Tampereen yliopisto

Enna Raerinne

TIEDONLOUHINTA OPINTOJEN OHJAUKSEN JA SUUNNITTELUN APUNA

Informaatioteknologian ja viestinnän tiedekunta
Pro gradu -tutkielma
Toukokuu 2020

TIIVISTELMÄ

Enna Raerinne: Tiedonlouhinta opintojen ohjauksen ja suunnittelun apuna
Pro gradu -tutkielma, 51 sivua, 18 liitesivua
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Toukokuu 2020

Korkeakouluilla on nykyään kiinnostusta hyödyntää toiminnoistaan muodostuvaa dataa näiden toimintojensa, kuten opetuksen, kehittämiseen tiedonlouhinnan avulla. Tässä tutkielmassa tehtiin katsaus tutkimuksiin tiedonlouhinnan käytöstä koulutusympäristössä ja havaittiin, ettei näissä tutkimuksissa ollut juurikaan hyödynnetty tiedonlouhintamenetelmiä opiskelijoiden opintoihin kiinnittymisen tutkimiseen. Myös opiskelijan koko opintopolkuun liittyvää tutkimusta tiedonlouhintaa hyödyntäen oli tehty vähemmän kuin yksittäisiin kursseihin liittyvää tutkimusta. Tässä työssä myös tutkittiin tiedonlouhinnan avulla opiskelijoiden kiinnittymistä opintoihinsa ja kurssien suosittelua opiskelijoille heidän ensimmäisenä lukuvuotenaan. Opiskelijoiden kiinnittymistä opintoihinsa selvitettiin jakamalla opiskelijat sopiviin, todellisuutta vastaaviin opiskelijaryhmiin klusteroinnin avulla heidän suoritustensa perusteella. Klusteroinnissa kokeiltiin kolmea klusterointimenetelmää, K-means ja EM-algoritmeja sekä hierarkkista klusterointia. Opintoihin kiinnittymistä tutkittiin myös assosiaatioanalyysin avulla etsimällä suosittuja kurssihahmoja opiskelijoiden ensimmäisen lukuvuoden suorituksista Apriori-algoritmilla. Kurssien suosittelua kokeiltiin luokittelupohjaisella menetelmällä, jossa käytettiin C4.5-päätöspuualgoritmia, ja klusteroinnin ja assosiaatioanalyysin yhdistelmällä, jossa oli käytössä K-means ja FP-growth -algoritmit. Louhinta tehtiin tiedonlouhintatyökaluilla Orange ja Weka.

Louhinnan tuloksista kävi ilmi, että klusteroimalla opiskelijat heidän suoritustensa perusteella saadaan esiin opiskelijaryhmiä, joista voidaan havainnoida opintoihin kiinnittymistä. Parhaiten eri opiskelijaryhmiä pystyi tarkastelemaan Orangessa hierarkkisella klusteroinnilla. Kurssihahmoista pystyttiin myös tarkastelemaan opintoihin kiinnittymistä ja vertailemaan erilaisista lähtökohdista aloittavien opiskelijoiden suorituksia. Opiskelijoiden ensimmäisen lukuvuoden suoritukset eivät riitä kurssien suosittelun perusteelliseen tutkimiseen, vaan dataa tarvittaisiin enemmän, mutta eri louhintamenetelmiä pystyttiin silti testaamaan. Tässä työssä tutkitut tiedonlouhintamenetelmät soveltuvat myös myöhempien opiskeluvuosien suoritusten analysointiin.

Avainsanat: tiedonlouhinta, luokittelu, assosiaatioanalyysi, klusterointi, datan esikäsittely, koulutusdatan louhinta, educational data mining, opiskelijoiden ryhmittely, kurssihahmot, kurssien suosittelu

Sisällys

1 Johdanto

Korkeakouluilla on nykyään kiinnostusta pyrkiä hyödyntämään toiminnoistaan, kuten opetuksesta, muodostuvaa dataa toimintojensa kehittämiseen. Dataa voidaan hyödyntää tiedonlouhinnan avulla, jossa data pyritään muuttamaan hyödylliseksi tiedoksi. Näin saatua tietoa voidaan käyttää esimerkiksi päätöksenteon tukena. Koulutusympäristössä tiedonlouhintaa voidaan käyttää muun muassa opiskelijoiden ryhmittelyssä, opiskelijoiden suoriutumisen ja pysyvyyden ennustamisessa, kurssien ja kurssimateriaalien suosittelussa ja oppimisen tutkimisessa. [Luan 2002, Romero and Ventura 2007, Baker and Yasef 2009, Goyal and Vohra 2012]

Tiedonlouhinnan hyödyntämisestä koulutusympäristössä on tehty tässä tutkielmassa katsaus, eikä tähän liittyvässä tutkimuksessa ole juurikaan käytetty louhintamenetelmiä opiskelijoiden opintoihin kiinnittymisen tutkimiseen tiedonlouhinnan avulla. Myös opiskelijan koko opintopolkuun liittyvää tutkimusta tiedonlouhintaa hyödyntäen on vähemmän verrattuna yksittäisiin kursseihin liittyvään tutkimukseen.

Tässä työssä tiedonlouhinnan avulla selvitetään opiskelijoiden kiinnittymistä opintoihinsa ensimmäisenä lukuvuotenaan, joka on mielenkiintoista tietoa korkeakoulun opetus- ja ohjaushenkilökunnalle. Opiskelijoiden kiinnittymistä opintoihinsa tutkitaan jakamalla opiskelijat ryhmiin heidän suoritustensa perusteella käyttäen kolmea eri klusterointimenetelmää ja tarkastelemalla muodostuneita opiskelijaryhmiä. Opintoihin kiinnittymistä selvitetään myös etsimällä opiskelijoiden suorituksista suosittuja kurssihahmoja assosiaatioanalyysin avulla. Opiskelijoille opintojen suunnittelua varten tutkitaan kurssien suosittelua eri tiedonlouhintamenetelmin. Tiedonlouhinnassa käytetty data on poimittu Tampereen yliopiston opiskelijatietojärjestelmästä ja se sisältää vuosina 2017 ja 2018 Tampereen yliopistossa tietojenkäsittelytieteiden tutkinto-ohjelmassa aloittaneiden opiskelijoiden ensimmäisen lukuvuoden opintosuoritukset yliopistossa. Tässä työssä louhintaan käytetään kahta tiedonlouhintatyökalua Orangea ja Wekaa [Orange 2020, Weka 2020].

Louhinnan tuloksista ilmeni, että klusteroimalla opiskelijat heidän suoritustensa perusteella saadaan esiin opiskelijoista muodostuvia ryhmiä, joista voi havainnoida opintoihin kiinnittymistä. Parhaiten opiskelijaryhmiä pystyi tarkastelemaan Orangessa hierarkisella klusteroinnilla. Kurssihahmoista pystyttiin myös tarkastelemaan opintoihin kiinnittymistä sekä vertailemaan erilaisista lähtökohdista aloittavien opiskelijoiden suorituksia. Kurssien suosittelun perusteelliseen tarkasteluun data ei ollut riittävää, mutta erilaisia louhintamenetelmiä pystyttiin testaamaan. Opiskelijoiden suoritusten analysointiin sopivat tässä työssä tutkitut menetelmät myös myöhempinä opiskeluvuosina.

Tämän työn louhintaprosessissa korostui datan valinta, sillä ilman sopivaa dataa ei voida tehdä perusteellista tutkimusta eikä löydetä kiinnostavia hahmoja, jos ne peittyvät

toisenlaisten hahmojen alle. Datan valinnan lisäksi sen huolellinen esikäsittely parantaa tulosten laatua ja paikkansa pitävyyttä. Lisäksi kahden eri louhintatyökalun käyttö toi esiin miten tärkeää tulosten havainnollistaminen on, jotta tulokset olisivat ymmärrettäviä. Tulosten laatuun vaikuttaa myös louhintatekniikoiden parametrien säätäminen, kuten esimerkiksi sopivan klusterien lukumäärän määrittäminen. Parametrien säätäminen voi kuitenkin olla vaikeaa, jos tiedonlouhinta ja eri tekniikat eivät ole käyttäjälle tuttuja. Tästä syystä esimerkiksi korkeakoulun opetus- ja ohjaushenkilökunnan voi olla hankalaa hyödyntää tiedonlouhintaa opintojen ohjauksessa.

Tässä työssä tutustutaan ensin tiedonlouhintaan yleisesti luvussa 2, jonka jälkeen seuraa katsaus tiedonlouhinnasta koulutusympäristössä luvussa 3. Luvussa 4 kerrotaan yksityiskohtaisemmin, millaista dataa työssä käytetään, millaisiin louhintatehtäviin sitä tarvitaan ja miten dataa on esikäsitelty louhintaa varten. Luvussa 5 käydään läpi louhinnassa käytettävät algoritmit ja luvussa 6 louhinnan tulokset, jossa niitä myös analysoidaan. Viimeisenä seuraa yhteenveto luvussa 7.

2 Tiedonlouhinta

Koko ajan kertyvät suuret datamäärät ja tarve muuttaa data hyödylliseksi tiedoksi ovat tuoneet tiedonlouhinnalle (data mining) paljon huomiota. Datan määrä ja kasvuvauhti on niin suuri, ettei ihminen pysty käsittelemään tai ymmärtämään dataa ilman tehokkaita työkaluja. Monesti kerätty data jää lojumaan säilytyspaikkaansa ilman käsittelyä eikä sitä käytetä päätöksentekoon, jonka avuksi se voisi antaa paljon tietoa. Usein tärkeät päätökset tehdään päätöksentekijän intuition varassa, koska hänellä ei ole työkaluja arvokkaan tiedon löytämiseen suuresta määrästä dataa. Tiedonlouhintatyökaluilla analysoidaan dataa ja voidaan löytää tärkeitä datahahmoja (pattern) ja -malleja (model), joilla voidaan edesauttaa muun muassa liiketoimintastrategioita ja tieteellistä tutkimusta. Datan ja tiedon välinen kasvava kuilu vaatii tiedonlouhintatyökalujen systemaattista kehitystä, joka muuntaa käyttämättömän datan arvokkaaksi tiedoksi. [Han *et al.* 2012]

Yksinkertaisesti määriteltynä tiedonlouhinta tarkoittaa tiedon tai tietämyksen (knowledge) louhintaa suuresta määrästä dataa [Han *et al.* 2012]. Louhintaprosessin täytyy olla automaattinen tai, useimmiten, puoliautomaattinen ja löydettyjen datahahmojen ja mallien täytyy olla merkityksellisiä niin, että niistä on hyötyä. Yleensä hyöty on taloudellista. [Witten *et al.* 2011]

Monet pitävät tiedonlouhintaa synonyyminä tietämyksen muodostamiselle (knowledge discovery from data(base)). Toiset vaihtoehtoisesti pitävät tiedonlouhintaa yhtenä tärkeänä osana tietämyksen muodostusprosessia. Tietämyksen muodostus koostuu datan esikäsittelystä (data preprocessing), tiedonlouhinnasta, (data)hahmojen ja mallien evaluoinnista (pattern evaluation) ja tiedon esittämisestä (knowledge presentation). Datan esikäsittelyssä on useita eri vaiheita, kuten datan puhdistaminen (data cleaning), datan yhdistäminen (data integration), datan valinta (data selection) ja muunnos (data transformation), joissa valmistellaan data louhintaa varten. Varsinaisessa tiedonlouhinnassa käytetään tekniikoita useilta eri tieteenaloilta kuten tietokantateknologiasta, tilastotieteestä, koneoppimisesta (machine learning), suurteholaskennasta (high-performance computing), hahmontunnistuksesta (pattern recognition), neuroverkoista (neural networks), datan visualisoinnista (data visualization), tiedonhausta (information retrieval), kuvan- ja signaalinkäsittelystä (image and signal processing) ja spatiaalisesta tai temporaalisesta dataanalytiikasta (spatial or temporal data analysis). Yleensä tiedonlouhintatehtävät voidaan jakaa kuvailevaan (descriptive) ja ennustavaan (predictive) tiedonlouhintaan. Kuvailevat louhintatehtävät luonnehtivat datan yleisiä ominaisuuksia ja ennustavat louhintatehtävät tekevät päätelmiä sen hetkisestä datasta antaakseen ennusteita. [Han *et al.* 2012]

Tiedonlouhinnalla on mahdollisuus generoida tuhansia, ellei jopa miljoonia hahmoja, malleja tai sääntöjä. Kuitenkaan kaikki löydetyt hahmot eivät ole kiinnostavia ja vain

pieni osa olisi oikeasti kiinnostavia käyttäjälle. Hahmo tai malli on kiinnostava, jos ihminen pystyy ymmärtämään sen helposti, jos se on jollain varmuudella validi uudella tai testidatalla kokeiltuna ja mahdollisesti hyödyllinen tai uudenlainen. Hahmo on myös mielenkiintoinen, jos se vahvistaa käyttäjän hypoteesin. Kiinnostava hahmo esittää tietoa. Hahmon kiinnostavuuden määrittämiseen on olemassa objektiivisia mittareita, jotka perustuvat löydetyn hahmon rakenteeseen ja siihen liittyviin tilastollisiin tunnuslukuihin. [Han *et al.* 2012]

Tiedonlouhinnassa esiintyy pohjimmiltaan neljää eri louhintatyyliä. Luokittelussa (classification) on aluksi joukko luokiteltuja esimerkkejä, joista luokittelumallin oletetaan oppivan tavan luokitella luokittelematonta dataa. Numeerisessa ennustamisessa (numeric prediction) ennustettu tulos ei ole diskreetti luokka vaan numeerinen suure. Esimerkiksi regressioanalyysi (regression) on numeerista ennustamista. Assosiaatioanalyysissä (association analysis) etsitään mitä tahansa assosiaatioita datan ominaisuuksista, ei vain niitä, jotka ennustavat tietyn luokan arvon, kuten luokittelussa. Klusteroinnissa (clustering) etsitään dataryhmiä, jotka kuuluvat yhteen. Näitä dataryhmiä kutsutaan klustereiksi. [Witten *et al.* 2011]

2.1 Data

Data koostuu tapauksista. Jokaisella tapauksella on attribuutteja, jotka kertovat tapauksen ominaisuuksista. Tietyn tapauksen attribuutin arvo on mitta siitä suureesta, johon attribuutti viittaa. Datamatriisissa rivit voivat olla tapauksia ja sarakkeet tapauksien attribuutteja, tai päinvastoin. Attribuutit voidaan jakaa numeerisiin ja laatueroattribuutteihin (nominal). Niiden välillä on selvä ero. Numeeriset attribuutit, joita joskus kutsutaan jatkuviksi (continuous) attribuuteiksi, mittaavat määrää, joko kokonais- tai reaalilukuarvoina. Huomioitavaa on, että termiä jatkuva käytetään tässä yhteydessä usein väärin, sillä kokonaisluvut eivät ole matemaattisesti jatkuvia. Laatueroattribuutit saavat arvonsa ennalta määritellystä, rajallisesta joukosta mahdollisia arvoja. Toinen tapa luokitella attribuutteja on tyypeillä järjestys (ordinal), välimatka, (interval) ja suhdeluku (ratio). Järjestysasteikolliset attribuutit kuvaavat luokkien välistä järjestystä ja välimatka-asteikkoisten attribuuttien luokkien välimatka on aina yhtä suuri ja ne voi myös järjestää. Suhdelukuasteikossa on aina absoluuttinen nollapiste ja niitä kohdellaan reaalilukuina. Binäärinen (binary) attribuutti on laatueroasteikollinen, mutta sillä on vain kaksi arvoa, kuten tosi ja epätosi. [Witten *et al.* 2011]

Tiedonlouhintaan käytettävä data on kerätty melko varmasti muuta tarkoitusta varten. Kun dataa on alun perin tallennettu, monet attribuutit eivät ole olleet niin tärkeitä ja ne on siksi jätetty tyhjiksi tai tarkastamatta. Jos tämä ei vaikuta datan alkuperäiseen käyttötarkoitukseen, ei ole myöskään kannustinta korjata tilannetta. Kuitenkin kun samaa dataa käytetään louhimiseen, virheet ja puutteet datassa saavat suuremman merkityksen. Siksi on tärkeää tarkistaa data huolellisesti poikkeavien attribuuttien ja arvojen varalta. [Witten

et al. 2011] Jos dataa on todella paljon ja se on peräisin useista eri lähteistä, se on usein epäjohdonmukaista, siitä puuttuu arvoja tai se sisältää virheellistä dataa (noise) [Han *et al.* 2012]. Redundantti data on myös yksi mahdollinen virheiden lähde louhinnassa. Monet louhintatyökalut antavat eri tuloksia, jos joitakin samoja tapauksia tai muuttujia on useamman kerran datajoukossa, koska toisto antaa niille enemmän vaikutusta tulokseen. Lopulta data menee myös vanhaksi. Monet tapaukset tietokannassa muuttuvat olosuhteiden muuttuessa. Täytyy siis ottaa huomioon, onko louhittava data yhä ajankohtaista. [Witten *et al.* 2011]

Huonolaatuinen data johtaa huonolaatuisiin louhintatuloksiin. Tästä syystä dataa esikäsitellään ennen louhintaa, sillä se parantaa datan laatua, siten auttaen parantamaan louhintaprosessin tarkkuutta ja tehokkuutta. Datan esikäsittely on tärkeä vaihe tietämyksen muodostusprosessissa, koska laadukkaiden päätösten täytyy perustua laadukkaalle datalle. Datan poikkeamien havaitseminen ja oikaiseminen ajoissa sekä analysoitavan datan tiivistäminen helpottaa lopullista päätöksentekoa suuresti. [Han *et al.* 2012]

Dataa voidaan esikäsitellä monella tavalla. Eri esikäsittelytekniikat eivät sulje toisiaan pois, vaan niitä voidaan käyttää yhdessä. Datan puhdistuksessa täytetään puuttuvat arvot, tasoitetaan häiriödataa, tunnistetaan ja poistetaan poikkeavat havainnot (outliers) ja selvitetään epäjohdonmukaisuudet. Datan valinnassa haetaan tietokannasta analyysitehtävälle merkityksellinen data. Jos datalähteitä on useita, tarvitaan datan yhdistämistä, jossa sulautetaan dataa useista eri lähteistä yhtenäiseen säilytyspaikkaan, kuten tietovarastoon. Yhdistämistä tarvitaan, jos esimerkiksi jotkut attribuutit esittävät saman asian eri nimillä eri tietokannoissa aiheuttaen epäjohdonmukaisuutta ja päällekkäisyyksiä. Täydentävää datan puhdistusta voidaan tehdä löytämään ja poistamaan tällaista dataa, jota on syntynyt datan yhdistämisestä. Datan muunnosoperaatiot, kuten normalisointi ja aggregointi, ovat täydentäviä esikäsittelymenetelmiä, jotka edistävät louhintaprosessin onnistumista. [Han *et al.* 2012]

Datan vähentämisellä (data reduction) saadaan vähennetty esitys datajoukosta, joka on paljon pienempi kooltaan, mutta tuottaa silti saman tai melkein saman analyyttisen tuloksen. Datan vähennystä voi tehdä monella tapaa. Niitä on esimerkiksi datan koostaminen, attribuuttien vähentäminen ja valinta esimerkiksi korrelaatioanalyysin avulla ja monilukuisuuden vähentäminen (numerosity reduction) korvaamalla dataa vaihtoehtoisilla, pienemmillä esityksillä, kuten klustereilla. Dataa voidaan vähentää myös yleistyksellä, korvaamalla matalan tason käsitteet korkeamman tason käsitteillä kuten kaupunkien korvaaminen osavaltioilla tai maakunnilla. Numeeriselle datalle yleistys voidaan tehdä esimerkiksi pilkkomalla jatkuvan attribuutin arvot sopiviksi intervalleiksi. [Han *et al.* 2012]

Useimmat tiedonlouhinta-algoritmit olettavat, että tapauksen attribuutin arvon puuttumisella ei ole erityistä merkitystä. Arvoa ei vain yksinkertaisesti tiedetä. Kuitenkin sille

voi olla hyvä syy miksi attribuutin arvo puuttuu. Ehkä on tehty päätös, ettei suoriteta jotain tiettyä testiä ja se voi välittää informaatiota tapauksesta eikä vain, että arvoa ei ole tallennettu. Jos näin on, olisi sopivampaa kirjata tapaus testaamattomaksi tai ehkä tehdä uusi attribuutti tätä varten. Vain joku, joka tuntee datan, voi tehdä asiantuntevan arvion siitä, onko kyseisen arvon puuttumisessa jotain ylimääräistä merkitystä vai pitäisikö sitä kohdella vain tavallisena puuttuvana arvona. Jos puuttuvia arvoja on montaa tyyppiä, se on selvä merkki, että on tekeillä jotain, josta täytyy ottaa selvää. [Witten *et al.* 2011]

Datasta on olennaista saada hyvä kokonaiskuva, jotta datan esikäsittely onnistuisi. Kuvailevia datan yhteenvetotekniikoita voidaan käyttää tunnistamaan datan tyypilliset ominaisuudet ja korostamaan mitä datan arvoja tulisi kohdella häiriöinä tai poikkeavina havaintoina. [Han *et al.* 2012] Esimerkiksi pylväsdiagrammit laatueroattribuuttien arvojen jakautumisesta ja sironta- (scatter plot) ja laatikko-jana -kuviot (box plot) numeeristen attribuuttien arvoista ovat todella hyödyllisiä. Datan alan asiantuntijoita tulisi konsultoida selittämään muun muassa datan poikkeamat, puuttuvat arvot ja kokonaisluvuin merkityt kategoriat. [Witten *et al.* 2011]

2.2 Luokittelu

Datan luokittelu on kaksiosainen prosessi. Ensimmäinen osa on oppimisvaihe (learning step, training phase), jossa muodostetaan luokittelumalli (classification model) eli miten uudet tapaukset tulisi luokitella. Toinen osa on varsinainen luokittelu, jossa ensimmäisessä vaiheessa muodostettua mallia käytetään ennustamaan tapauksille luokkatunnukset (class label), tai toisin sanoin, luokat. [Han *et al.* 2012]

Ensimmäisessä vaiheessa rakennetaan luokittelija (classifier), joka kuvailee ennalta määrätyt datan luokat ja käsitteet. Tämä on oppimisvaihe, jossa luokittelualgoritmi muodostaa luokittelijan analysoimalla opetusjoukkoa (training set) eli oppii datasta. Opetusjoukko koostuu datasta attribuutteineen ja tapauksille annetuista luokkatunnuksista. Luokkatunnus on laatueroasteikollinen ja opetusjoukko on koko datajoukosta satunnaisesti valittua dataa. Koska jokaiselle opetusjoukon tapaukselle on annettu luokkatunnus, tätä vaihetta kutsutaan myös ohjatuksi oppimiseksi (supervised learning). Luokittelijan oppiminen on ohjattua, koska jokainen tapaus kuuluu johonkin luokkaan. [Han *et al.* 2012]

Toisessa vaiheessa luokittelumallia käytetään luokitteluun. Ensimmäiseksi arvioidaan luokittelutarkkuus käyttämällä mallia dataan, jota ei ole käytetty mallin rakentamisessa, mutta joiden luokkatunnukset tiedetään. Tätä kutsutaan testausjoukoksi (test set). Luokittelutarkkuus testausjoukolle on se, kuinka moni testausjoukon tapaus on luokiteltu oikein, prosentteina annettuna. Jokaisen testausjoukon tapauksen luokkaa verrataan luokittelijan ennustamaan luokkaan. Jos luokittelijan tarkkuutta pidetään hyväksyttävänä, luokittelijaa voidaan käyttää luokittelemaan dataa, jonka luokkatunnuksia ei tiedetä. Luo-

kittelutarkkuuden arviointiin on useita menetelmiä kuten itse luokitteluunkin. Luokittelumenetelmiä ovat muun muassa päätöspuupäättely (decision tree induction), johon kuuluvat esimerkiksi algoritmit ID3, C4.5 ja CART, bayesilainen luokittelu (Bayesian classification), johon kuuluu naiivi Bayes -luokittelu (naïve Bayesian classification), sääntöpohjainen luokittelu (rule-based classification) käyttäen pohjana päätöspuuta tai käyttäen peräkkäistä peittämisalgoritmia (sequential covering) ja luokittelu erilaisilla neuroverkoilla (neural networks) sekä tukivektorikoneella (support vector machine). [Han *et al.* 2012]

2.3 Assosiaatioanalyysi

Assosiaatioanalyysissä ei ole tiettyä luokkaa vaan tehtävänä on löytää datasta mielenkiintoisia rakenteita [Witten *et al.* 2011]. Tällaisia rakenteita ovat esimerkiksi kattavat hahmot (frequent patterns). Kattavat hahmot ovat hahmoja, jotka esiintyvät datajoukossa toistuvasti. [Han *et al.* 2012] Esimerkiksi tietoalkiojoukko (itemset) eli attribuutti-arvoparien yhdistelmä on kattava hahmo [Han *et al.* 2012, Witten *et al.* 2011]. Tietoalkiojoukko, kuten maito ja leipä, jotka esiintyvät toistuvasti yhdessä kaupan ostosdatassa, on kattava joukko (frequent itemset) [Han *et al.* 2012]. Terminologia tulee ostoskorianalyysistä (market basket analysis), jossa tietoalkiot (item) eli attribuutti-arvo-parit ovat tavaroita ostoskorissa ja kaupan myymäläpäällikkö etsii assosiaatioita näiden hankintojen joukosta [Witten *et al.* 2011].

Ostoskorianalyysi on tyypillinen esimerkki kattavan joukon louhinnasta eli etsitään tietoalkioiden assosiaatioita ja korrelaatioita suurista (transaktio- ja relaatio-) datajoukoista. Ostoskorianalyysissä analysoidaan asiakkaan ostoskäyttäytymistä löytämällä assosiaatioita eri tavaroiden välillä, joita asiakas laittaa ”ostoskoriinsa”. Nämä assosiaatiot voivat auttaa kauppiaita kehittämään markkinointistrategioita antamalla tietoa siitä mitä tavaroita ostetaan yhdessä. Esimerkiksi jos asiakkaat ostavat maitoa, kuinka todennäköisesti he ostavat myös leipää (ja millaista leipää) samalla kauppareissulla? Tällainen tieto saattaa johtaa korkeampiin myyntilukuihin auttaen kauppiaita tekemään valikoivaa markkinointia ja suunnittelemaan hyllytilan käyttöä. [Han *et al.* 2012]

Datajoukosta löydetyt assosiaatiot (ja kattavat joukot) voidaan ilmaista assosiaatiosääntöinä (association rules). Esimerkiksi assosiaatio, että asiakkaat, jotka ostavat tietokoneen ostavat yleensä myös virustorjuntaohjelmiston samalla kertaa, voidaan esittää seuraavanlaisena assosiaatiosääntönä: tietokone ⇒ virustorjuntaohjelmisto [tuki = 2%, luottamus = 60%]. Kahden prosentin tuki assosiaatiosäännölle tarkoittaa, että kahdessa prosentissa kaikista analysoitavista transaktioista tietokone ja virustorjuntaohjelmisto ostettiin yhdessä. Kuudenkymmenen prosentin luottamus tarkoittaa, että kuusikymmentä prosenttia asiakkaista, jotka ostivat tietokoneen, ostivat myös virustorjuntaohjelmiston. Säännön tuella ja luottamuksella mitataan assosiaatiosäännön mielenkiintoisuutta. Ne kuvaavat löydettyjen sääntöjen hyödyllisyyttä ja varmuutta. Tyypillisesti assosiaatiosääntöjä pidetään mielenkiintoisina ja niitä pidetään vahvoina sääntöinä, jos ne ylittävät sekä

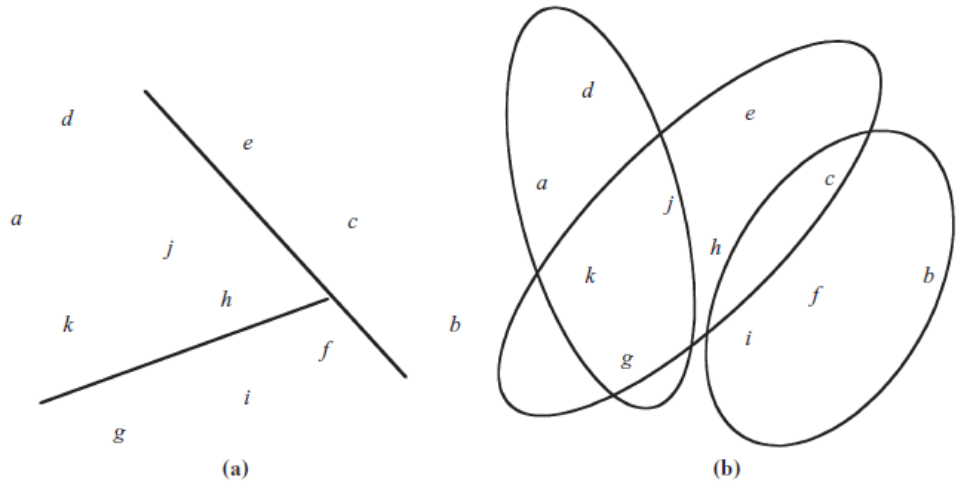
minimituen kynnysarvon, että minimiluottamuksen kynnysarvon. Kynnysarvot voi määritellä käyttäjä tai alan asiantuntija. [Han *et al.* 2012]

Assosiaatiosääntöjen louhinta voidaan supistaa kattavien joukkojen louhinnaksi. Yleensä assosiaatiosääntöjen louhinta on kaksivaiheinen prosessi. Ensin etsitään kaikki kattavat joukot. Jokaisen tällaisen joukon täytyy esiintyä vähintään minimituen verran analysoitavassa datassa. Toisessa vaiheessa generoidaan vahvoja sääntöjä kattavista joukoista. Näiden sääntöjen täytyy täyttää minimituen lisäksi minimiluottamus. [Han *et al.* 2012] Osasta joukoista tulee useampi sääntö, toisista ei tule yhtään luottamusta täyttävää sääntöä. Assosiaatiosääntöjä voi generoitua paljon ja haasteeksi voi muodostua niiden määrä. Määrää pyritään rajaamaan minimituen ja -luottamuksen kynnysarvoja säätämällä. Sääntöjä saattaa silti olla paljon ja niitä täytyy tarkastella manuaalisesti, jotta voidaan määritellä ovatko ne merkitseviä vai eivät. [Witten *et al.* 2011] Esimerkkejä assosiaatioanalyysimenetelmistä ovat Apriori-algoritmi ja FP-growth-algoritmi [Han *et al.* 2012].

2.4 Klusterointi

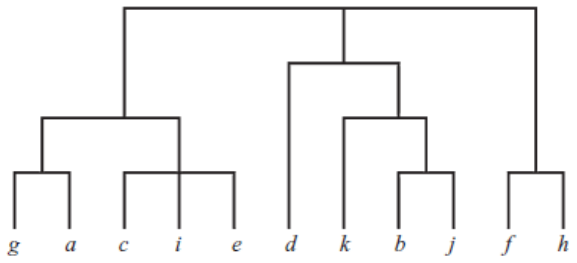
Klusterointia käytetään, kun ei ole ennustettavaa luokkaa, vaan tapaukset jaetaan luonnollisiin ryhmiin niiden samankaltaisuuden perusteella [Witten *et al.* 2011, Han *et al.* 2012]. Toistensa kanssa samankaltaiset tapaukset ovat yhdessä klusterissa ja ne eroavat toisissa klustereissa olevista tapauksista. Joskus klusterointia kutsutaan datan segmentoinniksi, koska klusterointi jakaa isoja datajoukkoja ryhmiin samankaltaisuuden perusteella. [Han *et al.* 2012] Haasteena on löytää klusterit, liittää tapaukset niihin ja pystyä liittämään myös uusia tapauksia klustereihin [Witten *et al.* 2011]. Klusterointi on ohjaamatonta oppimista (unsupervised learning). Se ei ole riippuvainen ennalta määrätyistä luokista ja luokitellusta opetusjoukosta. Siksi klusterointi on oppimista havainnoinnin kautta eikä oppimista esimerkkien kautta kuten luokittelu. [Han *et al.* 2012]

Klusteroinnin tuloksia voidaan ilmaista monella eri tapaa (kuva 1). Tunnistetut ryhmät voivat olla eksklusiivisia eli jokainen tapaus kuuluu vain yhteen ryhmään (kuva 1 (a)). Ryhmät voivat olla myös osittain päällekkäisiä eli tapaus voi kuulua useampaan eri ryhmään (kuva 1 (b)) tai tulokset voivat olla todennäköisyyksiä eli miten todennäköisesti tapaus kuuluu kuhunkin ryhmään (kuva 1 (c)). Tulokset voivat olla myös hierarkkisia eli tapaukset jaetaan karkeasti ryhmiin ja jokainen ryhmä jaetaan vielä osiin ja niin edelleen, ehkä jopa yksittäisiin tapauksiin asti (kuva 1 (d)) tai prosessi voidaan tehdä päinvastoin. Näiden eri tyyppisten tulosten valinta tulisi määräytyä sen mukaan, minkälaisen mekanismin arvellaan vaikuttavan tietyn klusterointi-ilmiön takana. Kuitenkin, koska mekanismia harvoin tiedetään ja juuri klusterien olemassaolo halutaan saada selville, tulostyypin valinnan määrittävät usein saatavilla olevat klusterointityökalut. [Witten *et al.* 2011]



	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1
…			

(c)



(d)

Kuva 1. Erilaisia tapoja esittää klustereita: eksklusiiviset klusterit (a), päällekkäiset klusterit (b), klustereihin kuulumisen todennäköisyydet (c) ja hierarkkinen klusterointipuu (d) [Witten *et al.* 2011].

Klusterointia käytetään sellaisenaan saamaan tietoa datan jakautumisesta, klustereiden ominaisuuksien havainnointiin ja keskittymään tiettyjen klustereiden lähempään analysointiin. Vaihtoehtoisesti klusterointi voi olla esikäsittelyvaihe muille tehtäville, kuten datan kuvailussa, attribuuttien osajoukon valinnassa ja luokittelussa, joissa käytettäisiin löydettyjä klustereita, valittuja attribuutteja ja ominaisuuksia. Klusteroinnilla voidaan myös havaita mielenkiintoisia korrelaatioita attribuuttien välillä. Yhtä klusteria, jossa on useita tapauksia, voidaan käsitellä yhtenä ryhmänä ja siten klusterointia voi käyttää myös datan vähentämiseen. Klusterointia voidaan käyttää myös poikkeamien havaitsemiseen,

jossa poikkeavat havainnot eli tapaukset, jotka ovat kaukana klustereista, voivat olla mielenkiintoisempia kuin yleiset tapaukset. Klusterianalyysiä on käytetty laajasti lukuisissa käyttökohteissa, kuten markkinatutkimuksessa, hahmontunnistuksessa, data-analyysissä ja kuvankäsittelyssä. [Han *et al.* 2012]

Klusterointialgoritmeja on useita erilaisia, muun muassa K-means (k:n keskiarvon klusterointimenetelmä) ja K-medoid (ja niiden variaatiot), jotka tuottavat eksklusiivia klustereita ja EM-algoritmi (Expectation–Maximization algorithm), jonka tulokset ovat todennäköisyyksiä klustereihin kuulumisesta [Han *et al.* 2012, Witten *et al.* 2011]. Hierarkkisia klusterointialgoritmeja ovat esimerkiksi AGNES (AGglomerative NESting) ja DIANA (DIvisive ANAlysis), jotka ovat nimiensä mukaisesti yhdistävä hierarkkinen klusterointimenetelmä ja jakava hierarkkinen klusterointimenetelmä [Han *et al.* 2012].

3 Koulutusdatan louhinta

Koulutusinstituutioista on tullut kompleksisia organisaatioita. Ne eivät enää tarjoa vain koulutusta, vaan hallinnoivat laajaa valikoimaa toimintoja, kuten instituutioiden markkinointia mahdollisille tuleville opiskelijoille, instituutioiden markkinointia yrityksille helpottamaan opiskelijoiden työhönsijoittumista, sisäisten toimintojen johtamista kuten kurssien sujuvaa järjestämistä tai henkilöstön rekrytointia ja motivointia, taloudellista suunnittelua ja yhteistyötä sääntely- ja lakisääteisten auktoriteettien kanssa. Näitä tehtäviä varten koulutusinstituutiot tarvitsevat moderneja johtamistapoja ja uusinta teknologiaa. [Goyal and Vohra 2012]

Korkeakouluopetuslaitokset ovat tulossa tietoisiksi koulutusdatan tutkimisen mahdollisuudesta parantaa niiden hallinnollisten päätösten laatua. Ne näkevät vaivaa ja sijoittavat tietojärjestelmien luomiseen, joilla voidaan kerätä koulutukseen liittyvää dataa, jota tutkitaan käyttäen tiedonlouhintatekniikoita. [Miguéis *et al.* 2018] Tiedonlouhinnan käyttöä koulutusympäristössä kertyvään dataan kutsutaan englanninkielisellä termillä Educational Data Mining (EDM) [Baker and Yasef 2009]. Data voidaan täten muuntaa hyödylliseksi tiedoksi tiedonlouhinnan avulla [Parack *et al.* 2012]. Louhinnalla saatua tietoa voidaan hyödyntää esimerkiksi oppimisen ja opetuksen kehittämisessä ja institutionaalisen tehokkuuden parantamisessa [Dutt *et al.* 2015]. Institutionaalinen tehokkuus tarkoittaa oppimisen arviointia laajemmassa mittakaavassa ja sillä on laajamittainen vaikutus korkeakouluopetukseen [Luan 2002].

Kuitenkaan useimmat korkeakouluista eivät ole pystyneet analysoimaan keräämäänsä dataa ja muuntamaan sitä hyödylliseksi informaatioksi, tiedonlouhinnalla tuetun data-analyysin lupaavasta potentiaalista huolimatta. Tosiasiassa, suurimmassa osassa tapauksia, vain tilastotieteen tukemia perinteisiä menetelmiä on käytetty dataan. Sen vuoksi näiden uuden sukupolven tekniikoiden käyttö on selvästi korkeakoulujen agendassa, jotta saadaan tuettua koulutusstrategioiden kehittämistä. [Miguéis *et al.* 2018]

Tiedonlouhintaa on käytetty onnistuneesti e-taloudessa (e-commerce) ja sillä on saatu myös lupaavia tuloksia EDM:ssä, esimerkiksi digitaalisessa oppimisessa (e-learning). Vaikkakin käytetyt menetelmät ovat samankaltaisia, alueissa on merkittäviä eroja. Esimerkiksi e-taloudessa tarkoitus on ohjata asiakkaita tekemään ostoja, kun taas digitaalisessa oppimisessa ohjata opiskelijoita oppimisessa. Datassa on myös eroja, sillä e-talouden data on yksinkertaista lokidataa ja digitaalisen oppimisen datassa on enemmän informaatiota opiskelijan interaktioista. Käyttäjämalli on myös kummassakin erilainen. E-taloudessa tiedonlouhinnan tavoitteena on liikevoitto, jota voi mitata muun muassa rahamäärällä, asiakkaiden määrällä ja asiakasuskollisuudella. Digitaalisessa oppimisessa louhinnan päämäärä on oppimisen parantaminen, joka on subjektiivisempi tavoite ja haastavampaa mitata. [Romero and Ventura 2007]

Korkeakouluopetuksessa löydetään laajempaa käyttöä tiedonlouhinnalle kuin sen vastineissa liikealalla, koska korkeakouluilla on kolme tiedonlouhintavaltaista tehtävää: tieteellinen tutkimus, joka on tiedon luomista, opetus, joka on tiedon välitystä, ja institutionaalinen tutkimus, joka koskee tiedon käyttöä päätöksenteossa. Kaikki edellä mainitut tehtävät kuuluvat tietämyksenhallinnan piiriin, joka ajaa tarvetta parempiin ja nopeampiin päätöksenteon työkaluihin ja menetelmiin. [Luan 2002]

Melkein kaikki tiedonlouhinnan algoritmit ja mallit, joita käytetään liikealalla, ovat suoraan käytettävissä tutkimukseen korkeakouluopetuksessa, erityisesti institutionaalisessa tutkimuksessa [Luan 2002]. Kuitenkin osa tutkijoista on sitä mieltä, että geneeriset tiedonlouhintatekniikat tai algoritmit eivät ole sopivia käytettäväksi EDM:ssä [Dutt *et al.* 2015]. Onkin ehdotettu, että EDM:ssä käytetyt menetelmät ovat usein erilaisia kuin normaalit tiedonlouhintamenetelmät, koska data on erilaista. Siinä voi olla monitasoista hierarkiaa ja riippuvuuksia, joista täytyy olla tietoinen, mutta toisaalta, joita voi myös hyödyntää. [Baker and Yasef 2009] Esimerkiksi digitaalisessa oppimisessa joitakin perinteisiä louhintatekniikoita voidaan soveltaa, joitakin ei [Romero and Ventura 2007]. EDM:n menetelmät tulevat siksi myös tiedonlouhinnan lisäksi muilta aloilta kuten koneoppimisesta, psykometriasta ja muilta tilastotieteen aloilta [Baker and Yasef 2009]. Lisäksi tarvitaan informaation visualisointia (information visualisation) ja laskennallista mallintamista (computational modeling) helpottamaan suuren tietomäärän analysointia [Baker and Yasef 2009, Romero and Ventura 2007].

3.1 Koulutusdatan louhinnan sovellusalueet

EDM voidaan jakaa korkealla tasolla neljään eri sovellusalueeseen, joista yksi pääkäyttöalue on opiskelijamallien parantaminen. Opiskelijamallit esittävät tietoa opiskelijan piirteistä tai tilasta, esimerkiksi opiskelijan sen hetkisestä tietämyksestä, motivaatiosta ja asenteista. Tutkijat ovat pystyneet laajentamaan opiskelijoiden mallintamisen kohti tekijöiden päättelemistä, jotka ennustavat opiskelijoiden kurssien tai opintojen keskeyttämistä tai kursseissa tai opinnoissa epäonnistumista. [Baker and Yasef 2009]

Toinen EDM-menetelmien käyttökohde on ollut koulutusalan tietostruktuurimallien kehittäminen. Niiden avulla on pystytty automatisoimaan tarkkojen struktuurimallien löytäminen suoraan datasta. Kolmas pääalue on ollut pedagogisen tuen tutkiminen, jotta voidaan havaita minkä tyyppinen oppimisen tukeminen on tehokkainta yleensä, eri tilanteissa ja eri ryhmille opiskelijoita. Neljäs sovellusalue on ollut empiiristen todisteiden löytäminen kasvatustieteiden teorioiden ja tunnettujen ilmiöiden jalostamiseen ja laajentamiseen, jotta saataisiin syvempi ymmärrys avaintekijöistä, jotka vaikuttavat oppimiseen ja jotta voidaan suunnitella parempia oppimisjärjestelmiä (learning systems), kuten erilaisia ympäristöjä tai välineitä oppimista varten. [Baker and Yasef 2009]

Tiedonlouhinnan soveltaminen voidaan orientoida eri tekijöiden mukaan, joilla on oma näkökulmansa; opiskelijat, opettajat ja vastuussa olevat opettajat ja tutkijat sekä hallinnossa työskentelevät. Opiskelijalle voidaan esimerkiksi yksittäisellä kurssilla suositella tehtäviä ja resursseja, jotka tukevat ja parantavat hänen oppimistaan. Näitä annetaan sen mukaan mitä hän on jo tehnyt ja miten hän on onnistunut ja myös sen perusteella mitä samankaltaiset oppijat ovat tehneet. [Romero and Ventura 2007] Tutkintotasolla opiskelijalle voidaan suositella vastaavasti kursseja [Wong 2018].

Opettajat voivat käyttää tiedonlouhinnalla löydettyä hyödyllistä informaatiota antamaan pedagogisen pohjan heidän päätöksiinsä suunnitellessaan ja muokatessaan oppimisympäristöä tai opetustapaa. Louhinnan tavoitteina on muun muassa saada enemmän objektiivista palautetta kurssista ja pystyä luokittelemaan oppijat sen perusteella miten paljon ohjausta ja tarkkailua he tarvitsevat. [Romero and Ventura 2007]

Vastaaville opettajille ja tutkijoille sekä hallinnossa työskenteleville tiedonlouhinta voi tarjota esimerkiksi tietoa siitä, miten resursseja voi kohdentaa paremmin ja miten parantaa kurssitarjontaa ja tutkinto-ohjelmia. He voivat louhinnan avulla ennakoida kurssille ilmoittautuneiden määrän aikataulutusta varten ja saada tietoonsa ketkä opiskelijoista ilmoittautuvat millekin kursseille ja ketkä tarvitsevat tukea valmistuakseen. [Romero and Ventura 2007] Tiedonlouhinnalla voidaan selvittää myös ne alumnit, jotka tekevät todennäköisimmin lahjoituksen tai osallistuvat alumnitoimintaan [Luan 2002].

Jotkin koulutusinstituutiot ovat jo tietoisia, että aikainen päätelmä opiskelijan mahdollisesta akateemisesta suoriutumisesta voi mahdollistaa opiskelijoille korkeampia akateemisia saavutuksia. Tämä voi johtaa eri toimien suunnitteluun, jotka kohdistuvat eri opiskelijaryhmiin heidän potentiaalinsa mukaan ja tehokkaampaan instituution resurssien kohdentamiseen. Opiskelijoiden suoriutumisen ennustaminen on yksi EDM:n suosituin ja hyödyllisin sovellusalue. Se muodostuu opiskelijan suoriutumisen, pistemäärän tai arvosanan tuntemattoman arvon päättelemisestä. Tämä on haastava ongelma ratkaista johtuen suuresta määrästä seikkoja, jotka voivat vaikuttaa opiskelijan suoriutumiseen, kuten sosioekonominen asema, entinen oppilaskokemus, vuorovaikutus muiden kanssa, demografiset ominaisuudet, psykologinen profiili ja kulttuuritausta. Opiskelijan suoriutumista on kirjallisuudessa ennustettu yksittäisen kurssin tasolla. Vähemmän on tehty tutkimusta opiskelijan suoriutumisesta tutkintotasolla. Joissain näistä tutkimuksista yritetään ennustaa saako opiskelija tutkinnon valmiiksi, kun taas toisissa pyritään päättelemään opiskelijan loppuarvosana. Moni muu tutkimus keskittyy opiskelijan suoriutumisen ennustamiseen ensimmäisen opiskeluvuoden lopussa. Suoriutumisen ennustamisessa käytetyimmät tekniikat ovat luokittelu ja regressio. [Miguéis *et al.* 2018]

Koulutuslaitosten ilmoittautumisdatan louhinta on myös yksi EDM:n avainalueista. Louhinnalla voidaan avustaa ilmoittautumisten hallinnointia (enrollment management),

johon kuuluu usein korkeakoulun markkinointi, sisäänpääsykäytännöt, ohjelmat opiskelijoiden pysyvyyden saavuttamiseen ja opiskelijoiden taloudellinen avustaminen. Näillä koulutuslaitokset pyrkivät vaikuttamaan opiskelijoiden ilmoittautumisiin. [Goyal and Vohra 2012] Esimerkiksi kun markkinointia avustetaan tiedonlouhinnalla, voivat korkeakoulut saada enemmän ilmoittautumisia mahdollisilta opiskelijoilta [Luan 2002].

Oppimisen tutkiminen on myös EDM:n avainalue [Goyal and Vohra 2012]. Tiedonlouhinnan avulla voidaan evaluoida ja kehittää digitaalisia oppimisjärjestelmiä (e-learning systems), kun löydetään hyödyllistä tietoa opiskelijoiden oppimisesta, josta voidaan muodostaa oppimisportfolio (learning portfolio). Louhinnalla saadaan myös muodostettua malli opiskelijan oppimisprosessista ja opiskelijasta itsestään. Verkkopohjaisista oppimisympäristöistä saadaan paljon dataa opiskelijan oppimiseen liittyen, sillä opiskelijan käyttäytymistä ympäristössä voidaan tallentaa. Tällaisen datan automaattiseen analyysiin on kasvava kiinnostus. Käytettyjä tekniikoita ovat muun muassa klusterointi, luokittelu, assosiaatioanalyysi, sekvenssien louhinta (sequential pattern mining) ja tekstin louhinta. [Romero and Ventura 2007]

EDM:ssä on myös tehty tärkeää tutkimusta opiskelijoiden pysyvyydestä, eli jatkavatko he opinnoissaan vai eivät, ja opintojen lopettamistasosta [Dutt *et al.* 2015]. On myös tutkittu yksittäisen kurssin tasolla mitkä tekijät vaikuttavat opiskelijoiden epäonnistumiseen ja keskeyttämiseen [Baker and Yasef 2009]. Tiedonlouhinnan avulla voidaan myös helposti ryhmitellä opiskelijoita muun muassa heidän henkilökohtaisten ominaisuuksiensa ja kustomoitujen piirteidensä perusteella eri tarkoituksia varten. Ryhmittelyssä louhintatekniikoina käytetään luokittelua ja klusterointia. Käytettyjä klusterointialgoritmeja ovat muun muassa kokoava hierarkkinen klusterointi, klusterointialgoritmi K-means ja mallipohjainen klusterointi (model-based clustering). [Goyal and Vohra 2012]

3.2 Haasteet

EDM:ssä ei kuitenkaan ole kyse vain datan muuntamisesta tiedoksi vaan myös louhitun tiedon suodattamisesta päätöksen tekoa varten [Romero and Ventura 2007]. On erittäin tärkeää, että organisaatiot omaksuvat data-analyysiin pohjautuvat kriittiset päätökset kuin, että laskevat kriittiset päätökset työntekijöidensä päiden varaan. Toisin sanoen joskus on tärkeämpää tutkia piilotettua tietoa organisaation datasta ja muuntaa se eksplisiittiseksi tietämykseksi parantamaan päätöksentekoprosessia kuin luottaa kokemukseen tai nyrkkisääntöihin. [Natek and Zwilling 2014]

Kuitenkin yliopistoissa kerätty data ei ole alun perin suunniteltu päätöksentekoanalyysia varten ja tämä johtaa useisiin eri dataformaatteihin tai skeemoihin, joita on hankala käsitellä ja prosessoida olemassa olevilla louhintatekniikoilla. Siksi on kriittisen tärkeää muuntaa ja esikäsitellä dataa, jotta syötedata on sopivaa koneoppimisen menetelmiin. Li-

säksi datan puhdistus ja puuttuvien arvojen täydentäminen ovat välttämättömiä, jotta voidaan tuottaa tarkkoja ja mielekkäitä malleja, esimerkiksi opiskelijan suoriutumisesta. [Trandafili *et al.* 2012]

EDM:ssä yksi suurimpia ongelmia on, että datajoukot ovat suhteellisen pieniä mielenkiintoisten hahmojen löytämistä varten [Dutt *et al.* 2015]. Kuitenkin oikeassa elämässä on paljon akateemisia tilanteita, joissa suhteellisen pienet datajoukot ovat normaaleja, ainakin korkeakoulun näkökulmasta. Opiskelijoiden käymistä kursseista kerätty data on hyvä esimerkki tällaisesta tilanteesta. Vaikka kurssilla kävisi suhteellisen iso määrä opiskelijoita, merkityksellistä dataa on silti pieni määrä. [Natek and Zwilling 2014]

Hyvin tunnettu fakta on, että tiedonlouhintatekniikat toimivat parhaiten suurella määrällä dataa. Useat tutkijat ovatkin päätyneet tulokseen, että pienet datajoukot rajaavat tiedonlouhintatekniikan ulottuvuutta. Tutkimustulokset pienten opiskelijadatajoukkojen louhinnasta kuitenkin tukevat selvästi johtopäätöstä, että tiedonlouhinta ei ole yleisesti rajoitettu suuriin datajoukkoihin, vaikka edeltävä tutkimus ei olettanut tällaista löytöä. Tiedonlouhintatekniikoiden tietynlainen käyttö pieniin rakenteellisiin datajoukkoihin voi myös tarjota käyttökelpoisia tuloksia. [Natek and Zwilling 2014]

3.3 Työkalut

Yleistä tiedonlouhintaa varten on kehitetty monia kaupallisia ja akateemisia työkaluja kuten Clementine (nykyään IBM SPSS Modeler), Intelligent Miner, Weka, Orange ja RapidMiner [Romero and Ventura 2007, IBM SPSS Modeler 2020, Intelligent Miner 2020, Weka 2020, Orange 2020, RapidMiner 2020]. Työkalut tarjoavat käyttöön louhinta-algoritmeja ja suodatus- ja visualisointitekniikoita. Näitä työkaluja ei ole kuitenkaan suunniteltu erityisesti pedagogisia tarpeita varten ja siksi ne ovat hankalia käyttää esimerkiksi opettajille, joilla ei ole laajaa tietämystä tiedonlouhinnasta. [Romero and Ventura 2007] Työkalujen ominaisuudet ovat liian laajoja siihen nähden mitä opettaja haluaisi tehdä. Yksi mahdollinen ratkaisu on kehittää työkaluja, jotka käyttävät jokaiseen tehtävään oletusalgoritmia ja parametrittomia tiedonlouhinta-algoritmeja yksinkertaistaakseen konfigurointia ja käyttöä ei-asiantuntevalle käyttäjälle. Toisekseen tiedonlouhintatyökalun olisi hyvä olla integroitu digitaaliseen oppimisympäristöön, tai muuhun järjestelmään, jotta louhintatekniikoilla saatuja tuloksia voitaisiin käyttää helposti ja suoraan. Lisäksi nykyiset työkalut, joiden datan louhinta liittyy tiettyihin kursseihin saattavat olla hyödyllisiä vain niiden kehittäjille. [Goyal and Vohra 2012] Joitakin erityisesti EDM:ään tarkoitettuja työkaluja on kehitetty, mutta ei ole olemassa yleisiä työkaluja tai niiden uudelleenkäyttöä, jotta niitä voitaisiin käyttää missä tahansa koulutusjärjestelmässä [Romero and Ventura 2007, Goyal and Vohra 2012]. Tästä syystä syöte- ja tulostedatan mallit tarvitsisivat standardisointia [Goyal and Vohra 2012].

Wekaa pidetään pätevänä analyysityökaluna, kun käsitellään koulutuksen hallintadataa ja lupaavana työkaluna datan analysoinnissa ja tulkinnassa parantamaan tietämyksenhallintaa organisaatioissa. Se vaikuttaa sopivan myös pienten datajoukkojen evaluointiin ja analysointiin erilaisissa tehtävissä. [Natek and Zwilling 2014] Weka on ilmainen ja avoimen lähdekoodin ohjelmisto, jossa algoritmeja pystyy käyttämään suoraan datajoukkoon tai algoritmeja voi kutsua omasta Java-koodista [Natek and Zwilling 2014, Aher and Lobo 2013]. Wekan kehitys aloitettiin vuonna 1992 Uuden-Seelannin valtion rahoittamana Waikaton yliopistossa. Weka on lyhenne sanoista *the Waikato Environment for Knowledge Analysis* ja on myös nimi Uudessa-Seelannissa tavattavalle linnulle, joka on ohjelmiston logossakin. Weka on kirjoitettu Javalla ja se sisältää kattavan kokoelman koneoppimisen algoritmeja, datan esikäsittelytyökaluja ja datan visualisointia. Algoritmeja on regressioanalyysiin, luokitteluun, klusterointiin, assosiaatiosääntöjen louhintaan ja merkittävien attribuuttien valintaan (attribute selection). Weka on laajalti hyväksytty akateemisissa ja liikealan piireissä ja sitä käytetään laajalti tiedonlouhinnan tutkimuksessa. [Hall *et al.* 2009]

Wekan lisäksi tässä työssä käytetään koneoppimisen ja tiedonlouhinnan yleistyökalua, Orangea. Sen monikerroksinen arkkitehtuuri sopii erilaisille käyttäjille tiedonlouhinnan aloittelijoista ohjelmoijiin, jotka käyttävät Orangen toiminnallisuuksia kutsumalla sitä suoraan Python-koodista. Suurin osa käyttäjistä hyödyntää Orangea kuitenkin sen graafisen käyttöliittymän kautta, mikä on pohjimmiltaan visuaalista ohjelmointia. Orangen kehitys alkoi vuonna 1997 Ljubljanan yliopistossa, Sloveniassa. Sen perusominaisuuksiin kuuluu useita koneoppimisen ja datan esikäsittelyn ja visualisoinnin algoritmeja. Toisin kuin esimerkiksi Weka, joka tarjoaa kaikkea mahdollista koneoppimiseen, Orangen päämääränä on ollut toteuttaa hyödyllisimpiä ja yleisesti käytetyimpiä tekniikoita joustavasti ja käyttäjäystävällisesti. Sen pääpaino on datan tutkimisessa. Orangea on käytetty niin tieteessä, teollisuudessa kuin opetuksessakin. Tieteellisesti sitä on käytetty uusien koneoppimisen algoritmien testiympäristönä, kuin myös uusien laskennallisten lähestymistapojen toteuttamiseen molekyylibiologiassa ja bioinformatiikassa. [Demšar and Zupan 2013]

3.4 Kurssien suosittelu

Korkeakouluopiskelussa opiskelijoiden tarvitsee valita kursseja, jotka sopivat tutkinnon valmistumisedellytyksiin [Wong 2018]. Yliopiston opetussuunnitelma on yleensä joustava. Opetussuunnitelma yliopistotutkinnon saamiseen vastaa useita kursseja, jotka on jaettu akateemisille jaksoille (lukukausille). Ennen jokaisen jakson/lukukauden alkua opiskelijan pitäisi ilmoittautua yhdelle tai useammalle kurssille, joista osa on pakollisia ja toiset ovat vapaavalintaisia, vastaten opiskelijan etenemistä. Kurssien sarjaa, joihin opiskelija on ilmoittautunut jokaisena lukukautena opiskelunsa aikana, kutsutaan opiskelijan akateemiseksi kulkureitiksi (academic itinerary). Akateeminen kulkureitti on onnistunut,

kun opiskelija saa hyvät tulokset jokaisesta kurssista, johon on ilmoittautunut, mikä tekee täten mahdolliseksi opiskelijan valmistumisen ajallaan ja hyvillä arvosanoilla. [Vialardi *et al.* 2009] Optimaalisen kurssien valinnan määräytymisessä tarvitsee ottaa huomioon henkilökohtaiset, institutionaaliset ja ulkoiset tekijät ja tätä määrittämistä voisi kuvailla aikatauluttamiseksi, graafien navigoimiseksi ja optimointiongelmaksi. Henkilökohtaiset tekijät pitävät sisällään oppijan valmiuden, hylätyt kurssit, työpaineet, terveysongelmat, sosiaaliset verkot, korvatut opinnot ja taloudelliset syyt. Institutionaalisiin tekijöihin kuuluvat kurssirajoitukset, kuten edeltävät kurssit, kurssien saatavuus ja ilmoittautumiskiintiöt. Ulkoisia tekijöitä ovat regulatiiviset(sääntely-) ja akkreditointi(hyväksyntä)vaatimukset. [Wong 2018]

Opiskelijan kurssille ilmoittautuminen riippuu ainoastaan hänen päätöksestään [Vialardi *et al.* 2009]. Kurssien valintatapoja on monia, kuten opaskirjat, verkkopalvelut ja henkilökohtainen avustus akateemiselta ohjaajalta. Eksplisiittistä ja implisiittistä tietoa kurssivalintojen tekemiseen saattaa olla eri järjestelmissä ja verkkosivuilla, mutta tieto ei ole personoitua, kuten ei ole opaskirjoissakaan. [Wong 2018] Opiskelijat voivat pyytää neuvoa akateemisilta ohjaajilta (tai professoreilta) tietääkseen kuinka monta ja mitä saatavilla olevista kursseista heidän tulisi ottaa pohjautuen heidän opintosuorituksiinsa [Vialardi *et al.* 2009]. Ohjaajat käyttävät usein kokemusta ja heuristiikkaa määrittääkseen parhaan mahdollisen polun näille opiskelijoille. Kuitenkin tämä voi johtaa epäjohdonmukaiseen neuvontaan, jos mukana on useita ohjaajia, eikä henkilökohtainen ohjaus ole skaalattavissa tuhansille opiskelijoille. [Wong 2018]

Osa opiskelijoista ei kysy neuvoa ohjaajilta tai kysyvät vain harvoin, jolloin ilmoittautuminen perustuu ainoastaan opiskelijan kokemukseen ja saatavilla olevaan informaatioon. Monilla opiskelijoista ei kuitenkaan ole tarpeeksi kokemusta ilmoittautumispäätösten tekemiseen, koska he eivät osaa yhdistää aikaa, työtä ja älyllisiä taitoja, joita kurssin onnistunut suoritus vaatii. Monesti valintakriteerit liittyvät läheisesti aikaan, jossa saada opinnot loppuun, suurelta osin opiskelijoiden elämäntilanteen vuoksi. Epäilemättä yliopistot tarjoavat tarvittavan kvantitatiivisen informaation, kuten saatavilla olevat kurssit, osa-alueet, luokkahuoneet ja professorit, mutta laadullista tietoa eli implisiittistä informaatiota entisten opiskelijoiden kokemuksista ei ole. [Vialardi *et al.* 2009]

Automatisoitu kurssien suosittelija on yksi menettelytapa skaalata ohjaus suurelle joukolle opiskelijoita [Wong 2018]. Esimerkiksi kurssien suosittelujärjestelmä digitaalisessa oppimisessa suosittelee parhaan yhdistelmän oppiaineita, joista opiskelijat ovat kiinnostuneita. Digitaalisen oppimisen järjestelmistä on tehty lukuisia tutkimuksia, jotka keskittyvät opiskelijan käyttäytymiseen ja oppimiseen. [Aher and Lobo 2013] Suosittelujärjestelmien käyttöä koulutusympäristössä auttamaan opiskelijoita päätöksenteossa on tutkittu ja kehitetty vain vähän [Vialardi *et al.* 2009].

Olemassa olevat koulutuksessa käytettävät suosittelujärjestelmät neuvovat opiskelijoita monella tasolla, korkean tason uravalinta-, tutkinto- ja pääainesuosituksista sisäisiin suosituksiin oppimistavoitteista kurssilla. Nämä järjestelmät käyttävät yhteistoiminnallista suodattamista (collaborative filtering), sisältöperustaista (content-based) suosittelua, assosiaatiolouhintaa (association mining) tai näiden tekniikoiden yhdistelmiä. Yhteistoiminnallinen suodattaminen tarkoittaa että, tarjotaan suosituksia opiskelijan profiilin ja menneiden valintojen tai muiden opiskelijoiden valintojen perusteella. [Wong 2018] Toiset opiskelijat ovat usein samankaltaisia akateemisen suorituksen ja muun henkilökohtaisen informaation osilta [Vialardi *et al.* 2009]. Sisältöperustainen (content-based) suosittelu on opiskelijan edellisten valintojen käyttämistä yhdessä kurssien metadatan kanssa suositusten antamiseen, ja assosiaatiolouhinnalla etsitään suosittuja kurssihahmoja [Wong 2018]. Klusterointia käytetään useimmiten suosittelujärjestelmissä tunnistamaan samankaltaiset opiskelijat [Asadi *et al.* 2019].

Kurssien suosittelijalla on omat haasteensa verrattuna klassisiin suosittelujärjestelmiin. Toisin kuin klassisten suosittelujärjestelmien, kurssien suosittelijan täytyy tarjota sarja rinnakkaisia kursseja ennemmin kuin yksittäinen kurssi. Järjestelmän täytyy ottaa huomioon käytyjen kurssien järjestyshistoria tarjotakseen parhaan mahdollisen suosituksen. Kurssien suosittelujärjestelmän tarvitsee ottaa huomioon myös monimutkaiset rajoiteongelmat kuten kurssien aikataulut, kurssien saatavuus, kurssien kiintiöt, opetusohjelman rajoitukset (edeltävyydet, yhdessä otettavat kurssit, sisällöllisesti päällekkäiset kurssit, kurssille pääsyn säännöt) ja tutkintosäännöt kuten pakollisten ja vapaavalintaisten kurssien vaatimukset. Järjestelmään voidaan tarvita lisäkriteerejä kuten opiskelijoiden uratavoitteet, akkreditointivaatimukset, henkilökohtaiset mieltymykset ja muut tilanteesta riippuvat rajoitteet (kuten stipendien rajoitukset jne). Tutkinto-ohjelmat ja kurssit kehittyvät jatkuvasti, joten kurssien historiallinen vastaavuus täytyy ottaa huomioon suosittelujärjestelmässä. Kurssi voi myös olla osa eri tutkintoa tai pääainetta, mikä vaatii järjestelmää mukauttamaan suositukset sopiviksi opiskelijan tilanteeseen. [Wong 2018]

Edellä mainittuja haasteita varten Wong kehitti prototyypin sekvenssipohjaisesta kurssien suosittelijasta, joka hyödyntää syväoppimisen tekniikkaa, neuroverkkoa Long Short-Term Memory (LSTM) Recurrent Neural Networks. Neuroverkolla muodostettu malli harjaannutetaan joka lukukausi uudelleen, jotta suosittelijan antamat ehdotukset pysyvät mahdollisten muutosten tasalla. [Wong 2018]

Erilaisia ja myös yksinkertaisempia tiedonlouhintamenetelmiä on käytetty kurssien suosittelujärjestelmiä kehitettäessä. Esimerkiksi Aher ja Lobo kehittivät kurssien suosittelujärjestelmän Moodlen yhteyteen, joka suosittelee kursseja opiskelijalle toisten opiskelijoiden ottamien kurssien perusteella. Opiskelija saa ehdotuksia, kun on ilmoittautunut vähintään viidelle kurssille ja kurssi on mukana suosituksissa, jos sen on valinnut vähin-

tään sata opiskelijaa. Tämä kurssien suosittelujärjestelmä auttaisi opiskelijoita valitsemaan sopivan yhdistelmän kursseja heidän kiinnostuksensa mukaan. Tällainen lähestymistapa kurssien suositteluun uusille opiskelijoille voi olla hyödyllinen MOOC eli Massively Open Online Courses -ympäristössä. Tutkimuksessa käytetty data kerättiin Moodlesta ja koostuu riveistä, joissa on opiskelijan tunniste ja attribuutteina eri kurssit, joissa arvoina kurssille ilmoittautuminen (’yes’, ’no’). Tiedonlouhintatyökaluna käytössä oli Weka. Louhinnassa testattiin erilaisia menetelmiä, kuten pelkkää Apriori-assosiaatiosääntöalgoritmia ja Apriorin ja eri klusterointialgoritmien yhdistelmiä. Testattuja klusterointialgoritmeja olivat K-means, Expectation-Maximization ja Farthest First. [Aher and Lobo 2013]

Aherin ja Lobon tutkimuksessa yhdistetyn algoritmin, K-means ja Apriori, tulokset olivat parempia kuin tulokset pelkällä Apriorilla ja toisilla yhdistelmillä klusterointia ja assosiaatioanalyysiä. Pelkän Apriorin käyttö vaati datan esikäsittelyä ja jos tukea kasvatettiin, saatiin vähemmän assosiaatiosääntöjä. Muiden klusterointialgoritmien, Expectation-Maximization ja Farthest First, ja Apriorin yhdistelmien tulokset eivät täsmänneet opiskelijoiden oikeiden valintojen kanssa. Mutta jos käytettiin K-means ja Apriorin yhdistelmää, ei tarvittu datan esikäsittelyä, kuten pelkän Apriorin kanssa, ja assosiaatiosääntöjä saatiin enemmän ja ne olivat vahvempia. Tämä yhdistelty lähestymistapa valittiin käytettäväksi louhintamenetelmäksi kurssien suositteluun. Siinä data klusteroidaan ensin K-means -algoritmilla, jolle annetaan parametreinä siemenluku ja klustereiden määrä (sopiva siemenluku ja klustereiden määrä etsitään testaamalla). Kun data on klusteroitu, käytetään Apriori -algoritmia jokaiseen klusteriin. Sen klusterin tulos valitaan suosittelemaan kursseja, jossa on oikeinlainen tulos eli assosiaatiosäännöt, joiden arvoina on ’yes’. [Aher and Lobo 2013] Aher ja Lobo kokeilivat toisessa tutkimuksessaan myös klusteroinnin (K-means), luokittelun (ADTree) ja assosiaatioanalyysin (Apriori) yhdistelmää kurssien suosittelussa samankaltaiseen dataan, ja totesivat sen toimivan paremmin kuin pelkän Apriori-algoritmin käytön [Aher and Lobo 2012].

Vialardi ja muut ovat kehittäneet yhteistoiminnallisen suosittelujärjestelmän, joka perustuu tiedonlouhintatekniikoihin ja jota käytetään koulutusympäristössä suosittelemaan kursseja. Järjestelmän on tarkoitus ennustaa, kuinka sopivaa tietyn opiskelijan on ottaa tietty kurssi. Pohjana käytetään toisten samankaltaisen opiskelijoiden tuloksia, jotka ovat jo käyneet kurssin. Tarkoitus on, että opiskelija saisi hyviä tuloksia jokaisella kurssilla hänen akateemisella kulkureitillään. Data koostui jokaisen opiskelijan demografisesta informaatiosta, kurssi-ilmoittautumisista, saaduista arvosanoista, kurssien määrästä jokaisena lukukautena, arvosanojen keskiarvosta ja kumulatiivisesta arvosanasta per lukukausi. He päätyivät käyttämään attribuutteina opiskelijan kurssi-ilmoittautumisten kokonaismäärää, kurssin nimeä, kumulatiivista arvosanaa lukukauden alussa ja kurssista saa-

tua arvosanaa, joka toimii luokkatunnuksena, kun se on muutettu joko arvoksi onnistuminen (success) tai epäonnistuminen (failure). Yksi tapaus datassa on yhden opiskelijan yksi suoritus. Datan suodattamisen ja puhdistamisen jälkeen käytettiin luokittelualgoritmia C4.5, josta saatiin säännöt, joita käytetään järjestelmässä ehdottamaan opiskelijalle, onko hänellä hyvä mahdollisuus onnistua kurssilla, johon ilmoittautuisi. Tällä tavalla opiskelijoilla on tukityökalu, joka auttaa heitä tekemään parhaat päätökset ennen ilmoittautumista. Kuitenkin luokittelun pitäisi ottaa huomioon, että oikeissa tilanteissa olosuhteet voivat muuttua vuodesta toiseen ja luokittelu voi kuvastaa ”vanhoja” malleja ja kuten minkä tahansa luokittelun kanssa, ennusteissa on odotettavissa virhemarginaali. On huonompi suositella opiskelijaa ottamaan kurssi, jota hän ei pääse läpi kuin, että ei suositella menemään kurssille, jonka hän voisi läpäistä. Noin 80% tapauksista järjestelmä osasi ennustaa opiskelijan suoriutumisen oikein. [Vialardi *et al.* 2009]

Bendakir ja Aïmeur kehittivät kurssien suosittelujärjestelmän RARE, jossa hyödynnetään Apriori-algoritmia suositusten muodostamisessa ja käytetään louhintatyökaluna Wekaa. RARE suosittelee kursseja opiskelijalle muiden opiskelijoiden aiempien kurssivalintojen perusteella. Järjestelmään kuuluu myös kurssien arviointi, joka vaikuttaa kurssien suositteluun. [Bendakir and Aïmeur 2006] Assosiaatioanalyysiä on hyödynnetty myös Zhangin ja muiden kurssien suosittelujärjestelmässä MCRS, joka on tarkoitettu MOOC-ympäristöille, joissa datamäärät ovat paljon suurempia kuin perinteisessä koulutusympäristössä [Zhang *et al.* 2018].

Asadin ja muiden kehittämässä kurssien suosittelumallissa klusterointia käytetään tunnistamaan opiskelijat, joilla on samankaltaiset kiinnostuksen kohteet ja taidot. Samankaltaisten opiskelijoiden kurssivalintoja tutkitaan sumealla assosiaatiosääntöjen louhinnalla (fuzzy association rule mining). Suosittelumallissa ennustetaan myös kurssin arvosana. Tutkimuksessa on myös koottu kattavasti yhteen, millaisia menetelmiä kurssien suosittelussa on viime aikoina käytetty. [Asadi *et al.* 2019]

4 Louhintatehtävät ja data

Tässä työssä louhinnalla pyritään saamaan opettajille ja ohjaajille tietoa opiskelijoiden kiinnittymisestä opintoihinsa ja opiskelijoille tietoa heille sopivista kurssivalinnoista opintojen alussa. Kurssien suosittelu sopii myös myöhempään vaiheeseen opiskelijan opintoja. Opintoihin kiinnittymistä tutkitaan ryhmittelemällä opiskelijat heidän ensimmäisen lukuvuotensa opintosuoritusten perusteella ja etsimällä suositut kurssihahmot ja moniko opiskelija on suorittanut ne. Myös ilman korvattavia opintoja aloittavia opiskelijoita verrataan opiskelijoihin, joilla on jo takana korkeakouluopintoja, joita he ovat hyväksilukeneet tutkintoonsa, ja selvitetään, millaisia opintoja korvataan ja mitä opintoja tehdään hyväksiluettujen lisäksi. Kurssihahmojen louhintaa ja opiskelijoiden ryhmittelyä voidaan hyödyntää myös myöhemmässä vaiheessa opiskelijan opintoja.

Louhinnassa käytettävä data on poimittu Tampereen yliopiston opiskelijatietojärjestelmästä. Poiminnan kohteena olivat vuosina 2017 ja 2018 Tampereen yliopistossa tietojenkäsittelytieteiden tutkinto-ohjelmassa aloittaneet opiskelijat ja heidän ensimmäinen lukuvuotensa yliopistossa. He ovat voineet aloittaa joko alemmassa tai ylemmässä korkeakoulututkinnossa ja ilmoittautua läsnä- tai poissaoleviksi. Data koostuu näiden opiskelijoiden tutkinto- ja ilmoittautumistiedoista yhdessä listassa sekä ensimmäisen lukuvuoden opintosuorituksista toisessa listassa. Opiskelijan tutkinnon tiedot ovat aloitusvuosi, opinto-oikeus ja koulutusohjelma. Opiskelijalla voi olla opinto-oikeus sekä alempaan (luonnontieteiden kandidaatti, LuK) että ylempään korkeakoulututkintoon (filosofian maisteri, FM) tai vain ylempään. Ilmoittautumistieto tarkoittaa, onko opiskelija ilmoittautunut läsnä- vai poissaolevaksi ensimmäiselle syys- ja kevätlukukaudelle erikseen. Yksi opintosuoritus sisältää opiskelijan pseudotunnisteen, suoritetun kurssin tunnuksen, nimen, arvosanan, suorituspäivän (tai hyväksilukupäivän) ja kurssista saadut opintopisteet. Suoritusmerkinnässä on myös tieto siitä, onko kurssi osittain tai kokonaan korvattu eli hyväksiluettu esimerkiksi jollain opiskelijan aiemmin eri oppilaitoksessa suorittamalla kurssilla. Korvatusta kurssista lisätietoja ovat alkuperäinen suorituspaikka, alkuperäinen suorituspäivä ja muut lisätiedot. Louhintatehtävän mukaan datan muotoa täytyy muuttaa sekä valita tehtävään sopivat attribuutit.

Datassa opiskelijoita on yhteensä 301 (vuonna 2017 150 opiskelijaa ja vuonna 2018 151 opiskelijaa), joista 20 opiskelijaa on ilmoittautunut poissaolevaksi koko lukuvuodelle eli heillä ei ole suoritusmerkintöjä ensimmäiselle lukuvuodelleen. Opiskelijoista 73 on tullut opiskelemaan vain ylempää korkeakoulututkintoa (vuonna 2017 30 opiskelijaa ja vuonna 2018 43 opiskelijaa). Suoritusmerkintöjä on yhteensä 3081, joista hyväksiluettuja suorituksia on 455. Suorituksia oli yhteensä 339 eri kurssista (kurssien koodien mukaan). Liitteessä 1 on esimerkkejä datan muodosta.

4.1 Opiskelijoiden kiinnittyminen opintoihinsa

Opiskelijat ryhmitellään heidän ensimmäisen lukuvuotensa opintosuoritusten perusteella käyttäen klusterointia. Kummallekin lukuvuodelle 2017-2018 ja 2018-2019 tarkastellaan millaisia ryhmiä (klustereita) saadaan. Opiskelijat jaetaan myös heidän opinto-oikeutensa mukaan eli tarkastellaan erikseen ylempää korkeakoulututkintoa suorittavia ja LuK- ja FM-tutkintoja suorittavia. Samalla etsitään parhainta klusterointialgoritmia tehtävää varten kokeilemalla hierarkkista klusterointia ja K-means ja EM-algoritmeja. Hierarkkinen klusterointi, käyttäen etäisyysmittoina Jaccardin kerrointa ja naapurikeskiarvoa, ja K-means -klusterointi tehdään työkalulla Orange ja EM-klusterointi tehdään Wekalla. Data muutetaan muotoon opiskelija per rivi, jossa attribuutteina ovat kurssit ja arvoina tieto suorittiko opiskelija kurssin. Jos hän on suorittanut kurssin, arvona on 1, muuten 0. Esimerkki datan muodosta on annettu taulukossa 1.

opiskelijan pseudotunniste	**TIEA2.1A**	**TIEA2.1B**	**TIEP1**	**TIEP3**	**TIEP5**
1	1	1	1	1	1
3	0	0	1	1	1
5	0	0	0	0	0

Taulukko 1. Esimerkki eri louhintatehtäviin tarvittavan datan muodosta.

Suositut kurssihahmot etsitään kattavilla joukoilla eli assosiaatioanalyysin avulla. Kattavista joukoista voidaan nähdä, moniko opiskelija on suorittanut tietyt kurssit. Kattavat joukot etsitään Wekalla Apriori-algoritmilla. Data on samassa muodossa kuin klusterointitehtävässä, mutta nollien tilalle päädyttiin kokeilujen jälkeen laittamaan puuttuvat arvot eli tyhjää. Tästä kerrotaan tarkemmin kohdassa 4.3. Esimerkki datan muodosta on annettu taulukossa 2. Suositut kurssihahmot etsitään sekä koko opintosuoritusdatasta ja lukuvuosista erikseen sekä opiskelijan opinto-oikeuden perusteella.

opiskelijan pseudotunniste	**TIEA2.1A**	**TIEA2.1B**	**TIEP1**	**TIEP3**	**TIEP5**
1	1	1	1	1	1
3			1	1	1
5					

Taulukko 2. Esimerkki datan muodosta kattavien joukkojen löytämiseen suorituksista.

Kun verrataan ilman korvattavia opintoja aloittavia opiskelijoita opiskelijoihin, joilla on jo takana korkeakouluopintoja, joita he ovat hyväksilukeneet tutkintoonsa, käytetään myös kattavia joukkoja. Kattavilla joukoilla saadaan selville, onko korvatuissa kursseissa toistuvia hahmoja ja mitä on suoritettu niiden lisäksi. Algoritmi, työkalu ja datan muoto

ovat samat kuin suosittuja kurssihahmoja etsiessä. Tätä tehtävää varten opintosuoritusdata jaetaan kahteen osaan sen perusteella, onko opiskelijalla korvattuja kursseja vai ei ja sen perusteella onko opintosuoritus korvattu vai ei.

4.2 Kurssien suosittelu

Kurssien suosittelussa vertaillaan eri suosittelumenetelmiä. Ensimmäinen on Aherin ja Lobon tapa, jossa käytetään klusterointia ja assosiaatioanalyysiä. Tästä on kerrottu tarkemmin kohdassa 3.4. Datan muoto on sama kuin taulukossa 1. Aherin ja Lobon käyttämien ’yes’ ja ’no’ arvojen tilalla käytetään vastaavasti ykköstä ja nollaa. Heidän valitsemansa algoritmien K-means ja Apriori (kummatkin Wekassa) tilalla käytetään Orangen K-means ja FP-growth -algoritmeja (Apriori ja FP-growth antavat samat tulokset).

Toinen suosittelumenetelmä on Vialardin ja muiden tapa, joka perustuu luokitteluun. Tästä on kerrottu tarkemmin kohdassa 3.4. Käytetty luokittelualgoritmi on C4.5, josta on Wekassa algoritmin Java-toteutus nimellä J48. Data on muodoltaan yhden opiskelijan suoritus per rivi, jossa on attribuutteina suoritettujen kurssien kokonaismäärä, kumulatiivinen arvosana lukukauden alussa, kurssin koodi ja kurssista saatu arvosana, joka toimii muunnoksen jälkeen luokkatunnuksena. Alun perin arvosana muutetaan joko arvoksi onnistuminen tai epäonnistuminen. Tämän työn datassa on vain hyväksyttyjä kursseja, joten selkeää rajaa onnistumiselle ja epäonnistumiselle ei ole. Kurssilla onnistuminen on myös subjektiivista, sillä opiskelijalle voi riittää joistain kursseista pelkkä läpipääsy. Tässä tapauksessa paremmat luokat olisivat erinomainen (ER, arvosana 5), hyvä (HV, arvosanat 3 ja 4), tyydyttävä (TY, arvosanat 1 ja 2) sekä hyväksytty (HYV). Datasta johdetaan attribuutit kurssien kokonaismäärä ja kumulatiivinen arvosana lukukauden alussa, joka lasketaan arvosanojen summa jaettuna kurssien määrällä. Esimerkki datan muodosta on annettu taulukossa 3.

opiskelijan pseudotunniste	**kurssien määrä**	**kumulatiivinen arvosana**	**kurssin koodi**	**tulos**
1	3	4	TIEP2.1A	HV
1	3	4	TIEP2.1B	HV
1	0	0	TIEP3	ER
3	0	0	TIEP1	TY
3	2	2,5	TIEP5	HV

Taulukko 3. Datan muoto luokittelupohjaista suosittelua varten.

4.3 Datan esikäsittely

Suurin datan esikäsittelyvaihe on datan muuntaminen kunkin louhintatehtävän kohdalla kuvattuun muotoon käyttäen apuna PostgreSQL-tietokantaa ja Microsoft Exceliä. Muut esikäsittelytoimet ovat datan yhtenäistäminen ja mahdollinen karsinta. Opintosuorituksissa on esimerkiksi kursseja, joista ei ole tullut opiskelijalle yhtään opintopistettä. Ne ovat usein osasuorituksia, jotka kuuluvat ylempään kurssisuoritukseen, josta on annettu opintopisteet. Tällaisia suorituksia on kuitenkin vain 48 kappaletta 3081 suoritusmerkinnässä, joten niitä ei tarvitse poistaa. Datassa on myös yleisesti muiden opiskelijoiden suorituksista poikkeavia suorituksia. Jos ne vaikuttavat louhinnassa saataviin tuloksiin negatiivisesti, ne on hyvä poistaa. Toisaalta nämä poikkeavat tapaukset voivat olla myös mielenkiintoisia, joten niitä ei kannata heti karsia. Data sisältää myös kursseja, joilla on eri kurssikoodit, mutta kyseessä on vastaava kurssi. Nämä on hyvä yhtenäistää niin, että niillä on sama koodi. Tästä esimerkkinä on kurssi Introduction to Academic English, jolla on kahta eri kurssikoodia KKENYHT ja KKENLUK. Näiden yhteiseksi kurssikoodiksi laitetaan KKENYHT/LUK.

Kurssista saatu tulos voi olla arvosana 1-5, merkintä HYV (hyväksytty), HV (hyvä) tai TY (tyydyttävä). Merkintöjä HV ja TY on käytetty vain opintojaksoilla ruotsin kielen kirjallinen viestintä ja ruotsin kielen suullinen viestintä. Sanalliset arvosanat täytyy muuntaa numeerisiksi esimerkiksi kumulatiivista arvosanaa laskettaessa. HYV muutetaan arvosanojen 1-5 keskiarvoksi eli arvosanaksi 3, HV arvosanojen 3-5 keskiarvoksi eli 4 ja TY arvosanojen 1-3 keskiarvoksi eli 2.

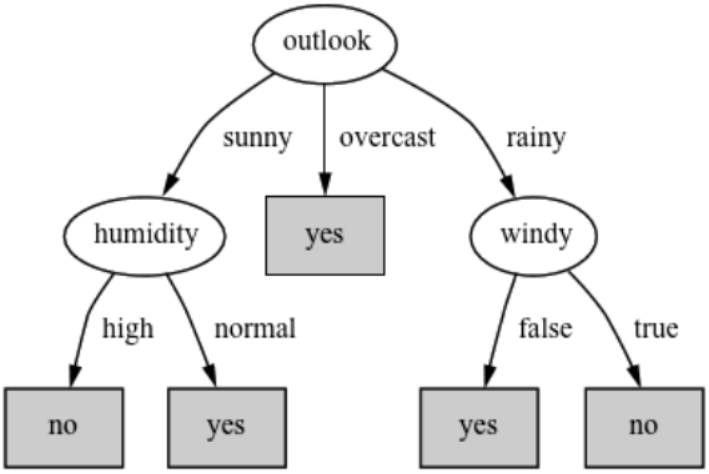
Suosittuja kurssihahmoja (kattavia joukkoja) louhittaessa ollaan kiinnostuneita kursseista, jotka on suoritettu (attribuutin arvo on 1). Jotta tällaiset kurssihahmot korostuisivat, voidaan datasta poistaa opiskelijat (rivit), jotka eivät ole suorittaneet mitään tai vain vähän opintoja sekä kurssit (attribuutit), joissa on vain muutama suorittaja. Tällä yritetään vähentää kattavia joukkoja, joissa attribuuttien arvot ovat 0. Tämä ei kuitenkaan tuonut suoritettuja kurssihahmoja tarpeeksi esille vaan suorittamattomat opinnot tulivat edelleen eniten esille. Tässä vaiheessa datasta poistettiin nolla-arvot ja tilalle jätettiin tyhjät arvot. Näin Apriori löysi suoritetut opinnot ja niistä kattavat joukot. Joissakin tiedonlouhintatyökaluissa on myös mahdollista saada halutut arvot esiin ilman tyhjien arvojen asettamista.

5 Louhinta-algoritmit

Tässä työssä käytettyjä algoritmeja tiedonlouhintaan ovat C4.5 luokitteluun, Apriori ja FP-growth assosiaatioanalyysiä varten ja klusterointiin K-means, EM (Expectation-Maximization) ja hierarkkinen klusterointi.

5.1 C4.5

C4.5 on luokitteluun tarkoitettu päätöspuualgoritmi. Sen on kehittänyt J. Ross Quinlan vuonna 1993 aiemmin kehittämästään päätöspuualgoritmista ID3, joka on laajennettu E.B. Huntin ja hänen kollegoidensa ideasta. Päätöspuupäättely on päätöspuun oppimista opetusjoukosta. Päätöspuussa on vuokaaviomainen puurakenne, jossa jokainen sisäsolmu (solmu, jolla on lapsisolmuja) merkitsee attribuuttiin tehtyä testiä ja jokainen oksa testin lopputulosta. Lehtisolmut (solmu, jolla ei ole lapsisolmuja) edustavat luokkatunnuksia. Ylin solmu on juurisolmu ja puu ”kasvaa alaspäin”. Kuvassa 2 on tyypillinen päätöspuu. Päätöspuu on luokittelumalli ja sen voi muuntaa helposti luokittelusäännöiksi. [Han *et al.* 2012]



Kuva 2. Esimerkki päätöspuusta, joka kuvaa hypoteettista sään vaikutusta jonkin pelin pelaamiseen [Witten *et al.* 2011].

C4.5 käyttää ahnetta lähestymistapaa, jossa päätöspuu muodostetaan ylhäältä alaspäin rekursiivisesti hajota ja hallitse -tyylillä. Opetusjoukko jaetaan rekursiivisesti pienempiin osajoukkoihin puuta muodostettaessa. Algoritmi tarvitsee kolme parametria: datan osajoukko *D*, lista attribuuteista ja attribuutin valintamitta. Aluksi *D* on koko opetusjoukko. Attribuutin valintamitalla valitaan datan jakokriteeri, joka ”parhaiten” jakaa datan *D* luokkatunnusten mukaisiin osiin. C4.5 käyttää mittoina informaation lisäystä (information gain) ja hyötysuhdetta (gain ratio) korjaamaan informaation lisäyksen aiheut-

tamaa vinoumaa. Informaation lisäys suosii attribuutteja, joista syntyy puuhun useita haaroja eli se suosii moniarvoisia attribuutteja. Hyötysuhde pyrkii normalisoimaan informaation lisäyksen. [Han *et al.* 2012]

Päätöspuun muodostaminen alkaa yhdestä solmusta N, joka esittää tapauksia joukossa D. Jos kaikki tapaukset kuuluvat samaan luokkaan, tulee solmusta N lehtisolmu, se nimetään luokan mukaan ja päätöspuu valmistuu. Muussa tapauksessa (luokkia on useita) valitaan jakokriteeri attribuutin valintamitalla. Valittu jakokriteeri kertoo mitä attribuuttia solmussa N testataan ja mitkä oksat (testin lopputulokset) siitä haarautuvat. Solmu N nimetään jakokriteerin mukaan ja tapaukset joukossa D jaetaan jakokriteerin mukaan osajoukkoihin. [Han *et al.* 2012]

Jos jakokriteerin attribuutti A on diskreetti, eli sen mahdolliset arvot voidaan luetella, testin lopputulokset vastaavat attribuutin mahdollisia arvoja. Jokaiselle arvolle a_j tehdään oma oksa ja se nimetään arvon mukaan. Osajoukko D_j sisältää joukon D tapaukset, joilla attribuutin A arvo on a_j. Attribuutti A voidaan poistaa attribuuttilistasta eikä sitä tarvitse harkita enää jakokriteerin attribuutiksi, koska data on jaettu A:n arvojen mukaan. [Han *et al.* 2012] On myös mahdollista ryhmitellä attribuutin A arvot kahteen tai useampaan oksaan [Wu *et al.* 2008].

Jos jakokriteerin attribuutti A on jatkuva, eli arvoja ei voi luetella, on solmun N testillä kaksi lopputulosta, $A \leq$ jakopiste ja $A >$ jakopiste. Attribuutin arvoväli jaetaan kahteen osaan jakopisteellä. Jakopiste saadaan attribuutin valintamitalta osana jakokriteeriä. Solmuun N lisätään kaksi oksaa ja ne nimetään kahden lopputuloksen mukaisesti. Tapaukset joukossa D jaetaan osajoukkoihin D_1 ja D_2. Osajoukossa D_1 ovat tapaukset, joille pätee $A \leq$ jakopiste ja osajoukossa D_2 tapaukset, joille pätee $A >$ jakopiste. [Han *et al.* 2012]

Algoritmi käyttää edellä kuvattua tapaa rekursiivisesti muodostamaan jokaiselle osajoukolle D_j päätöspuun. Rekursiivinen osiin jakaminen voi päättyä kolmella eri tavalla: [Han *et al.* 2012]

1. Kaikki tapaukset joukossa D (jota edustaa solmu N_j) kuuluvat samaan luokkaan.
2. Ei ole enää jäljellä attribuutteja, joiden mukaan tapauksia voidaan erotella. Solmusta N tulee lehtisolmu ja se nimetään luokkatunnuksen mukaan, jota vastaavia tapauksia on eniten joukossa D.
3. Jos osajoukko D_j on tyhjä. Tässä tapauksessa luodaan solmusta N, joka edustaa joukkoa D, lehtisolmu ja nimetään se joukon D enemmistöluokan mukaan.

Alkuperäinen päätöspuu karsitaan (prune), jotta vältetään ylisovittaminen (overfitting) [Wu *et al.* 2008]. Monet oksat ovat voineet muodostua opetusjoukon poikkeamista, joita aiheuttavat virheet ja poikkeavat havainnot datassa. Karsimisella pyritään poistamaan tämän tyyppiset oksat. Karsittu päätöspuu on yleensä pienempi ja yksinkertaisempi kuin alkuperäinen ja siten helpompi ymmärtää. Se on myös tavallisesti nopeampi ja pa-

rempi luokittelemaan uudet tapaukset oikein. [Han *et al.* 2012] Karsiminen tehdään yhdellä päätöspuun läpikäynnillä lehdistä juureen. Jokaisen alipuun kohdalla C4.5 laskee oksien arvioidun luokitteluvirheiden määrän yhteen ja vertaa sitä luokitteluvirheiden määrään, joka tulisi, jos alipuu korvataan lehdellä. Alipuu karsitaan, jos sen yhteenlaskettu virhemäärä on suurempi kuin sen korvaavalla lehdellä olisi. Alipuun voi korvata myös sen yhdellä oksalla ja tässäkin tapauksessa algoritmi vertaa arvioituja virhemääriä samalla tavalla. [Wu *et al.* 2008] C4.5 käyttää pessimististä virhemäärän arviointia pohjanaan opetusjoukko. C4.5-algoritmin karsimismenetelmää kutsutaan pessimistiseksi karsinnaksi (pessimistic pruning). [Han *et al.* 2012]

Komplekseja päätöspuita voi olla vaikea ymmärtää, vaikka ne olisi jo karsittu. Esimerkiksi yhteen luokkaan liittyvä tieto jakaantuu puussa yleensä useaan oksaan. C4.5 esitteli päätöspuulle vaihtoehtoisen formalismin, joka koostuu listasta sääntöjä. Säännöt ovat muodossa ”Jos *A* ja *B* ja *C* ja ... niin luokka *X*”, jossa kaikki yhden luokan säännöt on ryhmitelty yhteen. Tapaus luokitellaan etsimällä ensimmäinen sääntö, jonka ehdot täsmäävät tapaukseen. Jos sopivaa sääntöä ei löydy, tapaukselle annetaan oletusluokkatunnus. C4.5 muodostaa sääntöjoukot alkuperäisestä päätöspuusta. Jokaisesta polusta juuresta lehteen tulee prototyyppisääntö, jonka ehdot (*A*, *B*, *C*, ...) ovat attribuuttitestien tulokset polun varrella ja luokka *X* on lehden nimi. Prototyyppisääntöä yksinkertaistetaan vähentämällä ehtoja hill-climbing -algoritmilla niin, että pessimistinen virhemäärä on mahdollisimman pieni. Lopuksi yksinkertaistetuista säännöistä valitaan osajoukko jokaiselle luokalle erikseen. Nämä osajoukot järjestetään ja valitaan oletusluokkatunnus. Lopullisessa sääntöjoukossa on yleensä paljon vähemmän sääntöjä kuin karsitussa päätöspuussa lehtiä. [Wu *et al.* 2008]

5.2 Apriori

Apriori on R. Agrawalin ja R. Srikantin vuonna 1994 kehittämä algoritmi kattavien joukkojen louhimiseen totuusarvoisia assosiaatiosääntöjä varten. Apriori etenee iteratiivisesti tasoittaisena hakuna (level-wise search), jossa k:n alkion joukkoja käytetään tutkimaan (k + 1):n alkion joukkoja. [Han *et al.* 2012] Algoritmi on seuraavanlainen [Agrawal and Srikant 1994]:

1. Lasketaan alkioiden esiintymien määrä datajoukossa. Saadaan ensimmäinen joukko kandidaatteja C_1. Valitaan niistä yhden alkion joukot, jotka esiintyvät datassa minimituen verran. Saadaan joukko L_1, jossa ovat yhden alkion kokoiset kattavat joukot. Asetetaan k:n arvoksi 2. Siirrytään seuraavaan vaiheeseen, jos L_1 ei ole tyhjä.
2. Generoidaan k:n alkion joukot eli kandidaatit C_k joukosta L_{k-1} (minimituen täyttävät (k - 1):n alkion joukot) yhdistämällä joukon L_{k-1} jäsenet toisiinsa. Valitaan

kandidaateiksi ne alkiojoukot, joiden kaikki osajoukot ((*k* - 1):n alkion joukot) kuuluvat joukkoon L_{k-1} eli kaikkien osajoukkojen täytyy olla kattavia.

3. Lasketaan kandidaattijoukon C_k jäsenten esiintymät datassa. Valitaan ne, jotka esiintyvät datassa minimituen verran. Tästä saadaan uusi joukko L_k, jossa on kattavia *k*:n alkion joukkoja.
4. Jos joukko L_k ei ole tyhjä, siirrytään vaiheeseen 2 ja kasvatetaan *k*:n arvoa yhdellä. Muuten, jos L_k on tyhjä, yhdistetään lopputulokseksi kaikki minimituen täyttävät joukot $L_1, L_2, ..., L_{k-1}$.

Assosiaatiosäännöt generoidaan algoritmilla löydetyistä kattavista joukoista. Ensin etsitään jokaisen kattavan joukon *l* osajoukot, jotka eivät ole tyhjiä. Jokaiselle tällaiselle osajoukolle *a* muodostetaan sääntö $a \Rightarrow (l - a)$, jos joukon *l* tuen suhde osajoukon *a* tukeen on vähintään minimiluottamuksen verran. [Agrawal and Srikant 1994] Koska säännöt generoidaan kattavista joukoista, jokainen sääntö täyttää automaattisesti minimituen [Han *et al.* 2012].

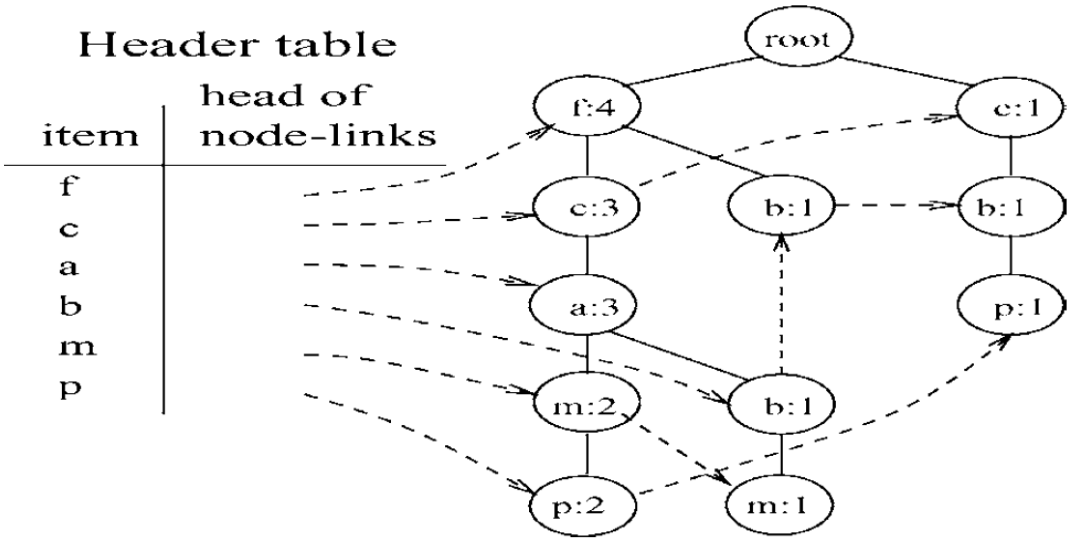
5.3 FP-growth

FP-growth (frequent-pattern growth) on algoritmi kattavien joukkojen louhintaan. Menetelmän on kehittänyt Jiawei Han kollegoineen 2000-luvun alussa. [Han *et al.* 2004] Siitä saadut kattavat joukot voidaan muuntaa assosiaatiosäännöiksi, kuten Apriorin kohdalla. FP-growth on yleisesti Aprioria nopeampi löytämään kattavat joukot, mutta nopeuteen vaikuttavat algoritmin lisäksi sen toteutus ja millaista dataa louhitaan. [Witten *et al.* 2011]

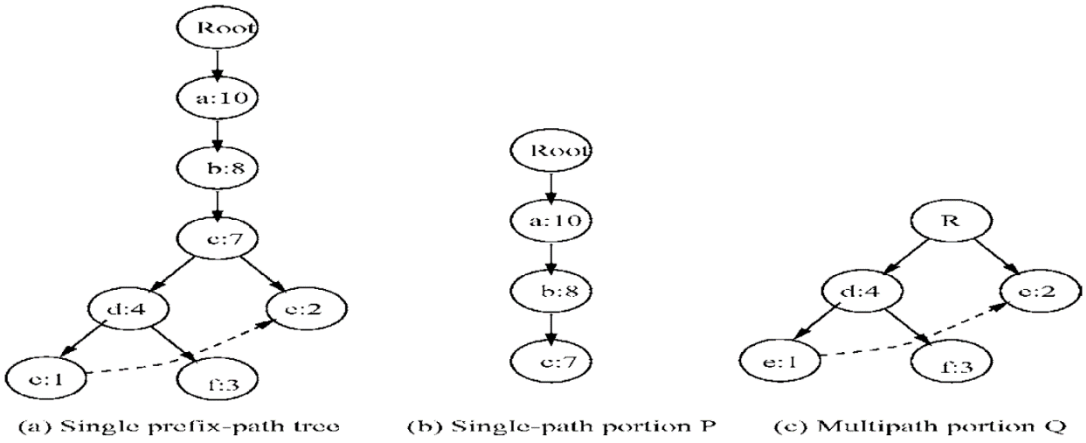
FP-growth käyttää hajoita ja hallitse -taktiikkaa. Se kokoaa ensin datan kompaktiin tietorakenteeseen FP-puu (FP-tree, frequent-pattern tree), jossa säilytetään tiedot datan kattavista hahmoista puurakenteessa ja aputaulukossa. Puun juurisolmu ei edusta mitään, mutta sen lapsisolmuista lähtevien (ali)puiden polut lehtiin vastaavat kattavia hahmoja. Yksi puun solmu edustaa yhtä kattavaa tietoalkiota. Solmulla on ominaisuuksina tietoalkion nimi, lukumäärä ja solmulinkki (node-link). Lukumäärä kertoo kuinka monessa datan tapauksessa se ja sitä edeltävät solmut esiintyvät yhdessä ja solmulinkki linkittyy toiseen vastaavan tietoalkion solmuun tai arvoon null, jos toista vastaavaa solmua ei ole. Aputaulukossa frequent-item-header table on kaksi saraketta, ensimmäinen kattavan tietoalkion nimeä varten ja toinen on osoitin ensimmäiseen tietoalkiota vastaavaan solmuun puussa. Aputaulukossa tietoalkiot on järjestetty tietoalkioiden tuen mukaan laskevaan järjestykseen. Puun muodostamiseen tarvitaan datan lisäksi minimituen arvo. Esimerkki FP-puusta on annettu kuvassa 3. FP-puu rakennetaan seuraavalla tavalla: [Han *et al.* 2004]

1. Data käydään kertaalleen läpi keräten kattavat tietoalkiot ja lasketaan niille tuet. Kattavien tietoalkioiden tulee esiintyä datassa vähintään minimituen verran. Kattavista tietoalkiosta muodostetaan lista, joka on järjestetty tuen mukaan laskevaan järjestykseen.

2. Luodaan puulle juurisolmu *T*, joka saa nimen ’null’.
3. Jokaiselle tapaukselle datassa:
    i. Valitaan tapauksen kattavat tietoalkiot ja järjestetään ne ensimmäisen vaiheen listan mukaan.
    ii. Tapauksen kattavien tietoalkioiden lista käydään rekursiivisesti läpi kutsumalla funktiota lisää_puuhun(*p*, *P*, *T*), jossa *p* on listan ensimmäinen alkio, *P* on loppulista ja *T* on tarkasteltava solmu (ensimmäisellä kutsulla juurisolmu). Funktio toimii seuraavalla tavalla. Jos solmulla *T* on lapsisolmu *N*, joka vastaa listan ensimmäistä solmua *p*, kasvatetaan solmun *N* tallettamaa lukumäärää yhdellä. Muussa tapauksessa luodaan uusi solmu *N*, jonka lukumäärä on yksi ja siitä tulee solmun *T* lapsisolmu. Uuden solmun *N* solmulinkki liitetään muihin vastaaviin solmuihin puussa. Jos lista *P* ei ole tyhjä, kutsutaan funktiota uudelleen parametreina listan *P* ensimmäinen alkio, loput listasta *P* ja solmu *N*.



Kuva 3. Esimerkki FP-puusta. Solmujen luvut ovat tuen arvoja. [Han *et al.* 2004]



Kuva 4. Esimerkki FP-puusta, joka ei haaraudu heti juurisolmun jälkeen ja sen jakamisesta kahdeksi puuksi louhittaessa. Solmujen luvut ovat tuen arvoja. [Han *et al.* 2004]

Kun FP-puu on rakennettu, sitä voidaan käyttää kattavien joukkojen louhintaan. Jos FP-puu, jossa juuren jälkeen on vain yksi solmu, josta lähtevä polku voi haarautua, käytetään algoritmissa tähän tapaukseen (kuva 4 (a)) optimoitua menetelmää. Puuta käsitellään silloin kahdessa osassa. Ensin haarautumaton polku mukaan lukien ensimmäinen solmu, joka haarautuu (kuva 4 (b)) ja toisena haarautuvat osat, joita käsitellään yhtenä puuna (kuva 4 (c)), kuten muita heti juuresta haarautuvia puita. Sen jälkeen tulokset kummastakin vaiheesta yhdistetään. Algoritmi koostuu täten kahdesta osasta, yksipolkuisen puun louhinta ja monipolkuisen puun louhinta. Algoritmi tarvitsee syötteenä FP-puun ja hahmon (ensimmäisellä kutsulla null) sekä minimituen ja antaa vastauksena kattavat joukot (hahmot). Algoritmi etenee rekursiivisesti seuraavalla tavalla: [Han *et al.* 2004]

1. Jos puu alkaa vain yhdellä polulla juuresta, jaetaan puu kahteen osaan. *P* on puun yksittäinen polku juuresta ensimmäiseen haarautuvaan solmuun ja *Q* on puun haarautuva osa, jossa ylin solmu korvataan juurisolmulla. Generoidaan polun *P* solmuista kaikki kombinaatiot (hahmot) ja annetaan jokaiselle kombinaatiolle tueksi siinä esiintyvien tietoalkioiden pienimmän tuen arvo.
2. Muussa tapauksessa koko puu on *Q*.
3. Jokaiselle alkiolle a_i puussa *Q*, pienimmän tuen omaavasta alkiosta suurimpaan:
    i. Luodaan hahmo yhdistämällä syötteenä annettu hahmo ja a_i ja annetaan uudelle hahmolle tueksi a_i:n tuki.
    ii. Muodostetaan uuden hahmon ehdollinen hahmopohja (conditional pattern-base) ja siitä ehdollinen FP-puu (conditional FP-tree). Ehdollinen hahmopohja muodostetaan hahmoa edeltävistä alkiosta puun *Q* eri poluissa, joissa hahmo esiintyy. Ehdolliseen puuhun valitaan ne alkiot, jotka

toistuvat hahmopohjassa minimituen verran. Esimerkki on annettu kuvassa 5.

iii. Jos uuden hahmon ehdollinen FP-puu ei ole tyhjä, kutsutaan algoritmia parametreina ehdollinen FP-puu ja hahmo.

4. Tehdään puiden *P* ja *Q* hahmojoukoista karteesinen tulo ja palautetaan puun *P* hahmojoukon, puun *Q* hahmojoukon ja karteesisen tulon unioni.

Item	Conditional pattern-base	Conditional FP-tree
p	$\{(fcam{:}2), (cb{:}1)\}$	$\{(c{:}3)\}\|p$
m	$\{(fca{:}2), (fcab{:}1)\}$	$\{(f{:}3, c{:}3, a{:}3)\}\|m$
b	$\{(fca{:}1), (f{:}1), (c{:}1)\}$	$\emptyset$
a	$\{(fc{:}3)\}$	$\{(f{:}3, c{:}3)\}\|a$
c	$\{(f{:}3)\}$	$\{(f{:}3)\}\|c$
f	$\emptyset$	$\emptyset$

Kuva 5. Kuvan 3 FP-puun ehdolliset hahmopohjat ja FP-puut alkioittain, joissa minimituki on 3 [Han *et al.* 2004].

5.4 K-means

K-means on klusterointialgoritmi, jonka useat tutkijat eri tieteenaloilla ovat kehittäneet erikseen, etenkin Lloyd vuonna 1957 (julkaistiin vasta vuonna 1982), Forgey vuonna 1965, Friedman ja Rubin vuonna 1967 ja MacQueen myös vuonna 1967 [Wu *et al.* 2008]. K-means tarvitsee datan lisäksi syötteeksi klustereiden määrän *k*. Se jakaa datajoukon annettuun määrään klustereita niin, että klustereiden sisällä samankaltaisuus on suuri ja klustereiden välillä pieni. Samankaltaisuutta mitataan klusteriin kuuluvien tapausten keskiarvon suhteen. Algoritmi etenee seuraavalla tavalla: [Han *et al.* 2012]

1. Valitaan datajoukosta mielivaltaiset *k* tapausta alustaviksi klustereiden keskipisteiksi.
2. Jokainen tapaus sijoitetaan klusteriin, joka on samankaltaisin tapauksen kanssa eli tapaus on lähimpänä klusterin keskipistettä (etäisyysmittana euklidinen etäisyys).
3. Päivitetään klusterien keskipisteet laskemalla uusi keskipiste tapausten keskiarvosta jokaisessa klusterissa.
4. Toistetaan vaiheita 2 ja 3, kunnes klustereissa ei tapahdu muutoksia.

K-means yrittää määritellä *k* klusteria, jotka minimoivat neliövirhesummafunktion (square-error function). Funktiossa lasketaan tapauksen etäisyys klusterinsa keskipisteestä (klusterin tapausten keskiarvo) korotettuna potenssiin kaksi jokaiselle tapaukselle jokaisessa klusterissa ja nämä etäisyyksien neliöt summataan yhteen. Tällä tavalla yritetään saada klustereista mahdollisimman kompakteja ja erillisiä. K-means ei sovi mielivaltaisen muotoisten klustereiden löytämiseen eikä datalle, jossa ei pysty laskemaan keskiarvoa. [Han *et al.* 2012]

K-means -algoritmin lopputulokseen voivat vaikuttaa mielivaltaisesti valitut alustavat klustereiden keskipisteet. Alustavien keskipisteiden vaikutusta on minimoitu paremmalla valintamenetelmällä algoritmin versiossa k-means++. [Witten *et al.* 2011] Algoritmista on myös muita variantteja, jotka eroavat alkuperäisestä esimerkiksi alustavien keskipisteiden valinnassa (kuten k-means++), samankaltaisuuden laskutavassa tai klustereiden keskipisteiden laskutavassa [Han *et al.* 2012].

5.5 Expectation-Maximization

Expectation-Maximization, lyhyemmin EM, on klusterointialgoritmi, jonka ovat kehittäneet Dempster, Laird ja Rubin vuonna 1977. EM voidaan nähdä K-means -algoritmin laajennoksena. Sen sijaan, että sijoitetaan jokainen tapaus sopivaan klusteriin, EM sijoittaa jokaisen tapauksen jokaiseen klusteriin painon mukaan, joka edustaa tapauksen todennäköisyyttä klusteriin kuulumisesta. Tästä seuraa, että klustereiden välillä ei ole tarkkoja rajoja. Klustereiden keskipisteet (keskiarvot) lasketaan täten painojen perusteella. [Han *et al.* 2012]

Käytännössä jokainen klusteri voidaan esittää matemaattisesti parametrisella todennäköisyysjakaumalla. Koko data on sekoitus (mixture) näitä jakaumia, jossa jokaista yksittäistä jakaumaa kutsutaan komponenttijakaumaksi (component distribution). Täten datan voi klusteroida käyttämällä *k*:n todennäköisyysjakauman äärellistä tiheyssekoitemallia (finite mixture density model), jossa jokainen jakauma esittää klusteria. [Han *et al.* 2012] Jakauma antaa todennäköisyyden, jolla klusteriin kuuluvalla tapauksella on tietyt attribuuttien arvot [Witten *et al.* 2011]. Tehtävänä on arvioida jakaumien parametrit niin, että ne sopivat dataan parhaiten [Han *et al.* 2012].

EM koostuu kahdesta vaiheesta alustuksen jälkeen, odotusarvojen laskeminen (expectation step) ja odotusarvojen maksimointi (maximization step), joista algoritmi on saanut nimensä. Algoritmille annetaan datan lisäksi klustereiden määrän *k*. Se etenee seuraavanlaisesti: [Han *et al.* 2012]

1. Tehdään alustava arvio sekoitemallin parametreista valitsemalla satunnaiset *k* tapausta klustereiden keskipisteiksi ja arvioimalla muut lisäparametrit.

2. Odotusarvojen laskeminen: Sijoitetaan jokainen tapaus jokaiseen klusteriin ja lasketaan klustereihin kuulumisen todennäköisyydet. Nämä todennäköisyydet ovat odotusarvoja tapauksen kuulumisesta klusteriin.
3. Odotusarvojen maksimointi: Käytetään edellisen vaiheen todennäköisyyksien arvioita (odotusarvoja) sekoitemallin parametrien uudelleen arviointiin.
4. Toistetaan vaiheita 2 ja 3 kunnes parametrien arvot konvergoituvat (nykyisen ja edellisen kierroksen arvot parametreille lähestyvät toisiaan).

Algoritmin tulokset ovat tapausten todennäköisyyksiä löydettyihin klustereihin kuulumisesta [Witten *et al.* 2011].

5.6 Hierarkkinen klusterointi

Hierarkkinen klusterointi ryhmittelee datan tapaukset klusterointipuuksi. Hierarkkiset menetelmät voidaan jakaa kahteen osaan, joko yhdistäviin (agglomerative hierarchical clustering) tai jakaviin (divisive hierarchical clustering) klusterointimenetelmiin. Yhdistävä hierarkkinen klusterointi, jonka ovat kehittäneet Day ja Edelsbrunner vuonna 1984, aloittaa klusteroinnin sijoittamalla jokaisen tapauksen omaan klusteriinsa ja sen jälkeen yhdistämällä vaihe vaiheelta klustereita toisiinsa yhä suuremmiksi klustereiksi jonkin kriteerin mukaan, kunnes kaikki tapaukset ovat samassa klusterissa tai jokin muu lopetusehto toteutuu. Suurin osa hierarkkisista klusterointimenetelmistä on yhdistäviä. Ne eroavat ainoastaan sen perusteella, miten klustereiden välinen samankaltaisuus määritetään. Jakava hierarkkinen klusterointi toimii päinvastaisesti kuin yhdistävä menetelmä. Se aloittaa yhdestä klusterista, johon kaikki tapaukset kuuluvat ja jakaa klusteria tai klustereita vaihe vaiheelta pienemmiksi osiksi jonkin kriteerin mukaan, kunnes kaikki tapaukset ovat omissa klustereissaan tai jokin lopetusehto täyttyy. Kummassakin menetelmässä lopetusehtona voi olla, että saavutetaan haluttu klustereiden määrä, tai klusterin halkaisija on maksimissaan jonkin kynnysarvon mittainen. Hierarkkisen klusteroinnin vaiheita kuvataan usein puurakenteella nimeltä dendrogrammi (kuva 1 (d)). Siitä näkee, kuinka tapaukset on ryhmitelty vaihe vaiheelta. [Han *et al.* 2012]

Klustereiden yhdistämisessä ja jakamisessa klustereiden välisiä etäisyyksiä eli klustereiden samankaltaisuutta voidaan mitata monella eri tavalla. Yksi tapa on mitata minimietäisyys klustereiden välillä eli mitata etäisyys kahden lähimmän tapauksen välillä, jotka ovat eri klustereissa. Tätä tapaa kutsutaan nimellä lähin naapuri (single-linkage). Toinen yleinen tapa on mitata maksimietäisyys klustereiden välillä eli mitata etäisyys kahden kauimmaisen tapauksen välillä. Tällöin kahden klusterin katsotaan olevan samankaltaisia vain, jos kaikki tapaukset niiden yhdistyessä olisivat suhteellisen samankaltaisia. Tätä kutsutaan nimellä kaukaisin naapuri (complete-linkage). Kummassakin tavassa poikkeustapaukset vaikuttavat tulokseen herkästi. Tämän ongelman ratkaisee menetelmä naapurikeskiarvo (average-linkage), joka on kompromissi aiempien tapojen välillä. Siinä

lasketaan keskiarvoetäisyys kaikkien kahden klusterin jäsenien välillä. [Witten *et al.* 2011] Tapausten välisen etäisyyden laskemiseen on myös useita vaihtoehtoja, kuten euklidinen etäisyys, Manhattan, Jaccardin kerroin jne. Tämän etäisyysmitan valinta vaikuttaa myös klusteroinnin tulokseen. [Choi *et al.* 2010]

Hierarkkinen klusterointi on yksinkertaista, mutta menetelmän ongelmakohtana on klusterien liitos- ja jakokohtien valinta. Valintaa ei pysty jälkikäteen vaihtamaan, kun se on tehty ja se vaikuttaa klustereihin menetelmän loppuun asti. Huonot valinnat johtavat huonoihin klustereihin. [Han *et al.* 2012]

6 Tulokset

Louhinnan tuloksia tarkastellaan louhintatehtävien mukaisesti, joita ovat opiskelijoiden ryhmittely klusteroimalla, suosittujen kurssihahmojen löytäminen assosiaatioanalyysillä ja kurssien suosittelun testaaminen eri louhintamenetelmin. Opiskelijoiden ryhmittelyllä ja suosittujen kurssihahmojen tutkimisella pystyttiin havainnoimaan opiskelijoiden kiinnittymistä opintoihinsa. Vaikka kurssien suositteluun tämän työn data ei sovi yhtä hyvin, eri suosittelumenetelmiä pystyttiin silti testaamaan ja vähän vertailemaan.

6.1 Opiskelijoiden ryhmittely

Opiskelijoiden ryhmittelyyn käytettävä data jaettiin ensin osiin opiskelijan aloitusvuoden ja opinto-oikeuden perusteella. Tästä saatiin neljä eri datajoukkoa, jotka klusteroitiin jokainen erikseen kolmella eri tavalla käyttäen hierarkkista klusterointia ja K-means ja EM-algoritmeja. Kaikilla menetelmillä kokeiltiin miten opiskelijat jakautuvat ryhmiin eri klusterimäärillä. Klustereiden ominaisuuksia tarkasteltiin työkalujen tarjoamien jakaumien avulla ja myös suoraan tietyn klusterin datasta. Kurssien koodit olivat datassa aakkosjärjestyksessä, mikä helpotti havainnointia. Jos klusterin ominaisuuksia oli vaikea nähdä edellä mainituin keinoin, etsittiin klusterin kattavat joukot Apriorin avulla Wekassa. Tätä varten klusterin datasta täytyi poistaa ensin nolla-arvot, jotta saatiin suoritetut kurssit esiin. Klusteroinnin tulokset on koottu tiivistetysti taulukkoon 4.

Molempina vuosina vain ylempää korkeakoulututkintoa suorittavista opiskelijoista oli selvästi erotettavissa Human-Technology Interaction (HTI) -kursseja suorittavat opiskelijat, joiden koulutusohjelma on Master's Degree Programme in Human-Technology Interaction. Jokainen klusterointimenetelmä löysi kyseisen opiskelijaryhmän, mutta algoritmilla K-means vastaavassa klusterissa oli joko muita opiskelijoita (aloitusvuosi 2018) tai ei ihan kaikkia kyseisiä opiskelijoita (aloitusvuosi 2017). K-means löysi lisäksi kummaltakin vuodelta klusterin, jossa on tehty eniten joitakin tietojenkäsittelyopin syventäviä ja aineopintoja sekä osa opiskelijoista oli suorittanut myös suomen kielen opintoja. Tämän kaltainen ryhmä tulee esiin myös muilla klusterointimenetelmillä. Suorituksista on pääteltävissä, että klusterissa on tietojenkäsittelyopin maisteriopintoja suorittavia opiskelijoita sekä opiskelijoita muista tietojenkäsittelyn maisterikoulutusohjelmista (Master's Degree Programme in Software Development, MDP in Computational Big Data Analytics), joista osa on kansainvälisiä opiskelijoita. Kahden edellisen koulutusohjelman opiskelijat eivät erotu selvästi omiksi klustereikseen kolmella klusterointimenetelmällä, koska opinnot ovat päällekkäisiä tietojenkäsittelyopin maisteriopintojen kanssa, kun taas HTI-koulutusohjelman kurssit ovat muista maisteriohjelmista erillisiä. K-means laittoi myös lukuvuodelta 2017-2018 kaksi kansainvälistä opiskelijaa omaan klusteriinsa, koska heillä oli eniten kieliopintoja, ja lukuvuodelta 2018-2019 kaksi muiden tieteenalojen opintoja suorittaneita opiskelijoita omiin klustereihinsa.

Hierarkkisella klusteroinnilla HTI-kursseja suorittavien opiskelijoiden klusterin lisäksi löytyi lukuvuodelta 2017-2018 yksi opiskelija, joka muodosti oman klusterinsa, koska hänellä oli muista poikkeavia suorituksia (muun tieteenalan opintoja). Näiden lisäksi muodostui myös kaksi muuta klusteria. Ensimmäisessä näistä kahdesta klusterista oli suoritettu eniten joitakin tietojenkäsittelyopin syventäviä ja aineopintoja ja toisessa yleisiä ja suomen kielen ja kulttuurin opintoja syventävien tietojenkäsittelyopin opintojen lisäksi. Suorituksista voidaan päätellä, että jälkimmäisessä klusterissa on kansainvälisiä opiskelijoita. Lukuvuodelta 2018-2019 hierarkkisella klusteroinnilla saatiin ryhmiteltyä HTI-kurssien suorittajat, yksi suorituksiltaan poikkeava opiskelija (muun tieteenalan opintoja), opiskelijat, joilla ei ole suorituksia sekä klusteri, jossa opiskelijat ovat suorittaneet yleisiä opintoja, tietojenkäsittelyn syventäviä ja aineopintoja ja osa heistä on tehnyt suomen kielen opintoja.

Ylempää korkeakoulututkintoa suorittavista opiskelijoista EM-algoritmilla löytyi HTI-koulutusohjelman opiskelijat omaan klusteriinsa vuonna 2017 aloittaneista opiskelijoista. Samalta lukuvuodelta EM laittoi samat kaksi kansainvälistä opiskelijaa kuin K-means omaan klusteriinsa ja muut maisteriopiskelijat jaettuna kolmeen eri klusteriin. Lukuvuonna 2018-2019 opiskelijat jakautuivat kahdella klusterilla HTI-koulutusohjelman opiskelijoihin ja muiden tietojenkäsittelyn maisteriohjelmien opiskelijoihin.

	K-means	**Hierarkkinen klusterointi**	**EM**
Ylempi korkeakoulututkinto 2017 (30)	C1 (2): yleisiä opintoja, suomen kielen opintoja ja syventäviä TIE-opintoja C2 (22): yleisiä opintoja, tietojenkäsittelyn syventäviä ja aineopintoja ja osalla suomen kielen opintoja C3 (6): HTI-kursseja	C1 (1): muun tieteenalan opintoja C2 (14): tietojenkäsittelyn syventäviä ja aineopintoja C3 (7): HTI-kursseja C4 (8): yleisiä opintoja, suomen kielen ja kulttuurin opintoja ja tietojenkäsittelyn syventäviä opintoja	C1 (2): yleisiä opintoja, suomen kielen opintoja ja syventäviä TIE-opintoja C2 (7): HTI-kursseja C3 (9): yleisiä ja suomen kielen kursseja ja muutamia tietojenkäsittelyn syventäviä opintoja C4 (6): tietojenkäsittelyn syventäviä ja aineopintoja C5 (6): tietojenkäsittelyn syventäviä ja aineopintoja

Ylempi korkeakoulututkinto 2018 (43)	C1 (1): muun tieteenalan opintoja syventävien tietojenkäsittelyn opintojen lisäksi C2 (15): yleisiä opintoja, tietojenkäsittelyn syventäviä ja aineopintoja ja osalla suomen kielen opintoja C3 (1): muun tieteenalan opintoja C4 (26): HTI-kursseja sekä opiskelijoita, joilla ei yhtään suorituksia tai suoritettuna muita kuin HTI-kursseja	C1 (17): HTI-kursseja C2 (23): yleisiä opintoja, tietojenkäsittelyn syventäviä ja aineopintoja ja osalla suomen kielen opintoja C3 (1): muun tieteenalan opintoja C4 (2): ei suorituksia	C1 (17): HTI-kursseja C2 (26): yleisiä opintoja, tietojenkäsittelyn syventäviä ja aineopintoja ja osalla suomen kielen opintoja sekä opiskelijat, joilla ei ole suorituksia
Alempi ja ylempi korkeakoulututkinto 2017 (120)	C1 (89): vaihtelevasti tietojenkäsittelyn yleis-, perus- ja aineopintoja C2 (1): muun tieteenalan opintoja C3 (30): tietojenkäsittelyn yleis-, perus- ja aineopintoja sekä pakollisia englannin ja matematiikan ja tilastotieteen opintoja	C1 (106): eniten tietojenkäsittelyn yleis-, perus- ja aineopintoja mutta myös näitä vähemmän suorittaneita opiskelijoita C2 (1): muun tieteenalan opintoja C3 (1): muun tieteenalan opintoja C4 (12): ei suorituksia	C1 (73): eniten tietojenkäsittelyn yleis-, perus- ja aineopintoja C2 (47): vaihtelevasti tietojenkäsittelyn yleis- ja perusopintoja, ei lainkaan tietojenkäsittelyn aineopintoja ja opiskelijoita, joilla ei yhtään suorituksia
Alempi ja ylempi korkeakoulututkinto 2018 (108)	C1 (67): eniten tietojenkäsittelyn yleis-, perus- ja aineopintoja C2 (40): vaihtelevasti tietojenkäsittelyn yleis-, perus- ja aineopintoja C3 (1): muun tieteenalan opintoja	C1 (95): eniten tietojenkäsittelyn yleis-, perus- ja aineopintoja ja muiden tieteenalojen opintoja C2 (1): muun tieteenalan opintoja C3 (12): ei suorituksia	C1 (50): vähemmän tietojenkäsittelyn yleis-, perus- ja aineopintoja, muiden tieteenalojen opintoja ja opiskelijoita, joilla ei ole yhtään suorituksia C2 (58): eniten tietojenkäsittelyn yleis-,

			perus- ja aineopintoja ja muiden tieteenalojen opintoja

Taulukko 4. Klusteroinnin tulokset jaoteltuna datan ja klusterointimenetelmän mukaan, jossa C merkitsee klusteria ja suluissa oleva luku klusterin jäsenten lukumäärää.

Alempaa ja ylempää korkeakoulututkintoa suorittavista opiskelijoista K-means -algoritmilla vuonna 2017 aloittavista opiskelijoista löytyi iso klusteri, jossa oli tehty vaihtelevasti tietojenkäsittelyn yleis-, perus- ja aineopintoja sekä muita opintoja. K-means laittoi myös yhden poikkeavan tapauksen omaan klusteriinsa ja neljäsosan tapauksista klusteriin, jossa opiskelijat ovat suorittaneet tietojenkäsittelyn yleis-, perus- ja aineopintoja sekä pakollisia englannin ja matematiikan ja tilastotieteen opintoja. Lukuvuodelle 2018-2019 K-means löysi jäsenmäärältään tasaisemman kokoiset kaksi klusteria ja taas yhden poikkeavan tapauksen. Ensimmäisessä klusterissa oli suoritettu eniten tietojenkäsittelyn yleis-, perus- ja aineopintoja kun taas toisessa klusterissa niitä oli suoritettu huomattavasti vähemmän tai ei ollenkaan.

Hierarkkisella klusteroinnilla saatiin kumpanakin vuonna yksi iso klusteri ja muutama pieni klusteri LuK- ja FM-tutkintoa suorittavista opiskelijoista. Lukuvuonna 2017-2018 kaksi poikkeustapausta, joilla on muiden tieteenalojen opintojen suorituksia, ja niitäkin vain muutama, muodostuivat omiksi klustereikseen. Lisäksi hierarkkisella klusteroinnilla pystyttiin erottamaan opiskelijat, joilla ei ole yhtään suoritusta, omaan klusteriinsa. Lukuvuonna 2018-2019 yksi poikkeustapaus muodostui yksittäiseksi klusteriksi ja opiskelijat, joilla ei ole suorituksia erottuivat taas omaksi ryhmäkseen. Molempien vuosien iso klusteri sisälsi sekä ahkerasti tietojenkäsittelyn yleis-, perus- ja aineopintoja suorittaneita opiskelijoita, että niitä vähemmän suorittaneita ja muita opintoja tehneitä opiskelijoita. Jos klusterien määrää kasvatti hierarkkisessa klusteroinnissa, poikkeavat tapaukset alkoivat erottua omiksi klustereikseen isosta klusterista ja lopulta iso klusteri alkoi pilkkoutua pienempiin klustereihin, joissa opiskelijat jakaantuivat paremmin ryhmiin.

EM-algoritmilla saatiin jäsenmäärältään tasaisemman kokoiset kaksi klusteria kumpanakin vuonna alempaa ja ylempää korkeakoulututkintoa suorittavista opiskelijoista. Suuremmassa klusterissa oli suoritettu eniten tietojenkäsittelyn yleis-, perus- ja aineopintoja ja pienemmässä huomattavasti vähemmän ja siellä on myös tapaukset, joilla ei ole suorituksia. Poikkeavat tapaukset jakaantuivat kumpaankin klusteriin.

Klusteroinnilla saaduista opiskelijaryhmistä (klustereista) voidaan nähdä opintoihinsa kiinnittyneet opiskelijat ja erottaa ne opiskelijat, jotka eivät ole tehneet yhtä paljon oman alansa opintoja. Myös poikkeavat tapaukset voivat olla mielenkiintoisia sekä sellaiset opiskelijat, jotka eivät ole suorittaneet kursseja lainkaan. Näistä opiskelijoista luonnollisesti osa on myös poissaoleviksi ilmoittautuneita.

Ylempää korkeakoulututkintoa suorittavat opiskelijat jakautuvat ryhmiin koulutusohjelmansa kurssitarjonnan perusteella, minkä takia kaikki eri koulutusohjelmien opiskelijat eivät näy erikseen omina ryhminään. Muista kuin HTI-koulutusohjelman opiskelijoista on kuitenkin erotettavissa kansainväliset opiskelijat, koska heille kuuluu pakollisia suomen kielen kursseja. Alemman ja ylemmän korkeakoulututkinnon suorittajat jakautuvat ryhmiin sen perusteella, kuinka paljon he ovat ottaneet tietojenkäsittelyn kursseja. Muita kuin tietojenkäsittelyn kursseja ottaneita opiskelijoita näkyy kaikissa neljässä datajoukossa poikkeustapauksina tai suurempien klustereiden sisällä. Mitään yhteistä ryhmää ei tällaisille opiskelijoille syntynyt.

Kaikki kolme menetelmää sopivat suoritusdatan klusterointiin, mutta hierarkkisen klusteroinnin etuna on, että se muodostaa klusterointipuun, josta klustereiden muodostumista pystyy seuraamaan. Poikkeustapaukset ovat helposti nähtävissä kuten myös identtiset suoritukset. Hierarkkinen klusterointi oli ainoa, joka pystyi erottamaan omaksi klusterikseen opiskelijat, joilla ei ollut suorituksia. Vaikka hierarkkinen klusterointi antoikin yhden ison klusterin LuK- ja FM-tutkintoja suorittaville opiskelijoille, klusterointipuun avulla pystyy näkemään, miten tämä iso klusteri pilkkoutuu mielekkäämpiin osiin ja niitä pystyy Orangessa tarkastelemaan erikseen. Jos K-means tai EM-algoritmilla haluaisi tehdä samaa joutuu klusterien määrää ensin muuttamaan ja sitten tarkastelemaan datasta, miten data jakautui sillä kertaa klustereihin. EM-algoritmin eduksi voi lukea jäsenmäärältään tasaisemman kokoiset klusterit. Data saatiin jaettua jo kahdella klusterilla suhteellisen mielekkäästi, kun taas K-means teki kahdella klusterilla yhden ison klusterin ja toisen yhden tai kahden tapauksen kokoisen klusterin. EM ei kuitenkaan nosta poikkeavia tapauksia esiin samalla tavalla kuin K-means tai hierarkkinen klusterointi.

6.2 Kurssihahmot

Kurssihahmoja louhittiin, jotta löydettäisiin ensimmäisen opiskeluvuoden suosituimmat kurssihahmot ja havaittaisiin opiskelijoiden kiinnittyminen opintoihinsa. Kurssihahmoja louhittiin myös, jotta voitaisiin vertailla ilman korvattavia opintoja aloittavia opiskelijoita jo korkeakouluopintoja suorittaneisiin opiskelijoihin ja että saataisiin selville millaisia kurssihahmoja opiskelijat ovat hyväksilukeneet opintoihinsa ja mitä kursseja niiden lisäksi on otettu. Kurssihahmot eli kattavat joukot louhittiin Apriori-algoritmilla Wekassa, jossa kattavat joukot ryhmiteltiin kätevästi niiden sisältämien tietoalkioiden lukumäärän mukaan. Kurssien koodeja vastaavat nimet on annettu liitteessä 2 ja liitteessä 3 on lueteltu kaikki löydetyt kurssihahmot.

Suosittuja kurssihahmoja etsittiin koko datasta, lukuvuosista erikseen, alemman ja ylemmän korkeakoulututkinnon suorittajista ja vain ylemmän korkeakoulututkinnon suorittajista eli viidestä eri datajoukosta. Parhaiten koko datasta nousee esiin LuK- ja FM-tutkintojen suorittajat, koska heitä aloittaa enemmän tietojenkäsittelytieteiden tutkinto-ohjelmassa kuin pelkän FM-tutkinnon opiskelijoita. Tästä syystä data oli myös tärkeää

jakaa osiin. Koko datassa, jossa on 301 opiskelijaa, suosituimpia yksittäisiä kursseja olivat tietojenkäsittelyn yleis- ja perusopinnot LUOYY003 (196 opiskelijaa), TIEP1 (165), TIEP2 (140), TIEP3 (147), TIEP5 (134), TIEY2 (157) ja TIEY4 (139). Kattavien joukkojen minimituki oli 40% eli 120 tapausta. Useamman tietoalkion kokoiset kattavat joukot sisälsivät suosituimpien kurssien eri kombinaatiota, kuten suurimman joukon {LUOYY003, TIEP1, TIEP3, TIEY2}, jonka oli suorittanut 124 opiskelijaa. Kurssit LUOYY003 ja TIEP1 oli suorittanut myös moni ylemmän korkeakoulututkinnon opiskelija.

Vuoden 2017-2018 datasta, jossa on 150 tapausta, 45% minimituella (67 tapausta) suosituimmat yksittäiset kurssit olivat myös tietojenkäsittelyn yleis- ja perusopintoja LUOYY003 (108 opiskelijaa), TIEP1 (79), TIEP2 (74), TIEP3 (74), TIEY2 (81) ja TIEY4 (72). Useamman tietoalkion kokoiset kattavat joukot ovat näiden kurssien eri kombinaatioita, kuten suurin joukko {LUOYY003, TIEP2, TIEY2}, jonka oli suorittanut 67 opiskelijaa. Opiskelijoista 45% oli täten suorittanut orientoivat opinnot ja kaksi johdantokurssia. Tässäkin datassa korostuvat alempaa ja ylempää korkeakoulututkintoa suorittavat opiskelijat.

Vuoden 2018-2019 datassa on 151 opiskelijaa. Suosituimmat yksittäiset kurssit 45% minimituella (68 tapausta) olivat tietojenkäsittelyn yleis- ja perusopintoja LUOYY003 (88 opiskelijaa), TIEP1 (86), TIEP3 (73), TIEP5 (72) ja TIEY2 (76). Suuremmat kattavat joukot ovat näiden kurssien eri kombinaatioita. Kaksi suurinta kolmen tietoalkion kattavaa joukkoa olivat {LUOYY003, TIEP1, TIEP3} ja {LUOYY003, TIEP1, TIEP5}. Kummatkin kurssihahmot oli suorittanut 68 opiskelijaa. Orientoivien opintojen lisäksi oli suoritettu eniten tietokantojen ja ohjelmoinnin peruskursseja toisin kuin edeltävänä vuonna.

Alempaa ja ylempää korkeakoulututkintoa suorittavissa opiskelijoissa, joita on kummaltakin vuodelta yhteensä 228, nousivat suosituimmiksi yksittäisiksi kursseiksi 55% minimituella (125 tapausta) tietojenkäsittelyn yleis- ja perusopinnot LUOYY003 (179 opiskelijaa), TIEP1 (152), TIEP2 (140), TIEP3 (147), TIEP5 (132), TIEY2 (157) ja TIEY4 (139), jotka ovat samat kuin koko datan suositut kurssit. Suurimmat kurssihahmot olivat kolmen tietoalkion kokoisia ja niitä on 6 kappaletta. Ne ovat seuraavanlaisia:

- {LUOYY003, TIEP1, TIEP3}, 133 kappaletta
- {LUOYY003, TIEP1, TIEY2}, 129 kappaletta
- {LUOYY003, TIEP2, TIEY2}, 126 kappaletta
- {LUOYY003, TIEP3, TIEY2}, 128 kappaletta
- {LUOYY003, TIEY2, TIEY4}, 125 kappaletta
- {TIEP1, TIEP3, TIEY2}, 126 kappaletta

Tuen ollessa 125 tapausta, jää koko datasta löytynyt suurin kattava joukko {LUOYY003, TIEP1, TIEP3, TIEY2} pois. Jos tukea lasketaan 45 prosenttiin eli 103 tapaukseen, tulee

neljän tietoalkion kokoisia kattavia joukkoja 31 kappaletta, viiden tietoalkion kokoisia joukkoja 11 kappaletta ja yksi kuuden tietoalkion joukko {LUOYY003, TIEP1, TIEP2, TIEP3, TIEY2, TIEY4}, jonka on suorittanut 105 opiskelijaa. Näistä kurssihahmoista voidaan nähdä miten ja kuinka moni opiskelija on kiinnittynyt opintoihinsa. Jos lukuvuosia tarkastelisi tästä datasta erikseen, voitaisiin suosituista kurssihahmoista nähdä opiskelijoiden kiinnittyminen oman alansa opintoihin vielä tarkemmin.

Ylemmän korkeakoulututkinnon suorittavista opiskelijoista, joita on kummaltakin lukuvuodelta yhteensä 73, löytyi vähemmässä määrin yhtenäistä linjaa opintojen suhteen, kuten voi olettaa, koska tässä vaiheessa erikoistutaan useaan suuntaan. Kuten näitä opiskelijoita ryhmitellessä huomattiin, HTI-kursseja suorittaneet erottuivat toisista opiskelijoista. Näin on myös kattavia joukkoja louhiessa. HTI-kurssit HTIS54 (17 opiskelijaa), HTIS60 (16), HTIS79 (16), HTIS81 (23), HTIS85 (17), HTIY005 (18) ja HTIY006 (24) olivat osa suosituimpia yksittäisiä kursseja 20% minimituella (15 tapausta). Muita suosittuja yksittäisiä kursseja olivat yleiset ja kieliopinnot KKENMP3 (17 opiskelijaa), KKSU1 (24), LUOYY003 (17) ja LUOYY006 (20). Useamman tietoalkion kokoiset kattavat joukot koostuivat pelkästään HTI-opinnoista.

Ilman korvattavia opintoja aloittavien opiskelijoiden vertaamiseen opiskelijoihin, joilla on hyväksiluettuja korkeakouluopintoja käytettiin neljää eri datajoukkoa: yksi opiskelijoista, joilla ei ole korvattuja suorituksia, toinen opiskelijoista, joilla on korvattuja suorituksia ja kolmas ja neljäs datajoukko, joissa viimeksi mainittujen opiskelijoiden suoritukset on jaettu sen perusteella, onko ne korvattu vai ei.

Opiskelijoita, joilla ei ole hyväksiluettuja suorituksia, on 205 kappaletta koko datassa. Heidän suorituksistaan saadaan tulokseksi 40% minimituella (82 tapausta) tietojenkäsittelyn yleis- ja perusopintoja LUOYY003 (138 opiskelijaa), TIEP1 (97), TIEP2 (93), TIEP3 (94), TIEY2 (107) ja TIEY4 (93). Näiden kurssien eri kombinaatioista muodostuvat suuremmat kattavat joukot, joista suurin on joukko {LUOYY003, TIEP1, TIEP3, TIEY2}, jonka on suorittanut 82 opiskelijaa. Opiskelijoilla, joilla on korvattuja opintoja, nousee esiin myös tietojenkäsittelyn yleis- ja perusopintoja, koska LuK- ja FM-tutkintojen suorittajat ovat vahvasti edustettuina kummassakin ryhmässä. Opiskelijoita, joilla on korvattuja kursseja, on yhteensä 96. Yksittäiset kurssit, jotka saadaan heidän suorituksistaan esiin 50% minimituella (48 tapausta) ovat LUOYY003 (58 opiskelijaa), TIEP1 (68), TIEP3 (53), TIEP5 (53) ja TIEY2 (50). Suurimpia kurssihahmoja ovat joukot {LUOYY003, TIEP1, TIEP3}, {LUOYY003, TIEP1, TIEP5} ja {TIEP1, TIEP3, TIEP5}, joita on suorittanut 48 opiskelijaa (kaksi ensimmäistä) ja 49 opiskelijaa.

Opiskelijoiden, joilla on hyväksiluettuja suorituksia, korvatut opinnot eivät ole tietenkään yhtä yhtenäisiä, mutta näistäkin löytyy mielenkiintoinen linja. Korvatuista opinnoista nousee esiin 15% minimituella (14 tapausta) tietojenkäsittelyn yleis-, perus-, ja aineopintoja TIEP1 (34 opiskelijaa), TIEP2 (18), TIEP3 (23), TIEP4 (17), TIEP5 (16),

TIETA16 (15) ja TIEY4 (25) sekä yksi englannin kielen kurssi KKENYHT/LUK, jonka on korvannut 15 opiskelijaa. Suuremmat kattavat joukot koostuvat TIEP ja TIEY-kurssien eri kombinaatioista, joista suurin on neljän tietoalkion kurssihahmo {TIEP1, TIEP3, TIEP4, TIEP5}, jonka on korvannut 14 opiskelijaa. Suurin osa korvatuista tietojenkäsittelyn peruskursseista oli suoritettu alun perin avoimessa yliopistossa. Jos tukea laski 10 prosenttiin eli 10 tapaukseen, ilmeni että 10% opiskelijoista, joilla on hyväksiluettuja opintoja, oli jo suorittanut tietojenkäsittelytieteiden perusopinnot ennen tutkinto-ohjelmassa aloittamista. Kahta HTI-kurssia on myös hyväksiluettu kumpaakin 11 opiskelijalle, sillä ne on ollut tarkoituskin suorittaa Tampereen teknillisessä yliopistossa.

Opiskelijoiden, joilla on hyväksiluettuja kursseja, suorituksista, joita ei ole korvattu, saatiin myös mielenkiintoisia tuloksia. Opiskelijoita on tässä 93 eli kolme vähemmän kuin aiemmin, koska kaikki eivät olleet suorittaneet muita opintoja korvattujen opintojen lisäksi. Suositut yksittäiset kurssit 25% minimituella (23 tapausta) olivat tietojenkäsittelyn yleis-, perus- ja aineopintoja LUOYY003 (58 opiskelijaa), TIEA1 (34), TIEA2.1A (26), TIEA2.1B (23), TIEP1 (34), TIEP2 (29), TIEP3 (30), TIEP4 (28), TIEP5 (37), TIETA9 (27) ja TIEY2 (38) sekä matematiikan ja tilastotieteen opintoja MTTMY1 (27 opiskelijaa) ja MTTTP1 (28). Suuremmat kurssihahmot koostuvat näiden kurssien eri kombinaatioista. Tästä on havaittavissa, että osa opiskelijoista jatkaa tietojenkäsittelyn opintoja hyväksiluettujen suoritusten jälkeen perinteisen kaavan mukaan.

6.3 Kurssien suosittelu

Kurssien suosittelua varten tässä työssä käytetty data ei ole ihanteellista, sillä dataa ei ole usealta opiskeluvuodelta. Näin saataisiin enemmän erilaisia kurssien suorituksia sekä erilaisia kurssikombinaatioita, joita on suoritettu. Tämän työn datassa suosituimpia kursseja ovat LUOYY003, TIEY2, TIEY4, TIEP1, TIEP2, TIEP3, TIEP4, TIEP5 ja TIEA1, joista on yli 100 suoritusta kahdelta vuodelta ja jotka käydään yleensä ensimmäisenä lukuvuonna. Muita yli neljänkymmenenviiden suorituksen kursseja olivat TIEA2.1, TIEA2.1A, TIEA2.1B, TIETA9, MTTTP1, MTTMY1, KKENYHT/LUK, ITIY3 ja ITIP4.

Aherin ja Lobon ehdottamassa kurssien suosittelutavassa käytettiin dataan ensin K-means -algoritmia klusterointiin ja jokaisesta klusterista louhittiin erikseen assosiaatiosäännöt Apriorilla. Sen klusterin assosiaatiosäännöt, joissa kaikki kurssit (attribuutit) olivat ’yes’ -arvoisia, valittiin kurssien suositteluun. Suosituksiin he olivat valinneet kurssit, jotka on suorittanut vähintään sata opiskelijaa. Tämän työn datalla sadan opiskelijan rajoitus olisi sisältänyt vain 9 kurssia, joten rajaa laskettiin ensin 10 opiskelijaan. Samalla datasta poistettiin opiskelijat, joilla ei ollut ollenkaan suorituksia jäljelle jäävistä kursseista. K-means -klusteroinnin ja FP-growth -assosiaatioanalyysin jälkeen klustereista ei saatu haluttuja ykkössääntöjä esiin vaan vain nollasäännöt korostuivat. Ykkössäännöillä tarkoitetaan tässä assosiaatiosääntöjä, joissa kaikkien kurssien (attribuuttien) arvot ovat

ykkösiä eli kurssit on suoritettu ja nollasäännöillä tarkoitetaan sääntöjä, joissa esiintyy ainakin yhden kurssin arvona nolla eli ainakin yksi kurssi on suorittamaton.

Kun rajaa nostettiin 45 opiskelijaan per kurssi, tulivat ykkössäännöt paremmin esiin, mutta mistään klusterista ei löytynyt pelkästään ykkössääntöjä. Klustereista voisi rajata pois kurssit, joissa ei ole kyseisessä klusterissa montaa suoritusta, ennen assosiaatiosääntöjen louhintaa parantamaan ykkössääntöjen näkyvyyttä. Nolla-arvot voisi myös muuttaa puuttuviksi arvoiksi kuten tehtiin kurssihahmojen louhinnan kohdalla. Löydettyjen sääntöjen määrä on myös erilainen Orangessa ja Wekassa. Orangen FP-growth antaa hyvin paljon assosiaatiosääntöjä ja minimirajaus sääntöjen maksimimäärälle on 10000 sääntöä. Wekassa Apriorille pystyy vapaasti määrittämään sääntöjen maksimimäärän, joka on oletukseltaan 10 sääntöä. Tässä tapauksessa ei pystytä valitsemaan sopivaa klusteria ja sääntöjä kurssien suositteluun samalla tavalla kuin alkuperäisessä menetelmässä. Kolmella klusterilla on kuitenkin erotettavissa yksi klusteri, jossa ykkössääntöjä on enemmän esillä. Klusterien määrän nostaminen ei tuonut klustereissa ykkössääntöjä merkittävästi enempää näkyviin, joten valittiin mainittu klusteri ja assosiaatiosäännöt lähempään tarkasteluun. Assosiaatiosäännöt louhittiin 50% minimituella ja 90% minimiluottamuksella.

Valittujen assosiaatiosääntöjen joukossa on muun muassa seuraavanlaisia ykkössääntöjä:

- TIEP1 ⇒ TIEA2.1A [tuki = 100%, luottamus = 100%]
- TIEA2.1A ⇒ TIEP1 [tuki = 100%, luottamus = 100%]
- TIEP3 ⇒ TIEA2.1A [tuki = 100%, luottamus = 100%]
- TIEA2.1A ⇒ TIEP3 [tuki = 100%, luottamus = 100%]
- TIEP5 ⇒ TIEA2.1A [tuki = 100%, luottamus = 100%]
- TIEA2.1A ⇒ TIEP5 [tuki = 100%, luottamus = 100%]
- TIEP1 ⇒ TIEP3 [tuki = 100%, luottamus = 100%]
- TIEP3 ⇒ TIEP1 [tuki = 100%, luottamus = 100%]
- TIEP1 ⇒ TIEP5 [tuki = 100%, luottamus = 100%]
- TIEP5 ⇒ TIEP1 [tuki = 100%, luottamus = 100%]
- TIEP5 ⇒ TIEP3 [tuki = 100%, luottamus = 100%]
- TIEP3 ⇒ TIEP5 [tuki = 100%, luottamus = 100%]

Näistä voisi antaa esimerkiksi opiskelijalle, joka on ilmoittautunut kurssille tai jo suorittanut kurssin TIEP1, ehdotuksen suorittaa myös kurssit TIEA2.1A, TIEP3 ja TIEP5. Kurssit TIEP3 ja TIEP5 ovat mielekkäitä suosituksia, kun taas kurssi TIEA2.1A ei ole hyvä suositus, jos opiskelija ei ole käynyt kurssia TIEP5, sillä kurssit TIEP1 ja TIEP5 ovat kurssin TIEA2.1A suositellut edeltävyydet. Suosittelujärjestelmä voisi tarkistaa

ovatko ehdotukset järkeviä kurssien suoritusjärjestyksen kannalta ennen kuin niitä suositellaan opiskelijalle. Jo tästä pienestä määrästä assosiaatiosääntöjä nähdään, että niistä saadut ehdotukset eivät ole välttämättä suoraan käyttökelpoisia kurssien suosittelussa vaan suosittelujärjestelmän täytyisi esimerkiksi tehdä tarvittavia tarkastuksia. Jos kurssien suosittelussa hyödynnettäisiin sekvenssien louhintaa, kurssien suoritusjärjestyksen pystyisi ottamaan paremmin huomioon. Tosin osalla opiskelijoista kurssien todellinen suorittamisjärjestys ei välttämättä vastaa ohjeellisen lukujärjestyksen mukaista: jonkin kurssin suoritusajankohta voi viivästyä esimerkiksi opiskelijan omien aikataulujen tai sairastumisen vuoksi.

Sen sijaan, että käyttäisi K-means ja assosiaatioanalyysi -yhdistelmää voisi käyttää vain assosiaatioanalyysia kuten joissakin kurssien suosittelujärjestelmissä on tehty. Esimerkiksi samalla tavalla kuin etsittiin kurssihahmoja assosiaation avulla muuttamalla nollat tyhjiksi arvoiksi, mutta käyttämällä FP-growth -algoritmia, saadaan liitteessä 4 annetut säännöt, jotka on järjestetty luottamuksen mukaan laskevaan järjestykseen. Assosiaatiosäännöt louhittiin 40% minimituella ja 90% minimiluottamuksella. Suurin osa saaduista säännöistä on suosittelun kannalta väärin päin kuten sääntö TIEP3, TIEP5 ⇒ TIEP1 [tuki = 41%, luottamus = 98%]. Sekvenssien louhinta voisi auttaa myös tässä tapauksessa. Jos sääntöä käyttäisi suositteluun toisin päin, saataisiin mielekäs suosittelu: kun opiskelija on suorittanut kurssin TIEP1, hänelle voitaisiin ehdottaa kursseja TIEP3 ja TIEP5. Mutta jos säännön kääntää toisin päin, luottamus ei välttämättä pädekään. Tällä menetelmällä, eli muuttamalla datassa nollat puuttuviksi arvoiksi ja louhimalla assosiaatiosäännöt, saadaan kurssisuosituksia ilman datan karsimista ja päästään eroon nollasäännöistä.

Tässä työssä kokeiltiin myös Vialardin ja muiden luokittelupohjaista kurssien suosittelutapaa. Data oli muodossa suoritus per rivi ja attribuutteina olivat opiskelijan suoritettujen kurssien kokonaismäärä, kumulatiivinen arvosana lukukauden alussa, kurssin nimi ja kurssista saatu arvosana, joka muunnettiin luokkatunnukseksi onnistuminen tai epäonnistuminen. Tässä työssä data on muuten samassa muodossa, mutta arvosana on muutettu moniarvoisemmaksi luokkatunnukseksi. Luokat ovat erinomainen (ER), hyvä (HY), tyydyttävä (TY) tai hyväksytty (HYV). Datassa opiskelijan ensimmäisen lukukauden suorituksissa kurssien lukumäärä ja kumulatiivinen arvosana ovat 0. Dataan käytettiin C4.5-luokittelualgoritmia Wekassa ja tulokseksi saatiin karsittu päätöspuu. Päätöspuussa on jokaiselle kurssille oma oksa tai alipuu. Päätöspuun kaksi erikseen tarkasteluun valittua osaa ovat seuraavanlaisia (luokkatunnuksen jälkeen suluissa on annettu, kuinka monta tapausta sai kyseisen luokkatunnuksen/kuinka monta tapausta luokiteltiin väärin):

```
...
kurssin_koodi = TIEVA36: HYV (14.0)
kurssin_koodi = TIEY4: HYV (139.0)
kurssin_koodi = LUOYY003: HYV (287.0)
kurssin_koodi = TIEP2: TY (140.0/63.0)
kurssin_koodi = TIEP3
|   kurssit_yht <= 4: HV (140.0/63.0)
|   kurssit_yht > 4: TY (7.0/1.0)
...
kurssin_koodi = ITIA7: HV (3.0)
kurssin_koodi = JOVP2: HV (1.0)
kurssin_koodi = TIEA2.1
|   kumulatiivinen_arvosana <= 2.5
|   |   kurssit_yht <= 6
|   |   |   kurssit_yht <= 2: ER (5.0/2.0)
|   |   |   kurssit_yht > 2: HV (2.0)
|   |   kurssit_yht > 6: TY (2.0)
|   kumulatiivinen_arvosana > 2.5: ER (41.0/4.0)
...
```

Päätöspuusta ilmeni heti ensimmäisenä se, että ensimmäisen opiskeluvuoden data ei ole tähän menetelmään ihanteellista. Yksittäiset kurssisuoritukset jäävät haarautumattomiksi oksiksi, joissa on vain yksi lopputulos kuten esimerkiksi kurssin JOVP2 kohdalla. Samalla tavalla tapahtuu kursseille, joista on vain muutama suoritus datassa ja niistä on tullut sama tulos, kuten kurssissa ITIA7. Haarautumattomat oksat muodostuvat myös kursseille, joista on annettu vain hyväksytty merkintä, sillä datassa ei ole hylättyjä kurssimerkintöjä. Tällaisia kursseja ovat esimerkiksi TIEVA36, TIEY4 ja LUOYY003.

Tätä suosittelumenetelmää varten dataa on ensimmäisenä lukuvuotena liian vähän, jotta voitaisiin muodostaa profiileja erilaisista opiskelijatyypeistä opiskelijoiden kurssien yhteenlasketun määrän ja kumulatiivisen arvosanan perusteella lukukauden alussa. Kun opiskelijat aloittavat heillä ei ole suoritettuja kursseja eikä siten myöskään kumulatiivista arvosanaa. Vasta ensimmäisen lukukauden jälkeen voidaan kerätä mielekkäämpää dataa, mutta silloinkin sitä on vielä varsin vähän. Jos dataa olisi enemmän, luokittelu voisi toimia paremmin, mutta pelkästään suoritettujen kurssien lukumäärän ja kumulatiivisen arvosanan perusteella luokittelusta ei voi tulla niin tarkkaa, koska moni muukin asia vaikuttaa opiskelijan suoriutumiseen. Vaikka dataa olisi enemmän haasteen tuo se mistä kohtaa lukuvuosi katkaistaan. Kaikista kursseista ei kerkeä välttämättä saamaan arvosanaa syyslukukauden loppuun mennessä vaan vasta esimerkiksi tammikuussa. Tähän vaikuttavat esimerkiksi kurssin kesto, tenttimisen ajankohta (tenttiikö lopputentissä vai uusintatentissä) ja harjoitustyöt. Tässä kokeilussa lukukausien raja vedettiin tammikuun viimeiselle päivälle.

Jos katsotaan otteita päätöspuusta ja kurssia TIEP2, nähdään että luokittelu on antanut kurssille vain yhden lopputuloksen, johon noin puolet tapauksista on luokiteltu väärin eli niillä on eri lopputulos. Tämä johtuu siitä, että kurssi on otettu ensimmäisenä lukukautena

eli silloin kun sen suorittajilla ei ole ollut aiempia suorituksia eli kurssien määrä ja kumulatiivinen arvosana on ollut kaikilla 0. Tästä syystä opiskelijan ensimmäisen lukukauden data on ongelmallinen. Kurssin TIEP2 kohdalla suorituksen eri tuloksia ei pysty siis päättelemään muista attribuuteista, joten luokaksi on tullut yleisin tulos. Kurssin TIEP3 kohdalla taas suurin osa on käynyt kurssin ensimmäisenä lukukautena, mutta seassa on myös suorituksia, jotka on tehty 31.1. jälkeen. Jos jollekin opiskelijalle tämän perusteella suositeltaisiin kurssia TIEP3 ja että se menisi hyvin, koska opiskelija on käynyt neljä kurssia tai alle, ennuste suoriutumisesta näillä perusteilla olisi kuin arvaus eikä niinkään tämän opiskelijan aiempaan suoriutumiseen, ja muiden samalla tavalla suoriutuneiden opiskelijoiden suorituksiin, perustuva ennuste.

Kurssin TIEA2.1 oli suurin osa suorittanut toisella lukukaudellaan. Jotta vain heidän suorituksensa nähtäisiin päätöspuussa, datasta poistettiin suoritukset, joissa kurssien määrä ja kumulatiivinen arvosana on 0. Tämän jälkeen muodostettiin uusi päätöspuu C4.5-luokittelulla. Uudessa päätöspuussa kurssin TIEA2.1 alipuu on seuraavanlainen:

```
kurssin_koodi = TIEA2.1
|   kumulatiivinen_arvosana <= 2.5
|   |   kurssit_yht <= 6: HV (2.0)
|   |   kurssit_yht > 6: TY (2.0)
|   kumulatiivinen_arvosana > 2.5: ER (41.0/4.0)
```

Puusta käy ilmi, että kurssia voisi suositella opiskelijoille, joiden kumulatiivinen arvosana lukukauden alussa on yli 2,5 tai jos se on 2,5 tai alle, niin opiskelijalla tulisi olla suoritettuna 6 kurssia tai vähemmän. Suositellaan siis kurssia, jos opiskelija voisi pärjätä siinä hyvin tai erinomaisesti, mukaillen Vialardin ja muiden alkuperäistä tapaa. Puusta on myös pääteltävissä, että mitä paremmin aiemmat opinnot ovat menneet sitä paremmin tälläkin kurssilla suoriutuu. Puuta voisi täten käyttää myös yleiseen kurssilla suoriutumisen tarkasteluun, mutta silloin kannattaisi laskea kurssien määrä ja kumulatiivinen arvosana koko lukuvuodelta. Tässä tapauksessa dataa on kuitenkin liian vähän, että päätöspuun avulla voitaisiin tehdä oikeita suosituksia tai muita johtopäätöksiä.

Toisin kuin Aherin ja Lobon suosittelumenetelmää tai pelkkää assosiaatioanalyysiä, tätä suosittelutapaa ei voi käyttää juuri aloittaneille opiskelijoille, koska heillä ei ole aiempia suorituksia. Koska heillä kurssien määrä ja kumulatiivinen arvosana olisivat kummatkin 0, he saisivat suosituksia väärin perustein. Esimerkiksi heille voitaisiin suositella kurssia TIEA2.1, koska luokittelun mukaan heidän pitäisi suoriutua siinä hyvin. Muissa suosittelutavoissa riittäisi, että opiskelija ilmoittautuu jollekin kurssille tai useammalle ja hän voisi alkaa saamaan jo suosituksia toisista kursseista, joita opiskella.

7 Yhteenveto

Opiskelijoiden opintoihin kiinnittymistä tutkittiin klusteroimalla opiskelijat heidän ensimmäisen lukuvuoden suoritustensa perusteella vertaillen kolmea eri klusterointimenetelmää: K-means ja EM-algoritmeja sekä hierarkkista klusterointia. Opintoihin kiinnittymistä selvitettiin myös etsimällä suosittuja kurssihahmoja opiskelijoiden suorituksista assosiaatioanalyysin avulla käyttäen Apriori-algoritmia. Lisäksi kurssihahmojen avulla vertailtiin eri lähtökohdista aloittavia opiskelijoita. Kurssien suosittelua kokeiltiin luokittelupohjaisella menetelmällä hyödyntäen C4.5-luokittelualgoritmia ja klusteroinnin ja assosiaatioanalyysin yhdistelmällä käyttäen K-means ja FP-growth -algoritmeja. Tässä työssä myös arvioitiin tiedonlouhintamenetelmien soveltuvuutta opintoihin kiinnittymisen tutkimiseen ja kurssien suositteluun.

Saaduista tuloksista nähdään, että opiskelijoiden kiinnittymistä opintoihinsa voidaan tutkia ryhmittelemällä opiskelijat klusteroinnin avulla. Klusteroinnissa sopiva, todellisia opiskelijaryhmiä vastaava klusterien lukumäärä löytyy ainoastaan testaamalla, miten data jakautuu eri klusterimäärillä. Tästä johtuen eri opiskelijaryhmiä pystytään parhaiten tarkastelemaan hierarkkisella klusterointimenetelmällä Orange -louhintatyökalussa, koska klustereiden muodostumista voidaan seurata klusterointipuusta ja klustereita pystytään tarkastelemaan erikseen. Eri opiskelijaryhmistä käy ilmi oman alansa opintoihin kiinnittyneet ja yleisesti yliopisto-opintoihin kiinnittyneet opiskelijat sekä opiskelijat, jotka eivät ole tehneet opintoja lainkaan tai vain vähän oman alansa opintoja. Louhimalla suosittuja kurssihahmoja assosiaatioanalyysin avulla pystytään myös hahmottamaan opiskelijoiden kiinnittymistä oman alansa opintoihin. Lisäksi kurssihahmojen avulla voidaan verrata erilaisista lähtökohdista aloittavien opiskelijoiden suorituksia, kuten ilman korvaavuuksia aloittavien opiskelijoiden vertaaminen jo korkeakouluopintoja suorittaneisiin opiskelijoihin. Vaikka ryhmittelyn ja kurssihahmojen tulokset olivat odotettuja, niistä voidaan nähdä esimerkiksi tarkat lukumäärät erilaisista opiskelijaryhmistä ja saada tarkkaa tietoa opintoihin kiinnittymisen tavoista. Tuloksia voidaan hyödyntää esimerkiksi tutorryhmien tai muiden ryhmien muodostamisessa, jos halutaan tuottaa taustatiedoiltaan homogeenisia ryhmiä, esimerkiksi laittamalla paljon korvaavuuksia saaneet opiskelijat omaan ryhmäänsä. Tulosten perusteella voisi myös mahdollisesti hienosäätää opetustarjontaa. Lisäksi menetelmien avulla saatavat tulokset voivat antaa arvokasta tietoa, kun eri vuosien tuloksia verrataan toisiinsa, jos esimerkiksi tehdään muutoksia opetussuunnitelmaan tai opetustarjontaan. Tässä työssä keskityttiin opiskelijoiden ensimmäiseen lukuvuoteen, mutta opiskelijoiden ryhmittely ja kurssihahmot ovat erittäin hyödyllisiä myös myöhemmin.

Kurssien suosittelu jäi tässä työssä eri suosittelumenetelmien kokeilun tasolle, koska muita louhintatehtäviä varten tarvittu data ei ollut yhtä sopivaa kurssien suositteluun. Eri

suosittelumenetelmiä testatessa pystyi kuitenkin vertailemaan niitä ja niiden sopivuutta ensimmäisen lukuvuoden opiskelijoille. Tarkoitukseen paremmin sopivalla datalla kurssien suosittelua olisi pystynyt tutkimaan perusteellisemmin.

Louhinnassa on syytä kiinnittää huomiota datan valintaan. Tässä työssä jo alkuperäisen datan valinnassa olisi ollut syytä ottaa huomioon kurssien suosittelu, johon muihin tehtäviin sopiva data ei ollutkaan yhtä hyvää. Dataa on myös mielekästä jakaa osiin louhiessa, jotta tulokset, joista ollaan kiinnostuneita, tulevat paremmin esille. Esimerkiksi opiskelijoiden jakaminen ryhmiin kannatti tehdä LuK- ja FM-tutkintoja suorittaville ja vain FM-tutkintoa suorittaville opiskelijoille erikseen, jotta saatiin näkyviin kiinnostavat ryhmät. Myös kurssihahmoja oli hyvä etsiä näistä opiskelijaryhmistä myös erikseen, koska LuK- ja FM-tutkintoja suorittavia opiskelijoita on datassa enemmän, joten heidän suorituksensa tulevat vahvemmin esille kurssihahmoissa.

Vaikka suurista datamääristä saa louhinnassa usein parempia tuloksia, tässä työssä pienestä datasta löydettiin kiinnostavia tuloksia ja datan koosta oli myös etua. Dataa oli helpompi tarkastella pienen kokonsa vuoksi kuten myös louhinnan tuloksia. Myös louhinta-algoritmien ajot olivat nopeita, koska dataa oli vähemmän eikä datasta tarvinnut määrän takia ottaa vain satunnaisia otoksia louhittavaksi.

Louhinnassa huomattiin, että käytetyt työkalut vaikuttavat paljon louhinnan sujuvuuteen ja tulosten tarkasteluun. Tässä työssä käytettiin kahta työkalua, Orangea ja Wekaa, koska kumpikaan ei tarjonnut yksin kaikkia tässä työssä tarvittavia algoritmeja. Työvuo ja datan ja tulosten esitystavat ovat kummassakin työkalussa erilaiset ja siksi työssä käytettiin esimerkiksi assosiaatioanalyysiin vaihtelevasti Orangen tarjoamaa FP-growthia ja Wekan tarjoamaa Aprioria. Opiskelijoiden ryhmittelyn tarkasteluun parhaiten sopiva hierarkkinen klusterointi ei välttämättä olisi ollut yhtä hyvä kandidaatti, ellei sen muodostamaa klusterointipuuta olisi ollut niin vaivatonta tarkastella Orangessa.

Tiedonlouhinnan käyttö opintojen ohjauksen apuna vaatisi opetus- ja ohjaushenkilökunnalta louhintatyökalujen käyttöä. Niiden käyttö edellyttää kuitenkin yleistä tiedonlouhinnan tuntemusta ja jotta tuloksia ymmärtäisi paremmin olisi hyvä tietää myös, miten käytetyt algoritmit toimivat. Louhinnassa voi helposti jäädä jumiin, jos ei tiedä mitä parametria tai miten sitä tulisi säätää, jotta saisi parempia louhintatuloksia. Näistä syistä tiedonlouhintaa voi olla vaikea ottaa käyttöön opintojen ohjauksen avuksi. Tietysti tiedonlouhinnan asiantuntijoita voidaan konsultoida louhintaprosessin aikana. Mikään ei myöskään estä korkeakouluja kehittämästä heidän omiin tarkoituksiinsa sopivampaa louhintatyökalua, joka ei vaadi käyttäjältä syvää ymmärrystä tiedonlouhinnasta.

8 Viiteluettelo

Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast algorithms for mining association rules. In: *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB) Conference*, 487–499.

Sunita B. Aher and L.M.R.J. Lobo. 2012. Combination of clustering, classification and association rule based approach for course recommender system in e-learning. *International Journal of Computer Applications*, 39, 7, 8-15.

Sunita B. Aher and L.M.R.J. Lobo. 2013. Combination of machine learning algorithms for recommendation of courses in e-learning system based on historical data. *Knowledge-Based Systems*, 51, 1-14.

Shahrokh Asadi, Seyed Mohammadbagher Jafari and Zohreh Shokrollahi. 2019. Developing a course recommender by combining clustering and fuzzy association rules. *Journal of AI and Data Mining*, 7, 2, 249-262.

Ryan S.J.D. Baker and Kalina Yasef. 2009. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, 1, 1, 3-16.

Narimel Bendakir and Esma Aïmeur. 2006. Using association rules for course recommendation. In: *Proceedings of the AAAI Workshop on Educational Data Mining*.

Seung-Seok Choi, Sung-Hyuk Cha and Charles C. Tappert. 2010. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8, 1, 43-48.

Janez Demšar and Blaž Zupan. 2013. Orange: Data mining fruitful and fun - A historical perspective. *Informatica*, 37, 55-60.

Ashish Dutt, Saeed Aghabozrgi, Maizatul Akmal Binti Ismail and Hamidreza Mahroeian. 2015. Clustering algorithms applied in educational data mining. *International Journal of Information and Electronics Engineering*, 5, 2, 112-116.

Monika Goyal and Rajan Vohra. 2012. Applications of data mining in higher education. *IJCSI International Journal of Computer Science Issues*, 9, 2, 113-120.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations*, 11, 1, 10-18.

Jiawei Han, Jian Pei, Yiwen Yin and Runying Mao. 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8, 1, 53–87.

Jiawei Han, Micheline Kamber and Jian Pei. 2012. *Data Mining: Concepts and Techniques (3rd ed)*. Elsevier/Morgan Kaufmann.

IBM SPSS Modeler. 2020. SPSS Modeler – Overview. https://www.ibm.com/products/spss-modeler. Checked 5.4.2020.

Intelligent Miner. 2020. Intelligent Miner. https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.dwe.navigate.doc/c_data_mining_container.html. Checked 5.4.2020.

Jing Luan. 2002. Data mining and knowledge management in higher education – Potential applications. Paper presented at the Annual Forum for the Association for Institutional Research (42nd, Toronto, Ontario, Canada, June 2-5, 2002)

V.L. Miguéis, Ana Freitas, Paulo J.V. Garcia and André Silva. 2018. Early segmentation of students according to their academic performance: A predictive modelling approach. *Decision Support Systems*, 115, 36-51.

Srečko Natek and Moti Zwilling. 2014. Student data mining solution – Knowledge management system related to higher education institutions. *Expert Systems with Applications*, 41, 14, 6400-6407.

Orange. 2020. Orange Data Mining. https://orange.biolab.si/. Checked 5.4.2020.

Suhem Parack, Zain Zahid and Fatima Merchant. 2012. Application of data mining in educational databases for predicting academic trends and patterns. In: *2012 IEEE International Conference on Technology Enhanced Education (ICTEE)*, 1-4.

RapidMiner. 2020. RapidMiner - Best Data Science & Machine Learning Platform. https://rapidminer.com/. Checked 5.4.2020.

Cristobal Romero and Sebastian Ventura. 2007. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33, 1, 135-146.

Evis Trandafili, Alban Allkoçi, Elinda Kajo and Aleksandër Xhuvani. 2012. Discovery and evaluation of student's profiles with machine learning. In: *BCI '12 Proceedings of the Fifth Balkan Conference in Informatics*, 174-179.

César Vialardi, Javier Bravo, Leila Shafti and Álvaro Ortigosa. 2009. Recommendation in higher education using data mining techniques. Paper presented at the International Conference on Educational Data Mining (EDM) (2nd, Cordoba, Spain, July 1-3, 2009)

Weka. 2020. WEKA – The workbench for machine learning. https://www.cs.waikato.ac.nz/ml/weka/index.html. Checked 5.4.2020.

Ian H. Witten, Eibe Frank and Mark A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques (3rd ed)*. Elsevier/Morgan Kaufmann.

Chris Wong. 2018. Sequence based course recommender for personalized curriculum planning. In: *Penstein Rosé C. et al. (eds) Artificial Intelligence in Education, AIED 2018, Lecture Notes in Computer Science, vol 10948*, Springer, 531-534.

Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua

Zhou, Michael Steinbach, David J. Hand and Dan Steinberg. 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14, 1, 1-37.

Hao Zhang, Tao Huang, Zhihan Lv, SanYa Liu and Zhili Zhou. 2018. MCRS: A course recommendation system for MOOCs. *Multimedia Tools and Applications*, 77, 7051–7069.

Liite 1: Esimerkkejä alkuperäisen datan muodosta

Esimerkkejä opiskelijoiden tiedoista:

opiskelijan pseudotunniste	aloitusvuosi	opinto-oikeus	koulutusohjelma	ilmoittautuminen: syksy	ilmoittautuminen: kevät
1	2018	LuF2	Tietojenkäsittelytieteiden tutkinto-ohjelma	L	L
2	2018	FM	Master's Degree Programme in Software Development	L	L
3	2018	LuF2	Tietojenkäsittelytieteiden tutkinto-ohjelma	P	L
4	2018	FM	MDP in Computational Big Data Analytics, Computer Sciences	L	L
5	2017	LuF2	Tietojenkäsittelytieteiden tutkinto-ohjelma	P	P
6	2017	LuF2	Tietojenkäsittelytieteiden tutkinto-ohjelma	L	P
7	2017	FM	Tietojenkäsittelyopin maisteriopinnot	L	L
8	2017	FM	Master's Degree Programme in Human-Technology Interaction	L	L

Esimerkkejä opiskelijoiden suorituksista:

opiskelijan pseudotunniste	opintojakson koodi	opintojakson nimi	arvosana	suoritus-päivä/hyväksiluku-päivä	opinto-pisteet	korvattu	alkuperäinen suoritus-paikka	alkuperäinen suoritus-päivä	lisätieto hyväksiluvusta
1	TIEP1	LAUSEKIELINEN OHJELMOINTI I	5	30/10/2018	5				
1	TIEP5	LAUSEKIELINEN OHJELMOINTI II	4	19/01/2019	5				
1	TIEY2	JOHDATUS TIETO-JENKÄSITTELY-TIETEISIIN	HYV	25/10/2018	3				
1	TIEA2.1A	OLIO-OHJELMOIN-NIN PERUSTEET I	4	27/03/2019	5				
2	TIETA8	SOFTWARE EN-GINEERING	HYV	20/01/2019	5				
2	MUUD9067	TIE-05206 MICRO-PROSESSORS	3	15/09/2018	4	K	SF TAM-PERE02	21/05/2016	
2	MUUD9048	TIE-31106 CRYP-TOGRAPHY EN-GINEERING	5	15/09/2018	5	K	SF TAM-PERE02	14/03/2016	

Liite 2: Opintojaksojen koodeja vastaavat nimet

HTIS54	EMOTIONS AND SOCIALITY IN HUMAN-TECHNOLOGY INTERACTION
HTIS60	INTERACTION TECHNIQUES
HTIS67	USER EXPERIENCE: DESIGN AND EVALUATION (TUT/IHTE)
HTIS79	MULTIMODAL INTERACTION
HTIS80	PSYCHOLOGY OF PERVASIVE COMPUTING (TUT/IHTE)
HTIS81	USABILITY EVALUATION METHODS
HTIS85	METHODS IN HUMAN-CENTERED DESIGN
HTIY005	STUDY SKILLS - BASICS OF INFORMATION LITERACY
HTIY006	ORIENTATION
ITIA7	GAMES AND INTERNET IN SOCIETY
ITIP4	PELIT JA PELILLISYYS
ITIY3	VERKKOJULKAISEMISEN PERUSTEET
JOVP2	VUOROVAIKUTUKSEN PERUSTEET
KKENYHT/LUK	INTRODUCTION TO ACADEMIC ENGLISH
KKENMP3	SCIENTIFIC WRITING - MODULE A - WRITING CLINIC
KKSU1	SUOMEN KIELEN ALKEISKURSSI 1
LUOYY003	ORIENTOIVAT OPINNOT, ALKUORIENTAATIO
LUOYY003	ORIENTOIVAT OPINNOT, TIEDONHANKINTATAIDOT I
LUOYY006	ORIENTATION
MTTMY1	MATEMATIIKAN PERUSKÄSITTEITÄ
MTTTP1	TILASTOTIETEEN JOHDANTOKURSSI
TIEA1	KÄYTTÖLIITTYMIEN PERUSTEET
TIEA2.1	OLIO-OHJELMOINNIN PERUSTEET
TIEA2.1A	OLIO-OHJELMOINNIN PERUSTEET I
TIEA2.1B	OLIO-OHJELMOINNIN PERUSTEET II
TIEP1	LAUSEKIELINEN OHJELMOINTI I
TIEP2	JOHDATUS VUOROVAIKUTTEISEEN TEKNOLOGIAAN
TIEP3	TIETOKANTOJEN PERUSTEET
TIEP4	TIETOJÄRJESTELMÄN SUUNNITTELUN PERUSTEET
TIEP5	LAUSEKIELINEN OHJELMOINTI II
TIEPT	TIETOJENKÄSITTELYTIETEIDEN PERUSOPINNOT
TIETA9	JOHDATUS WWW-TEKNIIKOIHIN
TIETA16	ERIKSEEN SOVITTAVA TIETOJENKÄSITTELYTIETEIDEN OPINTOJAKSO
TIEVA36	PRINCIPLES OF USABILITY, USER EXPERIENCE AND USER INTERFACES

TIEY2	JOHDATUS TIETOJENKÄSITTELYTIETEISIIN
TIEY4	TIETOTEKNIIKKATAIDOT

Liite 3: Kurssihahmojen (kattavien joukkojen) louhinnan tulokset

Koko datan kurssihahmot:
Tapauksia: 301
Attribuutteja: 335
Minimituki: 40% (120 tapausta)

Kattavat joukot:

Yhden tietoalkion kokoiset kattavat joukot: 7 kpl
LUOYY003 (196)
TIEP1 (165)
TIEP2 (140)
TIEP3 (147)
TIEP5 (134)
TIEY2 (157)
TIEY4 (139)

Kahden tietoalkion kokoiset kattavat joukot: 18 kpl
LUOYY003, TIEP1 (144)
LUOYY003, TIEP2 (134)
LUOYY003, TIEP3 (139)
LUOYY003, TIEP5 (129)
LUOYY003, TIEY2 (148)
LUOYY003, TIEY4 (135)
TIEP1, TIEP2 (125)
TIEP1, TIEP3 (139)
TIEP1, TIEP5 (130)
TIEP1, TIEY2 (132)
TIEP1, TIEY4 (121)
TIEP2, TIEP3 (127)
TIEP2, TIEY2 (131)
TIEP2, TIEY4 (122)
TIEP3, TIEP5 (124)
TIEP3, TIEY2 (131)
TIEP3, TIEY4 (121)
TIEY2, TIEY4 (128)

Kolmen tietoalkion kokoiset kattavat joukot: 14 kpl
LUOYY003, TIEP1, TIEP2 (122)
LUOYY003, TIEP1, TIEP3 (133)
LUOYY003, TIEP1, TIEP5 (125)
LUOYY003, TIEP1, TIEY2 (129)
LUOYY003, TIEP2, TIEP3 (124)
LUOYY003, TIEP2, TIEY2 (126)
LUOYY003, TIEP3, TIEP5 (120)

LUOYY003, TIEP3, TIEY2 (128)
LUOYY003, TIEY2, TIEY4 (125)
TIEP1, TIEP2, TIEP3 (120)
TIEP1, TIEP2, TIEY2 (120)
TIEP1, TIEP3, TIEP5 (122)
TIEP1, TIEP3, TIEY2 (126)
TIEP2, TIEP3, TIEY2 (121)

Neljän tietoalkion kokoiset kattavat joukot: 1 kpl
LUOYY003, TIEP1, TIEP3, TIEY2 (124)

Vuoden 2017-2018 kurssihahmot:
Tapauksia: 150
Attribuutteja: 196
Minimituki: 45% (67 tapausta)

Kattavat joukot:

Yhden tietoalkion kokoiset kattavat joukot: 6 kpl
LUOYY003 (108)
TIEP1 (79)
TIEP2 (74)
TIEP3 (74)
TIEY2 (81)
TIEY4 (72)

Kahden tietoalkion kokoiset kattavat joukot: 9 kpl
LUOYY003, TIEP1 (71)
LUOYY003, TIEP2 (72)
LUOYY003, TIEP3 (71)
LUOYY003, TIEY2 (77)
LUOYY003, TIEY4 (70)
TIEP1, TIEP3 (68)
TIEP1, TIEY2 (67)
TIEP2, TIEY2 (69)
TIEP3, TIEY2 (67)

Kolmen tietoalkion kokoiset kattavat joukot: 1 kpl
LUOYY003, TIEP2, TIEY2 (67)

Vuoden 2018-2019 kurssihahmot:
Tapauksia:151
Attribuutteja: 243
Minimituki: 45% (68 tapausta)

Kattavat joukot:

Yhden tietoalkion kokoiset kattavat joukot: 5 kpl
LUOYY003 (88)
TIEP1 (86)
TIEP3 (73)
TIEP5 (72)
TIEY2 (76)

Kahden tietoalkion kokoiset kattavat joukot: 6 kpl
LUOYY003, TIEP1 (73)
LUOYY003, TIEP3 (68)
LUOYY003, TIEP5 (69)
LUOYY003, TIEY2 (71)
TIEP1, TIEP3 (71)
TIEP1, TIEP5 (71)

Kolmen tietoalkion kokoiset kattavat joukot: 2 kpl
LUOYY003, TIEP1, TIEP3 (68)
LUOYY003, TIEP1, TIEP5 (68)

Alempaa ja ylempää korkeakoulututkintoa suorittavien opiskelijoiden kurssihahmot:
Tapauksia: 228
Attribuutteja: 234

Kattavat joukot (minimituki 55 % (125 tapausta)):

Yhden tietoalkion kokoiset kattavat joukot: 7 kpl
LUOYY003 (179)
TIEP1 (152)
TIEP2 (140)
TIEP3 (147)
TIEP5 (132)
TIEY2 (157)
TIEY4 (139)

Kahden tietoalkion kokoiset kattavat joukot: 14 kpl
LUOYY003, TIEP1 (143)
LUOYY003, TIEP2 (134)
LUOYY003, TIEP3 (139)
LUOYY003, TIEP5 (127)
LUOYY003, TIEY2 (148)
LUOYY003, TIEY4 (135)
TIEP1, TIEP2 (125)
TIEP1, TIEP3 (139)
TIEP1, TIEP5 (129)
TIEP1, TIEY2 (132)
TIEP2, TIEP3 (127)

TIEP2, TIEY2 (131)
TIEP3, TIEY2 (131)
TIEY2, TIEY4 (128)

Kolmen tietoalkion kokoiset kattavat joukot: 6 kpl
LUOYY003, TIEP1, TIEP3 (133)
LUOYY003, TIEP1, TIEY2 (129)
LUOYY003, TIEP2, TIEY2 (126)
LUOYY003, TIEP3, TIEY2 (128)
LUOYY003, TIEY2, TIEY4 (125)
TIEP1, TIEP3, TIEY2 (126)

Kattavat joukot (minimituki 45 % (103 tapausta)):

Yhden tietoalkion kokoiset kattavat joukot: 9 kpl
LUOYY003 (179)
TIEA1 (106)
TIEP1 (152)
TIEP2 (140)
TIEP3 (147)
TIEP4 (117)
TIEP5 (132)
TIEY2 (157)
TIEY4 (139)

Kahden tietoalkion kokoiset kattavat joukot: 27 kpl
LUOYY003, TIEP1 (143)
LUOYY003, TIEP2 (134)
LUOYY003, TIEP3 (139)
LUOYY003, TIEP4 (113)
LUOYY003, TIEP5 (127)
LUOYY003, TIEY2 (148)
LUOYY003, TIEY4 (135)
TIEA1, TIEP2 (105)
TIEP1, TIEP2 (125)
TIEP1, TIEP3 (139)
TIEP1, TIEP4 (112)
TIEP1, TIEP5 (129)
TIEP1, TIEY2 (132)
TIEP1, TIEY4 (121)
TIEP2, TIEP3 (127)
TIEP2, TIEP4 (104)
TIEP2, TIEP5 (111)
TIEP2, TIEY2 (131)
TIEP2, TIEY4 (122)
TIEP3, TIEP4 (113)
TIEP3, TIEP5 (124)

TIEP3, TIEY2 (131)
TIEP3, TIEY4 (121)
TIEP4, TIEY2 (108)
TIEP5, TIEY2 (118)
TIEP5, TIEY4 (108)
TIEY2, TIEY4 (128)

Kolmen tietoalkion kokoiset kattavat joukot: 40 kpl
LUOYY003, TIEP1, TIEP2 (122)
LUOYY003, TIEP1, TIEP3 (133)
LUOYY003, TIEP1, TIEP4 (109)
LUOYY003, TIEP1, TIEP5 (124)
LUOYY003, TIEP1, TIEY2 (129)
LUOYY003, TIEP1, TIEY4 (119)
LUOYY003, TIEP2, TIEP3 (124)
LUOYY003, TIEP2, TIEP5 (109)
LUOYY003, TIEP2, TIEY2 (126)
LUOYY003, TIEP2, TIEY4 (119)
LUOYY003, TIEP3, TIEP4 (109)
LUOYY003, TIEP3, TIEP5 (120)
LUOYY003, TIEP3, TIEY2 (128)
LUOYY003, TIEP3, TIEY4 (119)
LUOYY003, TIEP4, TIEY2 (105)
LUOYY003, TIEP5, TIEY2 (116)
LUOYY003, TIEP5, TIEY4 (106)
LUOYY003, TIEY2, TIEY4 (125)
TIEP1, TIEP2, TIEP3 (120)
TIEP1, TIEP2, TIEP5 (109)
TIEP1, TIEP2, TIEY2 (120)
TIEP1, TIEP2, TIEY4 (113)
TIEP1, TIEP3, TIEP4 (109)
TIEP1, TIEP3, TIEP5 (122)
TIEP1, TIEP3, TIEY2 (126)
TIEP1, TIEP3, TIEY4 (117)
TIEP1, TIEP4, TIEY2 (103)
TIEP1, TIEP5, TIEY2 (115)
TIEP1, TIEP5, TIEY4 (106)
TIEP1, TIEY2, TIEY4 (116)
TIEP2, TIEP3, TIEP5 (109)
TIEP2, TIEP3, TIEY2 (121)
TIEP2, TIEP3, TIEY4 (114)
TIEP2, TIEP5, TIEY2 (108)
TIEP2, TIEY2, TIEY4 (116)
TIEP3, TIEP4, TIEY2 (104)
TIEP3, TIEP5, TIEY2 (114)
TIEP3, TIEP5, TIEY4 (105)
TIEP3, TIEY2, TIEY4 (115)

TIEP5, TIEY2, TIEY4 (105)

Neljän tietoalkion kokoiset kattavat joukot: 31 kpl
LUOYY003, TIEP1, TIEP2, TIEP3 (118)
LUOYY003, TIEP1, TIEP2, TIEP5 (107)
LUOYY003, TIEP1, TIEP2, TIEY2 (117)
LUOYY003, TIEP1, TIEP2, TIEY4 (111)
LUOYY003, TIEP1, TIEP3, TIEP4 (106)
LUOYY003, TIEP1, TIEP3, TIEP5 (118)
LUOYY003, TIEP1, TIEP3, TIEY2 (124)
LUOYY003, TIEP1, TIEP3, TIEY4 (115)
LUOYY003, TIEP1, TIEP5, TIEY2 (113)
LUOYY003, TIEP1, TIEP5, TIEY4 (104)
LUOYY003, TIEP1, TIEY2, TIEY4 (114)
LUOYY003, TIEP2, TIEP3, TIEP5 (107)
LUOYY003, TIEP2, TIEP3, TIEY2 (118)
LUOYY003, TIEP2, TIEP3, TIEY4 (112)
LUOYY003, TIEP2, TIEP5, TIEY2 (106)
LUOYY003, TIEP2, TIEY2, TIEY4 (114)
LUOYY003, TIEP3, TIEP5, TIEY2 (112)
LUOYY003, TIEP3, TIEP5, TIEY4 (103)
LUOYY003, TIEP3, TIEY2, TIEY4 (113)
LUOYY003, TIEP5, TIEY2, TIEY4 (103)
TIEP1, TIEP2, TIEP3, TIEP5 (107)
TIEP1, TIEP2, TIEP3, TIEY2 (116)
TIEP1, TIEP2, TIEP3, TIEY4 (110)
TIEP1, TIEP2, TIEP5, TIEY2 (106)
TIEP1, TIEP2, TIEY2, TIEY4 (110)
TIEP1, TIEP3, TIEP5, TIEY2 (112)
TIEP1, TIEP3, TIEP5, TIEY4 (104)
TIEP1, TIEP3, TIEY2, TIEY4 (113)
TIEP1, TIEP5, TIEY2, TIEY4 (103)
TIEP2, TIEP3, TIEP5, TIEY2 (106)
TIEP2, TIEP3, TIEY2, TIEY4 (109)

Viiden tietoalkion kokoiset kattavat joukot: 11 kpl
LUOYY003, TIEP1, TIEP2, TIEP3, TIEP5 (105)
LUOYY003, TIEP1, TIEP2, TIEP3, TIEY2 (114)
LUOYY003, TIEP1, TIEP2, TIEP3, TIEY4 (108)
LUOYY003, TIEP1, TIEP2, TIEP5, TIEY2 (104)
LUOYY003, TIEP1, TIEP2, TIEY2, TIEY4 (108)
LUOYY003, TIEP1, TIEP3, TIEP5, TIEY2 (110)
LUOYY003, TIEP1, TIEP3, TIEY2, TIEY4 (111)
LUOYY003, TIEP2, TIEP3, TIEP5, TIEY2 (104)
LUOYY003, TIEP2, TIEP3, TIEY2, TIEY4 (107)
TIEP1, TIEP2, TIEP3, TIEP5, TIEY2 (104)
TIEP1, TIEP2, TIEP3, TIEY2, TIEY4 (107)

Kuuden tietoalkion kokoiset kattavat joukot: 1 kpl
LUOYY003, TIEP1, TIEP2, TIEP3, TIEY2, TIEY4 (105)

Ylempää korkeakoulututkintoa suorittavien opiskelijoiden kurssihahmot:
Tapauksia: 73
Attribuutteja: 161
Minimituki: 20% (15 tapausta)

Kattavat joukot:

Yhden tietoalkion kokoiset kattavat joukot: 11 kpl
HTIS54 (17)
HTIS60 (16)
HTIS79 (16)
HTIS81 (23)
HTIS85 (17)
HTIY005 (18)
HTIY006 (24)
KKENMP3 (17)
KKSU1 (24)
LUOYY003 (17)
LUOYY006 (20)

Kahden tietoalkion kokoiset kattavat joukot: 11 kpl
HTIS54, HTIS81 (17)
HTIS54, HTIY006 (17)
HTIS60, HTIS81 (16)
HTIS60, HTIY006 (16)
HTIS79, HTIS81 (16)
HTIS79, HTIY006 (16)
HTIS81, HTIS85 (16)
HTIS81, HTIY005 (18)
HTIS81, HTIY006 (23)
HTIS85, HTIY006 (16)
HTIY005, HTIY006 (18)

Kolmen tietoalkion kokoiset kattavat joukot: 5 kpl
HTIS54, HTIS81, HTIY006 (17)
HTIS60, HTIS81, HTIY006 (16)
HTIS79, HTIS81, HTIY006 (16)
HTIS81, HTIS85, HTIY006 (16)
HTIS81, HTIY005, HTIY006 (18)

Opiskelijoiden, joilla ei ole hyväksiluettuja kursseja, kurssihahmot:
Tapauksia: 205
Attribuutteja: 197

Minimituki: 40% (82 tapausta)

Kattavat joukot:

Yhden tietoalkion kokoiset kattavat joukot: 6 kpl
LUOYY003 (138)
TIEP1 (97)
TIEP2 (93)
TIEP3 (94)
TIEY2 (107)
TIEY4 (93)

Kahden tietoalkion kokoiset kattavat joukot: 11 kpl
LUOYY003, TIEP1 (93)
LUOYY003, TIEP2 (90)
LUOYY003, TIEP3 (90)
LUOYY003, TIEY2 (102)
LUOYY003, TIEY4 (91)
TIEP1, TIEP3 (87)
TIEP1, TIEY2 (86)
TIEP2, TIEP3 (82)
TIEP2, TIEY2 (88)
TIEP3, TIEY2 (87)
TIEY2, TIEY4 (86)

Kolmen tietoalkion kokoiset kattavat joukot: 6 kpl
LUOYY003, TIEP1, TIEP3 (85)
LUOYY003, TIEP1, TIEY2 (86)
LUOYY003, TIEP2, TIEY2 (86)
LUOYY003, TIEP3, TIEY2 (86)
LUOYY003, TIEY2, TIEY4 (85)
TIEP1, TIEP3, TIEY2 (82)

Neljän tietoalkion kokoiset kattavat joukot: 1 kpl
LUOYY003, TIEP1, TIEP3, TIEY2 (82)

Opiskelijoiden, joilla on hyväksiluettuja opintoja, kurssihahmot:
Tapauksia: 96
Attribuutteja: 246
Minimituki: 50% (48 tapausta)

Kattavat joukot:

Yhden tietoalkion kokoiset kattavat joukot: 5 kpl
LUOYY003 (58)
TIEP1 (68)
TIEP3 (53)

TIEP5 (53)
TIEY2 (50)

Kahden tietoalkion kokoiset kattavat joukot: 6 kpl
LUOYY003, TIEP1 (51)
LUOYY003, TIEP3 (49)
LUOYY003, TIEP5 (48)
TIEP1, TIEP3 (52)
TIEP1, TIEP5 (53)
TIEP3, TIEP5 (49)

Kolmen tietoalkion kokoiset kattavat joukot: 3 kpl
LUOYY003, TIEP1, TIEP3 (48)
LUOYY003, TIEP1, TIEP5 (48)
TIEP1, TIEP3, TIEP5 (49)

Opiskelijoiden, joilla on hyväksiluettuja opintoja, korvattujen suoritusten kurssihahmot:
Tapauksia: 96
Attribuutteja: 136

Kattavat joukot (minimituki 15% (14 tapausta)):

Yhden tietoalkion kokoiset kattavat joukot: 8 kpl
KKENYHT/LUK (15)
TIEP1 (34)
TIEP2 (18)
TIEP3 (23)
TIEP4 (17)
TIEP5 (16)
TIETA16 (15)
TIEY4 (25)

Kahden tietoalkion kokoiset kattavat joukot: 11 kpl
TIEP1, TIEP2 (16)
TIEP1, TIEP3 (20)
TIEP1, TIEP4 (16)
TIEP1, TIEP5 (16)
TIEP1, TIEY4 (16)
TIEP2, TIEP3 (15)
TIEP2, TIEP4 (15)
TIEP3, TIEP4 (15)
TIEP3, TIEP5 (15)
TIEP3, TIEY4 (15)
TIEP4, TIEP5 (14)

Kolmen tietoalkion kokoiset kattavat joukot: 7 kpl

TIEP1, TIEP2, TIEP3 (14)
TIEP1, TIEP2, TIEP4 (14)
TIEP1, TIEP3, TIEP4 (15)
TIEP1, TIEP3, TIEP5 (15)
TIEP1, TIEP3, TIEY4 (14)
TIEP1, TIEP4, TIEP5 (14)
TIEP3, TIEP4, TIEP5 (14)

Neljän tietoalkion kokoiset kattavat joukot: 1 kpl
TIEP1, TIEP3, TIEP4, TIEP5 (14)

Kattavat joukot (minimituki 10% (10 tapausta)):

Yhden tietoalkion kokoiset kattavat joukot: 12 kpl
HTIS67 (11)
HTIS80 (11)
KKENYHT/LUK (15)
TIEP1 (34)
TIEP2 (18)
TIEP3 (23)
TIEP4 (17)
TIEP5 (16)
TIEPT (10)
TIETA16 (15)
TIEY2 (12)
TIEY4 (25)

Kahden tietoalkion kokoiset kattavat joukot: 22 kpl
TIEP1, TIEP2 (16)
TIEP1, TIEP3 (20)
TIEP1, TIEP4 (16)
TIEP1, TIEP5 (16)
TIEP1, TIEPT (10)
TIEP1, TIETA16 (10)
TIEP1, TIEY2 (10)
TIEP1, TIEY4 (16)
TIEP2, TIEP3 (15)
TIEP2, TIEP4 (15)
TIEP2, TIEP5 (13)
TIEP2, TIEPT (10)
TIEP2, TIEY4 (11)
TIEP3, TIEP4 (15)
TIEP3, TIEP5 (15)
TIEP3, TIEPT (10)
TIEP3, TIEY4 (15)
TIEP4, TIEP5 (14)
TIEP4, TIEPT (10)

TIEP4, TIEY4 (13)
TIEP5, TIEPT (10)
TIEP5, TIEY4 (12)

Kolmen tietoalkion kokoiset kattavat joukot: 29 kpl
TIEP1, TIEP2, TIEP3 (14)
TIEP1, TIEP2, TIEP4 (14)
TIEP1, TIEP2, TIEP5 (13)
TIEP1, TIEP2, TIEPT (10)
TIEP1, TIEP2, TIEY4 (11)
TIEP1, TIEP3, TIEP4 (15)
TIEP1, TIEP3, TIEP5 (15)
TIEP1, TIEP3, TIEPT (10)
TIEP1, TIEP3, TIEY4 (14)
TIEP1, TIEP4, TIEP5 (14)
TIEP1, TIEP4, TIEPT (10)
TIEP1, TIEP4, TIEY4 (13)
TIEP1, TIEP5, TIEPT (10)
TIEP1, TIEP5, TIEY4 (12)
TIEP2, TIEP3, TIEP4 (13)
TIEP2, TIEP3, TIEP5 (13)
TIEP2, TIEP3, TIEPT (10)
TIEP2, TIEP3, TIEY4 (10)
TIEP2, TIEP4, TIEP5 (12)
TIEP2, TIEP4, TIEPT (10)
TIEP2, TIEP4, TIEY4 (11)
TIEP2, TIEP5, TIEPT (10)
TIEP3, TIEP4, TIEP5 (14)
TIEP3, TIEP4, TIEPT (10)
TIEP3, TIEP4, TIEY4 (12)
TIEP3, TIEP5, TIEPT (10)
TIEP3, TIEP5, TIEY4 (11)
TIEP4, TIEP5, TIEPT (10)
TIEP4, TIEP5, TIEY4 (11)

Neljän tietoalkion kokoiset kattavat joukot: 22 kpl
TIEP1, TIEP2, TIEP3, TIEP4 (13)
TIEP1, TIEP2, TIEP3, TIEP5 (13)
TIEP1, TIEP2, TIEP3, TIEPT (10)
TIEP1, TIEP2, TIEP3, TIEY4 (10)
TIEP1, TIEP2, TIEP4, TIEP5 (12)
TIEP1, TIEP2, TIEP4, TIEPT (10)
TIEP1, TIEP2, TIEP4, TIEY4 (11)
TIEP1, TIEP2, TIEP5, TIEPT (10)
TIEP1, TIEP3, TIEP4, TIEP5 (14)
TIEP1, TIEP3, TIEP4, TIEPT (10)
TIEP1, TIEP3, TIEP4, TIEY4 (12)

TIEP1, TIEP3, TIEP5, TIEPT (10)
TIEP1, TIEP3, TIEP5, TIEY4 (11)
TIEP1, TIEP4, TIEP5, TIEPT (10)
TIEP1, TIEP4, TIEP5, TIEY4 (11)
TIEP2, TIEP3, TIEP4, TIEP5 (12)
TIEP2, TIEP3, TIEP4, TIEPT (10)
TIEP2, TIEP3, TIEP4, TIEY4 (10)
TIEP2, TIEP3, TIEP5, TIEPT (10)
TIEP2, TIEP4, TIEP5, TIEPT (10)
TIEP3, TIEP4, TIEP5, TIEPT (10)
TIEP3, TIEP4, TIEP5, TIEY4 (11)

Viiden tietoalkion kokoiset kattavat joukot: 8 kpl
TIEP1, TIEP2, TIEP3, TIEP4, TIEP5 (12)
TIEP1, TIEP2, TIEP3, TIEP4, TIEPT (10)
TIEP1, TIEP2, TIEP3, TIEP4, TIEY4 (10)
TIEP1, TIEP2, TIEP3, TIEP5, TIEPT (10)
TIEP1, TIEP2, TIEP4, TIEP5, TIEPT (10)
TIEP1, TIEP3, TIEP4, TIEP5, TIEPT (10)
TIEP1, TIEP3, TIEP4, TIEP5, TIEY4 (11)
TIEP2, TIEP3, TIEP4, TIEP5, TIEPT (10)

Kuuden tietoalkion kokoiset kattavat joukot: 1 kpl
TIEP1, TIEP2, TIEP3, TIEP4, TIEP5, TIEPT (10)

Opiskelijoiden, joilla on hyväksiluettuja opintoja, korvattujen suoritusten lisäksi suoritettujen opintojen kurssihahmot:
Tapauksia: 93
Attribuutteja: 156
Minimituki: 25% (23 tapausta)

Kattavat joukot:

Yhden tietoalkion kokoiset kattavat joukot: 13 kpl
LUOYY003 (58)
MTTMY1 (27)
MTTTP1 (28)
TIEA1 (34)
TIEA2.1A (26)
TIEA2.1B (23)
TIEP1 (34)
TIEP2 (29)
TIEP3 (30)
TIEP4 (28)
TIEP5 (37)
TIETA9 (27)
TIEY2 (38)

Kahden tietoalkion kokoiset kattavat joukot: 20 kpl

LUOYY003, MTTMY1	(25)
LUOYY003, MTTTP1	(26)
LUOYY003, TIEA1	(32)
LUOYY003, TIEA2.1A	(25)
LUOYY003, TIEP2	(28)
LUOYY003, TIEP3	(27)
LUOYY003, TIEP4	(26)
LUOYY003, TIEP5	(34)
LUOYY003, TIETA9	(27)
LUOYY003, TIEY2	(36)
TIEA1, TIEP2	(24)
TIEA1, TIEP5	(23)
TIEA1, TIEY2	(24)
TIEP1, TIEP5	(23)
TIEP2, TIEP3	(23)
TIEP2, TIEP5	(24)
TIEP2, TIEY2	(25)
TIEP3, TIEP5	(25)
TIEP4, TIEP5	(26)
TIEP5, TIEY2	(25)

Kolmen tietoalkion kokoiset kattavat joukot: 8 kpl

LUOYY003, TIEA1, TIEP2	(23)
LUOYY003, TIEA1, TIEP5	(23)
LUOYY003, TIEA1, TIEY2	(24)
LUOYY003, TIEP2, TIEP5	(24)
LUOYY003, TIEP2, TIEY2	(25)
LUOYY003, TIEP3, TIEP5	(23)
LUOYY003, TIEP4, TIEP5	(25)
LUOYY003, TIEP5, TIEY2	(25)

Liite 4: Assosiaatiosäännöt FP-growth:lla kurssien suositteluun

Supp	Conf	Covr	Strg	Lift	Levr	Antecedent		Consequent
0.412	0.984	0.419	1.556	1.511	0.139	TIEP1=1, TIEP3=1, TIEY2=1	→	LUOYY003=1
0.405	0.984	0.412	1.331	1.795	0.179	TIEP3=1, TIEP5=1	→	TIEP1=1
0.429	0.977	0.439	1.485	1.501	0.143	TIEP1=1, TIEY2=1	→	LUOYY003=1
0.425	0.977	0.435	1.496	1.501	0.142	TIEP3=1, TIEY2=1	→	LUOYY003=1
0.415	0.977	0.425	1.531	1.500	0.138	TIEY2=1, TIEY4=1	→	LUOYY003=1
0.412	0.976	0.422	1.543	1.499	0.137	TIEP2=1, TIEP3=1	→	LUOYY003=1
0.405	0.976	0.415	1.568	1.499	0.135	TIEP1=1, TIEP2=1	→	LUOYY003=1
0.449	0.971	0.462	1.410	1.492	0.148	TIEY4=1	→	LUOYY003=1
0.432	0.970	0.445	1.231	1.770	0.188	TIEP5=1	→	TIEP1=1
0.415	0.969	0.429	1.279	1.768	0.180	LUOYY003=1, TIEP5=1	→	TIEP1=1
0.412	0.969	0.425	1.289	1.767	0.179	LUOYY003=1, TIEP3=1, TIEY2=1	→	TIEP1=1
0.429	0.963	0.445	1.463	1.478	0.139	TIEP5=1	→	LUOYY003=1
0.419	0.962	0.435	1.496	1.477	0.135	TIEP2=1, TIEY2=1	→	LUOYY003=1
0.419	0.962	0.435	1.260	1.755	0.180	TIEP3=1, TIEY2=1	→	TIEP1=1
0.415	0.962	0.432	1.508	1.477	0.134	TIEP1=1, TIEP5=1	→	LUOYY003=1
0.412	0.961	0.429	1.140	1.968	0.203	LUOYY003=1, TIEP1=1, TIEY2=1	→	TIEP3=1
0.445	0.957	0.465	1.400	1.470	0.142	TIEP2=1	→	LUOYY003=1
0.442	0.957	0.462	1.410	1.469	0.141	TIEP1=1, TIEP3=1	→	LUOYY003=1
0.442	0.957	0.462	1.187	1.745	0.189	LUOYY003=1, TIEP3=1	→	TIEP1=1
0.419	0.955	0.439	1.114	1.955	0.204	TIEP1=1, TIEY2=1	→	TIEP3=1
0.402	0.953	0.422	1.236	1.827	0.182	TIEP2=1, TIEP3=1	→	TIEY2=1
0.412	0.947	0.435	1.099	1.979	0.204	TIEP3=1, TIEY2=1	→	LUOYY003=1, TIEP1=1
0.462	0.946	0.488	1.333	1.452	0.144	TIEP3=1	→	LUOYY003=1
0.462	0.946	0.488	1.122	1.725	0.194	TIEP3=1	→	TIEP1=1
0.492	0.943	0.522	1.248	1.448	0.152	TIEY2=1	→	LUOYY003=1
0.419	0.940	0.445	1.172	1.803	0.186	LUOYY003=1, TIEP2=1	→	TIEY2=1
0.412	0.939	0.439	1.053	2.034	0.209	TIEP1=1, TIEY2=1	→	LUOYY003=1, TIEP3=1
0.405	0.938	0.432	1.131	1.922	0.194	TIEP1=1, TIEP5=1	→	TIEP3=1
0.435	0.936	0.465	1.121	1.794	0.193	TIEP2=1	→	TIEY2=1
0.415	0.933	0.445	1.075	1.950	0.202	TIEP5=1	→	LUOYY003=1, TIEP1=1
0.412	0.932	0.442	1.180	1.787	0.181	LUOYY003=1, TIEP1=1, TIEP3=1	→	TIEY2=1
0.415	0.926	0.449	1.163	1.775	0.181	LUOYY003=1, TIEY4=1	→	TIEY2=1
0.412	0.925	0.445	1.097	1.895	0.195	LUOYY003=1, TIEP2=1	→	TIEP3=1
0.412	0.925	0.445	1.097	1.895	0.195	TIEP5=1	→	TIEP3=1
0.402	0.924	0.435	1.122	1.891	0.189	TIEP2=1, TIEY2=1	→	TIEP3=1
0.402	0.924	0.435	1.069	1.986	0.200	TIEP3=1, TIEY2=1	→	TIEP2=1
0.442	0.924	0.478	1.021	1.891	0.208	LUOYY003=1, TIEP1=1	→	TIEP3=1
0.425	0.921	0.462	1.129	1.765	0.184	TIEY4=1	→	TIEY2=1
0.425	0.921	0.462	1.129	1.765	0.184	LUOYY003=1, TIEP3=1	→	TIEY2=1
0.405	0.910	0.445	1.231	1.661	0.161	LUOYY003=1, TIEP2=1	→	TIEP1=1
0.405	0.910	0.445	1.037	1.972	0.200	TIEP5=1	→	TIEP1=1, TIEP3=1
0.422	0.907	0.465	1.050	1.857	0.195	TIEP2=1	→	TIEP3=1
0.419	0.906	0.462	1.129	1.738	0.178	TIEP1=1, TIEP3=1	→	TIEY2=1
0.442	0.905	0.488	0.980	1.891	0.208	TIEP3=1	→	LUOYY003=1, TIEP1=1
0.419	0.900	0.465	1.057	1.830	0.190	TIEP2=1	→	LUOYY003=1, TIEY2=1