HAMK
Hämeen ammatti-
korkeakoulu

KONEOPPIMISEN HYÖDYNTÄMINEN VARASTON ALAVIRRAN SUUNNITTELUSSA

Ylemmän ammattikorkeakoulututkinnon opinnäytetyö

Tietojohtaminen ja älykkäät palvelut

Kevät 2024

Sauli Virtanen

HAMK
Hämeen ammatti-
korkeakoulu

Tietojohtaminen ja älykkäät palvelut — Tiivistelmä
Tekijä Sauli Virtanen — Vuosi 2024
Työn nimi Koneoppimisen hyödyntäminen varaston alavirran suunnittelussa
Ohjaaja Iivari Kunttu

Logistiikan toimitusketjujen hallinta on keskeinen osa yritystoimintaa. Materiaali- ja palveluvirtojen sujuvuus sekä tehokkuus ovat ratkaisevassa asemassa. Toimitusketjun optimointiin liittyy useita haasteita, kuten asiakkaiden vaihteleva kysyntä ja varastonhallinta. Erityisesti pitkät toimitusketjut välivarastoineen ja useine toimijoineen edellyttävät kokonaisvaltaista suunnittelua ja johtamista. Kysyntäketjun hallinnalla on merkittävä rooli logistiikan suunnittelussa, sillä asiakkaiden ennustamaton kysyntä voi vaikuttaa koko toimitusketjun suorituskykyyn.

Tutkimuksessa tarkastellaan koneoppimisen hyödyntämistä varaston alavirran suunnittelussa. Tutkimuksen kohteena on kuvitteellinen tavarantoimittajana toimiva tukkuliike, joka pyrkii optimoimaan varastokenttäänsä ja logistiikkaansa asiakkaiden tilaushistorian perusteella. Tutkimuksen tavoitteena on luoda koneoppimisen malli, joka ennustaa asiakkaiden tilauksia aiempien tilausten perusteella ja sitä kautta auttaa ylläpitämään oikean määrän oikeita tuotteita sopivissa varastoissa. Tutkimuksessa käytettävä aineisto on keinotekoista ja luotu vain tätä opinnäytetyötä varten.

Opinnäytetyössä käsitellään ensin koneoppimisen perusteita ja valitaan sopiva algoritmi tai algoritmit mallin rakentamiseen. Sen jälkeen toteutetaan koneoppimismalli ja analysoidaan mallin toimivuutta.

Tutkimustyön tuloksena syntyi päätöspuualgoritmin ja ryvästys algoritmin hyödyntämisestä esimerkki, jota voi hyödyntää rakennettaessa vastaavaa monimutkaisempaa koneoppimismallia. Synteettinen data palveli tarkoitustaan ja datan tuntemisesta oli hyötyä arvioitaessa koneoppimisalgoritmin suoriutumista. Yksinkertaisuutensa vuoksi data ei kuitenkaan vastaa tosielämän tarpeita eikä keinotekoisella datalla koulutettu koneoppimismalli todennäköisesti sellaisenaan toimi missään tilanteessa.

Avainsanat koneoppiminen, logistiikka, toimitusketjut
Sivut 39 sivua ja liitteitä 0 sivua

HAMK
Hämeen ammatti-
korkeakoulu

Data Leadership and Smart Services — Abstract
Author Sauli Virtanen — Year 2024
Subject Utilizing Machine Learning in Downstream Inventory Planning
Supervisors Iivari Kunttu

The management of logistics supply chains is a central part of business operations. The smoothness and efficiency of material and service flows are crucial. Supply chain optimization involves several challenges, such as inventory management and constantly changing customer demand. Particularly long supply chains with interim storage and multiple actors require comprehensive planning and management. Demand chain management plays a significant role in logistics planning, as unpredictable customer demand can affect the performance of the entire supply chain.

This study examines the utilization of machine learning in downstream inventory planning. The study focuses on a fictional wholesale company acting as a supplier, aiming to optimize its inventory and logistics based on customer order history. The goal of the study is to create a machine learning model that predicts customer orders based on previous orders and thereby helps maintain the right amount of the right products in suitable warehouses. The data used in the study is artificial and created solely for this thesis.

The thesis first discusses the basics of machine learning and selects a suitable algorithm or algorithms for building the model. Then, a machine learning model is implemented, and the functionality of the model is analyzed.

As a result of the research, examples of utilizing decision tree algorithm and clustering algorithm were created. The examples can be used when building a more complex machine learning model. Synthetic data served its purpose, and understanding the data was beneficial in evaluating the performance of the machine learning algorithm. However, due to its simplicity, the data does not meet real-life needs, and a machine learning model trained on artificial data is unlikely to work as is in any situation.

Keywords machine learning, logistics, supply chain management
Pages 39 pages and appendices 0 pages

Sisällys

Kuvat, taulukot ja kaavat

1 Johdanto

Logistiikan toimitusketjujen hallinta on keskeinen osa yritystoimintaa. Materiaali- ja palveluvirtojen sujuvuus sekä tehokkuus ovat ratkaisevassa asemassa. Toimitusketjun optimointiin liittyy useita haasteita, kuten asiakkaiden vaihteleva kysyntä ja varastonhallinta. Erityisesti pitkät toimitusketjut välivarastoineen ja useine toimijoineen edellyttävät kokonaisvaltaista suunnittelua ja johtamista. Kysyntäketjun hallinnalla on merkittävä rooli logistiikan suunnittelussa, sillä asiakkaiden ennustamaton kysyntä voi vaikuttaa koko toimitusketjun suorituskykyyn.

Tässä tutkimuksessa tarkastelen koneoppimisen hyödyntämistä varaston alavirran suunnittelussa. Alavirralla tarkoitetaan materiaalin kulkua toimittajalta asiakkaalle (Logistiikan maailma. n.d.). Tutkimuksen kohteena on kuvitteellinen tavarantoimittajana toimiva tukkuliike, joka pyrkii optimoimaan varastokenttäänsä ja logistiikkaansa asiakkaiden tilaushistorian perusteella. Tutkimuksen tavoitteena on luoda koneoppimisen malli, joka ennustaa asiakkaiden tilauksia aiempien tilausten perusteella ja sitä kautta auttaa ylläpitämään oikean määrän oikeita tuotteita sopivissa varastoissa. Tutkimuksessa käyttämäni aineisto on keinotekoista ja loin sen vain tätä opinnäytetyötä varten. Vaikka keinotekoisen datan käyttöön liittyy ongelmia, kuten aineiston yksinkertaisuus tai vinoumat, joiden vuoksi keinotekoisella aineistolla luotu malli ei toimisi tosielämän datalla, valitsin tämän lähestymistavan tietosuojasyistä ja siksi, että pystyn helpommin todentamaan koneoppimismallin toimisen halutulla tavalla.

Sovellan tutkimuksessa konstruktiivista lähestymistapaa, jonka keskeisenä tavoitteena on käytännön ongelmien ratkaisu konkreettisten tuotosten avulla. Tutkimusprosessi noudattelee Ojasalon, Moilasen ja Ritalahden (2015) esittelemää konstruktiivisen tutkimuksen prosessia.

Opinnäytetyössä käsittelen ensin koneoppimisen perusteita ja valitsen sopivan algoritmi tai algoritmit mallin rakentamiseen. Sen jälkeen toteutan koneoppimismallin ja analysoin mallin toimivuutta. Lopuksi arvioin laaditun mallin käyttökelpoisuutta.

2 Tutkimustehtävä

2.1 Tutkimusongelma

Logistiikan toimitusketjun kokonaisuus koostuu materiaali- ja palveluvirroista sekä niihin liittyvistä raha- ja tietovirroista. Toimitusketjuun kuuluu useita eri organisaatioita, joilla jokaisella on oma roolinsa. Toimitusketjun rakenne riippuu tuotteista, toimialasta ja asiakkaista. Toimitusketju yhdistää yrityksen ja sen tavarantoimittajat jakeluorganisaatioihin ja asiakkaisiin. (Ritvanen et al., 2011, s.22)

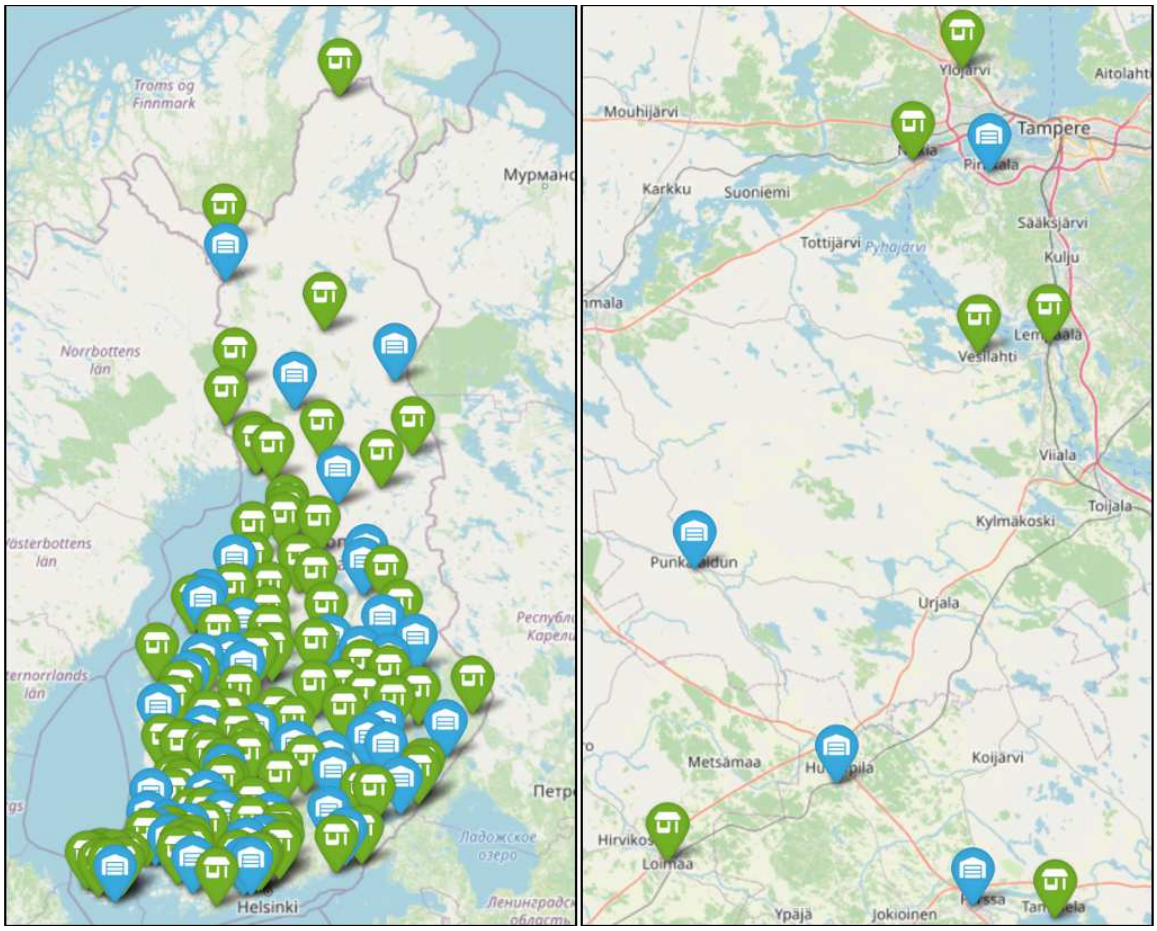
Pitkissä toimitusketjuissa esimerkiksi välivarastot, logistiikkayritykset ja tukkuliikkeet ovat yleisiä. Toimitusketjun hallinnalla tarkoitetaan yritysverkoston materiaalivirran ja siihen liittyvien tieto- ja rahavirtojen kokonaisvaltaista suunnittelua, ohjausta ja johtamista. Toimitusketjun hallinnan rinnalla puhutaan myös kysyntäketjun hallinnasta silloin, kun halutaan korostaa kysynnän merkitystä logistiikan kokonaisuuden suunnittelussa. Jos asiakkaiden kysyntää ei kyetä ennustamaan ja hallitsemaan, se johtaa koko toimitusketjun huonoon suorituskykyyn. (Ritvanen et al., 2011, s.23).

Tässä tutkimuksessa tavarantoimittajana toimivalla tukkuliikkeellä on tiedossaan asiakasyritystensä käyttämät tai edelleen myymät tuotteet sekä halutut varastotasot. Tunnistamalla asiakkaiden tuotenimikekohtaisen menekin toimittaja pyrkii optimoimaan omaa varastokenttäänsä ja logistiikkaansa.

Tutkimusta varten olen luonut skenaarion, jossa tavarantoimittajalla on maanlaajuinen 50 varaston verkosto. Tuotevalikoimaan kuuluu 5000 nimikettä. Varastokohtainen valikoima vaihtelee hieman, sillä tilan puutteen ja tuotteiden erilaisten varastointivaatimusten vuoksi kaikkea ei voi olla saatavilla kaikissa varastoissa.

Tavarantoimittajalla on 100 toimituspistettä, joista jokaista käyttää viidestä kymmeneen asiakasyritystä. Tyypillinen asiakkaan tilaus sisältää keskimäärin 100 nimikettä. Kuvassa 1 on esitetty varastojen (sininen merkki) ja toimituspisteiden (vihreä merkki) sijoittuminen maantieteellisesti. Vasemmanpuoleisessa kartassa ovat kaikki kohteet, kun taas oikeanpuoleinen on esimerkki Tampereen eteläpuoliselta alueelta.

Kuva 1 - Toimittajan varastojen ja toimituspisteiden sijoittuminen kartalle



Tilausten kuljettaminen toimituspisteisiin edellyttää lähetyksiä useilta tavarantoimittajan varastoilta. Logistiikan suunnittelu on työlästä ja aikaa vievää.

Tutkimuksen tavoitteena on luoda koneoppimisen malli, joka ennustaa erityyppisten asiakkaiden tilauksia aiempien tilausten perusteella ja sitä kautta auttaa ylläpitämään oikean määrän oikeita tuotteita sopivissa varastoissa. Koneoppimisen käsite on tarkemmin avattu luvussa 5.1.

Tutkimuskysymykset ovat:

1. Millaisen koneoppimisen mallin avulla voidaan ennustaa erityyppisten asiakkaiden tilauksia?

2. Millaisen koneoppimisen mallin avulla voidaan määritellä kunkin asiakkaan kannalta optimaaliset lähettävät varastot?

3. Mitä dataa on ylläpidettävä tutkimuksessa kuvattujen koneoppimisen mallien toimimiseksi?

2.2 Käytettävä empiirinen aineisto

Tutkimuksen aineisto sisältää

- Tiedot toimittajan varastoista
- Tiedot toimituspisteistä
- Asiakasrekisterin
- Toimittajan valikoimissa olevien nimikkeiden tiedot
- Historiatiedon aikaisemmin käsitellyistä tilauksista

Aineisto ei vastaa mitään todellista tilannetta vaan olen rakentanut sen ainoastaan tätä opinnäytetyötä varten käyttämällä OpenAI:n ChatGPT-kielimallia, Microsoft Exceliä ja Python-ohjelmointikieltä.

Varastotietotaulu (kuva 2) sisältää toimittajan 50 varastosta varaston tunnusnumeron (1-50 sekä varaston sijainnin koordinaatistossa (X,Y). Varaston tyyppi ja sijainti ovat satunnaisia.

Kuva 2 - Ote varastotietokannasta

	Store_Num	X	Y
0	1	29.25	63.23
1	2	20.42	60.01
2	3	28.48	62.97
3	4	24.80	60.46
4	5	26.75	63.23

Toimituspisteiden tietotaulu (kuva 3) on 100 tietueen taulukko. Toimituspisteen tietoihin kuuluvat tunnusnumeron (1-100) lisäksi satunnainen sijainti koordinaatistossa (X,Y) sekä tieto toimituspisteen operaattorista, joka on numeroitu tunnuksella 1000, 2000, 3000, 4000 tai 5000.

Kuva 3 - Ote toimituspistetietokannasta

	Delivery_Point_Num	X	Y	Operator
0	1	24.44	64.29	1000
1	2	20.77	60.25	1000
2	3	27.12	62.62	1000
3	4	30.92	62.67	1000
4	5	25.47	65.01	1000

Asiakasrekisterissä (kuva 4) on 4000 asiakkaan tiedot, jotka sisältävät 6-numeroisen satunnaisen asiakasnumeron, asiakkaan käyttämän toimituspisteen ja toimituspisteen

operaattorin sekä asiakkaan prioriteetin (1-4). Asiakaskohtaisten attribuuttien avulla on tarkoitus kyetä paremmin erottelemaan eri asiakkaat toisistaan.

Kuva 4 - Ote asiakasrekisteristä

	Customer_Num	Customer_Type	Customer_Subtype	Customer_Class	Operator	Delivery_Point	Priority
0	731081	J	1	J1	1000	19	4
1	456813	A	3	A3	4000	76	1
2	970429	E	1	E1	1000	1	2
3	960082	A	3	A3	4000	84	1
4	866245	E	2	E2	5000	99	3
5	953546	A	3	A3	3000	66	3
6	572605	D	1	D1	5000	96	4
7	778203	H	1	H1	1000	5	2
8	112748	A	2	A2	1000	14	3
9	581933	C	1	C1	4000	72	1

Kuva 5 esittää otetta toimittajan varastoissa olevien nimikkeiden tietokannasta. Erilaisia nimikkeitä on 541 kappaletta. Ne on jaettu kategorioihin, kuten ”SALAATIT” ja luokkiin, kuten ”A-SAL03 KANASALAATIT”. Jokaisella nimikkeellä on 8-numeroinen satunnainen tunnus.

Kuva 5 - Ote toimittajan nimiketietokannasta

	Item_Category	Item_Class_ID	Item_Class_Name	Item_ID	Item_Name
0	HEDELMÄT	A-HED01	SITRUKSET	86851856	APPELSIINI
1	HEDELMÄT	A-HED01	SITRUKSET	72713852	SITRUUNA
2	HEDELMÄT	A-HED01	SITRUKSET	40812828	LIME
3	HEDELMÄT	A-HED02	MARJAT	82928456	MANSIKKA
4	HEDELMÄT	A-HED02	MARJAT	84874866	MUSTIKKA
5	HEDELMÄT	A-HED03	TROOPPISET HEDELMÄT	40074101	ANANAS
6	HEDELMÄT	A-HED03	TROOPPISET HEDELMÄT	11380753	MANGO
7	HEDELMÄT	A-HED03	TROOPPISET HEDELMÄT	46951210	KIIVI
8	HEDELMÄT	A-HED04	OMENAT	32142806	GALA-ÖPPLE
9	HEDELMÄT	A-HED04	OMENAT	91623549	GRANNY SMITH -OMENA
10	VIHANNEKSET	A-VIH01	JUUREKSET	87775026	PORKKANA
11	VIHANNEKSET	A-VIH01	JUUREKSET	95960536	LANTTU
12	VIHANNEKSET	A-VIH01	JUUREKSET	84603879	PALSTERNAKKA
13	VIHANNEKSET	A-VIH02	LEHTIVIHANNEKSET	11134107	JÄÄSALAATTI
14	VIHANNEKSET	A-VIH02	LEHTIVIHANNEKSET	56577711	PUNASALAATTI
15	VIHANNEKSET	A-VIH02	LEHTIVIHANNEKSET	70022021	RUCOLA
16	LIHA	A-LIA01	NAUDAN LIHA	30781595	PAAHTOPAISTI
17	LIHA	A-LIA01	NAUDAN LIHA	14072716	KULMAFILEE
18	LIHA	A-LIA01	NAUDAN LIHA	61344917	JAUHELIHA
19	LIHA	A-LIA02	SIAN LIHA	87933108	KASSLER

Rakensin nimiketietokannan käyttäen apuna ChatGPT-kielimallia. Käskin kielimallia ensin keksimään esimerkiksi kymmenen lemmikkitarvikkeisiin liittyvää kategoriaa, jonka jälkeen teetin taulukon kuhunkin kategoriaan kuuluvista tuotteista esimerkiksi näin: ”Tee taulukko kategoriasta "KISSANRUOKA". Taulukon otsikot: "Kategoria";"Luokan tunnus";"Luokan nimi";"Tuotetunnus";"Tuotenimi". Taulukossa tulee olla 10 erilaista luokkaa KISSANRUOKIA”. Kielimallin avulla oli mahdollista keksiä, luokitella ja listata suuri määrä erilaisia tuotteita kohtalaisen nopeasti. Osaa mallin keksimistä tuotteista ei ole olemassa, mutta sillä ei ole tutkimustyön kannalta merkitystä.

Käsiteltyjen tilausten tietotaulu (kuva 6) sisältää 420039 tietuetta. Tallennetut tiedot ovat:

- Asiakasnumero (Customer_Num)
- Asiakkaan tyyppi (Customer_Type)
- Asiakkaan tyypin tarkenne (Customer_Subtype)
- Asiakkaan luokka (tyypin ja tarkenteen yhdistelmä) (Customer_Class)
- Toimituspisteen operaattori (Operator)
- Toimituspiste (Delivery_Point)
- Asiakkaan prioriteetti (Priority)
- Tilattavan materiaalin luokka (Class_ID, Class_Name)
- Tilattavan materiaalin nimike (Item_ID, Item_Name)

Toimitettu määrä (Provided)

Kuva 6 - Tiedot aikaisemmin manuaalisesti käsitellyistä tilauksista

	Customer_Num	Customer_Type	Customer_Subtype	Customer_Class	Operator	Delivery_Point	Priority	Class_ID	Class_Name	Item_ID	Item_Name	Provided
0	731081	J	1	J1	1000	19	4	F-AH-001	AUTOSHAMPOO	67042927	SONAX AUTOSHAMPOO	209
1	731081	J	1	J1	1000	19	4	F-AH-001	AUTOSHAMPOO	58908704	TURTLE WAX SUPER HARD SHELL AUTOSHAMPOO	531
2	731081	J	1	J1	1000	19	4	F-AH-002	AUTOVAHAT	76906724	MEGUIAR'S GOLD CLASS CARNAUBA PLUS PREMIUM AUT...	117
3	731081	J	1	J1	1000	19	4	F-AH-002	AUTOVAHAT	73059125	COLLINITE 845 INSULATOR WAX	95
4	731081	J	1	J1	1000	19	4	F-AH-003	SISÄPUHDISTUS	18755318	ARMOR ALL LEATHER CARE GEL	124
5	731081	J	1	J1	1000	19	4	F-AH-003	SISÄPUHDISTUS	75521224	CHEMICAL GUYS INNERCLEAN INTERIOR QUICK DETAIL...	124
6	731081	J	1	J1	1000	19	4	F-AH-004	ULKOPUHDISTUS	81149293	GTECHNIQ W5 CITRUS ALL PURPOSE CLEANER	0
7	731081	J	1	J1	1000	19	4	F-AH-004	ULKOPUHDISTUS	97884589	AUTOGLYM POLAR BLAST SNOW FOAM	276
8	731081	J	1	J1	1000	19	4	F-AH-005	VANTEIDENHOITO	60995090	SONAX XTREME WHEEL CLEANER PLUS	192
9	731081	J	1	J1	1000	19	4	F-AH-005	VANTEIDENHOITO	11916978	MEGUIAR'S HOT RIMS WHEEL & TIRE CLEANER	226

Koska koneoppimisalgoritmin on kyettävä löytämää aineistosta säännönmukaisuuksia, eivät taulun tiedot voi olla satunnaisia. Rakensin taulun asettamalla eri asiakkaan parametreille kertoimia suhteessa kunkin nimikkeen toimitettuun määrään.

Opinnäytetyössä käytetyt tietotaulut on tallennettu GitHubiin (https://github.com/sauvirtan/ONT2024.git). GitHub on kehittämisalusta, joka antaa mahdollisuuden luoda, tallentaa, hallinnoida ja jakaa koodia (Wikimedia Foundation. n.d.)

3 Tutkimusmenetelmät

3.1 Toteuttamistapa

Hämeen ammattikorkeakoulun opinnäytetyöohjeessa esitellään neljä erilaista opinnäytetyön toteuttamistapaa (HAMK, 2023):

- Tutkimuspainotteinen opinnäytetyö
- Toiminnallinen opinnäytetyö
- Portfolio-opinnäytetyö
- Artikkeliopinnäytetyö

Opinnäytetyö on mahdollista toteuttaa julkaisemalla ajankohtaisia ja uusia ajatuksia sisältäviä artikkeleita suurta yleisöä tai tiettyä ammattialaa kiinnostavasta aiheesta. Portfolio-opinnäytetyö taas koostuu opintojen aikana suunnitelmallisesti tehdyistä projekteista, jotka opinnäytetyössä kootaan yhdeksi tutkittavaksi kokonaisuudeksi. (HAMK, 2023)

Toiminnallisessa opinnäytetyössä kehitetään, toteutetaan ja arvioidaan uusia tuotteita, palveluja, toimintatapoja tai työkäytäntöjä (HAMK, 2023). Tämän opinnäytetyön toteuttaminen toiminnallisena edellyttäisi laadittavan koneoppimisen mallin soveltamista käytännössä osana jonkun toimijan logistiikan suunnittelua ja tulosten tarkastelua.

Koska tätä opinnäytetyötä ei ole sidottu minkään tietyn yrityksen tai yhteisön toimintaan eikä koneoppimisen mallin tuloksia tarkastella tosielämän datan perusteella, toteutan työn tutkimuspainotteisena. Tutkimuksen kohteena on koneoppimisen malli ja erilaiset vaihtoehdot sen toteuttamiseksi, ei esimerkiksi sen tuottama lisäarvo liiketoiminnalle. HAMK:n opinnäytetyöohjeen mukaan ”Tutkimuspainotteisen opinnäytetyön lähtökohtana on selkeästi muotoiltu työelämälähtöinen tutkimusongelma, johon haetaan vastausta käyttäen tarkoituksenmukaisia aineistoja ja yleisiä tutkimusmenetelmiä” (HAMK, 2023).

Katri Ojasalo, Teemu Moilanen ja Jarmo Ritalahti kuvaavat kirjassaan *Kehittämistyön menetelmät (2015)* selkeästi ja ymmärrettävästi tutkimuksellista kehittämistyötä. Teos on

kirjoitettu liiketoiminnan kehittämisen näkökulmasta ja sen esittämään toimintamalliin kuuluu oleellisena osana kehitystyön tuloksen arviointi ja testaaminen aidossa ympäristössä. Vaikka en kirjoitakaan tätä opinnäytetyötä minkään tietyn yrityksen toimintaan liittyen, pidän kirjan kuvaamaa tutkimuksellisen kehittämistyön prosessia (Ojasalo et al, 2015, s. 23–24) käyttökelpoisena oppaana tutkimusongelmani ratkaisemisessa.

Kuvassa 7 olen mukaillut kirjan mallia tutkimuksellisen kehittämistyön prosessista. Kuten tutkimusongelman määrittelyssä totesin, olen asettanut tutkimuksen tavoitteeksi luoda koneoppimisen malli, joka ennustaa asiakkaiden tilauksia aiempien tilausten perusteella ja sitä kautta auttaa ylläpitämään oikean määrän oikeita tuotteita sopivissa varastoissa. Tutkimustyö on uudistamisperustainen (Ojasalo et al, 2015, s. 26). En siis keskity käytännössä havaitun ongelman ratkaisemiseen vaan tutkin mahdollisuutta ottaa koneoppimisalgoritmeja käyttöön logistiikan toimintaprosessissa. Kehittämiskohde on logistiikan toimitusketjun hallinta, tarkemmin Efficient Consumer Response (ECR) ja Collaborative Planning, Forecasting and Replenishment (CPFR) (Ritvanen et al., 2011, s. 142). Nämä ovat toimintamalleja, joilla pyritään optimoimaan valikoima, vähentämään varastoja, parantamaan saatavuutta ja vastaamaan asiakkaiden kysyntään yhteistyössä vähittäiskaupan tavarantoimittajien ja jakeluketjun jäsenten kesken (Ritvanen et al., 2011, s. 142).

Kuva 7 - Tutkimuksellisen kehittämistyön prosessi (Ojasalo et al, 2015, s. 24)

Ennen kehittämistehtävän määrittämistä Ojasalo, Moilanen ja Ritalahti (2015) painottavat kehittämiskohteeseen perehtymisen tärkeyttä (kuva 7, kohta 2). Jos tavoitteenani olisi esimerkiksi tietyn yrityksen toiminnan parantaminen, edellyttäisi se perusteellista yrityksen toimitusketjun hallinnan tuntemusta. Tässä opinnäytetyössä lähtöasetelma on kuitenkin täysin fiktiivinen ja motivaationa työn tekemiselle on oppia soveltamaan koneoppimisen algoritmeja. Lopputulos on hyödyllinen, jos tutkimustyössä esitellyn tavan kaltainen menetelmä datan käsittelyssä on sovellettavissa tosielämään. Kehittämiskohteeseen perehtymisessä korostuvat toimialan (logistiikan) ymmärtäminen, käsitteistön tuntemus ja toisaalta koneoppimisen opiskelu. Tutkimuskirjallisuuden hankin HAMK:n Tietojohtaminen ja älykkäät palvelut-koulutusohjelman kurssimateriaalin lisäksi tekemällä Wikipedia-hakuja logistiikan ja koneoppimisen käsitteistä ja tutkimalla näin löytyneiden artikkeleiden lähdeluetteloja. Osan lähdeteoksista lainasin fyysisinä kirjoina, mutta pääosin käytin HAMK:n Finna-tietokannan e-kirjoja.

Kehittämistehtävää (kuva 7, kohta 3) ei pidä sekoittaa tutkimusongelmaan eikä sitä tarvitse pukea kysymysmuotoon (Ojasalo et al, 2015, s. 32). Kehittämistehtävän onnistumisen arvioimiseksi olisi myös luotava mittaristo, jonka avulla voitaisiin pisteyttää kehittämistehtävän onnistuminen sen käyttöönoton tai testaamisen jälkeen. Koska tämän opinnäytetyön lopputuloksena syntyvää koneoppimisen mallia ei voitane sellaisenaan käyttää eikä sen toimintaa käytännössä mitata, pidän mallin tuottamien ennusteiden laskennallista virhettä ainoana tapana mitata kehittämistehtävän, eli koneoppimisen mallin, onnistumista.

Tutkimuksellisen kehittämisprosessin mallin neljännen kohdan ”Tietoperustan laatiminen ja menetelmien suunnittelu” käsittelen seuraavissa alaluvuissa.

3.2 Tietoperusta

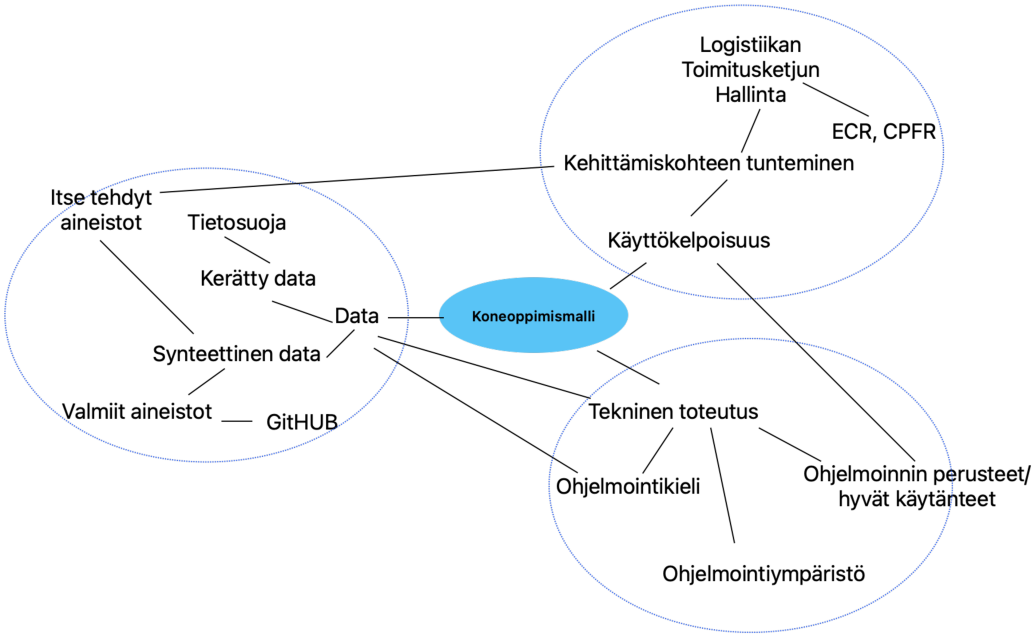
Tutkimustyön tietoperustalla tarkoitetaan olemassa olevaa teoria-aineistoa, jota käytetään tutkittavaan aihealueeseen perehtymiseen. Siitä voidaan käyttää myös termejä kirjallisuuskatsaus, teoreettinen viitekehys tai teoriatausta. (Ojasalo et al, 2015, s. 34).

Ojasalo, Moilanen ja Ritalahti (2015) tiivistävät tietoperustan käsitejärjestelmäksi, jossa käsitteet ja niiden väliset suhteet tulevat määritellyiksi. Käsitejärjestelmän tulisi sekä jäsentää kerättyä tietoa, että ohjata uuden tiedon etsintää. Käsitejärjestelmää koottaessa tutkijan tulisi selventää itselleen, mitkä ovat kehittämiskohteeseen liittyvät käsitteet ja niiden väliset suhteet sekä mahdolliset eri lähteiden ristiriidat tai eriävät näkökulmat. Käsitejärjestelmän

laatimisen voi aloittaa esimerkiksi miellekartan (mind map) laatimisella (Ojasalo et al, 2015, s. 35). Kirjassa Hyvä, parempi, valmis (2022) Juha Hakala suosittelee hieman vastaavalla tavalla miellekartan laatimista opinnäytetyön jäsentämiseksi. Tulkitsen tämän niin, että miellekarttaan päätyvät termit auttavat etsimään osuvaa lähdemateriaalia kehittämiskohteeseen liittyvistä aiheista.

Kuvassa 8 on yksinkertainen opinnäytetyön jäsentämiseksi ja tiedonhaun perustaksi luotu miellekartta. Olen asettanut tärkeimmäksi kehitystyöhön liittyväksi käsitteeksi kehittämistehtävän eli koneoppimismallin. Tähän liittyvät asiasanat olen jakanut kolmeen kokonaisuuteen, joista yksi liittyy kehittämiskohteen, logistiikan toimitusketjun hallinnan, tuntemiseen, toinen mallin rakentamisen tekniseen toteutukseen eli ohjelmointiin ja kolmas dataan, jota mallin on tarkoitus käsitellä. Alla tarkastelen joitakin näihin kokonaisuuksiin liittyvä opinnäytetyön lähteenä käyttämiäni julkaisuja.

Kuva 8 - Käsitekartta



*Logistiikan ja toimitusketjun hallinnan perusteet (*Ritvanen et al., 2011) on suunniteltu tarjoamaan ammattikorkeakouluille työelämän tarpeita vastaavaa opiskelumateriaalia logistiikan koulutukseen. Kirja esittelee logistiikan ja toimitusketjun hallinnan perusteita. Kirjan tiedot ovat osittain vanhentuneita ja sitä täydentää alati päivittyvä *Logistiikan maailma*-verkkosivusto, jota julkaisee Reijo Rautauoman säätiö. Verkkosivusto ja kirja yhdessä antavat logistiikan toimitusketjun hallinnasta riittävän ymmärryksen koneoppimismallin

tarvitseman datan ja itse mallin rakentamisesta niin, että sen voisi sopivasti sovellettuna kuvitella palvelevan jotain tosielämän käyttötapausta.

Hello world: Kuinka selviytyä algoritmien aikakaudella (Fry, 2019) kuvaa helposti ymmärrettävällä tavalla, mitä algoritmit ovat sekä miten tekoälyä sovelletaan nyt ja mahdollisesti tulevaisuudessa. Kirja on suunnattu suurelle yleisölle ja auttaa ymmärtämään tekoälyn käsitettä yleisesti.

Python: An Introduction to Programming. (Parker, 2021) on Python-ohjelmointikielen perustason oppikirja, jota tässä tutkimuksessa käytän sekä koneoppimisen mallin rakentamisen tukena, että apuna pohtiessani sitä, onko tutkimusongelma järkevintä ratkaista koneoppimisen vai perinteisen ehtoihin perustuvan ohjelmoinnin kautta.

Koneoppiminen. (Alpaydin, 2021) on kattava johdatus koneoppimisen periaatteisiin, menetelmiin ja sovelluksiin. Kirja alkaa käsittelemällä koneoppimisen peruskäsitteitä ja menetelmiä ja etenee myös syvemmälle erilaisten koneoppimisen tekniikoiden toteuttamiseen.

Koneoppimisen perusteet (Kämäräinen, 2023) on tarkoitettu ensimmäiseksi oppikirjaksi koneoppimiseen. Kirja on käyttämistäni lähteistä tuorein. Teos on erittäin helposti ymmärrettävä ja nimensä mukaisesti keskittyy perusteiden opettamiseen. Kirjailijan alkusanojen mukaisesti ”kirjassa on tehty muutamia myönnytyksiä suomen kielen suhteen, jotta esitystapa ei olisi ristiriidassa englanninkielisen kirjallisuuden kanssa”. Tämän myötä kirja on hyvä lähde valittaessa sopivia termejä käytettäväksi suomenkielisessä opinnäytetyössä.

Introduction to Machine Learning (Alpaydin, 2014) on suunniteltu opetuskäyttöön korkeakouluissa ja yliopistoissa, mutta se on myös hyödyllinen resurssi itsenäiseen opiskeluun. Alpaydin tarjoaa runsaasti esimerkkejä ja harjoituksia, jotka auttavat lukijoita ymmärtämään käsitteitä ja soveltamaan niitä käytännön ongelmiin. Lisäksi teos sisältää runsaasti kuvia ja kaavioita, jotka selventävät abstrakteja käsitteitä.

Tekoälyn perusteita ja sovelluksia (Tuominen & Neittaanmäki, 2019) on Jyväskylän yliopiston Informaatioteknologian tiedekunnan julkaisema kirja, joka on ladattavissa yliopiston JYX-julkaisuarkistosta.

Hands-On Machine Learning with Scikit-Learn and TensorFlow (Géron, 2017) on koneoppimisen konsultin ja entisen Googlen ohjelmoijan kirjoittama opas ohjelmoijille, jotka haluavat soveltaa koneoppimista. Teoksessa käsitellään ensin koneoppimisen perusteet käyttäen Scikit-Learn-kirjastoa, joka on Python-ohjelmointikielen yleisimmin käytetty koneoppimiskirjasto. Sen jälkeen siirrytään syväoppimisen ja neuroverkkojen maailmaan TensorFlow-kirjaston avulla. Kirjassa on paljon käytännön esimerkkejä ja harjoituksia, jotka auttavat ymmärtämään koneoppimisen käsitteitä ja soveltamaan niitä. Lisäksi kirja käsittelee hyvin esimerkiksi mallin arviointia ja virheiden analysointia. "Hands-On Machine Learning with Scikit-Learn and TensorFlow" on erinomainen resurssi kaikille, jotka haluavat oppia koneoppimisen käytännön sovelluksia Python-ohjelmointikielen avulla.

Lukuun ottamatta kolmea ensin mainittua kirjaa kaikki päälähteeni käsittelevät data-analyysiä ja koneoppimista. Niissä ei juurikaan esiinny Ojasalon, Moilasen ja Ritalahden (2015) mainitsemia ristiriitoja tai eriäviä näkökulmia, sillä kaikissa käsitellään samoja oikeaksi todistettuja matemaattisia kaavoja. ”Lähteiden keskustelua” ei ehkä sen vuoksi tämän aiheen tutkimustyössä merkittävästi synny, mutta itselleni koneoppimisen omaksumisen kannalta useampi samaa asiaa käsittelevä lähde on ollut hyödyllinen, sillä jossain kirjassa vaikeasti ymmärrettävä aihe on saatettu toisessa teoksessa selittää paremmin.

Aihealueen aikaisemmista tutkimuksista nostan esille seuraavat:

- Jeremias Penttilän opinnäytetyö *Koneoppiminen* Jyväskylän yliopistossa 2015 selventää, mitä koneoppiminen on ja mitä menetelmiä se pitää sisällään.
- Katia Nkulizan opinnäytetyö *Tekoäly: mahdollisuudet ja näkymät logistiikassa* Kaakkois-Suomen ammattikorkeakoulussa 2020 avaa tekoälyä käsitteenä ja sen yhteyttä logistiikkaan.
- Oiva Penttilän opinnäytetyö *Logistiikan kehittäminen datalla* Hämeen ammattikorkeakoulussa 2023 on tapaustutkimus Uusioaines Oy:n keräämän tai kerättävissä olevan datan käytön mahdollisuuksista.

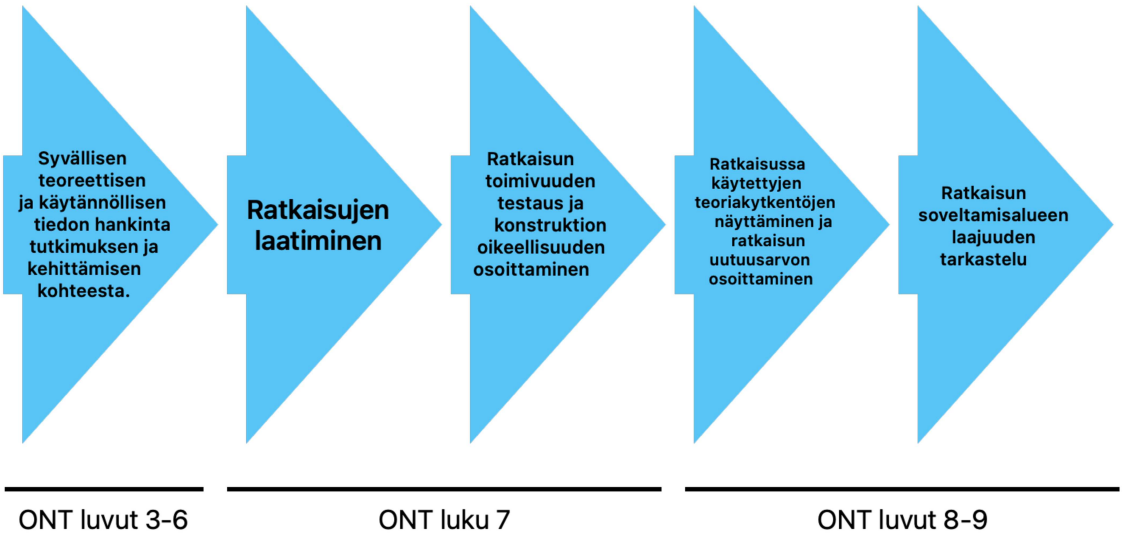
Anssi Lotvosen opinnäytetyö *Tekoälyn hyödyntäminen varastonohjauksessa* Jyväskylän ammattikorkeakoulussa 2021 on aiheeltaan lähellä omaa tutkimustyötäni. Lotvosen opinnäytetyön tavoitteena oli selvittää, voidaanko tekoälyä hyödyntää varastonohjauksessa ja kysynnän ennustamisessa. Myös Lotvonen hyödynsi tutkimustaan varten rakennettua koneoppimismallia, mutta mallin ohjelmoinnin ja yksityiskohtaisen tarkastelun sijasta tutkimus keskittyi algoritmin suorituskykyyn tietyn yrityksen tapauksessa.

3.3 Tutkimusmenetelmät

Ojasalo, Moilanen ja Ritalahti (2015) esittelevät kirjassaan viisi lähestymistapaa tutkimukselliseen kehittämistyöhön: tapaustutkimus, toimintatutkimus, konstruktiivinen tutkimus, palvelumuotoilu ja innovaatioiden tuottaminen. Kehittämistehtävä määrittää, mikä lähestymistapa sopii työhön parhaiten (Ojasalo et al, 2015, s. 36). Oma tutkimustyöni lienee lähimpänä konstruktiivista tutkimusta, jossa ”tavoitteena on käytännön ongelman ratkaisu luomalla uusi konstruktio eli jokin konkreettinen tuotos, esimerkiksi tuote, tietojärjestelmä, ohje tai käsikirja, malli, menetelmä tai suunnitelma” (Ojasalo et al, 2015, s. 37). Kuitenkin konstruktiivisen lähestymistavan keskeiseksi osaksi mainittu kehitetyn ratkaisun toteuttaminen ja käytännön toimivuuden arviointi jää tässä työssä tekemättä. Konstruktiivisen tutkimuksen kanssa samankaltaisia piirteitä on myös palvelumuotoilulla ja innovaatioiden tuottamisella ja kehittämishankkeessa voikin olla piirteitä monesta lähestymistavasta (Ojasalo et al, 2015, s. 36).

Kuvassa 9 olen mukaillut Ojasalon, Moilasen ja Ritalahden (2015) konstruktiivisen tutkimuksen prosessia sovitettuna opinnäytetyön eri vaiheisiin. Ensimmäisessä vaiheessa luon tietopohjan koneoppimisen mallin rakentamiselle. Tutkin ohjelmoinnin ja koneoppimisen perusteita sekä perustelen Python-ohjelmointikielen valintaa koneoppimisen mallin toteuttamiseksi. Tarkoitukseni on sekä avata käsitteitä opinnäytetyön lukijalle, että luoda perustaa tutkimuksessa käytettävän koneoppimisalgoritmin valinnalle. Lisäksi arvioin koneoppimisalgoritmien käytön mielekkyyttä tutkimusongelman ratkaisemiseksi ja niistä saatavia etuja verrattuna ehtoihin pohjautuviin algoritmeihin.

Kuva 9 - Konstruktiivisen tutkimuksen prosessi



Opinnäytetyön toisessa vaiheessa etenen tutkimuksen tavoitteeseen rakentamalla koneoppimisen mallin. Rakentamisen toteutan valitsemalla aikaisemman pohdinnan perusteella soveltuvimmaksi todetun koneoppimisalgoritmin tai algoritmit ja kirjoittamalla ohjelman, joka käyttää tehtävää varten luotua dataa. Ohjelman tuottamia tuloksia analysoimalla todennan algoritmin toimivuuden.

Kolmannessa vaiheessa pohdin tutkimustyön tuloksia ja arvioin laaditun mallin käyttökelpoisuutta. Tarkastelussa on kiinnitettävä huomiota synteettisen datan käyttöön tutkimuksessa: mitä hyötyä tai haittaa siitä voi olla?

4 Algoritmit ja koneoppiminen

4.1 Käsitteet

Kirjassa *Hello world: Kuinka selviytyä algoritmien aikakaudella (Fry, 2019, s. 20)* algoritmi on määritelty sarjana loogisia ohjeita, jotka kertovat alusta loppuun, miten jokin tehtävä on suoritettava. Vaikka mikä tahansa tämän määritelmän täyttävä ohje, esimerkiksi kirjahyllyn kokoamisopas, voisi teoriassa olla algoritmi, viitataan sanalla kuitenkin tavallisesti tietokoneohjelmien halutun tavoitteen saavuttamiseksi tekemiin laskutoimituksiin. Erilaiset algoritmit jaetaan neljään pääluokkaan (Fry, 2019, s. 21):

- Priorisointialgoritmit, jotka laativat käskyjen mukaisia luetteloita
- Luokittelualgoritmit, jotka jakavat datan kategorioihin
- Yhdistämisalgoritmit, jotka löytävät ja merkitsevät asioiden välisiä suhteita
- Suodattamisalgoritmit, jotka eristävät datasta oleellisia asioita

Yllä mainittu jako perustuu algoritmien toiminnan tavoitteisiin. Toimintatapansa perusteella algoritmit voidaan edelleen jakaa kahteen pääluokkaan: sääntöihin perustuvat algoritmit ja koneoppivat algoritmit. Näistä ensimmäiset toimivat suoraviivaisesti laatijansa luomien ohjeiden ja ehtojen mukaan. Jälkimmäiset taas kehittävät itse ohjeensa ja ehtonsa algoritmin laatijan määrittämällä tavalla (Fry, 2019, s. 24). Joni Kämäräinen käyttää sääntöihin perustuvista algoritmeista termiä ”Ohjelmointi 1.0” kirjassaan Koneoppimisen perusteet (2023, s.12).

Koneoppiminen on tekoälyn osa-alue. Tuomisen ja Neittaanmäen (2019, s. 6) mukaan sen tarkoituksena on ohjelmiston toimiminen paremmin paitsi pohjatiedon, myös käyttäjän toiminnan perusteella. Kone siis oppii toistuvista tapahtumista, sekä ihmisen tekemistä että

ohjelman toiminnan aikana tapahtuvista, ilman että sitä erikseen opetetaan. Ensimmäinen tunnettu koneoppimisen määritelmä lienee varhaisen tekoälyn kehittäjän Arthur Lee Samuelin julkaisema vuonna 1959: ”Koneoppiminen on oppiaine, joka antaa tietokoneille kyvyn oppia ilman, että niitä tarkasti ohjelmoidaan” (Géron, 2017, s. 4).

Tuominen ja Neittaanmäki (2019, s. 6) jakavat koneoppimisen tavat kolmeen eri kategoriaan: ohjattuun oppimiseen (*engl. supervised learning)*, ohjaamattomaan oppimiseen (*unsupervised learning)* ja vahvistettuun oppimiseen (*reinforcement learning)*. Géron (2017, s. 7) käyttää samaa jakoa lisäten joukkoon vielä puoliohjatun oppimisen (*semisupervised leraning)*. Hän esittelee myös seuraavia vaihtoehtoisia tapoja luokitella koneoppimisen järjestelmiä:

- Oppiiko järjestelmä lisää käsitellessään uutta dataa (*jatkuva oppiminen, engl. online learning)* vai toimiiko se alkuperäisen opetusdatan perusteella, kunnes se opetetaan uudestaan (*eräoppiminen, engl. batch learning)*?
- Toimiiko järjestelmä yksinkertaisesti vertaamalla uusia syötteitä tunnettuihin (*tapausoppiminen, engl. instance-based learning)* vai tunnistaako se säännönmukaisuuksia opetusdatassa luoden niihin perustuvan ennustavan mallin (*mallioppiminen, engl. model-based learning)*?

4.2 Ohjattu oppiminen

Ohjatun oppimisen periaate on, että koneoppimisjärjestelmälle annetaan opetusdatan muodossa tieto siitä, mikä on kuhunkin syötteeseen haluttu vaste. Ohjelma pystyy tämän jälkeen jaottelemaan saamansa uuden datan opetetulla tavalla. (Tuominen & Neittaanmäki, 2019, s. 13)

Ohjattu oppimisen algoritmit voidaan edelleen jakaa datan tyypin perusteella luokitteluun (*classification)* ja regressioon (*regression)*. Luokittelualgoritmi jakaa syötedatan tiettyihin ryhmiin. Regressioalgoritmin vaste taas on syötedatan perusteella luotu arvio esimerkiksi tuotteen hinnasta (Tuominen & Neittaanmäki, 2019, s. 13). Jako ei kuitenkaan välttämättä ole näin ehdoton, vaan regressioalgoritmia voidaan käyttää myös luokitteluun tai toisinpäin. Géron (2017, s. 9) mainitsee esimerkkinä logistisen regressioalgoritmin (*logistic regression)*, jonka tuottama vaste voi osoittaa esimerkiksi todennäköisyyttä, jolla syötedata kuuluu tiettyyn luokkaan.

4.3 Ohjaamaton oppiminen

Ohjaamattoman oppimisen ero ohjattuun on nimensä mukaisesti se, että opetusdata ei sisällä haluttuja vasteita, vaan järjestelmä oppii tunnistamalla datasta säännönmukaisuuksia ja poikkeamia (Tuominen & Neittaanmäki, 2019, s. 13). Géron (2017, s. 10–12) käsittelee aihetta kuvailemalla tyypillisimpiä ohjaamattoman oppimisen algoritmeja. Näitä ovat ryvästäminen (*clustering)*, visualisointi (*visualization)*, ulotteisuuden vähentäminen (*dimensionality reduction),* poikkeavuuksien havaitseminen (*anomaly detection)* ja riippuvuussääntöjen oppiminen (*association rule learning).*

Ryvästysalgoritmin vaste on samankaltainen kuin ohjatun luokittelualgoritmin. Algoritmi etsii datasta yhtymäkohtia ja luokittelee datan niiden perusteella. Visualisointialgoritmi taas tuottaa käyttäjän arvioitavaksi kuvan siitä, miten data on järjestäytynyt tiettyjen kriteerien mukaan helpottaen säännönmukaisuuksien ja yhtymäkohtien löytämistä. Ulotteisuuden vähentäminen liittyy visualisointiin. Sen tavoitteena on niputtaa datasta samankaltaiseen vasteeseen johtavia parametreja selkeämmän visualisoinnin tuottamiseksi. Géron (2017, s. 10–11)

Poikkeavuuksien havaitsemisalgoritmia hyödynnetään tyypillisesti esimerkiksi luottokorttien väärinkäytösten tai tuotteiden valmistusvirheiden havaitsemiseen. Algoritmi tunnistaa uuden datan, joka poikkeaa hyväksyttyjä tapauksia sisältävästä opetusdatasta. Géron (2017, s. 12)

Riippuvuussääntöjen oppimisen algoritmia hyödyntävät esimerkiksi tavaratalot tuotteidensa sijoittelussa. Algoritmi voi paljastaa riippuvuuksia kuluttajien ostotottumuksissa ja havaita esimerkiksi, että vaippojen ostajat ostavat myös paljon talouspaperia. Géron (2017, s. 12)

4.4 Puoliohjattu ja vahvistettu oppiminen

Puoliohjattu oppiminen on yleensä yhdistelmä ohjatun ja ohjaamattoman oppimisen algoritmeja. Puoliohjatun oppimisen järjestelmä voi toimia esimerkiksi siten, että se käyttää ohjaamatonta ryvästysalgoritmia datan järjestelyyn, jonka jälkeen käyttäjä osoittaa tuotetuille ryhmille halutut arvot eli vasteet. Tällöin kyse on ohjatusta luokittelusta. Géron (2017, s. 13)

Vahvistettu oppiminen toimii niin, että oppiva järjestelmä tekee opetusdataan pohjautuvia ratkaisuja, joista se saa käyttäjältä palkintoja tai rangaistuksia. Näiden perusteella järjestelmä pyrkii muodostamaan optimaalisen, siis eniten palkintoja tuottavan toimintamallin kuhunkin tilanteeseen. Menetelmää käytetään usein robottien opettamiseen. Géron (2017, s. 13–14)

4.5 Eräoppiminen ja jatkuva oppiminen

Eräoppiminen ja jatkuva oppiminen eivät ole vaihtoehtoinen menetelmä aiemmin luetelluille vaan erilainen tapa luokitella koneoppimisalgoritmeja.

Kun eräoppivan algoritmin toimintaa halutaan muuttaa, on sen toiminta keskeytettävä ja koulutusvaihe ajettava uudestaan, jonka jälkeen järjestelmän käyttöä voidaan jatkaa. Tämän menetelmän käyttökelpoisuus riippuu opetusdatan määrästä ja järjestelmän käyttötarkoituksesta. Géron (2017, s. 15)

Jatkuvan oppimisen järjestelmä koulutetaan syöttämällä sille opetusdataa joko tietue kerrallaan tai pienissä erissä. Toimintatapaa voidaan hyödyntää jatkuvasti toiminnassa olevissa järjestelmissä, jotka näin pystyvät reagoimaan nopeasti muutoksiin saamansa uuden datan kautta. Toinen käyttötarkoitus voi olla tallennustilan tai laskentatehon säästäminen: järjestelmä voi hylätä vanhan datan uuden saapuessa. Géron (2017, s. 16)

Käytettäessä jatkuvan oppimisen järjestelmää on huomioitava, että virheellisen datan syöttäminen algoritmille huonontaa järjestelmän toiminnan laatua. Erityisesti nopeasti uuteen dataan reagoivat järjestelmät tarvitsevat näin ollen enemmän valvontaa, jotta virheisiin voidaan puuttua. Valvontaa voi suorittaa ihminen tai erillinen poikkeavuuksien havaitsemisen algoritmi. Géron (2017, s. 16–17)

4.6 Tapausoppiminen ja mallioppiminen

Viimeinen Géronin (2017) mainitsema koneoppimisalgoritmien luokittelumenetelmä on jako tapaus- tai mallipohjaiseen oppimiseen. Tässä jaossa on kyse siitä, miten algoritmit yleistävät dataa.

Esimerkkinä tapausoppivasta järjestelmästä voisi olla ohjattu luokittelu- tai ohjaamaton ryvästysalgoritmi. Järjestelmä on jakanut opetusdatan ryhmiin ja yleistää datan ominaisuuksia uuden datan jakamiseksi opetusdatan kaltaisesti. Géron (2017, s. 17)

Mallioppiva järjestelmä koulutetaan valitsemalla opetusdataan parhaiten sopiva malli eli funktion kuvaaja. Luvussa 5.1.2 mainittu regressioalgoritmi on esimerkki tällä tavoin toimivasta järjestelmästä. Mallioppiva järjestelmä siis yleistää uutta dataa sen perusteella, miten se sopii valitun funktion kuvaajalle. Géron (2017, s. 18–21)

5 Koneoppimisen käytön mielekkyys tutkimusongelman ratkaisussa

Joni Kämäräinen vertaa kirjassaan Koneoppimisen perusteet (2023) perinteisiä algoritmeja ja koneoppimista. Vertailussaan hän käyttää esimerkkinä algoritmia, joka järjestää syötteenä annetut sanat tai numerot aakkos- tai suuruusjärjestykseen sekä algoritmia, joka löytää kaistaviivan itsestään ajavan auton etukameran kuvasta (Kämäräinen, 2023, s.13-14). Näistä ensimmäinen, järjestäminen, on ”suljetun maailman ongelma”, jolle voidaan kirjoittaa täsmällinen oikean tulosteen antava algoritmi. Kaistaviivan tunnistaminen kameran kuvasta taas on ”avoimen maailman ongelma”: aukottoman algoritmin ja siihen kuuluvien ehtojen kirjoittaminen on mahdotonta ilman merkittäviä pelkistyksiä. Liika pelkistäminen taas johtaa siihen, ettei lopputulos toimi tosielämässä (Kämäräinen, 2023, s.15). Koneoppiminen perustuu kerätyn opetusaineiston käyttöön, jonka vuoksi perinteisiä algoritmeja on mielekkäämpää käyttää tapauksissa, joissa ongelmaan on ohjelmoitavissa täsmällinen algoritmi. Koneoppimisratkaisun käytöstä tulisi siis olla jotain etua (Kämäräinen, 2023, s.15).

Tutkimusongelman määrittämisen yhteydessä nimesin opinnäytetyöni tavoitteeksi ”luoda koneoppimisen malli, joka ennustaa erityyppisten asiakkaiden tilauksia aiempien tilausten perusteella ja sitä kautta auttaa ylläpitämään oikean määrän oikeita tuotteita sopivissa varastoissa”. Kirjoitin siis tavoitteen jo etukäteen koneoppimisen käyttöä silmällä pitäen. Entä jos tavoite olisi koneoppimisen mallin luomisen sijaan “luoda algoritmi”? Koneoppimisen sijaan voisi olla mahdollista käyttää aikasarjaennustamista, joka perustuu joukkoon tietueita järjestettynä ajanhetken mukaan. Lineaarisen regressioalgoritmin tavoin kyse on trendin ja hajonnan havaitsemisesta aiemmasta datasta (Peixeiro, 2022).
Aikasarjaennustamisalgoritmien käyttö tutkimusongelman ratkaisemisessa olisi varmasti mahdollista esimerkiksi luomalla nykyisen asiakasnumeron sijaan tunnus yhdistämällä asiakkaan tyypistä kertovat attribuutit (kuva 4) asiakasnumeroksi ja tutkimalla näin syntyneille asiakasnumeroille tehtyjä tilauksia kunkin nimikkeen osalta aikajanalla.

Jos tutkimukseni kohteena olisi tosielämän data ja lopputulosta käytettäisiin jonkun tietyn yrityksen liiketoiminnan parantamiseen, olisi mielekästä tutkia tarkemmin eri algoritmien käytön tehokkuutta. Lopullisen lähestymistavan valinnassa tulisi huomioida ainakin ohjelman laatimisen monimutkaisuus ja sen myötä kustannukset, tarvittavan opetusdatan eli tarvittavan tallennustilan määrä, laskennan nopeus, laajennettavuus muuhun käyttötarkoitukseen ja tietysti eri algoritmien tuottamien ennustusten paikkansa pitävyys.

6 Työkalut

6.1 Python-ohjelmointikieli

Python on ohjelmointikieli, jonka kehittäminen alkoi vuonna 1989 ja ensimmäinen julkaisu tehtiin helmikuussa 1991 *(Van Rossum, 2009).* Nykyään Python on yksi maailman suosituimmista ohjelmointikielistä. Tämä perustunee ohjelmien helppoon kirjoitusasuun, toimivuuteen lähes kaikilla alustoilla, erinomaiseen dokumentaatioon ja kattavaan vakiokirjastoon, joka sisältää valmiita funktioita omissa ohjelmissa käytettäväksi (*Vorderman et al., 2017).*

IDLE (Integrated Development and Learning Environment) on Pythonin mukana asentuva ilmainen sovellus. Käytännössä se on yksinkertainen komentorivi Python-ohjelmien kirjoittamiseen ja ajamiseen (kuva 10). (*Vorderman et al., 2017).* Tehokkaampaa käyttöä varten on luotu useita kolmannen osapuolen IDE-kehitysympäristöjä (Integrated Development Environment). Näissä ympäristöissä on yleensä jonkinlainen kyky virheiden etsintään sekä kokonaisten ohjelmistoprojektien versionhallintaan *(Parker, 2021)*.

Kuva 10 - Python IDLE

```
Python 3.9.17 (main, Jun 15 2023, 07:46:17)
[Clang 14.0.3 (clang-1403.0.22.14.1)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Pythonin mittavan vakiokirjaston lisäksi sille on luotu ulkopuolisten kehittäjien toimesta runsaasti käyttökelpoisia kirjastoja muun muassa koneoppimiseen. Esimerkkinä näistä mainittakoon SciKit-Learn ja TensorFlow, joista ensimmäistä käytän myös tämän tutkimuksen koneoppimismallin rakentamiseen. Syynä kirjaston valintaan on sen funktioiden perusteellinen käsittely Gèronin kirjassa Hands-On Machine Learning with Scikit-Learn and TensorFlow (2017).

6.2 iPython ja Jupyter

Avoimen lähdekoodin iPython-alusta tukee Python-ohjelmointia tarjoamalla interaktiivisen komentorivin sekä selainpohjaisen Notebook-editorin *(Rossant, 2014)*. Erona edellisessä luvussa mainittuihin Pythonin IDLE- ja IDE-ympäristöihin iPythonin Notebook yhdistää

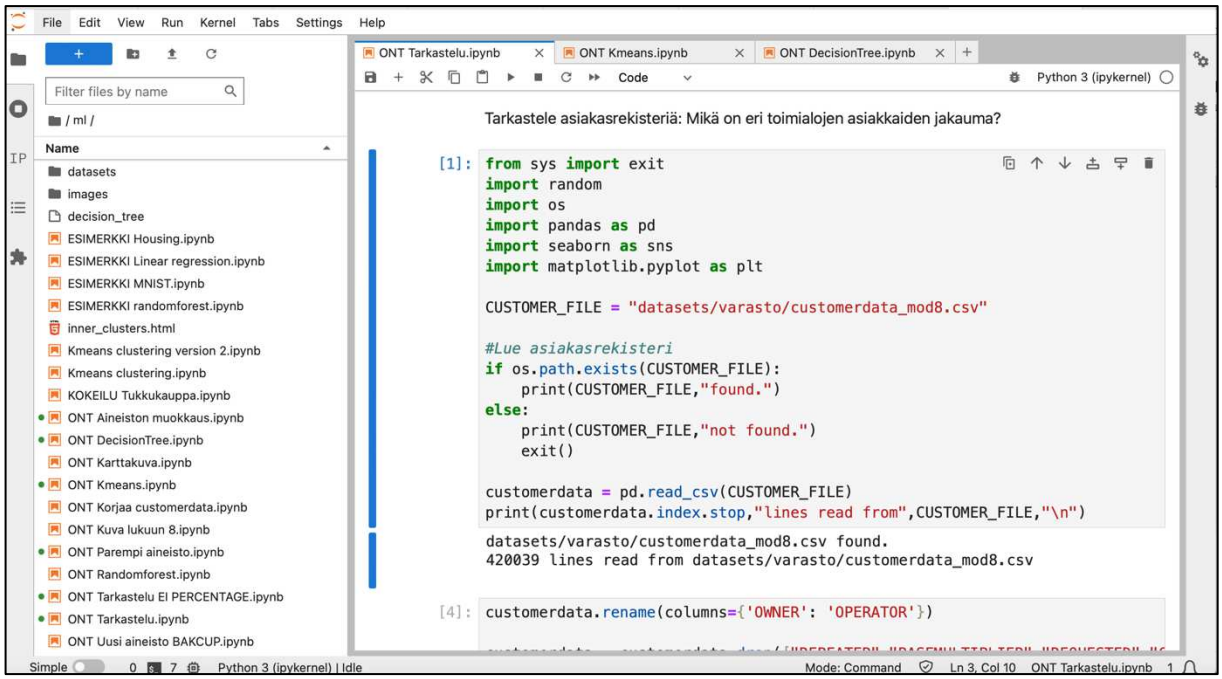
koodin, tekstin, matemaattisen ilmaisun, graafiset kuvaajat, animaation ja videot *(Rossant, 2014).*

iPython Notebook kehitettiin 1990- ja 2000-luvuilla rinnan tiettyjen Pythonin datatieteitä tukevien kirjastojen kanssa. Tällaisia kirjastoja ovat numeerista dataa käsittelevät *NumPy* ja *SciPy*, kuvaajia piirtäviä funktioita sisältävä *matplotlib* sekä *pandas*, jonka avulla ohjelmoija voi tehokkaasti analysoida ja muokata taulukoita, sarjoja ja tietokantoja. *(Rossant, 2014)*

Jupyter on avoimen lähdekoodin projekti, jonka tarkoitus on tukea datatiedettä ja ohjelmointikielestä riippumatonta tieteellistä laskentaa *(Kluyver, T. et al., 2016)*. Jupyter-projekti irtaantui iPython-projektista vuonna 2014 tarkoituksena keskittyä iPythonin ohjelmointikielestä riippumattomiin osiin *(Rossant, 2014)*. Jupyter-projekti tukee kymmeniä ohjelmointikieliä ja avoimen lähdekoodin vuoksi kuka tahansa voi kasvattaa listaa *(Kluyver, T. et al., 2016).* iPython toimii edelleen Jupyterin alkuperäisenä kääntäjänä Python-kielelle *(Rossant, 2014)*.

Jupyter Notebook yhdistää tekstiä, koodia ja mediaa kuten iPython Notebook. JupyterLab (kuva 11) taas on projektin viimeisin ja kehittynein ympäristö, joka tarjoaa modulaarisen alustan datatiede-, laskenta- ja koneoppimisprojektien kokonaisvaltaiseen hallintaan *(Kluyver, T. et al., 2016)*.

Kuva 11 - JupyterLab



7 Tulokset

7.1 Tuotteiden jakamiseen vaikuttavat asiakkaan tiedot

Aurélien Géronin kirja Hands-On Machine Learning with Scikit-Learn & Tensorflow (2017) tarjoaa erinomaisen muistilistan koneoppimisprojektin läpi viemiseksi:

1. Tarkastele ongelmaa
2. Hanki data
3. Tutki dataa oivallusten saamiseksi
4. Valmistele data paljastaaksesi säännönmukaisuudet
5. Tutki erilaisia koneoppimismalleja ja valitse parhaat
6. Viritä mallejasi ja yhdistä niitä parhaaseen ratkaisuun pääsemiseksi
7. Esittele ratkaisusi
8. Käynnistä, tarkkaile ja ylläpidä järjestelmää

Koska ongelman tarkastelu ja käytössä olevan datan esittely on tehty jo tutkimustyön kolmannessa luvussa, aloitan tutkimalla aikaisempien manuaalisesti käsiteltyjen tilausten tietokantaa, joka on esitetty kuvassa 6 (s. 8).

Géron (2017) ohjaa tekemään tutkittavasta datasta kopion ja tarvittaessa ottamaan siitä otoksen helpompaa käsittelyä varten. Tämän jälkeen tulisi tarkastella tietokannan kunkin sarakkeen sisältämien tietojen ominaisuuksia. Näitä ominaisuuksia ovat nimi, tyyppi, mahdollisten puuttuvien arvojen osuus, kohina ja virheet, käytettävyys ongelman ratkaisussa ja arvojen noudattama jakauma.

Kuvan 12 mukaisesti 420039 tietueen tietotaulu sisältää 12 saraketta, joista seitsemän on kokonaislukuja ja loput viisi tyyppiä ”object”. Jälkimmäisellä tarkoitetaan pandas-tietokannassa dataa, joka voi sisältää esimerkiksi merkkijonoja, listoja tai muita monimutkaisempia rakenteita.

Kuva 12 - Datan ominaisuudet

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 420039 entries, 0 to 420038
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Customer_Num      420039 non-null  int64
 1   Customer_Type     420039 non-null  object
 2   Customer_Subtype  420039 non-null  int64
 3   Customer_Class    420039 non-null  object
 4   Operator          420039 non-null  int64
 5   Delivery_Point    420039 non-null  int64
 6   Priority          420039 non-null  int64
 7   Class_ID          420039 non-null  object
 8   Class_Name        420039 non-null  object
 9   Item_ID           420039 non-null  int64
 10  Item_Name         420039 non-null  object
 11  Provided          420039 non-null  int64
dtypes: int64(7), object(5)
memory usage: 38.5+ MB
```

Asiakasnumero (Customer_Num) on jokaiselle asiakkaalle annettu satunnainen kahdeksannumeroinen luku. Uniikkeja asiakasnumeroita on 4000, kuten luvussa 2.2 totesin. Jokaisen operaattorin alue (Operator) sisältää useita toimituspisteitä ja jokainen asiakas käyttää vain yhtä toimituspistettä (Delivery_Point). Asiakkaan prioriteetti (Priority) ja luokka (Customer_Class) ovat asiakaskohtaisia parametrejä, jotka eivät muutu. Useat eri asiakkaat ovat samaa luokkaa. Asiakkaan luokka on yhdistelmä sarakkeiden ”Customer_Type” ja ”Customer_Subtype” tiedoista.

Tilauspohja sisältää vaihtelevan määrän tuotekategorioita ja lukumääriä. Erot eri asiakkaiden välillä syntyvät siten, että tilausten tuotekategoriat täytetään eri nimikkeiden tuotteilla. Esimerkiksi asiakkaat numero 592400 ja 731081 ovat molemmat tyyppiä J1 (huoltoasema), mutta heille on toimitettu eri määrä erilaisia nimikkeitä (kuva 13). Toimitettu määrä esitetään sarakkeessa ”Provided”.

Kuva 13 - Erot asiakkaiden välillä

	Customer_Num	Customer_Type	Customer_Subtype	Customer_Class	Operator	Delivery_Point	Priority	Class_ID	Class_Name	Item_ID	Item_Name	Provided
0	731081	J	1	J1	1000	19	4	F-AH-001	AUTOSHAMPOO	67042927	SONAX AUTOSHAMPOO	209
1	731081	J	1	J1	1000	19	4	F-AH-001	AUTOSHAMPOO	58908704	TURTLE WAX SUPER HARD SHELL AUTOSHAMPOO	531
2	731081	J	1	J1	1000	19	4	F-AH-002	AUTOVAHAT	76906724	MEGUIAR'S GOLD CLASS CARNAUBA PLUS PREMIUM AUT...	117
3	731081	J	1	J1	1000	19	4	F-AH-002	AUTOVAHAT	73059125	COLLINITE 845 INSULATOR WAX	95
4	731081	J	1	J1	1000	19	4	F-AH-003	SISÄPUHDISTUS	18755318	ARMOR ALL LEATHER CARE GEL	124

	Customer_Num	Customer_Type	Customer_Subtype	Customer_Class	Operator	Delivery_Point	Priority	Class_ID	Class_Name	Item_ID	Item_Name	Provided
363411	592400	J	1	J1	2000	30	3	F-AH-001	AUTOSHAMPOO	67042927	SONAX AUTOSHAMPOO	298
363412	592400	J	1	J1	2000	30	3	F-AH-001	AUTOSHAMPOO	58908704	TURTLE WAX SUPER HARD SHELL AUTOSHAMPOO	700
363413	592400	J	1	J1	2000	30	3	F-AH-002	AUTOVAHAT	76906724	MEGUIAR'S GOLD CLASS CARNAUBA PLUS PREMIUM AUT...	166
363414	592400	J	1	J1	2000	30	3	F-AH-002	AUTOVAHAT	73059125	COLLINITE 845 INSULATOR WAX	136
363415	592400	J	1	J1	2000	30	3	F-AH-003	SISÄPUHDISTUS	18755318	ARMOR ALL LEATHER CARE GEL	177

Asiakkaan tilaus siis sisältää tietyn määrän erilaisia tuoteluokkia ja -nimikkeitä. Aineistossa esiintyisi todennäköisesti tunnistettavia säännönmukaisuuksia, joiden tunnistaminen nopeuttaisi tilausten käsittelyä ja helpottaisi varastosaldojen ylläpitoa oikealla tasolla silloinkin, kun asiakaskunnassa tapahtuu muutoksia. Laadittavan mallin tavoitteena on siis oppia millä perusteilla asiakas- ja nimikekohtaiset ”Provided”-sarakkeen arvot on määritetty. Aineistosta on kyettävä tunnistamaan ne asiakasparametrit, joilla on vaikutusta toimitettujen nimikkeiden määrään, esimerkiksi: ”Prioriteetin kolme huoltoasema Keski-Suomessa tilaa Turtle Wax Super Hard Shell shampoota 1000kpl kuukaudessa”.

Nimikekohtainen ”Provided” on arvo, joka koneoppimisalgoritmin tulee kyetä ennustamaan asiakkaan tietojen perusteella. Koska aineistossa on jokaiselle tapaukselle esimerkki kohdemuuttujan arvosta, voidaan todeta kyseessä olevan ohjatun oppimisen tehtävä. Edelleen voidaan päätellä, että kyse on monimuuttujaregressiosta, sillä tarkoitus on ennustaa numeerista arvoa usean eri muuttujan perusteella. (Géron, 2017, s. 37)

Géron suosittelee erottamaan datasta opetus- ja testiaineistot ennen syvempää tutkiskelua. Perusteluksi annetaan ihmisaivojen kyky havaita säännönmukaisuuksia: testiaineiston tutkiminen voisi johtaa tietyn koneoppimisalgoritmin valintaan väärillä perusteilla. Tyypillisesti opetusaineiston koko on 80 % koko aineistosta, jolloin testaamiselle jää 20 %. (Géron, 2017, s. 49)

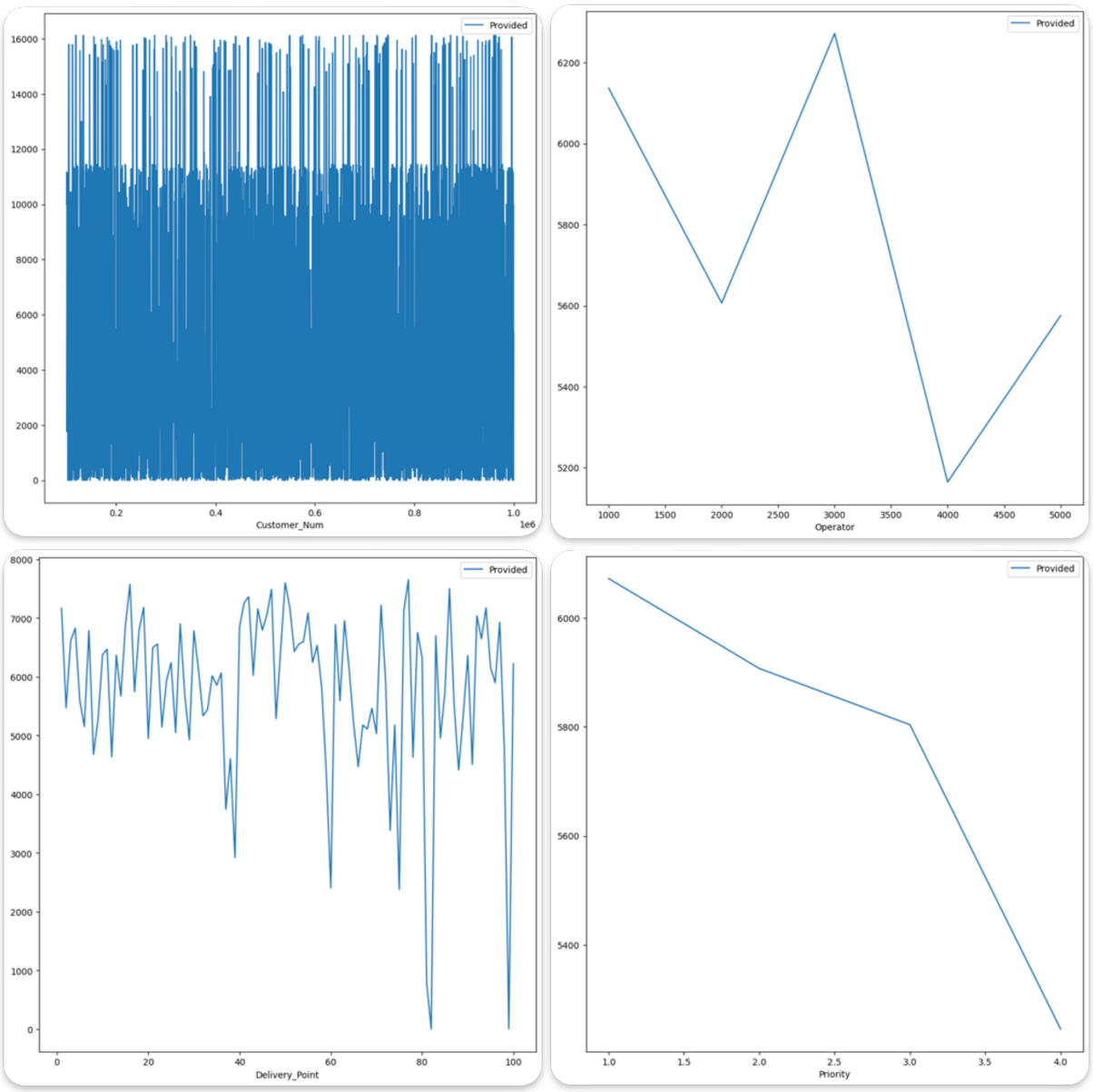
Satunnaista opetusaineistoa erotettaessa tulee varmistua, ettei aineistoa luoda eri perusteilla aina ohjelmaa ajettaessa. Tällöin testiaineistoa saatettaisiin ennen pitkää käyttää

koneoppimisalgoritmin opettamiseen. Lisäksi tulee huolehtia siitä, että opetusdata edustaa kattavasti koko aineistoa (Géron, 2017, s. 49–51). Tässä tapauksessa koko aineiston edustaminen tarkoittaa, että opetusaineistossa on oltava rivejä jokaiselta ”Item_ID”:n arvolta.

Tarkastelen opetusdatasta, paljonko kaikkia nimikkeitä keskimäärin lähetetään erityyppisille asiakkaille.

Kuvan 14 luomista varten olen poiminut tilaustietokannasta vain sarakkeet ”Customer_Num”, ”Customer_Subtype”, ”Operator”, ”Delivery_Point”, ”Priority”, ”Item_ID” ja ”Provided”. Kuvassa näkyy kunkin sarakeotsikon yllä kuvaaja, jossa x-akseli osoittaa sarakkeen arvon ja y-akseli ”Provided”-sarakkeen arvojen mediaanin.

Kuva 14 - Tuotteen jakautuminen tilaajille



Toimitetun määrän suhteessa asiakasnumeroon ja toimituspisteen numeroon kuvaajat näyttävät odotetusti melko satunnaisilta eikä niistä ole suoraan luettavissa säännönmukaisuuksia. Sen sijaan ”Operator”-sarakkeen tarkastelu antaa viitteitä siitä, että sarakkeen arvolla 4000 oleville asiakkaille tuotteita lähetetään huomattavasti vähemmän kuin muilla ”Operator”-arvoilla. ”Priority”-sarakkeen arvot taas näyttävät noudattavan kaavaa, jossa ”Provided”-arvo on kääntäen verrannollinen ”Priority”-arvoon.

Esimerkissä tarkastelin toimituspisteen suhdetta toimitettuun määrään toimituspisteen numeron perusteella. Luvun 2.2 kuvasta 3 kuitenkin selviää, että toimituspisteellä on myös paikkatieto. Koska toimituspisteen numero ei suoraan ilmaise maantieteellistä sijaintia, on sen sijaan tarkoituksenmukaista tarkastella numeroon liitettyjä koordinaatteja.

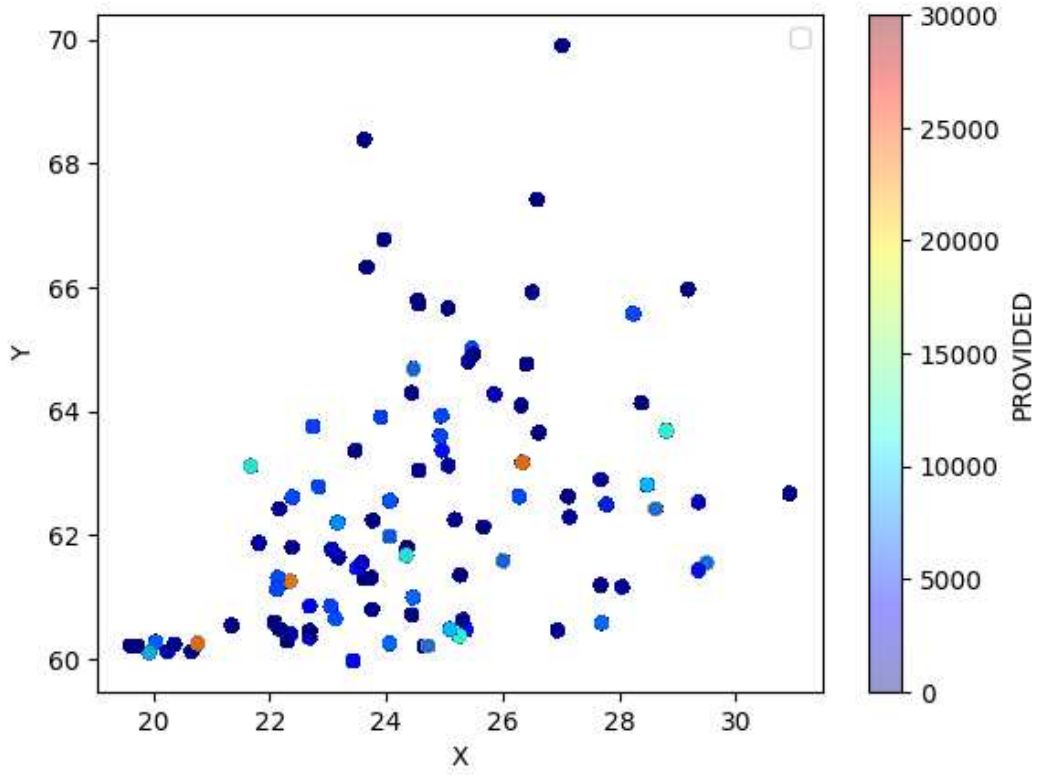
Kuvassa 15 tietotauluun on lisätty sarakkeet ”X” ja ”Y”, jotka sisältävät ”Delivery_Point”-sarakkeessa esitetyn toimituspisteen sijaintitiedot.

Kuva 15 - Lisätyt paikkatiedot

	Customer_Num	Customer_Subtype	Customer_Class	Operator	Delivery_Point	Priority	Item_ID	Provided	X	Y
0	731081	1	J1	1000	19	4	67042927	209	24.47	64.68
1	731081	1	J1	1000	19	4	58908704	531	24.47	64.68
2	731081	1	J1	1000	19	4	76906724	117	24.47	64.68
3	731081	1	J1	1000	19	4	73059125	95	24.47	64.68
4	731081	1	J1	1000	19	4	18755318	124	24.47	64.68

Kuva 16 esittää ”Provided”-sarakkeen arvojen mediaanin suhteessa toimituspisteiden koordinaatteihin. Kartan perusteella sijainnilla ei näytä olevan juurikaan merkitystä toimitettuun määrään, mutta tulosta todennäköisesti vääristää se, että eri nimikkeiden määrien vaihteluvälit poikkeavat toisistaan huomattavasti.

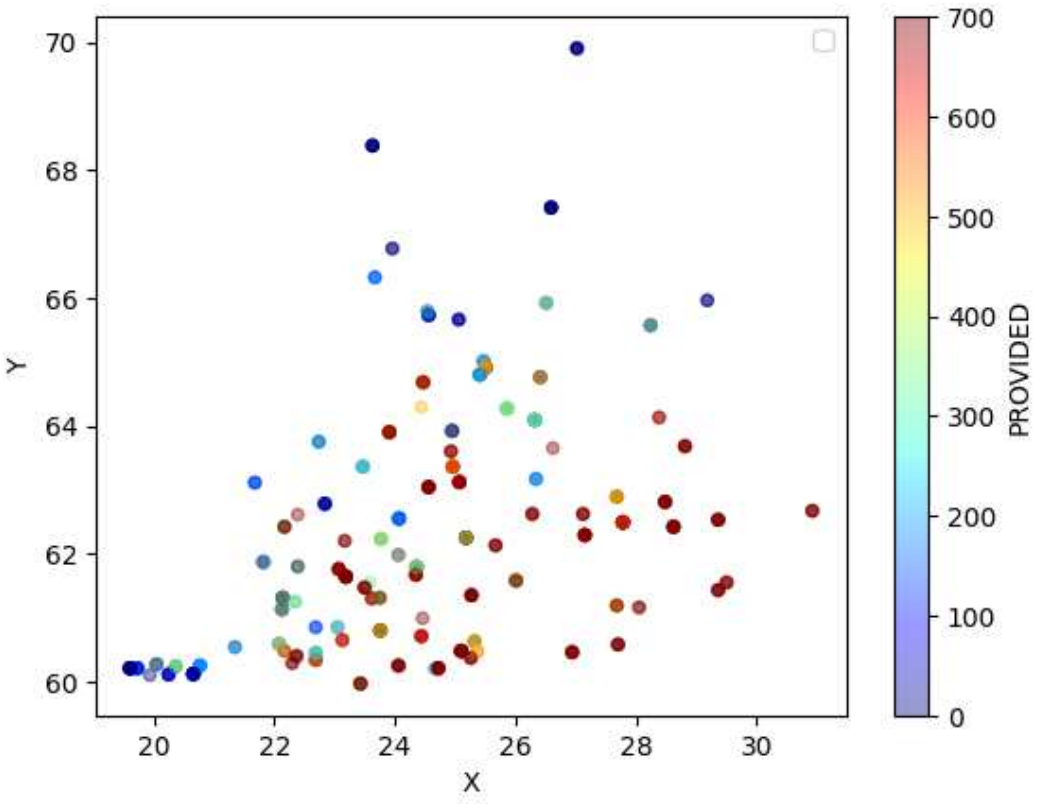
Kuva 16 - Sijainnin vaikutus



Kuva 17 esittää yksittäisen nimikkeen (58908704, TURTLE WAX SUPER HARD SHELL AUTOSHAMPOO) tilausmäärien mediaanin suhteessa toimituspisteiden koordinaatteihin.

Kuvasta on poikkeamista huolimatta havaittavissa korrelaatio toimituspisteen sijainnin ja lähetetyn määrän välillä siten, että kaakkoon mentäessä toimitettu määrä kasvaa.

Kuva 17 - Sijainnin vaikutus yksittäiseen nimikkeeseen



Kuvan 18 korrelaatiomatriisi puoltaa yllä esitettyjä johtopäätöksiä. Matriisissa on verrattu jokaisen taulukon muuttujan suhdetta ”Provided”-arvoon. Lähellä numeroa 1.0 oleva arvo kertoo vahvasta positiivisesta korrelaatiosta. Vastaavasti mitä lähempänä arvo on -1.0, sitä voimakkaammasta negatiivisesta korrelaatiosta on kyse. Matriisin perusteella merkittävin lineaarinen vaikutus on asiakkaan toimituspisteen sijainnilla (”X”, ”Y”) sekä prioriteetilla (”Priority”).

Kuva 18 - Korrelaatiotaulukko

```
Provided            1.000000
Customer_Subtype    0.508433
X                   0.039260
Item_ID             0.025786
Customer_Num        0.001431
Operator           -0.030026
Delivery_Point     -0.038313
Priority           -0.040167
Y                  -0.078311
Name: Provided, dtype: float64
```

Kuten kuvan 12 (Datan ominaisuudet) tarkastelun yhteydessä totesin, tilauspohjan tunniste ”Customer_Class” on yhdistelmä muuttujista ”Customer_Type” ja ”Customer_Subtype”. Jätin tähän saakka huomioimatta tämän muuttujan vaikutuksen ”Provided”-arvoon. Sillä kuitenkin on todennäköisesti merkitystä. Tarkastelen ensin, onko samoja nimikkeitä tilattu erityyppisille asiakkaille.

Kuvan 19 perusteella samaa nimikettä voi olla tilannut yhdestä kymmeneen erilaista asiakastyyppiä.

Kuva 19 - Eri asiakastyyppien lukumäärä per nimike

```
[38]: result = merged_df.groupby('Item_ID')['Customer_Class'].nunique().reset_index(name='Class_Count')

print(result)
```

```
      Item_ID  Class_Count
0    10019096            1
1    10116058            1
2    10691615            1
3    10694620            2
4    10834682            5
..        ...          ...
699  99357160            3
700  99468631            3
701  99795440            1
702  99811126            3
703  99915852            1

[704 rows x 2 columns]
```

```
[39]: max_value = result['Class_Count'].max()
min_value = result['Class_Count'].min()

print("Maximum Class_Count:", max_value)
print("Minimum Class_Count:", min_value)
```

```
Maximum Class_Count: 10
Minimum Class_Count: 1
```

```
[40]: max_ordersheet_count_rows = result[result['Class_Count'] == max_value]

item_ids_with_max_ordersheet_count = max_ordersheet_count_rows['Item_ID'].tolist()

print("Item_IDs with the maximum Class_Count:", item_ids_with_max_ordersheet_count)
```

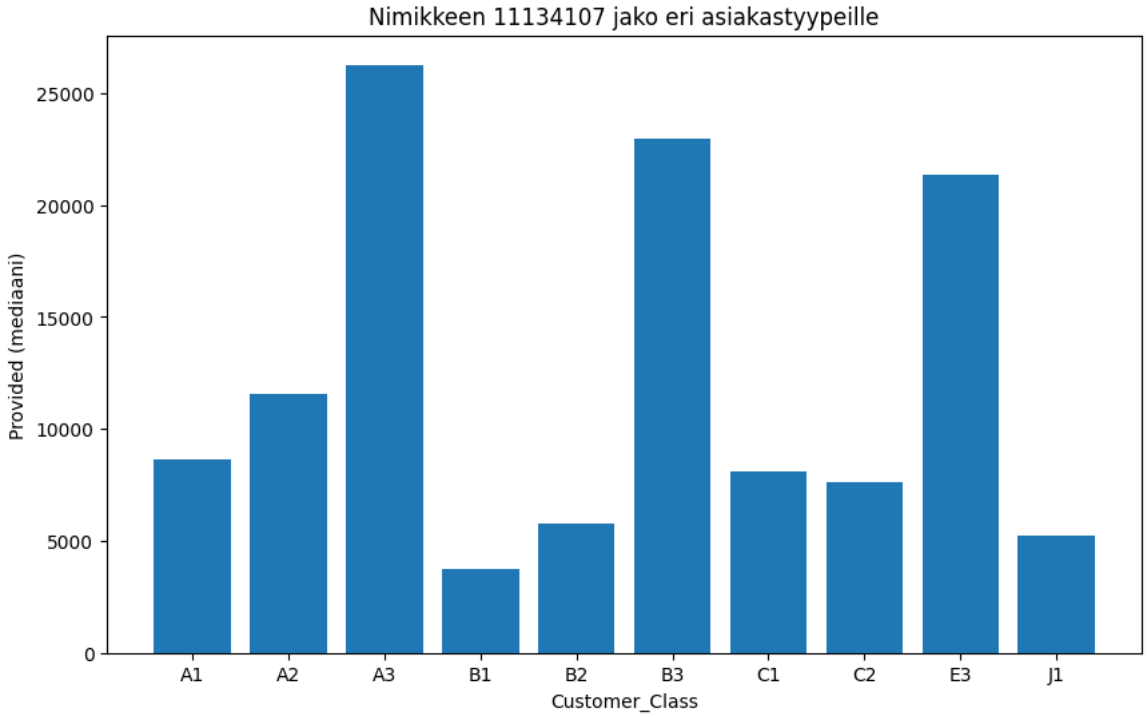
```
Item_IDs with the maximum Class_Count: [11134107, 11186587, 11666528, 11845047, 14731796, 21246110, 26683186, 29569057, 3101008
6, 33958130, 39403923, 43811323, 44359537, 47096577, 53761729, 54620935, 56577711, 58093969, 64594362, 65059887, 67790618, 6862
3437, 70022021, 70539600, 73160948, 73581836, 76565331, 77928503, 79264271, 80013614, 81301699, 81774563, 82369188, 87933108, 8
9032974, 91209504, 92086508, 97760116, 98156539, 98593785]
```

Nimikkeitä, joilla on maksimimäärä asiakastyyppejä, on 40. Näistä esimerkiksi nimikettä 11134107 JÄÄSALAATTI tilaavat seuraavan tyyppiset asiakkaat:

A1	LÄHIKAUPPA
A2	SUPERMARKET
A3	HYPERMARKET
B1	PIKARUOKA
B2	PERHERAVINTOLA
B3	FINE-DINING
C1	MOTELLI
C2	HOTELLI
E3	SAIRAALA
J1	HUOLTOASEMA

Kuvan 20 pylväsdiagrammi osoittaa, että nimikkeen 11134107 lähetykset eri asiakastyypeille vaihtelevat merkittävästi. Asiakkaan tyypillä on siis merkitystä ”Provided”-arvoa ennustettaessa.

Kuva 20 - Nimikkeen jako eri asiakastyypeille



Aineiston tarkastelun perusteella muuttujat, joista on hyötyä valmisteltaessa dataa koneoppimisalgoritmia varten, ovat ”Customer_Class”, Operator, X, Y ja Priority sekä tietenkin nimikkeen yksilöivä ”Item_ID”. Näistä ”Customer_Class” on kuitenkin ensin muutettava numeraaliseen muotoon. Lisäksi muuttujia on muokattava siten, että niiden mittakaava on sama. (Géron, 2017, s. 62–65)

”Customer_Class”-arvon muuttamiseksi numeraaliseen muotoon käytän Pythonin Scikit-Learn-kirjaston valmiita funktioita, joilla voi paitsi antaa kaikille mahdollisille ”Customer_Class”-sarakkeen arvoille tunnistenumeron, myös edelleen muokata numerolistauksesta matriisin, jonka riveillä tunnistenumeron tallentamisen sijaan osoitetaan sarakkeen arvoa vastaavan numeron sijainti numerolla 1 muiden rivin arvojen näyttäessä nollaa. Matriisi on tehtävä siksi, että muuten koneoppimisalgoritmi tulkitsisi esimerkiksi vierekkäisen numeroarvon 1 ja 2 saaneet asiakastyypit saman kaltaisiksi, vaikka niillä ei olisi mitään tekemistä keskenään. (Géron, 2017, s. 62–64)

Aineiston numeraalisten muuttujien ”Item_ID”, ”Operator”, ”X”, ”Y” ja ”Priority” mittakaava on keskenään hyvin erilainen. ”Operator”-muuttujan arvot ovat välillä 1000–5000 (kuva 14), mutta se on numeromuodostaan huolimatta luokitteleva kuten ”Customer_Class”. Sama koskee ”Item_ID”-muuttujaa, jonka saamat arvot ovat välillä 10019096–99915852.

Molemmat muuttujat on muutettava matriisimuotoon, kuten ”Customer_Class”. Priority-muuttujan arvot ovat välillä 1–4. Se on myös luokitteleva, mutta vaikutus ”Provided”-arvoon näyttää olevan lineaarinen (kuva 14). ”X” ja ”Y” ovat keskenään samankaltaisia liukulukuja ”X”:n arvojen sijoittuessa välille 19.61–30.92 ja ”Y”:n arvojen välille 59.97–69.9 (kuva 17).

Pythonin Scikit-Learn-kirjasto sisältää kaksi valmista lähestymistapaa eri mittakaavan muuttujien skaalaamiseen: arvot voidaan skaalata niin, että ne kaikki sijoittuvat liukulukuina välille 0 ja 1, tai ”standardisoida” siten, että mediaani on aina nolla. Jälkimmäinen vaihtoehto ei rajoita lukuja millekään tietylle välille, minkä vuoksi jotkut algoritmit saattavat ymmärtää sitä huonosti. Se voi kuitenkin olla parempi vaihtoehto aineistossa, jossa esimerkiksi virheestä johtuvien yksittäisten poikkeamien ei haluta vaikuttavan niin voimakkaasti lopputulokseen (Géron, 2017, s. 65). Tätä opinnäytetyötä varten luomassani synteettisessä aineistossa ei poikkeamia ole, joten kokeilen ensin mainittua tekniikkaa eli skaalaamista arvojen 0 ja 1 välille.

Laadin putken (pipeline), joka muuttaa sille annetun tietotaulun merkkijonomuotoiset muuttujat matriisimuotoon ja skaalaa numeeriset muuttujat 0 ja 1 välille. Tämän jälkeen koulutan Scikit-Learn-kirjaston lineaarisen regressioalgoritmin opetusaineiston tiedoilla. Lineaarinen regressio luo ennusteen laskemalla annettujen muuttujien painotetun summan lisättynä vinoumatermillä (bias term). Regressiomallin kouluttaminen tarkoittaa mallin parametrien sovittamista niin, että sen kuvaaja sopii parhaiten yhteen opetusaineiston kuvaajan kanssa (Géron, 2017, s. 107). Scikit-Learn-kirjasto tarjoaa valmiin LinearRegression-funktion mallin kouluttamiseen. Riittää, että opetusdata on ajettu yllä mainitun putken läpi ja siten muokattu funktion ymmärtämään muotoon.

Kuva 21 esittää lineaarisen regressiomallin kouluttamisen ja mallin tuottaman keskimääräisen virheen. Keskimääräinen virhe (Root Mean Square Error, RMSE) todennetaan laskemalla kullekin datapisteelle ennustetun arvon ja todellisen eli opetusdatasta tunnetun arvon ero. Ero korotetaan neliöön negatiivisten lukujen merkityksen poistamiseksi ja lasketaan erojen keskiarvo. Lopuksi keskiarvosta otetaan neliöjuuri. (Géron, 2017, s. 37)

Kuva 21 - Lineaarinen regressio ja virheen arviointi

```
[15]: #Linear regression

      from sklearn.linear_model import LinearRegression
      from sklearn.metrics import mean_squared_error
      import numpy as np

      lin_reg = LinearRegression()
      lin_reg.fit(storage_set_prepared, storage_set_labels)

      storage_set_predictions = lin_reg.predict(storage_set_prepared)
      lin_mse = mean_squared_error(storage_set_labels, storage_set_predictions)
      lin_rmse = np.sqrt(lin_mse)

      lin_rmse

[15]: 4396.838493147415
```

Mallin tulisi ennustaa ”Provided”-muuttujan arvoa asiakkaan tietojen perusteella. Opetusaineistossa ”Provided”-muuttujan arvot vaihtelevat välillä 0–30 000, joten lineaarisen regressiomallin keskimääräinen 4396,84 virhe on liian suuri. Mallin sovittaminen ei onnistu joko puutteellisten tietojen, tai vääränlaisen mallin valinnan vuoksi (Géron, 2017, s. 68). Epäilen syyn olevan tausta-aineiston useissa luokittelevissa muuttujissa, joiden vaikutus lopputulokseen ei ole lineaarinen.

Kokeilen seuraavaksi päätöspuualgoritmia (Decision tree), joka kykenee tehokkaasti sekä luokitteluun että regressioon ja havainnoimaan myös epälineaarisia suhteita (Géron, 2017, s. 167). Algoritmi luo päätöksentekosäännöistä hierarkkisen puumallin, joka koostuu päätöksentekosäännöistä. Päätöspuualgoritmin tulos on esitetty kuvassa 22.

Kuva 22 - Päätöspuualgoritmi

```
[19]: #DecisionTreeRegressor

      from sklearn.tree import DecisionTreeRegressor
      from sklearn.metrics import mean_squared_error
      import numpy as np

      tree_reg = DecisionTreeRegressor()
      tree_reg.fit(storage_set_prepared, storage_set_labels)

      storage_set_predictions = tree_reg.predict(storage_set_prepared)
      tree_mse = mean_squared_error(storage_set_labels, storage_set_predictions)

      tree_rmse = np.sqrt(tree_mse)
      tree_rmse

[19]: 0.0
```

Tuloksen perusteella päätöspuualgoritmi ennustaa oikean vastauksen virheettömästi. Tämä voi johtua synteettisen datani suhteellisen helposti yleistettävistä riippuvuuksista, tai kuten Aurélien Géronin kirjan esimerkissä, ylisovittamisesta (overfitting) (Géron, 2017, s. 69). Asian varmistamiseksi käytän saman kirjan esittelemää tekniikkaa, jossa opetusaineisto jaetaan kymmeneen osaan, joita koulutetaan ja evaluoidaan toistensa avulla.

Kuvan 23 perusteella mediaanivirhe on n. 1309. ”Provided”-arvojen mediaani on 5755,67 jolloin virhe olisi n. 23 %. Onko tämä hyväksyttävää vai ei riippunee mallin lopullisesta käyttötarkoituksesta. Toimintaa voi vielä hienosäätää hyperparametrien avulla esimerkiksi määrittämällä päätöspuun maksimisyvyys. Ilman rajoittavia parametreja päätöspuualgoritmi on herkkä ylisovittamaan koulutusdatan (Géron, 2017, s. 173).

Kuva 23 - Ristiinvalidointi

```
[16]: #Cross validation

from sklearn.model_selection import cross_val_score

scores = cross_val_score(tree_reg, storage_set_prepared, storage_set_labels,
                         scoring="neg_mean_squared_error", cv=10)

rmse_scores = np.sqrt(-scores)

def display_scores(scores):
    print("Scores:", scores)
    print("Mean:", scores.mean())
    print("Standard deviation:", scores.std())

display_scores(rmse_scores)
```

```
Scores: [1337.87187308 1254.24935973 1273.89691177 1281.5712459  1319.20629651
 1304.90941236 1336.03465023 1363.81428664 1349.93348213 1272.44754201]
Mean: 1309.3935060368362
Standard deviation: 35.61213897445737
```

7.2 Ryvästäminen ja kutakin toimituspistettä palvelevien varastojen määrittäminen

Edellisen luvun algoritmi tunnistaa toimitushistoriasta ne asiakkaan tiedot, joilla on vaikutusta asiakkaalle toimitettuun nimikekohtaiseen lukumäärään. Koska aineistossa ei ole määritetty varastoja, joista materiaalia tyypillisesti kuhunkin toimituspisteeseen lähetetään, käytän niiden määrittämiseen ryvästysalgoritmia. Kuten totesin luvussa 5.3, ryvästysalgoritmi etsii datasta yhtymäkohtia ja luokittelee datan niiden perusteella. Ryvästysalgorimeja ovat (Géron, 2017, s. 10):

- k-Means
- Expectation Maximization
- Hierarchical Cluster Analysis (HCA)

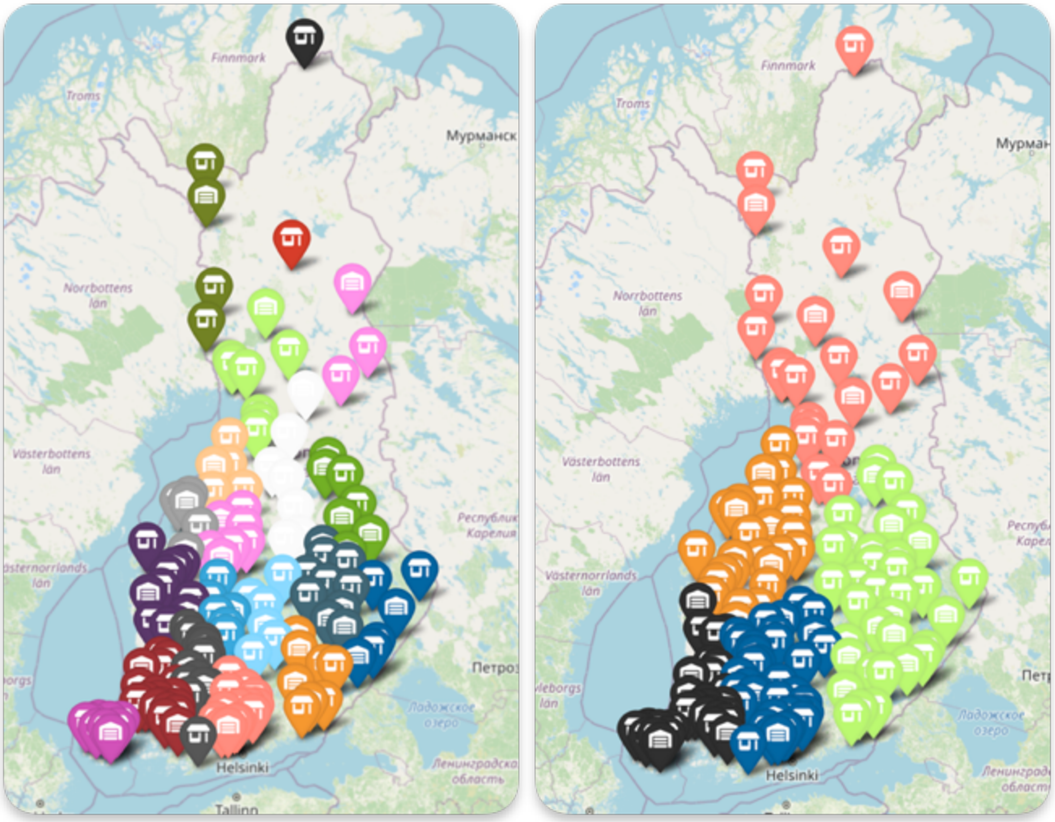
K-Means-algoritmi valitsee aluksi satunnaiset tietueet ryppäiden keskipisteiksi. Tämän jälkeen se luokittelee kaikki tietueet kuuluviksi johonkin ryppääseen perustuen niiden laskennalliseen ”etäisyyteen” keskipisteistä. Keskipisteen sijainti päivitetään vastaamaan kaikkien ryppääseen kuuluvien tietueiden mediaania. Luokittelu ja keskipisteen päivittäminen toistetaan, kunnes merkittävää muutosta ei enää tapahdu. (Alpaydin, 2020, s. 188-189)

Expectation Maximization-algoritmia käytetään tilastollisten mallien parametrien arviointiin, erityisesti tapauksissa, joissa mukana on piileviä (havaitsemattomia) muuttujia. Toisin kuin K-means-algoritmi, EM-algoritmi tuottaa arvion tietueen kuulumisesta ryppääseen jollain todennäköisyydellä. (Alpaydin, 2020, s. 190-195)

Hierarkkinen analyysi (Hierarchical Cluster Analysis) voi olla kasautuvaa (agglomerative) tai jakautuvaa (divisive). Kasautuva algoritmi aloittaa N ryhmästä, yhdistäen samankaltaisia ryhmiä, kunnes jäljellä on yksi. Jakautuva algoritmi taas toimii päinvastaisessa järjestyksessä. Hierarkkinen klusterianalyysi tuottaa puun, joka kuvaa havaintojen hierarkkista ryhmittelyä, ja tulkinta voi olla monimutkaisempaa kuin esimerkiksi selkeän ryhmittelyn tuottavassa k-means-algoritmissa. (Alpaydin, 2020, s. 199-201)

K-Means-algoritmi vaikuttaa käyttökelpoiselta yksinkertaiseen luokitteluun koordinaattien perusteella. Niputan varastojen ja toimituspisteiden koordinaatit samaan taulukkoon, jonka jälkeen voin helposti luokitella ne ryhmiin (kuva 24). Käytän kahta erikokoista ryhmittelyä, jotta kullekin toimituspisteelle syntyisi ensisijaiset (lähimmät) varastot, mutta kukin varasto palvelisi tarvittaessa toimituspisteitä myös laajemmalla alueella.

Kuva 24 - Varastot ja toimituspisteet ryvästettynä



Niputtamalla sekä toimituspisteet että varastot samoihin ryppäisiin voidaan edellisen luvun regressioalgoritmia apuna käyttäen arvioida nimikekohtaisesti erilaisten tuotteiden tarve kunkin ryppään varastoissa.

8 Pohdinta

Valitsin tutkimustyön aiheen oppiakseni koneoppimisalgoritmien ohjelmointia ja ymmärtääkseni paremmin koneoppimisen mahdollisuuksia. Minulla oli aikaisempaa kokemusta ohjelmoinnista, myös Pythonilla, mutta ei lainkaan koneoppimisesta käytännössä. Tutkimustyön tekeminen oli tämän vuoksi hidas, mutta erittäin palkitseva prosessi, jonka aikana opin tekstissä kuvattujen kokonaisuuksien ulkopuolelta myös datan keräämisen, dataputkien rakentamisen ja pilvipalveluiden käytön periaatteet. Mystinen tekoäly alkoi vaikuttaa varsin ymmärrettävältä.

Synteettisen datan rakentaminen tutkimustyön skenaariota varten osoittautui yllättävän työlääksi. Satunnaisten numeroiden sijaan sen tuli sisältää aitoja riippuvuuksia ja säännönmukaisuuksia algoritmien löydettäväksi. Osittain tästä syystä lopullisesta aineistosta

tuli verrattain yksinkertainen. Todellinen aineisto, jota tulisi käyttää varastonohjauksen tukena olisi huomattavasti moniulotteisempi. Anssi Lotvonen (2021) on omassa opinnäytetyössään *Tekoälyn hyödyntäminen varastonohjauksessa* avannut joitakin huomioitavia kokonaisuuksia. Näitä ovat nimikkeen taloudellinen tilauserä, tilaamisen ja varastoinnin kustannukset, määräalennukset, varmuusvarastot, laadun ja tuotannon ongelmat, nimikekohtainen arvon tuotto, varastokapasiteetti sekä tuotteiden eräkoot ja tilan tarve.

Keinotekoisen datan käytöstä voi ajatella olevan sekä hyötyä, että haittaa. Tietosuojan tai lupien kanssa ei synteettistä dataa käytettäessä synny ongelmia. Se myös auttaa tutkimusskenaarion rakentamisessa juuri halutun laiseksi eikä siis sisällä yllätyksiä tai poikkeamia, ellei niitä erikseen haluta. Toisaalta tämä voi johtaa siihen, että itse rakennetulla datalla koulutettu malli ei toimi tosielämän aineistolla, joka ei välttämättä ole yhtä säännönmukaista. Data voi jakautua epänormaalisti tai sisältää vinoumia, jotka koneoppimismalli omaksuu.

Tässä opinnäytetyössä itse tekemäni data palveli tarkoitustaan ja datan tuntemisesta oli hyötyä arvioidessani koneoppimisalgoritmin suoriutumista. Olisin voinut myös valita tarkasteltavan tietoaineiston jostain internetin kehittämisalustoista, kuten itse käyttämästäni GitHubista (luku 2.2) ja muokata luomaani skenaariota siten, että se sopii saatavilla olleeseen aineistoon. Jos nyt aloittaisin vastaavan työn teon uudestaan, kiinnittäisin vielä enemmän huomiota aineiston ja skenaarion suunnitteluun ennen ohjelmointiin ryhtymistä. Päättäisin tietotaulujen sarakkeiden yhtenäisestä otsikoinnista kielen, kirjainkokojen ja välimerkkien tarkkuudella sekä varmistuisin siitä, että eri taulut tukevat toisiaan eivätkä sisällä tarpeetonta toistoa.

Tutkimuksen tavoitteena oli luoda koneoppimisen malli, joka ennustaa erityyppisten asiakkaiden tilauksia aiempien tilausten perusteella ja sitä kautta auttaa ylläpitämään oikean määrän oikeita tuotteita sopivissa varastoissa. Käytännössä malli siis tunnisti datasta samat säännönmukaisuudet, jotka olin aineistoa laatiessa sinne erilaisia kertoimia käyttämällä syöttänyt. Tavoitteen toteutumisesta voinen todeta, että malli voi auttaa oikean varastotason ylläpitämisessä, mutta kuten aiemmin tässä luvussa totesin, toimivan kokonaisuuden rakentaminen edellyttäisi kattavampaa tietoaineistoa.

9 Lopputulokset ja suositukset

Päätöspuualgoritmia tai vastaavaa ohjatun oppimisen algoritmia voidaan tutkimustyön perusteella käyttää säännönmukaisuuksien tunnistamiseen asiakkaiden tilausdatasta. Algoritmin käyttö edellyttää datan tarkkaa analysointia ja dataputken rakentamista aineiston muokkaamiseksi algoritmille sopivaan muotoon sekä hyperparametrien säätämistä ylisovittamisen estämiseksi. Algoritmia voi hyödyntää esimerkiksi ennusteiden laatimisessa laajennettaessa toimintaa uudelle alueelle tai missä tahansa liiketoiminnassa, jossa on optimoitava materiaalin varastointia tai kuljettamista. Algoritmit voivat auttaa ennustamaan kysyntää ja tehostamaan varastonhallintaa, mikä auttaa vähentämään varastokustannuksia ja parantamaan toimitusketjun tehokkuutta. Lisäksi asiakasdatan perusteella asiakkaat voidaan segmentoida eri ryhmiin, jotta markkinointi voidaan kohdistaa ja personoida paremmin.

Ryvästysalgoritmia voidaan hyödyntää kunkin asiakkaan kannalta optimaalisten lähettävien varastojen valinnassa. Yksinkertaisimmillaan algoritmi voi luokitella asiakkaat ja varastot sijainnin perusteella, kuten tässä opinnäytetyössä, mutta ryvästäminen voidaan tehdä myös vanhan tilaus- ja toimitusdatan perusteella, jos esimerkiksi varaston tiedot tietojärjestelmässä ovat puutteelliset ja suunnittelu on aiemmin tehty järjestelmiin kirjaamattoman hiljaisen tiedon perusteella.

Johtopäätöksenä toimitusketjun hallinnassa auttavien koneoppimismallien toimimiseksi tulisi ylläpitää tietokantaa asiakkaiden tekemistä tilauksista. Tietokannasta tulisi ilmetä tilauksen tekijä, tarkka sisältö ja ajankohta ja mahdollisesti järjestelmä, jonka kautta tilaus tehtiin. Varastojen tietokanta voisi sisältää aikaleimatut tiedot saapuneesta ja lähteneestä materiaalista. Tämän lisäksi voidaan ylläpitää tietoja lähteneen materiaalin toimitusten etenemisestä ja käytetyistä kuljetusreiteistä. Mitä enemmän tietoa on käytettävissä asiakkaasta ja tilatuista nimikkeistä, sitä tarkempaan ennustamiseen ja asiakaskohtaiseen palveluun tai markkinointiin kyetään.

Opinnäytetyön tuloksia voinee sopivasti muokattuna hyödyntää käytännössä. Todennäköisesti tosielämän käyttötapaus olisi moniulotteisempi ja edellyttäisi monimutkaisempien dataputkien rakentamista puuttuvan datan eheyttämiseksi sekä tiedon käsittelemiseksi tarkoituksenmukaiseen muotoon.

Lähteet

Alasuutari, Pertti. (2012). *Laadullinen tutkimus 2.0*. Kustannusosakeyhtiö Vastapaino.

Alpaydin, Ethem. (2020). *Introduction to Machine Learning.* Massachusetts Institute of Technology.

Alpaydin, Ethem. (2021). *Koneoppiminen*. Terra Cognita.

Bister, Timo (2019). *Tietojenkäsittelyn opinnäytetyö: viittoja ja karttoja tutkimisen ja kehittämisen teille*. Jyväskylän ammattikorkeakoulu.

Fry, Hannah. (2019). *Hello world: Kuinka selviytyä algoritmien aikakaudella.* Bazar Kustannus Oy.

Géron, Aurélien. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'reilly.

Gori, M. (2017). *Machine Learning: A Constraint-Based Approach*. Elsevier Science & Technology.

Hakala, Juha T. (2022). *Hyvä, parempi, valmis. Opinnäytetyöopas ammattikorkeakouluille.* Gaudeamus Oy.

Hämeen ammattikorkeakoulu. (2023). *Opinnäytetyöohje*. https://www.hamk.fi/opiskelijan-ohjeet/opinnaytetyo/

Kluyver, T. et al. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. Positioning and Power in Academic Publishing. s. 87.90

Larrañaga, P., Atienza, D., Diaz-Rozo, J., Ogbechie, A., Puerto-Santana, C., Bielza, C. (2018). *Industrial Applications of Machine Learning*. Taylor & Francis Group.

Lotvonen, Anssi (2021). *Tekoälyn hyödyntäminen varastonohjauksessa.* [opinnäytetyö, Jyväskylän ammattikorkeakoulu] https://urn.fi/URN:NBN:fi:amk-2021060213377

Nkuliza, Katia (2020). *Tekoäly: Mahdollisuudet ja näkymät logistiikassa.* [opinnäytetyö, Kaakkois-Suomen ammattikorkeakoulu] https://urn.fi/URN:NBN:fi:amk-2020120125496

Ojasalo, K., Moilanen, T., Ritalahti, J. (2015). *Kehittämistyön menetelmät.* Sanoma Pro Oy.

Parker, James R. (2021). *Python: An Introduction to Programming.* Mercury Learning and information.

Peixeiro, Marco (2022). *Time Series Forecasting in Python*. Manning Publications Co.

Penttilä, Jeremias (2015). *Koneoppiminen*. [opinnäytetyö, Jyväskylän yliopisto] https://jyx.jyu.fi/bitstream/handle/123456789/49137/URN:NBN:fi:jyu-201603211906.pdf?sequence=1

Penttilä, Oiva (2023). *Logistiikan kehittäminen datalla*. [opinnäytetyö, Hämeen ammattikorkeakoulu] https://urn.fi/URN:NBN:fi:amk-2023070624523

Pérez, F., Granger, B. E. (2007). IPython: A System for Interactive Scientific Computing, Computing in Science and Engineering, vol. 9, no. 3, s. 21-29. doi:10.1109/MCSE.2007.53. URL: https://ipython.org

Project Jupyter (2023). Project Jupyter Documentation. https://docs.jupyter.org

Puusa, Anu & Juuti, Pauli. (2020). *Laadullisen tutkimuksen näkökulmat ja menetelmät*. Gaudeamus Oy.

Reijo Rautauoman säätiö. (n.d.). Logistiikan maailma. https://www.logistiikanmaailma.fi

Ritvanen, V., Inkiläinen, A., von Bell, A. & Santala, J. (2011). *Logistiikan ja toimitusketjun hallinnan perusteet*. Reijo Rautauoman säätiö.

Rossant, C. (2014). IPython Interactive Computing and Visualization Cookbook. Packt Publishing.

Toomey, D. (2018). Jupyter Cookbook. Packt Publishing.

Tuominen, H. & Neittaanmäki, P. (2019). *Tekoälyn perusteita ja sovelluksia.* Jyväskylän yliopisto.

Van Rossum, G. (2009). A Brief Timeline of Python. https://python-history.blogspot.com

Vorderman, C., Steele, C., Quigley, C., Goodfellow, M., McCafferty, D. & Woodcock, J. (2017). *Python-projekteja! Opiskelijan ohjelmointikirja*. Readme.fi.

Wikimedia Foundation. (n.d.). *GitHub*. https://en.wikipedia.org/wiki/GitHub