikkelin paikan mukaan.

Artikkelin paikka	***16***	***32***	***48***	***82***	***98***	***128***	***142***	***170***	***222***	***Yh-teensä***
Toteutunut	2576	1439	142	699	333	790	621	362	261	7223
Ennustettu	2581	1309	174	707	337	764	558	368	262	7064
Toteutunut - ennustettu	-5	130	-32	-8	-4	26	63	-6	-1	159
Suhteellinen virhe (%)	-0,2	9,0	-23,1	-1,3	-1,3	3,3	10,0	-1,7	-0,5	

Taulukosta havaitaan, että artikkeleiden kokonaisklikkausmääräennusteen (7064 klikkausta) virhe suhteessa toteutuneeseen klikkausmäärään (7223 klikkausta) on esimerkissä noin 2,2% (159 klikkausta). Yksittäisten artikkeleiden osalta virhe klikkausmäärien ennusteissa on 1 - 130 klikkausta ja suhteellisen virheen mediaani on 1,7%.

Seuraavassa taulukossa esitetään saman esimerkin avulla artikkelin paikan vaikutusta ennustettuihin klikkausmääriin. Laskelmat on tehty niin, että artikkelin korkeusluvuksi asetetaan vuorotellen luku väliltä 16-222 (9 eri paikkaa) ja lasketaan koneoppimismallin avulla klikkausmääräennuste. Taulukon sarakkeissa listataan alkuperäiset ja riveissä laskennalliset paikat. Lihavoidulla fontilla on korostettu alkuperäisessä paikassa toteutunut klikkausmäärä. Esimerkiksi alun perin paikassa 16 (2581 klikkausta) ollut artikkeli saa mallin mukaan 2611 klikkausta paikassa 32, 2528 klikkausta paikassa 48 ja 2405 klikkausta paikassa 222 (ks. rivit). Kun taasen alun perin paikassa 222 olleen artikkelin ennustetaan

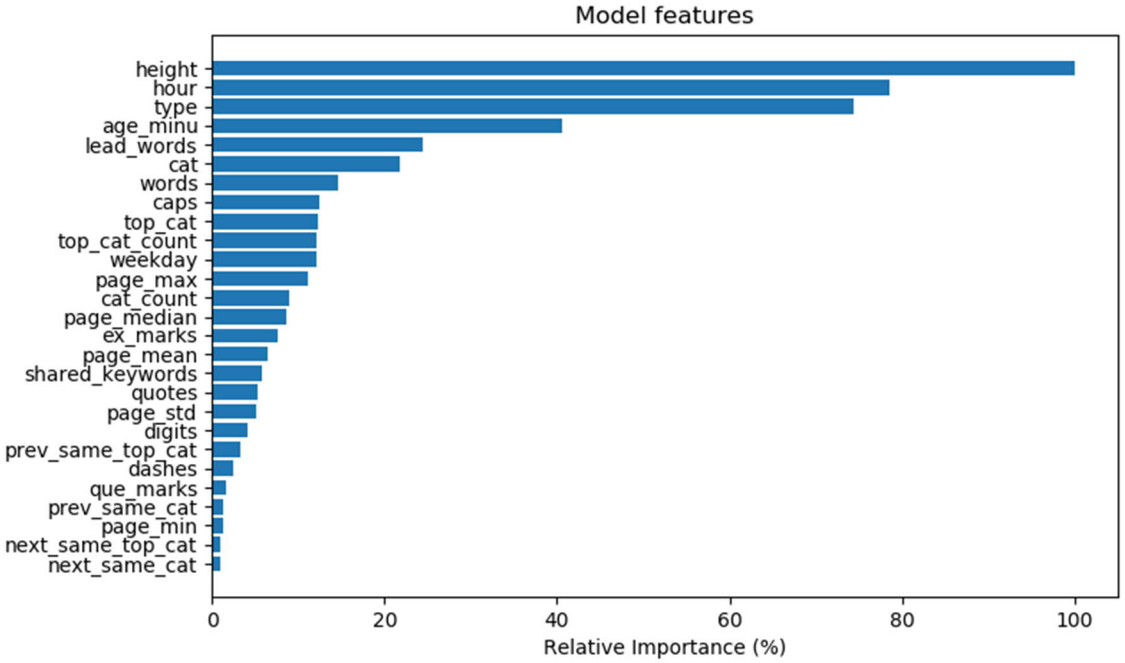
Taulukko 2. Artikkelin paikan vaikutus ennustettuun klikkausten määrään.

artikkelin paikka	***16***	***32***	***48***	***82***	***98***	***128***	***142***	***170***	***222***	***Yhteensä***
16	**2581**	1360	188	832	635	1213	811	630	585	8840
32	2611	**1309**	181	783	571	1072	711	571	447	8260 (93%)
48	2528	1287	**174**	728	520	999	653	503	378	7774 (88%)
82	2462	1227	159	**707**	344	832	616	465	362	7179 (81%)
98	2453	1219	166	701	**337**	804	587	436	357	7064 (80%)
128	2407	1157	121	674	324	**764**	560	393	325	6728 (76%)
142	2410	1185	124	674	322	762	**558**	391	319	6750 (76%)
170	2410	1165	113	747	303	742	550	**368**	291	6693 (76%)
222	2405	1163	77	862	273	719	520	354	**262**	6640 (75%)

saavan 585 klikkausta paikassa 16. Tässä esimerkissä paikassa 48 on ollut juttu, joka on saanut huomattavan vähän klikkauksia hyvästä sijainnista huolimatta. Tämän jutun siirtäminen alemmas olisi mahdollistanut korkeamman kokonaisklikkausmäärän, kun sen jälkeen olleet jutut olisi voitu sijoittaa yhtä paikkaa ylemmäs.

Jokaisella rivillä on myös yhteenlaskettu klikkausmäärä ja miten monta prosenttia se on parhaan sijainnin arvosta; nämä antavat suuntaa sijainnin vaikutuksesta. Nähdään, että etusivun alkuosassa pudotus klikkausmäärissä on selvä, mutta alempana eroa ei juurikaan ole. Tämä on linjassa yllä esitettyjen, sijaintikohtaisten klikkausmäärien jakaumien kanssa (Kuva 8).

Artikkelin sijainnin vaikutuksen arvioimisen lisäksi opetettua mallia pystyy käyttämään opetuksessa käytettyjen ominaisuuksien vaikutuksen arvioimiseen. Kuva 12 havainnollistaa koneoppimismallin opetuksessa käytettyjen valittujen ominaisuuksien suhteellista merkitsevyyttä mallin pohjalta tehtävään klikkausmäärän ennusteeseen, kun edellistä klikkausmäärää ei oteta huomioon. Ominaisuuden merkitsevyydellä (feature importance) tarkoitetaan tässä sitä, miten suuri vaikutus kyseisen ominaisuuden arvon muutoksella on lopputulokseen opetetun koneoppimismallin mukaan.



Kuva 12. Eri ominaisuuksien suhteellinen merkitsevyys klikkausmäärän ennusteeseen opetetussa koneoppimismallissa.

Artikkelin sijainnin etäisyysarvo yläreunasta (height) saa suurimman suhteellisen merkitysarvon (100%), mutta myös esimerkiksi vuorokauden tunti (hour), artikkelin tyyppi (yhden tai kahden palstan juttu), artikkelin ikä minuutteina (julkaisuajankohdasta), ingressissä näkyvien sanojen lukumäärä (lead_words), artikkelin kategoria (cat) ja saman pääkategorian juttujen määrä näytettävällä sivulla (top_cat_count) ovat melko merkitseviä ominaisuuksia. Ominaisuuksien vaikutussuunta voi olla positiivinen tai negatiivinen, mistä yllä esitetty kuva ei anna tietoa.

Tässä esitetyn analyysin ja kehitetyn mallin lähtötietona olivat toimituksen etusivulle sijoittamien juttujen etusivun kautta keräämien klikkausten määrät, juttujen sijaintitiedot ja joukko juttua ja sen otsikkoa kuvaavia metatietoja. Kehitetty koneoppimismalli hyödyntää oleellisena tietona aiemmin havaittua klikkausmäärää ja sitä voidaan näin pitää vahvistusoppimisen sovelluksena. Malli sisältää paikan vaikutuksen, joten mallia

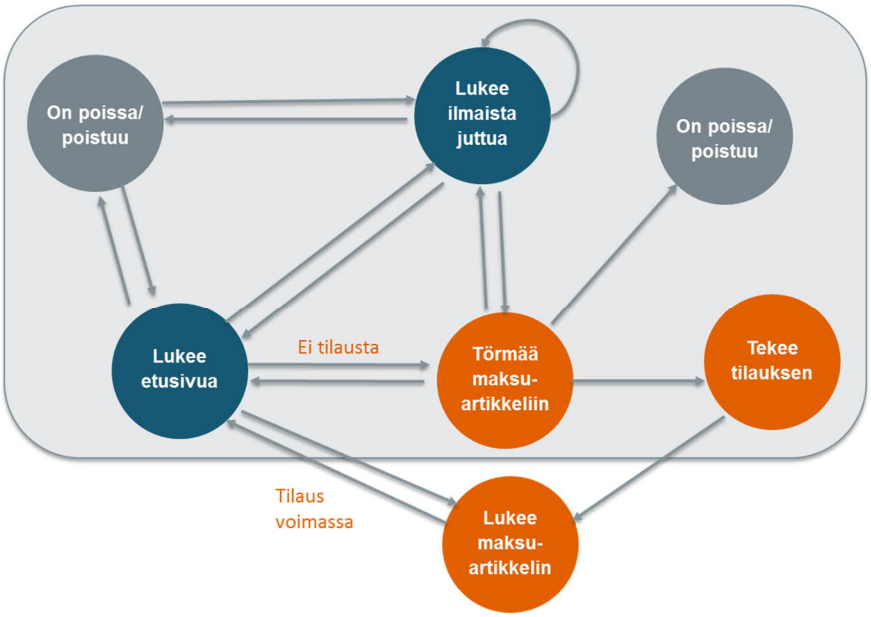
käytettäessä ei ole tarpeen testata jutun sijainnin vaikutusta monessa eri paikassa, kuten monikätisen rosvon ratkaisualgoritmit tekevät, vaan päätelmä sopivasta sijaintipaikasta voidaan tehdä yhdestä kohdasta saadun havainnon perusteella. Malli ei kuitenkaan pysty antamaan ehdotusta siitä, mille kohdalle uusi artikkeli tulisi sijoittaa, vaan mallia voi käyttää vasta kun on saatu havainto artikkelin tietyssä paikassa saamasta klikkausmäärästä. Jos käyttäjistä on taustatietoa, malli voidaan opettaa kullekin käyttäjäryhmälle erikseen, jolloin etusivun järjestystä voitaisiin optimoida myös käyttäjäryhmittäin.

5. Vahvistusoppimismalli tilausten tekemisen edistämiseksi

5.1 Mallin lähtöoletukset ja ohjelmiston kehittäminen

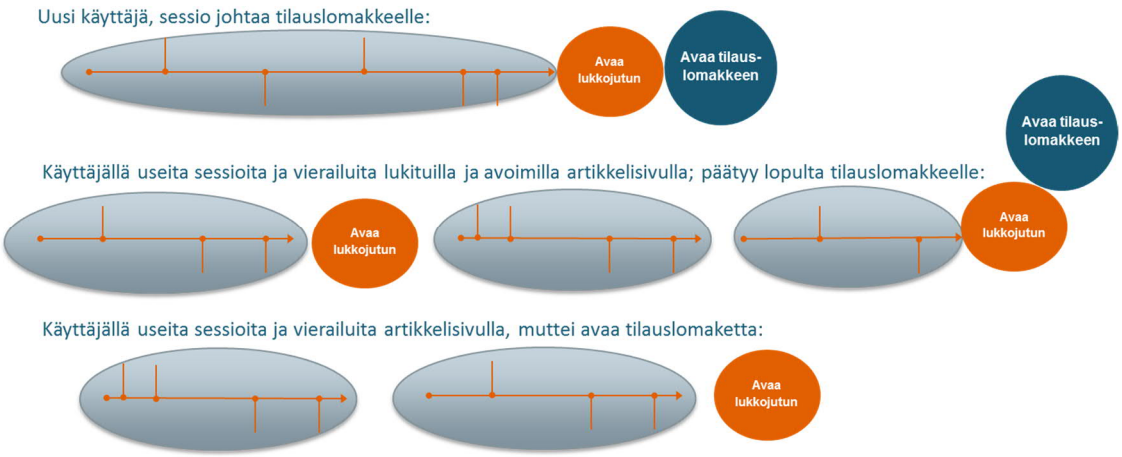
Toisessa esimerkkikohteessa kehitettiin vahvistusoppimiseen pohjautuva ohjelmisto, jonka avulla voidaan tuottaa suositus siitä, mitkä artikkelit kannattaa esittää, kun päätavoitteena on tilausten tekemisen edistäminen. Ohjelmiston kehittämiseen johtaneiden keskustelujen teemana oli oletus, että esittämällä käyttäjiä kiinnostavaa sisältöä sopivana yhdistelmänä ilmaisia ja tilauksen vaativia sisältöjä, käyttäjiä saadaan innostettua käyttämään palvelua ja vähitellen myös tekemään tilaus (kuva 13). Ohjelmiston on sisällytetty vain ei-tilaajat ja ainoa huomioon otettava kriteeri on tilausten edistäminen ohjelmistossa tehtyjen oletusten ja periaatteiden pohjalta.

Ohjelmistoon sisältyy data-aineiston generointi, eli ei tarvita todellisesta tuotantoympäristöstä kerättyä dataa. Simuloidun datan käytöllä on kaksi vahvaa etua: aineisto saadaan luotua nopeasti ja dataan voi sisältyä tietoja, joiden keräämistä ei ole käytännössä toteutettu. Haasteena on saada generoitu data ja käyttäytyminen vastaamaan riittävän tarkasti todellista dataa ja käyttäytymistä. Simulaation avulla voidaan joka tapauksessa saada suuntaviivaa sille, minkä tietojen keräämistä kannattaa kehittää.



Kuva 13. Käyttäjän eteneminen tilaukseen.

Vahvistusoppimisongelmaa tutkittiin lähtemällä käyttäjän sessioista (ks. kuva 14). Käyttäjä voi päätyä tilauslomakkeelle erilaisten sessioiden kautta ja tilauslomakkeen avaamista voi edeltää useampia käyntejä lukituilla ja avoimilla artikkelisivuilla. Tavoitteena siis on koneoppimisen keinoin löytää tilaukseen päättyneistä sessioista ja liittyvästä datasta samankaltaisuuksia ja riippuvuuksia, sekä sitten hyödyntää vahvistusopetettua mallia sopivien artikkeleiden näyttämisessä käyttäjälle.



Kuva 14. Esimerkkejä etenemispolusta tilaukseen.

Mallin luontiin ja hyödyntämiseen liittyi seuraavia vaiheita:

- Datan generointi koneoppimista varten: Käyttäjät, artikkelit, sessiot (luku 5.1.1)
- Mallin opetus testidatalla (luku 5.1,2 ja 5.1.3)
- Mallin inkrementaalinen päivitys saadun palautteen (palkkion) perusteella (luku 5.1.3)
- Mallin validointi testidatalla (luku 5.1.4)
- Mallin hyödyntäminen artikkelien valitsemiseksi sessiokohtaisesti (luku 5.2)

5.1.1 Testidatan generointi

Esimerkkiä varten generoitiin ohjelman avulla kuvitteelliselle lehdelle yhden päivän artikkelit, käyttäjät ja käyttäjien sessiot.

Artikkeleita generoitiin kaikkiaan 100, joista puolet oli vain tilaajien luettavissa (ns. lukkojuttuja) ja loput kaikille avoimia. Lisäksi jokaiselle artikkelille arvottiin kategoria (1-5) ja ”hot”-alkuarvo (0.0-1.0), joka kuvaa oletettavaa jutun kiinnostavuutta,

Simuloitua mallia varten generoitiin yhteensä 10000 käyttäjää ja jokaiselle käyttäjälle luotiin sessio, jolle arvottiin näytetyt ja klikatut (lukitut ja avoimet) artikkelit sekä lisäksi liitettiin tieto siitä, päättyikö sessio tilaukseen.

5.1.2 Opetusaineisto

Opetusaineisto koostui edellä kuvatusta generoidusta datasta ja siitä johdetuista artikkeli-, sessio- ja käyttäjäkohtaisista piirteistä (Taulukko 4):

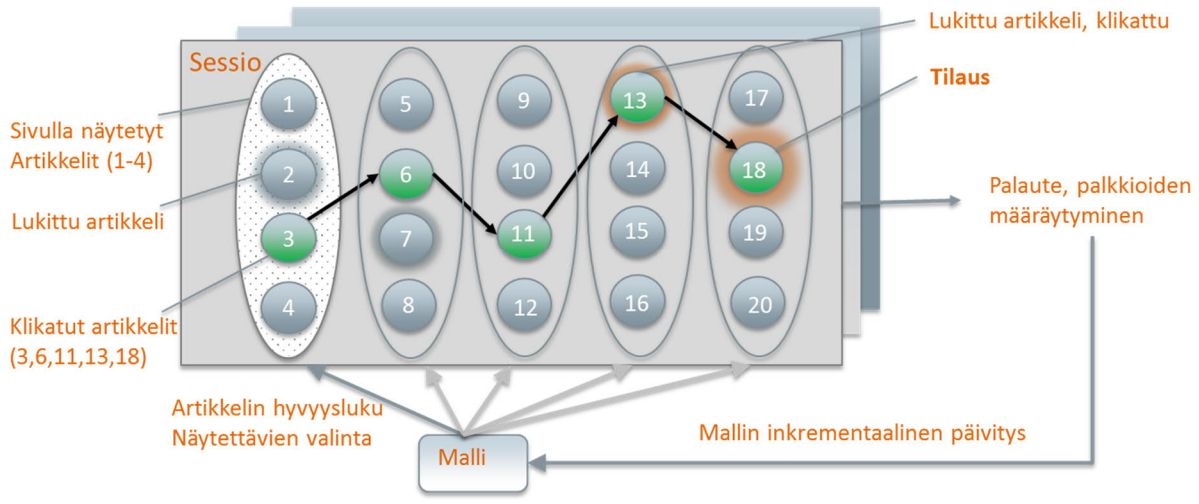
Taulukko 4. Opetusaineiston artikkeli-, sessio- ja käyttäjäkohtaiset piirteet

Artikkeli	▪ Kategoria ▪ Artikkelin ”hot” -alkuarvo, joka on artikkelille annettu kiinteä alkuarvo ▪ Artikkelin dynaaminen ”hot” -arvo, joka määräytyy toteutuneen käyttäytymisen perusteella korvaten alkuvaiheen arvot. ▪ Onko lukkojuttu (kyllä/ei)
Sessio	▪ Sessioon liittyvät artikkelit ▪ Päättyikö sessio tilaukseen (kyllä/ei)
Käyttäjä	▪ Klikattujen artikkelien määrät kategorioittain (yhteensä käyttäjän sessioista) ▪ Näytettyjen artikkeleiden määrät kategorioittain. ▪ Aktiivisuus (pohjautuen sessioiden ja klikattujen artikkeleiden määrään tietyllä aikavälillä)

5.1.3 Palkkion määräytyminen

Palkkio määrittelee vahvistusoppimisongelman tavoitteen (ks. luku 3). Tässä esimerkissä palkkio määräytyi jälkikäteen käyttäjän session lopputuloksen perusteella, eli päättyikö sessio tilaukseen. Sessiosta selviää toteutunut klikkauspolku.

Kuva 15 havainnollistaa toteutetun vahvistusoppimisohjelmiston toimintaa ja mallin inkrementaalista päivittämistä toteutuneen lopputuloksen perusteella. Vahvistusoppimisen kannalta ongelmaa tarkastellaan siis sessiokohtaisesti. Session päätyttyä määritellään lopullisten palkkioiden suuruus; ohjaavana tavoitteena on maksimoida kumulatiivinen palkkio. Sessioon liittyy useita vahvistusoppimisen tiloja. Käyttäjälle näytetyt jutut (esimerkiksi artikkelit 1-4 kuvassa 15) yhdessä käyttäjän ja artikkeleista johdettujen muiden piirteiden kanssa muodostavat yhden vahvistusoppimisen tilan (state). Tilaan liittyvä lopullinen palkkio määräytyy palautteena, kun koko sessio on toteutunut. Mallia päivitetään inkrementaalisesti toteutuneen lopputuloksen pohjalta. Käyttäjän klikkaama artikkeli johtaa taas seuraavaan tilaan, kunnes käyttäjän sessio päättyy.



Kuva 15. Mallin hyödyntäminen ja päivittäminen session päätyttyä.

Palkkion määrittelyllä voidaan painottaa eri asioita. Tässä esimerkissä tavoitteena oli korostaa tilaukseen johtanutta käyttäytymistä ja siksi tilaukseen johtaneen session

klikkauspolun (ks. kuvassa 11 artikkelit 3, 6, 11, 13, 18) artikkeleille annetaan suuri palkkio (toteutetussa esimerkissä palkkio oli 100). Jos sessio ei pääty tilaukseen, määritetään klikattujen artikkeleiden saama palkkio (välillä 1-20) funktiona klikkauspolkuun liittyvistä piirteistä kuten klikattujen lukkojuttujen lukumäärästä. Tällä palkitaan kiinnostavan sisällön esittämisestä, mikä mahdollisesti myöhemmin johtaa tilaukseen. Muusta toiminnasta ei saa palkkiota.

Nyt toteutetussa sovellusesimerkissä dataan ei sisällytetty klikkausten ajankohtaa. Aika voitaisiin ottaa huomioon esimerkiksi niin, että tilaukseen johtaneessa sessiossa painotetaan viimeisiä klikkauksia antamalla niille ensimmäisiä klikkauksia suuremman palkkion.

Myös optimaalisen palkkion suuruutta eri tilanteissa ja sitä, miten palkkio pitäisi määritellä niiden sessioiden osalta, jotka eivät päättyneet tilaukseen voitaisiin tutkia mallin opetusaineiston perusteella koneoppimisen keinoin. Mallin tarkempi evaluointi (seuraava luku 5.1.4) antaa myös mahdollisuuden palkkioiden sopivuuden arviointiin.

5.1.4 Mallin evaluointi

Esimerkkiä varten kehitettiin malli, joka ennustaa valitun toimenpiteen saamaa palkkiota, jonka perusteella arvioidaan artikkelin sopivuutta tilanteessa (artikkelin hyvyysluku, vrt kuva 15). Koneoppimiseen käytettiin MLP (Multi Layer Perceptron) -neuroverkkoa, joka on eteenpäin kytketty ja koostuu syötekerroksesta, yhdestä tai useammasta piilokerroksesta ja vastekerroksesta (Goodfellow et al., 2016). Mallia voi inkrementaalisesti päivittää, mikä mahdollistaa vahvistusoppimisen vaatiman askeleittaisen etenemisen. Kehitetty neuroverkkoon pohjautuva malli oppii arvofunktion toteutuneen palkkion mukaan (Deep Q learning, luku 3). Neuroverkon parametrien tarkempaan optimointiin ei tässä työssä kiinnitetty erityistä huomioita; evaluoinnissa käytettiin verkkoa, jossa on yksi piilokerros (100 neuronia) ja aktivoitiin käytettiin ReLU (Rectified Linear Unit) -funktiota, joka on yleisesti käytetty ja sopiva useimpiin tilanteisiin.

Mallin evaluointia varten opetusdata jaettiin kahteen osaan seuraavasti: 80% datasta käytettiin evaluoitavan mallin opetukseen ja 20 % testaamiseen. Lisäksi tuotettua mallia kokeiltiin kahdella erilaisella tilauksen syntymisperusteella simuloidussa datassa:

1. Sessio merkattiin päättyneeksi tilaukseen, jos sen aikana klikattiin yli yhdeksää lukkojuttua.

2. Täysin satunnaisesti valittu, yksi sadasosa sessioista merkittiin päättyneeksi tilaukseen.

Mallia evaluoitiin siten, että mallin avulla arvioitiin käyttäjän artikkelin valinnan hyvyyttä testidatasta johdetuissa tilanteissa toteutuneeseen palkkioon nähden.

Ensimmäinen tapaus on oletusarvoisesti erittäin helposti opittava, koska on olemassa täysi riippuvuus tuloksen (tilaus) ja mallin piirteen (käyttäjän klikkaamien lukkojuttujen määrä session aikana) välillä. Evaluoinnin tarkkuuden pitäisi olla hyvä, jos ohjelmisto ja malli toimii oikein. Tuloksena saatiin tarkkuus 0.99 (R^2 score), mikä on erittäin hyvä tulos ja osoittaa sen, että mallin avulla voidaan melkein täydellisesti määrittää artikkelin sopivuus käyttäjälle tässä tapauksessa.

Toisessa tapauksessa tilaus syntyi täysin satunnaisesti eikä käyttäjän toimilla ja tilauksella ollut loogista yhteyttä. Tässä arvion tarkkuudeksi saatiin 0.68 eli malli ei arvioi hyvin artikkelin sopivuutta.

Nämä evaluointiesimerkit valittiin kahdesta eri ääripäästä: toinen on täydellisesti opittavissa oleva ja toinen mahdoton oppia, koska se on satunnainen. Käytännössä todelliseen dataan perustuvissa tilanteissa tilanne on jossakin näiden ääripäiden välillä, satunnaisuutta on

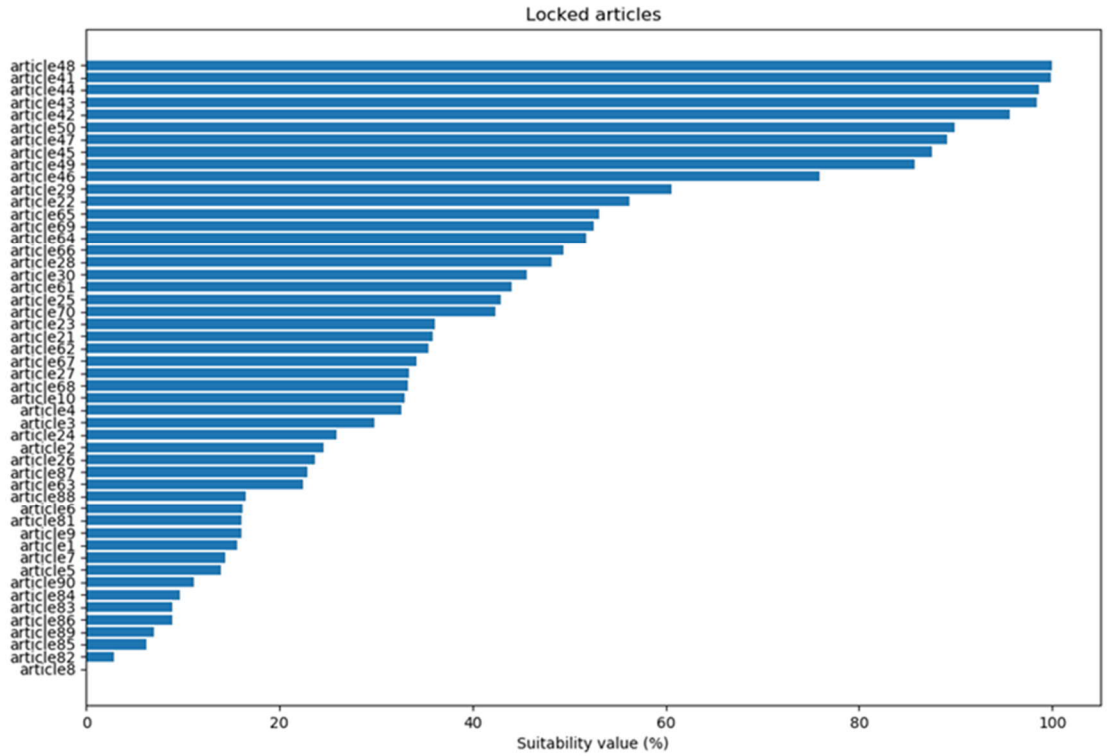
enemmän tai vähemmän, mutta toisaalta koneoppiminen voi tunnistaa monimutkaisia riippuvuussuhteita toteutuneiden sessioiden ominaisuuksien välillä.

Yleisesti ottaen optimaalisen palkkion suuruuden määrittämiseksi voidaan jälkikäteen arvioida valittujen palkkioiden sopivuutta suhteessa lopputuloksiin. Jos malli pystyy ennustamaan lopputulosta tarkasti, palkkio on todennäköisesti valittu hyvin. Toisaalta myös jos mallin ominaisuuksilla (piirteillä) ja lopputuloksella ei ole opittavissa olevaa riippuvuutta, mallin tarkkuus on todennäköisesti erittäin huono, jolloin myös piirteet pitäisi valita uudestaan tai tuoda mukaan enemmän piirteitä ja arvioida niiden merkitsevyyttä mallissa (vrt. Kuva 12 luvussa 4.2).

5.2 Malli ja sen hyödyntäminen

5.2.1 Näytettävien artikkeleiden sopivuuden arviointi

Toteutetun ohjelmiston avulla voidaan arvioida tarjolla olevien artikkelien sopivuutta käyttäjälle. Ohjelmisto järjestää erikseen sekä tarjolla olevat lukkojutut että avoimet jutut sopivuusjärjestykseen. Mallin pohjalta jokaiselle artikkelille voidaan laskea klikkauksen hyvyysarvo tietyssä käyttäjän session tilanteessa.



Kuva 16. Lukkojutut järjestettynä sopivuuden mukaan - esimerkki yhdelle käyttäjälle tehdystä hyvyyslistauksesta.

Yllä olevassa kuvassa tarjolla olevat lukkojutut on järjestetty opetetun mallin avulla yksittäistä käyttäjää varten. Käyttäjän sessiosta annetaan ohjelmalle lähtötietoina käyttäjälle näytettyjen ja käyttäjän klikkaamien artikkelien (avoimet ja lukitut) lukumäärät kategorioittain. Lisäksi ohjelma hakee tarjolla olevista artikkeleista muita tietoja kuten hot ja dhot-arvot. Esimerkin tilanteessa käyttäjä oli klikannut lukittuja juttuja kategorioittain [1, 0, 3, 0, 0] ja vapaita juttuja [0, 1, 3, 0, 0] vastaten kategorioita 1, 2, 3, 4, 5.

Kuvan esimerkin artikkelit kuuluvat viiteen eri kategoriaan (ks. luku 5.3): 1. kategoria (artikkelit 1-20), 2. kategoria (21-40), 3. kategoria (41-60), 4. kategoria (61-80) ja 5. kategoria (81-100). Yllä olevasta kuvasta havaitaan, että Top10 lukitut artikkelit ovat pääosin kolmantena olevan kategorian juttuja. Vastaavasti käyttäjälle voidaan järjestää avoimet artikkelit sopivuuden mukaan.

5.2.2 Lukkojuttujen ja avoimien juttujen suhde

Käyttäjälle näytettävien lukittujen ja avoimien juttujen suhde oletettiin edellä esitetyssä vakioksi. Mallin avulla voitaisiin arvioida tilannekohtaisesti myös sopivaa lukko- ja avoimien juttujen suhdetta. Tällöin voitaisiin esimerkiksi määrittää toimintapolitiikka (policy), joka arvioi lukkojuttujen hyvyyslukujen suhdetta avoimien juttujen hyvyyslukuihin ja sen funktiona määrittää sopivan lukkojuttujen ja avoimien juttujen suhdeluvun. Toinen vaihtoehtoinen tai täydentävä toimintapolitiikka on käyttää käyttäjän aikaisempien klikkaamien lukittujen ja avoimien juttujen suhdetta pohjana ja sen mukaan joko näyttää enemmän lukittuja juttuja, jos käyttäjä on aikaisemmin klikannut niitä keskimääräistä enemmän.

Kehitettyyn ohjelmaan sisällytettiin yksinkertaisena esimerkkinä yllä olevasta yhdistelty toimintapolitiikka (policy), joka tilannekohtaisesti määrittää sopivan lukkojuttujen osuuden tarjolla olevien lukittujen ja avoimien artikkeleiden hyvyyslukujen (vrt. kuva 16) sekä session ja käyttäjän piirteiden (ks. 5.1.2) funktiona. Ohjelmalle asetetaan lähtötietona oletusarvo näytettävien lukkojuttujen osuudelle, joka tässä esimerkissä oli 0.5 ja missä rajoissa suhdeluku voi muuttua (0.3-0.7). Kuvan 16 esimerkissä ohjelma antoi tulokseksi 0.45 eli tässä tilanteessa lukkojuttuja kannattaisi näyttää käyttäjälle hieman vähemmän kuin oletusarvon mukainen suhteellinen määrä.

6. Tulosten tarkastelu ja johtopäätökset

Tutkimuksen tavoitteena oli tarkastella koneoppimismenetelmien ja erityisesti vahvistusoppimisen hyödyntämismahdollisuuksia mediatuotannossa. Tarkennetuksi kohdealueeksi valittiin hankkeessa mukana olleiden yritysten kanssa mediasisältöjen esittäminen etusivulla, mitä tarkasteltiin kahdelta kannalta: paikan merkitys juttujen suosioon ja esitettävien juttujen valinta, kun tavoitellaan tilausten saamista.

- Paikan merkitystä suosioon tarkasteltiin todellisen aineiston pohjalta ja sen suhteen tärkeimmät päätelmät olivat seuraavat:
- Mitä korkeammalla etusivulla juttu on, sitä enemmän se keskimäärin saa klikkauksia, mutta paikan vaikutus tasoittuu suhteellisen nopeasti mentäessä etusivua alaspäin.
- Juttujen keskimääräinen kiinnostavuus samassa sijainnissa vaihtelee huomattavasti osaston perusteella.
- Jutun edeltävänä aikajaksona, esim. viiden minuutin aikana saamat klikkaukset ennustavat hyvin seuraavassa ajanjaksossa kertyviä klikkauksia, mutta aikavälin kasvaessa ennustettavuus heikkenee.
- Juttujen kiinnostavuus laskee nopeasti ajan kuluessa.

Kerätyn aineiston pohjalta tehtiin ennustava malli, joka edeltävän aikajakson klikkausten ja jutun ja etusivun dynaamisten ja staattisten ominaisuuksien pohjalta tekee ennusteen jutulle eri paikoissa kertyvästä klikkausmäärästä. Tällaista mallia voidaan käyttää juttujen esitysjärjestyksen määrittelemisessä.

Esitettävien juttujen valintaan liittyvässä työssä kehitettiin vahvistusoppimiseen ja neuroverkkoon pohjautuva ohjelmisto. Työssä käytettiin simuloitua dataa, eli generoitiin käyttäjät, jutut ja näiden sekä ohjelmiston parametrien perusteella käyttäjäsessiot. Generoidun datan avulla opetettiin malli, joka arvioi tarjolla olevien juttujen käyttäjäkohtaisen hyvyyden. Hyvyys määriteltiin tässä tapauksessa tilaukseen johtavan käyttäytymisen perustella. Tällä hankkeessa kehitetyllä uudella algoritmilla vahvistusoppimisen palaute tapahtuu viivästetysti toteutuneen session ja klikkauspolun perusteella.

Malli oppi generoidusta datasta varsin helposti dataan sisällytettyjä yhteyksiä klikkausten ja tilausten välillä (tässä esimerkissä tilaus merkittiin tehdyksi, kun käyttäjä on klikannut riittävän määrän lukittuja artikkeleita session aikana). Todellisuudessa klikkausten ja tilausten väliltä ei löydy näin suoraviivaista yhteyttä, mutta datasta voi kuitenkin löytyä tapahtumia ja käyttäytymistä, jotka ennakoivat tilauksen tekemistä. Kehitetty, vahvistusoppimiseen ja neuroverkon hyödyntämiseen perustuva malli voi oppia hyvinkin monimutkaisen riippuvuussuhteen piirteiden ja tilauksen välillä, kunhan aineistoa on riittävästi tarjolla. Neuroverkon käyttö mahdollistaa isonkin piirremäärän antamisen syötteenä ilman että tunnetaan syötteiden ja tuloksen riippuvuussuhteita.

Luonteva jatko kehitetyn algoritmin hyödyntämiseksi olisi käyttäjädatan analysointi tilaukseen johtaneiden käyttöpolkujen tunnistamiseksi, esimerkiksi miten käyntitiheys liittyy tilaamiseen. Näiden lisätietojen avulla simuloitua käyttäjädataa ja generoituja käyttäjäsessioita voidaan kehittää eteenpäin paremmin vastaamaan todellisuutta. Tämän jälkeen voidaan taas kehittää itse algoritmia ja testata erilaisten palkkioperusteiden käyttökelpoisuutta ennen isolla, todellisella datamassalla ja varsinkin todellisessa ympäristössä tehtävää käyttöä.

Algoritmin käyttämä inkrementaalinen päivittäminen edistää käyttöä tuotantoympäristössä, koska mallin päivittäminen voidaan tehdä useamman tunnin välein. Tuotantoympäristöstä saatavissa olevan datan laatuun liittyy epävarmuuksia, varsinkin kun tarkastellaan ei-tilaajien käyttäytymistä, jolloin yhden käyttäjän käyttöhistoria voi jakaantua moneen osaan. Voi myös olla hankalaa saada yhdistettyä käyttötietoja todelliseen tilaustietoon, jolloin mallin palkkio voidaan joutua antamaan pelkästään tilauslomakkeen klikkaamisesta.

Yksi mahdollisuus tilausten syntymisen edistämiseen voisi myös olla tilausmainoksen kohdistettu esittäminen hyödyntäen vahvistusoppimisella opittua käyttäytymistä: kun tunnistetaan käyttäjän olevan tilassa, joka on lähellä tilauksen tekemistä, käyttäjälle esitetään tilausmainos muutoinkin kuin ns. lukkojuttujen yhteydessä.

Kuva 17. Projektin kohdealueen menetelmiä ja niiden tyypillisiä sovellusalueita.

Kuva 17 havainnollistaa hankkeessa tarkasteltujen ja hankkeen kohdealueeseen liittyvien menetelmien keskinäisiä suhteita ja sovellusalueita. Työn alkuosassa tarkasteltiin erityisesti vahvistusoppimismenetelmiä, sekä yksivaiheiseen päätöksentekotilanteeseen sopivia monikätisen rosvon problematiikan ratkaisualgoritmeja (MAB, CMAB) että täysmittaisen vahvistusoppimisen menetelmiä. Case-esimerkeissä ei käytetty monikätisen rosvon problematiikan ratkaisualgoritmeja.

MAB- ja CMAB-menetelmillä etsitään dynaamisessa tilanteessa toimintapolitiikka, jolla käytettävissä olevat vaihtoehdot hyödynnetään tehokkaasti. Tyypillinen käyttökohde on mainosten esittäminen: mainosten käyttöikä on lyhyt ja on tärkeää esittää niitä mainoksia tai mainosten niitä versioita, joihin käyttäjä todennäköisimmin reagoi positiivisesti. CMAB-menetelmiä on sovellettu myös esitettävien uutisjuttujen valintaan.

AB-testauksella haetaan vastausta vastaavin kysymyksiin kuin MAB-ratkaisijoilla, mutta sen keskeisenä ideana on hakea tilastollisesti luotettava vastaus. Tämä sopii tilanteisiin, joissa ratkaisu tulee pitempiaikaiseen käyttöön.

Vahvistusoppimismenetelmät hakevat toimintapolitiikkaa, jonka avulla voidaan saavuttaa usean toimenpiteen sarjana saavutettavissa oleva tavoite. Tästä esimerkkinä hankkeessa oli tilauksiin johtava käyttäytyminen. Tilauksia oletettiin syntyvän, kun käyttäjälle esitetään kiinnostavia ilmaisia ja maksullisia juttuja sopivassa suhteessa. Kiinnostavien juttujen valintaa voidaan lähestyä suositusongelmana ja tässä siihen otettiin vahvistusoppimisen pohjautuva ratkaisu, eli artikkelien hyvyys perustui tilaukseen johtaneisiin käyttäytymispolkuihin.

Lähdeviitteet

Beysolow II, T. (2019) Applied Reinforcement Learning with Python. Apress, Berkeley, CA. https://link.springer.com/book/10.1007/978-1-4842-5127-0

Bouneffouf, D. & Rish, I (2019) A Survey on Practical Applications of Multi-Armed and Contextual Bandits. arXiv:1904.10040. 8 s.

Chapelle, O. & Li, L.: An empirical evaluation of thompson sampling. (2011) In: Advances in neural information processing systems. pp. 2249–22579 s.

Chen, S.-Y., Yu, Y., Da, Q., Tan, J., Huang, H.-K. & Tang, H.-H.. (2018). Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In SIGKDD. 1187–1196.

Collier, M. & Llorens, H.U. (2018). Deep Contextual Multi-armed Bandits. ArXiv, abs/1807.09809. 6 s.

van Emden, R. & Kaptein, M. (2018) contextual: Evaluating Contextual Multi-Armed Bandit problem in R. https://arxiv.org/abs/1811.01926. 55 s.

Gligic, L., Kormilitzin, A., Goldberg, P. & Nevado-Holgado, A. (2020). Named entity recognition in electronic health records using transfer learning bootstrapped neural networks. *Neural Networks, 121*, 132-139. doi:10.1016/j.neunet.2019.08.032

Goodfellow, I., Bengio, Y. & Courville, A. (2016). Deep learning. MIT press.

van Hasselt, H., Guez, A. & Silver, D. (2015) Deep Reinforcement Learning with Double Q-learning. https://arxiv.org/abs/1509.06461. 13 s.

Hastie, T.,Tibshirani, R. & Friedman, J.H. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer.

Johansson, F. & Mchome, M. (2018) Comparison of Arm Selection Policies for the Multi-Armed Bandit Problem. Master's thesis in Computer Science. Chalmers University of Technology. Gothenburg, Sweden. 46 s.

Li, L., Chu, W., Langford, J. & Schapire, R. E. (2010). "A Contextual-Bandit Approach to Personalized News Article Recommendation." In *Proceedings of the 19th International Conference on World Wide Web*, 661–70. ACM.

Li, S., Karatzoglou, A., & Gentile, C. (2015). Collaborative filtering bandits. https://arxiv.org/abs/1502.03473. 10 s.

Liu, R., Shi, Y., Ji, C., & Jia, M. (2019). A survey of sentiment analysis based on transfer learning. IEEE Access, 7, 85401-85412. doi:10.1109/ACCESS.2019.2925059

Mattos, D.I., Bosch, J., & Holmström Olsson, H. (2019). Multi-armed bandits in the wild: Pitfalls and strategies in online experiments. Information and Software Technology, 113(September), 68.

Silver, D., Huang, A., Maddison, C. et al. Mastering the game of Go with deep neural networks and tree search (2016) Nature 529, 484–489 doi:10.1038/nature16961

Sutton, R.S. & Barto, A.G. (2018) Reinforcement Learning: An Introduction. MIT Press. 548 s.

Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N.J., Xie, X. & Li, Z.. (2018). DRN: A Deep Reinforcement Learning Framework for News Recommendation. In WWW. 167–176.