Annele Ahola

Big data ja Weka API:n käyttö Java-sovelluksessa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Ohjelmistotekniikka

Insinöörityö

26.5.2016

Metropolia

Tekijä(t) Otsikko Sivumäärä Aika	Annele Ahola Big data ja Weka API:n käyttö Java-sovelluksessa 45 sivua + 1 liite 26.5.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Lehtori Vesa Ollikainen

Tässä insinöörityössä esitellään lukijalle big data eli massadata ja sen louhinta sekä opastetaan, kuinka avoimen lähdekoodin tiedonlouhintasovellus Wekan tarjoamaa ohjelmointirajapintaa voidaan käyttää osana omaa koodia Java-sovelluksessa.

Työssä esitellään termi big data kaikessa monipuolisuudessaan. Samalla tutkitaan kuinka laajasti sitä voidaan ja kannattaa elämän eri osa-alueilla hyödyntää, jotta saavutetaan uutta tietoa ihmisistä ja heidän toiminnastaan, laitteista ja ympäristöstä sekä yleisesti syy-seuraussuhteista. Työssä esitellään haaste, jonka big data suurilla tietomäärillään sekä niiden jatkuvalla kasvulla ja monipuolistumisella asettaa yksityisille ihmisille, yrityksille ja yhteiskunnalle, sekä ratkaisut, joilla big data-analyysit sekä tiedonlouhinta vastaavat haasteeseen ja miten tämä käytännössä tapahtuu.

Ohjelmointirajapinnan käyttöönoton esittelyä varten on luotu sovellus, jonka tarkoituksena on numeerisia ennusteita hyödyntäen ennakoida vastaajan ikä hänen vastaustensa perusteella. Lisäksi ohjelma vertailee erilaisia numeerisen ennusteen menetelmiä toisiinsa tuoden esille niiden vahvuutta tai heikkoutta ennustamisessa. Lopuksi esitellään luotu sovellus ja ohjataan lukijaa askel askeleelta, kuinka se on toteutettu sekä rakennettu ja arvioidaan ohjelman toimintaa alkuperäisiä ohjelman toiminnan tavoitteita vasten.

Avainsanat	big data, tiedonlouhinta, weka, api, java

Author(s) Title Number of Pages Date	Annele Ahola Big Data and Weka API in Java-application 45 pages + 1 appendice 26 May 2016
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Specialisation option	Software Engineering
Instructor(s)	Vesa Ollikainen, Senior Lecturer

This thesis presents the reader big data and data mining as well as gives guidance on how to use a programming interface (API) provided by an open source data mining software Weka, on Java-software.

The study introduces the diverse term of big data and how big data could and should be beneficially used in a variety of life areas in order to achieve new knowledge about people and their activities as well as about machines, environment and cause-effect relationships. This thesis presents the challenge that big data with its large data volumes, continuous growth and diversification, sets for individuals, companies and society at large. It also presents solutions for the challenges, i.e. big data analysis and data mining, and how these work in practice.

As a part of this thesis an application was created to demonstrate how to use the Weka API. The purpose of the application is to predict the age of the respondent on the basis of his/hers answers using numerical forecasts, which are a set of specific type of data mining methods, and to compare these methods with each other, bringing out their strengths and weaknesses. Finally the application itself is presented and the reader is guided step by step, how it was created and built. The functionality of the application is being evaluated and compared to the original goals of functionality.

Keywords	big data, data mining, weka, api, java

Sisällys

1 Johdanto

Tämän työn tarkoitus on esitellä lukijalle big data eli massadata ja sen louhinta sekä ohjeistaa, kuinka tiedonlouhintasovellus Wekan ohjelmointirajapintaa voi käyttää osana omaa koodia Java-sovelluksessa. Lisäksi rakennetaan rajapintaa hyödyntävä esimerkkisovellus, jota käytetään numeeriseen ennustamiseen. Saadun tuloksen luotettavuutta arvioidaan esimerkkitapauksessa.

Työ koostuu kolmesta osuudesta. Ensimmäisessä osuudessa (luku 2) esitellään big data terminä ja ilmiönä. Jotta big dataa voidaan tutkia ja esitellä, tulee myös määritellä, miten dataa ylipäätään tutkitaan ja jaotellaan. Koska big data ilmiönä on hyvin laaja, on sen tarjoamiin mahdollisuuksiin valittu esiteltäväksi lukijalle mahdollisesti kiinnostavimmat aiheet ja kohteet. Koska kyseessä on valtaisat määrät tietoa niin ihmisistä kuin ympäristöstäkin, aiheuttaa big data myös uusia haasteita. Näitä haasteita pyritään esittelemään monipuolisesti, mutta niihin haetaan myös ratkaisuja.

Seuraavassa osassa (luku 3) syvennytään tiedonlouhintaan eli käytännön toimenpiteisiin, miten big dataa louhitaan ja miten datan seasta etsitään ja saadaan esille meitä kiinnostavia asioita. Tässä luvussa esitellään tiedonlouhinnan perusajatus sekä –tavoite ja pohditaan tietokoneiden osuutta tiedonlouhinnassa. Koska tiedonlouhintaa varten on luotu useita eri sovelluksia, esitellään tässä kohtaa kaksi yleisintä, mutta toisistaan melko lailla eroavaa tiedonlouhintasovellusta, RapidMiner ja Weka. Myös tiedonlouhintamenetelmiä on useita ja niistä esitellään vain tämän työn käytännön esimerkissä käytetyt menetelmät.

Viimeinen osuus eli neljäs luku keskittyy esittämään, kuinka lukija voi käyttää Wekan tarjoamaa ohjelmointirajapintaa osana omaa Java-sovellustaan. Tätä osuutta varten on ohjelmoitu yksinkertainen Java-ohjelma, joka tekee numeerista ennakointia erilaisilla tiedonlouhintamenetelmillä ja vertailee, kuinka tarkkoja ennusteet ovat. Tiedonlouhintaa tehdään Helsingin Sanomien tarjoamaan vaalikonedataan. Luvussa ohjataan, kuinka lukija voi tehdä täsmälleen samanlaisen ohjelman itse ja näin ymmärtää paremmin sekä tiedonlouhinnan ja big datan hyödyt että Wekan käyttöä.

2 Big data

Koska työn tarkoitus on perehtyä ja tutustuttaa lukija esimerkkien kautta big dataan ilmiönä, on tarpeen selittää, mitä kaikkea big datalla tarkoitetaan.

Big datalla eli massadatalla eli suuraineistolla viitataan Salon mukaan nopeasti kasvavaan ja monipuolistuvaan dataan, sen luomaan haasteeseen organisaatioissa ja yhteiskunnassa, sekä näihin haasteisiin tarjolla oleviin ratkaisuihin. Big data on siis laaja käsite, jonka eri merkityksiä on käsiteltävä ja tutkittava monipuolisesti yhdessä ja erikseen saadakseen kokonaisen kuvan Big datan merkityksestä yrityksille ja yhteiskunnalle. (Salo 2013, 10.)

Big data -ilmiössä haetaan vastausta muun muassa kysymyksiin: Miten siirtää ja tallentaa dataa? Miten yhdistää ja analysoida dataa? Sekä miten tehokkaasti hyödyntää kaikkea käsillä olevaa dataa? (Salo 2013, 21.)

2.1 Tiedonmäärän kasvu

Kuten Salo toteaa, tämän päivän teknologia mahdollistaa yhä tehokkaamman datan jatkuvan luomisen ja sen pysyvämmän tallentamisen. Suuruusluokka, jossa liikutaan datan määrän suhteen, on datan määrän monikymmenkertaistuminen seuraavan kymmenen vuoden aikana. (Salo 2013, 11.)

Salon mukaan tässä on kysymys datan määrän kasvupaineen kerääntymisestä sietämättömän suureksi, joka yhdessä teknologian kehityksen kanssa on aiheuttanut tarpeen, johon big data vastaa. (Salo 2013, 16.) Tämän takia termi big data on paitsi tarpeellinen, myös kuvaava, kuten Davenport toteaa sanoessaan, että yhden arvion mukaan uutta dataa syntyy maailmassa 2,5 triljoonaa tavua päivässä (Davenport 2014, 11).

Lisää perspektiiviä datan määrään maailmassa ja sen jatkuvaan kasvuun antaa Price verkkodokumentissaan kertoessaan, että esimerkiksi nykyään keskiverto kotitalous tuottaa vuodessa dataa niin paljon, että se täyttäisi 65 iPhonea (yhdessä tilaa 32 GB) ja vuoteen 2020 mennessä määrä kasvaa 318 iPhoneen. (Price 2014.)

On todettu myös, että tiedon määrä kaikissa maailman tietokannoissa tuplaantuu joka kahdeskymmenes kuukausi. (Frank, Hall & Witten 2011, 4.)

Datan huimista määristä huolimatta, tai ehkä juuri sen takia iso osa siitä jää hyödyntämättä ja suuri osa siitä johdettavasta informaatiosta tallentamatta siitäkin huolimatta, että Salon mukaan kaikella datalla on alun perin joku sovellusalue ja käyttötarkoitus. (Salo 2013, 21.)

Työn loppupuolella esimerkissä käytetty aineisto, joka on koottu Helsingin Sanomien vaalikoneen vastauksista, on toki laaja, sisältäessään 1764 tietuetta ja 40 attribuuttia, mutta ei yleisen ajattelun mukaan täytä big datan määritelmää. Big datasta puhuttaessa tarkoitetaan yleensä sadoistatuhansista tai miljoonista tietueista koostuvaa aineistoa, mutta vähintään useista tuhansista. Kuitenkin esimerkissä käytetty aineisto on tarpeeksi laaja, jotta sitä voidaan käsitellä kuin big dataa ja tuoda sitä kautta analyysin hyödyt esille.

2.2 Strukturoitu, strukturoimaton ja semistrukturoitu data

Termi strukturoitu data viittaa yleisesti dataan, jolla on määrätty pituus ja muoto. Esimerkkejä strukturoidusta datasta ovat esimerkiksi numerot, päivämäärät ja Stringit eli esimerkiksi henkilön nimi tai osoite, joka voi sisältää sekä numeroita että kirjaimia. Strukturoitu data on yleensä tallennettu tietokantaan, johon voi suorittaa kyselyitä käyttäen SQL:ää (structured query language). (Halper & co. 2013, 50.)

Strukturoimaton data ei noudata tiettyä muotoilua. Jos 20 prosenttia yrityksille saatavilla olevasta datasta on strukturoitua dataa, niin loput 80 prosenttia on strukturoimatonta. Tähän asti teknologia ei ole tukenut datan hyödyntämistä muuten kuin tallentamalla sitä ja analysoimalla sitä manuaalisesti. (Halper & co. 2013, 53.)

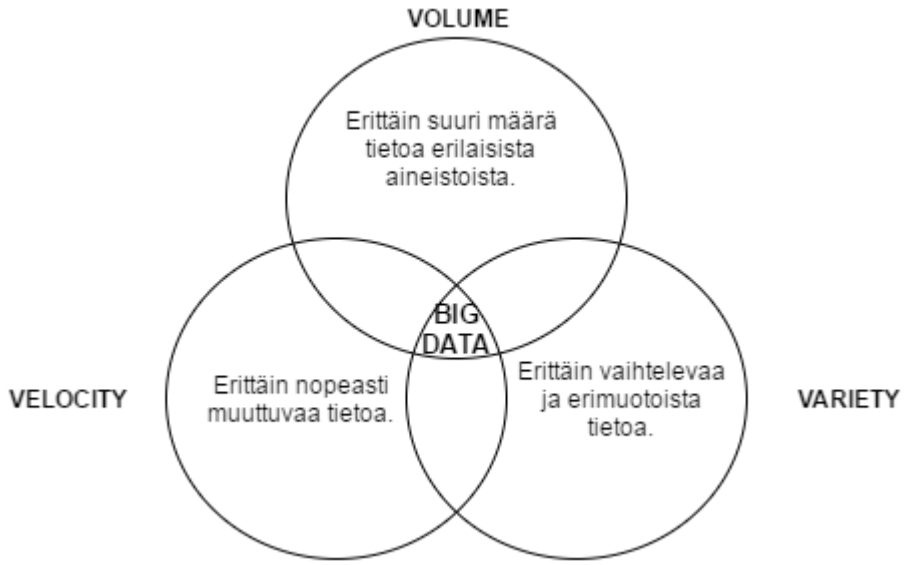
Esimerkiksi Helsingin vaalikoneen vastauksista kerätty aineisto on strukturoitua sisältäessään suurimmaksi osaksi numeroita, sekä jonkin verran string-tyyppistä dataa. Strukturoitua dataa on helpompi käsitellä ja analysoida nykyisillä big datan työkaluilla kuin strukturoimatonta dataa.

2.3 Big Datan 3V-käsitemalli

Halper ja muut (2013, 34.) määrittelevät big datan olevan mikä tahansa tiedon lähde, jolla on ainakin seuraavat kolme ominaisuutta.

1. erittäin suuri määrä tietoa (Volume)
2. erittäin nopeasti muuttuvaa tietoa (Velocity)
3. erittäin vaihtelevaa ja erilaista tietoa (Variety).

Nämä kolme ominaisuutta muodostavat kolme V:tä kuvan 1 osoittamalla tavalla.



Kuva 1. Big datan 3V-käsitemalli

Vaikka onkin kätevää yksinkertaistaa big datan käsite kolmeen V:hen, se voi olla myös harhaanjohtavaa ja liian yksinkertaistettua. Esimerkiksi saatat hallita suhteellisen pientä määrää hyvin erilaisia ja monimutkaisia tietoja tai saatatetaan käsitellä valtavaa määrää hyvin yksinkertaisia tietoja. Nuo yksinkertaiset tiedot voivat olla kaikki strukturoitua tai strukturoimatonta dataa. Tässä tilanteessa neljäs V: veracity, totuudenmukaisuus on vieläkin tärkeämpää. Kuinka tarkkaa tieto on? Ovatko big data analyysin tulokset oikeasti järkeviä? (Halper ym. 2013, 40.)

2.4 Mahdollisuudet

Sanonnan mukaan tieto merkitsee valtaa. Nykymaailmassa kuitenkin lähes kaikki tieto alkaa olla erittäin helposti saatavilla ja kaikille tasapuolisesti tarjolla. Niinpä tieto, jota muilla ei ole ja jota muut eivät osaa käyttää ja hyödyntää, on se uusi valttikortti.

Salon näkemyksen mukaan pilvipalvelu- ja big data -ajan menestyvän yrityksen tunnistaa siitä, että se onnistuu keräämään eniten dataa, yhdistämään sen muualta saadun datan kanssa ja analysoimalla kokonaisuutta tuottamaan siitä hyödyksi muutettavaa lisäarvoa. (Salo 2013, 142.)

Mitään lopullista ratkaisua ei big data kilpailuedun ongelmaan tuo, mutta datasta tulee yksi tärkeimmistä ja jopa ainoista kestävän kilpailuedun lähteistä. Data on arvokasta vain, jos sitä osaa hyödyntää, mutta hyödynnettynä se sitten tarjoaakin etumatkan, jota on vaikea kuroa kiinni. (Salo 2013, 138.)

Tärkeintä on analysoida data ja muuntaa se näkemyksiksi, innovaatioiksi sekä taloudelliseksi hyödyksi sen sijaan, että arvostetaan vain datan suurta määrää. (Davenport 2014, 2.)

2.4.1 Lääketiede sekä terveydenhuolto ja hyvinvointi

Terveydenhoito on yksi tietointensiivisimmistä teollisuudenaloista tällä hetkellä. Periaatteessa neljä päälähdettä tuottaa kaiken terveydenhuollon datan: Terveydenhoidon tarjoajat, julkiset ja yksityiset maksajat, lisäpalveluiden tarjoajat (farmaseuteista laboratorioihin) ja terveydenhoidon käyttäjät/kuluttajat. Haasteena ei ole pelkästään tietokannat ja niihin pääsy vaan myös tiedon tekeminen käytettäväksi ja hyödylliseksi. (Knight 2015a.)

Tulosten parantamisen ja kustannusten leikkaamisen lisäksi Big Dataa käytetään terveydenhuollossa epidemioiden ennakoimiseen, tautien parantamiseen, elämänlaadun parantamiseen ja ennaltaehkäistävien sekä ennakoitavien kuolemien estämiseen. (Marr 2015a.)

Viime vuonna IDC Health Insightsin tekemä tutkimus osoitti, että 50 prosenttia sairaaloista ja terveydenhuollon vakuutusfirmoista asettivat analyysivalmiuksien kasvattamisen tärkeimmäksi prioriteetikseen ensi vuoden aikana. (Marr 2015b.)

McKinsey & Company kokosi raportin Center for US Health System Reformille, jossa tunnistettiin neljä big datan päälähdettä terveydenhuollon alalla. Näitä ovat: toiminta- ja kustannusdata, kliininen data, lääketestauksista kerätty data sekä potilaiden arkielämästä kerätty data.

Toiminta- ja kustannusdata koostuu perusluvuista, jotka osoittavat muun muassa, kuinka paljon tiettyjä sairauksia on hoidettu ja millä kustannuksilla johtaen tietoon kaikista kustannustehokkaimmista hoitomenetelmistä. Data antaa tietoa myös tiettyjen sairauksien levinneisyydestä ja millä tärkeydellä niitä tulisi hoitaa.

Kliininen data koostuu kunkin potilaan terveydellisistä tiedoista, jotka on kerätty lääkärikäyntien ja -tutkimusten aikana, mikä sisältää myös esimerkiksi lääkärin muistiinpanoja potilaasta.

Lääketestauksista kerätty data koostuu lääkkeille suoritettujen testien tuloksista ja testien koehenkilöistä. Tällaisten tietokantojen avulla voidaan löytää yhä sopivampia ehdokkaita lääketestaukseen ja siten parantaa testien tuloksia.

Potilaiden arkielämästä kerätty data koostuu esimerkiksi sykemittarilla kerätyistä arvoista, jotka kuvaavat potilaan arkielämän toimia ja kehon toimintaa normaalitilanteissa. Muita mitattavia arvoja ovat esimerkiksi sydänkäyrä, ruumiinlämpö ja sykevälivaihtelut. (Marr 2015c.)

Konsulttiryhmä McKinsey & Company arvioi, että Big Datan käyttö tulee säästämään 12-17 prosenttia USA:n terveydenhuolto kustannuksissa. Palveluidentarjoajat, jotka käyttävät parhaita käytäntöjä, pystyvät välittömästi tehostamaan/lisäämään prosessien määrää, kuten koodaamista, laskuttamista ja tarjonnan hallinnan käytäntöjä? Tärkeimpänä, Big Data tarjoaa niin pienille kuin suurillekin terveydenhuollon tarjoajille mahdollisuuden analysoida hoitotuloksia ja osoittaa, mihin rahaa kannattaa käyttää ja missä sitä kannattaa säästää. (Knight 2015b.)

Knight (2015a.) esittelee verkkodokumentissaan kuusi tapaa, joilla Big Data muuttaa terveydenhuoltoa:

- Tukee tutkimuksia: Tieto-lähtöiset ratkaisut voivat johtaa parempiin diagnosointimenetelmiin sekä parannuskeinoihin, mikä kehittää sekä parantaa lopputulosta ja leikkaan koko prosessin kustannuksia.

- Datan muutos tiedoksi: Muutos parantaa tiedon kulkua sekä laatua, ja tästä seuraa tehokkaampaa hoitoa.

- Tukee itsehoitoa: Useat yritykset tutkivat Big Dataa ja kuinka sitä voi käyttää ”auttamaan meitä auttamaan itseämme.” Näköpiirissä/tiedossa on parempi terveydellinen tieto saavutettavissa yhteisöllisellä tasolla.

- Tukee tarjoajia: Palveluntarjoajien vastustuksesta huolimatta yritykset kehittävät uusia käyttäjäystävällisempiä tuki järjestelmiä, lievittääkseen/helpottaakseen/vähentääkseen paineita, säästämällä aikaa ja rahaa sekä parantamalla lopputulosta.

- Kasvattaa tietoisuutta: Uudet teknologiat ja niitä tarjoavat yritykset ottavat listan edelliset pointit hoitaakseen globaalien ja julkisten terveydellisten ongelmien ehkäisyn ja sen priorisoinnin kannalta.

- Kerää tietoa (varanto): Big Data ei ehkä paranna syöpää tänään, mutta se voi paljastaa tietoja, jotka auttavat parantamaan hoitomenetelmiä huomenna.

(Knight 2015a.)

Tuttu, moderni esimerkki suuraineiston, big datan, avulla rakennetusta paremmasta yhteiskunnasta on Google Flu -järjestelmä. Järjestelmä ennustaa influenssan puhkeamisen laskemalla, kuinka monta hakua sanalla ”flu” on tehty kussakin Yhdysvaltain osavaltiossa tai alueella. Kun näiden verkkohakujen lukumäärä kasvaa alueella voimakkaasti, siellä olevien tautitapausten määrä on todennäköisesti kasvussa. Tällaiset tekniikat auttavat Yhdysvaltain Centers for Disease Contolia (tarttuvien tautien tutkimuskeskus) havaitsemaan uudet influenssakannat ja ennustamaan, miten paljon lääkettä tarvitaan, ja ennakoimaan sairastuvien määrän. (Pentland 2014, 130.)

Koska hengityselinten oireiden, kuumeen, influenssan, stressin ja masennuksen kaltaisiin ongelmiin liittyvät käyttäytymisen muutokset ovat samankaltaiset kaikilla, ihmisen

kattava terveydentila voidaan luokitella pelkästään heidän käyttäytymisensä perusteella. Joukkoistamalla käyttäytymisinformaatio koko väestöön ja sen jälkeen yhdistämällä saatu informaatio aineistoon, joka kertoo, missä ja milloin ihmiset ovat liikkuneet edellisten päivien mittaan koko alueen sairastumisriski voidaan arvioida ja todeta, missä ihmisillä on suurin ja pienin todennäköisyys sairastua influenssaan tiettynä päivänä ja kellonaikana. (Pentland 2014, 131.)

Toinen esimerkki big datan hyödyntämisestä terveydenhuollossa on The Pittsburgh Health Data Alliance, joka edustaa yhteistyötä Carnegie Mellon Universityn, The University of Pittsburghin ja UPMC:n välillä.

The University of Pittsburgh tuo mukaan asiantuntemuksen lääketieteellisessä tutkimuksessa ja Carnegie Mellon University tietojenkäsittelytieteessä sekä koneoppimisessa. UPMC tuo datan, hoitoympäristön ja kirjatut tiedot onnistuneesta kaupallistamisesta.

Ratkaisut, joita liitto tarjoaa keskittyvät sairauksien puhkeamisen ehkäisyyn, diagnoosin kehittämiseen ja hoidon laadun parantamiseen. Näillä eväillä voidaan päätyä jopa tilanteeseen, jossa terveydenhoidon rutiinitarkastuksista kerätyn datan avulla voitaisiin ennakoida, milloin henkilö päätyy teho-osastolle. Tulevaisuudessa voidaan mahdollisesti käyttävää geneettistä dataa kehittämään personalisoituja syövän hoito-ohjelmia. (Pittsburgh Health Data Alliance, 2016.)

Myös Apple ja IBM ovat julkistaneet yksityiskohtia ja tietoja heidän kumppanuudestaan, joka tarjoaa iPhonen sekä Apple Watch -älykellon käyttäjille mahdollisuuden jakaa tietojaan IBM:n Watson Health pilvipohjaiseen terveydenhuollon analysointi palveluun. Marr kuvailee tätä hanketta verkkodokumentissaan (2015b).

Watson Health käyttää toimiakseen IBM:n Watson Analyticsia, joka taas käyttää hienostuneita koneoppimisen algoritmeja, jotka perustuvat luonnollisen kielen käsittelyyn. Teoriassa käyttäjät voivat pyytää sitä suorittamaan monimutkaisia analyyttisia operaatioita käyttäen yksinkertaista luonnollista kieltä, ilman tarvetta osata ja ymmärtää koodaamista kommunikoidakseen tietokoneen kanssa. Se suunniteltiin, onnistuneesti, alun perin olemaan mestari Jeopardy!-kilpailussa.

Uusi järjestely, uuden kumppanuuden hedelmiä, tarkoittaa, että iPhonelle, iPadille sekä Apple Watchille kehitetyt sovellukset, jotka käyttävät Applen Healthkit-kehityssovelluksia, voivat suoraan ladata tietoja käyttäjiensä terveydestä ja aktiivisuudesta IBM:n turvattuun pilvipalveluun. Pääsy palveluun on rajoitettu yksityisyysasetusten mukaisesti.

Data kerätään eri laitteiden sensorien kautta, kuten GPS:n osoittama liikehdintä älypuhelimissa ja sydänkäyrä ja sykkeenmittaus Apple Watchissa, sekä manuaalisesti kirjaamalla.

Älylaitteita, jotka on linkitetty Applen ja IBM:n tarjoaman palvelun kaltaiseen analyyttiseen sovellukseen, voidaan käyttää myös kaukomonitorointiin mahdollistaen lääkärien puuttumisen tilanteeseen, jos hälyttävää tietoa on kerätty.

IBM arvioi, että lähes 5 miljoonaa ihmistä ympäri maailmaa on liittynyt heidän tarjoamaan terveydenhuollon palveluun ensi vuoteen mennessä.

2.4.2 Rikollisuus

Yksi mielenkiintoisimpia big datan hyödyntämisen osa-alueita on mahdollinen rikosten ennustaminen ja siten niiden ennaltaehkäiseminen.

Van Rijmenam kertoo verkkodokumentissaan, että tällä hetkellä Los Angelesin ja Santa Cruzin poliisilaitokset käyttävät big dataan ja tiedonlouhintaan perustuvia menetelmiä, joilla ennustetaan, missä rikos todennäköisesti tapahtuu. Menetelmän käyttö on ollut menestys, sillä alueella tapahtuneiden murtojen määrä on laskenut 33 prosenttia, väkivaltarikosten määrä 21 prosenttia ja omaisuusrikosten määrä 12 prosenttia.

Koska rikoksista kerätty data osoittaa tiettyä kaavamaisuutta, mallille pystyttiin syöttämään 13 miljoonaa rikosta viimeiseltä 80 vuodelta. Tämä tiedon aarreaitta auttaa nyt ymmärtämään rikosten luonnetta paremmin. Huomattiin, että rikoksen sattuessa tietyllä alueella oli myös todennäköistä, että lisää rikoksia tapahtui kyseisellä alueella.

Poliisit määrätään tietynkokoiselle maantieteelliselle alueelle, jossa rikoksia mallin algoritmin mukaan todennäköisimmin tapahtuu heidän 12 tunnin työvuoronsa aikana. Vuoronsa aikana poliisit ovat ohjeistettuja partioimaan näitä alueita niin usein kuin mahdollista sekä tarkkailemaan ja etsimään mahdollista rikollista tai siihen viittaavaa toimintaa.

Nykyään mallia päivitetään jatkuvasti uusilla rikoksista saaduilla tiedoilla, jotta ennusteista saataisiin vieläkin tarkempia. (van Rijmenam, 2014.)

Kiinnostavaa on myös, että esimerkiksi turvallisuuspalvelut voisivat tunnistaa ja napata terroristit ennen kuin he edes hyökkäävät. Yksi keino olisi antaa valtion virastojen käyttöön kaikki mahdollinen metadata etukäteen, jotta ne voisivat ”kylvää” tietokantaan tunnisteita, kuten puhelinnumeroita. Tällaisella taktiikalla olisi mahdollisuus paljastaa piilokaavoja, jotka voisivat auttaa analyytikoita yhdistämään muita tietolähteitä ja päätellä, onko hyökkäys tapahtumassa. Big datan analyysit voisivat mahdollistaa myös epäiltyjen terroristien tunnistamisen, kun heidän puhelimensa on tunnistettu. (Richards & King 2014a, 407.)

2.4.3 Yritysmaailma

Kuitenkin suurin big datan käyttöä ja kehittämistä eteenpäin vievä voima on raha ja kilpailu. Organisaatiot ovat aina luottaneet kykyynsä luoda raportteja, jotka auttavat ymmärtämään, mitä kerätty data kertoo niille esimerkiksi kuukausittaisista myyntiluvuista tai tulevaisuuden kasvunäkymistä. Big data muuttaa tiedon hallintaa ja käyttötarkoituksia. Jos yritys voi kerätä, hallita ja analysoida riittävästi tietoa, se voi käyttää aivan uudenlaisia työkaluja, jotka auttavat yrityksen johtoa todella ymmärtämään tiedonkeruun ja siitä johtuvan ongelmanratkaisun merkityksen. (Halper ym. 2013, 23.)

Crivat & muut (2009, 4-5.) ovat listanneet liike- ja yritysmaailman ongelmia, joihin big data ja tiedonlouhinta saattavat tuoda vastauksen:

- Suositukset asiakkaille – Mitä tai minkälaisia tuotteita tai palveluita voisit suositella asiakkaillesi? Asiakkaat, joille tarjotaan ajoittain heille sopivia suosituksia, tulevat olemaan yritykselle todennäköisemmin arvokkaita, sillä he ostavat enemmän ja lojaalimpia, sillä he tuntevat voimakkaamman yhteyden tuotteen- tai palveluntarjoajaan.

- Poikkeuksien tunnistaminen – Tiedonlouhinnalla voidaan selvittää, mitkä palaset datasta poikkeavat muista. Luottokorttiyritykset käyttävät tiedonlouhintaan perustuvaa poikkeusten tunnistamista selvittääkseen, ovatko kaikki korttitapahtumat valideja. Jos tiedonlouhintajärjestelmä merkitsee tietyn korttitapahtuman poikkeukselliseksi, voi asiakas saada puhelun, jossa tarkistetaan, onko hän todella

itse käyttänyt korttiaan. Myös vakuutusfirmat käyttävät poikkeusten tunnistamista selvittääkseen, onko jokin vakuutusilmoitus kenties vilpillinen eli vakuutuspetos.

- Liikkuvuus analyysi – Mitkä asiakkaat todennäköisimmin vaihtavat käyttämään kilpailija palveluita tai tuotteita? Analyysi auttaa markkinointipäälliköitä tunnistamaan asiakkaat, jotka todennäköisimmin jättävät yrityksen palvelut ja miksi. Analyysin seurauksena yritys voi parantaa asiakassuhteitaan ja jopa palauttaa entisiä asiakkaita palveluidensa piiriin.

- Riskien hallinta – Voiko tietylle asiakkaalle myöntää lainan? Tiedonlouhinnalla selvitetään lainahakemuksen riskit.

- Asiakassegmentointi – Mitä itse ajattelet asiakkaistasi? Asiakassegmentointi määrittää asiakkaiden käyttäytymisen sekä tunnistamisen profiilit. Näitä profiileita käytetään tarjotakseen yksilöllisiä markkinointiohjelmia sekä -strategioita, jotka kohdistetaan näihin profiilijoukkoihin.

- Kohdistettu mainonta – Käyttämällä verkkosivujen navigointidataa ja asiakkaiden ostoskäyttäytymisessä huomattuja kaavoja, voidaan verkkosivujen selaajille tarjota kohdistettuja mainoksia.

- Ennusteet – Kuinka monta laatikkoa viiniä tullaan tietyssä myymälässä myymään ensi viikolla? Miltä näyttää varastotilanne kuukauden kuluttua? Tiedonlouhintaan perustuvilla ennusteilla voidaan vastata tämän kaltaisiin aikariippuvaisiin kysymyksiin.

Markkinatalous näkee big datan puhtaana mahdollisuutena: markkinoijat käyttävät sitä kohdennettuun mainontaan, vakuutusfirmat käyttävät sitä optimoimaan tarjontaansa ja pankit käyttävät sitä markkinoiden lukemiseen. (Boyd & Crawford 2012, 664.)

2.5 Haasteet

2.5.1 Yksityisyys ja eettisyys

Dataan ja sen säilyttämiseen liittyy myös olennaisia tietoturvahuolia. Usein big datassa on kyse ihmisiin ja heidän elämäänsä sekä toimiinsa liittyvistä tiedoista. Käyttäjät ja ihmiset ylipäätään haluavat, että heistä kerätty tieto on varmassa tallessa siten, etteivät siihen pääse käsiksi palveluntarjoajan oma henkilöstö tai ulkopuoliset tahot. (Salo 2013, 107.)

Mikään tieto ei ole kuitenkaan yhtä henkilökohtaista ja yksityistä, kuin esimerkiksi lääketieteellinen data, joten erityisen vahva tietoturva on tarpeen, jottei tämä tieto pääse vääriin käsiin. (Marr 2015a.)

Yksi este tiedonhankinnassa ja säilyttämisessä on, että ihmiset pelkäävät heidän terveydellisten tietojensa päätyvän vakuutusfirmojen käsiin ja vaikuttavan negatiivisesti heidän mahdolliseen vakuutuksensaantiin esimerkiksi epäterveellisten elintapojen seurauksena. Yksityisyydensuojan tulisi estää tietojen joutuminen vääriin käsiin eli kenellekään sopimuksen ulkopuolella olevalle. Kuitenkin julkinen anonyymi/nimetön data olisi hyödyllistä myös vakuutusfirmoille auttamaan yleisten trendien ja riskitekijöiden määrittämisessä ja tunnistamisessa sekä vähentämään vakuutuspetoksien syntyä. Tämä voisi taas teoriassa johtaa kaikille reilumpiin vakuutusmaksuihin. (Marr 2015b.)

Yrityksen tai yksityisen henkilön, joka on huolissaan datansa turvallisuudesta tietosuojamielessä, kannattaa ennen tietojen pilveen siirtämistä tai niiden siellä säilyttämistä arvioida, kuinka esimerkiksi yrityksen kannalta liiketoimintakriittisiä ne ovat ja mitä seuraisi, jos ne päätyisivät vääriin käsiin. (Salo 2013, 108.)

Kuluneen vuosikymmenen aikana tutkijat ja toimittajat laajalti eri aloilta ovat vakuuttuneet siitä, että digitaalisen tulevaisuutemme on sisällettävä merkittävä suojaus suuresti välittämillemme arvoille, kuten yksityisyys, tasa-arvoisuus sekä identiteetti. (Richards & King 2014b, 2.)

Ensinnäkin meidän on ajateltava yksityisyyttä informaatio sääntöinä. Big datan aikakaudella yksityisyys tulisi ymmärtää paremminkin tarpeena laajentaa sääntöjä, joita käytämme hallitsemaan henkilökohtaisen datan virtaa. Tallennettavien tai tallennettujen

henkilökohtaisten tietojen määrä on selkeästi kasvussa, joten myös tarve säännöille, joilla säännellä tätä sosiaalista muutosta, kasvaa. (Richards & King 2014a, 395.)

Laajamittainen hakudata saattaa auttaa meitä kehittämään parempia välineitä, palveluita sekä julkisia hyödykkeitä. Toisaalta se saattaa johtaa hyökkäyksiin yksityisyyttä kohtaan sekä päällekäyvän, häiritsevän markkinoinnin uuteen aaltoon. Tiedon analysointi saattaa auttaa meitä ymmärtämään verkkoyhteisöjä ja poliittisia liikkeitä. Toisaalta se saattaa johtaa mielenosoittajien jäljitykseen ja siten keskustelun ja sananvapauden hiljenemiseen. Muuttavatko suuret määrät dataa tapaamme tutkia ihmisten keskinäistä viestintää ja kulttuuria vai vähentääkö se tutkimuksen vaihtoehtoja ja muuttaa tutkimuksen tarkoitusta? (Boyd & Crawford 2012, 663.)

Big data nähdään tehokkaana välineenä käsitellä erilaisia yhteiskunnallisia ongelmia tarjoten mahdollisuuden uusiin näkökulmiin koskien niinkin moninaisia aiheita kuin syöpätutkimus, terrorismi ja ilmastonmuutos. Toisaalta big data nähdään huolestuttavana Big Brother -ilmiönä, joka mahdollistaa tunkeutumisen yksityisyyteen, kutistuvat kansalaisoikeudet ja -vapaudet sekä lisääntyvän valvonnan valtion ja yritysten puolesta. Big datan vaikutus ihmisen yksityisyyteen herättää paljon kysymyksiä, joihin vasta mietitään vastuksia. Onko oikein, että jonkun julkinen nettikirjoitus erotetaan asiayhteydestään ja analysoidaan sen jälkeen tavalla, jota kirjoittaja ei olisi voinut ikinä kuvitella? Saako yksityisen ihmisen asettaa analysoitavaksi, tämän tietämättä asiasta? Kuka on vastuussa siitä, etteivät yksilöt ja yhteisöt loukkaannu tutkimusprosessin seurauksena? Miten määritellään tietoinen suostumus tässä asiassa? (Boyd & Crawford 2012, 672.)

Big datan aikakaudella kaikista innovatiivisimpia toissijaisia käyttötarkoituksia datalle ei ole vielä osattu edes kuvitella siinä vaiheessa, kun data on ensikerran kerätty. Kuinka yritykset voisivat ilmoittaa tarkoitusperistään datan suhteen, jos niitä ei ole vielä keksitty. Kuinka yksilöt voivat antaa tietoisen suostumuksen tuntemattomaan asiaan? Suostumuksen uupuessa, mikä tahansa big datan uusi analyysi liittyen henkilökohtaisiin tietoihin vaatisi uutta suostumuksen anomista kultakin tutkimuksen kohteelta, jokaiselle datan uudelleen käytölle erikseen. Kallista. (Cukier & Mayer-Schönberger 2013, 153.)

Myös Chassell (2014) on pohtinut big datan aiheuttamia kysymyksiä yksityisyydestä ja järjestöjen toimintojen vaikutuksista:

- Konteksti, asiayhteys, tarkoitus – Mihin (käyttö)tarkoitukseen data on alun perin luovutettu? Mihin tarkoitukseen dataa nyt käytetään? Kuinka kaukana alkuperäisestä tarkoituksesta uusi käyttötarkoitus on? Onko tämä sopivaa?

- Suostumus ja vaihtoehto/valinta – Minkälaisia vaihtoehtoja asianosaiselle annetaan/on annettu? Tietävätkö asianosaiset, tekevänsä päätöstä? Ymmärtävätkö asianosaiset todella, mihin ovat suostumassa? Onko asianosaisella todella mahdollisuus kieltäytyä? Mitä vaihtoehtoja asianosaiselle tarjotaan?

- Kohtuullisuus – Onko käytetyn datan syvyys ja leveys sekä siitä johdetut suhteet kohtuullisia sovellukselle, johon niitä käytetään?

- Perusteltavuus – Ovatko käytetyt tietolähteet asianmukaisia, arvovaltaisia/virallisia, kokonaisia/täydellisiä sekä sopivia sovellukselle?

- Omistajuus – Kuka omistaa lopputulokset? Mitkä ovat heidän vastuunsa lopputulosten suojelun ja niihin liittyvien velvollisuuksien suhteen?

- Reiluus/oikeudenmukaisuus – Kuinka tasapuolisia menetelmän tulokset ovat kaikille osapuolille? Saavatko kaikki asianmukaiset korvaukset?

- Harkittu – Mitkä ovat datan keruun ja analysoinnin mahdolliset seuraukset?

- Pääsy tietoihin – Minkälainen pääsy/oikeus tutkimuksen kohteelle annetaan tietoihin?

- Vastuullisuus – Kuinka virheitä ja tahattomia seurauksia havaitaan ja korjataan? Voivatko asianomaiset tarkistaa tuloksia, jotka koskevat heitä tai vaikuttavat heihin?

Vaarana voi myös olla, että yksilöitä arvioidaan todennäköisyyksien kautta: algoritmit ennustavat yksilön todennäköisyyden saada sydänkohtaus, josta seuraa, että henkilö joutuu maksamaan enemmän sairausvakuutuksestaan. Kun lasketaan henkilön todennäköisyys laiminlyödä asuntolainansa takaisinmaksu, seuraa siitä, että yksilöltä mahdollisesti evätään lainansaanti. algoritmien laskiessa henkilön todennäköisyyden tehdä ri-

kos, seuraa siitä, että yksilö saatetaan pidättää ennen rikoksen toteuttamista. Tämä johtaa eettiseen tutkiskeluun vapaan tahdon roolista verrattuna tietojenhallinnan diktatuuriin. Pitäisikö yksilön vapaan tahdon tärkeys mennä big datan tärkeyden edelle, vaikka tilastot väittävät toisin? Big datan aikakausi edellyttää uusia sääntöjä yksilöllisyyden turvaamiseksi. (Cukier & Mayer-Schönberger 2013, 16-17.)

Tärkeintä ei ole kuitenkaan se, lisääkö big data yksityisyyden menettämisen uhkaa (kyllä lisää), vaan muuttaako se uhkan luonnetta. Jos uhka olisi yksinkertaisesti sama, mutta suurempi, lait ja säännöt, jotka suojaavat yksityisyyttä voisivat edelleen toimia big datan aikakaudella; nykyiset toimet tulisi vain kaksinkertaistaa. Jos ongelman luonne muuttuu, saatamme tarvita uusia ratkaisuja. (Cukier & Mayer-Schönberger 2013, 153.)

Vaikka data olisi karsittu mahdollisimman anonyymiksi, big data yleensä sisältää niin suuren määrän tietoa, että jopa yksittäiset ihmiset voidaan silti tunnistaa yhdistämällä tuloksia toisiinsa tai muualta saataviin tietoihin.

2.5.2 Ammattitaidon puute

Big datan käsittely eli tässä työssä viitattu louhinta vaatii tekijältään ammattitaitoa useilta eri aloilta. Se vaatii teknistä kokemusta erilaisten sovellusten parissa työskentelystä, kuin myös yleistä taitoa ja ymmärrystä tietojen analyysista, tilastotieteistä ja matemaattisesta mallinnuksesta. Lisäksi, että tuloksia voidaan tulkita oikein, tekijä tarvitsee kattavan käsityksen kyseisen alan, oli sitten kyseessä terveydenhoito, markkinointi tai viihdemaailma, ympäristötekijöistä ja vaikuttimista. Dataa pitää ymmärtää todella läheisellä tasolla saadakseen siitä merkittäviä tuloksia irti tiedonlouhinnan avulla. Tällaista poikkitieteellistä ammattitaitoa ei välttämättä helpolla löydy tai se on hyvin kallista. Siksi saatetaankin tyytyä yhdistelemään eri tekijöiden ammattitaitoa, jolloin ongelmaksi muodostuu ihmisten kyky kommunikoida toistensa kanssa ja jakaa osaamistaan sekä taitojaan ja tietoaan muille. Liekö sitten parempi vaihtoehto olla data-analyysin rautainen ammattilainen ja opiskella siihen päälle kunkin tutkittavan datan olennaisia ominaisuuksia vai olla tietyn alan ammattilainen, joka tuntee keräämänsä datan perinpohjaisesti ja opiskella sitten datan analysointia ja tiedonlouhintaa? Joka tapauksessa nämä kaksi taitoa on jollain tavalla yhdistettävä big datan louhinnassa, ja se voi tuottaa ongelmia.

3 Tiedonlouhinta

Kuten aiemmin on todettu, joka kerta käyttäessämme ostoksia tehdessämme kaupan kanta-asiakaskorttia tai luottokorttia, tai surffailemme netissä, luomme uutta dataa maailmaan. Näitä tietoja talletetaan päivittäin kyseisten kauppojen sekä yritysten omistamille valtaville ja tehokkaille tietokoneille ja palvelimille. Näistä tietomääristä voidaan johtaa kaavoja, jotka kertovat meidän mielenkiinnon kohteistamme, tavoistamme ja käyttäytymismalleistamme. Tiedonlouhinta auttaa löytämään ja tulkitsemaan näitä malleja.

Tiedonlouhinnassa tieto on digitaalisesti tallennettua ja louhinta suoritetaan automaattisesti tietokoneella tai sitä on vähintäänkin tehostettu tietokoneiden avulla. Tiedonlouhinnan tarkoitus on ratkaista ongelmia analysoimalla olemassa olevaa tietoa. (Frank, Hall & Witten 2011, 4.)

Tiedonlouhinta on prosessi, jossa etsitään ja toivottavasti myös löydetään kaavoja tiedon joukosta. Löydettyjen kaavojen tulee olla tarkoituksenmukaisia ja johtaa jonkinlaiseen hyötyyn tai kehitykseen, yleensä taloudelliseen sellaiseen. Hyödylliset kaavat ovat sellaisia, joiden avulla pystytään tekemään monimuotoisia ennusteita käyttämällä uutta tietoa. (Frank & muut 2011, 5.)

Esimerkiksi, kun yrityksellä on tietoa asiakkaiden toimista eli tuotteiden klikkauksista ja niiden ostamisesta verkkokaupassa, yritys voi luoda valmiita kaavoja jo ostoksensa tehneiden asiakkaiden klikkausten ja niistä seuranneiden ostosten perusteella ja käyttää noita kaavoja ennustaakseen uuden asiakkaan klikkausten perusteella tämän mahdolliset tulevat ostokset ja mainostaa niitä erityisesti hänelle oston tekemisen nopeuttamiseksi, helpottamiseksi ja kasvattaakseen oston todennäköisyyttä. Näin jo olemassa olevan tiedon perusteella tehdään uudesta tiedosta ennusteita.

Tiedonlouhinta tarjoaa keinot ymmärtää valtavia tietomääriä automatisoimalla analysointiprosessin, jossa esimerkiksi kategorisoidaan ja klusteroidaan tietoa eri joukkoihin, ja siten etsitään ja tunnistetaan yleisiä suuntauksia sekä myös poikkeuksia datasta. (Crivat, MacLennan & Tang 2009, xxix.)

Han, Kamber & Pei määrittelevät tiedonlouhinnan olevan iteratiivinen tapahtumaketju, joka sisältää seuraavat kohdat:

1. Tiedon siistiminen – poistetaan kohina ja epäjohdonmukainen tieto.

2. Tiedon integrointi ja yhdistäminen – yhdistetään mahdollisesti useampia tiedonlähteitä.

3. Tiedon valinta – valitaan ja noudetaan analyysiin vain olennainen tieto.

4. Tiedonmuunnos – muutetaan tieto haluttuun, sopivaan muotoon analyysia varten.

5. Tiedonlouhinta – itse tiedonlouhinnan prosessi, jossa käyttämällä eri menetelmiä haetaan tiedosta kaavoja.

6. Kaavojen arviointi – etsitään ja havaitaan tuotoksista vain kiinnostavimmat kaavat, jotka tuottavat kiinnostavaa uutta tietoa.

7. Tiedon ja tulosten esittäminen – esitetään löydetyt tulokset havainnollisesti käyttämällä esimerkiksi visualisointia.

(Han, Jiawei, Kamber, Micheline & Pei, Jian 2012, 6-8.)

Tämän työn tiedonlouhintaesimerkissä nämä työvaiheet on jaettu kolmeen pakettiin: vaiheet 1-4 kuuluvat otsikon “Datan valmistelu” alle, 5 kuuluu otsikon “Ohjelma ja prosessi” alle ja 6-7 otsikon “Lopputulokset ja arviointi” alle.

3.1 Data ja datan valmistelu

Kuten edellisessä kappaleessa esitettiin, on datan valmistelu suuri osa koko tiedonlouhinnan pakettia. Ilman datan valmistelua itse louhintaa ei ole järkevää suorittaa. Datan valmistelu koostuu tiivistettynä neljästä vaiheesta: siistiminen, integrointi, valinta sekä muunnos. Järjestyksellä ei ole varsinaisesti väliä, mutta kaikki vaiheet kannattaa käydä huolella läpi, vaikka ne eivät vaatisikaan lopulta toimenpiteitä.

Siistimisellä tarkoitetaan kohinan ja epäjohdonmukaisen tiedon poistamista. Tähän tarkoitukseen on kehitelty sopivia laskukaavoja, mutta sen voi myös tehdä silmämääräisesti

tarkkailemalla esimerkiksi kunkin numeerisen attribuutin pienimpiä ja suurimpia arvoja ja poistamalla sieltä selvästi pienuudessaan tai suuruudessaan poikkeavat arvot.

Tietenkin todella suurissa aineistoissa manuaalinen poisto-operaatio veisi aivan liikaa aikaa ja se kannattaa automatisoida. Tällöin esimerkiksi voidaan käyttää kaavaa, jossa yksitellen jokaisen attribuutin kaikkien arvojen keskiarvo lasketaan. Siihen lisätään keskiarvoinen poikkeama kerrottuna kahdella ja näin saadaan arvojen korkein arvo. Kaikki tuon arvon ylittävät tietueet poistetaan siistimisen nimissä. Sama kuvio toistuu pienimmän arvon kohdalla vähentämällä keskiarvosta poikkeaman keskiarvo kerrottuna kahdella.

Siistimistä on myös tyhjien ruutujen poistaminen. Jos tyhjiä ruutuja on tietyssä attribuutissa erityisen paljon, voi olla järkevää poistaa koko attribuutti analyysistä tarpeettomana, sillä se ei tarjoa mitään tietoa. Toisaalta, jos yhdellä esimerkkitapauksella on paljon tyhjiä ruutuja, voi olla viisasta poistaa kyseinen tapaus analyysistä, sillä tapaus ei tarjoa tarpeeksi tietoa analyysille. Näitä täytyy toisinaan tarkastella täysin tapauskohtaisesti ja tiedon siistiminen voi viedä paljonkin aikaa, mutta on erityisen tärkeä etappi validin lopputuloksen kannalta ja parantaa tiedonlouhinnan luotettavuutta.

Integroinnilla ja yhdistelyllä tarkoitetaan sitä, että toisinaan tarvitaan tietoa useasta eri aineistosta, jolloin niitä saatetaan yhdistellä halutun lopputuloksen saavuttamiseksi. Tällöin on erityisen tärkeää olla tarkkana, että tieto pysyy validina eikä vääriä yhdistelmiä tehdä. Yleensä kuitenkin useammasta eri lähteestä kerättyä ja yhdistettyä tietoa analysoimalla saadaan paljon mielekkäämpiä ja hyödyllisempiä lopputuloksia.

Valinnalla tarkoitetaan sitä, että kun analyysimenetelmä on päätetty, valitaan mukaan vain menetelmää tukevat ja sille olennaiset attribuutit. Muunnoksella taas sitä, että muutetaan tieto haluttuun muotoon, jotta sitä voidaan analyysissä käsitellä oikein ja jotta se on esimerkiksi vertailukelpoista muun tiedon kanssa.

3.2 Prosessi ja menetelmät

Tiedonlouhinnan prosessi ja menetelmät syntyvät algoritmeista, joita käyttämällä saadaan olemassa olevaan tietoon perustuvia laskukaavoja. Tiedonlouhinnan menetelmiä on useita erilaisia, jotka kaikki soveltuvat parhaiten jonkin tietyn tai tietyn tyyppisen datan

analysointiin. Tämän työn esimerkissä (ks. luku 4) sovelletaan käytännössä numeeristen ennusteiden käyttöä ja tutkitaan niiden luotettavuutta, joten seuraavaksi esitellään erilaisia menetelmiä tuottaa numeerisia ennusteita.

3.2.1 Regressioanalyysit

Regressioanalyysi on tunnetuimpia sekä käytetyimpiä tekniikoita analysoitaessa monimuotoista dataa. Tekniikka perustuu loogiseen prosessiin, jossa hyödynnetään yhtälöä osoittamaan ja kuvaamaan suhdetta tietyn muuttujan sekä joukon ennustavia muuttujia kanssa. Tämän yksinkertaisen loogisuuden takia regressioanalyysi vetoaa laajalti ja koetaan hyödylliseksi. (Montgomery, Peck & Vining (2012, xiii.)

Esimerkiksi lineaarisessa regressiossa muodostetaan suoran yhtälöllä kaikista ennakkotapauksista (harjoitteludata) mahdollisimman suora viiva ja tarkastellaan kuinka lähelle viivaa tapaukset osuvat. Näin viivalla voidaan ennustaa tiettyjen arvojen avulla lopputuloksia sekä kuinka tarkkoja ne ovat. Osa ennakkotapausten attribuuteista on parempia lopputuloksen ennakoijia kuin toiset, ja ne saavat enemmän painoarvoa, kun ennakoidaan lopputulosta.

Tämän työn esimerkkiohjelmassa (ks. luku 4) tarkastellaan lineaarisen (Linear) ja isotonisen (Isotonic) regression sekä Pace Regression tarjoamia ennusteita ja niiden luotettavuutta.

3.2.2 Multilayer Perceptron

Multilayer Perceptron on myötäkytkentäinen neuroverkko, jossa on yksi tai useampia kerroksia tulokerroksen (input) ja lähtökerroksen (output) välillä. Se, että verkko on myötäkytkentäinen, tarkoittaa, että data virtaa yhdensuuntaisesti tulosta lähtöön päin. Jokainen kerros koostuu yksiköistä (units), jotka ottavat tulonsa alapuolella olevan kerroksen yksiköistä ja lähettävät lähtönsä yläpuolella olevan kerroksen yksiköihin. Multilayer Perseptronin tapauksessa oppimiseen (training) käytetään taaksepäin etenevää oppimisen algoritmia. Multilayer Percetron on laajalti käytetty muun muassa kaavojen luokitteluun, tunnistamiseen, ennustamiseen ja arvioimiseen. Tällä tekniikalla ratkaistaan ongelmia, joihin lineaariset tekniikat eivät muun muassa sovellu. (Neuro AI, 2013.)

3.2.3 Gaussian Processes ja SMOreg

Gaussian Processes on yksi suosituimpia mallintamisen menetelmiä käsitellessä dataa, jota havaitaan ajan, tilan tai sekä ajan että tilan kautta. Se on stokastinen prosessi ja perustuu Gaussin jakaumaan eli todennäköisyyksiin. Prosessien sanotaan olevan laskennallisesti raskaita. (Davis päiväämätön.)

SMOreg perustuu SMO (Sequential minimal optimization) algortimin käyttöön. SMO algoritmiä käytetään tukivektorikoneen opettamiseen (training) uudella tavalla ja johtaa näin parempiin yleistyksiin. Aikaisempiin tukivektorikoneen algoritmeihin verrattuna SMO pystyy käsittelemään suurempia tietoaineistoja ja nopeammin. (Platt, 1998.)

3.3 Tiedonlouhinnan automatisointi, onko se mahdollista ja miltä osin?

Koska tiedonlouhinta on määritelty tarkoittavan osittain automatisoitua datan analyysia, sen automatisointi on toki joiltain osin mahdollista. Tiedonlouhinta on tässä työssä jaettu kolmeen pääosaan: *datan valmistelu, prosessi* sekä *lopputulokset ja arviointi.* Louhinnan ensimmäinen vaihe, datan valmistelu, voidaan automatisoida vasta siinä vaiheessa, kun aineistoon ja sen tarjoamaan tietoon sekä mahdollisuuksiin on tarpeeksi perehdytty ja päätetty, mitä kyseisestä aineistosta halutaan saada irti. Kaikille aineistoille ei missään tapauksessa tehdä samanlaisia valmisteluja, vaan datan valmistelu riippuu täysin aineiston tarjoamasta tiedosta, sen sisältämistä attribuuteista ja sen yleisestä eheydestä ja johdonmukaisuudesta.

Kuitenkin kuten tässä työssä aiemmin mainittiin, voidaan tietyt raja-arvot ylittävä data karsia valmiilla kaavoilla laskettuna. Lisäksi, jos tiedetään, että tullaan käsittelemään useita samanlaisia aineistoja, voidaan näihin kaikkiin käyttää ensimmäisen kerran jälkeen automatisoidusti samoja datan valmistelumenetelmiä. Kuitenkin tämä tehdään niin, että aina datan valmistelu tulee aloittaa tutkimalla datan rakennetta ja tarjontaa täysin omilla silmillä ja pohtia dataa ja sen sisältöä täysin omilla aivoilla.

Kun data on valmisteltu, yleensä myös louhintamenetelmä on päätetty. Louhintamenetelmä on tässä vaiheessa suurilta osin automatisoitu, mutta saattaa vaatia jotain raja-arvojen asettamista tai muuta tarkennusta haluttaessa parempia tuloksia. Kuitenkin lähes minkä tahansa louhintamenetelmän voi ajaa läpi oletusasetuksilla ja saada esille tuloksia. Siksi voidaan sanoa, että tämä kohta voidaan automatisoida, mutta ihmisen

päättelykyky ja ammattitaito on myös tässä vaiheessa tärkeää ja johtaa todennäköisesti parempiin ja hedelmällisempiin lopputuloksiin.

Lopputulosten esittämiseen liittyy tiettyä automatisointia, sillä usein on mielekästä esittää tulokset graafisesti ja visuaalisesti, jotta ne on helpompi ja nopeampi sisäistää sekä ymmärtää. Tulosten esittämisen voi siis vielä automatisoida.

Lopputuloksia arvioitaessa taas ammattitaito ja osaaminen ovat suuressa osassa eikä tätä osiota ole enää mitään järkeä automatisoida. Ainoa tapa saada tuloksista mitään irti on lukea ja ymmärtää ne kukin omalla tavallaan.

Tiivistettynä tiedonlouhinta on jatkuvaa asioiden priorisointia sekä pohtimista ja omilla aivoilla tehtävää päättelyä louhinnan alusta loppuun saakka, eli vain pieniä osia tiedonlouhinnasta voidaan automatisoida, mutta ne osat sitten kannattaakin, ettei työmäärä yhden aineiston kohdalla kasva pilviin. Siksi on kehitetty näppäriä sovelluksia, joilla louhinnan automatisoitavat vaiheet voidaan suorittaa nopeasti ja vaivattomasti ja niin, että siihen kykenee lähes kuka tahansa perustasolla.

3.4 Sovellukset

Tiedonlouhintaan on kehitetty useita eri sovelluksia. Suurin osa niistä on yritysmaailmaan suunnattuja maksullisia ohjelmia, joihin ei tässä työssä perehdytä lainkaan. Avoimen lähdekoodin sovelluksista tunnetuimmat, mutta kuitenkin kovin erilaiset keskenään ovat RapidMiner sekä Weka. Koska tämän työn esimerkkilouhinnassa on käytetty kumpaakin sovellusta eri käyttötarkoituksiin, ne esitellään molemmat varsinaisesti vertailematta toisiinsa.

3.4.1 RapidMiner

Suosituimpia avoimen lähdekoodin tiedonlouhintasovelluksia on RapidMiner. Tämän työn esimerkissä RapidMineria on käytetty datan valmisteluun, sillä RapidMiner selkeällä graafisella käyttöliittymällään sopii datan valmisteluun erinomaisesti. Tämä toimii lisäksi selkeänä esimerkkinä lukijalle RapidMinerin käytöstä.

RapidMiner osaa käsitellä ja hyödyntää useita eri tiedostotyyppejä ja datanlähteitä, kuten Excel, Access, Oracle Microsoft SQL, MySQL, SPSS ja Salesforce. Se on kehitetty Java-ohjelmointikielellä erityisesti pärjäämään myös big datan louhinnassa muun data-analyysin ohella. RapidMiner on kehitetty toimimaan useimmilla ja suurimmilla alustoilla ja käyttöjärjestelmillä. RapidMiner-ohjelma sisältää laajan valikoiman tiedonlouhinta menetelmiä ja algoritmeja.

RapidMiner tarjoaa vahvan visuaalisen ympäristön tiedonlouhinnalle sekä laajat mahdollisuudet tehdä lopputuloksista juuri itsesi näköiset ja juuri käyttöösi sopivat. RapidMinerissa on mahdollista valita helppokäyttöisen graafisen käyttöjärjestelmän tai koodin syöttöön perustuvan ohjauksen väliltä. RapiMiner on AGPL-lisensoitu avoimen lähdekoodin ohjelma.

3.4.2 Weka

Weka on Waikaton yliopiston kehittämä työkalu tiedonlouhintaan, joka on erikoistunut myös big datan käsittelyyn. Weka sisältää siis kokoelman erilaisia algoritmeja tiedonlouhintaan ja sitä voidaan käyttää joko sen omalla yksinkertaisella ja riisutulla käyttöliittymällä tai kutsumalla Weka API:N kautta algoritmeja omassa Java-koodissa. Louhinta-algoritmien lisäksi Weka sisältää datan valmisteluun käytettäviä työkaluja sekä tulosten esittämiseen käytettäviä visualisoinnin työkaluja. Weka on avoimen lähdekoodin sovellus ja on GNU GPL (General Public License) -lisensoitu.

3.4.3 Muut

Muita avoimen lähdekoodin tiedonlouhintasovelluksia ovat muun muassa R-Programming, Orange ja NLTK. R-Programming on R-ohjelmointikielellä kirjoitettu tiedonlouhintaohjelma, jonka maine on kirinyt viime vuosien aikana. Orange on Python-ohjelmointikielellä kehitetty yksinkertainen ja helposti opittava ohjelma tiedonlouhintaan. NLTK on Pythonilla kehitetty työkalu kielen ja tekstin prosessointiin.

4 Sovellus ja Wekan API

4.1 Tarkoitus

Jotta lukija ymmärtäisi, mitä iloa big datasta ja tiedonlouhinnasta voi olla ja miten kuka tahansa voi hyödyntää esimerkiksi Wekan tarjoamaa ohjelmointirajapintaa, on tätä työtä varten rakennettu Java-sovellus. Käyttäen Wekan ohjelmointirajapintaa sovellus suorittaa kuusi eri tiedonlouhintamenetelmää, jotka kaikki ovat numeerisia ennusteita eli numeerisen datan louhinnasta seuraavia ennustuksia lopputuloksesta. Ohjelman on tarkoitus päätellä Helsingin Sanomien tarjoaman vaalikonedatan vastauksista kyseisen vastaajan ikä ja tutkia samalla, kuinka luotettava numeerinen ennuste voi olla ja mikä numeerisista ennusteista on mahdollisesti luotettavin menetelmä.

Numeeriset ennusteet valikoituivat menetelmäksi, sillä käytettävissä oleva aineisto (ks. alaluku 4.4) sisältää paljon puhdasta numeerista dataa. Jotta ennusteiden osuvuutta voidaan arvioida, tulee meidän tietää oikeat lopputulokset kunkin ennusteen kohdalla. Tämän takia päädyttiin ennustamaan jotain, mikä oikeasti on jo tiedossa. Tarkoitus ei siis varsinaisesti tällä tiedonlouhinnalla ole keksiä ja löytää uutta tietoa olemassa olevan tiedon avulla, vaan esitellä, kuinka näin voisi tehdä. Mielenkiintoisen näkökulman esimerkkiin tuo vertailu siitä, kuinka lähelle oikeaa tulosta päästää ja kuinka usein.

Koska HS:n vaalikoneen vastaukset sisältävät paljon muutakin kuin numeerista tietoa, olisi analyysia voinut tehdä monella muullakin tavalla. Aineistoa tutkimalla olisi voitu kehittää kaavoja, joista selviää esimerkiksi, miten vastaajan koulutus vaikuttaa vastaukseen tietyssä kysymyksessä tai siihen, mihin puolueeseen hän kuuluu, miten vastaajaan puolue vaikuttaa vastauksiin kysymyksissä tai esimerkiksi miten sukupuoli ja ikä vaikuttavat siihen, onko vastaajalla mahdollisesti Twitter- tai Facebook-tili.

Tämän työn louhintaosuus on mukana, jotta Weka API:n käyttöönotto saadaan esiteltyä mielekkäällä sekä selkeällä tavalla, joten nähtiin viisaimmaksi käyttää tarjolla olevaa puhdasta numeerista dataa, joka vaatii hyvin paljon vähemmän valmistelua kuin aineiston polynomiset attribuutit ja niiden vastaukset.

4.2 Weka API

Weka API on Wekan kehittämä ohjelmointirajapinta, joka mahdollistaa Wekan tiedonlouhintamekanismien integroimisen omaan sovellukseesi. Weka API:lla voit tuoda kaikkien eri tiedonlouhintamenetelmien paketit omaan koodiisi ja käyttää kaikkia samoja funktioita, joita itse Weka-ohjelman sisälläkin käytetään. Tärkeimmät tarvittavat komponentit ovat:

Instances (instanssit) - niihin haetaan oma louhittava data.

Filter (filtteri) - valmistellaan oma louhittava data.

Attribute Selection (attribuuttien valinta) - valitaan mukaan otettavat attribuutit myöskin datan valmisteluvaiheessa.

Classifier/Clusterer - haluttu louhintamenetelmä, jossa esimerkiksi classifier opettaa alkuperäisellä datalla, miten uutta dataa tulee käsitellä.

Evaluating (arviointi) – arvio siitä, kuinka hyvä ja tarkka valittu menetelmä on ollut.

4.3 Suunnitelma ja tavoitteet

Esimerkkiohjelman suunnittelu lähti liikkeelle tutkimalla tarjolla olevaa dataa ja selvittämällä kokeilemalla, mitä siitä voisi saada irti. Koska aineisto sisältää paljon numeerisia arvoja, tuntui luontevalta alkaa tutkimaan lähemmin erilaisia numeerisia ennusteita. Numeerisia ennusteita on useita, ja ne käyttävät eri algoritmeja. Siksi ajattelin olevan ainakin itselleni mielenkiintoista sekä mielekästä tutkia, mitkä ennusteista olisivat mahdollisesti tarkimpia ja luotettavimpia. Kuitenkin ohjelman toinen tavoite on esitellä, kuinka Wekan tarjoamaa API:a voidaan käyttää Java-ohjelmointikielellä ja siihen tarkoitukseen ei ole mielekästä olla liikaa esimerkkejä eri algoritmeista. Koska tavoitteena on opettaa yksinkertainen ja selkeä tapa hyödyntää API:a omassa sovelluksessaan, oli tärkeää dokumentoida selkeästi kuvien kera ohjelman syntyminen.

4.4 Data

HS:n vaalikoneen vastaukset valikoitui käytettäväksi datan lähteeksi, sillä data oli erityisen siistiä ja yhdenmukaista. Siinä on hyvin vähän puuttuvia arvoja ja paljon numeerisia vastauksia, joita on helppo käsitellä. Tässä aineistossa on yhteensä jopa 40 attribuuttia, mikä selittyy sillä, että vaalikoneen kysymyksiä on 30, ja ne ovat jokainen oma attribuuttinsa. Attribuutit ovat: *id*, *name*, *district*, *party*, *age*, *gender*, *www*, *facebook*, *twitter*, *education* sekä kysymykset *q1-q30*. Seuraavassa on kuvattu kaikki attribuutit yksitellen, paitsi kysymykset q1-q30, jotka on kuvattu kaikki kerralla.

id – Automaattisesti generoitu numerotunniste jokaiselle vastaajalle, numeerinen.

name – Vastaajan nimi, muodossa polynominal. Vastaaja on itse vapaasti kirjoittanut nimensä vastatessaan kyselyyn.

party – Vastaajan puolue, muodossa polynominal. Vastaaja on valinnut puolueensa listasta puoleita.

age – Vastaajan ikä, muodossa numeerinen. Vastaaja on itse vapaasti kirjoittanut ikänsä vastatessaan kyselyyn.

gender – Vastaajan sukupuoli, muodossa polynominal. Vastaaja on valinnut sukupuolensa vaihtoehdoista MALE (mies), FEMALE (nainen) tai NULL (ei vastausta).

www – Vastaajan mahdollinen kotisivuosoite, muodossa polynominal. Vastaaja on itse vapaasti kirjoittanut mahdollisen kotisivujensa osoitteen vastatessaan kyselyyn tai jättänyt kohdan tyhjäksi, jolloin arvo on NULL.

facebook - Vastaajan mahdollinen Facebook-tilin osoite, muodossa polynominal. Vastaaja on itse vapaasti kirjoittanut mahdollisen Facebook-tilinsä osoitteen vastatessaan kyselyyn tai jättänyt kohdan tyhjäksi, jolloin arvo on NULL.

twitter - Vastaajan mahdollinen Twitter-tilin osoite, muodossa polynominal. Vastaaja on itse vapaasti kirjoittanut mahdollisen Twitter-tilinsä osoitteen vastatessaan kyselyyn tai jättänyt kohdan tyhjäksi, jolloin arvo on NULL.

education – Vastaajan koulutustaso, muodossa polynominal. Vastaaja on itse vapaasti kirjoittanut koulutustasonsa vastatessaan kyselyyn tai jättänyt kohdan tyhjäksi, jolloin arvo on NULL.

q1-q30 – Vastaajan vastaukset kysymyksiin 1-30, muodossa numeerinen. Vastaaja on valinnut vastauksensa numeerisista vaihtoehdoista 1-5 tai jättänyt vastaamatta ollenkaan, jolloin arvo NULL.

4.5 Datan valmistelu

Koska datan valmistelua ei voi automatisoida, se tulee tehdä ennen varsinaista tiedonlouhintaa. Tässä esimerkissä käytetty Helsingin Sanomien tarjoama vaalikonedata sisältää paljon epäolennaisia attribuutteja ja joitakin puuttuvia arvoja. Lisäksi esimerkin luonteen vuoksi tarvitaan kaksi eri aineistoa: toinen, jossa attribuutilla age on olemassa numeeriset arvot, ja toinen, jossa arvot on poistettu ja korvattu kysymysmerkillä. Koska esimerkissä käytetään tiedonlouhintaan Wekan tarjoamaa API:a ja Weka on suunniteltu tukemaan arff-tiedostojen louhintaa, tulee molemmat aineistot muuttaa csv-tiedostomuodosta arff-tiedostomuotoon. Esimerkkiohjelman käyttämät tiedostot on siis valmisteltu ja luotu käyttämällä RapidMiner-tiedonlouhintasovelluksen versiota 5.3.000. Käytetyn ohjelman valinta perustui sen helppokäyttöisyyteen ja selkeyteen datan valmistelussa. Graafinen käyttöliittymä helpotti huomattavasti monimutkaisen datan valmistelua tähän esimerkkiin sopivaksi.

Datan valmistelu lähtee liikkeelle hakemalla alkuperäinen HS:n tarjoama csv-tiedosto (Retrieve HSVaalidata). Koska aineistossa vastanneiden henkilöiden id on merkitty normaalin attribuutin sijaan roolilla ”id” se tulee muuttaa normaaliksi attribuutiksi (Set Role), jotta sen voi myöhemmin tarpeettomana poistaa käytöstä (id-roolilla merkittyä attribuuttia ei voi poistaa). Seuraavaksi poistetaan käytöstä esimerkin tiedonlouhinnan kannalta tarpeettomat attribuutit ja niiden vastaukset (Select Attributes).

Koska työn esimerkissä halutaan tehdä numeerisia ennusteita vastauksien q1-q30 perusteella ja yritetään ennustaa ikää, tarvitaan nyt vain attribuutteja ”age” (ikä) ja ”q1”-”q2” (vastaukset monivalintakysymyksiin). Ylimääräisiä ovat attribuutit ”district” ”education”, ”facebook”, ”gender”, ”id”, ”name”, ”party”, ”twitter” ja ”www”. Koska kysymyksen 1 koh-

dalla on jonkin verran puuttuvia vastauksia, muttei kuitenkaan niin paljon, että koko kysymys tulisi poistaa louhinnasta, poistetaan kaikki vastaajat (instances), jotka eivät ole vastanneet kysymykseen 1. Näin päästään eroon puuttuvista arvoista, jotka saattavat häiritä tiedonlouhinnan suorittamista sekä väärentää johtopäätöksiä.

Kuten aiemmin todettiin, tarvitsemme kaksi aineistoa esimerkin tiedonlouhinnan toteuttamiseen, joten tässä kohtaa aineisto jaetaan kahtia siten (Split Data), että saadaan kaksi erillistä aineistoa, joista toinen sisältää 70 prosenttia instansseista ja toinen 30 prosenttia instansseista. Näiden aineistoa valmistelua jatketaan siis tästä eteenpäin erillään toisistaan. Aineiston, joka sisältää 70 prosenttia instansseista, attribuutit uudelleen järjestetään (Reorder Attributes), jotta esimerkissä tutkimamme age-attribuutti saadaan viimeiseksi. Tämä toimenpide liittyy Wekan tapaan käsitellä aineistoja ja selkenee myöhemmin. Tällä kertaa riittää, että attribuutit järjestetään laskevaan aakkosjärjestykseen. Tämän jälkeen aineisto tallennetaan arff-tiedostomuodossa tietokoneelle nimellä Trainingdata. Seuraavaksi jatketaan toisen aineiston (30 prosenttia instansseista) valmistelua. Tästä aineistosta tulee Scoringdata eli siitä tulee puuttumaan arvot attribuutille ”age”. Tämänkin aineiston attribuutit halutaan järjestää uudelleen samaan järjestykseen kuin edellisen eli laskevaan aakkosjärjestykseen, jotta ”age”-attribuutti on viimeisenä. Tässä kohtaa, kun ”age”-attribuutin arvoja ei ole vielä poistettu, tallennetaan aineisto arff-tiedostomuodossa nimellä TrueAges, jotta voidaan myöhemmin vertailla, kuinka lähelle oikeita arvoja on päästy. Seuraavaksi poistetaan ”age” attribuutti kokonaan ja lisätään se uudestaan, jotta saadaan kaikkiin instansseihin sen kohdalle kysymysmerkki eli tyhjä. Järjestetään attribuutit taas uudelleen, jotta ”age”-attribuutti saadaan viimeiseksi ja tämän jälkeen tallennetaan arff-tiedostomuodossa nimellä Scoringdata.

Nyt meillä on siis opetusaineisto, joka sisältää 70 prosenttia alkuperäisestä aineistosta ja jossa on myös ”age” arvot, Scoring-aineisto, joka sisältää 30 prosenttia alkuperäisestä aineistosta ja, josta on poistettu arvot attribuutilta ”age” sekä TrueAges-aineisto, joka on muuten sama, kuin Scoring-aineisto, mutta joka sisältää instanssien todelliset arvot attribuutille ”age”. Näitä aineistoja lähdetään louhimaan Wekan avulla Java-pohjaisessa ohjelmassa.

Aineistot ovat nyt arff-muodossa, joka on tekstitiedosto, joka muodostuu siten, että ensimmäinen rivi antaa aineistolle nimen tunnisteen ”@relation” jälkeen, seuraavat rivit kuvaavat aineiston attribuutit aina tunnisteetn ”@attribute” jälkeen asettaen ensin attribuutin nimen ja sen jälkeen tyypin, tässä tapauksessa kaikissa real eli kaksoistarkkuuden

liukuluku. Nämä ensimmäiset rivit Scoring-aineiston tekstitiedostosta esitellään kuvassa 2.

```
@RELATION RapidMinerData

@ATTRIBUTE 'q9' real
@ATTRIBUTE 'q8' real
@ATTRIBUTE 'q7' real
@ATTRIBUTE 'q6' real
@ATTRIBUTE 'q5' real
@ATTRIBUTE 'q4' real
@ATTRIBUTE 'q30' real
@ATTRIBUTE 'q3' real
@ATTRIBUTE 'q29' real
@ATTRIBUTE 'q28' real
@ATTRIBUTE 'q27' real
@ATTRIBUTE 'q26' real
@ATTRIBUTE 'q25' real
@ATTRIBUTE 'q24' real
@ATTRIBUTE 'q23' real
@ATTRIBUTE 'q22' real
@ATTRIBUTE 'q21' real
@ATTRIBUTE 'q20' real
@ATTRIBUTE 'q2' real
@ATTRIBUTE 'q19' real
@ATTRIBUTE 'q18' real
@ATTRIBUTE 'q17' real
@ATTRIBUTE 'q16' real
@ATTRIBUTE 'q15' real
@ATTRIBUTE 'q14' real
@ATTRIBUTE 'q13' real
@ATTRIBUTE 'q12' real
@ATTRIBUTE 'q11' real
@ATTRIBUTE 'q10' real
@ATTRIBUTE 'q1' real
@ATTRIBUTE 'age' real
```

Kuva 2. VaalidataScoring.arff-tiedoston muoto.

Seuraavat rivit tunnisteen ”@data” jälkeen sisältävät itse datan eli jokaisen tietueen ja arvot kaikkiin edellä mainittuihin attribuutteihin yhden desimaalin tarkkuudella. Nämä rivit Scoring-aineiston tekstitiedostosta nähdään kuvassa 3.

```
@DATA
1.0,4.0,4.0,5.0,4.0,4.0,5.0,3.0,1.0,5.0,1.0,2.0,5.0,5.0,5.0,5.0,1.0,1.0,4.0,5.0,4.0,1.0,5.0,4.0,1.0,2.0,4.0,2.0,2.0,5.0,?
1.0,3.0,3.0,2.0,3.0,2.0,5.0,3.0,1.0,3.0,2.0,1.0,2.0,4.0,4.0,3.0,1.0,4.0,2.0,3.0,3.0,4.0,5.0,1.0,3.0,2.0,2.0,3.0,4.0,5.0,?
2.0,2.0,2.0,4.0,4.0,2.0,5.0,4.0,2.0,2.0,2.0,2.0,2.0,5.0,4.0,4.0,1.0,4.0,4.0,4.0,2.0,5.0,2.0,2.0,2.0,4.0,4.0,2.0,2.0,4.0,?
2.0,2.0,4.0,2.0,4.0,2.0,4.0,4.0,4.0,2.0,2.0,2.0,2.0,5.0,5.0,5.0,1.0,1.0,2.0,5.0,4.0,2.0,4.0,4.0,1.0,4.0,4.0,4.0,1.0,5.0,?
1.0,4.0,4.0,2.0,4.0,2.0,4.0,1.0,2.0,4.0,1.0,2.0,4.0,5.0,2.0,3.0,1.0,4.0,4.0,4.0,1.0,2.0,4.0,4.0,4.0,2.0,4.0,2.0,2.0,5.0,?
1.0,4.0,5.0,3.0,4.0,3.0,4.0,2.0,2.0,4.0,3.0,4.0,2.0,5.0,5.0,4.0,1.0,4.0,4.0,4.0,3.0,4.0,2.0,5.0,4.0,2.0,3.0,5.0,2.0,4.0,?
4.0,4.0,2.0,2.0,5.0,4.0,4.0,1.0,2.0,4.0,4.0,2.0,4.0,4.0,2.0,1.0,5.0,5.0,2.0,4.0,1.0,1.0,4.0,1.0,4.0,2.0,5.0,4.0,1.0,4.0,?
5.0,2.0,2.0,2.0,5.0,4.0,5.0,2.0,2.0,5.0,2.0,2.0,4.0,4.0,4.0,4.0,5.0,2.0,2.0,4.0,4.0,2.0,5.0,4.0,5.0,5.0,1.0,2.0,4.0,4.0,?
4.0,2.0,4.0,2.0,4.0,4.0,5.0,4.0,2.0,4.0,3.0,2.0,4.0,4.0,4.0,4.0,5.0,2.0,1.0,4.0,4.0,1.0,5.0,4.0,4.0,5.0,2.0,4.0,4.0,5.0,?
4.0,2.0,4.0,2.0,4.0,2.0,4.0,4.0,4.0,2.0,4.0,2.0,4.0,5.0,4.0,2.0,4.0,4.0,1.0,5.0,4.0,1.0,5.0,4.0,5.0,4.0,4.0,4.0,1.0,5.0,?
5.0,2.0,4.0,4.0,4.0,2.0,4.0,5.0,4.0,5.0,4.0,2.0,4.0,4.0,4.0,4.0,5.0,2.0,1.0,5.0,4.0,1.0,5.0,4.0,4.0,4.0,2.0,2.0,2.0,4.0,?
1.0,4.0,4.0,5.0,5.0,2.0,2.0,5.0,4.0,5.0,4.0,1.0,4.0,5.0,5.0,1.0,1.0,2.0,2.0,5.0,4.0,1.0,5.0,2.0,4.0,4.0,2.0,5.0,1.0,5.0,?
5.0,4.0,4.0,2.0,5.0,4.0,5.0,2.0,1.0,5.0,4.0,2.0,4.0,4.0,2.0,5.0,5.0,1.0,1.0,4.0,4.0,2.0,5.0,4.0,4.0,4.0,2.0,2.0,4.0,4.0,?
5.0,4.0,4.0,2.0,5.0,4.0,4.0,5.0,2.0,5.0,2.0,1.0,4.0,4.0,4.0,1.0,4.0,2.0,1.0,5.0,4.0,1.0,5.0,2.0,5.0,4.0,1.0,4.0,1.0,5.0,?
2.0,4.0,4.0,2.0,4.0,4.0,2.0,4.0,4.0,4.0,2.0,2.0,4.0,5.0,4.0,4.0,4.0,2.0,2.0,4.0,4.0,1.0,4.0,2.0,4.0,5.0,4.0,4.0,2.0,4.0,?
2.0,2.0,1.0,5.0,4.0,2.0,2.0,5.0,4.0,1.0,4.0,5.0,4.0,5.0,5.0,1.0,1.0,5.0,4.0,5.0,1.0,1.0,1.0,2.0,5.0,2.0,5.0,1.0,1.0,4.0,?
5.0,2.0,2.0,5.0,5.0,2.0,2.0,2.0,4.0,5.0,1.0,2.0,2.0,5.0,4.0,1.0,2.0,5.0,5.0,5.0,1.0,1.0,1.0,1.0,5.0,1.0,5.0,2.0,1.0,5.0,?
2.0,4.0,2.0,5.0,4.0,2.0,2.0,4.0,4.0,4.0,1.0,2.0,2.0,5.0,1.0,1.0,1.0,4.0,4.0,5.0,1.0,1.0,1.0,1.0,5.0,4.0,4.0,1.0,1.0,2.0,?
2.0,1.0,4.0,5.0,4.0,4.0,5.0,4.0,2.0,4.0,1.0,2.0,1.0,4.0,5.0,1.0,2.0,5.0,2.0,4.0,2.0,3.0,2.0,2.0,5.0,2.0,4.0,1.0,1.0,4.0,?
5.0,5.0,4.0,5.0,5.0,4.0,5.0,4.0,4.0,4.0,2.0,2.0,2.0,4.0,1.0,2.0,3.0,5.0,4.0,4.0,1.0,2.0,1.0,2.0,5.0,5.0,5.0,4.0,2.0,5.0,?
4.0,2.0,1.0,2.0,1.0,4.0,5.0,2.0,1.0,1.0,1.0,4.0,2.0,4.0,2.0,4.0,5.0,4.0,4.0,4.0,2.0,2.0,4.0,4.0,5.0,4.0,1.0,5.0,1.0,4.0,?
2.0,4.0,2.0,2.0,2.0,3.0,5.0,2.0,1.0,1.0,1.0,5.0,1.0,4.0,1.0,4.0,5.0,1.0,4.0,1.0,1.0,5.0,5.0,1.0,4.0,1.0,4.0,1.0,4.0,2.0,?
2.0,2.0,1.0,2.0,2.0,4.0,4.0,1.0,2.0,2.0,1.0,5.0,1.0,2.0,2.0,4.0,5.0,4.0,4.0,2.0,1.0,5.0,5.0,5.0,2.0,4.0,1.0,2.0,1.0,4.0,2.0,?
2.0,4.0,2.0,2.0,2.0,2.0,5.0,5.0,2.0,4.0,2.0,4.0,2.0,4.0,4.0,4.0,4.0,4.0,4.0,4.0,4.0,1.0,5.0,4.0,1.0,2.0,1.0,4.0,1.0,2.0,4.0,?
4.0,4.0,1.0,3.0,1.0,4.0,5.0,1.0,1.0,1.0,1.0,5.0,1.0,1.0,1.0,5.0,5.0,5.0,5.0,3.0,1.0,3.0,1.0,1.0,5.0,3.0,4.0,1.0,1.0,1.0,?
2.0,2.0,1.0,1.0,1.0,4.0,5.0,1.0,1.0,1.0,1.0,5.0,1.0,2.0,2.0,5.0,5.0,2.0,5.0,2.0,1.0,5.0,2.0,4.0,4.0,1.0,5.0,1.0,1.0,2.0,?
1.0,1.0,5.0,1.0,4.0,4.0,5.0,1.0,1.0,5.0,1.0,4.0,1.0,5.0,5.0,5.0,5.0,1.0,5.0,1.0,1.0,4.0,5.0,5.0,5.0,1.0,1.0,1.0,4.0,4.0,?
3.0,4.0,2.0,1.0,1.0,5.0,5.0,1.0,1.0,1.0,1.0,5.0,1.0,1.0,2.0,5.0,5.0,2.0,5.0,1.0,1.0,5.0,2.0,1.0,4.0,1.0,4.0,1.0,2.0,1.0,?
2.0,4.0,2.0,5.0,1.0,4.0,4.0,1.0,3.0,2.0,2.0,4.0,2.0,2.0,2.0,5.0,3.0,5.0,2.0,1.0,3.0,4.0,3.0,4.0,2.0,4.0,1.0,2.0,1.0,?
1.0,1.0,1.0,5.0,1.0,4.0,5.0,2.0,1.0,1.0,1.0,5.0,1.0,5.0,1.0,5.0,5.0,5.0,5.0,5.0,1.0,5.0,1.0,1.0,5.0,1.0,1.0,1.0,1.0,1.0,?
4.0,1.0,1.0,5.0,2.0,4.0,5.0,1.0,1.0,2.0,1.0,4.0,2.0,1.0,1.0,4.0,5.0,1.0,2.0,2.0,1.0,4.0,5.0,4.0,2.0,1.0,2.0,1.0,4.0,2.0,?
2.0,5.0,4.0,2.0,2.0,4.0,5.0,1.0,1.0,2.0,1.0,4.0,2.0,3.0,2.0,5.0,5.0,1.0,4.0,4.0,2.0,5.0,4.0,1.0,5.0,2.0,4.0,1.0,4.0,5.0,?
5.0,4.0,4.0,3.0,4.0,4.0,3.0,3.0,4.0,4.0,3.0,3.0,3.0,4.0,2.0,2.0,2.0,1.0,1.0,4.0,4.0,3.0,4.0,4.0,4.0,4.0,1.0,2.0,2.0,4.0,?
1.0,1.0,1.0,1.0,2.0,4.0,5.0,1.0,1.0,1.0,1.0,5.0,1.0,5.0,1.0,5.0,5.0,2.0,4.0,4.0,1.0,1.0,5.0,1.0,5.0,1.0,4.0,1.0,1.0,2.0,?
2.0,2.0,4.0,2.0,5.0,2.0,4.0,2.0,3.0,4.0,2.0,2.0,4.0,5.0,5.0,2.0,1.0,4.0,2.0,5.0,4.0,4.0,3.0,2.0,2.0,4.0,4.0,4.0,2.0,4.0,?
4.0,4.0,4.0,1.0,4.0,4.0,5.0,1.0,2.0,4.0,4.0,2.0,4.0,4.0,2.0,4.0,2.0,1.0,1.0,4.0,4.0,1.0,4.0,5.0,2.0,4.0,1.0,5.0,2.0,5.0,?
1.0,4.0,4.0,1.0,5.0,4.0,4.0,2.0,2.0,2.0,1.0,4.0,4.0,5.0,1.0,4.0,1.0,4.0,5.0,5.0,3.0,1.0,3.0,1.0,4.0,5.0,4.0,1.0,2.0,2.0,?
4.0,2.0,2.0,1.0,2.0,4.0,5.0,1.0,1.0,4.0,2.0,4.0,2.0,4.0,4.0,5.0,5.0,1.0,2.0,4.0,1.0,2.0,5.0,2.0,4.0,2.0,4.0,1.0,2.0,2.0,?
5.0,5.0,4.0,4.0,4.0,4.0,5.0,2.0,1.0,4.0,1.0,2.0,4.0,3.0,1.0,5.0,5.0,1.0,2.0,2.0,4.0,2.0,5.0,2.0,2.0,4.0,4.0,4.0,1.0,4.0,?
2.0,4.0,4.0,2.0,4.0,2.0,4.0,4.0,2.0,4.0,2.0,2.0,4.0,4.0,2.0,4.0,3.0,2.0,2.0,4.0,2.0,2.0,4.0,2.0,2.0,2.0,4.0,2.0,1.0,4.0,?
4.0,1.0,3.0,3.0,4.0,4.0,4.0,2.0,2.0,4.0,4.0,1.0,4.0,4.0,3.0,5.0,5.0,2.0,2.0,4.0,2.0,3.0,4.0,4.0,2.0,2.0,5.0,4.0,1.0,4.0,?
3.0,5.0,4.0,4.0,3.0,2.0,4.0,5.0,4.0,2.0,1.0,2.0,4.0,4.0,2.0,5.0,5.0,1.0,5.0,5.0,1.0,3.0,5.0,5.0,2.0,4.0,5.0,1.0,2.0,5.0,?
2.0,2.0,2.0,2.0,2.0,4.0,4.0,1.0,2.0,2.0,4.0,4.0,2.0,4.0,2.0,5.0,2.0,1.0,4.0,2.0,2.0,2.0,4.0,2.0,2.0,2.0,4.0,1.0,2.0,4.0,?
```

Kuva 3. VaalidataScoring.arff-tiedoston muoto.

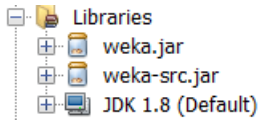
4.6 Ohjelma ja prosessi

Työn tarkoitus on esitellä Wekan tarjoaman API:n käyttöä omassa Java-sovelluksessa. Tätä varten suunniteltiin ja toteutettiin Javalla Netbeansissa esimerkkiohjelma, jolla hyödynnetään Wekan tarjoamaa ohjelmointirajapintaa tiedonlouhintaan.

Aivan ensimmäiseksi on suositeltavaa ladata ja asentaa Weka-sovellus, sillä sen mukana tulee myös NeatBeansissä tarvittava weka-src.jar-tiedosto. Tämän kirjastotiedoston voi myös ladata erikseen samalta sivulta kuin itse ohjelmankin: http://www.cs.waikato.ac.nz/ml/weka/downloading.html (Luettu 23.4.2016.).

Tässä työssä on käytetty Wekan Stable-versiota 3.6, joka asentaa myös Javan version vM 1.8. Käyttöjärjestelmänä on 64-bittinen Windows 10. Tämän työn esimerkkiohjelma on toteutettu käyttäen NetBeans-ohjelmointialustan versiota 8.1.

Työn tekeminen lähti käyntiin luomalla uusi projekti NetBeansissa ja lisäämällä projektin kirjastoiksi (Libraries) Wekan tarjoamat weka-src.jar sekä weka.jar (kuva 4). Näiden kirjastojen avulla voidaan käyttää Wekan API:a.



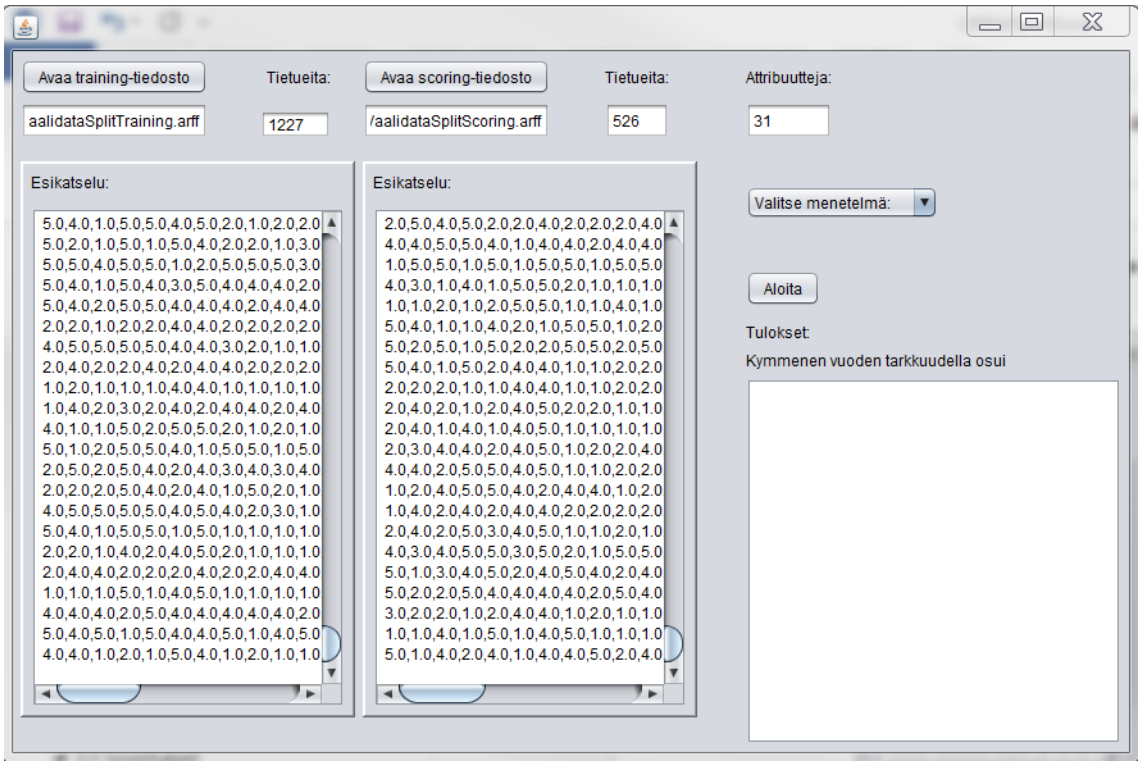
Kuva 4. Projektin kirjastoihin haettu Wekan tarjoamat .jar-tiedostot

Koska ohjelma on hyvin yksinkertainen, jotta myös esimerkki olisi yksinkertainen ja helppo ymmärtää, projekti sisältää vain kolme luokkaa. Ensimmäinen luokka DataMiningMain.java (kuva 5) eli pääluokka sisältää pelkästään käyttöliittymän avaamisen eli esille laiton.

```
public class DataMiningMain {

    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new DataMiningUI().setVisible(true);
            }
        });
    }

}
```

Kuva 5. Sovelluksen Main-luokka ja sen toiminnot.

Toisessa luokassa DataMiningUI:ssa rakennetaan sovelluksen käyttöliittymä (kuva 6) käyttäen NetBeansin tarjoamaa käyttöliittymän rakennusapuria Swing Palettea. Käyttöliittymässä valitaan FileChooserilla tietokoneen tiedostoista valmiiksi sekä training- että scoring-data. Haettuaan ne liittymä esittelee valitut tiedostot esikatselu-ikkunassa ja laskee, kuinka paljon tietueita sekä attribuutteja aineistot sisältävät.



Kuva 6. Sovelluksen käyttöliittymä, jossa aineistot on jo haettuna.

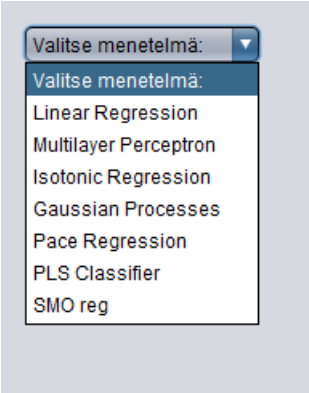
Tähän liittyvä kaikki toiminnallisuus suoritetaan DataMiningUI:n sisällä. Kaikki mitä tapahtuu hakemalla training-datan ja avaamalla sen sovelluksessa kuvataan alempana kuvassa 7. Kun arff-tiedosto on avattu FileChooserilla, se luetaan bufferiin ja asetetaan uuden instanssijoukon (import weka.core.Instances) trainingdata-nimisen muuttujan arvoksi. Koska instanssijoukosta on tiedettävä se attribuutti, jonka arvoa tullaan jatkossa ennustamaan, asetetaan ClassIndex training-datan viimeiseen attribuuttiin, joka on tässä tapauksessa age eli ikä. Weka API:n Instances-luokan metodeilla saadaan haettua training-datan sisältämien tietueiden eli kyseisten instanssien määrä (numInstances()) sekä attribuuttien määrä (numAttributes()).

Koska scoring-dataa haettaessa tehdään siihen liittyen täsmälleen samat vaiheet, sitä ei tässä sen enempää aukaista.

```
private void avaaTrainingActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int returnVal = fileChooser.showOpenDialog(this);
    if (returnVal == JFileChooser.APPROVE_OPTION) {
        trainingfile = fileChooser.getSelectedFile();
        trainingNimi.setText(trainingfile.getName());
        try {
            trainingdata = new Instances(new BufferedReader(new FileReader(trainingfile)));
        } catch (IOException ex) {
            Logger.getLogger(DataMiningUI.class.getName()).log(Level.SEVERE, null, ex);
        }
        trainingdata.setClassIndex(trainingdata.numAttributes() - 1);
        attribuutteja.setText(Integer.toString(trainingdata.numAttributes()));
        trainingTietueita.setText(Integer.toString(trainingdata.numInstances()));
        br = null;
        try {
            br = new BufferedReader(new FileReader(trainingfile));
        } catch (FileNotFoundException ex) {
            Logger.getLogger(DataMiningUI.class.getName()).log(Level.SEVERE, null, ex);
        }
        EsikatseluTraining.setText(null);
        text = null;
        try {
            while ((text = br.readLine()) != null) {
                EsikatseluTraining.append(text + "\n");
            }
        } catch (IOException ex) {
            Logger.getLogger(DataMiningUI.class.getName()).log(Level.SEVERE, null, ex);
        }
    } else {
        System.out.println("File access cancelled by user.");
    }
}
```

Kuva 7. Training-datan avaaminen käyttöliittymässä ja siitä seuraavat toimet.

Kun sekä training- että scoring-data on noudettu ja asetettu esikatseluun, valitaan louhintamenetelmä alasvetovalikosta kuvan 8 osoittamista vaihtoehdoista.



Kuva 8. Sovelluksen valikko tiedonlouhintamenetelmistä.

Tämä valikko on toteutettu ComboBox-valikkona, johon asetetaan vaihtoehdot ja annetaan niille numeeriset arvot nollasta eteenpäin. Tässä esimerkissä Valitse menetelmä –vaihtoehto on saanut arvoksensa 0 ja siitä eteenpäin Linear Regression 1 ja SMO reg 7. ComboBox-valikon toteuttamiseen käytettyä koodia esitellään kuvassa 9.

```
private void aloitaBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int i = comboBoxMenetelmat.getSelectedIndex();
    if(i == 1){
        controller = new DataMiningController(trainingdata, scoringdata);
        text2=controller.linearRegression();
        Tulokset.append("Linear Regression: " + text2 + " " + scoringdata.numInstances() + ":sta" + "\n");
    }
    if(i == 2){
        controller = new DataMiningController(trainingdata, scoringdata);
        text2=controller.multilayerPerceptron();
        Tulokset.append("Multilayer Perceptron: " + text2 + " " + scoringdata.numInstances() + ":sta" + "\n");
    }
    if(i == 3){
        controller = new DataMiningController(trainingdata, scoringdata);
        text2=controller.isotonicRegression();
        Tulokset.append("Isotonic Regression: " + text2 + " " + scoringdata.numInstances() + ":sta" + "\n");
    }
    if(i == 4){
        controller = new DataMiningController(trainingdata, scoringdata);
        text2=controller.gaussianProcesses();
        Tulokset.append("Gaussian Processes: " + text2 + " " + scoringdata.numInstances() + ":sta" + "\n");
    }
    if(i == 5){
        controller = new DataMiningController(trainingdata, scoringdata);
        text2=controller.paceRegression();
        Tulokset.append("Pace Regression: " + text2 + " " + scoringdata.numInstances() + ":sta" + "\n");
    }
    if(i == 6){
        controller = new DataMiningController(trainingdata, scoringdata);
        text2=controller.plsClassifier();
        Tulokset.append("PLS Classifier: " + text2 + " " + scoringdata.numInstances() + ":sta" + "\n");
    }
    if(i == 7){
        controller = new DataMiningController(trainingdata, scoringdata);
        text2=controller.smoReg();
        Tulokset.append("SMO reg: " + text2 + " " + scoringdata.numInstances() + ":sta" + "\n");
    }
}
```

Kuva 9. ComboBoxin indexien valintaa if-lauseilla.

Kun ComboBoxista on valittu jokin menetelmä, asetetaan sen menetelmän arvo valituksi id:ksi ja tämän jälkeen yksinkertaisilla if-lauseilla valitaan, millä menetelmällä seitsemästä training- sekä scoring-dataa lähdetään louhimaan eli mikä funktio viimeisen luokan DataMiningControllerin tarjoamasta toteutetaan, kun käyttöliittymässä painetaan painiketta Aloita. Kaikki funktiot palauttavat arvonaan stringin, joka kertoo, kuinka monta ennustetta on osunut oikeaan kyseisellä menetelmällä ja ne tulostetaan Tulos-tauluun, jotta niitä voi vertailla keskenään.

Kolmas luokka DataMiningController, jonka koodia louhintamenetelmälle Linear Regression esitellään kuvassa 10, on raskaampi ja sisältää suuren osan koko sovelluksen varsinaisesta toiminnallisuudesta.

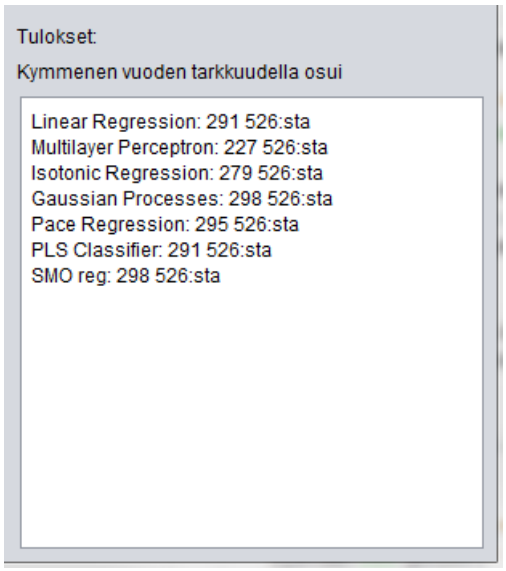
Kun DataMiningUI:n puolella valitaan louhintamenetelmäksi Linear Regression, kotrollerille lähetetään ladatut training- sekä scoring-data ja toteutetaan kontrollerin puolelta funktiota linearRegression(). Tämä funktio toimii niin, että aluksi ladataan tiedossa oleva vertailudata instanssiin comparedata. Tämä vertailudata on aineisto, joka sisältää scoring-datasta puuttuvat varsinaiset ikä-attribuutin arvot. Tällaisen aineiston tulee olla olemassa sitä varten, että voidaan todentaa, kuinka tarkkoja numeeriset ennusteet todella ovat ja kuinka oikeaan ne voivat osua. Compare-datalle tehdään samat toimenpiteet, kuin training- ja scoring-datalle aikaisemmin.

```
public String linearRegression() {
    laskuri=0;
    Instances comparedata = null;
    try {
        // TODO add your handling code here:

        comparedata = new Instances(new BufferedReader(new FileReader("C:\\Users\\Annu\\O
    } catch (IOException ex) {
        Logger.getLogger(DataMiningUI.class.getName()).log(Level.SEVERE, null, ex);
    }
    comparedata.setClassIndex(comparedata.numAttributes() - 1);
    LinearRegression lr = new LinearRegression();

    try {
        lr.buildClassifier(training);
    } catch (Exception ex) {
        Logger.getLogger(DataMiningUI.class.getName()).log(Level.SEVERE, null, ex);
    }
    System.out.print(lr + "\n");
    for (int i = 0; i < scoring.numInstances(); i++) {
        try {
            pred = lr.classifyInstance(scoring.instance(i));
        } catch (Exception ex) {
            Logger.getLogger(DataMiningUI.class.getName()).log(Level.SEVERE, null, ex);
        }

        age = comparedata.instance(i).value(30);
        if(age-pred>=-10 && age-pred<=0){
            laskuri++;
        }
        if(age-pred<=10 && age-pred>=0){
            laskuri++;
        }
        System.out.printf("%.0f", pred);
        System.out.print("   |   ");
        System.out.printf("%.0f", age);
        System.out.print("\n");
    }
    System.out.print(laskuri + " osui 10 vuoden tarkkuudella.");
    return Integer.toString(laskuri);
}
```

Kuva 10. LinearRegression-classifierin muodostaminen ja tulosten printtaaminen konsoliin.

Tässä kohtaa hyödynnetään Wekan tarjoamaa ohjelmointirajapintaa. Luodaan muuttuja lr, joka on tyyppiä LinearRegression (impoweka.classifiers.functions.LinearRegression). Muuttujan lr funktiota buildClassifier käytetään opettamaan ohjelmalle training-datan avulla, miten attribuuttien q1-q30 arvot vaikuttavat attribuutin age arvoon. Tämän jälkeen tulostetaan rakennettu kaava, jolla age-attribuutin tuloksia voidaan laskea q1-q30-attribuuttien perusteella. Sitten kyseistä kaavaa käytetään funktiolla classifyInstance aikaisemmin hakemaamme scoring-dataan. Samalla käydään läpi kaikki scoring-datan instanssit.

Funktio palauttaa tuloksen Stringinä, joka kertoo, kuinka monta ennustetta yhteensä kaikista instansseista osui 10 vuoden tarkkuudelle todellisesta iästä. Nämä tulokset esitellään käyttöliittymän puolella (kuva 11).



Kuva 11. Tulokset omassa ikkunassaan

Koska kaikkien louhintamenetelmien perusrakenne on samanlainen kuin Linear Regressionissa, niitä ei tässä erikseen esitellä.

4.7 Tulosten tulkinta

Jotta lukija ymmärtää, millä tavoin tehdyn tiedonlouhinnan tuloksia voidaan tulkita, tässä esitellään esimerkinomaisesti ohjelman laatijan tulkinta louhinnan tuloksista.

Kuvassa 12 esitellystä kaavasta näkee Linear Regressionin kannalta tärkeimmät kysymykset, jotka vaikuttavat vastaajaan ikään. Kun lähtöikä on 43,1351 vuotta ja katsotaan, mitä kukin vastaaja on vastannut esimerkiksi kysymykseen 9 (q9), huomataan, että tällä kysymyksellä on suuri merkitys henkilön iän arvioinnissa. Eli, mitä enemmän samaa mieltä vastaaja on väittämän ”Alkoholia tarjoavien ravintoloiden pitäisi saada olla auki nykyistä vapaammin.” kanssa, sitä nuoremmaksi hänen ikäennusteensä tipahtaa. Seuraavaksi vahvin painoarvo vastaajan iän kannalta on kysymyksellä 21 (q21) ” Homo- ja lesbopareilla pitää olla samat avioliitto- ja adoptio-oikeudet kuin heteropareilla.” Myös tämän kysymyksen kohdalla toimii periaate, että mitä enemmän on väittämän kannalla,

sitä nuorempi on. Vähiten merkitystä iän ennustamisessa on ollut kysymyksillä, jotka puuttuvat kaavasta eli niillä ei ole varsinaista painoarvoa. Tällaisia ovat olleet kysymykset Suomen velkaantumisesta, ydinvoimasta, Sote-uudistuksesta ja Natosta. Linear Regressionin avulla saatiin ennustettua 291 ikää 526:sta kymmenen vuoden tarkkuudella oikein.

```
Linear Regression Model

age =

     -1.4245 * q9 +
      1.2597 * q7 +
     -1.0166 * q6 +
      0.8442 * q5 +
     -1.2527 * q4 +
      0.8321 * q30 +
      0.5133 * q29 +
     -0.9048 * q28 +
      0.5035 * q24 +
      0.8226 * q22 +
     -1.3785 * q21 +
      0.9983 * q2 +
     -0.4306 * q16 +
     -0.4566 * q15 +
      1.1633 * q12 +
      0.7687 * q10 +
     43.1351
```

Kuva 12. Linear Regression kaava

Multilayer Perceptron on monivaiheisempi laskutoimitus, jossa lasketaan ensin yhden solmun (Node) sisällä jokaisen kysymyksen painoarvo kuvan 14 mukaisesti. Tässä tapauksessa solmuja on 15 ja lopulta jokaiselle solmulle on saatu laskettua myös oma painoarvo, jotka sitten yhteen laskemalla saadaan ennuste vastaajan iästä (kuva 13). Koska tämä menetelmän on monivaiheisempi ja vaatii enemmän laskuja, ohjelmalla kestää myös kauemmin ajaa se läpi. Lisäksi Multilayer Perceptronin avulla saatiin ennustettua vain 227 ikää 526:sta kymmenen vuoden tarkkuudella oikein. Tämä on kaikista menetelmistä huonoin määrällinen tulos.

```
Linear Node 0
    Inputs    Weights
    Threshold    -0.03226942525165664
    Node 1    -1.2460861193623423
    Node 2    0.4291053406964216
    Node 3    -1.335637763192456
    Node 4    -0.6395399156429182
    Node 5    1.7193002040200327
    Node 6    -1.1179897227710152
    Node 7    1.578995839639686
    Node 8    1.1329475992489937
    Node 9    -0.7942962515481948
    Node 10    -0.9677970744672743
    Node 11    -0.4355474774004282
    Node 12    2.1667123849758623
    Node 13    -1.3674498371706647
    Node 14    0.715313867452937
    Node 15    -1.2536804999951474
```

Kuva 13. Multilayer Perceptron solmut

```
Sigmoid Node 1
    Inputs    Weights
    Threshold    -5.838747780848905
    Attrib q9    0.2951574327698269
    Attrib q8    0.09682383574887504
    Attrib q7    3.0060792433180565
    Attrib q6    1.332752524755242
    Attrib q5    -4.070007550914141
    Attrib q4    4.262109939686646
    Attrib q30    0.9334620507200785
    Attrib q3    -0.9379030693556605
    Attrib q29    6.354064056292418
    Attrib q28    -2.491449468245702
    Attrib q27    3.2152166007874863
    Attrib q26    -1.7384165905910174
    Attrib q25    -5.146929869392292
    Attrib q24    -1.4339130205297523
    Attrib q23    3.2018641415954656
    Attrib q22    -0.012960962682888445
    Attrib q21    -2.967300869543191
    Attrib q20    3.0630170513420842
    Attrib q2    -0.021124642700541138
    Attrib q19    -4.524675452995906
    Attrib q18    1.9799690652934507
    Attrib q17    -2.147732815920681
    Attrib q16    -2.2820567395764613
    Attrib q15    -1.4971131361303551
    Attrib q14    -0.25219551437282245
    Attrib q13    0.3953743381481687
    Attrib q12    -0.46865265952453317
    Attrib q11    3.7676005775067276
    Attrib q10    4.647359652772454
    Attrib q1    -0.28658854936926065
```

Kuva 14. Multilayer Perceptron kysymysten painotukset solmussa 1

Kuten Linear Regressioinissa myös Isotonic Regressionissa (kuva 15) on annettu painoarvo kysymykselle 9 (q9). Kaikille vastausvaihtoehdoille yhdestä viiteen on luotu oma ennuste, joka vielä tuloksissa pyöristetään tasaluvuksi. Tämän seurauksena kysymyk-

seen 9 vastausvaihtoehdon yksi valinneiden on ennustettu olevan 52-vuotiaita, vaihtoehdon kaksi valinneiden 49-vuotiaita, vaihtoehdon kolme valinneiden 44-vuotiaita, vaihtoehdon neljä valinneiden 44-vuotiaita ja vaihtoehdon viisi valinneiden 39-vuotiaita. Eli samalla tavoin kuin Linear Regressionissa: mitä enemmän ollaan väittämän 9 kanssa samaa mieltä, sitä nuorempi on. Isotonic Regressiolla ohjelma ennusti kymmenen ikävuoden tarkkuudella 279 ikää oikein 526:sta.

```
Isotonic regression

Based on attribute: q9

prediction:      51.65          cut point:       1.5
prediction:      48.56          cut point:       2.5
prediction:      44.33          cut point:       3.5
prediction:      43.73          cut point:       4.5
prediction:      39.32
```

Kuva 15. Isotonic Regression kaava perustuen kysymykseen 9

Gaussian Processes käyttää laskukaavassaan matriiseja kuvan 16 mukaisella tavalla. Tällä kaavalla ohjelma ennusti kymmen vuoden tarkkuudella iän oikein 298:lle henkilölle 526:sta.

```
Gaussian Processes

Kernel used:
  RBF kernel: K(x,y) = e^-(1.0* <x-y,x-y>^2)

Average Target Value : 44.7256046705588
Inverted Covariance Matrix:
    Lowest Value = -0.226771437896869
    Highest Value = 0.7921186928276341
Inverted Covariance Matrix * Target-value Vector:
    Lowest Value = -19.85475205215417
    Highest Value = 24.119905574663054
```

Kuva 16. Gaussian Processes-kaava

Pace Regressionin laskukaavasta kuvassa 17 näkee taas, kuinka kysymyksillä 9 (q9) ja 21 (q21) on selkeästi suurin merkitys iän ennustamisessa. Lisäksi kysymys 4 (q4) Suomen luopumisesta kivihiilen, turpeen ja maakaasun käytöstä nousee esille tässä kaavassa. Arvion mukaan mitä enemmän samaa mieltä on väittämän kanssa, sitä nuorempi on. Pace Regressionilla ohjelma ennustaa 295 ikää 526:sta oikein kymmenen vuoden

tarkkuudella. Se on tämän menetelmän keveyden kannalta oikein hyvä tulos verrattuna muihin.

```
Pace Regression Model

age =

     43.2203 +
     -1.5449 * q9 +
      1.175  * q7 +
     -1.003  * q6 +
      0.4831 * q5 +
     -1.2713 * q4 +
      0.6003 * q30 +
      0.3204 * q29 +
     -0.6319 * q28 +
      0.2603 * q24 +
      0.4992 * q22 +
     -1.4064 * q21 +
      0.2371 * q20 +
      1.0181 * q2 +
      0.2494 * q19 +
      0.2749 * q17 +
     -0.0381 * q16 +
     -0.2249 * q15 +
      0.3202 * q13 +
      1.1107 * q12 +
      0.439  * q10
```

Kuva 17. Pace Regression kaava

Myös SMO regin käyttämä laskukaava (kuva 18) nostaa esille kysymykset 9 ja 21 eniten vaikuttavina tekijöinä iän ennustamisessa. Tällä menetelmällä ohjelma ennusti vastaajien iät oikein kymmenen vuoden tarkkuudella 298:henkilön kohdalla 526:sta.

```
SMOreg

weights (not support vectors):
 -       0.1203 * (normalized) q9
 -       0.0066 * (normalized) q8
 +       0.087  * (normalized) q7
 -       0.0867 * (normalized) q6
 +       0.0524 * (normalized) q5
 -       0.0897 * (normalized) q4
 +       0.0513 * (normalized) q30
 +       0.0089 * (normalized) q3
 +       0.0183 * (normalized) q29
 -       0.0485 * (normalized) q28
 -       0.0328 * (normalized) q27
 -       0.0167 * (normalized) q26
 +       0.0138 * (normalized) q25
 +       0.0429 * (normalized) q24
 -       0.0083 * (normalized) q23
 +       0.0715 * (normalized) q22
 -       0.1129 * (normalized) q21
 +       0.0444 * (normalized) q20
 +       0.0891 * (normalized) q2
 +       0.043  * (normalized) q19
 +       0.0336 * (normalized) q18
 +       0.0326 * (normalized) q17
 -       0.0217 * (normalized) q16
 -       0.0061 * (normalized) q15
 -       0.0137 * (normalized) q14
 +       0.0084 * (normalized) q13
 +       0.0656 * (normalized) q12
 +       0      * (normalized) q11
 +       0.0204 * (normalized) q10
 +       0.0074 * (normalized) q1
 +       0.3999
```

Kuva 18. SMO reg-kaava

5 Yhteenveto

Tämän työn tarkoitus oli esitellä lukijalle big data eli massadata ja sen louhinta sekä ohjeistaa, kuinka tiedonlouhintasovellus Wekan ohjelmointirajapintaa voi käyttää osana omaa koodia Java-sovelluksessa. Ohjelmointirajapinnan käyttöönoton esittelyä varten luotiin sovellus, jonka tarkoitus oli numeerisia ennusteita hyödyntäen ennakoida vastaajan ikä hänen vastaustensa perusteella ja vertailla erilaisia numeerisen ennusteen menetelmiä toisiinsa tuoden esille niiden vahvuutta ennustamisessa.

Työssä tulee ilmi, kuinka moniulotteinen termi big data on ja kuinka laajasti sitä voidaan ja kannattaa elämän eri osa-alueilla hyödyntää saavuttaakseen uutta tietoa ihmisistä ja heidän toiminnastaan, laitteista ja ympäristöstä sekä yleisesti syy-seuraussuhteista. Työssä esitellään haaste, jonka big data suurilla tietomäärillään sekä niiden jatkuvalla kasvulla ja monipuolistumisella asettaa yksityisille ihmisille, yrityksille ja yhteiskunnalle, sekä ratkaisut, joilla big data -analyysit sekä tiedonlouhinta vastaavat haasteeseen.

Tavoite opastaa lukijalle, kuinka Wekan ohjelmointirajapintaa voidaan käyttää omassa Java-sovelluksessa, toteutuu työssä. Yksinkertaisen numeerisia ennusteita käyttävän louhintasovelluksen rakentaminen ja siinä Weka API:n käyttöönotto on esitelty vaihe vaiheelta, niin että kuka tahansa pystyy tekemään saman. Koska Weka API:n käyttö vaatii myös yleistä ymmärrystä tiedonlouhinnasta, alussa luodun laajan teoriaosuuden merkitys tulee tässä kohtaa huomatuksi, kuten myös selkeä opastus datan valmistelusta tiedonlouhintaa varten.

Itse ohjelman tarkoitus kokeilla erilaisia numeerisia ennusteita ja vertailla niiden toimintaa toisiinsa totetutuu, kun jokaista menetelmää voidaan käyttää samaa aineistoa vasten ja sen jälkeen vertailla tuloksia keskenään. Mikään menetelmistä ei räikeästi esiintynyt edukseen ja kaikki menetelmät ennustivat suhteellisen epäluotettavasti vastaajan iän vastausten perusteella. Tämän päätellään johtuvan siitä, että analysoitavan aineiston koko oli kuitenkin hieman liian pieni saadakseen luotettavia tuloksia. Kuitenkin mielenkiintoisesti muutama kysymys pomppasi selkeästi esille ollen painoarvoltaan merkittävämpiä, kun ennustetaan vastaajan ikää. Näitä kysymyksiä olivat kysymys 9 (q9) liittyen ravintoloiden aukioloaikoihin, 21 (q21) liittyen homo- ja lesboliittoihin, 4 (q4) liittyen kivihiilen, turpeen ja maakaasun käyttöön ja 7 (q7) liittyen velvoitteeseen ottaa tarjottu työpaikka vastaan.

Lähdettäessä tekemään tiedonlouhintaa ei voida etukäteen tietää, minkälaista uutta tietoa sieltä paljastuu vai paljastuuko mitään. Tämän työn tiedonlouhintaesimerkin tarkoituksena ei ollut löytää uutta tietoa, vaan toimia yksinkertaisena ja selkeänä esimerkkinä lukijalle, miten tiedonlouhintaa tehdään ja minkälaisin välinein, erityisesti opastaen Wekan API:n käytössä.

Lähteet

Boyd, Danah & Crawford, Kate (2012) CRITICAL QUESTIONS FOR BIG DATA, *Information, Communication & Society*, 15:5, sivut 662-679. Verkkodokumentti. <http://dx.doi.org/10.1080/1369118X.2012.678878> (Luettu 19.1.2016).

Chessell, Mandy (2014) Ethics for big data and analytics. Verkkodokumentti. <http://www.ibmbigdatahub.com/sites/default/files/whitepapers_reports_file/TCG%20Study%20Report%20-%20Ethics%20for%20BD%26A.pdf> (Luettu 19.1.2016).

Crivat, Bogdan, MacLennan, Jamie & Tang, ZhaoHui (2009) Data mining with Microsoft SQL server 2008. Indianapolis, IN: Wiley Publishing, Inc.

Cukier, Kenneth & Mayer-Schönberger, Viktor (2013) Big data. A revolution that will transform how we live, work and think. London: John Murray.

Davenport, Thomas H. (2014) Big data at work: dispelling the myths, uncovering the opportunities. Boston, Massachusetts: Harvard Business School Publishing Corporation.

Davis (päiväämätön), Richard A. Gaussian processes. <http://www.stat.columbia.edu/~rdavis/papers/VAG002.pdf> Verkkodokumentti. (Luettu 23.5.2016).

Frank, Eibe, Hall, Mark A. & Witten, Ian H. (2011) Data mining: practical machine learning tools and techniques. Burlington, MA: Morgan Kaufmann Publishers.

Fricke Torsten & Novak Ulrich (2015) Tapaus Google. Munchen: F.A. Herbig Verlagsbuchhandlung GmbH.

Halper, Fern; Hurwitz, Judith & Nugent, Alan (2013) Big Data For Dummies. Hoboken, New Jersey: John Wiley & Sons, Inc.

Han, Jiawei; Kamber, Micheline & Pei, Jian (2012) Data Mining: concepts and techniques. Waltham, MA: Morgan Kaufmann Publishers.

Jeskanen, Elina (2015) Miten tiedon visualisoinnilla parannetaan terveydenhoitoa? Verkkodokumentti. <https://www.cgi.fi/blogi/miten-tiedon-visualisoinnilla-parannetaan-terveydenhoitoa> (Luettu 26.1.2016).

King, Jonathan H. & Richards, Neil M. (2014a) Big Data Ethics, *Wake Forest Law Review*, 49, sivut 393-432. Verkkodokumentti. <http://ssrn.com/abstract=2384174> (Luettu 19.1.2016).

King, Jonathan H. & Richards, Neil M. (2014b) Big Data and the Future for Privacy. Verkkodokumentti. <http://ssrn.com/abstract=2512069> (Luettu 19.1.2016).

Knight, Meredith (2015a) Healthcare Dives into Big Data. Verkkodokumentti. <http://usa.healthcare.siemens.com/news-and-events/mso-big-data-and-healthcare-1> (Luettu 13.1.2016).

Knight, Meredith (2015b) Smart Use of Big Data: The Key to the Future. Verkkodokumentti. <http://www.healthcare.siemens.se/news-and-events/mso-big-data-and-healthcare-2> (Luettu 13.1.2016).

Macrae, Duncan (2015) How Big Data Is Changing The Sports Industry. Verkkodokumentti. <http://www.techweekeurope.co.uk/data-storage/how-big-data-is-changing-the-sports-industry-182365> (Luettu 21.3.2016).

Marr, Bernard (2015a) How Big Data Is Changing Healthcare. Verkkodokumentti. <http://www.forbes.com/sites/bernardmarr/2015/04/21/how-big-data-is-changing-healthcare/#2715e4857a0b31e4987a32d9> (Luettu 13.1.2016).

Marr, Bernard (2015b) Apple & IBM Team Up For New Big Data Health Platform. Verkkodokumentti. <https://www.linkedin.com/pulse/apple-ibm-team-up-new-big-data-health-platform-bernard-marr?trk=mp-reader-card> (Luettu 13.1.2016).

Marr, Bernard (2015c) 4 Ways Big Data Is Transforming Healthcare. Verkkodokumentti. <http://www.datasciencecentral.com/profiles/blogs/4-ways-big-data-is-transforming-healthcare> (Luettu 13.1.2016).

Marr, Bernard (2015d) Big Data: The Winning Fromula in Sports. Verkkodokumentti. <http://www.forbes.com/sites/bernardmarr/2015/03/25/big-data-the-winning-formula-in-sports/#2ad77aef26dc> (Luettu 21.3.2016).

Montgomery, Douglas C., Peck, Elizabeth A. & Vining, G. Geoffrey (2012) Introduction to Linear Regression Analysis. Hoboken, New Jersey: John Wiley & Sons, Inc.

Neuro AI (2013 Backpropagation. Verkkodokumentti. <http://www.learnartificialneuralnetworks.com/backpropagation.html> (Luettu 23.5.2016).

North, Matthew A. (2012) Data Mining for the Masses. Lexington, KY: A Global Text Project Book.

Pentland, Alex (2014) Sosiaalifysiikka: Miten hyvät ideat leviävät: uuden tieteenalan opetuksia. Helsinki: Terra Cognita.

Pittsburgh Health Data Alliance (2016) Verkkodokumentti. <https://healthdataalliance.com/ > (Luettu 13.2.2016).

Platt, John C. (1998) Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Verkkodokumentti. <http://research.microsoft.com/pubs/69644/tr-98-14.pdf > (Luettu 23.5.2016).

Price, Gary (2014) How Large is the Digital Universe? How fast is It Growing? Verkkodokumentti. <http://www.infodocket.com/2014/04/16/how-large-is-the-digital-universe-how-fast-is-it-growing-2014-emc-digital-universe-study-now-available> (Luettu 19.1.2016).

Salo, Immo (2013) Big data – Tiedon vallankumous. Jyväskylä: Docendo Oy.

Salo, Immo (2014) Big Data & pilvipalvelut. Jyväskylä: Docendo Oy.

van Rijmenam, Mark (2014) The Los Angeles Police Department Is Predicting and Fighting Crime With Big Data. Verkkodokumentti. < https://datafloq.com/read/los-angeles-police-department-predicts-fights-crim/279 > (Luettu 16.3.2016).

Liite 1. Helsingin Sanomien vaalikoneen kysymykset q1-q30

Kaikissa kysymyksissä vastausvaihtoehdot olivat 1-5, 1:n ollessa Täysin eri mieltä, 3:n En osaa sanoa ja 5:n ollessa Täysin samaa mieltä.

1. Suomen velkaantuminen on käännettävä laskuun vaikka se samalla tarkoittaisi leikkauksia palveluihin ja etuuksiin.

2. Hyvin ansaitsevien palkkaverotusta pitäisi kiristää.

3. Suomi tarvitsee lisää ydinvoimaa.

4. Suomen tulee luopua kivihiilen, turpeen ja maakaasun käytöstä vuoteen 2025 mennessä.

5. Suomen julkinen sektori on liian suuri ja sitä on syytä pienentää.

6. Ruotsin pakollisesta opiskelusta tulisi luopua.

7. Velvoitteita ottaa vastaan tarjottu työpaikka pitäisi kiristää.

8. Jos ihmisellä on varallisuutta, hänen omaisuuttaan pitäisi vanhana käyttää hoivapalveluiden kustantamiseksi.

9. Alkoholia tarjoavien ravintoloiden pitäisi saada olla auki nykyistä vapaammin.

10. Kuntien määrää tulee vähentää merkittävästi, vaikka pakolla.

11. Yritysten pitäisi voida maksaa työehtosopimuksia pienempiä palkkoja, jotta Suomeen saataisiin työpaikkoja.

12. Koko Suomi on syytä pitää asuttuna ja valtion tuettava tätä verovaroin.

13. Sote-uudistuksen yhteydessä yksityiset terveyspalvelut pitäisi nostaa julkisten palveluiden rinnalle samanlaiseen asemaan.

14. Kuntien pitää saada itse tuottaa sosiaali- ja terveyspalveluita, eikä sote-uudistuksessa niiltä saa viedä liikaa pois valtaa.

15. Taksien pitäisi antaa kilpailla vapaasti asiakkaista ilman että valtio säätää hinnat tai rajoittaa taksilupien määriä.

16. EU:sta on Suomelle enemmän hyötyä kuin haittaa.

17. Hallitus päätti, että Suomi jää ulos osan EU-maiden valmistelemasta rahoitusmarkkinaverosta. Suomen pitäisi mennä mukaan rahoitusmarkkinaveroon.

18. Suomen tulisi tällä vaalikaudella ryhtyä valmistelemaan hakemista Natoon.

19. Puolustusvoimille on annettava nykyistä enemmän rahaa.

20. EU- ja ETA-alueen ulkopuolelta tulevien kohdalla käytetään nyt "tarveharkintaa" eli työlupien saantia rajoitetaan. Tästä on pidettävä kiinni, eikä työperäistä maahanmuuttoa pidä helpottaa.

21. Homo- ja lesbopareilla pitää olla samat avioliitto- ja adoptio-oikeudet kuin heteropareilla.

22. Jos valtio tarjoaa turvapaikanhakijoiden vastaanottokeskuksen perustamista kotikuntaani, tarjous pitää hyväksyä.

23. Kouluissa kohdellaan koululaisia liian lepsusti. Tiukempi kuri tekisi kouluista parempia.

24. Perinteiset arvot - kuten koti, uskonto ja isänmaa - muodostavat hyvän arvopohjan politiikalle.

25. Julkisia palveluita tulisi ulkoistaa entistä enemmän yksityisten yritysten tuotettavaksi.

26. Jos tulee eteen tilanne, jossa on välttämätöntä joko leikata julkisia palveluita ja sosiaalietuuksia tai korottaa veroja, veronkorotukset ovat parempi vaihtoehto.

27. Suuret tuloerot ovat hyväksyttäviä, jotta erot ihmisten lahjakkuudessa ja ahkeruudessa voidaan palkita.

28. Nykyisen kaltaiset palvelut ja sosiaalietuudet ovat pitemmän päälle liian raskaita julkiselle taloudelle.

29. Talouskasvu ja työpaikkojen luominen tulisi asettaa ympäristöasioiden edelle, silloin kun nämä kaksi ovat keskenään ristiriidassa.

30. Kaikessa päätöksenteossa pitäisi arvioida vaikutukset ympäristöön ja tarvittaessa luopua ympäristölle haitallisista hankkeis