LAPPEENRANNAN-LAHDEN TEKNILLINEN YLIOPISTO LUT
School of Engineering Science
Tuotantotalous

Aaro Hassinen

Ohjelmistoekosysteemin ISV-kumppaneiden klusterointi ja priorisointi

Diplomityö

Tarkastajat: Professori Timo Kärri
TkT Antti Ylä-Kujala

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT
School of Engineering Science
Tuotantotalous

Aaro Hassinen

Ohjelmistoekosysteemin ISV-kumppaneiden klusterointi ja priorisointi

Diplomityö
2021
85 sivua, 21 kuvaa, 5 taulukkoa ja 3 liitettä
Tarkastajat: Professori, TkT Timo Kärri ja tutkijatohtori, TkT Antti Ylä-Kujala

Hakusanat: Ohjelmistoekosysteemi, ISV-kumppani, koneoppiminen, klusterointi, usean kriteerin päätösanalyysi

Nykypäivän kovasti kilpailulla ohjelmistokehityksen markkinoilla asiakkaat vaativat yrityksiltä entistä enemmän personointia, kätevyyttä ja hyviä kokemuksia. Näihin vaatimuksiin ohjelmistoja tarjoavan yrityksen on helpompi vastata kumppanuuksien avulla. ISV-kumppanien avulla ohjelmistoekosysteemi voi kehittää kattavammin ekosysteemin ominaisuuksia ja palveluita. ISV-kumppanit, eli Independent Software Vendor-kumppanit, ovat kolmannen osapuolen ohjelmistokehittäjiä, jotka hyödyntävät rajapintoja uusien ominaisuuksien ja palveluiden kehittämisessä. Kumppanien avulla ekosysteemin ohjelmistoyritys voi paremmin keskittyä heidän ydinliiketoimintaansa ja saada tukea asiakkaiden toiveiden vastaamiseen kumppaneilta.

Tämän diplomityön tarkoituksen on ollut luoda menetelmä, jonka avulla ohjelmistoekosysteemin ISV-kumppaneita voidaan priorisoida ja klusteroida helpommin hallinnoitaviin joukkoihin. Työn tarkoituksena on tehostaa ohjelmistoekosysteemin ISV-kumppanien hallintaa ja selvittää, miten ISV-kumppaneita voidaan verrata toisiinsa ja määritellä millainen on hyvä ISV-kumppani. Työtä varten on tehty kattava kirjallisuuskatsaus, jossa perehdytään ohjelmistoekosysteemeihin ja ISV-kumppanien valintakriteereihin. Kirjallisuuskatsauksessa syvennytään myös erilaisiin koneoppimisen menetelmiin ja erityisesti klusterointiin ja data-analyysin prosessiin DIKW-pyramidin avulla.

Tämän työn lopussa esitellään data-analyysin prosessi, jossa ISV-kumppaneista kerätystä datasta luodaan usean kriteerin päätösanalyysillä parhaimmuusjärjestys, jota voidaan käyttää priorisoinnin työkaluna ja työn lopussa esitellään menetelmä, jossa hyödynnetään koneoppimista ISV-kumppanien klusteroinnissa. Tämän työn menetelmien kehittämisessä hyödynnettiin Python-ohjelmointikieltä ja sen useita kirjastoja, joiden avulla dataa ISV-kumppaneista yhdistettiin ja esikäsiteltiin. Python-ohjelmointikielen avulla suoritettiin myös ISV-kumppaneiden klusteroiminen K-means algoritmin avulla, sekä usean kriteerin päätösanalyysin SAW-menetelmä ohjelmoitiin ja toteutettiin Python-ohjelmointikielen avulla.

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Industrial Engineering and Management

Aaro Hassinen

Prioritizing and clustering ISV-partners in Software Ecosystem

Master's thesis
2021
85 pages, 21 figures, 5 tables and 3 appendices
Examiners: Professor, D.Sc. (Tech). Timo Kärri and Post-Doctoral Researcher, D.Sc. (Tech.) Antti Ylä-Kujala

Keywords: software ecosystem, ISV-partner, machine learning, clustering, multi-criteria decision analysis, MCDA, Independent software vendor

In today's highly competitive software development market, customers are demanding more personalization, convenience, and good experiences from companies. These requirements are easier for a software company to meet through partnerships. With the help of ISV partners, the software ecosystem can more comprehensively develop ecosystem features and services. ISV Partners, or Independent Software Vendor Partners, are third-party software developers who take advantage of interfaces to develop new features and services. With the help of partners, an ecosystem software company can better focus on their core business.

The purpose of this thesis has been to create a method by which ISV partners in a software ecosystem can be prioritized and clustered into more manageable sets of ISV partners. The purpose of this thesis is to improve the management of ISV partners in the software ecosystem and to find out how ISV partners can be compared with each other and to determine what makes a good ISV partner. A comprehensive literature review has been conducted for the thesis, which examines software ecosystems and selection criteria for ISV partners. The literature review also delves into various machine learning methods and in particular the process of clustering and data analysis using the DIKW pyramid.

At the end of this work, a process of data analysis is presented, in which data collected from ISV partners is used to create a ranking order by multi-criteria decision analysis that can be used as a prioritization tool, and at the end, a method is utilized to utilize machine learning in ISV partner clustering. The methods of this work were developed using the Python programming language and its several libraries to combine and pre-process data about ISV-partners. The Python programming language was also used to cluster ISV partners using the K-means algorithm, and the SAW method for multi-criteria decision analysis was programmed and implemented using the Python programming language.

ALKUSANAT

Opinnot LUT-yliopistossa ovat olleet ikimuistoiset ja ne alkavat olla takana tämän diplomityön valmistuttua. Haluan kiittää Vismaa tämän diplomityön aiheesta. Kiitokset kaikille Vismalaisille, ketkä ovat olleet työn toteutuksessa mukana ja erityisesti kiitokset Jouni Tiaiselle työn ohjaamisesta. Diplomityö oli haastava, mutta mielenkiintoinen ja pitkä projekti. Pääsin oppimaan valtavasti uutta ja näillä eväillä on hyvä jatkaa eteenpäin.

Lopuksi vielä kiitos kaikille ystäville ja perheelle tuesta koko opintojen aikana.

Lappeenranta 25.9.2021
Aaro Hassinen

SISÄLLYSLUETTELO

1 JOHDANTO 3

1.1 Työn tausta 3

1.2 Tavoitteet ja rajaus 4

1.3 Työn menetelmät 5

1.4 Työn rakenne 5

2 OHJELMISTOLIIKETOIMINTA JA -EKOSYSTEEMI 6

2.1 ISV-Kumppaniohjelma ja sen hyödyt 11

2.2 ISV-kumppanien valintakriteerit 14

2.3 ISV-kumppanien priorisointi 17

3 DATAN JALOSTAMINEN 19

3.1 Data-analytiikka 19

3.2 Datasta viisautta 21

4 KONEOPPIMINEN 26

4.1 Ohjattu oppiminen 26

4.2 Ohjaamaton oppiminen 27

4.3 Osittain ohjattu oppiminen ja vahvistettu oppiminen 29

5 USEAN KRITEERIN PÄÄTÖSANALYYSI JA SEN MENETELMÄT 31

5.1 Painotettu summa -malli 32

5.2 Analyyttinen hierarkiaprosessi 33

5.3 TOPSIS 34

5.4 Promethee 35

6 VISMA SOLUTIONS OY 36

6.1 ISV-kumppaniohjelma Vismassa 37

6.2 Netvisor ja integraatio rajapinnat 38

7 DATA-ANALYYSIN PROSESSI 41
7.1 Datan kerääminen ja esikäsittely 42
7.2 Datan esittely 47
7.3 ISV-kumppanien klusterointi 54
7.4 ISV-kumppanien priorisointi 62
8 TULOKSET JA NIIDEN ARVIOINTI 67
8.1 Työn keskeiset tulokset 67
8.2 Tulosten arviointi 70
8.3 Jatkotoimenpiteet ja suositukset 71
9 YHTEENVETO 73
Lähteet 74

Liite 1. Katsaus partnerin valintakriteereistä
Liite 2. Tulostaulukko osa 1.
Liite 3. Tulostaulukko osa 2.

1 JOHDANTO

Saatavilla olevan datan määrä kasvaa eksponentiaalista vauhtia ja globaalisti käytetyn datamäärän odotetaan kasvavan lähes 300 % vuodesta 2020 vuoteen 2025 mennessä, eikä kasvu näytä hidastuvan (Holst, 2021). Kasvavan datan määrän ansiosta dataohjautuva päätöksenteko on tullut suositummaksi, ja sen tarkoituksena on hyödyntää saatavilla olevaa dataa päätöksenteon tukena. Dataohjautuvan päätöksenteon avulla voidaan tehdä luottavampia päätöksiä nopeammin tehostaen perinteisiä päätöksenteon prosesseja. Tämän työn tarkoituksena on hyödyntää saatavilla olevaa dataa ohjelmistoekosysteemin ISV-kumppaneista ja koneoppimisen avulla luoda klustereita näistä ISV-kumppaneista ja selvittää ISV-kumppanien parhaimmuusjärjestys, jonka avulla niitä voidaan priorisoida. ISV lyhenne tulee sanoista Independent Software Vendor ja tarkoittaa itsenäistä ohjelmisto toimittajaa. (Stobierski, 2021)

EY:n 2020 tekemän tutkimuksen mukaan yli kaksi kolmasosaa tutkimukseen osallistuneista yrityksistä sanovat, että ekosysteemit ja kumppanuudet ovat ainut mahdollinen tapa menestyä nykypäivänä. Asiakkaat vaativat entistä enemmän personointia, kätevyyttä ja hyviä kokemuksia. Yrityksillä on haasteena vastata kaikkiin asiakkaiden tarpeisiin, mutta kumppanuuksien avulla näihin tarpeisiin voidaan vastata. (Higgins, 2020)

1.1 Työn tausta

Työ on tehty yhteistyössä Visma Solutions Oy:n kanssa (myöhemmin Case-yritys). ISV-kumppaneilla case-yrityksen tapauksessa viitataan yhteistyökumppaneihin, jotka kehittävät omia ohjelmistoja ja ne liitetään case-yrityksen tuotteisiin rajapintojen avulla. ISV-kumppaneita voi olla myös sellaiset yhteistyökumppanit, jotka kehittävät uutta liiketoimintaa case-yrityksen rajapintojen avulla. PAM:ien, eli Partner account managerien työtehtävänä on hallinnoida suhteita kumppaneiden kanssa. Kumppanien määrän kasvaessa ja heidän tukitarpeiden lisääntyessä olisi tärkeä pystyä määrittämään tärkeimmät kumppanit. Tehostamalla tärkeimpien kumppanien määrittämistä, voitaisiin paremmin arvioida nykyisten ja tulevien kumppanien arvoa, sekä pystyttäisiin määritellä kumppanit, joiden kanssa yhteistyöstä olisi mahdollista saada eniten lisäarvoa, esimerkiksi uusien asiakkaiden muodossa.

Tehokkaamman kumppanien arvon määrityksen avulla pystyttäisiin myös kohdentamaan tärkeimmille kumppaneille enemmän resursseja ja tämän avulla varmistamaan heidän tyytyväisyytensä ja parantamaan heidän sitoutumistansa.

1.2 Tavoitteet ja rajaus

Työn tavoitteena on suunnitella menetelmä, jonka avulla voidaan automaattisesti määritellä tärkeimmät ISV-kumppanit. Tavoitteena on myös haastaa nykyistä case-yrityksen kumppanitasoajattelua etsimällä vaihtoehtoisia tapoja ryhmitellä nykyisiä ISV-kumppaneita esimerkiksi koneoppimisen avulla. Työn tuloksien avulla pystytään arvioimaan nykyisten kumppanien arvoa ja vertailemaan nykyisiä kumppaneita keskenään. Tutkimuksen tuloksien perusteella voitaisiin myös arvioida uusien kumppanien arvoa jo hakuprosessin aikana. Työ rajataan käsittelemään vain yhtä case-yrityksen tuotteen, Netvisorin ISV-kumppaneita, koska saatavilla olevassa datassa on merkittäviä eroja eri tuotteiden välillä ja tämän työn puitteissa ei ole mahdollista ottaa käsittelyyn kaikkien tuotteiden ISV-kumppaneita. Työssä hyödynnetään case-yrityksen eli Visma Solutions Oy:n dataa, mahdollisista yrityssalaisuuksista johtuen työn lopputulokset ovat anonymisoidut. Tämän työn tutkimuskysymykset ovat seuraavat:

1. Millainen on hyvä ISV-kumppanit?
2. Miten nykyiset ISV-kumppanitasot voidaan jakaa uudelleen koneoppimisen avulla?
3. Miten määritellä ISV-kumppaneille tärkeysjärjestys?

Ensimmäinen tutkimuskysymys luo perustaa tämän työn toiselle ja kolmannelle tutkimuskysymyksille. Ensimmäisessä tutkimuskysymyksessä selvitetään, miten kirjallisuudessa määritellään hyvä ISV-kumppani ja mitä asioita liittyy hyvän ISV-kumppanin valintaan. Tarkoituksena on erityisesti löytää kriteereitä, joiden avulla voidaan määritellä, kuinka hyvä tietty ISV-kumppani on. Toisessa tutkimuskysymyksessä perehdytään case-yrityksen nykyisiin ISV-kumppanitasoihin ja klusteroidaan koneoppimista hyödyntämällä nykyiset ISV-kumppanit heistä saatavilla olevan datan avulla ja selvitetään, olisiko nykyisiä ISV-kumppanitasoja mahdollista jakaa useampiin tasoihin. Hypoteesina on, että ISV-kumppanien jakaminen useampaan partneritasoon voisi helpottaa tärkeysjärjestyksen luomista ja helpottaa partnerien hallinnoimista. Kolmannessa tutkimuskysymyksessä luodaan ISV-

kumppaneista ensimmäisen tutkimuskysymyksen tuloksien pohjalta ISV-kumppanien tärkeysjärjestys. Hypoteesina on, että ISV-kumppanit voidaan laittaa tärkeysjärjestykseen heistä saatavilla olevan datan avulla ja sitä tietoa voidaan hyödyntää, kun priorisoidaan ISV-kumppaneita.

1.3 Työn menetelmät

Työssä käytetyt tutkimusmenetelmät ovat kirjallisuuskatsaus ja kvantitatiivinen, eli määrällinen tutkimus, jota hyödynnettiin lähestymistapana tutkimuskysymyksien ratkaisemiseen. Kirjallisuuskatsauksessa hyödynnetään mahdollisimman paljon akateemisia lähteitä eri tietokannoista. Tietolähde hakuihin hyödynnetään LUT-tiedekirjaston tietokantahakua ja Google Scholarin hakutoimintoa. Kirjallisuuskatsauksessa hyödynnetään myös avoimia WWW-lähteitä. Kirjallisuuskatsauksen tavoitteena on saada mahdollisimman laaja kuva ja ymmärrys työn aihealueesta empiriaosuuden tueksi. Työssä hyödynnetään myös laskennallisia ja tilastollisia menetelmiä lopputuloksien saavuttamiseksi. Laskennalliset ja tilastolliset menetelmät toteutetaan Python-ohjelmointikielen avulla. Laskennallisia ja tilastollisia menetelmiä varten hyödynnettiin case-yritykseltä saatua ja kerättyä dataa ISV-kumppaneista.

1.4 Työn rakenne

Työ on jaettu kahteen isompaan osakokonaisuuteen, teorialukuihin ja soveltaviin lukuihin. Teorialuvuissa kuvataan työn aihepiiriin liittyvää teoriaa työn soveltavan osuuden tueksi. Teorialuvut on jaettua teorioiden aihealueittain omiksi luvuikseen. Luvut 2–6 ovat työn teorialukuja. Luvussa 6 on käsitelty tarkemmin case- yrityksen, eli case-yrityksen nykytilannetta työn aihealueen suhteen. Soveltavassa luvussa 7 esitetään, kuinka työn ongelmat saadaan ratkaistuiksi. Luvussa 7 yhdistetään kirjallisuuskatsauksen löydöksiä käytäntöön hyödyntämällä case-yritykseltä saatavissa olevaan dataan ISV-kumppaneista. Luvun 7 jälkeen esitetään työn tulokset ja niiden arviointi. Työn tuloksien jälkeen esitetään mahdolliset jatkotoimenpiteet ja kehitysehdotukset. Työn luvussa 8 käydään läpi työn tulokset ja niiden arviointi ja työn viimeisessä luvussa 9 on työn yhteenveto.

2 OHJELMISTOLIIKETOIMINTA JA -EKOSYSTEEMI

Tässä luvussa käsitellään lyhyesti ohjelmistoliiketoimintaa ja sen toiminta periaatteita erityisesti Software as a Service liiketoiminnassa. Tämän jälkeen syvennytään ohjelmistoekosysteemiin ja sen tuomiin etuihin. Tämän luvun alaluvuissa perehdytään ohjelmistoekosysteemeihin liittyviin ISV-kumppaneihin, jotka ovat tärkeä osa tätä työtä.

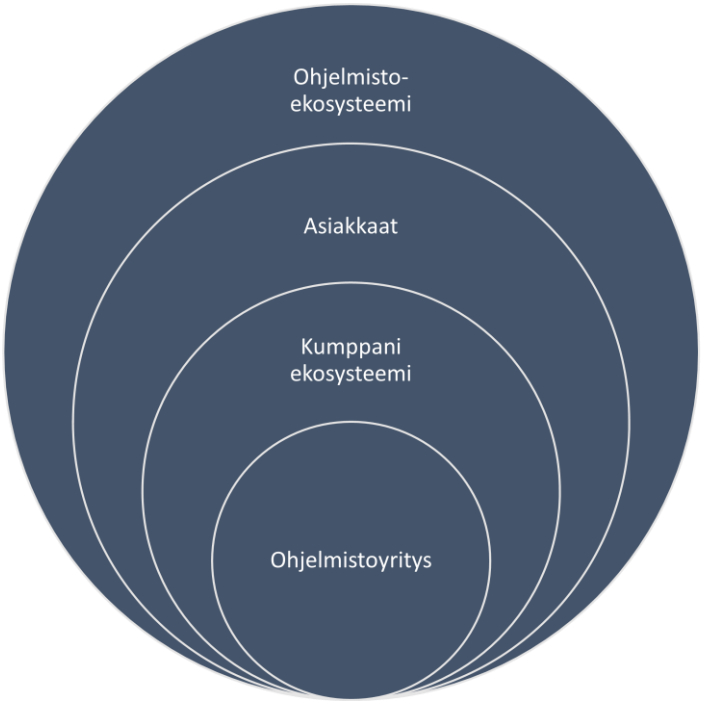
Ohjelmistoliiketoiminnalla tarkoitetaan liiketoimintaa, joka keskittyy ohjelmistojen kehittämiseen ja niiden myyntiin lisenssillä tai palveluna (Nousiainen, 2018 s. 11). Hoch et. al. (1999 s.27) mukaan ohjelmistotuotteet voidaan jakaa kuluttajille ja yrityksille suunnattuihin ohjelmistoihin. Kuluttajille suunnatut ohjelmistot ovat esimerkiksi käyttöjärjestelmät ja videopelit ja yrityksille suunnatut ohjelmistot ovat esimerkiksi toiminnanohjaus ja varastonhallinta ohjelmistot. (Hoch et. al. 1999 s.32–34)

Software as a Service (SaaS), eli ohjelmisto palveluna. SaaS on ohjelmiston jakelumalli, jossa sovellusta hallinnoi sen palveluntarjoaja. Palveluntarjoaja on vastuussa muun muassa sovelluksen infrastruktuurista, toiminallisuudesta ja sen ylläpidosta. Tällä tavoin asiakkaan ei tarvitse itse asentaa sovellusta erikseen tai ylläpitää sovellusta erillisellä palvelimella, koska sovelluksen palveluntarjoaja hoitaa sovelluksen tarjoamisen pilvipalvelimien avulla. Yleisesti asiakas maksaa SaaS palvelusta kiinteää kuukausihintaa tai käyttäjäkohtaisen hinnan perusteella, SaaS palveluissa ei yleensä myöskään ole kiinteitä aloituskustannuksia. SaaS -mallista hyötyvät merkittävästi palveluntarjoaja ja asiakas. Asiakkaalla on aina käytössään tuotteen uusin version, joka samalla helpottaa palveluntarjoajan ylläpitoa. Pilvipalvelut ovat myös helposti skaalattavissa isommille asiakasmäärille, joka helpottaa palveluntarjoajan liiketoiminnan kehittymistä. Asiakkaat voivat käyttää SaaS ohjelmistoa helposti verkkoselaimen välityksellä ja käytettävissä missä tahansa. (Weiping, 2009 s. 110; Ma, 2007; Yiu, 2020)

Popp ja Meyer määrittelevät ohjelmistoekosysteemin olevan ekonominen ekosysteemi, joka keskittyy yhden ohjelmistotuottajan ympärille ja heidän mukaansa aiemmat tutkimukset ovat tuoneet ilmi sen, että nopeasti kehittyvä ohjelmistoala on vienyt kilpailun siihen, että menestys määrittyy koko ekosysteemin menestyksen perusteella, eikä pelkän ohjelmistoyrityksen

menestyksen perusteella (Popp & Meyer, 2010, s. 131). Jansen ja Cusumano määrittelevät ohjelmistoekosysteemin olevan joukko toimijoita, jotka toimivat yhdessä ja näitä toimijoita yhdistää teknologinen alusta tai markkina ja nämä toimijat työskentelevät jakamalla informaatiota, resursseja ja heidän tuloksiansa. (Jansen & Cusumano, 2013)

Ohjelmistoekosysteemit kehittyvät ajan myötä yhden ison ohjelmistotuottajan ympärille, joka kehittää omaa ohjelmistoalustaa. Ohjelmistoekosysteemissä kaikki osapuolet hyötyvät, koska se mahdollistaa myös pienempien toimijoiden pääsyn isoille markkinoille. Isoimman hyödyn ohjelmistoekosysteemistä kerää ekosysteemiä hallinnoiva taho, joka yleensä on ohjelmistoyritys, jonka ympärille ekosysteemi on rakentunut. (Kaistinen, 2017 s. 27–28; Jansen & Cusumano, 2013)



Kuva 1. Ohjelmistoekosysteemi

Kuvassa 1 on havainnollistettu ohjelmistoekosysteemin rakennetta. Kuten aiemmin mainittu, ohjelmistosekosysteemin ytimessä on itse ohjelmistoyritys, jonka ohjelmistoalusta on ohjelmistoekosysteemin käytössä. Ohjelmistoyrityksen ympärillä on kumppaniekosysteemi, joka pitää sisällään lisäarvoa tarjoavat jälleenmyyjät, lisäarvoa tarjoavat jakelijat ja järjestelmä

integraattorit. Kumppaniekosysteemin rooli on toimia esimerkiksi jälleenmyyjinä, solmia yhteisiä markkinointikampanjoita ohjelmistoyrityksen kanssa, täydentää ohjelmistoekosysteemin tarjontaa omilla ohjelmistoratkaisuilla tai rakentaa integraatioita toisiin ohjelmistoihin, joita asiakkaat käyttävät. (Bech, 2015 s.129–133; Popp & Meyer, 2010, s. 153; Kaistinen, 2017 s. 27, 33–35)

Ohjelmistoekosysteemi tuo mukanaan useita hyötyjä, jotka näkyvät kaikille ekosysteemin toimijoille. Loppukäyttäjät esimerkiksi hyötyvät paremmasta tarjoamasta ydintuotteen ympärillä, he pystyvät esimerkiksi tehostamaan omia prosessejaan ohjelmistoekosysteemin tuomien integraatioiden avulla. Ohjelmistoekosysteemi houkuttelee myös samalla uusia asiakkaita ja parantaa asiakkaiden sitoutuneisuutta, koska laajalti yrityksessä käyttöön otetusta ohjelmistosta siirtyminen toiseen, esimerkiksi kilpailijan ohjelmistoon on erittäin työlästä. Ohjelmistoyritys pystyy paremmin keskittymään ydintuotteen kehittämiseen, sillä välin kun kumppaniekosysteemi tarjoaa loppukäyttäjille ekosysteemin ydintuottetta tukevia toimintoja. Tällä tavoin pystytään tehokkaasti jakamaan tuotekehityksen kustannuksia ekosysteemin sisällä. Toimivan ohjelmistoekosysteemin avulla koko ekosysteemin pystyy skaalautumaan tehokkaammin, koska ekosysteemin toimijat täydentävät toistensa osaamista ja sen avulla pystytään tavoittamaan isompi joukko potentiaalisia asiakkaita ja saamaan syvempää ymmärrystä eri toimialoihin. (Bosch, 2009; Kim, 2019)

Tämän työn tarkastelukohteena on erityisesti kumppanit, jotka tekevät integraatioita omiin ohjelmistoihinsa, eli integraatiokumppanit. Integraatiokumppaneita ovat myös kumppanit, jotka tekevät integraatioita kolmannen osapuolen ohjelmistoihin. Integraatiokumppaneita varten ohjelmistoyrityksen täytyy luoda ja avata ohjelmistoon rajapinnat heitä varten. Nämä rajapinnat antavat kumppaneille pääsyn ohjelmiston keskeisiin prosesseihin ja tiedon keräämiseen ja siirtämiseen ohjelmistojen välillä. Integraatiokumppanit ovat myös merkittäviä osallisia ohjelmistoekosysteemissä, koska he ovat lähtökohtaisesti hyvin sitoutuneita ekosysteemin menestykseen ja kehitykseen, koska integraatioiden käyttöönotto ja heidän oman ohjelmistonsa kehittäminen on vaatinut resursseja, ennen kuin on päästy osaksi ekosysteemiä. (Moilanen et al.2019 s.83–85; Popp, 2010, s. 185)

Integraatiokumppanit hyödyntävät ohjelmistoyrityksen kehittämän ohjelmiston rajapintoja integraatioiden rakentamiseen. Näitä rajapintoja voidaan kutsua termillä API, eli Application programming interface, suomennettuna ohjelmointirajapinta. Yksinkertaisimmillaan ohjelmointirajapinnan avulla useat eri sovellukset voivat kommunikoida keskenään. Sen avulla voidaan siirtää tietoa helposti ja automaattisesti ohjelmien välillä. Esimerkkinä ohjelmistot voivat automaattisesti siirtää palkanlaskennan ohjelmistosta tiedot kirjanpitojärjestelmään.

Ohjelmointirajapinnat voidaan luokitella neljään eri kategoriaan. Avoimet ohjelmointirajapinnat on tarkoitettu julkiseen käyttöön, ja niissä ei ole ollenkaan tai hyvin vähän rajoituksia, ketkä niitä pääsee käyttämään. Tämä mahdollistaa ulkopuolisille tahoille hyvinkin nopean pääsyn dataan ja nopeuttaa kehitysvaihetta. Avointen rajapintojen avulla ulkopuoliset kehittäjät voivat rakentaa omia ohjelmistojaan alkuperäisen ohjelmiston rinnalle ja tällä tavoin luoda asiakkaille lisäarvoa. Avoimet rajapinnat lisäävät tietoturva uhkia, koska kaikilla on silloin vapaa pääsy ohjelmiston taustajärjestelmiin (De, 2017, s.7).

Toisena kategoriana on sisäiset ohjelmointirajapinnat ja ne on tarkoitettu vain yrityksen sisäiseen käyttöön siirtämään dataa vain sisäisten ohjelmistojen välillä. Sisäisiin rajapintoihin ei anneta pääsyä kolmansille osapuolille ja niiden tietoturvatoimet ovat hyvin tiukat. (Moilanen et al., 2019 s.54; Henderson, 2020; De, 2017 s.6–8)

Kolmantena on kumppani ohjelmointirajapinta ja se on tarkoitettu erityisesti ohjelmistoekosysteemin partnereille. Kumppani ohjelmointirajapinnoilla on kattavampi pääsy prosesseihin ja dataan, mutta myös tietoturva vaatimukset ovat kattavammat verrattuna avoimeen ohjelmointirajapintaan. Kumppani ohjelmointirajapintaan pääsy on myös rajattu vain valituille kumppaneille. (Moilanen et al., 2019 s.54; Henderson, 2020; De, 2017 s.6–8)

Neljäntenä on yhdistelmäohjelmointirajapinta, jonka tarkoituksena on mahdollistaa useiden rajapintakutsujen tekeminen yhdellä kutsulla. Tämän ohjelmointirajapinnan avulla kehittäjä pystyy lähettämään ja hakemaan tietoja useista rajapinnoista yhdellä kutsulla, joka yksinkertaistaa tiedonsiirtoa ja lisää ohjelmointirajapinnan käyttötarkoituksia. (Moilanen et al., 2019 s.54; Henderson, 2020; De, 2017 s.6–8)

Taulukko 1. Yleisimpiä ohjelmistoprotokollia (Riley, 2019; AltexSoft, 2020)

Nimi	Julkaistu	Formaatti	Vahvuudet
SOAP	1999	XML	Laajasti käytetty ja käytetään erityisesti tietoturvaa vaativissa ratkaisuissa.
REST	2000	JSON, XML, HTML ja teksti	Tiedonsiirto formaattien joustavuus ja helppokäyttöisyys
JSON-RPC	2005	JSON	Yksinkertaisuus
GraphQL	2015	JSON	Joustava datan kysely ja kevyt ratkaisu

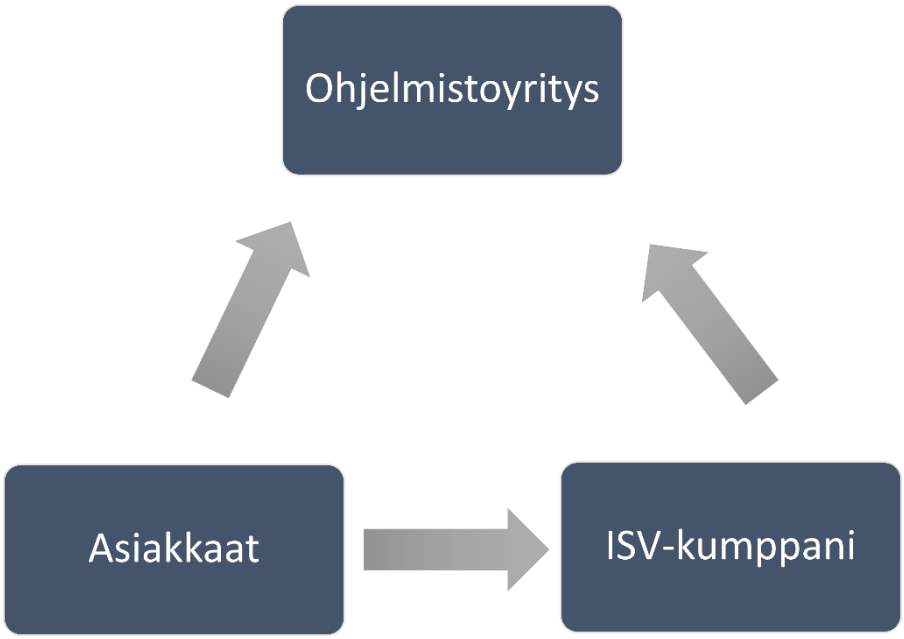
Yllä olevassa taulukossa 1. on listattu neljä ohjelmointirajapinta protokollaa. Ohjelmointirajapinnat noudattavat tiettyjä protokollia, joista tunnetuimpia ovat SOAP (Simple Object Access Protocol) ja REST (Representational State Transfer). SOAP-protokollan avulla ohjelmistot voivat lähettää toisilleen sovitun standardin mukaisia XML-formaatissa olevia tiedostoja, jotka sisältävät haluttua tietoa. XML formaatti on XML-kielen mukaisesti esitetty rakenteellista tekstidataa. REST on SOAP-protokollaan verrattuna kevyempi vaihtoehto ja se sallii tiedon lähettämisen useammassa eri tiedosto formaatissa esimerkiksi JSON-formaatissa. JSON-formaatti on XML kaltainen tekstidataa sisältävä formaatti, mutta se on huomattavasti yksinkertaisempi ja sisältää objekteja, jotka sisältävät avain-arvo-pareja. REST ohjelmointirajapinnat on kasvattanut suosiota merkittävästi sen helppokäyttöisyyden ansiosta. (Kankamge 2012, s. 11)

JSON-RPC on hyvin samankaltainen REST:in kanssa, koska molemmat hyödyntävät tiedon siirtämiseen JSON-formaattia, mutta JSON-RPC on yksinkertaisempi ja ei tarjoa samanlaista joustavuutta REST:iin verrattuna (Riley, 2019). GraphQL:n avulla ohjelmointirajapintaan voidaan tehdä pyyntöjä, joiden avulla saadaan haettua vain tarvittava tieto, jonka avulla tietomallin hallinta on helpompaa (Altexsoft, 2020).

2.1 ISV-Kumppaniohjelma ja sen hyödyt

Tässä kappaleessa käsitellään, sitä mitä ISV-kumppani ohjelma tarkoittaa ja mitä hyötyjä ISV-kumppaniohjelma tuo sitä tarjoavalle ohjelmistolle ja sen alkuperäiselle kehittäjälle ja myös mitä etuja ISV-kumppani ohjelmasta on sen partnereille. ISV-kumppaniohjelma on ohjelmistoyrityksen strateginen päätös laajentaa omaa palvelua ja toimintaa luomalla lisäarvoa ohjelmistoekosysteemin loppukäyttäjille. ISV-kumppani tuo ekosysteemiin oman ohjelmistotuotteensa, joka liitetään integraatioiden avulla ohjelmistoekosysteemin alkuperäiseen tuotteeseen. Tämän integraation avulla saadaan tuotettua loppuasiakkaalle lisäarvoa. (Kumpon, 2004) (Ramirez, 2020)

Yhtenä merkittävänä etuna ISV-kumppaniohjelmasta on sen tuottama liikevaihto molemmille osapuolille. ISV-kumppanit saavat tuloa siitä, että he voivat laskuttaa loppuasiakasta heidän tuotteensa ja rajapinnan käyttämisestä. Erilaisia laskutusperiaatteita on useita ja ne ovat useasti tuotekohtaisia. Ohjelmistoekosysteemin alkuperäinen omistaja ja kehittäjä hyötyy tästä siten, että he voivat esimerkiksi laskuttaa osuuden loppuasiakkaan ja kumppanin välisestä liikevaihdosta. Edellä mainitulla tavalla on toiminut esimerkiksi ohjelmistojätti Salesforce. Salesforce tarjoaa internetissä heidän kauppapaikallaan useita ISV-kumppanien tuottamia lisäosia heidän ohjelmistoonsa. Kauppapaikalta loppuasiakkaat voivat ostaa itselleen käyttöön ohjelmistoon ISV-kumppanien tuottamia lisäominaisuuksia. Salesforce ottaa itselleen tietyn prosentin ISV-kumppanin tuottamasta liikevaihdosta. Salesforce auttaa ISV-kumppaneita saamaan enemmän myyntiä generoimalla myyntivihjeitä, markkinoimaan ja auttaa viemään loppuun ISV-kumppanin myyntiä. Alla olevassa kuvassa 2 on havainnollistettu, kuinka ISV-kumppaniohjelman rahavirrat syntyvät. Asiakkaat maksavat ohjelmistoyritykselle ohjelmistoekosysteemin ohjelmiston käytöstä ja asiakkaat maksavat ISV-kumppanille heidän ohjelmistonsa käytöstä. Ohjelmistoyrityksen ja ISV-kumppanin välillä on tulonjakoa. ISV-kumppani saattaa joutua maksamaan ohjelmistoyritykselle vielä lisäksi kumppaniohjelmaan mukaan pääsemisestä lisämaksuja. (Ricadela, 2006, Popp & Meyer, 2010, s. 137)



Kuva 2. Rahavirrat ISV-kumppaniohjelmassa (Popp & Meyer, 2010 s. 137)

ISV-kumppaniohjelma on tärkeä myös siitä syystä, että ohjelmistoyrityksen on vaikea saavuttaa kaikkia heidän potentiaalisia asiakkaitaan ilman ohjelmistoekosysteemin ISV-kumppanien tuomaa apua ja lisäarvoa. Ohjelmistoyrityksen ei ole välttämättä kannattavaa itse luoda omaan ohjelmistoonsa kaikkia niche-ominaisuuksia, jos niitä varten löytyy ISV-kumppani, joka voi tuottaa kyseiset ominaisuudet (Kumpon, 2004 s.52). Ohjelmistoyrityksen on nykypäivänä erittäin vaikea olettaa, että pystyisi vastaamaan kaikkiin heidän asiakkaidensa tarpeisiin yksin, ja tässä tilanteessa on hyvä hyödyntää esimerkiksi ISV-kumppaneita. ISV-kumppanien avulla ohjelmistoekosysteemiin voidaan lisätä asiakkaille tärkeitä palveluita ja ominaisuuksia, joiden johdosta ekosysteemiin syntyy tahmeutta ja lisäarvoa loppuasiakkaille. Tahmeudella viitataan siihen, että loppuasiakkaan on vaikeampi siirtyä käyttämään kilpailijan tuotetta, koska heillä on niin monta palvelua jo yhdestä paikasta saatavilla. Ramirez (2020) antaa tästä hyvän esimerkin, jossa viittaa Amazon Alexan ja Spotifyn yhteistyöhön, loppukäyttäjä voi hyödyntää Amazon Alexa:a ohjaamaan hänen musiikkiaan Spotifyn kautta. Tämä johtaa siihen, että käyttäjät epätodennäköisemmin lopettavat Spotify tilauksensa, koska voivat käyttää samaa palvelua myös Amazon Alexan avulla. Blanda (2019) korostaa erityisesti sitä, että kun asiakkaan toimintamallit on upotettu syvälle ohjelmistoekosysteemiin se lisää merkittävästi sen käyttöä ja vähentää samalla asiakaspoistumaa. (Ramirez, 2020; Blanda, 2019)

Markkinointi on yksi ISV-kumppaniohjelman eduista, joista hyötyy molemmat osapuolet. Ohjelmistoyritys voi tarjota ISV-kumppaneille mahdollisuutta mainostaa heidän tuotteitaan ohjelmistoyrityksen kanavien kautta isommalle asiakasmäärälle, samalla ohjelmistoyritys voi markkinoida ISV-kumppanin asiakkaille omaa tuotettansa. Molemmat osapuolet pystyvät tällä tavoin kasvattamaan omaa brändi näkyvyyttä ja tehostaa markkinointia (Rickmann, 2014 s.8). Markkinoinnin avulla molemmat osapuolet voivat kasvattaa omaa asiakaskuntaa ja laajentua uusille markkinoille. Ohjelmiston myynti tehostuu ISV-kumppanin avulla erityisesti ohjelmistoekosysteemin kasvaessa, kun uusia ISV-kumppaneita haluaa orgaanisesti tulla osaksi ekosysteemiä ja samalla tuoden heidän loppuasiakkaansa ekosysteemiin (Kumpon, 2004 s. 57). (Blanda, 2019; Ricadela, 2016 s. 51; Moilanen et al. 2018 s. 86)

ISV-kumppanien avulla ohjelmistokehitystä voi tehostaa myös siten, että ISV-kumppani on voinut ratkaista jo esimerkiksi ohjelmiston vaatimuksiin liittyviä haasteita. Moilanen et al. tuo kirjassaan esimerkin esille, jossa ISV-kumppanin avulla pystyttiin ohjelmistoon lisäämään ominaisuus, jonka avulla pystyttiin vastaamaan regulaation tuomiin vaatimuksiin esimerkiksi rahaliikenteen hallitsemisessa. (Moilanen et al. 2018 s. 87)

ISV-kumppaniohjelmassa on omat haasteensa ja riskinsä kaikille osapuolille ja riskit tärkeä pystyä huomioimaan ja minimoimaan. Haasteena voi olla esimerkiksi, että ISV-kumppanit kilpailevat keskenään liian aggressiivisesti ja kieltäytyvät tekemään yhteistyötä. Haasteena ja riskinä voi myös olla se, kuinka ISV-kumppani hoitaa asiakastyytyväisyyttä. Jos ISV-kumppanin asiakkaan ovat tyytymättömiä heidän saamaan palveluun, se voi heijastua tyytymättömyytenä myös ohjelmistoekosysteemin ohjelmistoyritykseen. ISV-kumppani saattaa myös toimia vastoin ohjeita ja sääntöjä ja näin aiheuttaa haittaa koko ohjelmistoekosysteemille. Yksikin huono partneri voi aiheuttaa merkittävää imago haittaa koko ohjelmistoekosysteemille. ISV-kumppaniohjelma tuo ekosysteemiin lisää kompleksisuutta, joka hankaloittaa ekosysteemin hallinnointia ja vaatii ohjelmistoyritykseltä resursseja. ISV-kumppaniohjelman riskejä ja haasteita voidaan minimoida onnistuneella ISV-kumppanin valinta prosessilla. (Kaistinen, 2017 s. 62–64)

2.2 ISV-kumppanien valintakriteerit

Tässä luvussa käsitellään ISV-kumppaniohjelmaan liittyvien ISV-kumppanien valintaan liittyviä kriteereitä ja valintaprosessia. Tarkoituksena on tuoda esille kirjallisuudesta, mitä kriteerejä käytetään, kun ollaan valitsemassa uutta ISV-kumppania ohjelmistoekosysteemiin. Tämän työn keskiössä on case-yrityksen Netvisor ohjelmistoekosysteemi, joka hyötyy verkostoefektistä. Verkostoefektillä tarkoitetaan sitä, että ohjelmistoekosysteemin arvo ja houkuttelevuus uusille partnereille ja asiakkaille kasvaa samalla, kun uusia partnereita ja asiakkaita liittyy osaksi ekosysteemiä. Ohjelmistoekosysteemin ja kumppanien määrän kasvaessa on tärkeä pystyä hallinnoimaan ja valitsemaan parhaimmat kumppanit osaksi ekosysteemiä. Kuten edeltävässä kappaleessa mainittiin, että ISV-kumppaniohjelman riskien hallitsemista ja välttämistä varten partnerien valintaprosessi on erityisen tärkeä. (Rickman, 2014 s.3)

ISV-Kumppanin valintaan voi olla useita eri syitä ja Chen et. al. (2008, s.20) tuovat tutkimuspaperissa esille neljä yleistä syytä valita ISV-kumppani. Ensimmäisenä on strategiset syyt. Strategisilla syillä viitataan siihen, että ISV-kumppanin avulla yritetään kasvattaa tuottoja tai parantaa markkina-asemaa. Toisena syynä on kustannuslähtöinen. Kustannuslähtöisillä syillä viitataan siihen, että ISV-kumppanin avulla pyritään vähentämään ohjelmistoekosysteemistä aiheutuvia kuluja, esimerkiksi tehostamalla tuotekehitystä ja täten vähentämällä tuotekehityksen kuluja. Kolmantena syynä on resurssilähtöinen. Resurssilähtöisillä syillä viitataan siihen, että ISV-kumppanilla on jotain resursseja tai osaamista, jotka ovat hyvin arvokkaita ohjelmistoekosysteemille. Tällaisia resursseja voi esimerkiksi olla patentit tai markkinointikanavat. Neljäntenä syynä on oppimislähtöinen. Oppimislähtöisillä syillä viitataan siihen, että ISV-kumppani voi hyödyttää ohjelmistoekosysteemiä tuomalla mukaan uusia teknologioita ja tuoda sitä osaamista koko ekosysteemiin. Partanen ja Möller jakavat ISV-kumppanit kahteen kategoriaan; strategisiin ja ei-strategisiin kumppanit, strategisetkumppanit ovat kumppanit, jotka tuovat ekosysteemiin hyvin tärkeitä ja korvaamattomia tuotteitta tai palveluita ja ei-strategiset kumppanit tuovat ekosysteemiin korvattavia ja ei niin tärkeitä tuotteita tai palveluita (Partnen & Möller, 2012). (Chen et. al. 2008, s.20)

Kirjallisuudessa on lukuisia tapoja määrittää kriteereitä ISV-kumppaneille, koska kriteerit on yleensä linjattu ohjelmistoekosysteemin ohjelmistoyrityksen strategian mukaisesti ja kriteerit on yrityskohtaisesti suunniteltu siten, että niiden avulla voidaan varmistaa, onko potentiaalisen ISV-kumppanin ja heidän strategiansa linjassa ohjelmistoekosysteemin ohjelmistoyrityksen strategian kanssa. Beeleen (2019, s. 19–24) on tehnyt erittäin kattavaa tutkimusta partnerin valintaprosessista ja erityisesti kumppanin valintakriteereistä. ISV-kumppanien valintaa voidaan osaltaan rinnastaa myös muihin konteksteihin, joissa valitaan kumppani valittujen kriteerien pohjalta. Esimerkiksi Vankatesh, et. al (2019) hyödyntää usean kriteerin päätösanalyysin menetelmiä toimitusketju kumppanin valinnassa, jossa valintakriteerien avulla pyritään valitsemaan parhaimmat kumppanit.

Beelen (2019, s. 19–24) ei ota tutkimuksessa kantaa yrityksien strategioihin, vaan on kerännyt useista eri tutkimuksista partnerin valinta kriteereitä, joiden avulla ohjelmistoekosysteemin hallinnoiva taho voi valita parhaimmat partnerit. Beelen (2019, s. 19–24) oli kirjallisuudesta löytänyt 57 eri kriteeriä partnerin valintaprosessia varten. Liitteessä 1 on listattu kaikki Beelen (2019, s. 19–24) löytämät kriteerit. Kaikki kriteerit on jaettu kuuteen kategoriaan. Ensimmäinen kategoria on partnerin piirteet, eli niiden avulla voidaan kuvailla ja testata potentiaalisen ISV-kumppanin luonnetta ja potentiaalia. Toinen kategoria on partnerin kyvykkyydet, eli niiden avulla määritellään mitä ohjelmistoyritys on hakemassa potentiaaliselta ISV-kumppanin organisaatiolta ja työntekijöiltä. Kolmas kategoria on partnerin tuote, eli millaisia erityispiirteitä potentiaalisen ISV-kumppanin tuotteella on ja mitä hyötyjä niistä olisi ohjelmistoekosysteemille. Neljäs kategoria on partnerin omistukset, eli omistaako potentiaalinen ISV-kumppani esimerkiksi jotain immateriaalioikeuksia, joista olisi hyötyä ohjelmistoekosysteemille. Viides kategoria on partnerin myyntikyvyt, eli millaiset on potentiaalisen ISV-kumppanin myyntikanavat ja markkinaosuus. Kuudes ja viimeinen kategoria käsittää kriteerejä, jotka ovat ohjelmistoyrityksen näkökulmasta, eli millainen potentiaali ISV-kumppanilla on. (Beelen, 2019, s.19–24; Wann, et. al. 2009))

Beelen (2019, s. 19–24) haastatteli tutkimuksessaan kuutta eri isoa ohjelmistoekosysteemiä hallinnoivaa ohjelmistoyritystä ja kysyi heidän mielipidettänsä tärkeimmistä kriteereistä. Alle olevassa taulukossa 2. on esitetty jokaisen kategorian kolme tärkeintä partnerin valintakriteeriä.

Taulukko 2. Yhteenveto suosituimmista kriteereistä. (Beelen, 2019 s. 35)

Kategoria	1. Suosituin	2. Suosituin	3. Suosituin
Partnerin piirteet	Sitoutuneisuus yhteistyöhön	Asiakastyytyväisyys	Luotettavuus
Partnerin kyvykkyydet	Innovointikyvyt	Ainutlaatuinen osaaminen	Innovaatio keskeisyys
Partnerin tuote	Luotettavuus	Laatu	Integraation tehokkuus
Partnerin omistukset	Partneriyrityksen osaaminen	Tekninen osaaminen	Tuotekehityksen investoinnit
Partnerin myyntikyvyt	Asiakaskunta	Markkinaosuus	Markkinakattavuus
Ohjelmistoyrityksen näkökulma	Portfoliota täydentävä	Yhteistyön tuotto	Suositukset

Ramirez on Beelen (2019, s. 19–24) tavoin jakanut partnerin valinta kriteerit kategorioihin. Ramirez (2021) on jakanut kriteerit neljään omaan kategoriaan; asemointi, sitoutuneisuus, vakaus ja myyntikyvykkyys. Potentiaalisia partnereita arvioitaessa ohjelmistoyritys tarkastelee, kuinka potentiaalinen partneri on asemoitunut ohjelmistoekosysteemin ohjelmistoyritykseen verrattuna. Tähän liittyy selvitystyötä, jonka tarkoituksena on saada selville, sopiiko potentiaalisen partnerin brändi ohjelmistoyrityksen brändiin, onko molemmilla tahoilla samanlaiset pitkän aikavälin suunnitelmat. Tässä vaiheessa ohjelmistoyritys myös selvittää miten hyvin potentiaalisen partnerin tuote tukisi ohjelmistoekosysteemiä ja onko se kyseinen partneri jo jonkun toisen ohjelmistoyrityksen partnerina. (Wann, et. al. 2009)

Toinen kategorioista on sitoutuneisuus. Tähän kategoriaan kuuluu ohjelmistoyrityksen rooli selvittää, kuinka motivoitunut potentiaalinen partneri on yhteistyöhön. Osa selvitystyötä on käydä keskusteluja potentiaalisen partnerin kanssa ja tutustua heidän tuotteeseensa ja kehitystiimiin. Samalla ohjelmistoyritys voi tiedustella asioita, joita potentiaalinen partneri voisi parantaa. Potentiaalisen partnerin motivaation perusteella ohjelmistoyritys voi hyvin jo päätellä, onko mahdollista rakentaa yhteisiä suunnitelmia esimerkiksi markkinointiin ja tuotekehitykseen. (Wann, et. al. 2009)

Kolmas kategoria on vakaus, jolla viitataan siihen millainen vaikutus potentiaalisella partnerilla, on ohjelmistoekosysteemiin. Jos potentiaalisen partnerin brändi tai partneri on hyvin uusi tekijä markkinoilla, voi sillä olla negatiivinen vaikutus koko ekosysteemiin. Jos potentiaalisella partnerilla on pitkäaikaisia asiakassuhteita ja vaikutusta hänen markkinallansa, on sillä positiivinen vaikutus ekosysteemiin. Ohjelmistoyrityksen on myös hyvä ottaa huomioon, mitä vaikutusta potentiaalisen partnerin brändillä on parhaassa ja huonoimmassa skenaariossa. Potentiaalisen asiakkaan asiakassuhteista on myös hyvä tarkastella, mikä on asiakkaiden tyytyväisyys tuotteeseen ja asiakaspalveluun. (Wann, et. al. 2009)

Neljäntenä kategoriana on myyntikyvykkyys. Myyntikyvykkyyteen liittyy kaikki kolme aiemmin mainittua kategoriaa ja tällä kategorialla on ohjelmistoyritykselle isoin vaikutus. Tarkasteltavia asioita tässä kategoriassa on muun muassa se, että onko potentiaalinen partneri luotettava konsultti loppuasiakkaille, ottaako partneri koko ohjelmistoekosysteemin osaksi omaa myyntiprosessia, tuleeko partneri säästämään myyntiin kuluvia resursseja ohjelmistoyritykseltä, tekeekö partneri myyntiä tehokkaasti ja pystyykö potentiaalinen partneri auttamaan ohjelmistoekosysteemin nykyisiä asiakkaita. Ramirezin (2021) listaamat kategoriat ja kriteerit sisältävät paljon yhtäläisyyksiä Beelen (2019, s. 19–24) tutkimuksen kanssa. (Ramirez, 2021; Wann, et. al. 2009)

2.3 ISV-kumppanien priorisointi

Tässä luvussa käsitellään, miten kirjallisuudessa käsitellään ISV-kumppanien välistä priorisointia. ISV-kumppanien priorisointi on tärkeää, koska varsinkin isommissa ohjelmisto ekosysteemeissä voi helposti tulla tilanteita, joissa kaksi kumppania tarjoaa hyvin samanlaista palvelua tai ovat muuten hyvin samanlaisessa asemassa. Tällaisessa tilanteessa ohjelmistoekosysteemiä hallinnoivan ohjelmistoyrityksen tulisi pystyä päättämään priorisoitavien kumppanien väliltä.

Beelen (2019 s.45–46) tekemän tutkimuksen haastatteluissa selvisi useampi peruste sille, kuinka kumppaneita voidaan priorisoida. Ensimmäinen haastateltava yritys on ottanut hyvin asiakaslähtöisen tavan priorisoida asiakkaita. Heidän priorisointinsa perustuu asiakastarpeeseen

ja he eivät rajoita kumppanien pääsyä heidän ohjelmistoekosysteemiinsä, vaan asiakkaat saavat päättää mitä ominaisuuksia ja palveluita he haluaisivat mukaan. Kyseisen yrityksen myynti ja markkinointi tiimillä on myös vaikutusta kumppanien priorisointiin. Myynniltä ja markkinoinnilta voi tulla tietoa loppuasiakkaan tarpeesta, jonka avulla voidaan kasvattaa myyntiä ja, jos joku kumppani pystyy vastaamaan tähän tarpeeseen, saavat he heti korkeamman prioriteetin. Toinen Beelen (2019 s.45–46) haastattelema yritys ensisijaisesti priorisoi kumppaneita sen perusteella, onko heidän palvelunsa tai tuote parempi, kuin muut tällä hetkellä saatavilla olevat ja onko jollain kumppanilla enemmän kysyntää markkinalla, kuin toisella. Samainen yritys priorisoi myös sen perusteella, kuinka innokas kumppani on toimimaan yhteistyössä ohjelmistoekosysteemissä. Kolmas Beelen (2019 s.45–46) haastattelema yritys on samalla kannalla kuin ensimmäinen ja heidän priorisointinsa perustuu asiakaslähtöiseen kysyntään. Beelen (2019 s.45–46) haastatteli useita yrityksiä tutkimuksessaan ja priorisointi perusteet ohjelmistoyrityksillä painottuivat loppuasiakkaiden tarpeisiin, myynnin ja markkinoinnin tarpeisiin, kumppanin tekniseen osaamiseen, kumppanin motivaatioon, markkinoiden kysyntään ja viimeisimpänä kumppanin arvioituun liikevaihdon kehitykseen ja kumppanin saavutuksiin. (Beelen, 2019 s.45–46)

Weinbaum tuo artikkelissaan esille menetelmän, jossa kumppanien priorisointi perustuu siihen, että kumppanit pisteytetään tiettyjen kriteerien perusteella. Weinbaumin priorisointi menetelmässä jokaisesta kriteeristä annetaan pisteitä kouluarvosanojen tyylisellä asteikolla ja kumppanit priorisoidaan parhaimman keskiarvon perusteella. Weinbaumin käyttämät kriteerit ovat hyvin samanlaiset, kuin Beelen (2019 s.45–46) tutkimuksessa löytämät yleisesti käytetyt kumppanin valinta kriteerit. (Weinbaum, 2017)

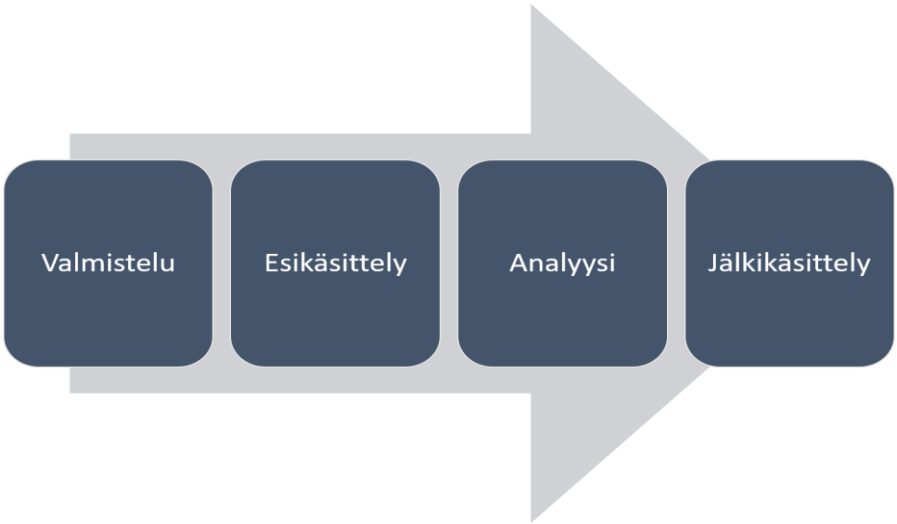
3 DATAN JALOSTAMINEN

Tässä luvussa käsitellään data-analytiikalle merkittävät vaiheet, eli mitä data on ja missä muodossa se yleensä on, ja kuinka datasta saadaan jalostettua tietoa. Tämän työn keskiössä on se, kuinka saatavilla olevasta datasta saadaan luotua informaatiota päätöksenteon tueksi. Tämä kappale on jaettu kolmeen pienempää osa-alueeseen; data-analytiikkaan, jossa käsitellään data-analytiikkaa ja analytiikan prosesseja pintapuolisesti. Toisena osa-alueena on datan määrittely, jossa syvennytään mitä data on ja mitä haasteita sen kanssa saattaa ilmetä. Kolmantena osa-alueena on DIKW-hierarkia, jota käytetään apuna datan jalostamisprosessin havainnollistamiseen.

3.1 Data-analytiikka

Data-analytiikka on hyvin laaja käsite, joka pitää sisällään erilaisia data käsittelyn menetelmiä, kuten tilastotieteitä, koneoppimista, kuvion tunnistusta ja tekoälyä. Data-analytiikan avulla pyritään löytämään datan sisältämää informaatiota ja sen avulla voidaan mahdollisesti löytää informaatiota, joka normaalissa tilanteessa jäisi huomaamatta. Data-analytiikan avulla voidaan esimerkiksi yritykset voivat löytää tuotannon pullonkauloja, tehostaa markkinointia tai reagoimaan markkinoiden tapahtumiin nopeammin. (Frankenfield, 2020)

Data-analytiikkaan prosessi voidaan jakaa neljään data-analyysin vaiheeseen. Data-analyysin vaiheet on esitetty alla olevassa Kuvassa 3. Ensimmäisenä vaiheena on valmistelu. Valmistelu vaihe pitää sisällään suunnitteluvaiheen, datan keräämisen, datan ominaisuuksien generoimisen ja datan valinnan. Valmisteluvaihe nimen mukaisesti vain valmistelee dataa seuraavaa vaihetta, eli esikäsittelyä varten. Esikäsittelyssä data siivotaan ja sieltä korjataan tai poistetaan virheellisiä arvoja. Käyttötarkoituksesta riippuen dataa voidaan tässä vaiheessa myös standardisoida ja muuttaa tarvittavaan muotoon analyysiä varten. Kolmantena vaiheena on analyysi, jossa käsitelty data voidaan visualisoida ja sen pohjalta voidaan luoda esimerkiksi ennusteita tai hyödyntää klusteroinnissa tai regressio analyyseissä. Viimeisimpänä vaiheena on jälkikäsittely, jossa dokumentoidaan ja arvioidaan lopputulosten luotettavuus. (Runkler, 2020 s.2–3)



Kuva 3. Data-analytiikka prosessin vaiheet. (Runkler, 2020)

Data-analytiikan käyttökohteita voidaan ymmärtää helpommin jakamalla se neljään alla listattuun data-analyysiin kategoriaan:

- Kuvaileva analytiikka
- Diagnosoiva analytiikka
- Ennustava analytiikka
- Ohjaileva

Kuvailevan analytiikan avulla voidaan vastata kysymykseen, ”mitä tapahtui?”. Kuvailevan analytiikan avulla tutkia historiadataa ja kuvailla tarkemmin mitä tapahtui. Kuvailevan analytiikan tärkeimpänä tehtävänä on muokata raaka data sellaiseen muotoon, että se voidaan visualisoida ja siitä voidaan ymmärtää mitä tapahtui. Kuvaileva analytiikan avulla voidaan huomata, jos joku asian on väärin mutta sen avulla ei pystytä vastaamaan, että miksi niin tapahtui. (Bekker, 2017; Mehta, 2017; Stevens, 2021)

Diagnosoivan analytiikan avulla voidaan vastata kysymykseen, ”miksi jotain tapahtui?” Diagnosoiva analytiikan avulla pystytään löytämään datasta poikkeamia ja korrelaatioita, joiden avulla pystytään selvittämään, miksi jotain tapahtui. Diagnosoivaan analytiikkaan tekniikoihin kuuluu esimerkiksi regressioanalyysi, klassifiointi ja herkkyysanalyysit. (Bekker, 2017; Mehta, 2017; Stevens, 2021)

Kolmantena on ennustava analytiikka ja se vastaa kysymykseen, mitä tulee todennäköisesti tapahtumaan. Ennustavassa analytiikassa hyödynnetään kuvailevan ja diagnosoivan analytiikan löydöksiä ja luo niiden pohjalta ennustettavia trendejä. Ennustavaa analytiikkaa voidaan käyttää esimerkiksi ravintolan asiakasmäärien ennustamiseen tai sosiaalisen median käyttäjien reaktiota johonkin asiaan. Ennustava analytiikka ei ole täysin tarkkaa ja ennustuksen tarkkuus riippuu pitkälti saatavilla olevan datan laadusta, ja siitä kuinka satunnaista ennustettava asia on. (Bekker, 2017; Mehta, 2017; Stevens, 2021)

Neljäntenä on ohjaileva, ja sen tarkoituksen on kertoa mihin toimiin kannattaa ryhtyä, jotta voidaan välttää tulevaisuuden ongelmat, tai hyötyä eniten trendeistä. Ohjaileva analytiikka on erittäin monimutkaista, koska siinä on otettava huomioon kaikkia mahdolliset lopputulokset ja niiden vaikutukset ja pystyä valitsemaan niistä paras. Tähän tehtävään hyödynnetään algoritmeja, koneoppimista, tilastollisia menetelmiä ja mallinnusta. Hyvä esimerkki ohjailevasta analytiikasta on hakukoneet, ne pyrkivät ennustamaan mikä hakutulos olisi hakijan kannalta kaikista parhain ja tarjoaa sitä ensimmäisenä. Toisena esimerkkinä on esimerkiksi parhaimman kuljetusreitin valinta, jossa on otettava huomioon eri kuljetustavat, reitit, tietyöt ja ruuhka. (Bekker, 2017; Mehta, 2017; Stevens, 2021)

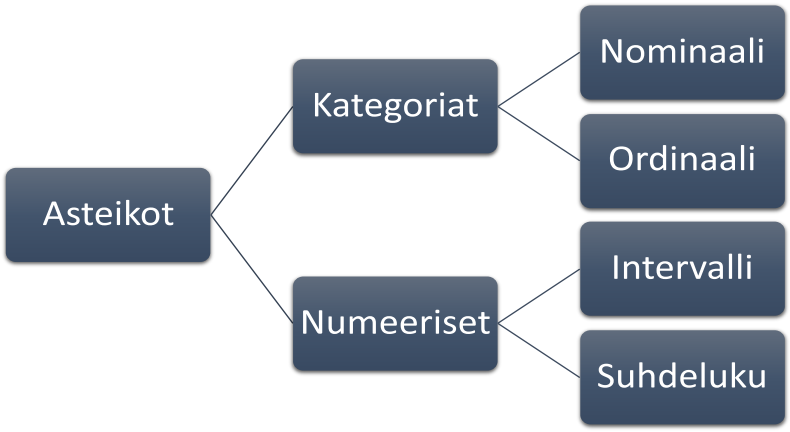
3.2 Datasta viisautta

Ahonen et. al (2017, s. 19) määritteleen datan olevan ”Tarkasteltavasta kohteesta kerättyä ja tallennettua numeriista ja ei-numeerista aineistoa”, Ahonen et. al. viittaa tällä raakadataa, joka sellaisenaan ei ole käyttökelpoista päätöksenteon tukena (Ahonen et al. 2017 s. 19). Ackoff (1999) taas määrittelee datan olevan symboleita, jotka määrittelevät objekteja ja tapahtumia (Ackoff, 1999). Ahsan ja Shah (2019) määrittelevä datan artikkelissaan olevan mitattua tai tilastollista tietoa, joka sisältää merkityksellistä ja merkityksetöntä informaatiota ja se täytyy prosessoida, ennen kuin siitä on hyötyä (Ahsan & Shah 2019).

Jotta datasta voidaan jalostaa käyttökelpoista informaatiota, täytyy ensin ymmärtää missä muodossa se on. Isossa kuvassa kaikki data voidaan jakaa strukturoituun ja strukturoimattomaan dataan. Taylor kuvailee artikkelissa, että strukturoitu data on sellaista dataa, josta voidaan hakea tietoa (Taylor, 2018). Pickel (2018) viittaa artikkelissaan

strukturoidun datan olevan dataa, joka sopii relaatiotietokantojen ja laskentataulukoiden kiinteisiin kenttiin ja sarakkeisiin (Pickel 2018). Strukturoitu data sisältää siis organisoitua ja helposti ymmärrettävää dataa, jota on helppo käsitellä. Strukturoimaton data on Balduccin ja Marinovan (2018) mielestä yleisesti ymmärretty olevan tietoa, jolla ei ole ennalta määrättyä tietomallia tai tietoa, jota ei ole järjestelty ennalta määrätyllä tavalla (Balducci & Marinova, 2018). Strukturoimaton data voi olla esimerkiksi tekstitiedostoja, äänitiedostoja, valokuvia, videoita. Strukturoimaton data voi olla ihmisen tai tietokoneen generoimaa dataa. Isoimmat erot strukturoidun ja strukturoimattoman datan välillä on sen analysoinnissa. Strukturoimattoman datan analysoiminen on merkittävästi haastavampaa. Strukturoidun datan analysointiin löytyy paljon hyviä työkaluja ja menetelmiä, mutta strukturoimattoman datan analysointiin tarkoitetut työkalut ja menetelmät kehittyvät jatkuvasti. Strukturoidun datan analysoimiseen voidaan käyttää esimerkiksi koneoppimiseen perustuvia menetelmiä. (Taylor, 2018; Pickell 2018)

Viittaus siihen, onko data strukturoitua vai strukturoimatonta viittaa siihen, missä muodossa se on saatavilla. Datan ominaisuudet voidaan jakaa neljään asteikkoon. Devi ja Murty kirjassa esittävät neljä eri data asteikkoa; nominaali, ordinaali, intervalli ja suhdelukuun (Devi &Murty 2015 s. 41). Gupta käy kirjassaan läpi datan asteikot vastaavasti Devin ja Murtyn (2015 s. 41) kanssa, mutta on jakanut jakanut asteikot eri tavalla alla olevan kuvan 4. kaltaisesti. Gupta on jakanut nominaalisen ja ordinaalisen asteikon kategoristen asteikkojen alle ja jakanut intervallin ja suhdeluvun numeeristen asteikkojen alle. Djangone tekee artikkelissaan samanlaisen jaottelun asteikoille kuin Gupta, mutta lisää sen, että kategorinen data on kvalitatiivista dataa ja numeerinen data on kvantitatiivista dataa (Djangone, 2021). (Gupta 2016, s. 11)



Kuva 4. Data asteikkojen neljä eri tyyppiä. (Gupta 2016, s. 11)

Nominaalinen data on kategorisoivaa dataa ilman järjestystä. Nominaaliasteikossa oleva data voi olla binääristä tai ei-binääristä. Binäärisessä muodossa oleva datalla on vain kaksi kategoriaa, esimerkiksi ”Tosi” tai ”Epätosi”. Ei-binäärisessä datassa on useita eri vaihtoehtoja, esimerkiksi henkilön puhuma äidinkieli tai henkilön hiustenväri. (Donges 2018; Singh 2020; Rennemeyer 2019)

Ordinaalinen data on kategorisoivaa dataa, mutta sillä on selkeä järjestys. Ordinaalisella asteikolla mitataan ei-numeerisia tietoja, kuten asiakastyytyväisyyttä. Ordinaalinen asteikko ei ota kantaa siihen, kuinka iso eri vaihtoehtojen välillä on. Asiakastyytyväisyyttä mitattaessa voitaisiin käyttää ordinaalista asteikkoa hyväksi kysymällä, ”Kuinka todennäköisesti suosittelisit tätä palvelua tuttavallesi”, ja vastausvaihtoehtoina olisi ”Todennäköisesti”, ”Mahdollisesti”, ”Ehkä” ja ”Epätodennäköisesti”. (Donges 2018; Singh 2020; Rennemeyer 2019)

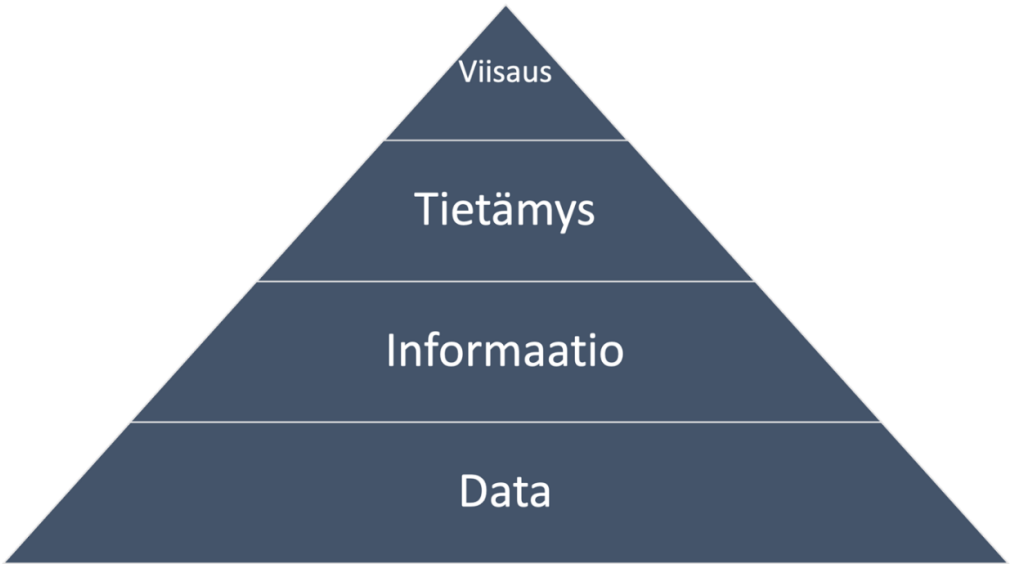
Numeerinen data voidaan ajatella myös diskreettina ja jatkuvana datana. Diskreetti data on kokonaislukuna, eikä sitä voida jakaa. Esimerkki diskreetistä datasta on henkilöiden määrä perheessä, tai opiskelijoiden määrä luokassa, koska luokassa ei pysty olemaan esimerkiksi 4.5 opiskelijaa, vaan määrän täytyy olla kokonaisluku. Jatkuva data on diskreetin datan vastakohta ja se voidaan jakaa ja esittää desimaalilukuna, esimerkiksi pituus tai paino. (Donges 2018; Singh 2020; Rennemeyer 2019)

Intervalliasteikko ja suhdelukuasteikko kuuluu numeerisiin asteikkoihin. Intervalliasteikon avulla voidaan kuvata datan järjestystä ja eroja arvojen välillä. Intervalliasteikon avulla voidaan tarkastella esimerkiksi lukujen keskijakaumia. Intervalliasteikolla voidaan kuvata esimerkiksi lämpötiloja, koska voidaan todeta, että 10 °C on suurempi kuin 0 °C. Intervalliasteikossa ei kuitenkaan ole mahdollista soveltaa jako- tai kertolaskuja, vaan siitä voidaan laskea pelkästään eroja ja yhteen- ja vähennyslaskujen avulla. (Donges 2018; Singh 2020; Rennemeyer 2019)

Suhdelukuasteikolla voidaan esittää datan järjestystä ja niiden välistä eroa. Suhdelukuasteikolla on ”absoluuttinen nolla”, jolla viitataan siihen, että esimerkiksi esineen korkeus ei voi olla 0 cm, koska silloin sitä ei olisi. Suhdelukuasteikolla mikään luku ei voi olla negatiivinen ja sen

arvon ollessa 0, sitä kuvastavaa asiaa ei ole olemassa. (Donges 2018; Singh 2020; Rennemeyer 2019)

Edellä on kuvattu sitä, millaista data on ja missä muodossa se ilmenee. Data on tärkein elementti, kun siitä lähdetään jalostamaan viisautta. Alla oleva kuva 5. esittää DIKW-hierarkiaa. DIKW-hierarkia tulee sanoista data-informaatio-tieto-viisaus-hierarkia, ja sillä esitetään tiedon jalostusastetta (Kortelainen et. al 2017, s. 20).



Kuva 5. DIKW-hierarkia. (Rowley, 2007)

Ackoff (1999) tuo esille sen, että informaatio palasta varten tarvitaan paljon dataa ja yhtä lailla tiedon palasta varten tarvitaan paljon informaatiota ja viisauden palasta varten tarvitaan paljon tietoa (Ackoff, 1999). Kuvassa 5. esitetty DIKW-hierarkia noudattaa pyramidin muotoa, juuri edellä mainitusta syystä. Jokaista jalostustasoa varten tarvitaan aina enemmän resursseja aikaisemmalta tasolta. Tässä kappaleessa syvennyttiin alussa hyvin paljon dataan ja mitä se on, koska se on DIKW-hierarkian perusta. Informaatio on DIKW-hierarkian toinen aste ja informaatiolla kuvataan dataa, joka on prosessoitu ihmiselle helposti ymmärrettävään muotoon. Informaation avulla voidaan vastata kysymyksiin, kuka, mitä, missä tai milloin? Informaatioksi voidaan myös kutsua dataa, jolle on annettu tarkoitus ja se on apuna tietyn aihealueen ymmärtämisessä. Informaatiojärjestelmät tuottavat, keräävät, tallentavat ja prosessoivat dataa yksinkertaisella tasolla. Esimerkki informaatiosta, on kun halutaan tietää tietyn kuukauden keskilämpötila, silloin on saatavilla datana kuukauden eri päivien lämpötiloja, mutta dataksi ei

ole välttämättä tallennettu suoraan keskilämpötilaa, vaan se voidaan laskea datan pohjalta ja tulkita informaationa. (Rowley, 2007; Frické, 2009; Ackoff, 1999)

Tietämys on DIKW-hierarkian kolmas aste ja sillä viitataan tietotaitoon ja osaamiseen, kuten siihen, että henkilö muistaa tai tietää ulkomuistista jonkin asian. Tietämyksellä voidaan vastata kysymykseen, miten? Tietämys rakentuu informaation päälle, joka on luotu datan pohjalta. Tietämyksen saavuttamiseksi informaatioon on lisätty tietotaitoa ja kontekstuaalista ymmärrystä ja se vaatii kykyä tulkita informaatiota ja sen pohjalta tunnistaa tarvittavia toimenpiteitä. Tietämyksenä voidaan myös pitää kykyä tunnistaa datasta toistuvia tapahtumia ja se on perustana ennustavien mallien luomiselle. (Rowley, 2007; Frické, 2009; Ackoff, 1999; Figueroa 2019)

DIKW-hierarkian neljäs ja viimeinen aste on viisaus. Viisaudella viitataan siihen, että sen avulla voidaan vastata kysymykseen, miksi jotain tapahtui? Viisaus perustuu kykyyn hyödyntää edellä mainittujen tasojen tietoja tulevaisuuden ennustamiseen. Figueroa esittää artikkelissaan käytännön esimerkin DIKW-hierarkiasta. Datan avulla voidaan tietää, että ulkona sataa. Informaation avulla voidaan tietää, että ulkona lämpötila on pudonnut viisi astetta ja kosteus on noussut viisi prosenttia ja tämän jälkeen ulkona alkoi satamaan. Tietämyksen avulla voidaan sanoa, että todennäköisesti ulkona alkaa satamaan, jos lämpötilaa laskee äkillisesti ja kosteus nousee. Viisauden avulla voidaan todeta, että aiemman tietämyksen avulla voidaan ennakoita tehokkaasti, milloin ulkona alkaa satamaan. (Rowley, 2007; Frické, 2009; Ackoff, 1999; Figueroa 2019)

4 KONEOPPIMINEN

Koneoppimiseen liittyviä algoritmeja on ollut käytössä jo 1970-luvulta lähtien. Koneoppiminen on kasvattanut suosiotaan 2000-luvulla, koska koneoppimisalgoritmit vaativat paljon laskentatehoa ja tietokoneiden laskentatehon kasvaessa koneoppiminen on kasvattanut suosiota. Laskentatehon kasvaessa pystytään ratkaisemaan entistä vaikeampia ongelmia. Datan määrän kasvaessa pystytään entistä paremmin hyödyntämään koneoppimista ja saamaan siitä entistä enemmän hyötyjä irti. Koneoppimisessa tietokone oppii algoritmien avulla tekemään päätöksiä tai tehtäviä, kun sille on annettu tarpeeksi dataa ja esimerkkejä oikeista päätöksistä ja tehtävistä. Koneoppimis algoritmeja ja käyttötarkoituksia on monenlaisia, mutta koneoppimista voidaan käyttää esimerkiksi kasvojen tai esineiden tunnistamiseen kuvista tai videosta tai ennusteiden mallintamiseen. (Louridas & Ebert 2016, s. 110)

Koneoppiminen voidaan jakaa neljään eri kategoriaan, ohjattuun oppimiseen, ohjaamattomaan oppimiseen, vahvistettuun oppimiseen ja osittain ohjattuun oppimiseen. Tässä kappaleessa keskitytään tarkemmin ohjattuun- ja ohjaamattomaan oppimiseen, koska ne ovat yleisimmät ja helppo käyttöisimmät koneoppimisen menetelmät. (Wakefield 2021; Li 2020)

4.1 Ohjattu oppiminen

Ohjattu oppinen on koneoppimisen menetelmä, jossa koneoppimisalgoritmia opetetaan esimerkkien avulla. Ohjatun oppimisen tärkeimpänä tekijänä on se, että koneoppimisalgoritmille syötettävässä datassa on oltava mukana oikeat vastaukset. Esimerkiksi, jos halutaan koneoppimisalgoritmin ennustavan asuntojen hintoja, silloin algoritmille syötettävässä datassa on hintaan vaikuttavia parametreja ja myös toteutuneita asuntojen myyntihintoja. Ohjatussa oppimisessa koneoppimisalgoritmille syötetään tarpeeksi dataa, jonka jälkeen sen avulla voidaan ennustaa edeltävän esimerkin perusteella asunnon myyntihintoja pelkästään saatavilla olevien hintaan vaikuttavien parametrien perusteella, kuten pinta-ala, sijainti ja kunto. (Edwards, 2018; Vaseekaran, 2018; Joshi 2020, s. 10; Bonaccorso 2018, s. 13–14)

Ohjatun oppimisen yleisimmät käyttökohteet ovat klassifiointi ja regressio. Klassifioinnissa koneoppimisalgoritmin tarkoituksena on etsiä datasta sääntöjä, joiden avulla voidaan erottaa datapisteet toisistaan. Klassifioinnin avulla data voidaan jakaa erilaisiin kategorioihin. Jos vaihtoehtoina on vain kaksi kategoriaa, silloin kyseessä on binäärinen klassifiointi ja jos vaihtoehtoja on useampi kuin kaksi, silloin on kyseessä moniluokkainen klassifiointi. Joshi (2020) tuo kirjassaan esille useita esimerkkejä klassifioinnista. Klassifiointia voidaan hyödyntää esimerkiksi siihen, että määritetään, onko saatu sähköposti roskapostia vai ei. Roskapostiviestit voidaan klassifioinnin avulla myös jaotella omiin kategorioihinsa. Klassifiointia voidaan hyödyntää myös esimerkiksi kuvien tunnistamiseen, jossa esimerkiksi tunnistetaan kuvasta eläimiä ja jaetaan eri eläimet omiin kategorioihin. (Wakefield 2021; Li 2020; Edwards, 2018; Joshi 2020, s. 179–180)

Ohjatun oppimisen regressio algoritmeja hyödynnetään ennustamiseen. Regressio koneoppimisalgoritmin on ymmärrettävä datan sisältämiä suhteita ja korrelaatioita ja hyödyntää niitä ennustamisessa. Regressiota hyödynnetään numeerisissa ongelmissa, eli esimerkiksi sään tai osakekurssin ennustamisessa. Jo aiemmin mainittu asunnon hinnan ennustaminen tapahtuu regressiota hyödyntämällä. Asunnon hinnan määrittäminen sopii täydellisesti regressioon, koska kyseessä on numeerinen lopputulos, mutta tärkeää on myös ottaa huomioon mitkä asiat vaikuttavat lopputulokseen. (Wakefield 2021; Edwards, 2018; Joshi 2020, s. 188–190)

4.2 Ohjaamaton oppiminen

Ohjaamaton oppiminen on koneoppimisen menetelmä, jossa koneoppimisalgoritmia opetetaan pelkästään lähdetietojen perusteella. Ohjaamattomassa oppimisessa koneoppimisalgoritmille ei anneta oikeita vastauksia. Ilman oikeita vastauksia koneoppimisalgoritmien avulla pystytään silti löytämään mielenkiintoisia ja monimutkaisia kuvioita ja asioita pelkästään lähtödatan avulla. Klassinen esimerkki ohjaamattomasta oppimisesta on klusterointi. Klusteroinnissa koneoppimisalgoritmi oppii pelkän lähtödatan pohjalta lajittelemaan datan joukkoihin. Koneoppimisalgoritmi voisi esimerkiksi oppia lajittelemaan kolikoita koon ja värin perusteella siten, että samat kolikot ovat omassa joukossaan. (Edwards, 2018; Vaseekaran, 2018; Joshi 2020, s. 11; Bonaccorso 2018, s. 17)

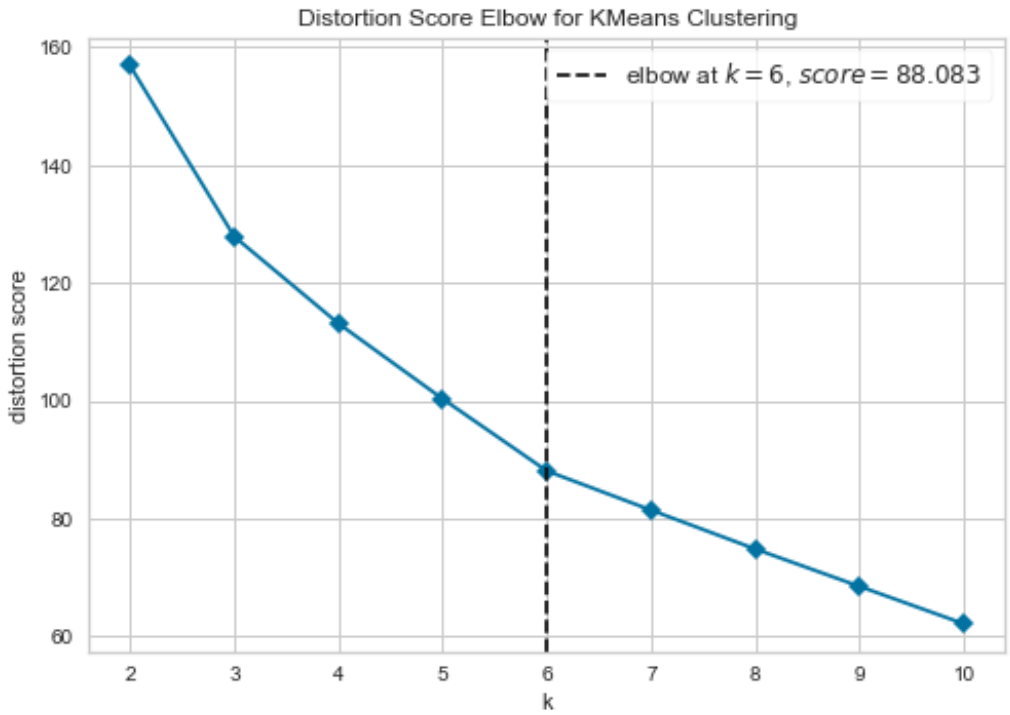
Ohjaamaton oppiminen voidaan jakaa klusterointiin ja datan ulottuvuuksien vähentämiseen. Kuten aiemmin mainittiin klusteroinnissa datapisteet jaotellaan ennalta määrättyjen kriteerien pohjalta joukkoihin, eli klustereihin. Kriteerien avulla voidaan erottaa klusterit toisistaan ja kriteerien avulla pystytään määrittämään mihin klusteriin tietty datapiste kuuluu. (Wakefield 2021; Edwards, 2018; Joshi 2020, s. 134; Bonaccorso 2018, s. 182)

Datan ulottuvuuksien, eli attribuuttien vähentämisessä on kyse siitä, että voi olla tilanne, jossa datassa on erittäin paljon attribuutteja ja ei ole laskennallisesti tehokasta käyttää kaikkia attribuutteja mukana koneoppimisessa. Datan ulottuvuuksia vähentämällä voidaan säilyttää suurin osa datan sisältämästä varianssista samalla, kun vähennetään datan määrää. Datan varianssin säilymisellä viitataan siihen, että mahdollisimman vähän informaatiota häviää, kun raakadatasta poistetaan muuttujia, jotka ovat analysoinnin kannalta tarpeettomia. Datan ulottuvuuksia vähentämällä voidaan myös helpommin löytää datasta oikeita korrelaatioita ja sen käsittely yksinkertaistuu. (Wakefield 2021; Edwards, 2018; Bonaccorso 2018, s. 474–475)

Ohjaamattoman oppimisen klusterointialgoritmeista tunnetuimpia ovat K-means ja K-modes algoritmit. K-means ja K-modes algoritmien periaate on hyvin samanlainen, ne pyrkivät jakamaan datasetissä olevat datapisteet klustereihin. Klusterien määrää esittää kirjain K. K-means ja K-modes algoritmit etsivät datasta samanlaisuuksia ja korrelaatioita ja yhdistää ne niiden pohjalta klustereihin. Molemmat algoritmit perustuvat samoihin välivaiheisiin. (Ahdmed, 2020; Maklin, 2018; Yildirim, 2020)

Ensimmäisessä vaiheessa algoritmille annetaan klusterien määrä K. Toisessa vaiheessa algoritmi määrittelee satunnaisesti K-määrän klustereita ja niiden keskipisteitä. Kolmannessa vaiheessa algoritmi laskee datapisteiden etäisyyden klustereiden keskiosaan. K-means algoritmissa nimenmukaisesti lasketaan etäisyyksien keskiarvoa ja K-modes algoritmissa lasketaan etäisyyksien moodi keskipisteestä. Algoritmi minimoi iteroimalla kaikkien datapisteiden etäisyydet suhteessa klustereiden keskipisteisiin. Algoritmin laskenta on valmis, kun etäisyyden keskiarvo tai moodi, ei enää muutu seuraavalla iteraatiolla. (Ahdmed, 2020; Maklin, 2018; Yildirim, 2020)

Klusterien määrän voi selvittää kyynärpää menetelmän avulla. Kyynärpää menetelmän avulla kuvataan klusterien määrän suhdetta vääristymään eli klusterin sisältävien datapisteiden etäisyyksien neliön summaan. Mitä enemmän klustereita on, silloin klusterien sisäiset etäisyydet pienenevät, mutta on tärkeää löytää se kohta, jossa klusterien kasvavalla määrällä on kaikista pienin vaikutus. Alla olevassa kuvaajassa 6. on havainnollistettu kyynärpää menetelmän avulla luotua kuvaajaa. Kuvaajasta voidaan nähdä, että vääristymän määrä laskee nopeasti kuudenteen klusteriin asti ja hidastuu merkittävästi sen jälkeen. Tästä voidaan todeta, että kyseiselle datalle K:n arvo tulisi olla kuusi. (Ahdmed, 2020; Maklin, 2018; Yildirim, 2020)



Kuva 6. PyCaret kirjastolla luotu kuvaaja K-means algoritmin kyynärpääkuvaajasta.

4.3 Osittain ohjattu oppiminen ja vahvistettu oppiminen

Osittain ohjattu oppiminen on hyvin samanlaista, kuin ohjattu oppiminen. Ainut ero osittain ohjatun ja ohjatun oppimisen välillä on se, että koneoppimisalgoritmille annetussa oppimisdatassa on saatavilla vain osa oikeista vastauksista. Koneoppimisalgoritmi yrittää oppia osittain puutteellisen datan avulla ja oppimisen lopputuloksena koneoppimisalgoritmi pystyy jatkossa täydentämään opetusdataan oikeat vastaukset. Osittain ohjatun oppimisen haasteena

on kuitenkin se, että puutteellinen data voi aiheuttaa vääriä lopputuloksia (Singh et. al. 2008). Vahvistetussa oppimisessa (eng. Reinforcement learning) koneoppimis algoritmille annetaan tietyt parametrin ja tehtävät ja algoritmille annetaan palautetta sen suoriutumisen perusteella. Epäonnistuneesta suorituksesta algoritmille annetaan huonoa palautetta ja onnistumisesta hyvää palautetta. Vahvistettua oppimista voidaan hyödyntää esimerkiksi videopeleissä, algoritmin epäonnistuessa se häviää pelin ja onnistuessaan voittaa. (Wakefield 2021; Kahler 2021)

5 USEAN KRITEERIN PÄÄTÖSANALYYSI JA SEN MENETELMÄT

Tässä luvussa käsitellään usean kriteerin päätösanalyysia ja sen eri menetelmiä. Tarkoituksena on myös syventyä eri menetelmiin ja niiden eroihin ja käyttötarkoituksiin. Usean kriteerin päätösanalyysi on yksi päätöksenteko prosessin tekniikoista. Yrityksien johtajat joutuvat säännöllisesti tekemään rationaalisia päätöksiä ja ratkaisemaan päätöksentekoon liittyviä ongelmia. Rationaalisen päätöksenteon tueksi päättäjät tarvitsevat ammattilaisen näkemystä aihealueesta, tilastollista tietoa ja dataa aiheeseen liittyen ja päättäjän oma mielipide asiaan. Päätöksenteko prosessi voidaan yksinkertaistaa kolmeen vaiheeseen; informaation etsimiseen, vaihtoehtojen ja kriteerien määrittämiseen ja viimeisenä parhaimman vaihtoehdon valinta. Tätä päätöksentekoa voidaan helpottaa työkalujen, ohjelmistojen tai algoritmien avulla ja monen kriteerin päätösanalyysi kuuluu näihin algoritmeihin. Usean kriteerin päätösanalyysiä käytetään tilanteissa, joissa päätöksentekoon vaikuttavia kriteereitä on useita ja vaihtoehtoja on useita. (Hudson, 2015 s. 2,4; Thakkar, 2021 s. 1–3)

Usean kriteerin päätösanalyysi tunnetaan myös termillä monen kriteerin päätöksenteko. Molemmilla termeillä viitataan samanlaiseen päätöksenteon menetelmään, jossa paras vaihtoehto valitaan arvioimalla vaihtoehtoja keskenään useiden kriteerien pohjalta. Tyypillinen Usean kriteerin päätösanalyysi alkaa kriteerien ja vaihtoehtojen määrittely saatavilla olevan tiedon pohjalta. Toisena vaiheena on asettaa numeeriset arvot kriteereille, joiden avulla voidaan määritellä mikä kriteeri on tärkeämpi kuin toinen ja kolmantena vaiheena on matemaattisten menetelmien hyödyntäminen vaihtojen tärkeysjärjestykseen laittamisessa. (Thakkar, 2021 s. 2–4

Usean kriteerin päätösanalyysiä voidaan helposti hyödyntää monessa eri käyttötarkoituksessa, ja käyttötarkoituksesta riippuen täytyy vain valita siihen sopivin monen kriteerin päätösanalyysin menetelmä. Usean kriteerin päätösanalyysia voidaan hyödyntää esimerkiksi tuotetoimittajan valinnassa, jossa kriteereinä voisi olla toimitusaika, toimitushistoria, laatu ja muut tekijät. Usean kriteerin päätösanalyysia voidaan hyödyntää myös yhtä lailla esimerkiksi oikean tuotantolaitteiston hankinnassa, jossa kriteereinä voi toimia hinta, toimivuus, turvallisuus ja tuotto. Yhtä lailla usean kriteerin päätösanalyysiä terveellisen ruokavalion valintaan, jossa vaihtoehtoina on tietyt ruoka-aineet ja kriteereinä on ruuan sisältämät

ravintoarvot. Widianta et al. hyödyntää artikkelissaan usean kriteerin päätösanalyysiä henkilöstöhallinnon tehostamisessa ja erityisesti uusien työntekijöiden valitsemisprosessia tehostamalla, jossa päätösanalyysin kriteereinä toimii työnhakuun liittyvät työntekijän osaaminen ja tietämys. Huang et al. hyödyntää artikkelissaan usean kriteerin päätösanalyysiä ekosysteemin partnerien valintaprosessissa. (Thakkar, J. J. 2021, s.9–10; Adriyendi, S. 2015; Widianta et al., 2018; Huang et al., 208)

5.1 Painotettu summa -malli

Triantaphyllou et al. (1998), Afshari et al. (2010) ja Thakkar (2021) tuovat esille yksinkertaisen ja yhden eniten käytetyistä menetelmistä, eli painotettu summa -malli, joka on englanniksi weighted sum model. Weighted sum model (WSM) tunnetaan myös nimellä Simple Additive Weighting (SAW), eli yksinkertainen additiviinen painotus. SAW-menetelmän tarkoituksena on antaa jokaiselle kriteerille painoarvo, ja näiden painoarvojen avulla jokaiselle päätöksen vaihtoehdolle lasketaan painotettu summa. SAW-menetelmän laskeminen on helpompi havainnollistaa alla olevien kaavojen 1–5 avulla;

$$A = \{A_i \,|i = 1,2,3,..,\ n\} \tag{1}$$

$$C = \{C_j \,|j = 1,2,3,..,\ n\} \tag{2}$$

$$W = \{w_1,\ w_2, w_3,, \dots, w_n\} \tag{3}$$

$$\mathrm{X} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \tag{4}$$

$$A_i = \sum w_j \times\ x_{ij}\ \ i = 1, \dots, n, j = 1, \dots, n \tag{5}$$

Kaavassa 1. on havainnollistettu joukko A, joka kuvaa usean kriteerin päätösanalyysin kaikkia eri vaihtoehtoja $A_1, A_2, \dots, A_n$, kaavassa 2. on havainnollistettu joukko C, joka kuvaa vaihtoehtojen joukkoon A kohdistuvia kriteereitä. Kaavassa 3. on havainnollistettu kriteerien joukkoon C kohdistuvia painokertoimia. Neljännessä kaavassa on päätösmatriisi, jonka arvot x_{mn} kuvaavat vaihtoehdon A_i suhdetta kriteerin C_j. Kaavassa 5. on havainnollistettu painotetun summan laskeminen kertomalla jokainen päätöksen vaihtoehdon kriteeri sitä koskevalla painoarvolla ja nämä tulokset lasketaan yhteen. Yhteenlaskettu summa on yksittäisen vaihtoehdon kokonaispisteet ja se vaihtoehto, mikä saa eniten pisteitä on tällä menetelmällä

kaikista merkittävin. Painotettua summaa laskiessa on kuitenkin tärkeä huomata, että päätösmatriisin arvot on normalisoitava, ennen kriteerien painotuksien laskemista. Päätösmatriisin arvot saadaan normalisoitua kahdella tavalla, jos tietty kriteeri halutaan maksimoida, silloin sen normalisointi lasketaan kaavalla: $n_{ij} = \frac{c_{ij}}{c_{ij}{}^{max}}$, jossa arvo r_{ij} jaetaan kriteerisarakkeesta löytyvällä maksimiarvolla. Jos tietty kriteeri halutaan minimoida, silloin sen normalisointi lasketaan kaavalla: $n_{ij} = \frac{c_{ij}{}^{max}}{c_{ij}}$, jossa kriteerisarakkeen maksimiarvo jaetaan arvolla c_{ij}. Kriteerien maksimoinnilla ja minimoinnilla viitataan siihen, tilanteeseen jos halutaan, että tietyn kriteerin arvo on mahdollisimman suuri, silloin arvot maksimoidaan ja jos halutaan, että tietyn kriteerin arvo on mahdollisimman pieni, silloin sen kriteerit arvot minimoidaan. SAW-menetelmän etuina on sen yksinkertaisuus ja se säilyttää raakadatassa olevat suuruuserot ja ottaa ne laskennassa huomioon. Rajoituksena SAW-menetelmässä on se, että se vaatii kaikkien kriteeriarvojen olevan positiivisia ja painoarvojen määrityksessä voi syntyä harha-arvioita. (Afshari et al., 2010; Triantaphyllou et al., 1998; Thakkar, 2021, S.27; Adriyendi, S. 2015; Widianta et al., 2018)

5.2 Analyyttinen hierarkiaprosessi

Kirjallisuudessa paljon käytetty usean kriteerin päätösmenetelmä on AHP, eli analyyttinen hierarkiaprosessi tai englanniksi Analytic Hierarchy Process. AHP on hyvin samanlainen aiemmin mainittuun SAW-mallin kanssa. Isoin ero SAW-malliin on se, että AHP:ssa otetaan huomioon kaikki päätöksentekoon vaikuttavat kriteerit ja näiden kriteerien välille määritellään keskinäiset painoarvot. AHP-mallia hyödynnetään useasti tilanteissa, joissa päätöksentekoon vaikuttavia kriteereitä on useita, AHP-malli perustuu siihen, että päätöksenteon kriteereitä verrataan keskenään ja kriteeri pareille annetaan luku väliltä 1–9 sen mukaan, kuinka paljon merkittävämpi toinen kriteeri on toiseen verrattuna. Luku 1 viittaa siihen, että kriteerit ovat yhtä tärkeitä ja luku 9 viittaa siihen, että toinen kriteeri on merkittävästi tärkeämpi. Tämä parittainen vertailu tehdään kaikille kriteereille. Kaavassa 6. on havainnollistettu kriteereistä tehtyä matriisia, jossa on kriteerien väliset suhdeluvut. Kaavan 6 matriisi muodostaa suhdematriisin.

$$A = \begin{bmatrix} w_1/w_1 & w_1/w_2 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \cdots & w_2/w_n \\ \vdots & \cdots & \ddots & \vdots \\ w_n/w_1 & w_n/w_2 & \cdots & w_n/w_n \end{bmatrix} \tag{6}$$

Kaavassa 5 esitetystä suhdematriisista saadaan johdettua ominaisvektori. Suhdematriisin pohjalta johdettu ominaisvektori sisältää AHP-menetelmän painoarvot. Painoarvojen johtamisen jälkeen AHP-menetelmä seuraa SAW-menetelmää samalla tavalla ja vaihtoehtoille lasketaan pisteet kertomalla normalisoidut kriteerien arvot niiden painoarvoilla. AHP-menetelmän etuna on se, että sen rakenne on yksinkertainen mutta samalla sen avulla voidaan huomioida kompleksisia oikean elämän ongelmia. AHP-menetelmän etuina on myös se, että se voidaan helposti integroida muihin kvantitatiivisiin tai kvalitatiivisiin lähestymistapoihin, eli esimerkiksi peliteorian tai SWOT analyysin yhteyteen. AHP-menetelmän avulla päättäjät voivat hyödyntää omaa osaamista ja intuitiota paremmin, kun painoarvoissa on kyse kriteerien välisistä suhteellisista tärkeyksistä. AHP-menetelmän haasteena kuitenkin on kriteeriparien välille muodostettavien suhteellisten tärkeyksien rakentaminen, erityisesti tilanteessa, jossa kriteereitä on paljon. AHP-menetelmän parittainen kriteerien arvioiminen vaatii osaamista ja tietämystä sovellettavasta kontekstista. AHP-menetelmä vie merkittävästi enemmän aikaa verrattuna yksinkertaisempaan SAW-menetelmään. AHP-menetelmää on hyödynnetty muun muassa toimitusketjun partnerinvalinnassa. (Klutho, S. 2013, s. 8–11; Thakkar J. J., 2021, s.33–36, s. 60–61; Widianta et al., 2018; Triantaphyllou et al., 1998; Velasquez, M & Hester, P, 2013; Macharis et al., 2004; Venkatesh et al., 2019)

5.3 TOPSIS

TOPSIS, eli tekniikka priorisointijärjestykselle samankaltaisesti, kuin ihanteellinen ratkaisu on AHP- ja SAW-menetelmien rinnalla yleinen usean kriteerin päätöksenteon menetelmistä, mutta ei kuitenkaan niin yleinen. TOPSIS-menetelmä perustuu siihen, että kaikkia päätöksenteon vaihtoehtoja verrataan parhaimpaan mahdolliseen ratkaisuun ja jokaiselle vaihtoehdolle lasketaan etäisyys tästä parhaimmasta mahdollisesta ratkaisusta ja lasketaan etäisyys huonoimmasta vaihtoehdosta. Paras vaihtoehto on se, jonka pituus on lähimpänä parasta tilannetta ja kauimpana huonoimmasta tilanteesta. Etäisyyksien laskennassa hyödynnetään euklidisen etäisyyden laskemista ja sen tulosta. Ideaali tilanne muodostetaan sen perusteella, että jokaisesta kriteeristä otetaan paras ja huonoin arvo ja näitä arvoja verrataan muihin kriteereihin. Tässä vaiheessa on tärkeä olla määriteltynä, onko kriteeri hyödyllinen vai haitallinen. Hyödylliset kriteerit ovat sellaisia kriteereitä, jossa halutaan maksimoida kyseisen

kriteerin luku ja haitallisissa kriteereissä halutaan minimoida kyseisen kriteerin luku. Etuina TOPSIS-menetelmässä on, että se laskenta ja priorisointi on yksinkertaista ja päättäjä pystyy itse sen hyvin ymmärtämään. TOPSIS-menetelmän avulla pystytään saamaan myös nopeasti tuloksia, joiden avulla voidaan huomata nopeasti paras vaihtoehto. Haasteena TOPSIS-menetelmässä on se, että se ei ota kantaa kriteerien välisiin suhteisiin tai korrelaatioihin ja olettaa kriteerien laskevan tai kasvavan monotonisesti, eli arvot vain kasvavat tai vähenevät. Haasteena on myös painoarvojen määrittäminen. Thakkar J. J., 2021, s.90–92; Triantaphyllou et al., 1998; Velasquez, M & Hester, P, 2013; Widianta et al., 2018; Uppal S., 2020)

5.4 Promethee

Promethee (Preference Ranking Organization Method for Enrichment Evaluations) on kirjallisuudessa useasti käytetty usean kriteerin päätöksenteon menetelmä. Promethee on aiemmin esiteltyihin menetelmiin verrattuna uusin. Promethee menetelmällä vaihtoehdot laitetaan parhaimmuus järjestykseen preferenssifunktioiden avulla. Promethee-menetelmä käyttää aiempien menetelmien tavoin normalisoitua päätösmatriisia ja promethee-menetelmä vaatii jokaiselle kriteerille painoarvot. Promethee-menetelmä ei ota kantaa siihen, miten painoarvot määritetään, vaan se on päätöksentekijä vastuulla. Päätöksentekijän pitää myös päättää kriteereille sopivat preferenssifunktiot. Preferenssifunktioita on kuusi erilaista funktioita ja niiden avulla määritetään, miten preferenssi tiettyyn kriteeriin muuttuu. Preferenssi voi muuttua esimerkiksi lineaarisesti, porrasmaisesti, eksponentiaalisesti tai gaussilaisesti. Kriteerit ja niihin yhdistetyt preferenssifunktiot aggregoidaan summafunktion avulla. Summafunktion avulla voidaan laskea jokaiselle vaihtoehdolle. Pisteytyksen avulla voidaan suoraan verrata vaihtoehtoja keskenään ja se vaihtoehto, joka saa enemmän pisteitä kuin toinen on tässä tapauksessa parempi vaihtoehto. Promethee menetelmää voidaan hyödyntää päätöksenteossa, joka sisältää kvalitatiivista ja kvantitatiivista dataa ja promethee menetelmän avulla voidaan helpommin verrata vaihtoehtoja keskenään, joita olisi muilla menetelmillä vaikeampi verrata. Haasteine promethee menetelmässä on painoarvojen määrittäminen, koska sille ei ole varsinaista ohjeistusta ja promethee menetelmän luomaan vaihtoehtojen paremmuusjärjestykseen ei pysty lisätä uusia vaihtoehtoja, ilman että koko prosessi suoritetaan uudestaan. (Thakkar J. J., 2021, s.120–124; Papathanasiou J. & Ploskas N., 2018; Macharis et al., 2004; Velasquez, M & Hester, P, 2013; Widianta et al., 2018)

6 VISMA SOLUTIONS OY

Case-yrityksys, eli Visma Solutions Oy on yksi Visma konsernin tytäryhtiöistä. Case-yritys perustettiin vuonna 1996 ja sen pääkonttori sijaitsee Norjassa, Oslossa. Visma-konsernissa on yli 11 000 työntekijää ja vuoden 2019 liikevaihto oli 1526 miljoonaa euroa. Visma-konserniin kuuluu yli 200 yritystä 20 eri maasta. Visma toimii kaikissa Pohjoismaissa, Benelux-maissa sekä Keski- ja Itä-Euroopassa. Vismalla on myös laaja jälleenmyyjien ja kumppanien verkosto ja maarajojen yli toimiva tuotekehitysorganisaatio. Visma tarjoaa liiketoiminalle kriittisiä järjestelmiä ja palveluita, jotka tukevat yrityksiä ja julkishallinnon organisaatioita. (Visma a)

Suomessa Visma on ollut vuodesta 2001 lähtien ja suomessa Visma konserniin kuuluu kymmenen tytäryhtiötä, jotka työllistävät noin 1500 työntekijää ja palvelevat yli 100 00 asiakasta. Visman tytäryhtiöitä suomessa ovat Visma Ai Works, Visma Consulting, Visma Enterprise, Visma Financial Solutions, Visma Megaflex, Visma Passeli, Visma Public, Visma Software, Visma Solutions ja Visma Tampuuri. Suomessa Visma tarjoaa ohjelmistoratkaisuja muun muassa toiminnanohjaukseen, taloushallintoon, sähköiseen allekirjoitukseen, ajanvarauksiin, hankintaan, henkilöstöhallintaan ja työmaiden hallintaan. (Visma b)

Case-yritys on yksi Suomen johtavista SaaS-ohjelmistojen, eli Software as a Service toimittaja. Case-yrityksellä on tällä hetkellä viisi eri SaaS-palvelua. Tämä palvelut ovat Visma Netvisor, Visma Severa, Maventa, Visma Sign ja ValueFrame. Visma Netvisor on tarkoitettu taloushallinnon työkaluksi. Sen avulla voidaan hallita yrityksen myynnit, ostot, henkilöstö, palkanlaskenta ja kirjanpito. Visma Severa on projektinhallintaohjelma, jonka avulla voidaan yhdistää projektit, työtunnit ja laskutuksen. Maventa on ohjelma verkkolaskutukseen ja automaattiseen maksuvalvontaan. Visma Sign on työkalu sähköiseen allekirjoittamiseen, joka hyödyntää vahvaa tunnistautumista. Visma Signin allekirjoituksia käytetään esimerkiksi sopimuksien ja dokumenttien allekirjoittamiseen. ValueFrame on asiantuntijayrityksille suunnattu toiminnanohjausjärjestelmä. (Visma Solutions)

6.1 ISV-kumppaniohjelma case-yrityksessä

Tässä luvussa käsitellään sitä, kuinka ISV-kumppaniohjelma toimii case-yrityksessä ja mitä se tarkoittaa partnereille. ISV-kumppaniohjelma on ohjelmistokehittäjille suunnattu yhteistyömalli, jonka tavoitteena on luoda elinvoimainen ja kasvava liiketoimintaekosysteemi, joka tuo lisäarvoa loppuasiakkaille ja mahdollisuuden kannattavaan liiketoimintaan kolmansille osapuolille.

Kumppaniohjelman tarkoituksena on kehittämään kolmansien osapuolien avulla palveluita, jotka eivät ole case-yrityksen ydinliiketoimintaa. Kumppaniohjelmaan suositaan erityisesti kumppaneita, jotka kehittävät mobiili- sekä pilvipohjaisia ratkaisuja. Kumppaniohjelman kumppaneille tarjotaan käyttöön ohjelmointirajapinnat ja kattavat lisämateriaalit. Case-yritys auttaa kumppaneita ratkaisujen kaupallistamisessa ja tarjoaa apua ohjelmistorajapintojen käytössä.

Kumppaniohjelmaan hakevilta ohjelmistokehittäjiltä vaaditaan selkeää liiketoimintasuunnitelmaa markkinointikanavien ja myyntikanavien käytöstä. Kumppani on myös vastuussa kehittämän palvelunsa jakelusta, ylläpidosta ja asiakaspalvelusta. Kaikkia kumppaneita ei automaattisesti hyväksytä kumppaniohjelmaan, vaan kumppanin täytyy myös täyttää case-yrityksen asettamia laadullisia ja kaupallisia vaatimuksia. Kumppaniohjelmasta ei aiheudu liittymiskuluja, vaan asiakkailta, jotka käyttävät kumppanin hyödyntämiä palveluita veloitetaan rajapintojen käytöstä API-maksu ja siihen liittyvät transaktiokulut. Case-yritys tarjoaa kumppanien kehittämille palveluille näkyvyyttä case-yrityksen markkinapaikoilla, yhteisöissä ja verkkosivuilla ja ratkaisu esitellään myös myyjille. Myyjät voivat mahdollisuuksien mukaan tarjota kumppanin kehittämää ratkaisua lisänä suoraan asiakkaalle.

Kumppaniohjelman kumppanit voidaan pääsääntöisesti jakaa kahteen kategoriaan; partnereihin ja Premium kumppaneihin. Premium kumppaniksi ohjelmistokehittäjä voi päästä, kun se on merkittävä osa case-yrityksen ohjelmistoekosysteemiä. Premium kumppaneille tarjotaan korotettua näkyvyyttä, yhteisen liiketoiminnan suunnittelua case-yrityksen kanssa, myyntivinkkejä sekä aktiivista asiakaskartoitusta, kehitysympäristöjen laajennettu

käyttöoikeus. Laajennettu käyttöoikeus avaa kehitysympäristöihin enemmän ominaisuuksia. Case-yritys tekee tiiviimpää yhteistyötä Premium kumppanien kanssa ja fokuksena on kehittää molempien osapuolien liiketoimintaa Premium ja Partner tason kumppanien lisäksi case-yrityksessä on myös käytössä Single ja Software Partner tasot. Single Partnerit ovat kumppaneita, jotka ovat luoneet integraation yhden asiakkaan tarpeen täyttämiseksi, eli esimerkiksi integroineet asiakkaan sisäisiä järjestelmiä case-yrityksen tuotteisiin. Software Partner on kumppani, joka on sertifioitu kehittäjä tai ohjelmistotalo, jota voidaan suositella loppuasiakkaille tuottamaan yksittäisiä integraatioita heidän omiin sisäisiin järjestelmiinsä. ISV-kumppaniohjelmassa on myös mukana AO-partnereita, AO on lyhenne sanoista Accounting Office, eli tilitoimisto. AO-partnerit, ovat case-yrityksen tilitoimistokumppaneita, mutta ovat myös ISV-kumppaneita. AO-partnereilla on omaa kehitystoimintaa ja ISV-kumppaniohjelman integraatiot liittyvät kyseisten tilitoimistojen omaan ohjelmiston kehitystoimintaan.

6.2 Netvisor ja integraatio rajapinnat

Tämä työn on rajattu koskemaan vain pelkästään case-yrityksen Netvisor palvelua, joka on yksi case-yrityksen SaaS-palveluista. Netvisor on yrityksen taloushallinnon ja liiketoimintaprosessit yhdistävä pilvipalvelu. Netvisorin avulla yritys voi hallita sen myynnit, ostot, henkilöstönhallinnon, palkanlaskennan ja kirjanpidon. Tällä hetkellä Netvisoria käyttää noin 20 000 yritystä monilta eri toimialoilta ja yli 500 tilitoimistoa Suomessa. Netvisorin avulla yritys pystyy esimerkiksi automatisoimaan sen laskutusta ja kirjanpitoa. Netvisor tarjoaa yrityksille myös mahdollisuuden seurata tarkasti reaaliajassa omaa kassavirtaa ja maksuvalmiutta. Yrityksille on saatavilla Netvisorista neljä eri hinnoittelumallia. Netvisorin hinnoittelumallit on suunnattu eri kokoisten organisaatioiden tarpeiseen ja hinnat vaihtelevat 15 euron ja 90 euron kuukausimaksun välillä.

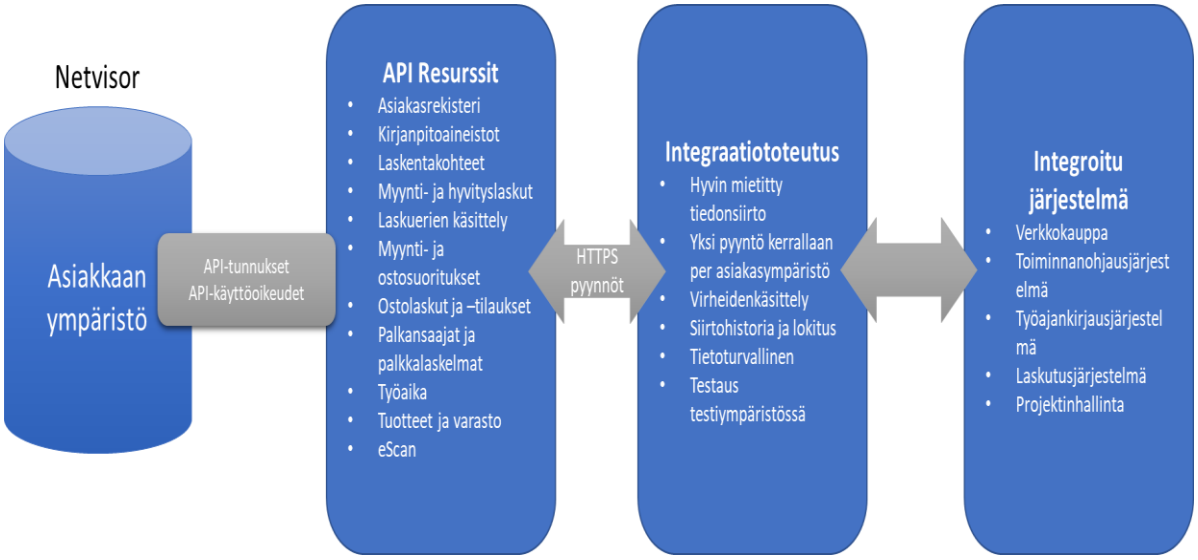
Netvisori on valmiiksi integroitu ISV-kumppanien ansiosta jo yli 200 ohjelmistoon ja toimialaratkaisuun. Netvisorin käyttäjä pystyy helposti ottamaan käyttöön valmiit integraatiot Netvisorin tarjoamasta Netvisor Storesta. Netvisorin valmiiden integraatioiden avulla yritys pystyy esimerkiksi yhdistämään heidän verkkokauppansa suoraan Netvisoriin, tai yhdistää laskutuksen suoraan erilliseen perintäpalveluun, tai siirtää tietoja suoraan erilliseen raportointia

ja seurantaa tarjoavaan ohjelmistoon. Integraatiot ovat yleensä erikseen hinnoiteltu ja ne eivät suoraan sisälly Netvisorin kuukausimaksuun. Integraatioiden hinnat ovat integraatio kohtaisia ja niiden laskutus vaihtelee palvelusta riippuen. (Netvisor, 2020)

Netvisor Marketplace tarjoaa paikan asiakkaille tutkia ja perehtyä tarkemmin saatavilla oleviin Netvisorin integraatioihin. Marketplacelle integraatiot on jaettu kahteen pääkategoriaa: Store-kumppaneihin ja tuotteistettuihin integraatioihin. Jos integraatiotuote sisältää Store-merkinnät Marketplacella, silloin sen voi käyttäjä ottaa sen omatoimisesti käyttöön suoraan Netvisorissa muutamassa minuutissa. Tuotteistetut integraatiot vaativat yhteydenottopyyntöä ohjelmiston toimittajaan ja he hoitavat asiakkaan integraation loppuun.

Marketplacelta löytyvät integraatiotuotteet voidaan jakaa myös tuotekategorioihin. Näitä tuotekategorioita ovat digitaalinen tehokkuus, henkilöstö, integraatioalustat ja -palvelut, liiketoiminta, myynti, osto, rahoitus, raportointi ja BI, talous, toiminnanohjaus ja verkkokauppa. Yksittäinen integraatiotuote voi kuulua useampaan tuotekategoriaan. Tuotekategorioiden avulla asiakas voi etsiä esimerkiksi henkilöstöhallintoon sopivia integraatioita tai verkkokauppa integraatiolla yhdistää oman yrityksensä verkkokaupan integraatioilla Netvisoriin. (Netvisor b, 2021)

Alla olevassa kuvaajassa 7. on havainnollistettu Netvisorin kaksisuuntaista rajapintapalvelua. Kaksisuuntaisen rajapintapalvelun avulla voidaan yhdistää asiakkaan järjestelmä Netvisoriin. Kaksisuuntainen rajapintapalvelu sopii pienten sekä isompien järjestelmien liittämiseksi Netvisoriin. (Netvisor b, 2021)



Kuva 7. Netvisorin kaksisuuntainen rajapintapalvelu (Netvisor b, 2021)

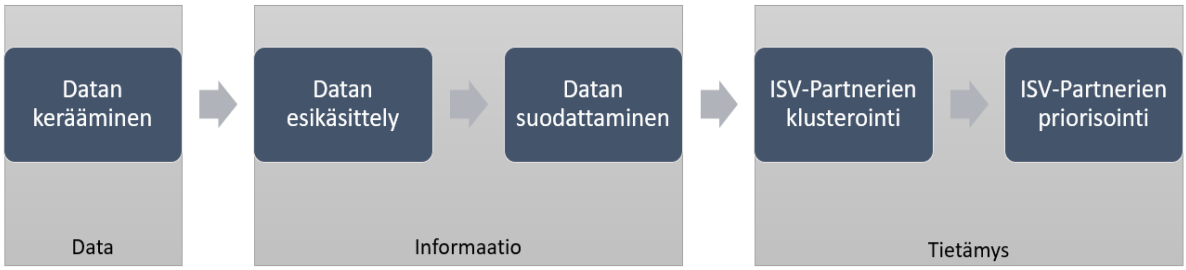
Integroitavaan järjestelmästä voidaan hakea Netvisorista myyntilaskuja, ostolaskuja, ostotilauksia, tuotteita, asiakkaita, kirjanpidon tositteita, maksutoimeksiantoja, palkanlaskennan palkkaperusteita ja resurssienhallinnan työaikakirjaustietoja. Integroitavasta järjestelmästä voidaan hakea Netvisoriin suorituksia, pankkitapahtumia, asiakkaita, ostotilauksia, tuotteita, laskuja ja laskun saldoja ja kirjanpidon tietoja. Ohjelmistorajapinnan avulla voidaan Netvisoriin yhdistää esimerkiksi verkkokauppoja, toiminnanohjausjärjestelmiä, työajanseurantajärjestelmiä, laskutusjärjestelmiä ja projektinhallinnan järjestelmiä. (Netvisor b, 2021)

Netvisorin integraatiorajapinta toimii REST-mallin mukaisesti. Kommunikoitu integroidun järjestelmän ja Netvisorin välillä tapahtuu http-pyynnöillä. Tietoa tuotaessa Netvisoriin, lähettävä taho muodostaa aineistoa kuvaavan XML-muotoisen sanoman ja lähettää sen http-pyynnön välityksellä Netvisoriin. Netvisorista tietoa poispäin vietäessä pyynnön ei tarvitse olla XML-muotoinen sanoma, vaan pyyntö osoitetaan haluttuun resurssiin tarvittavilla parametreilla ja Netvisor palauttaa vastauksen XML-muodossa. (Netvisor b,2021)

7 DATA-ANALYYSIN PROSESSI

Tässä luvussa käsitellään tämän diplomityön data-analyysiin liittyvä prosessi. Tässä kappaleessa nojataan laajalti teoriassa esille tuotuun DIKW-pyramidiin. Ensimmäisessä alakappaleessa käsitellään, miten data on kerätty ja miten sitä on esikäsitelty. Ensimmäinen alaluku viittaa DIKW-pyramidin ensimmäiseen tasoon, eli dataan. Toisessa alaluvussa käsitellään, millaista dataa jäi jäljelle esikäsittelyn jälkeen. Toisen alaluvun tarkoituksena on havainnollistaa DIKW-pyramidin toista tasoa, eli informaatiota. Tässä vaiheessa data on kerätty ja jalostettu helpommin ymmärrettävään ja käytettävään muotoon. DIKW-pyramidin kolmas taso eli tietämys saavutetaan, kun informaatiota jalostetaan. Informaatiota jalostetaan tietämykseksi työn seuraavassa vaiheessa.

Alla olevassa kuvaajassa 8. on havainnollistettu tämän työn data-analyysin prosessi. Prosessi on jaettu kolmeen eri osakokonaisuuteen, data, informaatio ja tietämys. Osakokonaisuudet viittaavat suoraan aiemmin mainittuihin DIKW-pyramidin tasoihin. Prosessin ensimmäisessä vaiheessa kerätään ja yhdistetään datalähteitä. Prosessin ensimmäinen vaihe käsitellään luvussa 7.1. Prosessin toinen vaihe, eli informaatio pitää sisällään datan esikäsittelyn ja datan suodattamisen. Tässä vaiheessa prosessin ensimmäisestä vaiheesta eli datasta jalostetaan informaatiota. Datan esikäsittelyn ja suodattamisen avulla data saadaan sellaiseen, muotoon että sitä voidaan analysoida, mitata tai visualisoida. Prosessin toinen vaihe käsitellään kappaleessa 7.1 ja 7.2. Prosessin kolmas ja viimeinen vaihe, eli tietämys saavutetaan jalostamalla informaatiota. Tässä prosessin vaiheessa esikäsiteltyyn dataan suoritetaan laskennallisia ja tilastollisia menetelmiä, eli klusterointia ja usean kriteerin päätösanalyysi. Prosessin kolmas vaihe käsitellään kappaleissa 7.3 ja 7.4.



Kuva 8. Data-analyysin prosessi havainnollistettuna

7.1 Datan kerääminen ja esikäsittely

Tässä luvussa käsitellään, kuinka tätä työtä varten kerättiin tarvittava data ja kuinka se esikäsiteltiin jatkoprosessointia varten. Työ rajattiin koskemaan vain ainoastaan case-yrityksen Netvisor tuotteen ISV-kumppaneita, joten datan kerääminen alkoi selvittämällä, mitä tietoa partnereista on saatavilla. Case-yritys käyttää Tableau ohjelmistoa datan jatkojalostamiseen ja visualisointiin.

Tableau on 2003 perustettu SaaS-ohjelmisto, jonka avulla sen käyttäjät voivat jalostaa dataa sellaiseen muotoon, että sitä voidaan käyttää apuna luomaan organisaatiosta enemmän data ohjautuneempi. Tableau on business intelligence ohjelmisto, kuten esimerkiksi Microsoftin PowerBI tai Qlik Solutionin Qlik ohjelmisto. (Tableau, 2019)

Data haettiin suoraan Tableausta, koska sinne data oli aggregoitu useista eri tietolähteistä jo valmiiksi. Tämä säästi merkittävän määrän työtä, koska muuten olisi täytynyt jokaisesta tietolähteestä erikseen käydä keräämässä tarvittava data partnereista, mutta nyt kaikki data oli saatavilla yhdestä lähteestä. Tableau on jatkokehityksen kannalta myös hyvä alusta, koska se tarjoaa mahdollisuuksia jatkojalostaa dataa esimerkiksi omien Python-skriptien avulla.

Data oli Tableausta haettava silti useasta eri taulusta, mutta datan pystyi lataamaan CSV-tiedostoina käsittelyä varten. Kaikki datan esikäsittelyyn ja tutkimiseen liittyvä prosessointi hoitui Python ohjelmointikielen Pandas ohjelmointikirjastolla. Python on erittäin yleinen ja se soveltuu moniin eri käyttötarkoituksiin ja erityisesti laajalti käytetty data-analytiikassa (Python, 2021).

Pandas ohjelmointikirjasto on nopea, tehokas ja monipuolinen työkalu datan käsittelyyn ja analysointiin ja se on rakennettu Python ohjelmointi kielen päälle. Pandas tukee erinomaisesti useita eri tietoformaatteja ja hyödyntää erityisesti nopeaa ja tehokasta DataFrame objekteja, jotka auttavat datan manipuloinnissa ja indeksoinnissa. DataFramien avulla pystytään käyttämään ja muokkaamaan hyvin nopeasti isojakin datamassoja. (Pydata, 2015)

Netvisorin ISV-kumppaneista oli kerätty dataa vuodesta 2017 alkaen, mutta datan seasta huomattiin jatkuvuus ongelmia ja osa tiedoista saattoi olla väärin. Tässä työssä käsitelty data on pelkästään vuodelta 2020, koska vuoden 2020 alusta kerätty data oli kaikista kattavinta ja parhaiten saatavilla. Vuoden 2020 datasta oli myös saatavilla enemmän tietoja partnereista, kuin esimerkiksi vuodelta 2017. Jos olisi voitu varmistaa datan oikeellisuus ja eheys vuodesta 2017 lähtien nykyhetkeen, olisi datan pohjalta voinut saada paremman kuvan mihin suuntaan partnerin ISV-toiminta on kehittynyt. Muutamasta data attribuutista saatiin luotettavasti kerättyä edeltävän, eli vuoden 2019 data, jotta voitiin laskea vuotuinen muutos. Datan attribuutteihin palataan myöhemmin tässä kappaleessa.

ISV-kumppaneista saatu data oli hyvin hajanaista, joten CSV tiedostojen keräämisen jälkeen ne ladattiin yksikerrallaan omiin Pandasin tarjoamiin DataFrame objekteihin käsittelyä varten. Datasetistä etsittiin puuttuvia arvoja ja ne täytettiin NaN-arvoilla, eli määrittelemättömillä numeroarvoilla. Jos kyseessä oli tekstiä sisältävä sarake, silloin puuttuvat ja vialliset tietueet täytettiin ’Not Found’-tekstillä. Tällä tavoin datasta löytyneet puutteet saatiin yhtenäistettyä, jotta datan eheyttä olisi helpompi seurata. Samassa yhteydessä datasarakkeiden arvot käytiin läpi ja varmistettiin, että kaikissa sarakkeissa oli vain sille tarkoitettua dataa, esimerkiksi varmistettiin, että numeerisia arvoja sisältävä sarake sisältää pelkästään numeroita. Tämän jälkeen datan tietueet aggregoitiin Partnerin nimen perusteella. Aiemmin datan joukossa saattoi olla samalla partnerilla useita eri tietueita, mutta jatkokäsittelyn kannalta jokaisella Partnerilla oli tärkeää olla vain yksi tietue. Datan aggregoinnin jälkeen kaikki CSV-tiedostoista kerätyt DataFrame objektit pystyttiin yhdistämään yhdeksi DataFrame objektisi Partnerin nimen perusteella.

ISV-kumppanien nimet oli kirjattu hieman eri muodoissa eri tietolähteissä, joten partnerien nimi parametri piti yhtenäistää eri tietolähteiden välillä. Partnerin nimen sisältävä parametri sisälsi alun perin partneri yrityksen nimen ja heidän tuotteen nimen samassa parametrissa. Tästä piti parsia erilleen yrityksen ja tuotteen nimi. Parsimisen jälkeen piti yhtenäistää tietolähteisiin partnerin nimi, jota käytettiin tietolähteiden yhdistämiseen. Yhtä tietolähdettä käytettiin lähtötietotauluja ja muiden tietolähteiden partnerin nimiä verrattiin tähän tauluun.
Vertailussa käytettiin Python ohjelmointikielen FuzzyWuzzy-kirjastoa, jonka avulla voidaan suorittaa sumeaa tekstin yhdistämistä. FuzzyWuzzy-kirjasto hyödyntää sumeaa logiikkaa

siihen, kun se etsii tekstistä yhtäläisyyksiä. Yhtäläisyyksien etsimiseen kyseinen kirjasto hyödyntää Levensteinin etäisyyden laskemista. Levensteinin etäisyydellä voidaan kertoa kuinka samanlaisia kaksi eri tekstiä ovat ja Levensteinin etäisyydellä voidaan myös kertoa, kuinka monta operaatiota tekstiin täytyy tehdä, jotta molemmat tekstit olisivat samanlaisia. Operaatioita ovat merkkien lisääminen, poistaminen ja korvaaminen toisilla merkeillä. Kyseisen kirjaston avulla pystyi yhdistämään kaikkien tietolähteiden partneri nimet toisiinsa todennäköisyyksien perusteella. Yhdistämisen jälkeen tulokset käytiin manuaalisesti vielä läpi tarkkuuden varmistamiseksi. (Cohen, 2011)

DataFrame-objektien yhdistämisen jälkeen ylimääräiset parametrit poistettiin, koska niitä käytettiin pelkästään apuna datan yhdistämiseen. Datasettien yhdistämisen jälkeen kaksi kategorista-muuttujaa muokattiin numeeriseksi muuttujaksi. Tämä tapahtui One-Hot encoding-menetelmää hyödyntäen, jossa kategorisista muuttujista tehdään numeerisia muuttamalla ne riveistä sarakkeiksi. One-Hot encoding menetelmää hyödynnettiin sen takia, koska sen avulla pystyttiin paremmin hyödyntämään tilastollisia menetelmiä korrelaatioiden tarkasteluun. Tietyt menetelmät eivät pysty käsittelemään parametreja, jotka ovat kategorisessa muodossa, vaan se täytyy muokata numeeriseen muotoon. Alla olevassa taulukoissa 3. ja 4. on esimerkki, kuinka one-hot encoding -menetelmä toimii.

Taulukko 3. Esimerkki datasta ilman one-hot-encodausta.

Indeksi	Ammatti
0	Siivooja
1	Kokki
2	Kirjanpitäjä
3	Kokki

Taulukko 4. Esimerkki datasta, joka on one-hot-encodattua.

Indeksi	Ammatti_siivooja	Ammatti_kirjanpitäjä	Ammatti_kokki
0	1	0	0
1	0	0	1
2	0	1	0

3	0	0	1

Käytännössä One-Hot encoding -menetelmällä muutetaan kategoriset arvot binäärisiksi sarakkeiksi. Tämän avulla saadaan säästettyä kategoristen parametrien luonnollinen suhde toisiinsa. One-Hot encoding -menetelmää tulee erityisesti käyttää silloin, kun kategorisilla parametreilla ei ole suuruusjärjestystä. Jos kategorisilla arvoilla olisi suuruusjärjestys, esimerkiksi S, M, L ja XL, silloin One-Hot encoding menetelmä menettäisi parametrin sisäisen suuruusjärjestyksen. (Bownlee, 2017) One-Hot encoding -menetelmää hyödynnettiin muuntamaan kolme kategoria muotoista parametria. Tämän toimenpiteen haittapuolena on se, että datan parametrien määrä kasvoi merkittävästi. Datasettien yhdistämisen jälkeen käytettävissä oli 15 parametriä, mutta One-hot encoding -menetelmän jälkeen parametrejä oli 171. Eniten uusia parametreja One-hot encoding -menetelmän johdosta tuotti partnerin käyttämien API-rajapintojen listaus. Tämä listaus tuotti uusia parametreja 146 kappaletta.

Tietojen esikäsittelyn ja yhdistämisen jälkeen datasta suodatettiin pois epäaktiiviset partnerit. Epäaktiiviseksi partneriksi määriteltiin sellainen, jolla on alle viisi asiakasta, ja ei ole tehnyt yhtään rajapintakyselyä vuoden 2020 aikana, tai partnerin tuottaman kuukausituotto on alle 500 euroa. Partnereista suodatettiin myös Single-tason partnerit, tähän päädyttiin sen takia, koska case-yrityksen ISV-kumppani strategiaan ei kuulu Single-tason partnerien tarkka seuraaminen, koska niillä on suhteessa hyvin pieni vaikutus isossa kuvassa. Suodattamisen jälkeen jäljelle jäi 177 partneria.

ISV-kumppanien suodattamisen jälkeen tarkastelin datasetin parametrien välistä korrelaatiota, jotta voisi päätellä onko kaikki saatavilla olevat parametrit tarpeellisia. Erityisesti mielenkiintoa herätti, onko API-rajapintojen listaukset merkittäviä parametreja, ja onko niillä korrelaatiota muiden parametrien kanssa.Pythonin Pingouin-kirjaston avulla suoritettiin korrelaatio laskenta, joka laski korrelaation jokaisen parametrin välille hyödyntäen Spearmanin korrelaatiota. Päädyin käyttämään Spearmanin korrelaatiota, koska käytettävissä oleva data ei sopinut perinteiseen Pearsonin korrelaatioon. Pearsonin korrelaatiota varten datan olisi oltava intervalli tai suhdeluku muodossa. Kuten aiemmin mainittu, datan seassa on useampi kategorinen parametri, joten Pearsonin korrelaatio ei sopinut tähän tarkoitukseen. Korrelaatiokertoimen avulla pystyy selvittämään kahden muuttujan välistä riippuvuutta. Spearmanin

korrelaatiokerroin on tilastollinen menetelmä määrittämään kahden parametrin välisen monotonisen suhteen vahvuutta. Monotoninen suhde on sellainen, jossa molemmat parametrit muuttuvat toisen muuttuessa. Epämonotoninen suhde on sellainen, jossa parametrien arvot välillä nousevat ja välillä laskevat toistensa mukaan, toisin kuin monotonisessa suhteessa parametrien arvot aina pelkästään nousevat tai laskevat. Spearmanin korrelaatio kerroin asettuu -1 ja +1 välille. Mitä lähempänä korrelaatio kerroin on 1, sitä positiivisesti vahvempi suhde kahden parametrin välillä on ja taas mitä lähempänä korrelaatio kerroin on -1, sitä negatiivisesti vahvempi suhde on näillä parametreillä. Parametrien välistä korrelaatiota voidaan myös havainnollistaa verbaalisesti, jos korrelaatio kerroin on välillä:

- 0.00–0.19, suhde on erittäin heikko
- 0.20–0.39, suhde on heikko
- 0.40–0.59, suhde on keskinkertainen
- 0.60–0,79, suhde on vahva
- 0.80–1.0, suhde on erittäin vahva.

Sama verbaalinen asteikko toimii myös negatiivisille korrelaatio kertoimille, mutta silloin muutoksen suhde on negatiivinen. (Laerd.com, 2018, Laerd.com, 2012)

ISV-kumppanien käyttämät API-rajapintojen parametrit oli jaettu kahteen luokkaan. Jokaista rajapintaa kohden oli kaksi parametria, kuinka monta loppuasiakasta sitä käyttää ja kuinka monta rajapintakutsua kyseiseen rajapintaan asiakas on tehnyt. Kaikille parametreille laskettiin Spearmanin korrelaatiokertoimet ja vertasin, korreloiko ne esimerkiksi partnerin kuukausittaisen liikevaihdon kanssa. Suurimmat korrelaatiokertoimet kuukausilaskutuksen kanssa olivat vain 0.33 ja 0.35 välillä, joten korrelaation suhde oli heikko. Tästä syystä poistin datasta yksittäisiä API-rajapintoja kuvaavat parametrit, mutta jätin dataan edelleen parametrit, jotka kuvaavat kuinka monta loppuasiakasta käyttää partnerin rajapintoja yhteensä ja kuinka monta rajapinta kutsua niihin tehdään yhteensä. Tämän muutoksen ansiosta parametrien määrä väheni 171 parametrista 31 parametriin. Jatkokäsittelyn kannalta palautin partnerin partneritasoa ja tuotekategoriaa kuvaavat parametrit normaaliin kategoriseen muotoon, joka vähensi parametrien määrään 20 parametriin.

Jäljelle jääneille 20 parametrille tein vielä tarkistavana toimenpiteenä Phik korrelaatio laskennan avulla. Phik korrelaation laskenta perustuu aiemmin mainitun Pearsonin korrelaation mukaan, mutta sitä on kehitetty enemmän. Phik eroaa Pearsonin korrelaatiosta siten, että se pystyy käsittelemään kategorisia, ordinaalisia ja intervalli muotoisia parametreja. Pearson korrelaatiosta poiketen Phik korrelaatio pystyy myös selvittämään ei-lineaarisia korrelaatioita. Phik korrelaatio eroaa myös aiemmin mainituista Pearsonin ja Spearmanin korrelaatioista siten, että sen antamat korrelaatio kertoimet ovat nollan ja ykkösen välillä, tarkoittaen sitä, että Phik korrelaatio ei ota kantaa siihen, onko korrelaatio positiivinen vai negatiivinen. Phik korrelaation haittana on sen vaatima laskentateho, eikä se tästä syystä toimi datasettien kanssa, joissa on paljon attribuutteja. Python-ohjelmointikielelle tarkoitetun Phik-kirjaston avulla loin korrelaatiomatriisin ja koeffisienttimatrsiin, joka kuvastaa korrelaatioiden tilastollista merkitsevyyttä. (Baak et al., 2019; Lewinson,2021)

7.2 Datan esittely

Tässä luvussa käydään tarkemmin läpi tässä työssä käsiteltävän datan parametrit ja missä muodossa data on. Parametrien esittelyn jälkeen perehdytään aiemmin laskettuihin korrelaatioihin ja niiden merkitykseen työn jatkon kannalta. Alla olevan taulukon 5. ensimmäisessä sarakkeessa on listattu parametrin nimi, toisessa sarakkeessa on listattu missä muodossa data on ja kolmannessa sarakkeessa on tarkempi kuvaus mitä kyseinen tietue tarkoittaa.

Taulukko 5. Partneridatan listaus

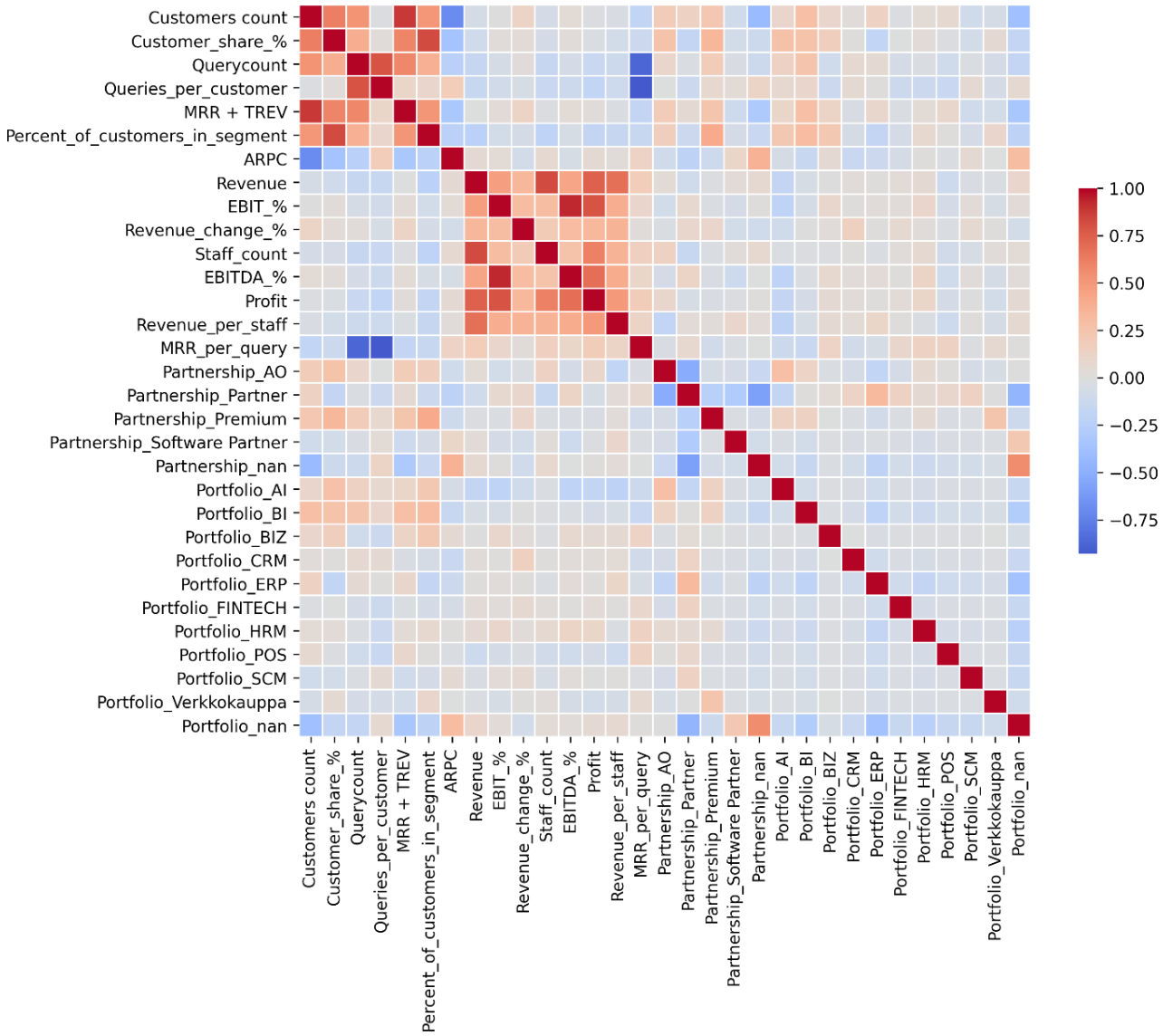
Nimi	**Formaatti**	**Kuvaus**
1. PartnerName	Teksti	Partnerin nimi
2. Portfolio	Teksti kategoria	Tuoteportfolio, johon partnerin tuote kuuluu
3. Partnership	Teksti kategoria	Kyseisen partnerin ISV-kumppaniohjelman taso
4. Product_name	Teksti	Integraatio tuotteen nimi

5. Company_name	Teksti	Partneri yrityksen virallinen nimi
6. Customers count	Kokonaisluku	Loppuasiakkaiden määrä
7. Customer_share_%	Desimaaliluku	Ekosysteemin markkinaosuus loppuasiakkaista
8. Querycount	Kokonaisluku	API-rajapintoihin tehtyjen kutsujen määrä
9. Queries_per_customer	Kokonaisluku	API-rajapintoihin tehtyjen kutsujen määrä suhteessa loppuasiakkaiden määrään
10. MRR + TREV	Kokonaisluku	Kuukausittainen liikevaihto ja transaktiomaksut
11. Percent_of_customers_in_segment	Desimaaliluku	Ekosysteemin markkinaosuus tietyssä asiakassegmentissä
12. ARPC	Kokonaisluku	Keskimääräinen liikevaihto per loppuasiakas
13. Revenue	Kokonaisluku	Partneriyrityksen liikevaihto
14. EBIT_%	Desimaaliluku	Partneriyrityksen liiketulos-%
15. Revenue_change_%	Desimaaliluku	Partneriyrityksen liikevaihdon kasvu
16. Staff_count	Kokonaisluku	Partneriyrityksen henkilökunnan määrä
17. EBITDA_%	Desimaaliluku	Partneriyrityksen käyttökate-%
18. Profit	Kokonaisluku	Partneriyrityksen tuotto
19. Revenue_per_staff	Desimaaliluku	Partneriyrityksen liikevaihto per työntekijä
20. MRR_per_query	Desimaaliluku	Kuukausittainen liikevaihto ja transaktiomaksut per API-rajapintaan tehty kutsu.

Datassa on kaksi parametria, jotka ovat tekstimuotoisia kategorioita eli Portfolio ja Partnership. Portfolio-parametri sisältää tiedon, mihin tuoteportfolioon kyseinen integraatiotuote kuuluu. Portfolio kategoriaan kuuluu ERP, BI, HRM, POS, AI, FINTECH, SCM, CRM, Verkkokauppa ja BIZ kategoriat. Yleisimmät kategoriat olivat ERP 21 %, BI 13 % ja HRM 10 %. Koko datasta portfolio tieto puuttui 35 % tuotteilta. Partnership-parametri on toinen tekstimuotoinen parametri, joka sisältää tiedon, mille partneritasolle kyseisen tuotteen partneri kuuluu. Aiemmassa kappaleessa 7.1. on esitelty tarkemmin case-yrityksen ISV-kumppaniohjelman partneritasot. 68 % tuotteiden partnereista kuului normaalille partner-tasolle, 11 % kuului AO-partneri tasolle, 4 % kuului Software Partneri-tasolle ja 2.8 % kuului Premium partneritasolle. Partnership tieto puuttui kuitenkin 14 % tuotteista.

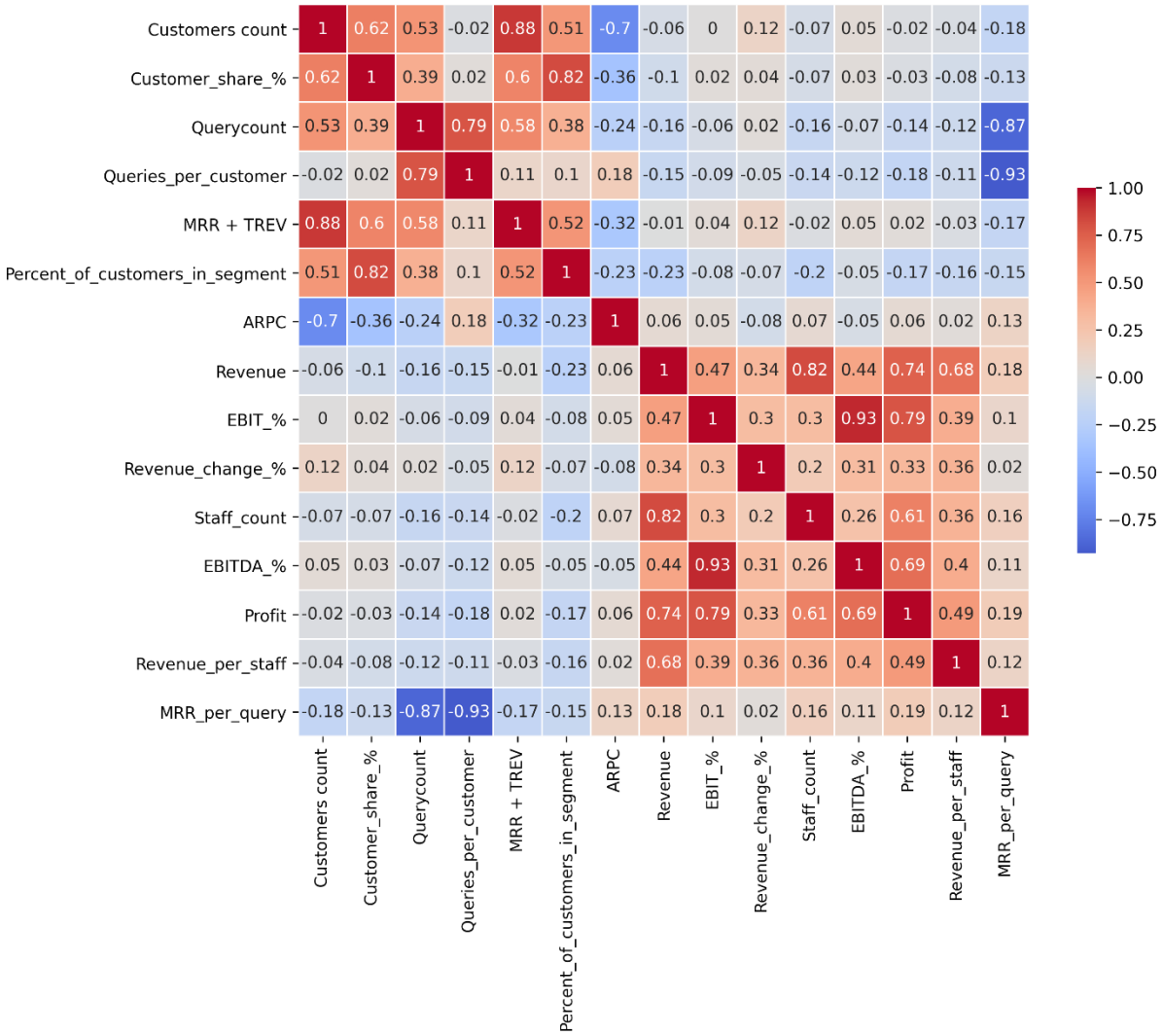
Partneriyrityksien taloustiedoissa on puutteita, mutta tarkoista puutteista ei ole tietoa, koska tietojen puuttumista ei ole erikseen ilmoitettu, vaan luvut on merkattu nollaksi. Esimerkiksi partneriyrityksen liikevaihto tieto on 4 % tapauksista nolla, mutta ei voida olla varmoja onko kaikilla liikevaihto oikeasti 0 euroa, vai onko tieto vain puutteellista. Nolla-arvoja löytyy, myös “EBIT_%”, “Revenue_change_%”, “Staff_count”, “EBITDA_%”, “Profit” ja “Revenue_per_staff” parametreista. Nolla-arvojen määrä on kuitenkin yleisesti ottaen 5–10 % välillä ja osassa parametreista nolla-arvot voivat todennäköisesti kuvata oikeitakin arvoja.

Alla olevassa kuvaajassa 9. on havainnollistettu parametrien välisiä korrelaatioita. Korrelaatiot on laskettu Spearmanin korrelaation perusteella. Kuvassa 9. tekstimuotoiset kategoriset parametrit on muokattu One-hot encoding menetelmällä. Kuvaaja on matriisimuotoinen, jossa X- ja Y-akseleilla on listattu kaikki datan sisältämät parametrit. Parametrien välille on laskettu Spearmanin korrelaatio ja korrelaatiokerrointa visualisoidaan värin perusteella. Väriskaala on -1 ja 1 välillä. Korrelaatiokertoimen ollessa lähellä -1, silloin parametriparia kuvastava neliö on sinisempi ja korrelaatiokertoimen ollessa lähellä 1, silloin parametriparia kuvastava neliö on punaisempi. Neutraalina, eli nolla-arvona on valkoinen väri. Kuvaajasta voidaan nähdä nopeasti, että kategorisilla arvoilla ei ole merkittäviä korrelaatioita, joten on selkeämpi tarkastella suoraan korrelaatioita, joista on poistettu kategoriset parametrit.



Kuva 9. Spearmanin korrelaatiomatrisi kategorisilla parametreilla

Alla olevassa kuvaajasta 10. on poistettu kategoriset parametrit. Korrelaatiomatriisiin on myös lisätty korrelaatiokertoimet, jotta korrelaatioiden havainnoiminen on selkeämpää. Parametriparien korrelaatiokertoimia kuvastaa myös väri, samalla tavalla kuin aiemmassa kuvaajassa 9. Korrelaatiomatriisista voidaan nähdä, että korrelaatiot ovat ryhmittäytyneet. Ryhmittäytyminen johtuu siitä, että alaosan ”Revenue – Revenue_per_staff”, parametrit kuvaavat partneriyrityksen taloustietoja ja yläosa ”Customers count – ARPC” kuvaa partnerin integraation tuottoa ja rajapinnan käyttöä.

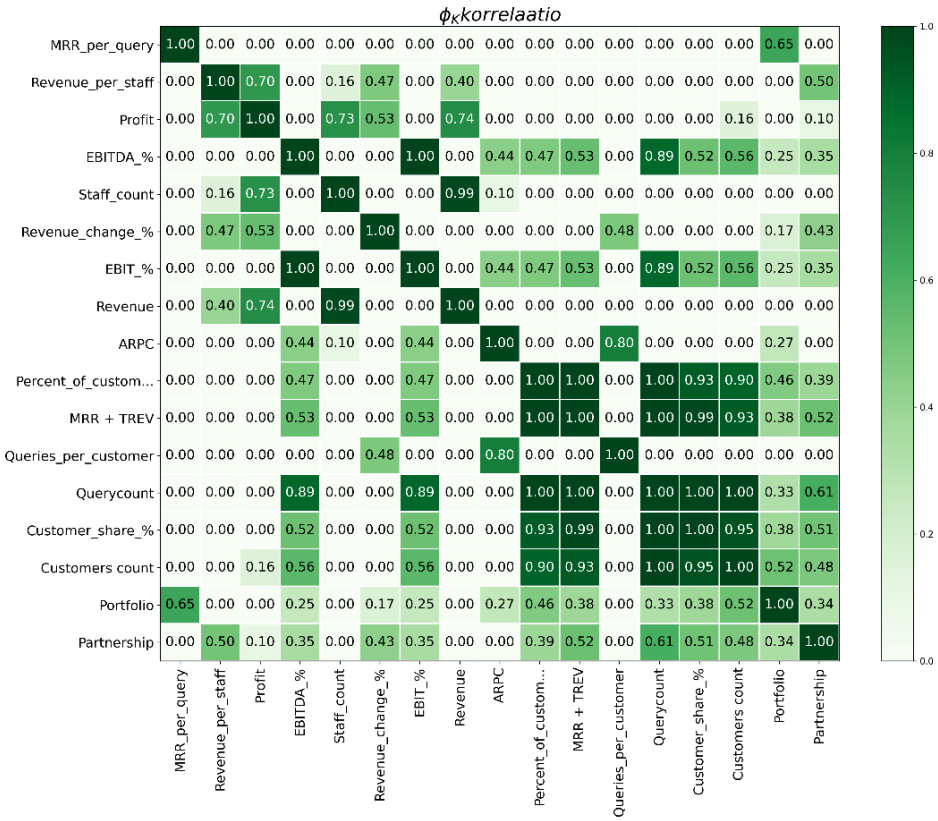


Kuva 10. Spearmanin korrelaatiomatriisi ilman kategorisia parametreja.

Ohjelmistoekosysteemin kannalta tärkeimpiä parametreja ovat ”Customers count”, ”MRR + TREV”, ”ARPC” ja partneriyrityksen kasvuluvut. ”Customers count” on tärkeä parametri sen takia, koska mitä enemmän loppuasiakkaita on käyttämässä integraatioita, sitä enemmän se luo ohjelmistoekosysteemiin tahmeutta. ”MRR + TREV” ja ”ARPC” ovat tärkeitä parametreja sen takia, koska niiden avulla voidaan seurata integraatioiden tuottamia rahavirtoja. ”ARPC” parametrista voidaan nähdä se, että se korreloi negatiivisesti -0.59 parametrin ”Customers count” kanssa. Tämä on hyvä havainto siinä mielessä, että yleisesti integraatiot, jotka ovat laajamittaisessa käytössä useilla loppuasiakkailla on yksinkertaisempia integraatioita, joiden laskutus on samalla alhaisempi ja toisin päin, monimutkaiset ja isot integraatiot ovat harvinaisempia, mutta niiden laskutus per asiakas on silloin taas merkittävästi isompi. ”MRR + TREV” parametrilla on vahva positiivinen korrelaatio +0.77 parametrin ”Customers count”

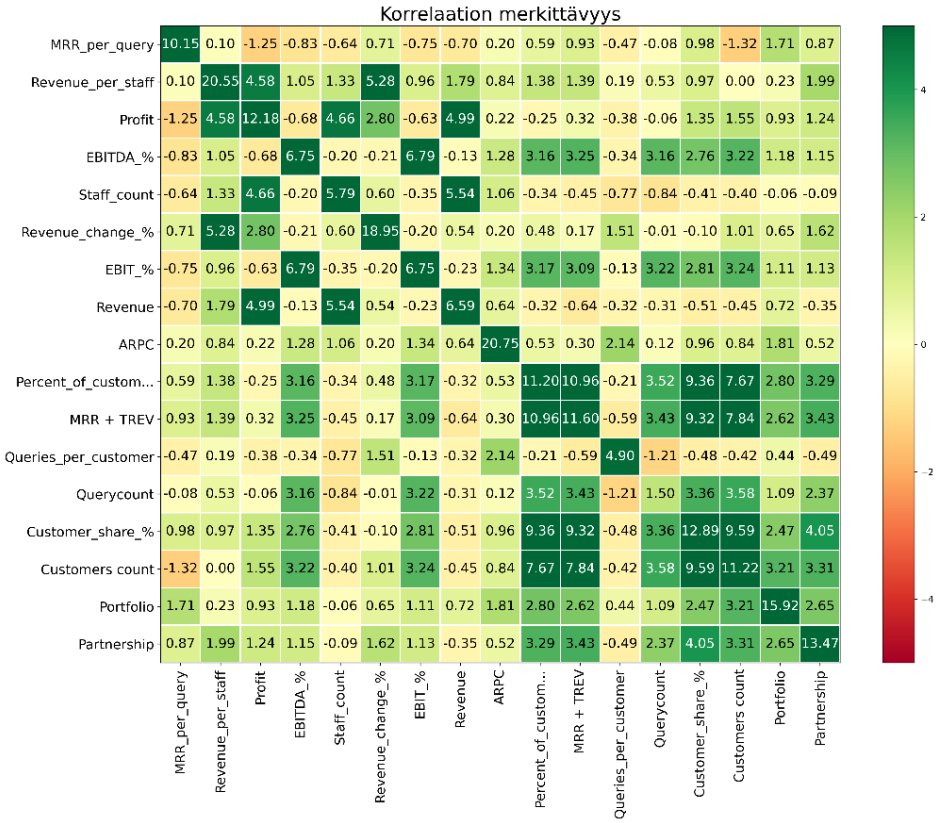
kanssa. Vahva positiivinen korrelaatio selittyy sillä, kun loppuasiakkaiden määrä integraatiota kohden kasvaa, silloin myös kasvaa kuukausittainen liikevaihto ja transaktioiden määrä samassa suhteessa. Vahvoja korrelaatioita löytyy myös integraatio partnereiden taloudellisten parametrien väliltä.

Mielenkiintoista olisi ollut, jos korrelaatioita olisi löytynyt taloudellisten tietojen ja rajanpintojen käyttöön liittyvien parametrien väliltä. Alla olevassa kuvassa 11. on aiemmin esitellyn Phik korrelaatiomenetelmän avulla laskettu korrelaatiomatriisi. Phik korrelaatiomatriisissa on nyt myös mukana tekstikategoriset parametrit ”Portfolio” ja ”Partnership” ilman One-hot encoding menetelmää. Phik korrelaatiomatriisista voidaan tehdä samoja johtopäätöksiä, kuin Spearmanin korrelaatiomatriisista. Phik korrelaatiomatriisista voidaan kuitenkin tarkastella helpommin partnerien portfolioon ja partneritasoon liittyviä korrelaatioita. Phik korrelaatiomatriisista voidaan huomata, että portfolio, johon partnerin tuote kuuluu, korreloi vahvasti (0.65) parametrin ”MRR per query” kanssa, joka oli integraation liikevaihto suhteessa rajapintaan tehtyihin kutsuihin.



Kuva 11. Phik korrelaatiomatriisi

Korrelaation tilastollinen merkittävyys voidaan tarkastaa kuvaajasta 12. Tilastollinen merkittävyys portfolion ja ”MRR per query” parametrilla on matala, koska sen arvo on vain 1.71. Parametrit ovat tilastollisesti merkittävästi toisiinsa korreloivia, jos niiden tilastollinen merkittävyys on yli 2.5. Tilastollisen merkittävyyden perusteella voidaan todeta, että ISV-kumppanin portfolio korreloi asiakasmäärän, liikevaihdon ja markkinaosuuden kanssa. ISV-kumppanin partneritasolla on hyvin samanlaiset korrelaatiot, kuin portfolio kategorialla, mutta ne ovat tilastollisesti merkittävämpiä. Tämä eroavaisuus voi selittyä sillä, että partneritasoja on vähemmän, kuin portfolioita. Mielenkiintoinen löydös Phik korrelaatiosta on se, että ISV-kumppanin liikevoitto-% ja käyttökate-% korreloi rajapinnan käyttöön liittyvien parametrien kanssa, kuten asiakasmäärä, rajapintakutsujen määrä, markkinaosuus ja rajapinnan tuottaman liikevaihdon kanssa. Kyseiset korrelaatiot ovat tilastollisesti merkittäviä. Phik korrelaatio ei kuitenkaan ota kantaa, mihin suuntaan parametrit korreloivat.



Kuva 12. Phik korrelaatiomatriisin tilastollinen merkittävyys matriisi.

7.3 ISV-kumppanien klusterointi

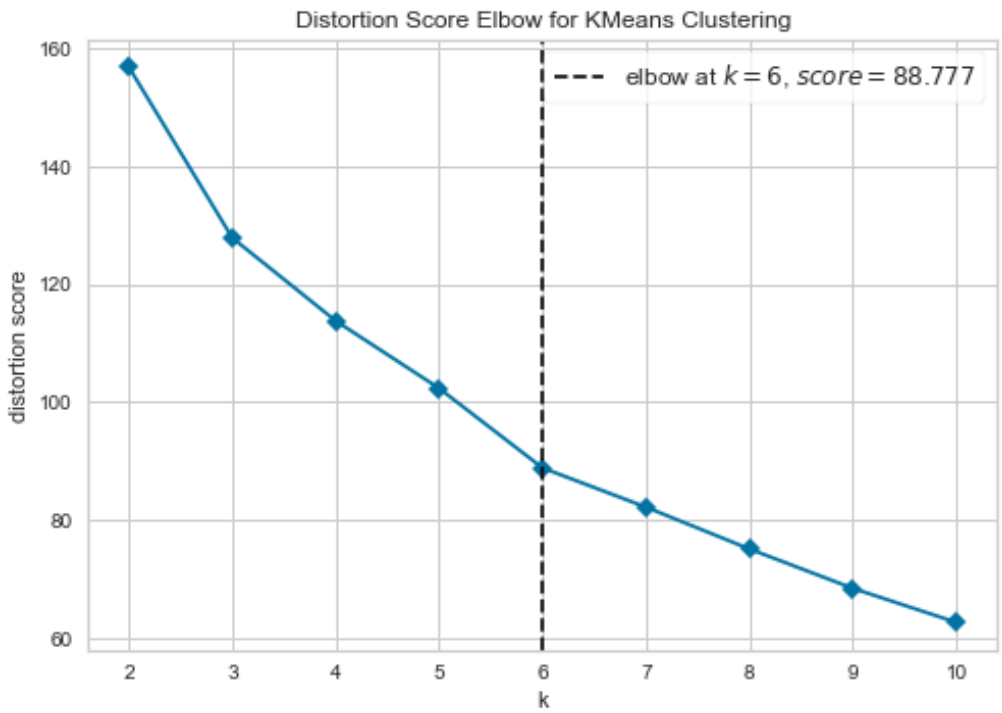
Tässä kappaleessa käsitellään partnerien klusteroimista ja perehdytään siihen, kuinka aiemmassa kappaleessa esiteltyä dataa voidaan hyödyntää klusteroimisessa ja käydään tarkemmin läpi klusteroimisen lopputuloksia ja lopputuloksien hyödyllisyyttä. Tutkimuskysymyksen 2. hypoteesina oli pyrkiä selvittämään, pystyisikö klusteroinnilla jakamaan isompia partneritasoja pienempiin joukkoihin. Crispim ja Sousa (2009) hyödynsivät klusterointia heidän artikkelissansa, jossa klusteroitiin yrityksiä aiemmin määrättyjen kriteerien pohjalta.

Datan käsittelyn ja tutkimisen jälkeen lähdin testaamaan erilaisia klusterointialgoritmeja ja niiden tehokkuutta ja toimivuutta. Datan klusterointi toteutettiin Python-ohjelmointikielellä. Klusterointia varten hyödynsin Pythonin PyCaret kirjastoa, joka mahdollistaa helposti ja useiden eri koneoppimisalgoritmien testaamisen ja tulosten tarkastelun. PyCaret on avoimen lähdekoodin kirjasto, jonka tarkoituksena on vähentää tarvittavaa koodin määrää, kun hyödynnetään koneoppimista. (PyCaret, 2021)

Ennen ISV-kumppanien datan klusteroimista datasta poistettiin muutamia attribuutteja. Poistettavia attribuutteja olivat ”PartnerName”, ”Product_name” ja ”Company_name”. Kyseiset attribuutit ovat jokaiselle tietueelle uniikkeja ja tarkoitettu vain identifioimaan muita attribuutteja. Tämän jälkeen datajoukko normalisoitiin hyödyntämällä minmax-algoritmia. Datan normalisointi suoritettiin siitä, syystä että voidaan varmistua siitä, että data on standardi muodossa. Mohamadin ja Usmanin (2013) tekemässä tutkimuksessa todettiin, että datan standardisointi ennen klusterointia johtaa tarkempiin lopputuloksiin, tehostaa käsittelyä ja klusterointi saa tarkemmat tulokset. Min-Max normalisointi on prosessi, jossa kaikki attribuutit skaalataan nollan ja ykkösen välille. Attribuutin pienin arvo (min) saa luvukseen 0.0 ja suurin (max) 1.0 ja loput arvot lasketaan kaavan 6. avulla. Normalisointi helpottaa attribuuttien vertailua keskenään, jos niiden alkuperäiset skaalat ovat merkittävästi erilaiset.

$$MM(X_{ij}) = \frac{X_{ij} - X_{min}}{X_{max} - X_{min}} \tag{6}$$

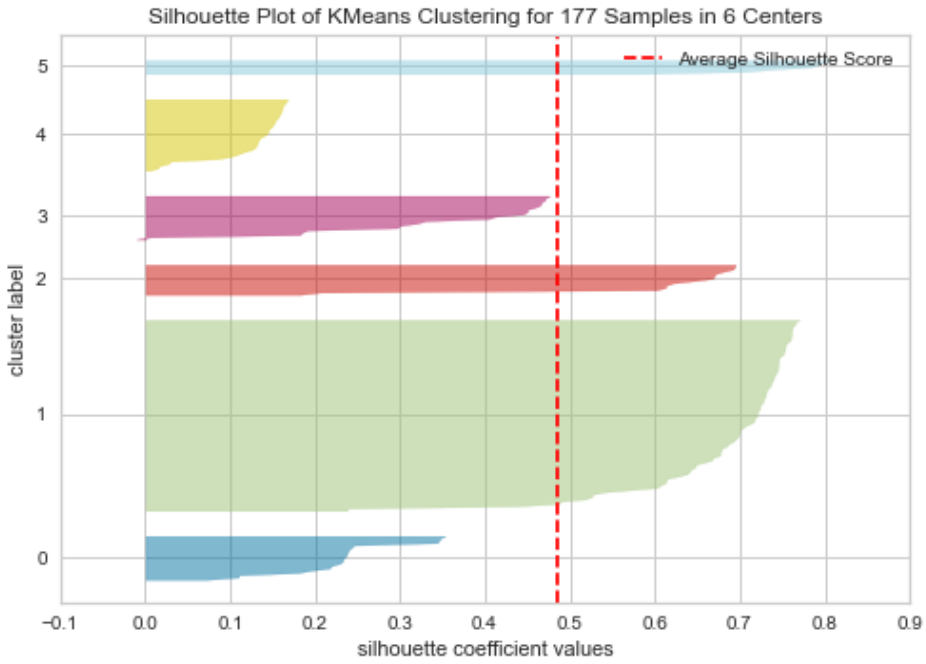
Datan normalisoinnin jälkeen klusterointia varten täytyi määrittää optimaalinen klusterien määrä. Optimaalisen klusterien määrän pystyy selvittämään kyynärpää kuvaajan avulla. Kyynärpääkuvaajan avulla voidaan katsoa optimaalisin klusterien määrä siitä kohtaa kuvaajasta, jossa käyrän lasku taittuu tasaisemmaksi. Graafisesti tarkasteltuna kyynärpää menetelmän kuvaajan luoma käyrä näyttää kyynärpään muotoiselta. Alla olevassa kuvaajassa 13. on havainnollistettu Python-ohjelmointikielen PyCaret kirjastolla luotua kuvaajaa K-Means klusterointialgoritmin kyynärpään kuvaajalle. Kuvaajasta voidaan nähdä, että optimaalisin klusterien määrä on kuusi. Kyynärpää kuvaaja on klusterointialgoritmikohtainen ja optimaalinen klusterien määrä voi vaihdella algoritmista riippuen. (Mohamad & Usman, 2013)



Kuva 13. PyCaret kirjastolla luotu kuvaaja K-means algoritmin kyynärpääkuvaajasta.

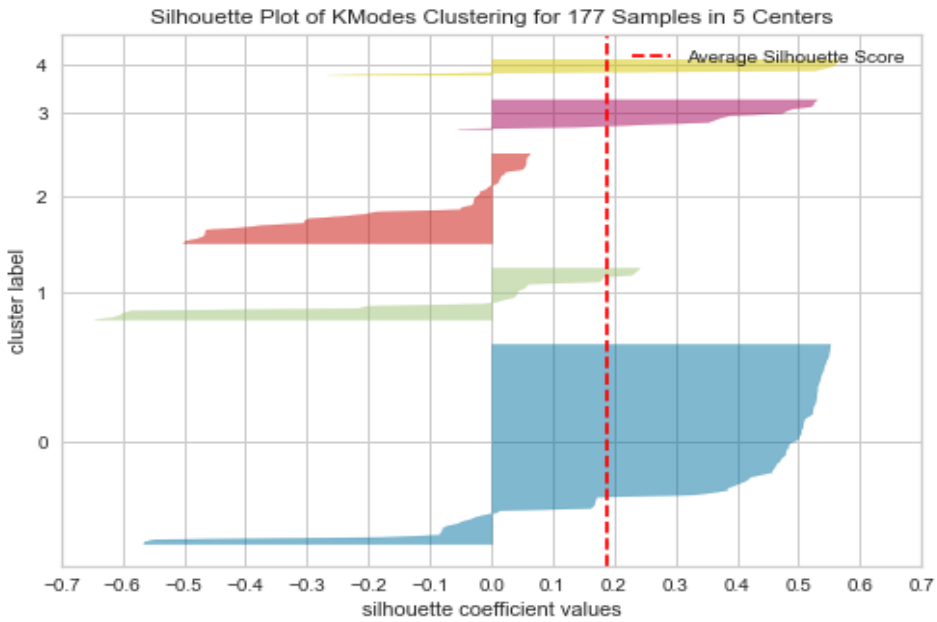
Optimaalisen klusterimäärän selvittämisen jälkeen suoritettiin datan klusterointi PyCaret kirjaston avulla. Optimaalisen klusterimäärän valinta voidaan tarkastaa siluetti kuvaajan avulla. Siluettikuvaajan avulla voidaan tarkastella datapisteiden keskimääräistä etäisyyttä viereisten klusterien datapisteisiin. Siluettikoeffisientti on -1 ja 1 välillä, sen ollessa 1 kyseinen datapiste on erittäin kaukana viereisestä klusterista. Siluettikoeffisientin 0 arvo tarkoittaa, että datapiste on hyvin lähellä viereistä klusteria ja negatiiviset arvot tarkoittavat, että kyseinen datapiste

saattaa olla väärässä klusterissa. Siluettikuvaajassa tarkastellaan datapisteiden keskiarvoa, ja mitä suurempi siluettikoeffisienttien keskiarvo on, sen parempi, koska silloin klusterit ovat mahdollisimman erillään toisistaan. Alla olevassa kuvaajassa 14. on K-Means algoritmilla tuotettu siluettikuvaaja. Kuvaajasta voidaan nähdä, että siluettikoeffisientin keskiarvo on noin 0.5, eli kohtalaisen hyvä. Merkittävän huomiona on se, että mukana ei ole lähes yhtään negatiivista arvoa. Kolmannessa, eli violetissa klusterissa on nähtävillä, yksi datapiste, joka sijoittuu siluettikoeffisientti asteikolla alle nollan. (Scikit-learn, 2020)



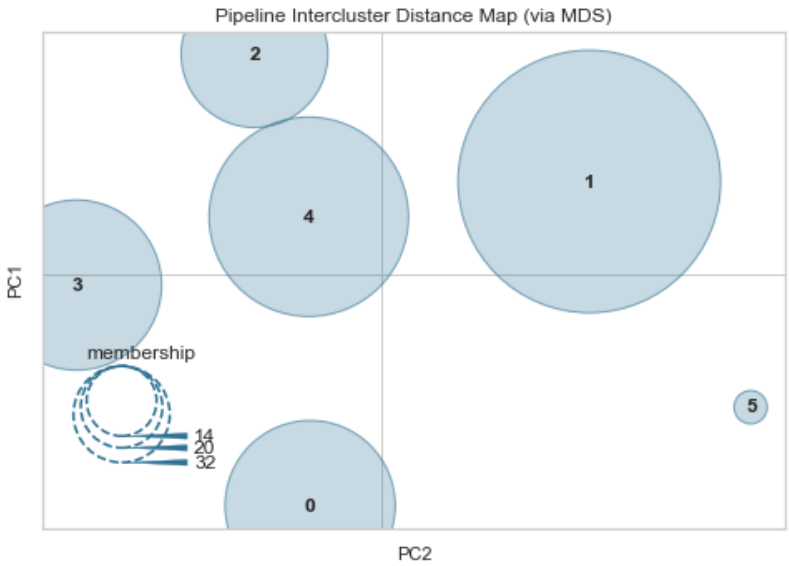
Kuva 14. Siluettikuvaaja K-Means klusterointialgoritmille.

Klusterointia testattiin myös K-Modes menetelmän avulla. K-Modes menetelmässä optimaalinen klusterien määrä oli viisi klusteria. Klusterien optimaalisen määrän jälkeen ajettiin koneoppimismallin opetus ja tarkasteltiin siluettikuvaajan tuloksia. Alla olevasta kuvaajasta 15. voidaan heti nähdä, että klusterit sisältävät negatiivisia siluettikoeffisientin arvoja, eli datapisteet kuuluvat mahdollisesti vääriin klustereihin. Negatiiviset siluettikoeffisientin arvot vaikuttavat myös klusterien siluettikoeffisienttien keskiarvoihin ja se on K-modes algoritmin kohdalla vain noin 0.2, eli aika huono. Tämän pohjalta voidaan todeta, että kyseisen datan klusteroinnissa K-means algoritmi toimii merkittävästi paremmin.



Kuva 15. Siluettikuvaaja K-Modes klusterointialgoritmille.

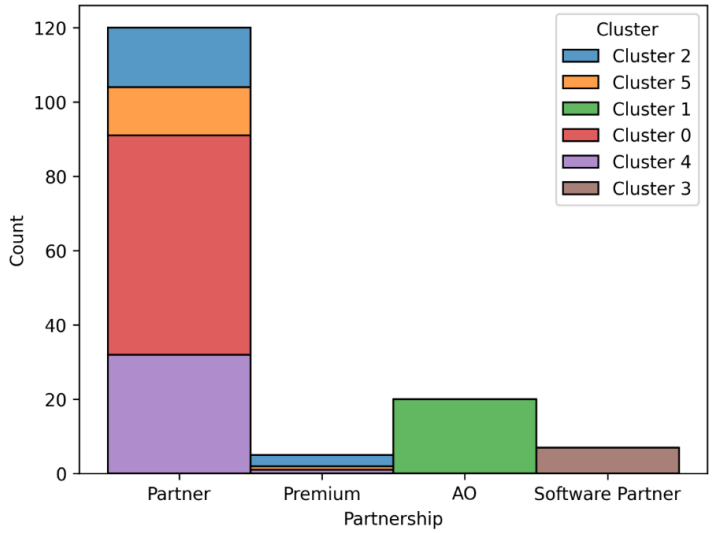
Alla olevassa etäisyys kuvaajassa 16. on havainnollistettu klusterit kaksiulotteisessa tasossa, jossa on säilytetty klusterien keskipisteiden väliset etäisyydet toisiinsa. Kuvaajan 15. Y-akseli havainnollistaa ensimmäistä pääkomponenttia ja X-akseli havainnollistaa toista pääkomponenttia. Pääkomponentit on luotu PyCaret kirjaston avulla hyödyntäen pääkomponenttianalyysiä. Pääkomponenttianalyysi on menetelmä, jossa vähennetään datan parametrien määrää tiivistämällä niitä erilaisten menetelmien avulla (Joliffe & Cadima, 2016). Pääkomponenttien avulla pystytään helpommin visualisoimaan klusterien sijaintia suhteessa toisiin klustereihin.



Kuva 16. K-Means klusterien etäisyyskuvaaja.

Kuvaajassa olevat ympyrät havainnollistavat klustereita ja ympyröiden koot määräytyvät sen mukaan, kuinka monta datapistettä klusteriin kuuluu. Vaikka klusterit olisivat kaksiulotteisessa tasossa päällekkäin, se ei tarkoita, että klusterit olisivat päällekkäin normaalitilanteessa. Etäisyyskuvaajan avulla voidaan huomata, että klusterit ovat suhteessa toisiinsa lähes samaa kokoluokkaa ja selkeästi erillään toisistaan, kuten voitiin nähdä jo aiemmin K-Means siluettikuvaajan perusteella. Poikkeuksena tähän on viides klusteri, joka on muita klustereita pienempi. (Yellowbrick, 2021)

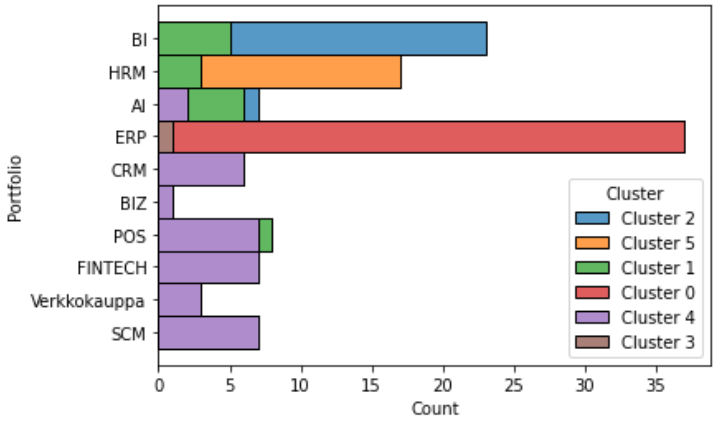
Siluettikuvaajan ja etäisyyskuvaajan perusteella voidaan päätellä, että klusterointi on onnistunut hyvin K-Means algoritmin avulla. Seuraavaksi tarkastellaan kuvaajien avulla, miten ISV-kumppanit ovat klusteroitu. Klusteroinnissa käytettiin aiemmin luvussa 7.2. esiteltyä dataa ja kaikkia sen attribuutteja, pois lukien ISV-kumppanien nimet. Alla olevan kuvaajan 17. avulla voidaan tarkastella, miten eri ISV-kumppanien partneritasot ovat jakautuneet eri klustereihin.



Kuva 17. Partneritasojen jakautuminen klustereihin

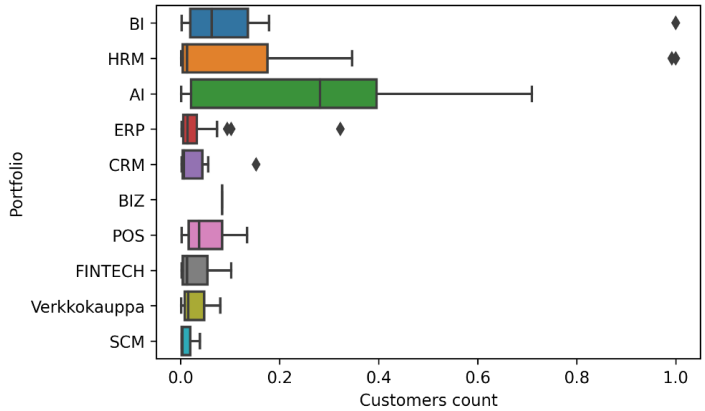
Kuvaajasta voidaan nähdä, että isoin partneritaso, eli ”Partner” on jakautunut neljään erilliseen klusteriin. ”Premium” partneritason ISV-kumppanit ovat jakautuneet kolmeen erilliseen klusteriin. ”AO” partneritason ISV-kumppanit ovat jakautuneet kokonaan omaksi klusterikseen, samalla tavalla kuin ”Software Partner” tason ISV-kumppanit.

Alla olevassa kuvaajassa 18. on havainnollistettu, minkä eri portfolion ISV-kumppaneita klusterit sisältävät. Kuvaajasta voidaan heti havaita, että Klusteri 0 sisältää pelkästään ERP-portfolion ISV-kumppaneita. Samalla tavalla Klusteri 5 sisältää vain HRM-portfolion ISV-kumppaneita, Klusteri 2 sisältää BI- ja AI-portfolion kumppaneita. Klusteri 1 sisältää useita eri portfolioita, mutta se johtuu selkeästi siitä, koska se sisälsi pelkästään ”AO”-tason ISV-kumppaneita, joita löytyy useasta eri portfoliosta. Klusteri 4 pitää sisällään käytännössä, kaikki muut pienemmät portfoliot, joissa on lukumäärällisesti huomattavasti vähemmän ISV-kumppaneita, kuin muissa portfolioissa. Klusterointi on siis jakanut omiksi klustereiksiin muutamat kumppanitasot, yhdistänyt useampia pienempiä portfolioita yhdeksi joukoksi ja jakanut isommat portfoliot omiksi klustereiksi.



Kuva 18. Portfolioiden jakautuminen klustereihin.

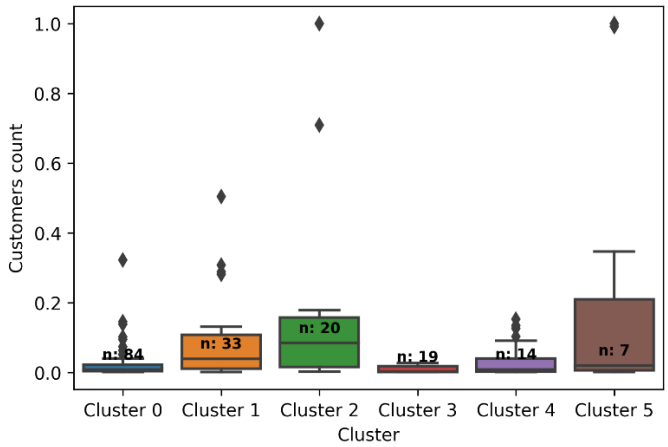
Alla olevassa kuvaajassa 19. on havainnollistettu portfolioihin kuuluvien ISV-kumppanien loppuasiakkaiden määrää laatikkokuvaajien avulla. Y-akselilla on kuvattu kaikki portfoliot ja X-akselilla on tietojen anonymisoinnin vuoksi näytetty vain normalisoitu luku loppuasiakkaiden määrästä. Kuvaajasta voidaan heti nähdä, että loppuasiakkaiden määrät eivät ole tasaisesti jakautuneet eri portfolioiden välillä, vaan niissä on merkittäviä eroja. Mielenkiintoisena huomiona aiempaan on se, että Klusteri 4 sisälsi ISV-kumppani määrältä pienempiä portfolioita, mutta loppuasiakkaiden suhteen Klusteri 4 on hyvin onnistunut keräämään pienet portfoliot ja pienet loppuasiakasmäärät yhteen joukkoon.



Kuva 19. Laatikkokuvaaja portfolioiden loppuasiakkaiden määrän jakautumisesta.

Alla olevassa kuvaajassa 20. on kuvattu laatikkokuvaajien avulla klusteri-tasolla loppuasiakkaiden määrää. Y-akselilla on loppuasiakkaiden määrä normalisoidussa muodossa ja X-akselilla on jokainen klusteroinnilla luotu klusteri. Klusterien loppuasiakasmäärät ovat selkeästi jakautuneet kahteen osaan. Klusterissa 0, 3 ja 4 on huomattavasti vähemmän

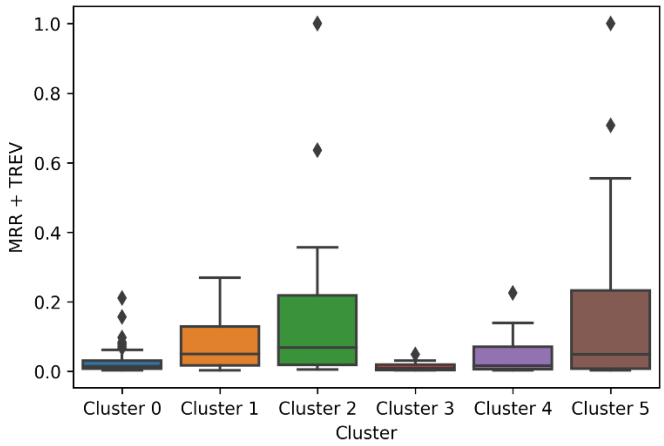
loppuasiakkaita, kuin klustereissa 1,2 ja 5. Selkeästi datajoukossa on ISV-kumppaneita, joiden loppuasiakasmäärät ovat muuhun joukkoon suhteutettuna merkittävästi suurempia, joten laatikkokuvaajia tarkastellessa voidaan nähdä, että klusterit sisältävät selkeästi joukosta poikkeavia lukuja. Näitä poikkeavia lukuja ei tässä tapauksessa voida poistaa, koska ne ovat oikeita partnereita ja oikeita lukuja, mutta heidän loppuasiakas määrät ovat vain merkittävästi suurempia. Jokaisen klusterin kohdalla on myös kirjattu ISV-kumppanien lukumäärä kyseisessä klusterissa. Klusterissa 0 on eniten ISV-partenereita, eli 84 kappaletta. Klusterissa 1 on 33 ISV-kumppania, klusterissa 2 on 20, klusterissa 3 on 12, klusterissa 4 on 14 ja klusterissa 5 on vähiten, eli 7 ISV-kumppania. Klusteri 0 sisältää sen takia merkkittävästi enemmän ISV-partenereita, kuin muut klusterit koska se on keskittynyt vain ERP-portfolion ISV-kumppaneihin ja se on portfolioista kaikista suurin.



Kuva 20. Loppuasiakkaiden määrät klustereissa.

Alla olevassa kuvaajassa 21. on laatikkokuvaajan avulla havainnollistettu ISV-kumppanien tuottamaa liikevaihtoa jokaisessa klusterissa. Y-akselilla on ”MRR + TREV”-parametri, joka kuvaistaa kuukausittaista liikevaihtoa ja transaktiomaksuja. Y-akseli on normalisoitu anonymisoinnin takia. X-akselilla on kaikki klusteroinnin luomat klusterit. Eniten tuottavat partnerit kuukausitasolla löytyy klustereista 2,5 ja 1. Kuten aiemmasta kuvaajasta nähtiin, kyseiset klusterit sisältävät myös loppuasiakasmäärällisesti isoimmat ISV-kumppanit. Loppuasiakkaiden määrä huomattiin aiemmin korreloivan vahvasti parametrin ”MRR + TREV” kanssa. Klusterit 4, 3 ja 0 tuottavat kuukausittain suhteessa vähemmän per ISV-

kumppani, mutta se johtuu selkeästi loppuasiakkaiden määrästä per ISV-kumppani. Lukumäärällisesti näissä klustereissa on merkittävästi suurin osa ISV-kumppaneista. Klusterien avulla voidaan huomata, että suurin osa ISV-kumppanien tuotoista tulee tietyistä pienemmistä klustereista.



Kuva 21. Klusterien MRR + TREV

7.4 ISV-kumppanien priorisointi

Tässä luvussa perehdytään siihen, kuinka määritellä case-yrityksen tapauksessa tärkeimmät ISV-kumppanit ja kuinka kirjallisuudesta olevan tiedon avulla pystyttäisiin luomaan ISV-kumppaneista oma ranking, jota voidaan käyttää työkaluna ISV-kumppanien priorisoinnissa. ISV-kumppanien parhaimmuusjärjestystä varten on käytössä aiemmissa kappaleissa esitelty data, joka sisältää käsiteltyä dataa kaikista case-yrityksen Netvisor tuotteen ISV-kumppaneista ja heidän tuotteistansa.

Kirjallisuudesta löytyi paljon materiaalia ohjelmistoekosysteemeihin ja tarkemmin vielä kumppaniekosysteemin roolista koko ohjelmistoekosysteemiin. Kirjallisuudesta löytyi hyvin informaatiota siitä, mitkä ovat kumppaniekosysteemin hyödyt kaikille osapuolille. Tämän jälkeen lähdin syventymään ISV-kumppaneihin ja siihen liittyvään kirjallisuuteen erityisesti ISV-kumppanien valintaprosessiin ja priorisointiin liittyen.

Haastavaksi tekijäksi ilmeni, se että kirjallisuutta ISV-kumppanien priorisointiin valintaprosessin jälkeen oli erittäin vaikea löytää, ja sitä oli vain hyvin rajallisesti saatavilla.

ISV-kumppanien valinta kriteereistä on kasattu luvussa 2.2. taulukko 2., jossa on tuotu esille kategorioittain erilaisia kriteereitä, joiden avulla partnereita pystyisi priorisoimaan ja hyödyntämään valintaprosessissa. ISV-kumppanien priorisointia kirjallisuudessa tuotiin esille menetelmä, jossa ISV-kumppaneita pisteytettiin heidän suoriutumisensa perusteella valittujen kriteerien pohjalta ja ISV-kumppanit laitettiin saatujen pisteiden perusteella parhaimmuusjärjestykseen. Kirjallisuuden pohjalta oli tarkoitus etsiä menetelmä, jonka avulla voidaan hyödyntää case-yrityksen ISV-kumppaneista saatavilla olevaa dataa, ja verrata ISV-kumppaneita keskenään ja luoda pisteytys datan pohjalta ja pisteytyksen avulla laittaa ISV-kumppanit parhaimmuus järjestykseen.

Työn teoriaosuudessa käsiteltiin usean kriteerin päätösanalyysiä ja sen menetelmiä. Usean kriteerin päätösanalyysin avulla voidaan edesauttaa päätöksentekoa pisteyttämällä päätöksen eri vaihtoehtoja ennalta määrättyjen kriteerien pohjalta. Usean kriteerin päätösanalyysiä voidaan hyödyntää monessa eri käyttötarkoituksessa ja työn teoriaosuudessa esiteltiin erilaisia käyttökohteita. ISV-kumppanien priorisoinnissa yhdistyy useat eri päätökseen vaikuttavat kriteerit ja ISV-kumppaneista saatavilla oleva data sopii usean kriteerin päätösanalyysiä varten. Usean kriteerin päätösanalyysissä on useita erilaisia menetelmiä, joiden avulla voidaan määrittää vaihtoehtojen joukosta paras vaihtoehto. Usean kriteerin päätösanalyysi jaetaan kolmeen vaiheeseen. Ensimmäisenä on datan ja informaation kerääminen, toinen vaihe on vaihtoehtojen ja kriteerien määrittäminen. Kolmas vaihe on parhaimman vaihtoehdon valinta.

Ensimmäinen vaihe on työn aiemmassa vaiheessa tehty, jossa data kerättiin ja käsiteltiin ISV-kumppanien klusterointia varten. ISV-kumppaneista saatavilla oleva data on nähtävillä luvun 7.2. taulukossa 5. saatavilla oleva data on hyvin rajallista ja usean kriteerin päätösanalyysin toisessa vaiheessa on tarkoitus määrittää vaihtoehdot ja kriteerit. Päätöksenteon vaihtoehdot ovat jo ISV-kumppaneista saatavilla olevassa datassa mukana, koska vaihtoehtoina ovat kaikki ISV-kumppanit. Kriteerien määrittämistä rajoitti merkittävästi saatavilla oleva data. Kirjallisuudessa oli paljon erilaisia kriteereitä, joita pystyy hyödyntämään ISV-kumppanin kyvykkyyksien ja potentiaalien arvioinnissa, mutta tässä tapauksessa rajoittavana tekijänä oli saatavilla oleva data.

Kirjallisuudessa keskityttiin hyvin paljon vertailemaan ISV-kumppaneita valintaprosessin aikana, jolloin on helppo selvittää ja kysyä potentiaaliselta ISV-kumppanilta vastaukset haluttuihin valinta kriteereihin. Tässä tapauksessa ei ole mahdollista selvittää puuttuvia tietoja, koska tarkoituksena on verrata nykyisiä ISV-kumppaneita tämänhetkisen tiedon perusteella ja lopputuloksen on tarkoitus olla sellainen, jota voidaan päivittää automaattisesti, ilman että se vaatii manuaalista työtä.

Ensin kriteerien selvittäminen aloitettiin siitä, miten tällä hetkellä case-yrityksessä hoidetaan ISV-kumppanien valintaprosessia. Valintaprosessissa hyödynnetään hyvin samanlaisia kriteereitä, kuin kirjallisuudessa mutta iso osa kriteereistä vaatii ammattilaisen arvioimista, eikä niihin ole suoria vastauksia saatavilla, esimerkiksi on hyvin vaikea arvioida ilman substanssiosaamista, kuinka hyvin tietty ISV-kumppani sopisi tiettyyn portfolioon, tai on erittäin vaikea arvioida ilman ammattilaisen arviointikykyä, kuinka sitoutunut potentiaalinen ISV-kumppani olisi.

Tästä johtuen, kriteerit määriteltiin täysin sen datan pohjalta, jota oli saatavilla. Saatavilla oleva data käytiin case-yrityksen Netvisorin PAM:ien (eng. Partner Account Manager) kanssa läpi ja he pistivät datan pohjalta luodut kriteerit tärkeysjärjestykseen. Tämän tärkeysjärjestyksen avulla pystytään määrittämään tarvittavat tiedot usean kriteerin päätösanalyysin kolmatta vaihetta varten. Viideksi tärkeimmäksi parametriksi valittiin loppuasiakkaiden määrä, integraation liikevaihto, keskimääräinen liikevaihto per loppuasiakas, ISV-kumppanin osuus asiakassegmentistä ja ISV-kumppanin yrityksen liikevaihdon kasvu edeltävästä vuodesta. Kaikki edellä mainitut parametrit valittiin tärkeimmiksi kriteereiksi ja tavoitteena on niiden maksimointi.

Usean kriteerin päätösanalyysin kolmannessa vaiheessa suoritetaan vaihtoehtojen vertailu keskenään aiemmin valittujen kriteerien ja vaihtoehtojen pohjalta. Kirjallisuudessa on esitetty useita erilaisia menetelmiä vaihtoehtojen vertailua varten ja niitä on esitelty tämän työn luvussa 5. case-yrityksen ISV-kumppanien vertailua varten parhaimmaksi menetelmäksi valikoitui SAW-menetelmä, koska se on yksinkertainen toteuttaa, käytettävissä oleva data sopii kyseiseen menetelmään, ottaa huomioon datassa olevat suuruuserot. Isoimpana tekijänä on SAW-menetelmän yksinkertaisuus, koska se mahdollistaa jatkossa helpon jatkokehityksen, koska

kriteerien painokertoimien muuttaminen ja tuloksien iteroiminen on nopeaa ja helppoa. Kaikilla menetelmillä oli omat hyvät- ja huonot puolensa, mutta muut menetelmät olivat, joko laskentatehollisesti merkittävästi vaativampia isompien datamassojen kanssa, tai menetelmät olivat monimutkaisempia, jolloin niiden jatkokehitys ja iteroiminen on hitaampaa.

SAW-menetelmän valinnan jälkeen seuraavaksi piti luoda menetelmä, jonka avulla se voidaan hyödyntää valmiiksi käsiteltyyn dataan ISV-kumppaneista. Yksinkertaiseksi vaihtoehdoksi valikoitui Python-ohjelmointikielellä implementoitu ratkaisu, jossa hyödynnetään SAW-menetelmän laskentalogiikkaa ja aiemmin esiteltyjä kaavoja 1–6. Ennen SAW-menetelmän suorittamista, täytyi määrittää sopivat painokertoimet valituille kriteereille. Painoarvot valittiin yhdessä Netvisori:n PAM:ien kanssa siten, että jokainen sai määritellä itse sopivat painokertoimet ja annetuista painokertoimista laskettiin keskiarvo. Painokertoimet olivat desimaalilukuja nollan ja ykkösen väliltä siten, että niiden yhteenlaskettu summa olisi tasan yksi. Tarkkoja painokertoimia ei voida työn anonymiteetin takia esittää.

SAW-menetelmän toteutus alkaisi normalisoimalla data, mutta se oli tehty työn aikaisemmassa vaiheessa. Tarkoituksena oli laskea neljä eri parhaimmuusjärjestystä, joten datasta tehtiin neljä kopiota. Data kopioitiin klusterien, portfolion, partneritason perusteella ja neljäs data sisälsi kaikki ISV-kumppanit. Tämän jälkeen jokainen datan tietue iteroitiin läpi. Jokainen tietueen parametri kerrottiin sille tarkoitetulla painokertoimella. Tässä analyysissä priorisoitiin vain viittä parametria, joten muut parametrit saivat painoarvoksi nollan, jolloin näiden parametrien lopulliseksi arvoksi tuli myös nolla. Viimeiseksi laskettiin yhteispisteet laskemalla yhteen kaikkien parametrien saamat arvot.

SAW-menetelmän toteutusta on havainnollistettu liitteissä 2. ja 3. ne sisältävät tulostaulukon, joka on jaettu kahteen osaan. Tulostaulukko pitää sisällään normalisoidut parametrien arvot, jotka kerrottiin niille annetuilla painokertoimilla. Kertolaskujen summaamisen jälkeen saatiin neljä eri pistesummakategoriaa, partneri-, portfolio, klusteri ja pistekategoria. ISV-kumppaneista laskettiin neljä eri pistesummaa, joista saadaan muodostettua parhaimmuusjärjestys.

Ensimmäisenä ISV-kumppanit laitettiin parhaimmuusjärjestykseen portfolio ja partneritasolla, josta saatiin partnerikategorian pisteytys ja portfoliokategorian pisteytys. Seuraavaksi ISV-kumppanit laitettiin parhaimmuusjärjestykseen aiemmin luotujen klusterien pohjalta, jonka tuloksena saatiin klusterikategorian pisteytys. Viimeisenä laskettiin kaikille ISV-kumppaneille yhteinen parhaimmuusjärjestys, tämä vaihe jätettiin viimeiseksi sen takia, että tämä mahdollistaa tarvittaessa aiemmin laskettujen parhaimmuusjärjestyksen ottamisen mukaan kriteereiksi, kun lasketaan lopullista parhaimmuusjärjestystä, jonka tuloksena saatiin pistekategoria, eli kokonaispisteet. Viimeisenä kaikki ISV-kumppanit lajiteltiin suurimmasta pienimpään heidän kokonaispisteiden perusteella. Mitä isompi kokonaispistemäärä sitä tärkeämpi kyseinen ISV-kumppani on.

Liitteissä 2. ja 3. on havainnollistettu kolmen tärkeimmän ISV-kumppanin data. Liitteistä on kuitenkin anonymisoitu tiedot, joiden avulla kyseiset ISV-kumppanit voitaisiin mahdollisesti tunnistaa. Liitteiden tulostaulukoissa ”Partner_rank”-sarake viittaa ISV-kumppanin partnerikategoriassa saatuihin pisteisiin, ”Portfolio_rank”-sarake viittaa portfoliokategoriassa saatuihin pisteisiin, ”Cluster_rank”-sarake viittaa klusterikategoriassa saatuihin pisteisiin ja ”Score_rank” sisältää kokonaispisteet.

Tämä lopputulos antoi jokaiselle partnerille neljä eri sijalukua, joita voidaan hyödyntää partnerin prioriteetin arvioinnissa. Useat sijaluvut mahdollistavat esimerkiksi sen, että voidaan etsiä parhaimmat partnerit tietystä portfoliosta tai partneritasosta, tai verrata miten tietyn partneritason ISV-kumppanien sijaluvut suhteutuvat muihin ISV-kumppaneihin Klusterien sijaluvut antavat myös mahdollisuuden tutkia, miten klusteroinnin avulla jaetut ISV-kumppanit sijoittuvat eri sijalukujen perusteilla. Laskenta toteutettiin täysin Python-ohjelmointikielen avulla ja se mahdollistaa ketterän painoarvojen muokkaamisen ja uusien sijalukujen laskemisen.

8 TULOKSET JA NIIDEN ARVIOINTI

Tässä kappaleessa tarkastellaan tämän diplomityön tuloksia ja perehdytään tarkemmin lopputulosten arviointiin ja erityisesti työn haasteisiin, jatkotoimenpiteisiin ja suosituksiin. Seuraavassa alakappaleessa esitetään vastaukset tämän työn tutkimuskysymyksiin.

8.1 Työn keskeiset tulokset

Työn aikana kohdattiin merkittävä määrä haasteita kappaleen 8 alussa esitellyn data-analyysin prosessin alkuvaiheissa, erityisesti siinä vaiheessa, kun dataa kerättiin ja esikäsiteltiin. Datalähteitä oli useita ja datalähteiden yhdistäminen toisiinsa ei ollut yksinkertaista. Data-analyysin prosessi eteni onnistuneesti ja lopputulokset näyttävät lupaavilta ottaen huomioon datan saatavuuden ja laadun. Työn jatkotoimenpiteet ja suositukset keskittyvät siihen, miten nyt kehitettyä data-analyysi prosessia pystyisi parantamaan ja tehostamaan ja myös siihen, kuinka sitä voisi jatkokehittää. Data-analyysi prosessissa päästiin kappaleessa 4.2. esitetyn DIKW-pyramidin kolmannelle tasolle eli tietämyksen tasolle, mutta potentiaalia olisi päästä viimeiselle viisauden tasolle.

Työn ensimmäinen tutkimuskysymys oli ”Millainen on hyvä ISV-kumppani?”. Tutkimuskysymykseen etsittiin vastausta kirjallisuuskatsauksen avulla. Hyvän ISV-kumppanin määrittäminen implisiittisesti on hyvin vaikeaa ja siihen löytyi kirjallisuudesta useita eri näkökulmia. Kaikista eniten kirjallisuudessa oli esillä ISV-kumppanin valintaprosessi, jossa hyvä ISV-kumppani määritellään, ennen kuin se on osana ohjelmistoekosysteemiä. ISV-kumppanin valintaprosessista löytyi hyvin kirjallisuutta ja yhteenveto suosituimmista ISV-kumppanin valintakriteereistä esitettiin luvussa 2.2. taulukossa 2. viisi eri ISV-kumppanin valintakriteerien kategoriaa, ja niistä kolme suosituinta kriteeriä. Valintakriteerien kategoriat ovat partnerin piirteet, partnerin kyvykkyydet, partnerin omistukset, partnerin myyntikyvyt ja ohjelmistoyrityksen näkökulma. Työn luvussa 2.3. käsiteltiin, kuinka kirjallisuudessa käsitellään ISV-kumppanien priorisointia. Kirjallisuuden pohjalta löytyi hyvin vaihtelevia tapoja priorisoida partnereita, mutta aihealueet, joita ISV-kumppaneissa verrattiin priorisointia varten, olivat hyvin samanlaisia, kuin kappaleessa 2.2. taulukossa 2. esitetyt valinta kriteerit.

Näiden havaintojen pohjalta voitiin päätellä, että ISV-kumppanien valintakriteerejä voi hyödyntää myös nykyisten ISV-kumppanien priorisoinnissa ja ISV-kumppanien valintakriteerien avulla voidaan määritellä hyvä ISV-kumppani. Kirjallisuudesta löydettyjä valintakriteerejä hyödynnetään laajemmin kolmannen tutkimuskysymyksen tuloksissa.

Työn toinen tutkimuskysymys oli ”Miten nykyiset ISV-kumppanitasot voidaan jakaa uudelleen koneoppimisen avulla?” case-yrityksen Netvisor tuotteen ISV-kumppanit ovat tällä hetkellä jaettu kahteen pääkategoriaan, partnereihin ja premium partnereihin. Näiden lisäksi on kategoriat ”AO” ja ”Software Partner”. ”AO”-partnerit sisältävät vain tilitoimisto kumppaneita, ketkä ovat samalla ISV-kumppaneita ja ”Software Partner”-partnerit sisältävät vain ohjelmistoyrityksiä, jotka tekevät yksittäisiä integraatioita muille case-yrityksen asiakkaille. Lähtötasona kaikille ISV-kumppaneille on ”partner” partneritaso ja jos heistä tulee merkittävä osa case-yrityksen ohjelmistoekosysteemiä muun muassa liikevaihdon ja loppuasiakkaiden määrän mukaan, heistä voi tulla premium partnereita.

Toista tutkimuskysymystä varten täytyi ensin selvittää mitä dataa ISV-kumppaneista on saatavilla, seuraavaksi kerätty data piti käsitellä ja siivota jalostamista varten. Seuraavaksi oli selvitettävä kirjallisuudesta, miten koneoppimista pystyisi hyödyntämään saatavilla olevaan dataan. Koneoppiminen voidaan jakaa ohjaamattomaan ja ohjattuun oppimiseen. Tässä tutkimuskysymyksen ongelman asettelussa ainut vaihtoehto oli hyödyntää ohjaamatonta oppimista ja siihen kuuluvaa klusterointia. Klusterointialgoritmien avulla voidaan opettaa koneoppimismalli lajittelemaan saatavilla oleva data joukkoihin datapisteiden samankaltaisuuden perusteella. Tutkimuskysymyksessä oli kyse jakaa ISV-kumppanien data joukkoihin, tai toisin sanoen kategorioihin, jotka olisivat verrattavissa nykyisiin ISV-kumppanitasoihin. Kirjallisuuskatsauksen jälkeen klusterointialgoritmiksi valikoitui K-Means-algoritmi, jota käytettiin datan klusteroinnissa. Klusteroinnilla saatiin kattavia ja hyvin eriteltyjä partnerijoukkoja. Klusteroinnin tuottamat joukot olivat tarpeeksi erillään toisistaan, eikä niissä ollut päällekkäisyyksiä. Joukot olivat järkevän kokoisia ja klusterointi oli onnistunut huomioimaan myös ISV-kumppanien portfoliot ja niiden vaikutus oli selkeästi huomattavissa.

Työn kolmas tutkimuskysymys oli ”Miten määritellä ISV-kumppaneille tärkeysjärjestys”. Tämä tutkimuskysymys linkittyy suoraan ensimmäiseen tutkimuskysymykseen, jossa haettiin

määritelmää hyvälle ISV-kumppanille. Tämän tutkimuskysymyksen ratkaisussa hyödynnettiin kirjallisuudesta löydettyjä priorisoinnin kriteerejä ja menetelmiä. Kirjallisuudessa oli hyödynnetty ISV-kumppanien priorisoinnissa kriteerien pisteytystä, joten tämän tutkimuskysymyksen ratkaisuun pääsemiseen sovellettiin usean kriteerin päätösanalyysiä.

Usean kriteerin päätösanalyysi on päätöksenteon työkalu, jossa hyödynnetään erilaisia menetelmiä, joita hyödyntämällä voidaan valita paras vaihtoehto ennalta annettujen kriteerien pohjalta. Kirjallisuuskatsauksen pohjalta tätä työtä varten usean kriteerin päätösanalyysin menetelmäksi valikoitui SAW-menetelmä, jossa jokaiselle kriteerille annetaan painoarvo. Painoarvojen avulla voidaan jokaiselle päätöksenteon vaihtoehdolle laskea painotettu summa. SAW-menetelmään päädyttiin sen yleisen suosion ja yksinkertaisuuden takia. Usean kriteerin päätösanalyysiä varten on tärkeää määritellä kriteerit ja niiden painokertoimet, joiden avulla ISV-kumppanit pystytään pisteyttämään.

Kriteerien määrittelyn rajoittavana tekijänä oli saatavilla oleva data, joka on esitelty tarkemmin luvun 7.2. taulukossa 5. Saatavilla olevaa dataa verrattiin kirjallisuudessa yleisesti käytettyihin kriteereihin ja datasta valittiin viisi parametria usean kriteerin päätösanalyysin kriteereiksi. Tämän jälkeen kriteereille määriteltiin painoarvot ja itse SAW-menetelmä toteutettiin Python-ohjelmointikielen avulla. Tuloksia voi tarkastella liitteistä 2. ja 3. tulokset on jaettu kahteen taulukkoon, jossa on listattu kolmen tärkeimmän ISV-kumppanin data normalisoituna ja SAW-menetelmän lopputulokset. Usean kriteerin päätösanalyysissä laskettiin jokaiselle ISV-kumppanille SAW-menetelmän avulla pisteytykset neljässä eri kategoriassa ja pisteytykset muutettiin sijaluvuiksi suurimmasta pistemäärästä pienimpään. Sijalukujen neljä kategoriaa olivat partneri-, portfolio, klusteri ja kokonaissijaluku. Jos halutaan tarkastella parhaimpia ISV-kumppaneita tietyllä partneritasolla, voidaan tarkastella partnerikategorian sijalukuja, tai jos halutaan tarkastella tietyn portfolion parhaimpia ISV-kumppaneita, voidaan tarkastella portfoliokategorian sijalukuja. Kokonaissijaluku sisältää kaikki ISV-kumppanit yhdessä paikassa.

Usean kriteerin päätösanalyysillä saatiin hyviä tuloksia, ja oli erityisen mielenkiintoista hyödyntää työn aiemmassa vaiheessa tehtyjä klustereita yhtenä sijalukujen kategoriana. Python ohjelmointikielellä tehty toteutus usean kriteerin päätösanalyysistä SAW-menetelmällä

mahdollistaa helpon jatkokehityksen ja tuloksien iteroimisen jatkossa. Painoarvoja on helppo tarvittaessa muuttaa, ja mikäli uutta dataa tulee saataville, se on myös helppo lisätä nykyiseen toteutukseen. Nykyisellään toteutuksesta käyttäjä saa itselleen CSV-formaatissa olevan tiedoston, jossa kaikki ISV-kumppanit on listattu liitteen 2. ja 3. tapaan.

8.2 Tulosten arviointi

Tässä kappaleessa käydään läpi tulosten arviointia, työn haasteita ja onnistumisia. Työn ongelmatilanne onnistuttiin määrittelemään hyvin jo työn alkuvaiheessa, joka mahdollisti työn jouhevan etenemisen. Työn lopputuloksiin päästiin ja tutkimuskysymyksiin saatiin kattavat vastaukset. Työn aikana vastaan tuli huomattava määrä haasteita, mutta ne onnistuttiin aina ratkaisemaan. Haasteiden pohjalta pystyi rakentamaan hyvä käsityksen ideoista jatkokehitystä varten. Isoimmat haasteet liittyivät dataan. Dataa ISV-kumppaneista oli hyvin rajallisesti saatavilla, eikä sitä kerätty kovin kattavasti. Datan rajallisuus rajoitti merkittävästi mahdollisia priorisoitavia asioita ISV-kumppaneista. Datassa oli paljon puutteita, kuten esitelty luvussa 8.2. Data sisälsi myös kirjoitusvirheitä ja oli paikoittain vaikea selvittää, oliko data puutteellista vai paikkansa pitävää. Datasta löytyy nolla-arvoja, joista ei pysty tulkitsemaan onko oikeasti kyseisen parametrin arvo nolla, vai puuttuuko kyseinen parametri kokonaan. Datan tarkastelu ja yhdistäminen oli erittäin haasteellista, koska kaikissa datalähteet, jotka liittyivät ISV-kumppaneihin eivät sisältäneet parametria, jonka avulla datalähteet voitaisiin yhdistää toisiinsa. Työtä varten piti kehitellä menetelmä, jossa datasta parsitaan ISV-kumppanin nimi ja yrityksen nimi ja tätä tietoa käytettiin yhdistämään eri tietolähteitä.

Kaikkiin työn tutkimuskysymyksiin saatiin tyydyttävät vastaukset, mutta parannettavaa jäi jokaisella rintamalla. Kirjallisuutta tämän työn aiheesta oli hyvin rajallisesti saatavilla, joten kirjallisuutta piti yrittää soveltaa tämän työn kontekstiin. ISV-kumppanien klusterointi onnistui hyvin, vaikka saatavilla data oli rajattua ja sen laatu oli vaihtelevaa. Klusterointi olisi voinut olla vielä tarkempaa siinä mielessä, että klusterit voisi olla vielä selkeämmin toisistaan erillään. Optimaalisessa tilanteessa klusterit olisivat myös enemmän samankokoisia toistensa kanssa. ISV-kumppanien priorisoinnin toteutus onnistui hyvin ja valittu usean kriteerin päätösanalyysin SAW-menetelmä sopi erinomaisesti työn kontekstiin ja työtä varten kehitetyn Python-koodin

avulla ISV-kumppanien priorisointi on helppo ottaa vakituisesti käyttöön osaksi case-yrityksen PAM:ien päivittäistä työtä. Työn tulokset olivat rajoitteista huolimatta hyviä ja merkittäviä.

8.3 Jatkotoimenpiteet ja suositukset

Tässä kappaleessa käydään läpi työn kannalta oleellisimmat jatkotoimenpiteet ja suositukset niihin liittyen. Edellisessä luvussa käytiin läpi työn tuloksia ja syvennyttiin erityisesti työn aikana ilmenneisiin haasteisiin erityisesti datan suhteen. Työn tuloksien ja havaintojen pohjalta suosittelisin case-yrityksen kehittämään omaa datakulttuuria erityisesti ISV-kumppanien suhteen. Datakulttuurin rakennuspilarina voisi toimia tietoallas tai tietovarasto. Tietoallas ja tietovarasto ovat arkkitehtuurisia ratkaisuja, joihin voidaan tallentaa isoja datamassoja strukturoitua tai strukturoimatonta dataa. Dataa tietoaltaisiin ja varastoihin voidaan kerätä dataa useasta eri lähteestä ja yhdistää niitä tietoaltaan tai varaston avulla. Tämä mahdollistaisi ISV-kumppanidatan helpomman käytön ja jatkokehityksen jatkossa. (AWS, 2021)

Tärkeänä datakulttuurissa on nostaa datan tärkeys korkealla prioriteetille. Hyvänä lähtökohtana on keskittyä datan laadun valvontaan ja nykyisen datan eheyden varmistamiseen ja virheiden korjaamiseen. Hyvänä jatkotoimenpiteenä olisi tutkia, voisiko ISV-kumppaneista olla lisää arvokasta dataa saatavilla. Dataa voisi ehdottomasti kerätä lisää ja ehdotuksena olisi pyrkiä keräämään ISV-kumppanien loppuasiakkailta dataa asiakastyytyväisyydestä. Loppuasiakkaiden tyytyväisyydellä voitaisiin arvioida paremmin ISV-kumppaneilta monesta eri näkökulmasta, esimerkiksi tuotteen ja asiakaspalvelun laatua. Datan määrän kasvaessa nykyiseen toteutukseen on mahdollista lisätä erilaisia kasvulukuja ISV-kumppaneista ja sen avulla voitaisiin mahdollisesti ennustaa ISV-kumppanin kasvua. Kasvuluvut ja niiden pohjalta luodut ennusteet voisivat olla hyvä lisä nykyiseen ratkaisuun. Haasteena ja kehityskohteena on myös se, että kaikki tieto Netvisorin ISV-kumppaneista on saatavilla vain tuotetasolla. Tarkoittaen sitä, että kaikki attribuutit viittaavat tietyn partnerin tiettyyn tuotteeseen, eli siinä tilanteessa, kun partnerilla on useampi tuote tarjolla, se on listauksessa mukana jokaisen tuotteensa kanssa. Tämä luo sellaisen haasteen, että ISV-kumppanin tuottoja ei voi suoraan aggregoida jokaisen tuotteen pohjalta.

Jatkotoimenpiteenä voisi olla nykyisen ratkaisun soveltaminen myös ISV-kumppanien valintaprosessiin. ISV-kumppaniksi hakevilta yrityksiltä voitaisiin pyytää valmiiksi tiettyjä tietoja ja koneoppimista hyödyntämällä voitaisiin pyrkiä ennustamaan potentiaalisen ISV-kumppanin menestystä ISV-kumppaniohjelmassa.

9 YHTEENVETO

Tämän työn tarkoituksena oli tutkia ohjelmistoekosysteemin ISV-kumppaneiden klusterointia ja priorisointia. Työ tehtiin yhteistyössä case-yrityksen, Visma Solutions Oy:n kanssa. ISV-kumppanit ovat case-yrityksessä merkittävässä roolissa osana heidän ohjelmistoekosysteemiänsä. ISV-kumppanien hallintaa tehostamalla pystytään paremmin hallinnoimaan kasvavaa ohjelmistoekosysteemiä. ISV-kumppaneilla on merkittävä rooli osana ohjelmistoekosysteemiä, koska he ovat osana luomassa lisäarvoa koko ekosysteemille kehittämällä lisäpalveluita loppuasiakkaille. ISV-kumppaneita priorisoimalla voitaisiin paremmin tehostaa yhteistyötä tärkeimpien ISV-kumppanien kanssa esimerkiksi markkinoinnin tai yhteisen tuotekehityksen muodossa.

Työn ensimmäisen tutkimuskysymyksen tavoitteena oli kirjallisuuskatsauksen avulla määritellä millainen olisi hyvä ISV-kumppani. Kirjallisuuskatsauksen suorittaminen oli haastavaa, koska lähteitä oli saatavilla rajallisesti. Hyvän ISV-kumppanin määrittely keskittyi kirjallisuudessa ISV-kumppanin valintaprosessiin ja valintakriteereihin. Ratkaisuna tutkimuskysymykseen on määritellä hyvä ISV-kumppani valintakriteerien avulla.

Toisen tutkimuskysymyksen tavoitteena oli hyödyntää koneoppimista ISV-kumppanien jakamisessa joukkoihin. Tällä hetkellä kaikki ISV-kumppanit kuuluvat tietylle partneritasolle. Tutkimuskysymystä lähdettiin ratkaisemaan kirjallisuuskatsauksen avulla. Kirjallisuudessa perehdyttiin erilaisiin koneoppimisen menetelmiin ja algoritmeihin. Menetelmäksi valikoitui klusterointi ja tarkemmin K-means klusterointialgoritmi, jonka avulla ISV-kumppanit saadaan klusteroitua joukkoihin. Klusterointi osoittautui tehokkaaksi tavaksi jakaa ISV-kumppaneita uusiin ryhmiin ja klusteroinnin tulokset näyttivät lupaavilta.

Työn kolmannen tutkimuskysymyksen tavoitteena oli määritellä ISV-kumppaneille tärkeysjärjestys priorisointia varten. Tärkeysjärjestyksen rakentamisen ongelmaan löytyi apuja aiemmin suoritetun kirjallisuuskatsauksen avulla. Kirjallisuuskatsauksen avulla ISV-kumppaneita oli verrattu keskenään ennalta valittujen kriteerien pohjalta. Usean kriteerin päätösanalyysiä ja ISV-kumppaneista saatavilla olevaa dataa pystyttiin luomaan ISV-kumppaneille tärkeysjärjestys, jota voidaan hyödyntää ISV-kumppaneita priorisoitaessa.

LÄHTEET

Ackoff, R. 1999. From Data to Wisdom. Ackoff's Bests. New York: John Wiley & Sons. s. 170 – 172.

Afshari, A., Mojahed, M. & Yusuff, R.M., 2010. Simple Additive Weighting approach to Personnel Selection problem. International Journal of Innovation, Vol. 1. No. 5, 511-515

Adriyendi, S. 2015. Multi-Attribute Decision Making Using Simple Additive Weighting and Weighted Product in Food Choice. International Journal of Information Engineering and Electronic Business Vol. 7. No. 6, 8–14. https://doi.org/10.5815/ijieeb.2015.06.02

Ahmed, M. 2020. K-means Clustering Clearly explained - Artificial Intelligence in Plain English. [WWW-dokumentti] Medium. Saatavilla: https://ai.plainenglish.io/k-means-clustering-clearly-explained-4b61d013ed46 [Viitattu 27.8.2021].

Amazon. 2021. What is a data lake? [WWW-dokumentti] Saatavilla: https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/?nc=sn&loc=2 [Viitattu 20.9.2021].

Anushka, M. 2017. Four Types of Business Analytics to Know | Analytics Insight. [WWW-dokumentti] Analytics Insight. Saatavilla: https://www.analyticsinsight.net/four-types-of-business-analytics-to-know/ [Viitattu 24.3.2021].

AltexSoft. 2020. Comparing API Architectural Styles: SOAP vs REST vs GraphQL vs RPC. [WWW-dokumentti] Saatavilla: https://www.altexsoft.com/blog/soap-vs-rest-vs-graphql-vs-rpc/ [Viitattu 20.3.2021].

Baak, M., Koopman, R., Snoek H. & Klous S. 2019. A new correlation coefficient between categorical, ordinal and interval variables with Pearson characteristics. Saatavilla: http://arxiv.org/abs/1811.11440/ [Viitattu 25.8.2021].

Balducci, B & Marinova, D. 2018. Unstructured data in marketing. Journal of the Academy of Marketing Science. Vol. 46. S. 557–590. doi: https://doi.org/10.1007/s11747-018-0581-x.

Bech, H. P. 2015. Building Successful Partner Channels: Channel Development & Management in the Software Industry. TBK Consult.

Beelen, L. Are You of Value to Me? A Partner Selection Framework for Software Ecosystem Orchestrators. Department of Information and Computing Sciences. Utrecht University.

Bekker, A. 2017. 4 Types of Data Analytics to Improve Decision-Making. [WWW-dokumentti] Saatavilla: https://www.scnsoft.com/blog/4-types-of-data-analytics [Viitattu 24.3.2021].

Blanda, S. 2019. When is a Tech Partnership a Waste of Time? [WWW-dokumentti] Crossbeam.com. Available at: https://blog.crossbeam.com/tech-partnership-integration-saas [Viitattu 25.7.2021].

Bonaccorso, G. 2018 Machine learning algorithms: popular algorithms for data science and machine learning. Second edition. Birmingham Packt Publishing.

Bosch, J. 2009. From Software Product Lines to Software Ecosystems. 13th International Software Product Line Conference. 24-28. doi: 10.1145/1753235.1753251

Brownlee, J. 2017. Why One-Hot Encode Data in Machine Learning? [WWW-dokumentti] Machine Learning Mastery. Saatavilla: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/ [Viitattu 29.6.2021].

Chen, S., Lee, H. & Wu, Y., 2008. Applying ANP approach to partner selection for strategic alliance. Management Decision Vol. 46. No.3., 449–465. doi: https://doi.org/10.1108/00251740810863889

Cohen, A., 2011. FuzzyWuzzy: Fuzzy String Matching in Python. [WWW-dokumentti] Saatavilla: https://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/ [Viitattu 23.8.2021].

Crispim, J. A., & Pinho, J. (2009). Partner selection in virtual enterprises: a multi-criteria decision support approach: Virtual Enterprises: Coalition Formation in Collaborative Networks. International Journal of Production Research. Vol. 47 No.17. s. 4791–4812

De, B., 2017. API Management. Apress, Berkeley, CA. doi: 10.1007/978-1-4842-1305-6

Djangone, A. Analyzing Your Data: Why Do Data Types Matter. [WWW-dokumentti] Saatavilla: www.medium.com/swlh/analyzing-your-data-why-do-data-types-matter-part-1-1aad04e543e7. [Viitattu 23.8.2021].

Donges, N. Data Types in Statistics - Towards Data Science. [WWW-dokumentti] Saatavilla: www.towardsdatascience.com/data-types-in-statistics-347e152e8bee. [Viitattu 23.8.2021].

Edwards, G. 2018. Machine Learning | An Introduction - Towards Data Science. [WWW-dokumentti] Saatavilla: https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0 [Viitattu 17.8.2021].

Eryk L. 2021. Phik (ϕk) — get familiar with the latest correlation coefficient. [WWW-dokumentti] Medium. Saatavilla: https://towardsdatascience.com/phik-k-get-familiar-with-the-latest-correlation-coefficient-9ba0032b37e7 [Viitattu 20.8.2021].

Figueroa, A. 2019. Data Demystified – DIKW model. [WWW-dokumentti] [Viitattu 2.4.2021] Saatavilla: https://towardsdatascience.com/rootstrap-dikw-model-32cef9ae6dfb

Frankenfield, J. 2020 Data-analytics. [WWW-dokumentti] [Viitattu 20.3.2021] Saatavilla: https://www.investopedia.com/terms/d/data-analytics.asp

Gowthamy V. 2018. Machine Learning: Supervised Learning vs Unsupervised Learning. [WWW-dokumentti] Saatavilla: https://gowthamy.medium.com/machine-learning-supervised-learning-vs-unsupervised-learning-f1658e12a780 [Viitattu 17.8.2021].

Henderson, C. 2020. Types of API Explained- How Are They Different? [WWW-dokumentti] Saatavilla: https://anyconnector.com/api-integration/types-of-api.html [Viitattu 20.3.2021].

Higgins, D. 2020. Four essential ingredients of successful ecosystem partnerships. [WWW-dokumentti] Ey.com. Saatavilla: https://www.ey.com/en_gl/consulting/four-essential-ingredients-of-successful-ecosystem-partnerships [Viitattu 19.9.2021].

Hoch, D., Roeding, C., Purkert, G., Lindner, S., & Müller, R. 1999. Secrets of Software Success: Management Insights from 100 Software Firms Around the World. Boston: Harvard Business School Press.

Holst, A. 2021. Total data volume worldwide 2010-2025 [WWW-dokumentti] Saatavilla: https://www.statista.com/statistics/871513/worldwide-data-created/ [Viitattu 12.9.2021]

Huang, B., Lihui, B., Arnab & R., Ningning, M. 2018. A multi-criterion partner selection problem for virtual manufacturing enterprises under uncertainty. Internal Journal of Production Economics. Vol. 196. s. 68-81 doi: 10.1016/j.ijpe.2017.08.024.

Hudson, R. 2015. Decision-making: processes, behavioral influences and role in business management. New York, New York: Nova Science Pub Inc.

Jansen, S & Cusumano, M.A., 2013. Defining software ecosystems: a survey of software platforms and business network governance, in: Software Ecosystems. Edward Elgar Publishing. 13- 28.

Joliffe, I. & Cadima, J. 2016. Principal component analysis: a review and recent developments. Philosophical Transactions Of The Royal Society A. Vol. 374 No. 2065. doi: https://doi.org/10.1098/rsta.2015.0202

Joshi, A. V. 2020. Machine Learning and Artificial Intelligence. Cham: Springer International Publishing. doi: https://doi.org/10.1007/978-3-030-26622-6

Kahler, S. 2021. Video: Reinforcement learning using Deep-Q Networks with SAS Viya. [WWW-dokumentti] The SAS Data Science Blog. Saatavilla: https://blogs.sas.com/content/subconsciousmusings/2021/03/18/reinforcement-learning-deep-q-networks/ [Viitattu 17.8.2021].

Kankanamge, C. 2012. Web Services Testing with SoapUI. Birmingham, Packt Publishing. 332 s. ISBN 1-62198-901-1

Kaistinen, J. 2019. Partner ecosystems in enterprise software: cause and effect of the business model from vendor, partner, and customer perspectives. Tieto- ja palvelutalouden laitos. s.83.

Karl P. & Meyer R. 2010. Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry. Books on Demand. 242 s. ISBN-10 3842300514.

Kim, J. 2019. Why we integrate: Why smart companies build ecosystems. [WWW-dokumentti] Saatavilla: https://www.appcues.com/blog/why-companies-build-product-ecosystems [Viitattu 20.3.2021].

Klutho, S. 2013. Mathematical Decision Making. An Overview of the Analytics Hierarchy Process. [WWW-dokumentti] Saatavilla: https://www.whitman.edu/Documents/Academics/Mathematics/Klutho.pdf [Viitattu 23.8.2021]

Laerd.com. 2018. Spearman’s Rank-Order Correlation - A guide to when to use it, what it does and what the assumptions are. [WWW-dokumentti] Saatavilla:

https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php [Viitattu 29.7.2021].

Laerd.com. 2012. Pearson Product-Moment Correlation - When you should run this test, the range of values the coefficient can take and how to measure strength of association. [WWW-dokumentti] Saatavilla: https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php [Viitattu 29.7.2021].

Li, H. 2020. Which machine learning algorithm should I use? [WWW-dokumentti] The SAS Data Science Blog. Saatavilla: https://blogs.sas.com/content/subconsciousmusings/2020/12/09/machine-learning-algorithm-use/ [Viitattu 17.8.2021].

Louridas, P. & Ebert, C. 2016. Machine Learning. IEEE Software, Vol. 33 No. 5, s.110–115. doi: 10.1109/MS.2016.114.

Ma, D., 2007. The Business Model of “Software-As-A-Service.” IEEE International Conference on Services Computing. doi: 10.1109/SCC.2007.118.

Macharis, C., Springael, J., De Brucker, K., & Verbeke, A. 2004. PROMETHEE and AHP: The design of operational synergies in multicriteria analysis.: Strengthening PROMETHEE with ideas of AHP. European Journal of Operational Research. Vol. 153 No. 2, s. 307–317. doi: S0377-2217(03)00153-X

Maklin, C. 2018. K-means Clustering Python Example - Towards Data Science. [WWW-lähde] Saatavilla: https://towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203 [Viitattu 23.8.2021].

Mohamad, I. & D. Usman. 2013. Standardization and Its Effects on K-Means Clustering Algorithm. Research Journal of Applied Sciences, Engineering and Technology. S. 3299–3303. doi:10.19026/rjaset.6.3638

Moilanen, J., Niinioja, M., Seppänen, M. & Honkanen M. 2018 API-talous 101. Helsinki: Alma.

Netvisor. (2020). Visma Netvisor | Älykäs Taloushallinto ja Palkanlaskenta - kokeile maksutta. [WWW-lähde] Saatavilla: https://netvisor.fi/ [Viitattu 20.3.2021].

Netvisor a. Ohjelmistorajapinta. [WWW-lähde] Saatavilla: https://support.netvisor.fi/fi/support/solutions/articles/77000466610-ohjelmistorajapinta [Viitattu 15.5.2021]

Netvisor b. Netvisor Marketplace [WWW-lähde] Saatavilla: https://marketplace.netvisor.fi/kaikki-integraatiot/ [Viitattu 20.3.2021].

Nousiainen, R. 2018. Suomalaisten ohjelmistoyritysten kumppanuussuhteet. Tietojärjestelmätiede. Saatavilla: http://urn.fi/URN:NBN:fi:jyu-201803141724

Papathanasiou, J. & Ploskas, N. 2018 Multiple Criteria Decision Aid: Methods, Examples and Python Implementations. Cham: Springer International Publishing. doi: 10.1007/978–3–319–91648–4.

Partanen, J., & Möller, K. 2012. How to build a strategic network: A practitioner-oriented process model for the ICT sector. Industrial Marketing Management. Vol. 41 No. 3. s. 481–494.

Pickell, D. 2018. Structured vs Unstructured Data – What's the Difference? [WWW-lähde] Saatavilla: https://learn.g2.com/structured-vs-unstructured-data [Viitattu 5.6.2021]

Popp, K. M. 2010. Goals of Software Vendors for Partner Ecosystems – A Practitioner´s View. Software Business. Berlin, Heidelberg: Springer Berlin Heidelberg. S. 181–186.

PyCaret. 2021. Home - PyCaret. [WWW-lähde] Saatavilla: https://pycaret.org/ [Viitattu 27.8.2021].

Pydata. 2015. pandas - Python Data Analysis Library. [WWW-lähde] Saatavilla: https://pandas.pydata.org/about/index.html [Viitattu 11.7.2021].

Python. 2021. Welcome to Python.org. [WWW-lähde] Saatavilla: https://www.python.org/ [Viitattu 11.6.2021].

Ramirez, O. 2020. Partnerships 101: ISVs, VARs, SIs, MSPs, and the Glue that Holds them Together. [WWW-lähde] Crossbeam.com. Saatavilla: https://blog.crossbeam.com/partnerships-101-isv-vs-var-vs-si-vs-msp-vs-oem [Viitattu 12.7.2021].

Ramirez, O. 2021. Vetting 100 s of Channel Partners? Use This 4-Criteria Checklist to Speed Up The Process. [WWW-lähde] Crossbeam.com. Saatavilla: https://blog.crossbeam.com/vetting-channel-partners-speed-up-process [Viitattu 21.7.2021].

Rennemeyer, A. 2020. Types of Data in Statistics - Nominal, Ordinal, Interval, and Ratio Data Types Explained with Examples. [WWW-lähde] Saatavilla: www.freecodecamp.org/news/types-of-data-in-statistics-nominal-ordinal-interval-and-ratio-data-types-explained-with-examples [Viitattu 21.7.2021].

Ricadela, A. 2006. Salesforce Taps ISV Partners To Feed The Growth Beast. [WWW-lähde] Saatavilla: https://www.informationweek.com/it-life/salesforce-taps-isvs-to-feed-the-growth-beast [Viitattu 21.7.2021].

Rickmann, T., Wenzel, S. & Fischbach, K., 2014. Software Ecosystem Orchestration: The Perspective of Complementors. 20th Americas Conference on Information Systems, AMCIS 2014. [WWW-lähde] Saatavilla: https://www.researchgate.net/publication/286307057_Software_ecosystem_orchestration_The_perspective_of_complementors [Viitattu 15.7.2021].

Riley, C. 2019. Know your API protocols: SOAP vs. REST vs. JSON-RPC vs. gRPC vs. GraphQL vs. Thrift. [WWW-lähde] Saatavilla: https://www.mertech.com/blog/know-your-api-protocols [Viitattu 20.3.2021].

Rowley, J., 2007. The wisdom hierarchy: representations of the DIKW hierarchy. Journal of Information Science. Vol. 33. No. 2. S. 163–180. doi: 10.1177/0165551506070706.

Runkler, T.A., 2020. Data Analytics: Models and Algorithms for Intelligent Data Analysis. Springer Fachmedien Wiesbaden, Wiesbaden. doi: https://doi.org/10.1007/978-3-658-29779-4 S. 161

Scikit-learn. 2020. Selecting the number of clusters with silhouette analysis on KMeans clustering — scikit-learn 0.24.2 documentation. [WWW-lähde] Saatavilla: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html [Viitattu 30.8.2021].

Shaurya U. 2020. Ranking and Selection of the best with Multi-Criteria Decision Making (MCDM) — Part Two. [WWW-lähde] Saatavilla: https://medium.datadriveninvestor.com/ranking-and-selection-of-the-best-with-multi-criteria-decision-making-mcdm-part-two-78b3adc6cca5 [Viitattu 7.8.2021].

Singh, M. 2020. Types & Scales of Data in Descriptive Statistics - Analytics Vidhya - Medium. [WWW-lähde] Saatavilla: www.medium.com/analytics-vidhya/types-scales-of-data-in-descriptive-statistics-d3aa439c0b1e. [Viitattu 7.8.2021].

Singh, A., Nowak, R., & Zhu, X. 2008. Unlabeled data: Now it helps, now it doesn't. Department of Computer Sciences. University of Wisconsin – Madison. Advances in Neural Information Processing Systems 21 (NIPS 2008)

Stevens, S. 2021. What are the different types of data analysis? [WWW-lähde] Saatavilla: https://careerfoundry.com/en/blog/data-analytics/different-types-of-data-analysis/ [Viitattu 24.3.2021].

Stobierski, T. 2021.The advantages of data driven decision-making. [WWW-lähde] Saatavilla: https://online.hbs.edu/blog/post/data-driven-decision-making [Viitattu 12.9.2021]

Tableau. 2019. What is Tableau? [WWW-lähde] Saatavilla: https://www.tableau.com/why-tableau/what-is-tableau [Viitattu 11.7.2021].

Taylor, C. 2018. Structured vs. Unstructured Data | Datamation. [WWW-lähde] Datamation. Saatavilla: https://www.datamation.com/big-data/structured-vs-unstructured-data/ [Viitattu 5.7.2021].

Thakkar, J. J. 2021. Multi-criteria decision making. Springer: Singapore. doi: 10.1007/978-981-33-4745-8.

Triantaphyllou, E., Shu, B., Sanchez, S.N., & Ray, T., 1998. Multi-Criteria Decision Making: An Operations Research Approach. Encyclopedia of Electrical and Electronics Engineering. Vol. 15. S. 175-186.

Velasquez, M. & Hester, P. T. 2013. An Analysis of Multi-Criteria Decision Making Methods. International Journal of Operations Research. Vol. 10 No. 2. s. 56 – 66.

Venkatesh, V., Zhang, A., Deakins, E., Luthra, S. & Mangla, S. 2019. A Fuzzy AHP-TOPSIS Approach to Supply Partner Selection in Continuous Aid Humanitarian Supply Chains. Annals of Operations Research.Vol. 283. No. 1-2. s. 1517–1550, doi:10.1007/s10479-018-2981-1.

Visma a 2019. Organisation. [WWW-lähde] Saatavilla: https://www.visma.com/organisation/ [Viitattu 3.1.2021].

Visma b 2019. Tietoa Vismasta. [WWW-lähde] Saatavilla: https://www.visma.fi/tietoa-vismasta/ [Viitattu 9.1.2021].

Visma Solutions. 2021. Fiksut ohjelmistot suomalaisille yrityksille | 100 % pilvipalveluna. [WWW-lähde] Saatavilla: https://vismasolutions.com/tuotteet/ [Viitattu 10.1.2021].

Wakefield, K. 2021. A guide to the types of machine learning algorithms. [WWW-lähde] Saatavilla: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html [Viitattu 17.8.2021].

Weinbaum, A. 2017. Classifying Partner Investment Prioritization via Partner Scorecards. [WWW-lähde] Computer Market Research. Saatavilla: https://computermarketresearch.com/partner-investment-prioritization/ [Viitattu 24.6.2021].

Weiping, L. 2009. An analysis of new features for workflow system in the SaaS software. Proceedings of the 2nd International Conference on Interaction Sciences Information Technology, Culture and Human. doi: 10.1145/1655925.1655946

Wu, W., His-An, S. & Hui-Chun, C. 2009. The analytic network process for partner selection criteria in strategic alliances. Expert Systems with Applications. Vol. 36. No. 3. S. 4646-4653. doi: 10.1016/j.eswa.2008.06.049.

Widianta, M.M.D., Rizaldi, T., Setyohadi, D.P.S. & Riskiawan, H.Y., 2018. Comparison of Multi-Criteria Decision Support Methods (AHP, TOPSIS, SAW & PROMENTHEE) for Employee Placement. Journal of Physics: Conf. Series 953. doi: 10.1088/1742-6596/953/1/012116

Yellowbrick 2021. Intercluster Distance Maps — Yellowbrick v1.3.post1 documentation. [WWW-lähde] Saatavilla: https://www.scikit-yb.org/en/latest/api/cluster/icdm.html [Viitattu 5.9.2021].

Yildirim, S. 2020. K-Means Clustering — Explained - Towards Data Science. [WWW-lähde] Medium. Saatavilla: https://towardsdatascience.com/k-means-clustering-explained-4528df86a120 [Viitattu 25.8.2021].

Yiu, T. 2020. Business Models: SaaS (Software as a Service) - Alpha Beta Blog - Medium. [WWW-lähde] Medium. Saatavilla: https://medium.com/alpha-beta-blog/business-models-saas-software-as-a-service-65a6a9a9dd14 [Viitattu 13.5.2021].

Liite 1. Katsaus partnerin valintakriteereistä. (Beelen, 2019, s.19–24)

Kriteeri	**Kuvaus**	**Lähde**
Partnerin piirteet		
1 Luotettavuus	Ohjelmistoyritys etsii luotettavaa partneria. Luotettavuus on elintärkeä piirre ohjelmistoekosysteemille	(Robson, 2002), (Duysters, Kok,&Vaandrager,1999), (Shah &Swaminathan,2008), (Das &He,2006),(Keung&Griffiths,2008), (Chen etal.,2008),(Angeles&Nath,2000), (Solesvik &Westhead,2010),(Franco,2010), (Bierly III&Gallagher,2007), (Kraus, Meier,Niemand,&etal.,2018)
2 Maine ja uskottavuus	Maine ja uskottavuus partnerissa vaikuttaa siihen, kuinka ohjelmistoekosysteemin loppuasiakkaat ottavat uuden partnerin vastaan.	(Mat, Cheung,&Scheepers,2009), (Das &He,2006),(Wu,Shih,&Chan,2009), (Solesvik &Westhead,2010),(Dickson,1966), (Franco,2010),(BierlyIII&Gallagher,2007) (Schreiecketal.,2016)
3 Yhteistyöhistoria	Ohjelmistoyrityksen on hyvä tietää, onko partnerilla kokemusta yhteistyön tekemisestä	(Mat etal.,2009),(Wuetal.,2009), (Solesvik &Westhead,2010), (Das &He,2006),(Chenetal.,2008), (Jarimo, Salkari,&Bollhalter,2006)
5 Yhteistyö tavoitteet	Onko partnerilla samat tavoitteet yhteistyön suhteen, kuin ohjelmistoyrityksellä	(Angeles &Nath,2000)
6 Kulttuuri	Onko partnerin kulttuuri ja yrityskulttuuri linjassa ohjelmistoyrityksen kanssa.	(Geringer, 1988),(Chenetal.,2008), (Bierly III&Gallagher,2007), (Cartwright&Cooper,1993), (Robson, 2002),(Zutshi&Tan,2009),

		(Wuetal.,2009), (Franco,2010)
7 Tavoitteet	Onko partnerin tavoitteet linjassa ohjelmistoyrityksen kanssa.	(Partanen&Möller,2012),(Wueta l.,2009), (Bierly III&Gallagher,2007)
8 Organisaation struktuuri ja koko	Onko partneriyrityksen organisaation koko sellainen, mitä ohjelmistoyritys on etsimässä. Esimerkiksi, haluaako ohjelmistoyritys pieniä yrityksiä partneriksi.	(Geringer, 1988),(Das&He,2006), (Chen et al.,2008)
9 Taloudelliset mittarit	Ohjelmistoyrityksen on hyvä tietää potenttiaalisen partneriyrityksen taloudellinen tilanne	(Geringer, 1988),(Dickson,1966), (Shah &Swaminathan,2008), (Solesvik &Westhead,2010), (Das &He,2006)
10 Tuottavuus	Onko partneriyritys tuottava. Onko partnerilla mahdollista investoida heidän tuotteeseen.	(Chen et al.,2008)
11 Kasvupotenttiaali	Millainen on partnerin kasvupotenttiaali	(Chen et al.,2008)
12 Yritys-suunnitelma	Millainen on partneriyrityksen tulevaisuuden suunnitelmat	(Doherty,2009)
13 Yhteistyösuhteen kemiat	Onko partneriyrityksen kanssa käydyissä keskusteluissa yhteiset kemiat	(Doherty,2009)
14 Kommunikaation läpinäkyvyys	Onko partneriyrityksen kommunikaatio läpinäkyvää ja tehokasta	(Duysters et al.,1999), (Angeles &Nath,2000), (Alvesetal.,2017),(Benlianetal.,2 015)
15 Tavoitteiden läpinäkyvyys	Onko partneriyrityksen tavoitteet läpinäkyviä	(Benlian etal.,2015)
16 Tiedon jakamisen kulttuuri	Onko partneriyritys valmis jakamaan informaatiota ja osaamista	(Wuetal.,2009),(Solesvik&West head,2010) (Kreiner &Schultz,1993), (Emden, Calantone,&Droge,2006),(Baars &Jansen,2012)
17 Sitoutuneisuus	Onko partneriyritys valmis sitoutumaan ohjelmistoyrityksen sopimuksiin ja sääntöihin	(Benlian etal.,2015)
18 Sitoutuneisuus yhteistyöhön	Onko partneriyritys sitoutunut yhteistyöhön	(Partanen&Möller,2012), (Mat et al.,2009),(Chenetal.,2008), (Angeles &Nath,2000),(Franco,2010)

19 Lojaliteetti	Kuinka lojaali partneriyritys on ohjelmistoyritystä kohden	(Benlian etal.,2015)
20 Investointi halu	Onko partnerityritys valmis investoimaan rahaa yhteistyöhön	(Alves et al.,2017)
21 Omatoimisuus	Onko partneriyritys valmis ja kyvykäs toimimaan omatoimisesti	(Alves et al.,2017)
22 Joustavuus	Onko partneriyritys valmis joustamaan yhteistyötä varten	(Wuetal.,2009),(Wu&Barnes,2012)
23 Yleisten toimintatapojen noudattaminen	Noudattaako partneriyritys yleisiä ohjelmistokehityksen toimintatapoja	(Alves et al.,2017)
24 Asiakastyytyväisyys	Kuinka tyytyväisi partnerityrityksen asiakkaat ovat heidän tuotteeseensa	(Kannan &Haq,2007),(Dickson,1966), (Schreieck et al.,2016)
Partnerin kyvykkyydet		
25 Resurssien käytettävyys	Onko partnerilla resursseja käytettävissä tuotekehitykseen ja kasvuun	(Shah &Swaminathan,2008), (Kraus et al.,2018),(Das&He,2006), (Meffert &Swaminathan,2017), (Hitt, Dacin,Levitas,Arregele,&Borza, 2000), (Möller &Rajala,2007)
26 Ainutlaatuinen osaaminen	Erottuuko partneriyritys muista vastaavista toimijoista omalla osaamisella	(Hitt et al.,2000),(Das&He,2006), (Wuetal.,2009)
27 Kustannusten vähentäminen	Pystyykö ohjelmistoyritys laskemaan omia kuluja partneriyrityksen avulla	(Huang &Wu,2003),(Wu&Barnes,2012)
28 Innvointi mahdollisuudet	Onko partnerilla mahdollisuuksia ja osaamista innovoida ja kehittää uusia ominaisuuksia, joilla erottua eduksi.	(Alves et al.,2017)
29 Innovaatio keskeisyys	Onko partneriyritys koko ajan keskittynyt miettimään uusia innovaatioita	(Chen etal.,2008)
30 Hallinnointi kyvyt	Onko partneriyrityksen tapa hallinnoida omaa yritystä linjassa ohjelmistoyrityksen kanssa	(Zhong &Ren,2015),(Das&He,2006), (Wuetal.,2009)
Partnerin tuote		
31 Laatu	Miten laadukas partneriyrityksen tuote on	(Wu&Barnes,2011,2012),(Das&He,2006),

		(Solesvik &Westhead,2010)
32 Hinnoittelu	Onko partneriyrityksen hinnoittelu linjassa ohjelmistoyrityksen kanssa.	(Wu&Barnes,2011),(Kannan&Haq,2007), (Xia &Wu,2007),(Abratt,1986), (Dickson,1966)
33 Luotettavuus	Kuinka luotettava on partneriyrityksen tuote	(Xia &Wu,2007),(Abratt,1986)
34 Dokumentaatio	Onko partneriyrityksen dokumentaatio heidän tuotteestansa selkeä	(Benlian et al.,2015)
35 Tietoturvallisuus	Onko partneriyrityksen ja heidän tuotteensa tietoturva ajan tasalla ja standardien mukainen	(Schreieck et al.,2016)
36 Integraation tehokkuus	Kuinka tehokkaasti partneriyritys pystyy käyttämään ohjelmistoyrityksen rajapintoja. Mitä tehokkaampaa rajapinnan käyttö, sen parempi	(Benlian et al.,2015)
37 Jatkuva kehitys	Panostaako partneriyritys jatkuvasti heidän tuotteensa kehittämiseen	(Chen et al.,2008)
38 Asiakaspalvelu	Onko partnerityrityksen asiakaspalvelu korkealaatuista	(Benlian et al.,2015)
Partnerin omistukset		
39 In-house osaaminen	Onko partneriyrityksellä omaa osaamista organisaatiossa	(Dekker&VandenAbbeele,2007)
40 Tekninen osaaminen	Onko partneriyrityksellä omaa teknistä osaamista ja tuotekehitystä	(Wuetal.,2009)
41 Immateriaalioikeudet	Onko partnerilla täydet immateriaalioikeudet omaan tuotteeseen	(Huang &Wu,2003), (Al-Khalifa &EggertPeterson,1999)
42 Patentit	Onko partneriyrityksellä patentteja	(Huang &Wu,2003),(Wuetal.,2009), (Al-Khalifa &EggertPeterson,1999)
Partnerin myyntikyvyt		
43 Pääsy markkinoille	Miten laaja pääsy partneriyrityksellä on markkinoille ja millainen on markkinan kasvupotenttiaali	(Geringer, 1988),(Doherty,2009), (Wuetal.,2009)

44 Myyntikanavat	Onko partneriyrityksellä käytössä useita myyntikanavia	(Metallo, Agrifoglio,Schiavone,&Mueller, 2018), (Das &He,2006),(Wuetal.,2009)
45 Myyntiosaaminen	Kuinka paljon partneriyrityksellä on kokemusta myynnistä	(Metallo etal.,2018),(Das&He,2006)
46 Asiakaskunta	Kuinka iso asiakaskunta partneriyrityksellä on	(Das &He,2006)
47 Markkinatietämys	Kuinka hyvin partneriyritys tuntee omat markkinansa. Voiko ohjelmistoyritys hyötyä tästä tiedosta	(Mat etal.,2009),(Das&He,2006), (Franco,2010),(Wuetal.,2009), (Solesvik &Westhead,2010)
48 Markkinaosuus	Kuinka iso osuus partneriyrityksellä on heidän markkinastansa	(Wuetal.,2009)
49 Asiakkaiden diversiteetti	Kuinka laaja-alaisesti partneriyritys palvelee erilaisia asiakkaita, vai onko kaikki asiakkaat samanlaisia	(Wuetal.,2009)
50 Partnerin verkostot	Millaiset verkostot partneriyrityksellä on ja voiko ohjelmistoyritys hyötyä niistä	(Giessmann & Stanoevska-Slabeva,2012), (Angeles & Nath,2000), (Bierly III & Gallagher,2007)
Ohjelmistoyrityksen näkökulma		
51 Suhde kilpailijoihin	Onko partneriyrityksellä suhdetta ohjelmistoyrityksen kilpailijoihin	(Giessmann &Stanoevska-Slabeva,2012)
52 Monialainen	Onko partneriyrityksen tuotteen tarkoitus palvella monialaisesti vai tiettyä kohderyhmää	(Schreiecketal.,2016)
53 Avoimuus kilpailijoiden väliseen yhteistyöhön	Onko partneryritys valmis tekemään yhteistyötä potentiaalisen kilpailijan kanssa yhteistä hyvää varten.	(Schreiecketal.,2016)
54 Avoimuus yhteiskehitykseen	Onko partneriyritys valmis yhteiseen tuotekehitykseen ohjelmistoyrityksen kanssa	(Hitt etal.,2000),(Möller&Rajala,2007), (Meffert &Swaminathan,2017)
55 Portfolion täydentäminen	Täydentääkö partnerin tuote ohjelmistoyrityksen portfoliota	(Schreiecketal.,2016)
56 Yhteistyön sijoitetun pääoman tuotta	Kuinka iso on yhteistyöhön sijoitetun pääoman tuotto	(Chen etal.,2008)

57 Suositukset	Onko partneriyrityksellä suosittelijoita	(Keung &Griffiths,2008)
58 Regulaatioiden tietämys	Onko partneriyrityksellä tietämystä paikallisesta regulaatiosta	(Mat et al.,2009), (Das&He,2006)

Liite 2. Tulostaulukko osa 1.

Customers count	Customer_share_%	Querycount	Queries_per_customer	MRR + TREV	Percent_of_customers_in_segment	ARPC	Revenue	EBIT_%	Revenue_change_%	Staff_count	EBITDA_%	Profit
1,00	1,00	1,00	1,00	1,00	1,00	1,00	0,00	0,00	1,00	0,00	0,00	1,00
1,00	0,99	1,00	1,00	1,00	0,28	0,08	0,96	0,10	0,09	0,95	0,42	1,00
1,00	0,98	1,00	1,00	1,00	0,14	0,00	0,96	0,17	0,00	0,97	0,24	1,00

Liite 3. Tulostaulukko osa 2.

Revenue_per_staff	MRR_per_query	Partner_rank	Portfolio_rank	Cluster_rank	Score_rank	Partner Name	Product_name	Company_name	Portfolio	Partnership	Cluster
1,000	0,464	0,216	0,580	0,233	1,000	-	-	-	-	-	Cluster 2
0,844	0,006	0,233	0,591	0,244	0,963	-	-	-	-	-	Cluster 2
0,289	0,004	0,239	0,568	0,222	0,955	-	-	-	-	-	Cluster 5