

Lappeenrannan-Lahden teknillinen yliopisto LUT  
School of Engineering Science  
Tietotekniikan koulutusohjelma

Kandidaatintyö

**Juho Hupanen**

**OHJELMISTOON INTEGROITAVIEN TEKOÄLYMODUULIEN  
NYKYTILA-ANALYYSI**

Työn tarkastaja: Tekniikan Tohtori Ari Happonen

Työn ohjaaja: Tekniikan Tohtori Ari Happonen

# TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT

School of Engineering Science

Tietotekniikan koulutusohjelma

Juho Hupanen

## Ohjelmistoon integroitavien tekoälymoduulien nykytila-analyysi

Kandidaatintyö 2020

38 sivua, 9 kuvaa, 3 taulukkoa

Työn tarkastajat: Tekniikan Tohtori Ari Happonen

Hakusanat: AI, tekoäly, koneoppiminen, kehys

Keywords: AI, artificial intelligence, machine learning, framework

Tekoäly on viimeisimpien vuosien aikana kehittynyt yhdeksi tietotekniikan trendikkäimmistä osa-alueista. Samalla myös erilaisten tekoälyratkaisujen määrä on kasvanut merkittävästi. Tämän työn tavoitteena on selvittää näiden ratkaisujen nykytilanne ja luoda katsaus suosituimpiin kirjastoihin sekä neljään ekosysteemiin. Lisäksi perehdytään tekoälyn keskeisiin käsitteisiin. Tarkastelun kohteena ovat ratkaisujen määrä, laajuus ja integroitavuus. Kartoitettaviksi valittiin Amazonin, Googlen, IBM:n ja Microsoftin ekosysteemit. Työssä selvisi, että markkinoilta löytyy kymmeniä koneoppimiskirjastoja ja kehitystyötä helpottavia työkaluja. Jotkut näistä, kuten TensorFlow, PyTorch ja Keras, ovat saavuttaneet vakiintuneen aseman tekoäly-yhteisön keskuudessa. Tulevaisuudessa tekoäly tulee olemaan yhä olennaisempi osa yritysten liiketoimintaa ja näin ollen kysyntä tekoälyratkaisuille tulee vain kasvamaan.

## **ABSTRACT**

Lappeenranta-Lahti University of Technology LUT  
School of Engineering Science  
Degree Programme in Software Engineering

Juho Hupanen

### **An analysis of the current state of integrable AI software modules**

Bachelor's Thesis 2020

38 pages, 9 figures, 3 tables

Examiners: D.Sc. (Tech.) Ari Happonen

Keywords: AI, artificial intelligence, machine learning, framework

Artificial intelligence has developed into one of the trendiest fields in information technology during the last few years. At the same time, the number of different artificial intelligence solutions has also grown significantly. The aim of this thesis is to find out the current state of these solutions and to provide an overview of the most popular libraries as well as four ecosystems. Additionally, the work looks into the most essential concepts of artificial intelligence. The focus is on the quantity, the scope and the integrability of the solutions. The ecosystems of Amazon, Google, IBM and Microsoft were selected for the mapping. It was found out that there are dozens of machine learning libraries and tools making the development easier on the market. Some of these, such as TensorFlow, PyTorch and Keras, are well established frameworks among the artificial intelligence community. In the future, artificial intelligence will be an increasingly integral part of the businesses and therefore the demand for artificial intelligence solutions is only set to grow.

# SISÄLLYSLUETTELO

<b>1</b>	<b>JOHDANTO.....</b>	<b>3</b>
1.1	TAVOITTEET JA RAJAUKSET .....	3
1.2	TYÖN RAKENNE .....	4
<b>2</b>	<b>TUTKIMUSMENETELMÄ.....</b>	<b>5</b>
<b>3</b>	<b>TEKOÄLY.....</b>	<b>6</b>
3.1	TEKOÄLYN MÄÄRITELMÄ .....	6
3.2	KONEOPPIMINEN JA SEN ALGORITMIT .....	6
3.3	SYVÄOPPIMINEN JA NEUROVERKOT .....	8
3.4	MUITA SOVELLUKSIA JA OSA-ALUEITA .....	9
<b>4</b>	<b>TEKOÄLYKIRJASTOJEN JA -ALUSTOJEN MARKKINATILANNE .....</b>	<b>11</b>
4.1	TEKOÄLYKIRJASTOT .....	11
4.2	TEKOÄLYALUSTOJEN TAUSTAA .....	14
4.3	TARKASTELTAVIEN EKOSYSTEEMIEN ESITTELY .....	16
4.4	AMAZON.....	19
4.5	GOOGLE.....	21
4.6	IBM.....	23
4.7	MICROSOFT.....	24
<b>5</b>	<b>TULOKSET, POHDINTA JA TULEVAISUUS .....</b>	<b>27</b>
5.1	TULOKSET.....	27
5.2	POHDINTA JA TULEVAISUUS .....	28
<b>6</b>	<b>YHTEENVETO.....</b>	<b>30</b>
	<b>LÄHTEET.....</b>	<b>31</b>

## **SYMBOLI- JA LYHENNELUETTELO**

ACM	Association for Computing Machinery
AI	Artificial Intelligence
AMI	Amazon Machine Image
API	Application Programming Interface
AWS	Amazon Web Services
CNN	Convolutional Neural Network
CNTK	Computational Network ToolKit, Microsoft Cognitive Toolkit
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
FNN	Feedforward Neural Network
GCP	Google Cloud Platform
IaaS	Infrastructure as a Service
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
ML	Machine Learning
MLaaS	Machine Learning as a Service
NLP	Natural Language Processing
NN	Neural Network
ONNX	Open Neural Network Exchange
PaaS	Platform as a Service
RNN	Recurrent Neural Network

# 1 JOHDANTO

Teknologian kehittyminen on varsinkin 2010-luvun aikana mahdollistanut erilaisten tekoälyratkaisujen määrän räjähdysmäisen kasvun. Suuremman laskentatehon, kehittyneempien algoritmien ja kasvaneen tallennuskapasiteetin ansiosta tekoäly peittoaa ihmisen yhä useammassa tehtävässä. Tästä kuuluisimpia esimerkkejä ovat IBM:n Deep Blue, joka voitti shakissa senhetkisen hallitsevan maailmanmestarin Garry Kasparovin vuonna 1997 ja Googlen AlphaGo, joka löi go-pelin korkeatasoisen ammattilaispelaaja Lee Sedolin vuonna 2016 (Li & Du, 2007). Tekoälylle löytyy jatkuvasti uusia käyttökohteita ja sitä hyödynnetäänkin jo myös perinteisemmällä vauhdikkaasti digitalisoituvilla aloilla, kuten teollisuudessa. Kyseisillä aloilla on aiemmin oltu vahvasti tuotekeskeisiä ja siirrytään nyt kohti informaatio- ja palvelukeskeisiä liiketoimintamalleja (Kortelainen et al., 2019).

Ohjelmistojen kehittäminen ja ylläpitäminen vaatii runsaasti työtunteja. Jos näitä työtunteja ei käytetä tehokkaasti, yritys voi jäädä alati kasvavilla markkinoilla jälkeen kilpailijoistaan. Näitä tulee markkinoille sekä uusien IT-yritysten ja startupien että perinteisten palvelu- ja tuoteyritysten muodossa, näiden muokatessa liiketoimintaansa tietolähtöisemmäksi (Kinnunen et al., 2019; Kortelainen et al., 2019) ja koko tuotekalustoansa hallinnoiviksi liiketoimintaratkaisuuksi (Kortelainen et al., 2016) osana globaalia digitalisaatioaaltoa. Kehitystyön helpottamista ja tehostamista varten onkin olemassa valmiita ratkaisuja, jotka ovat integroitavissa kehitettävään sovellukseen. Tällaisia ratkaisuja ovat esimerkiksi ohjelmistokehityksessä käytettävät kirjastot, jotka tarjoavat joukon aliohjelmia, joita voidaan uudelleenkäyttää hyödyntämällä niitä omassa ohjelmakoodissa.

## 1.1 Tavoitteet ja rajaukset

Tässä kandidaatintyössä tutustutaan ensin tekoälyn tärkeimpiin käsitteisiin, jonka jälkeen tutkitaan erilaisia tekoälyalustoja ja -kehyksiä. Tutkimuskysymyksinä ovat: ”Kuinka paljon kullekin alustalle on tällä hetkellä tarjolla erilaisia tekoälyä hyödyntäviä ohjelmistoon integroitavia ratkaisuja?”, ”Kuinka laajoja nämä ratkaisut ovat?” ja ”Miten kattavaa on näiden ratkaisujen dokumentaatio ja kuinka helposti integroitavia ne ovat?”

Työ on rajattu siten, että tutkimuksen kohteena on vain neljän suurimman (Amazon, Microsoft, Google ja IBM) pilvipalveluita tarjoavan ohjelmistoyhtiön tekoälykehitysekosysteemit ja näiden tukemat ohjelmistokehykset. Tähän rajaukseen päädyttiin, koska työssä haluttiin keskittyä erityisesti pilvipohjaisiin ratkaisuihin, ja kyseiset yhtiöt kattavat yli puolet maailmanlaajuisista alusta- ja infrastruktuuripilvipalveluiden markkinoista (ITCandor, 2020). Ottaen huomioon kandidaatintyön aikarajoitteet arvioitiin, että neljän suurimman toimijan tarjoamat tuotteet ja palvelut antanevat tässä tapauksessa riittävän kattavan yleiskuvan markkinatilanteesta.

## **1.2 Työn rakenne**

Johdannon jälkeen toisessa luvussa kuvaillaan työssä käytettäviä tutkimusmenetelmiä. Kolmannessa luvussa olevan kirjallisuuskatsauksen tarkoituksena on tarkastella lähemmin tekoälyn tärkeimpiä peruskäsitteitä ja alalajeja. Kirjallisuuskatsauksen jälkeen kartoitetaan markkinoiden tarjontaa tekoälykirjastojen ja -kehysten osalta, ja analysoidaan näiden käyttökohteita ja toimintaperiaatteita. Tässä kohtaa käydään läpi myös käsiteltävät ekosysteemit. Viidennessä luvussa keskustellaan työn tuloksista ja pohditaan tekoälyratkaisujen tulevaisuutta. Luvussa 6 tehdään yhteenveto.

## 2 TUTKIMUSMENETELMÄ

Työn teoriaosuudessa käytettiin kirjallisuuskatsausta. Lähteet kirjallisuuskatsaukseen haettiin verkkotietokannoista, kuten Google Scholar, ACM Digital Library (Association for Computing Machinery) ja IEEE Xplore (Institute of Electrical and Electronics Engineers). Hakukoneena toimi LUT Finna. Haut tapahtuivat yhdistelemällä erilaisia hakusanoja muun muassa tekoälyyn ja koneoppimiseen liittyen. Tuloksista karsittiin pois lähteet, jotka oli julkaistu ennen vuotta 2010, ja mukaan otettiin vain englanninkieliset julkaisut. Perusteena ajalliselle rajaukselle oli tekoälyn viime vuosikymmen aikana syväoppimisen myötä tieteenalana kokema uudelleensyntyminen. Ottamalla mukaan vain tuoreimmat julkaisut pyrittiin varmistamaan lähdemateriaalin tuoreus ja relevanttius. Jäljelle jäävistä tuloksista luettiin tiivistelmät, jonka jälkeen kirjallisuuskatsauksessa hyödynnettiin soveltuvien osien valittuja julkaisuja.

Tutkimuksessa noudatettiin pääosin kvalitatiivista tutkimusmenetelmää, mutta sen tukena käytettiin myös kvantitatiivisia menetelmiä. Aineiston analysoinnissa sovellettiin vertailevaa tutkimusta. Työssä tarkasteltiin tekoälykehyksiä sekä -alustoja ja näiden tarjoamia API-rajapintoja. Arvioinnin kohteeksi valittiin neljä markkinoiden suosituinta ekosysteemiä ja näiden tukemat tekoälykehykset. Tämän jälkeen jokainen ekosysteemi kartoitettiin tapaustutkimusten avulla.



### **3 TEKOÄLY**

Tässä luvussa luodaan teoreettinen pohja suoritettavalle tutkimukselle kirjallisuuskatsauksen avulla. Tarkoituksena on kartoittaa erilaisia tekoälyyn rinnastettavia käsitteitä siten, että tässä luvussa tehtyjä havaintoja voidaan käyttää työssä tarkasteltavien tekoälyratkaisujen vertailun pohjana.

#### **3.1 Tekoälyn määritelmä**

Tekoälyn (Artificial Intelligence, AI) määritelmästä ei ole olemassa selkeää yksittäistä kuvausta. Tämä johtunee siitä, että se on käsitteenä monitulkintainen ja sille on vaikea kehittää kaikenkattavaa selitystä. Määrittelemisen vaikeus piilee nimenomaan termin osassa äly (intelligence), sillä sen merkityksestä ei ole päästy täysin yhteisymmärrykseen (Turner, 2019). Erilaisia versioita määritelmästä on kuitenkin vuosien varrella esitetty, joista ehkä yksi klassisimmista esimerkeistä on John McCarthyn vuonna 1955 luoma määritelmä. Hän määrittelee tekoälyn tavoitteeksi kehittää koneita, jotka käyttäytyvät kuin ne olisivat älykkäitä. (Ertel, 2011) Toinen kuuluisa esimerkki on Alan Turingin kehittämä Turing-testi, jossa koehenkilön tehtävänä on tunnistaa, onko testin toinen osapuoli ihminen vai kone. Jos kone onnistuu esittämään ihmistä uskottavasti, se on osoittautunut älykkääksi. (Turner, 2019) Edellä mainittujen ihmiskeskeisten määritelmien rinnalle on sittemmin tullut useita rationaalisia määritelmiä, jotka pyrkivät jättämään ihmisen pois yhtälöstä.

Tässä työssä termiä tekoäly käytetään eräänlaisena sateenvarjokäsitteenä, joka kattaa useita eri tekoälyyn yleisesti rinnastettavissa olevia teknologioita. Kyseiset teknologiat on pyritty käsittelemään tässä kirjallisuuskatsauksessa mahdollisimman kattavasti.

#### **3.2 Koneoppiminen ja sen algoritmit**

Koneoppiminen (Machine Learning, ML) on yksi tekoälyn keskeisimmistä osa-alueista. Sen tarkoituksena on kouluttaa tekoäly suoriutumaan tehtävästään antamalla sille syötteeksi ennalta määriteltyä koulutusdataa (Louridas & Ebert, 2016). Koulutettua algoritmia voidaan myöhemmässä vaiheessa soveltaa uuteen dataan. Tekoälyn pioneeri Arthur Samuel

määrittelikin koneoppimisen vuonna 1959 tutkimusalana, joka antaa tekoäylle kyvyn oppia jotakin ilman, että sitä olisi siihen erityisesti ohjelmoitu (Turner, 2019).

Koneoppimisalgoritmit voidaan yleisesti jakaa kolmeen eri alatyypin: ohjattu oppiminen (Supervised Learning), ohjaamaton oppiminen (Unsupervised Learning) ja vahvistusoppiminen (Reinforcement Learning) (Turner, 2019). Näistä ensimmäisessä, ohjatussa oppimisessa, syötteenä olevan datan mukana annetaan myös oikeat tulokset. Toiset ohjatun oppimisen algoritmeista perustuvat luokitteluun (classification) ja toiset regressioon (regression). Luokittelussa syötteinä ovat opetusdata ja jokaisen opetusdataan kuuluvan alkion luokka. Näiden tietojen avulla on tarkoitus opettaa tekoäly luokittelemaan uutta dataa. Regressiossa algoritmi pyrkii ennustamaan jatkuva-arvoisen muuttujan arvon. (Louridas & Ebert, 2016)

Ohjaamaton oppiminen eroaa ohjatusta oppimisesta siten, että siinä algoritmille ei anneta palautetta eikä oikeita tuloksia vaan sen on löydettävä ne itse (Louridas & Ebert, 2016). Ohjaamattoman oppimisen edut pääsevätkin oikeuksiinsa etsittäessä uutta tietoa datasta, koska tekoäly kykenee itsenäisesti löytämään erilaisia malleja ja yhteyksiä alkioden välillä ja voi löytää ihmisiltä huomioimatta jääneitä seikkoja (Turner, 2019). Ohjaamattomassa oppimisessa voidaan käyttää menetelmänä joko klusterointia (clustering) tai ulottuvuuden vähentämistä (dimensionality reduction). Klusterointialgoritmit ryhmittelevät dataa klustereihin jonkin tietyn kriteerin perusteella. Ulottuvuuden vähentämistä käyttävät algoritmit poistavat datasta ylimääräisiä ulottuvuuksia. Näin pyritään saamaan paremmin selville datan tärkeimmät ominaisuudet. (Louridas & Ebert, 2016)

Vahvistusoppiminen on siinä mielessä erityinen menetelmä, että algoritmi ei käytä oppiessaan minkäänlaista opetusdataa, vaan se oppii ihmisen lailla yrityksen ja erehdyksen kautta. Tällä tavalla algoritmi oppii toimimaan kussakin tilanteessa parhaaksi havaitulla tavalla. (Ertel, 2011) Oppimisprosessissa päätösten hyvyden arvioi joko ihminen tai automaattinen palautemekanismi, joka antaa päätöksille pisteitä sen mukaan, kuinka haluttu saatu tulos on (Turner, 2019).

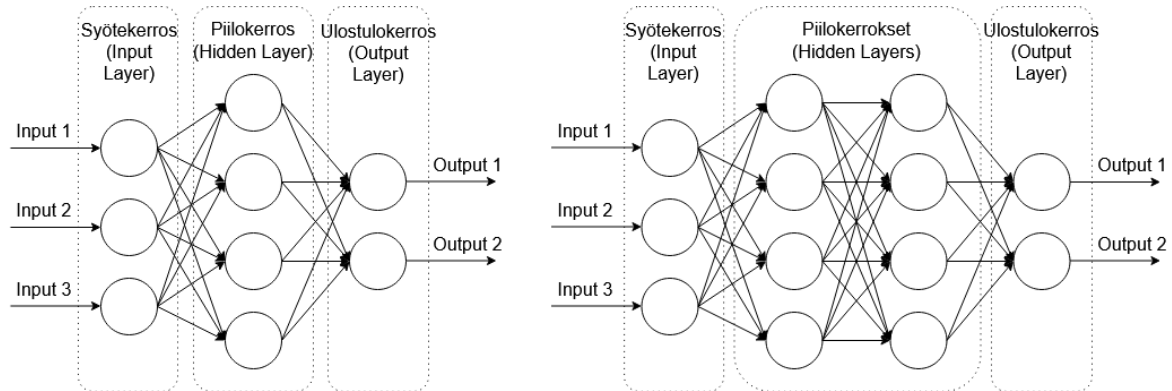
### 3.3 Syväoppiminen ja neuroverkot

Syväoppimisen (Deep Learning, DL) suosio on viime aikoina kasvanut runsaasti sen avulla saatujen lupaavien tulosten ansiosta. Se on lyönyt muut koneoppimismenetelmät esimerkiksi hahmontunnistuksessa, hiukkaskiihdytindatan analysoimisessa ja aivojen hermoverkkojen rekonstruktiossa (LeCun et al., 2015). Tämä on näkynyt muun muassa erilaisissa kilpailuissa, joissa konenäön syväoppimisalgoritmit ovat osoittautuneet joillain aloilla jopa ihmistä paremmiksi (Schmidhuber, 2015).

Syväoppiminen perustuu suurilta osin neuroverkkoihin (Neural Network, NN), tarkemmin niiden alalajiin syviin neuroverkkoihin (Deep Neural Network, DNN), ja näiden opettamiseen. Yleisesti neuroverkko koostuu keinotekoisista neuroneista, joiden toimintamalli perustuu näiden biologisten esikuviansa toiminnan yksinkertaistettuun imitoimiseen. Keinotekoiset neuronitkin ovat samalla tavalla yhteydessä muihin neuroneihin ja prosessoivat tulevia signaaleja, jonka jälkeen ne lähettävät generoidun ulostulon eteenpäin syötteenä muille neuroneille. Neuroverkot koostuvat syötekerroksesta (input layer), yhdestä tai useammasta piilokerroksesta (hidden layer) sekä ulostulokerroksesta (output layer). Kuten kuvasta 1 on nähtävissä, siinä missä tavallisella ei-syvällä neuroverkolla on vain yksi piilokerros, on syvällä neuroverkolla näitä aina kaksi tai useampia. (Sewak, 2019) Käytännössä neuroverkkojen opettaminen tarkoittaa sellaisten painoarvojen löytämistä neuroneille, jotka saavat verkon käyttäytymään halutulla tavalla (Schmidhuber, 2015).

Erilaisia neuroverkkoarkkitehtuureja on olemassa kymmenittäin, mutta etenkin seuraavat kolme rakennemallia ovat olleet syväoppimisessa merkittävässä osassa: eteenpäin kytketyt neuroverkot (Feedforward Neural Network, FNN), takaisinkytketyt (Recurrent Neural Network, RNN) neuroverkot ja konvoluutioneuroverkot (Convolutional Neural Network, CNN). FNN-verkossa signaalit kulkevat vain yhteen suuntaan eli edellisestä kerroksesta seuraavaan. (Sewak, 2019) RNN-verkko käsittelee dataa vaiheittain. Tästä syystä sitä käytetään usein puheen ja kielen prosessoimiseen. (LeCun et al., 2015) CNN-verkot koostuvat pääosin konvoluutio- (convolutional layer) ja koontikerroksista (pooling layer). Konvoluutiokerrokset tuottavat ulostuloinaan piirrekarttoja (feature map), jotka ne siirtävät syötteiksi seuraavalle kerrokselle. (Sewak, 2019) Konvoluutiokerroksen tehtävänä on

havaita edellisen kerroksen lokaaleiden piirteiden yhteyksiä, kun taas koontikerros yhdistää semanttisesti samanlaiset piirteet. CNN-verkot ovat menestyneet erityisen hyvin konenäön sovelluksissa, koska ne ovat tehokkaita löytämään kuvista erilaisia piirteitä. (LeCun et al., 2015)



**Kuva 1.** Tavallinen ei-syvä neuroverkko (vas.) ja syvä neuroverkko (oik.)

### 3.4 Muita sovelluksia ja osa-alueita

Tietokonenäkö (Computer Vision, CV) on tieteenala, joka keskittyy sellaisten algoritmien kehittämiseen, jotka ovat erikoistuneet tunnistamaan objekteja visuaalisesta datasta. Toisin sanoen tavoitteena on ikään kuin jäljitellä ihmisen näkökykyä pyrkimällä kuvailemaan tarkastelun kohteena olevaa materiaalia analysoimalla sen visuaalisia ominaisuuksia. Tietokonenäön sovelluksia käytetään muun muassa sormenjälkitunnistuksessa, 3D-mallinnuksessa, kasvojen tunnistuksessa ja itseohjautuvissa autoissa. (Szeliski, 2011)

Luonnollisen kielen prosessointi (Natural Language Processing, NLP) tutkii keinoja saada algoritmit ”ymmärtämään” ihmisten käyttämiä eli luonnollisia kieliä. NLP:tä käytetään esimerkiksi puheentunnistuksessa, konekääntämisessä, sentimenttianalysissä ja luonnollisen kielen generoinnissa. NLP:ssä voidaan prosessoida joko tekstiä tai puhetta. Puhetta käsitellään ikään kuin tekstin meluisana versiona, josta on poistettava ylimääräinen häiriö ennen kuin sitä pystytään analysoimaan. Viime aikoina syväoppiminen sekä laskentatehon ja tallennuskapasiteetin kasvu ovat mahdollistaneet NLP-algoritmien kehittymisen täysin uudelle tasolle. (Deng & Liu, 2018)

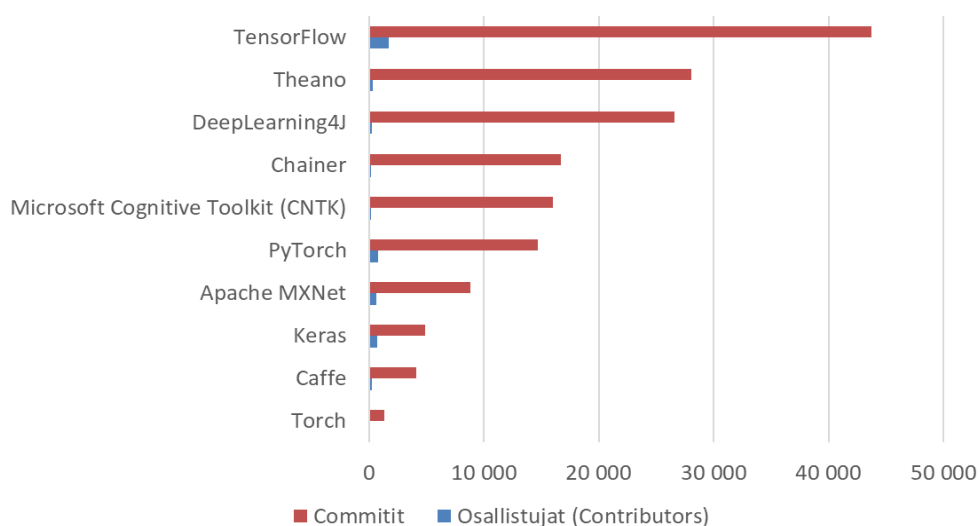
Asiantuntijajärjestelmät (expert systems) ovat ohjelmia, jotka on suunniteltu tekemään päätelmiä ja antamaan vastauksia kysymyksiin niille ohjelmoitujen sääntöjen perusteella. Päättelyprosessi tapahtuu päätöspuiden avulla, joten tulos on aina deterministinen. (Turner, 2019)

## 4 TEKÖÄLYKIRJASTOJEN JA -ALUSTOJEN MARKKINATILANNE

Tämän luvun tavoitteena on luoda katsaus tekoälykehysten ja -alustojen markkinatilanteeseen. Tarkoituksena on selvittää, mitkä ovat markkinoiden suosituimmat tekoälyratkaisut, ja tutustua tarkemmin kuhunkin ratkaisuun. Ensin käsitellään tekoälykirjastoja yleisellä tasolla, jonka jälkeen käydään läpi neljä valittua alustaa case-tutkimus-tyyppisesti.

### 4.1 Tekoälykirjastot

Tekoälykirjastot, -ohjelmistokehityspaketit ja -ohjelmistokehykset tarjoavat valmiita ratkaisuja rungoksi oman tekoälyprojektin toteuttamiselle. Ne sisältävät tavallisesti API-rajapinnan (Application Programming Interface), jonka kautta kehittäjä voi tehdä pyyntöjä kirjaston tarjoamille palveluille. Kuvassa 2 on luokiteltu vuoden 2018 lopun suosituimpia tekoälykehymiä niiden suosion GitHubissa perusteella. Näistä lähes kaikki perustuvat jollakin tavalla syväoppimiseen. Kehyksien palveluja hyödyntämällä toteutettujen syväoppimisalgoritmien avulla voidaan opettaa malleja, jotka soveltuvat erilaisiin käyttötarkoituksiin, kuten kuvantunnistukseen, NLP:hen tai päätöksentekoon.

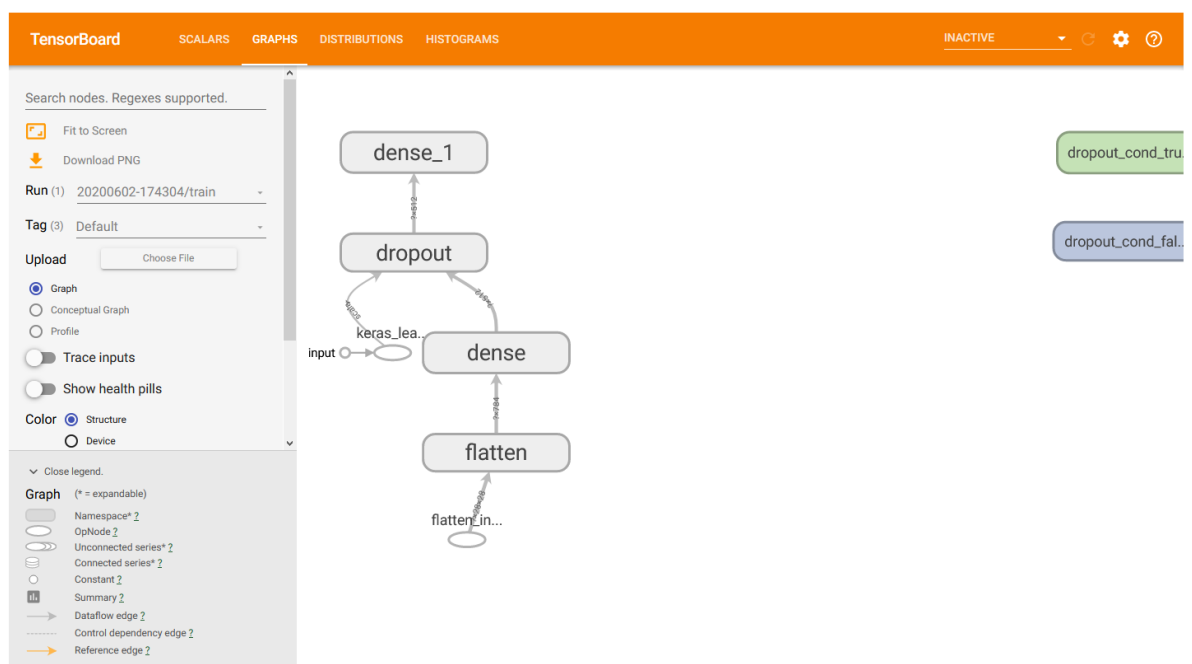


**Kuva 2.** Yksittäisten osallistujien ja committien määrä GitHubissa kirjastoittain. (Statista, 2018)

Syväoppimiskirjastoista vanhin laajalti käyttöön otettu kirjasto, vaikkakaan ei enää nykyään virallisesti tuettu, on Montrealin yliopiston koneoppimisinstituutin kehittämä Theano. Kyseessä on Python-kirjasto, joka mahdollistaa matemaattisten lausekkeiden manipuloimisen, arvioimisen ja optimoimisen tehokkaasti. (Theano, 2017) Nykypäivänä selvästi suosituin kirjasto on puolestaan Googlen Brain -tiimin TensorFlow. Sitä voisi luonnehtia jopa kokonaiseksi alustaksi, jonka keskeisimpänä osana toimivat eri tasoiset API-rajapinnat. Keras on valittu näistä viralliseksi korkeimman tason rajapinnaksi. Alusta tarjoaa myös erilaisia työkaluja, kuten TensorBoardin, joka auttaa etenkin mallien ja tulosten visualisoinnissa. Kuvassa 3 on TensorBoardin käyttöliittymä. (TensorFlow, 2020) Tämän lisäksi suurta suosiota on kerännyt Facebookin tekoälytutkimuslaboratorio FAIR:n kehittämä PyTorch, joka perustuu vanhempaan Torch-kirjastoon. Facebook on julkaissut myös kirjaston nimeltään Caffe2, joka on Kalifornian yliopistossa kehitetyn Caffe-kirjaston seuraaja. Sen mainostetaan olevan sekä modulaarinen, skaalautuva että kevyt kirjasto. Caffe2 yhdistettiin osaksi PyTorchia vuonna 2018. (Caffe2, 2018; Pytorch, 2020) Microsoft Cognitive Toolkit, joka tunnetaan myös nimellä CNTK (Computational Network ToolKit), on Microsoftin avoimen lähdekoodin syväoppimiskirjasto, jonka virallinen kehitystyö päättyi vuoden 2019 alussa (Microsoft, 2019). Apache MXNet on syväoppimiskirjasto, joka on rakennettu ensisijaisesti Pythonille. Kirjaston merkittävin moduuli on korkean tason API-rajapintana toimiva Gluon-kirjasto. (Apache MXNet, 2020) Scikit-learn on Pythonille tehty koneoppimiskirjasto, joka on rakennettu SciPy-kirjaston päälle. Scikit-learn-kirjasto tukee muun muassa luokittelua, regressiota ja klusterointia. (scikit-learn, 2020) Chainer on japanilaisen Preferred Networksin kehittämä Python-pohjainen syväoppimiskehys, jonka erityisominaisuutena on ”define-by-run” -lähestymistapa. Se eroaa perinteisemmästä ”define-and-run” -menetelmästä siten, että siinä neuroverkon rakenne määritellään dynaamisesti vasta ajon aikana sen sijaan, että se olisi jo valmiiksi määritelty. Myöhemmin myös muut kirjastot, kuten PyTorch, ovat omaksuneet kyseisen menetelmän. Tähän suurimpana syynä voitaneen pitää lähestymistavan merkittävimpana etuna mainittua ominaisuutta eli joustavuutta, joka mahdollistaa monimutkaistenkin rakenteiden luomisen suhteellisen vaivattomasti. (Chainer, 2020)

MXNetin lisäksi Apachen tekoälytarjontaan kuuluu myös Spark-klusteriohjelmistoon sisältyvä MLlib (Machine Learning Library). Sen tarkoituksena on helpottaa

koneoppimisprosessin toteuttamista hajautetuissa järjestelmissä. (Apache Spark, 2020) Eclipse Deeplearning4j on Java-kielellä kirjoitettu syväoppimiskirjasto. Deeplearning4j on yhteensopiva hajautettujen ohjelmistokehysten, kuten Hadoopin ja Apache Sparkin, kanssa, mikä mahdollistaa mallien opettamisen klusterissa. (Deeplearning4j, 2020) Mallien hajautettuun opettamiseen perustuu myös Uberin Horovod, joka on yhteensopiva niin TensorFlow'n, Kerasin, PyTorchin kuin MXNetin kanssa. Horovodin tehtävänä on tehostaa ja helpottaa hajautettujen syväoppimisprosessien toteuttamista edellä mainituilla kirjastoilla. (Horovod, 2019) XGBoost on Gradient Boosting -tekniikkaa hyödyntävä hajautettu koneoppimiskirjasto. Sen ideana on tarjota mahdollisimman nopeita, mutta samalla tarkkoja, algoritmeja koneoppimisen ongelmiin. Tehokkuutensa ansiosta XGBoost on voittanut useita koneoppimiskilpailuita ja noussut yhdeksi suosituimmista tekoälykirjastoista. (XGBoost, 2020)



**Kuva 3.** TensorBoardin käyttöliittymä

Vuonna 2017 Microsoft alkoi yhteistyössä Facebookin kanssa kehittämään avoimen lähdekoodin ONNX (Open Neural Network Exchange) -ekosysteemiä, jotta kehukset olisivat paremmin yhteensopivia toistensa kanssa. ONNX:n avulla voidaan siirtää malleja eri kehysten välillä. Tämä mahdollistaa esimerkiksi sen, että kehittäjät voivat ensin kehittää

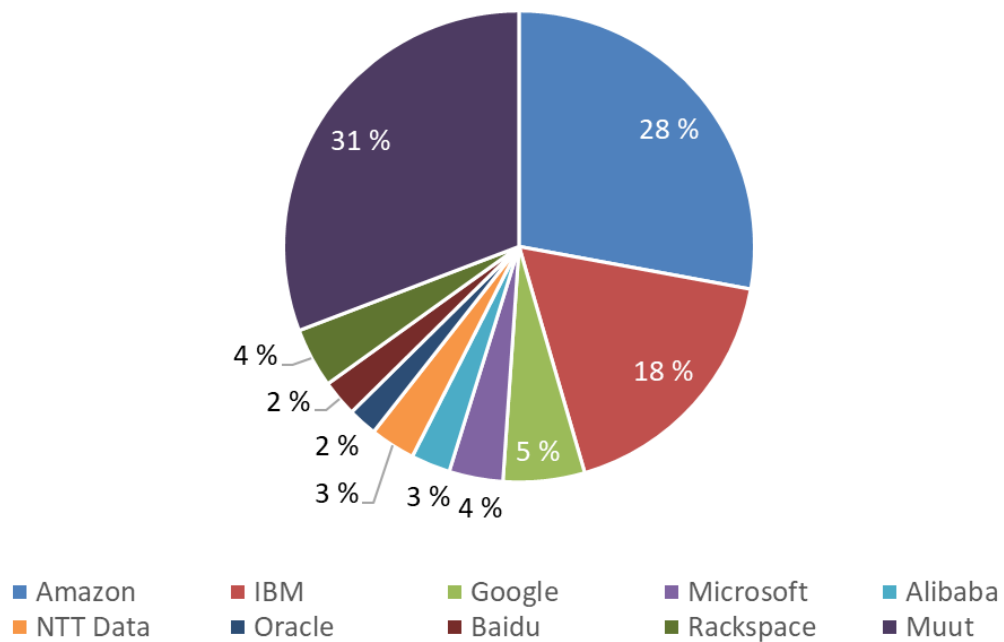


syväoppimismallinsa käyttäen yhtä kehystä ja julkaista sen myöhemmin useammille kehyksille. Julkaisun jälkeen ONNX-projektiin on liittynyt useita alan toimijoita. Tällä hetkellä jo lähestulkoon kaikki suosituimmista kehyksistä tukevatkin ONNX:ää. (ONNX, 2020) ONNX:n lisäksi vakiintuneeksi työkaluksi on muodostunut Googlella syväoppimistutkijana työskentelevän François Chaletin ylläpitämä Keras. Sen tarkoituksena on toimia ikään kuin korkean tason neuroverkkorajapintana käytettävien kehysten välillä. Vahvuuksina sillä ovat mallien prototypoinnin nopeus, modulaarisuus ja käyttäjäystävällisyys. Keras tukee markkinoiden suosituimpia kirjastoja, kuten TensorFlow'ta, CNTK:ta ja Theanoa. (Keras, 2020)

## **4.2 Tekoälyalustojen taustaa**

Tekoälyalustat koostuvat algoritmeista, API-rajapinnoista, työkaluista ja muista kehitystyötä helpottavista palveluista. Niiden perimmäinen tehtävä onkin minimoida toistavan ja manuaalisen työn määrä ja tarjota mahdollisimman kattava paketti ratkaisuja, joiden avulla tekoälyprojektien toteuttaminen olisi sujuvampaa. Laskentateho on myös verrattain kallista, varsinkin yksityishenkilöille tai pienemmän kokoluokan kehitystiimeille, jonka vuoksi tekoälyalustojen valmistajat tarjoavat yleensä laskenta- ja tallennuskapasiteettia pilvipalveluissaan. (Masood & Hashmi, 2019)

Pilvipalveluiden vallankumous on mahdollistanut myös tekoälyalustojen syntymisen. Pilvialustojen tarjoajat ovatkin tuoneet viime vuosina markkinoille myös omat tekoälyratkaisunsa muiden palvelujen rinnalle. PaaS (Platform as a Service) on yleisnimitys sellaisille pilviperustaisille palvelualustoille, joissa työkalut sekä käyttöjärjestelmä tarjotaan valmiina asiakkaalle, ja asiakkaan vastuulle jää sovellusten valitseminen. PaaS mahdollistaa kokonaisten ohjelmistojen kehittämisen pilvialustan kautta siten, että kehittäjiä ei tarvitse huolehtia laitteiston tai alustan ylläpidosta. IaaS (Infrastructure as a Service) eli infrastruktuuri palveluna toimii samalla tavalla kuin PaaS, lukuun ottamatta infrastruktuuria, jonka ylläpitäminen on siinä asiakkaan vastuulla. (Srinivasan, 2014) Kuvassa 4 on esitetty suurimpien IaaS- ja PaaS-palveluntarjoajien markkinaosuuksia.

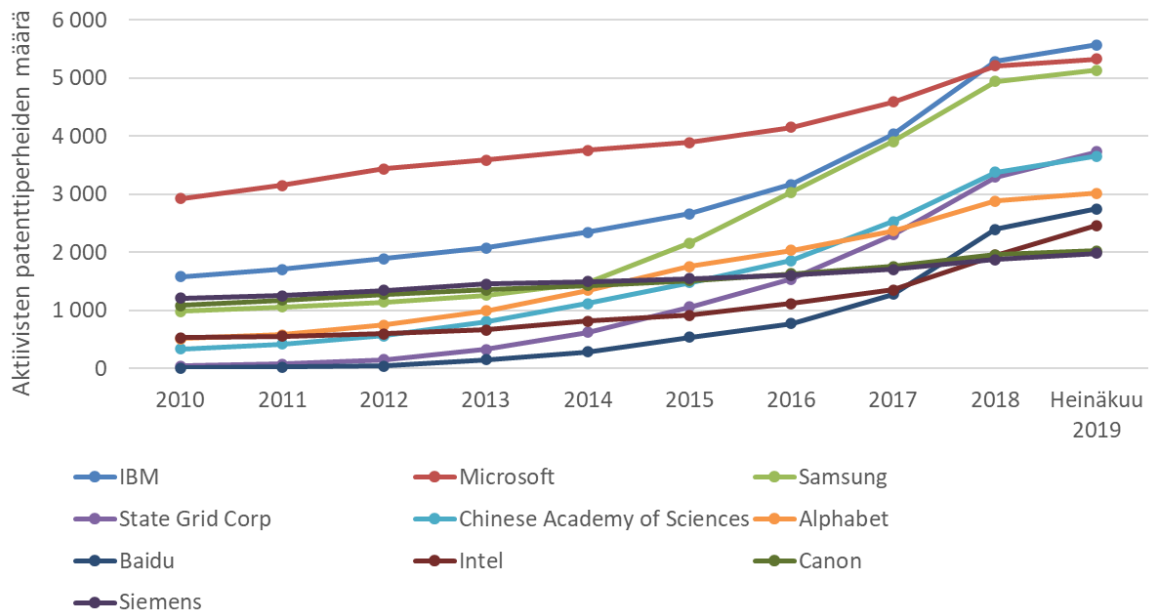


**Kuva 4.** IaaS- ja PaaS-ratkaisujen palveluntarjoajien markkinaosuudet liikevaihdon mukaan vuonna 2019 (ITCandor, 2020)

MLaaS (Machine Learning as a Service) eli koneoppiminen palveluna tarkoittaa sellaisia palveluita kuin esimerkiksi datan esiprosessointia tai valmiita malleja, jotka ovat saatavilla jossakin pilvipalvelussa. Näitä palveluja pääsee normaalisti käyttämään REST API:n kautta. MLaaS-ratkaisujen etuna on niiden käyttöönottamisen ja käyttämisen helppous, kun taas varjopuolena voi olla se, että mallit ovat liian yleistettyjä eivätkä näin ollen ole tarpeeksi erikoistuneita kyseessä olevaan ongelmaan. MLaaS:stä esimerkkejä ovat Amazon ML, Microsoft Azure ML, Google Cloud AI ja IBM Watson. (Kuznetsov, 2018)

Jotta saataisiin parempi kuva yritysten panostuksesta tekoälyyn ja koneoppimiseen, voidaan tarkastella esimerkiksi myönnettyjen patenttien määriä. Kuvassa 5 on kuvattu suurimpien patentinomistajien aktiivisten tekoälyyn liittyvien patenttien määrän kehitystä viimeisimmän vuosikymmenen aikana. Kuten kuvasta nähdään, ohjelmistoyrityksistä varsinkin IBM, Microsoft ja Alphabet (Google) omistavat runsaasti tekoälypatentteja ja ovat myös lähivuosina hankkineet näitä huomattavan määrän lisää. Tässä kohtaa on hyvä ottaa huomioon, se että patenttien määrä itsessään ei välttämättä täysin korreloi tekoälyyn panostamisen kanssa. Liiketoimintamallit voivat poiketa toisistaan merkittävästi, mikä tekee vertailemisesta haasteellista. Paremman indikaattorin puutteessa tässä työssä päätettiin

kuitenkin tarkastella suurimpia patenttien omistajia, sillä ne antavat joka tapauksessa riittävän kokonaiskuvan alan suurimmista toimijoista. Koska Amazon, IBM, Google ja Microsoft lukeutuvat tämän hetken suosituimpiin pilvipalveluiden tarjoajiin ja niiden palvelut ovat myös suhteellisen tekoälyorientoituneita, valittiin niiden tekoälyalustat tarkasteltaviksi tähän työhön.



**Kuva 5.** Maailman suurimmat tekoäly- ja koneoppimispatenttien omistajat vuodesta 2010 vuoden 2019 heinäkuuhun aktiivisten patenttiperheiden määrän mukaan. (LexisNexis PatentSight, 2019)

### 4.3 Tarkasteltavien ekosysteemien esittely

Suurimmat ohjelmistoalan toimijat ovat ryhtyneet koneoppimisbuumin myötä kovaan kilpailuun tekoälyalustojen ja -työkalujen kehityksessä. Yritykset, jotka on perinteisesti tunnettu pilvipalvelujen ja ohjelmistojen tarjoajina, ovat viimeisimpinä vuosina alkaneet investoimaan tekoälyratkaisujen kehittämiseen osana tuote- ja palvelutarjontaansa. Tällaisia yrityksiä ovat myös tässä työssä tarkasteltavien ekosysteemien tarjoajat Microsoft, Amazon, Google ja IBM.

Keskeisin osa kunkin yhtiön tekoälyekosysteemiä on yleensä vahvasti pilvipalvelun yhteyteen rakennettu koneoppimisalusta, joka tarjoaa työkalut koneoppimismallien luomiseen, opettamiseen, käyttöönottoon ja hallintaan. Vaikka alustat on suunniteltu

käytettäväksi suurimmaksi osaksi pilviympäristössä, on esimerkiksi mallien kouluttaminen usein mahdollista myös paikallisesti työpöytäympäristössä. Tästä voi olla hyötyä erityisesti, mikäli data on sensitiivistä ja se halutaan tietoturvasyistä pitää yrityksen oman infrastruktuurin sisäpuolella. Taulukossa 1 on listattu kunkin käsiteltävän yrityksen pilvipalvelut sekä ML-alustat.

**Taulukko 1.** Yritysten pilvipalvelut ja ML-alustat.

<b>Palveluntarjoaja</b>	<b>Pilvipalvelu</b>	<b>ML-alusta</b>
<b>Amazon</b>	Amazon Web Services (AWS)	Amazon SageMaker
<b>Google</b>	Google Cloud Platform (GCP)	Google AI Platform / Google Cloud Machine Learning Engine
<b>IBM</b>	IBM Cloud	IBM Watson Studio
<b>Microsoft</b>	Microsoft Azure	Azure Machine Learning

Toinen tärkeä osa yritysten tekoälytarjontaa on valmiit tekoälypalvelut, joita on mahdollista hyödyntää näiden API:en avulla. Nämä palvelut perustuvat valmiiksi opetettuihin malleihin, joita voi käyttää ilman sen erityisempää koneoppimisosaaamista. Palveluita on tarjolla muun muassa näön, puheen, kielen ja tekstinkäsittelyn sovelluksiin sekä esimerkiksi keskustelubottien luomiseen. Tekoälypalvelut yhtiöittäin on lueteltu taulukossa 2.

Lisäksi alustat tukevat yleisimpiä avoimen lähdekoodin koneoppimiskehyksiä. Taulukossa 3 on listattu alustojen tukemat kirjastot ja kehykset. Tuetulla kehyksellä tarkoitetaan tässä tapauksessa sellaista kehystä, joka on dokumentaation mukaan virallisesti tuettu alustan tarjoajan toimesta. Myös kokeellisessa vaiheessa olevat tuet on sisällytetty vertailuun.

**Taulukko 2.** Yritysten tekoälypalvelut.

Palveluntarjoaja	Näkö	Puhe, teksti ja kieli	Botit ja agentit	Muut (analyysi, haku, suositteleva jne.)
<b>Amazon</b>	Rekognition	Comprehend, Lex, Polly, Textract, Translate, Transcribe	Lex	CodeGuru, Forecast, Fraud Detector, Kendra, Personalize
<b>Google</b>	Video, Vision	Cloud Speech-to-Text, Cloud Text-to-Speech, Natural Language, Translation	Dialogflow	AutoML Tables, Cloud Inference API, Recommendations AI
<b>IBM</b>	Visual Recognition	Language Translator, Natural Language Classifier, Natural Language Understanding, Personality Insights, Speech to Text, Text to Speech, Tone Analyzer	Assistant	Discovery
<b>Microsoft</b>	Computer Vision, Custom Vision, Face, Form Recognizer, Ink Recognizer, Video Indexer	Immersive Reader, Language Understanding, Speaker Recognition, Speech to Text, Speech Translation, Text Analytics, Text to Speech, Translator Text	QnA Maker	Anomaly Detector, Bing Search API:t (Custom Search, Entity Search, Image Search jne.), Content Moderator, Personalizer

**Taulukko 3.** Ekosysteemien tuki kehysille ja kirjastoille.

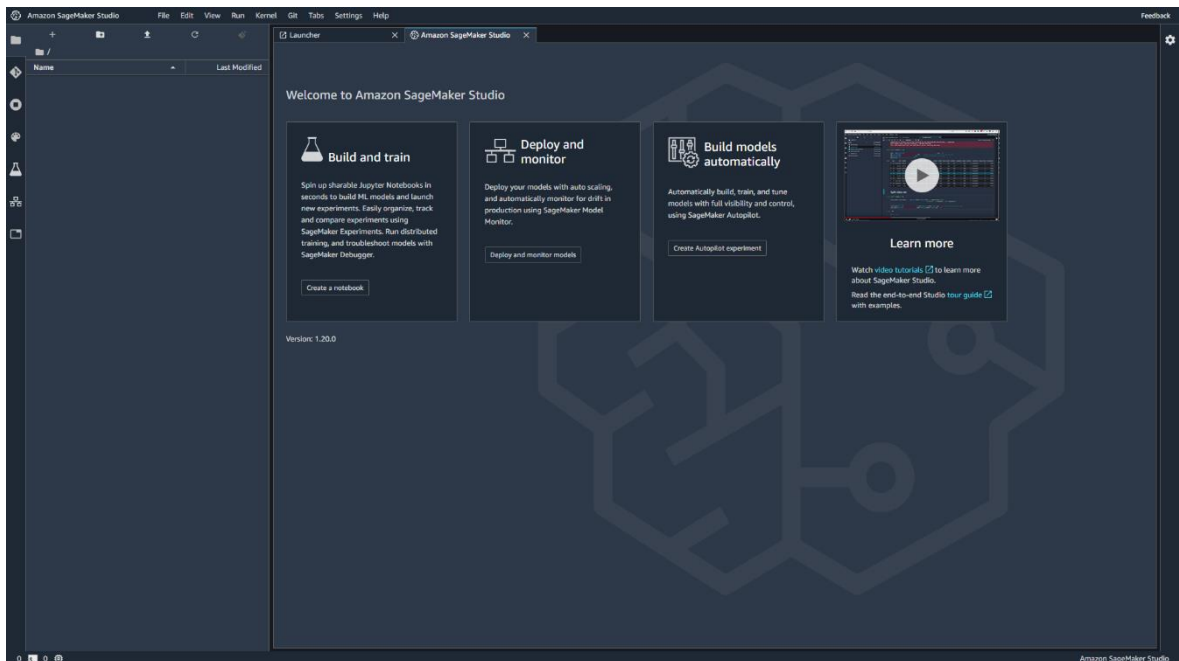
	TensorFlow	Keras	scikit-learn	PyTorch	XGBoost	Horovod	MXNet	CNTK	Chainer	Spark MLlib	Theano
<b>Amazon</b>	x	x	x	x	x	x	x	x	x	x	x
<b>Google</b>	x	x	x	x	x	x	x	x	x		
<b>IBM</b>	x	x	x	x	x	x				x	x
<b>Microsoft</b>	x	x	x	x	x	x	x	x	x	x	x

## 4.4 Amazon

Amazon.com on verkkokirjakauppana vuonna 1994 aloittanut yhdysvaltalaisyritys, joka on nopeasti kasvanut yhdeksi tämän päivän suurimmista yhtiöistä. Nykypäivänä Amazon tunnetaan verkkokauppansa lisäksi myös AWS (Amazon Web Services) -pilvipalveluiden tarjoajana. AWS johtaakin tällä hetkellä pilvimarkkinoita noin 30 %:n markkinaosuudellaan IaaS- ja PaaS-palveluista (ITCandor, 2020). Amazon on myös viime vuosien aikana alkanut investoida tekoäly- ja koneoppimiskäytännön kehittämiseen. Yhtiön tekoälytarjonta muodostuu useista erilaisista tuotteista ja palveluista aina Alexa-virtuaaliavustajasta ohjelmistoon integroitaviin palveluihin ja kehitystä helpottaviin työkaluihin asti.

Amazonin SageMaker-alusta tarjoaa työkaluja kaikkiin koneoppimiskehityksen vaiheisiin. Näistä SageMaker Ground Truth on tarkoitettu datan nimeämisen ja luokittelun automatisoimiseen. Sen mainostetaan vähentävän luokitteluun kuluja kustannuksia jopa 70%:lla. Ground Truth toimii siten, että raaka data annetaan ensin koneoppimismallille luokiteltavaksi, jonka jälkeen tulkinvarainen data lähetetään ihmisen luokiteltavaksi. Lopuksi ihmisen luokittelema data annetaan takaisen koneoppimismallin prosessoitavaksi, ja oikein luokiteltu data on valmiina käytettäväksi. SageMaker Studio on varta vasten ML-kehitystä varten luotu web-pohjainen ohjelmointiympäristö, jota voidaan käyttää mallien rakentamiseen, opettamiseen ja käyttöönottamiseen. Kuvassa 6 on SageMaker Studion oletusnäkyvä. Sen käyttöliittymä on johdettu suoraan JupyterLab-käyttöliittymästä. Vasemmassa laidassa sijaitsevat ikonit muun muassa tiedostoselaimelle, terminaalille, Git-työkaluille ja erilaisille komennoille. Näkymän oikealla puolella on työskentelyalue, joka tarjoaa mahdollisuuden avata useita välilehtiä työnkulun sujuvoittamiseksi. SageMaker Notebooks perustuu Jupyter Notebook -web-sovellukseen. Sen avulla on mahdollista luoda eräänlaisia muistioita, jotka ovat nopeasti suoritettavissa ja helposti jaettavissa. Malleja voi jakaa esimerkiksi Amazonin AWS Marketplacessa. Koneoppimisen automatisoimisessa auttaa SageMaker Autopilot, joka automaattisesti valitsee sopivan algoritmin, opettaa ja säätää sitä. Autopilot antaa myös käyttäjälle mahdollisuuden valita graafisesta käyttöliittymästä parhaiten omia tarpeita vastaavan mallin. Mallien opettamista varten Amazon tarjoaa SageMaker Experiments ja Debugger -työkalut. Experiments auttaa koulutusiteraatioiden hallinnassa tallentamalla niiden parametrit, asetukset ja tulokset omiin

tiedostoihinsa. Debugger antaa erilaisia mittareita opetusprosessin tueksi ja varoittaa mahdollisista yleisistä opetusvirheistä. SageMaker Neo, Model Monitor ja Augmented AI on tarkoitettu mallien käyttöönottovaiheeseen. Neo optimoi malleja tekemällä niistä tehokkaampia, kompaktimman kokoisia ja siirrettäviä. Model Monitorilla voi tarkkailla käyttöön otetun mallin tarkkuutta. Mikäli mallille syötettävä data poikkeaa merkittävästi opetusdatasta, ei malli välttämättä enää toimi halutulla tarkkuudella. Model Monitor pyrkii havaitsemaan tällaiset tilanteet ja selvittämään ongelmien syyt. Jos opetusdatan määrittelemiä rajoituksia on rikottu, raportoidaan käyttäjälle tuloksissa kaikista rajoitusrikkomuksista ja niiden yksityiskohdista. Augmented AI on palvelu, joka ohjaa automaattisesti mallin matalan varmuustason ennusteet ihmisen arvioitavaksi. (Amazon, 2020a)



**Kuva 6.** SageMaker Studion käyttöliittymä

Valmiita (ohjelmistoon integroitavia ja erillisiä) AI-palveluja on AWS:ssä yhteensä 12 kappaletta. Useimmille niistä on tarjolla kuukaudesta vuoteen kestävä kokeilujakso, jonka jälkeen niiden laskutus menee kulutuksen mukaan. Amazonin tarjonnasta löytyy niin tavanomaisia konenäkö- ja NLP-palveluja kuin myös erikoisempia algoritmeja, kuten Forecast ja CodeGuru. Forecast on esimerkiksi liiketoiminnan tulosten ennakkointiin tarkoitettu palvelu, joka tekee ennusteita sille syötetyn datan perusteella. CodeGurua

voidaan käyttää koodin katselmoinnin automatisointiin ja suorituskyvyn kannalta kalliiden rivien löytämiseen. (Amazon, 2020b) Amazon tarjoaa EC2-palvelussaan pilvilaskentakapasiteettia paljon suorituskykyä vaativiin syväoppimistehtäviin. Pilvessä on mahdollista suorittaa AMI (Amazon Machine Image) -instansseja, jotka tukevat suosituimpia ML-kehyskiä ja on konfiguroitu optimaalisesti parhaan suorituskyvyn saavuttamiseksi. (Amazon, 2020c)

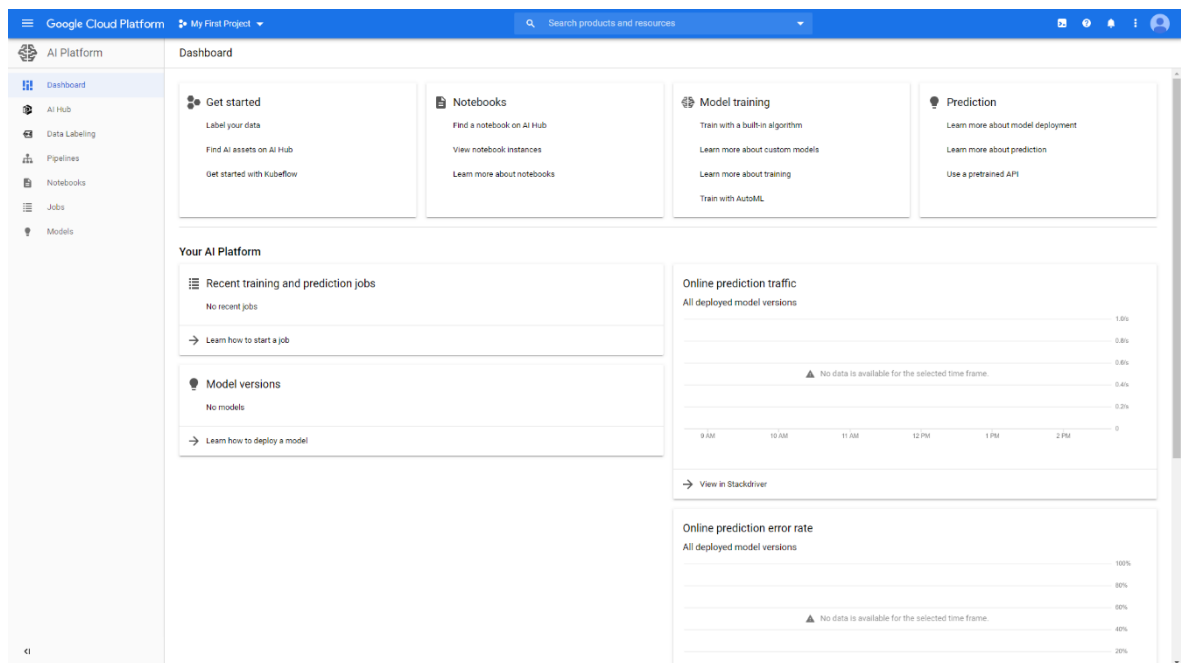
## 4.5 Google

Google on vuonna 1998 Yhdysvalloissa perustettu ja hakukoneestaan tunnettu yhtiö, joka on vuosien myötä tehnyt paljon uusia aluevaltauksia niin ohjelmisto- kuin laitteistopuolella. Tällä hetkellä Google kehittää maailman suosituinta mobiilikäyttöjärjestelmää Androidia sekä käytetyintä verkkoselainta Chromea, tarjoaa erilaisia palveluja sähköpostista GCP (Google Cloud Platform) -pilvipalveluihin ja on tuonut markkinoille laitteita kuten Chromecastin tai Chromebookin. Tekoälysektorilla Google omistaa Lontoolaisen DeepMindin ja kehittää suosittua TensorFlow-kirjastoa.

Googlen tekoälykehitystyökalut ovat saatavilla AI Platform -alustalla, jonka käyttöliittymästä on kuvakaappaus kuvassa 7. Vasemmasta laidasta löytyy erilaisia työkaluja sisältävä valikko. Oikealla näkyy yleiskatsaus, jossa on esiteltynä muun muassa mallien toimintaan liittyvää informaatiota. Datan luokitteluun on tarjolla Data Labeling Service, johon lähetetään luokiteltava data, kaikki mahdolliset nimikkeet sekä ohjeet dataa luokitteleville henkilöille. Kun data on luokiteltu, se lähetetään takaisin, jonka jälkeen sitä voidaan käyttää mallien opettamisessa. ML-sovelluksia on mahdollista rakentaa integroidussa JupyterLab-ympäristössä Jupyter Notebook -pohjaisen AI Platform Notebooks -palvelun avulla. Deep Learning VM Image tarjoaa nopean tavan suorittaa suosituimmille kehyksille valmiiksi konfiguroituja virtuaalikoneinstansseja. Mallien opettamiseen pilviympäristössä on olemassa AI Platform Training, joka tukee natiivisti TensorFlow, scikit-learn ja XGBoost -kirjastoja. Myös muita kirjastoja on mahdollista käyttää mukautettujen konttien avulla. AI Platform Prediction -palvelun avulla voi suorittaa valmiita malleja pilvessä. Vastaavasti mallien opettamisen ja käyttöönoton paikallisesti Kubernetes-alustalla mahdollistaa avoimen lähdekoodin Kubeflow. Koneoppimisella



saatujen tulosten ymmärtämistä pyrkii helpottamaan tällä hetkellä vielä beetavaiheessa oleva Explainable AI, joka antaa erilaisia mittareita mallin tekemien ennusteiden selittämiseksi. AI Hub on kaikenlaisten AI-komponenttien jakamiseen tarkoitettu palvelu. Sen avulla voi jakaa resursseja joko yrityksen sisäisesti tai hyödyntää Googlen kehittämää ja julkaisemaa sisältöä. Kubeflow'n, AI Hubin ja Notebooksin etuna on se, että ne eivät itsessään maksa käyttäjälle mitään, vaan ainoat kustannukset tulevat pilviresurssien käyttämisestä. (Google, 2020a) Esimerkiksi virtuaalikone neljällä virtuaalisella suorittimella ja 16 gigatavun keskusmuistilla maksaa noin 0,05-0,20 \$ tunnilta ja 30-140 \$ kuulta riippuen resurssien sijainnista ja virtuaalikoneen tyypistä (Google, 2020b).



**Kuva 7.** AI Platformin yleiskatsausnäkyvä

Google tarjoaa kahden tyyppisiä valmiita tekoälykomponentteja: Cloud AutoML -tuotteita ja esiopetettuja API-rajapintoja. AutoML-palvelut mahdollistavat omavalintaisten mallien luomisen pilviympäristössä ilman sen suurempaa ML-osaamista. Tekoälypalvelut on Googlessa jaoteltu näkö-, kieli-, puhe- ja jäsenneilyjen tietojen kategorioihin. Näköpalveluihin kuuluvat algoritmit löytävät kuvista ja videoista yksityiskohtia. Kielelliset palvelut tunnistavat kieliä ja kykenevät kääntämään niitä. Puhepalveluista löytyvät keskustelukäyttöliittymien rakentamiseen soveltuva Dialogflow sekä tekstistä puheeksi ja

puheesta tekstiksi -palvelut. Jäsenneltyjen tietojen palveluihin kuuluu esimerkiksi suosittelualgoritmi. (Google, 2020c)

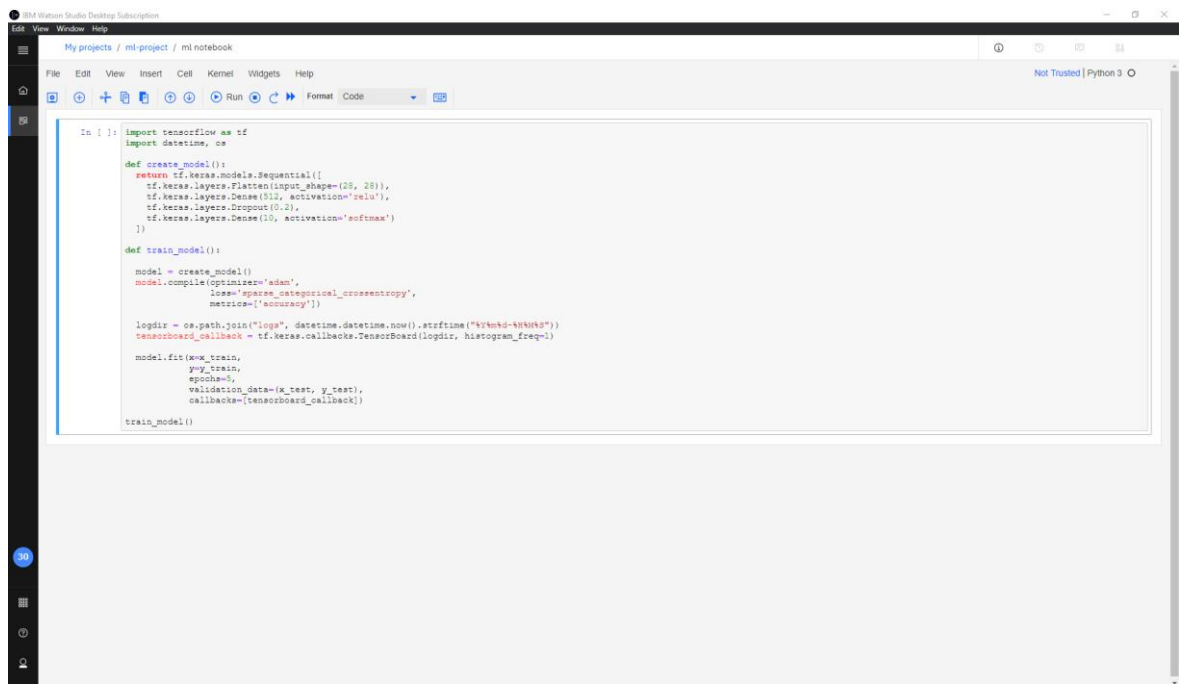
## 4.6 IBM

IBM (International Business Machines Corporation) (alun perin Computing-Tabulating-Recording Company, CTR) perustettiin New Yorkissa vuonna 1911. Yritys on vuosien varrella kehittänyt huomattavan määrän uusia innovaatioita ja sen portfolio on historiallisesti ollut hyvin laaja. Tällä hetkellä IBM:n liiketoiminta koostuu pääasiassa kognitiivisista ratkaisuista, teknologiapalveluista ja pilvialustoista, yrityspalveluista sekä järjestelmistä (IBM, 2020c). Watson on IBM:n tekoälyratkaisujen keskeisin tuote ja se on suunnattu erityisesti yrityksille.

Watson Studio on IBM:n työkalu mallien rakentamiseen ja opettamiseen. Kuvassa 8 on näkyvillä Watson Studion työpöytäversion käyttöliittymä Notebooks-tilassa. Tämän lisäksi käyttöliittymä tarjoaa muun muassa mittareita mallien suoriutumisesta eikä kaikkea tarvitse välttämättä tehdä kuvan mukaisesti koodia kirjoittamalla. Yksi Watson Studion sisältämistä ominaisuuksista on AutoAI, joka on tarkoitettu ML-mallien rakentamiseen ja käyttöönottamisen automatisoimiseen. Se muun muassa esiprosessoi syötedatan ja valitsee automaattisesti sopivimman mallin. Mallien visuaaliseen valmisteleminen ja rakentamiseen Watson Studio tarjoaa SPSS Modeler flows -ominaisuuden. Sen avulla voidaan luoda liiketoimintaa hyödyttäviä ennustemalleja käyttämällä olemassa olevaa dataa. Käyttöönottamiseen ja suorittamiseen on tarjolla Watson Studioon integroitu Watson Machine Learning, joka antaa työkalut mallien dynaamiseen uudelleenopettamiseen, API-rajapintojen automaattiseen generoimiseen ja sisältää Watson OpenScale-työkalun integroituna. OpenScalella on mahdollista hallita ja monitoroida malleja arvioimalla niiden tuloksia sekä esimerkiksi vähentää mahdollisia vääristymiä. (IBM, 2020b)

Watsonin API:t kattavat kuusi eri osa-aluetta, jotka ovat kuvantunnistus, tekstistä puheeksi, kielten kääntäminen, kielellinen luokittelu sekä persoonallisuuden ja emootioiden analysointi. Näistä palveluista erityisesti tunteiden ja persoonallisuuden analysointiin tarkoitettut Tone Analyzer ja Personality Insights erottuvat muiden palveluntarjoajien

tarjonnasta. Tone Analyzer käyttää kielellistä analyysiä havaitakseen emotionaalisia sävyjä tekstistä. Teknologiaa voidaan hyödyntää esimerkiksi asiakaspalvelussa tai keskustelubottien rakentamisessa. Personality Insights pyrkii vastaavasti arvioimaan luonteenpiirteitä, tarpeita, arvoja ja kulutustottumuksia kirjoitetun tekstin perusteella, mistä voi olla hyötyä varsinkin yksilöllisen asiakaskokemuksen luomisessa. Ohjelmistoon integroitavien palvelujen lisäksi IBM:ltä löytyy myös valmiita tekoälyä hyödyntäviä työkaluja, joiden tarkoituksena on tarjota tekoälyn edut, vaikka osaamista ei koneoppimisesta sen enempää olisikaan. Esimerkkeinä tällaisista työkaluista ovat Watson Assistant, jonka avulla voi rakentaa keskustelurajapintoja ja Watson Discovery, jota voidaan käyttää yritysdatan hakuun ja analysointiin. (IBM, 2020a)



```
In [ ]: import tensorflow as tf
import datetime, os

def create_model():
    return tf.keras.models.Sequential([
        tf.keras.layers.Flatten(input_shape=(28, 28)),
        tf.keras.layers.Dense(512, activation='relu'),
        tf.keras.layers.Dropout(0.2),
        tf.keras.layers.Dense(10, activation='softmax')
    ])

def train_model():
    model = create_model()
    model.compile(optimizer='adam',
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])

    logdir = os.path.join("logs", datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
    tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir, histogram_freq=1)

    model.fit(x=x_train,
            y=y_train,
            epochs=1,
            validation_data=(x_test, y_test),
            callbacks=[tensorboard_callback])

train_model()
```

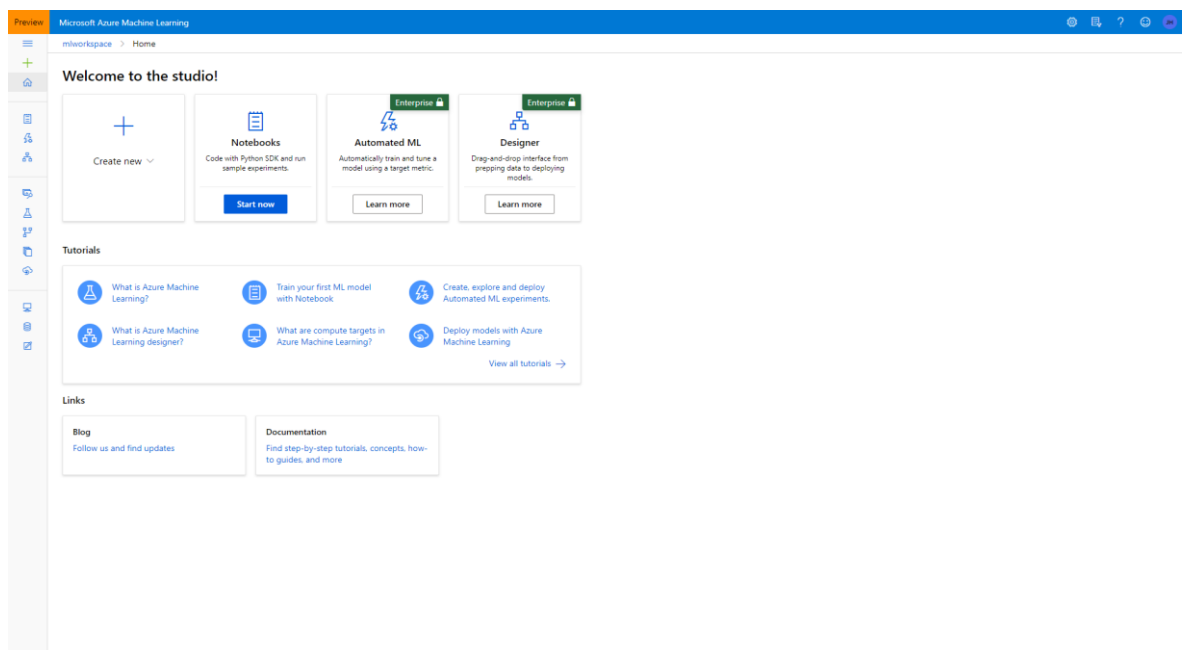
**Kuva 8.** IBM Watson Studion Notebook-integraatio

## 4.7 Microsoft

Vuonna 1975 perustettu Microsoft nousi 1980-luvulla MS-DOSin ja myöhemmin Windows-käyttöjärjestelmänsä ansiosta yhdeksi maailman suurimmista teknologiayhtiöistä. Yrityksen muihin tunnettuihin tuotteisiin ja palveluihin lukeutuvat muun muassa Xbox-pelikonsolit, Office-toimisto-ohjelmistot ja Edge-selain. Viime vuosina Microsoft on alkanut painottamaan strategiassaan yhä enemmän pilvipalveluita ja yrityksille suunnattuja

ratkaisuja. Yhtiön Azure-pilvipalvelu onkin pilvimarkkinoiden toiseksi suosituin alusta. Microsoft on investoinut paljon myös tekoälyyn ja avoimen lähdekoodin ohjelmistoihin, mistä hyvänä esimerkkinä voidaan pitää ONNX-projektia.

Azure Machine Learning on työkalu, joka on tarkoitettu mallien rakentamiseen, opettamiseen ja käyttöön ottamiseen. ML-kehitykseen se sisältää lähtökohtaisesti kaksi eri lähestymistapaa: code-first-menetelmän ja raahaa ja pudota -työkalun. Code-first-ominaisuus tarjoaa Jupyter notebooks -sovelluksen integroituna. Azure Machine Learning mahdollistaa iteratiivisten vaiheiden automatisoimisen, mallien hienosäätämisen ja datan visualisoinnin helposti graafisen käyttöliittymän kautta. Automaattiset yhteenvedot auttavat ymmärtämään mallien toimintaperiaatteita ja valitsemaan parhaiten sopivan mallin. MLOps eli koneoppimisen DevOps on kehitetty sujuvoittamaan koneoppimiselämänkaaren hallintaa tekemällä datatieteilijöiden ja IT-tiimien yhteistyöstä tiiviimpää. Tähän tarkoitukseen löytyy Azure Machine Learningista ratkaisuja muun muassa jatkuvan integraation ja toimituksen työkalujen muodossa. Kuvassa 9 on Azure Machine Learningin yleisnäkymä. Vasemmalla näkyvästä valikosta käyttäjä voi valita esimerkiksi AutoML-ominaisuuden, visuaalisen käyttöliittymän mallien rakentamiseen tai Notebooks-editorin. Lisäksi valikosta pääsee muun muassa tarkastelemaan dataa, malleja ja resursseja. (Microsoft, 2020b)



**Kuva 9.** Azure Machine Learning, käyttöliittymä

Azure Cognitive Services -tekoälypalvelut on lajiteltu kategorioihin päätöksenteko, kieli, puhe, näkö ja web-haku. Hakupalvelut koostuvat kymmenestä erilaisesta Bing-API:sta, joihin kuuluvat esimerkiksi visuaalinen, uutis-, kuva- ja videohaku, oikeinkirjoituksen tarkistus sekä autosuggest-algoritmi. (Microsoft, 2020c) Azure Bot Service on bottien kehittämiseen tarkoitettu palvelu. Se tarjoaa muun muassa avoimen lähdekoodin työkaluja, mahdollisuuden integroida Cognitive Services -palveluita botteihin ja ottaa botteja käyttöön suosituimmilla kanavilla. (Microsoft, 2020a)

## 5 TULOKSET, POHDINTA JA TULEVAISUUS

Tässä luvussa käsitellään kartoitusvaiheessa saatuja tuloksia ja tehdään niiden perusteella tarvittavat johtopäätökset. Lisäksi pohditaan tekoälyn ja koneoppimisen tulevaisuudennäkymiä.

### 5.1 Tulokset

Ensimmäiseen tutkimuskysymykseen liittyen eli ”Kuinka paljon kullekin alustalle on tällä hetkellä tarjolla erilaisia tekoälyä hyödyntäviä ohjelmistoon integroitavia ratkaisuja?” voidaan todeta, että ratkaisuja alkaa olla markkinoilla jo huomattavasti ja niiden määrä kasvaa koko ajan. Esimerkiksi Deep Learning -sivuston listauksessa (2017) on lueteltu yhteensä 46 erilaista ML-kirjastoa. Tämä on seurausta niin suurten ohjelmistoyritysten tekemistä investoinneista kuin avoimen lähdekoodin yhteisöjen aktiivisuudesta tekoälyratkaisujen kehittämisessä. Kuten pilvialustojen kartoituksessa voitiin huomata, ovat useimmat suurimmista PaaS-palveluiden tarjoajista (esimerkkeinä Microsoft, IBM, Google, Amazon, Oracle, Alibaba ja Baidu) lähteneet kehittämään omia tekoälypalvelujaan ja -työkalujaan osana omia alustojansa.

Toinen tutkimuskysymys koski ratkaisujen laajuutta. Tähän vastatessa voidaan ratkaisut jaotella kahteen eri luokkaan ja pohtia tilannetta kummankin kohdalla erikseen. Ensimmäiseen luokkaan kuuluvat avoimen lähdekoodin kirjastot. Näistä esimerkiksi TensorFlow on keskimääräistä laajempi tarjotessaan tuen hajautetulle opettamiselle, integraatiomahdollisuuden mobiili- ja web-sovelluksiin sekä kokonaisen ML-elämänkaaren kattavan alustan. Toiset kirjastot voivat sen sijaan olla keskittyneitä ainoastaan johonkin tiettyyn käyttötarkoitukseen, kuten korkean tason DL-rajapinnoiksi tarkoitettut Keras ja Gluon. Toisaalta on aika selvää, että ei ole olemassakaan sellaista kirjastoa, joka sopisi kaikkiin mahdollisiin skenaarioihin, minkä takia kirjaston valintaa kannattaakin harkita joka tilanteessa erikseen. Toinen luokka on pilvialustojen tarjoajien tekoälyekosysteemit. Näiden tarjonnassa voidaan nähdä selkeitä trendejä työkalujen ja integroitavien palveluiden suhteen. Kaikki neljästä tarkastellusta yhtiöstä (Amazon, Google, IBM, Microsoft) ovat tuoneet markkinoille omat alustansa, jotka tarjoavat automatisoituja ja kehitystyötä helpottavia

ratkaisuja koneoppimismallien luomisen jokaiseen vaiheeseen. Lisäksi on voitu havaita, että jokainen ekosysteemi sisältää ainakin yhden palvelun niin konenäköön, NLP:hen kuin bottien kehittämiseen. Hakuun tarkoitettujen palvelujen lisäksi on myös suhteellisen yleisiä. Näistä havainnoista voidaan tehdä sellainen johtopäätös, että ainakin tarkasteltujen ekosysteemien perusteella markkinoilta löytyy jo vähintäänkin kymmeniä valmiita tekoälyä hyödyntäviä ratkaisuja.

Kolmannessa tutkimuskysymyksessä haluttiin saada vastaus dokumentaation kattavuuteen ja ratkaisujen integroitavuuden helppouteen. Dokumentaatioon liittyen tulokset vastasivat hyvin pitkälti odotuksia. Niin avoimen lähdekoodin kirjastoihin kuin maksullisiin API-rajapintoihin on useimpiin olemassa kattava dokumentaatio, joskin avoimen lähdekoodin ratkaisuihin löytyy luonnollisesti enemmän materiaalia niiden vakiintuneen aseman, suosion ja avoimuuden ansiosta. Yksi esimerkki tällaisesta perusteellisesta dokumentoinnista on CNTK, jonka dokumentaatio (Microsoft Docs, 2017) on selkeät kuvaukset kustakin ominaisuudesta ja oppaita erilaisiin käyttötapauksiin. Yksi integroimista helpottava asia on AutoML-palvelut, jotka auttavat koneoppimisen automatisoinnissa. Automaatiosta voi olla erityisesti hyötyä, mikäli esimerkiksi jokin taho ilman ML-osaamista haluaa hyödyntää koneoppimisteknologiaa parantaakseen yrityksensä tuottavuutta. Lisäksi AutoML nopeuttaa prosessia kokonaisuudessaan minimoimalla tarvittavan manuaalisen työn määrän ja parhaassa tapauksessa valitsee mallit ihmistä paremmin. ONNX ja Keras ovat myös hyviä esimerkkejä integroimisen helppouteen ja joustavuuteen positiivisesti vaikuttavista ratkaisuista. ONNX-standardi parantaa mallien siirrettävyyttä yhtenäisen formaatin avulla ja Keras tekee DL-kehityksestä käyttäjäystävällistä tarjoamalla modulaarisen ympäristön mallien rakentamiseen.

## **5.2 Pohdinta ja tulevaisuus**

Yksi mielenkiintoisimmista työn suorittamisen yhteydessä havaituista ilmiöistä oli se, miten suuret yhtiöt ovat alkaneet panostaa avoimeen lähdekoodiin. Parhaiten tämä on huomattavissa ONNX-formaatin tapauksessa, jota tukeekin jo kymmenet eri teknologiayhtiöt. Sille, että miksi Microsoft ja Facebook päättivät tällaiseen projektiin alun perin ryhtyä, voi olla monta eri syytä. Tärkeimpiin syihin kuuluu varmasti se, että avoimen

lähdekoodin ratkaisut tulevat tulevaisuudessa mitä luultavimmin saavuttamaan yhä enemmän suosiota. Kun nämä yhtiöt jo aikaisessa vaiheessa liittyvät mukaan tähän trendiin ja mahdollisesti saavat vakiinnutettua asemansa osana yhteisöä, voivat ne mahdollisesti myöhemmin hyötyä siitä monin eri tavoin. Lisäksi kun malleista tulee paremmin siirrettäviä kehysten välillä, pääsevät yritykset itse hyödyntämään laajempaa skaalaa erilaisia malleja.

Tulevaisuudessa tekoälyratkaisujen tarjonta ja varsinkin kysyntä tulevat varmasti kasvamaan samalla kun investoinnit AI-teknoologiaan lisääntyvät. Vuoden 2010 ja 2018 välillä sijoitukset tekoäly-startup-yrityksiin ovatkin kasvaneet maailmanlaajuisesti keskimäärin 48 %:n vuosivauhtia (Perrault et al., 2019). Tekoälyratkaisuissa varsinkin avoin lähdekoodi tulee olemaan tärkeässä osassa. Tämän takia onkin tärkeää, että avoimen lähdekoodin tärkeimmät perusajatukset, eli avoin yhteistyö, standardit ja parempi yhteentoimivuus sekä informaation saatavuus, eivät pääse unohtumaan eikä pieni ryhmä suurimpia yhtiöitä pääse saavuttamaan liian suurta valta-asemaa näillä markkinoilla (Haff, 2018).



## 6 YHTEENVETO

Tämän työn tavoitteena oli tarkastella tarjolla olevia tekoälyratkaisuja ja luoda katsaus niiden integroitavuuteen ja laajuuteen tämän hetken markkinatilanteen perusteella. Lisäksi perehdyttiin alan kirjallisuuteen ja tutustuttiin tärkeimpiin käsitteisiin. Työssä käytiin läpi suosituimmat avoimen lähdekoodin kirjastot ja tutustuttiin näiden käyttömahdollisuuksiin. Kaupallisista tekoälypalvelujen tarjoajista valittiin pilvialustan suosion perusteella neljä yhtiötä, jotka olivat Amazon, Google, IBM ja Microsoft. Näistä jokaisen yhtiön tekoälyekosysteemistä tehtiin case-tutkimus-tyyppinen selvitys.

Ratkaisuja tutkittaessa selvisi, että eräät kehykset, kuten TensorFlow ja Keras, ovat saavuttaneet vakiintuneen aseman markkinoilla ja niiden taustalla on laaja ja omistautunut yhteisö (Masood & Hashmi, 2019). Kaupallisissa ratkaisuissa tiettyjä toistuvia trendejä, joista esimerkkeinä voidaan mainita pilvipalvelujen ympärille rakennetut ML-kehitystyökalut ja API-rajapinnan tarjoavat palvelut. Palveluissa suosittuja sovelluskohteita havaittiin olevan erityisesti tietokonenäkö, NLP ja botit, sillä näillä osa-alueilla tarkastelluilla palveluntarjoajilla on olemassa eniten ratkaisuja.

Työssä saatujen tulosten perusteella voidaan todeta, että valmiiden tekoälyratkaisujen integrointiin on olemassa kattava valikoima työkaluja niin tekoälyasiantuntijalle kuin vähemmän aiheeseen perehtyneellekin henkilölle. On toki pidettävä mielessä, että automatisoidut ratkaisut eivät usein ole ilmaisia eikä niiden tekemisiin päätöksiin välttämättä kannata ainakaan vielä luottaa yhtä paljon kuin ihmiseen. Automatisoitua koneoppimista voidaan kuitenkin pitää monille yrityksille mahdollisena vaihtoehtona. Seuraavina vuosina ratkaisujen määrä tulee varmasti vielä kasvamaan ja näiden laatu paranemaan yritysten ja yksityishenkilöiden hyödyntäessä tekoälyä enenevässä määrin erilaisten ongelmien ratkaisemiseen.

## LÄHTEET

- Amazon. (2020a). Amazon SageMaker. Amazon Web Services, Inc. [verkkodokumentti]. [Viitattu: 20.5.2020]. Saatavissa: <https://aws.amazon.com/sagemaker/>
- Amazon. (2020b). Artificial Intelligence Services. Amazon Web Services, Inc. [verkkodokumentti]. [Viitattu: 21.5.2020]. Saatavissa: <https://aws.amazon.com/machine-learning/ai-services/>
- Amazon. (2020c). Machine Learning on AWS. Amazon Web Services, Inc. [verkkodokumentti]. [Viitattu: 15.4.2020]. Saatavissa: <https://aws.amazon.com/machine-learning/>
- Apache MXNet. (2020). Apache MXNet. Apache MXNet. [verkkodokumentti]. [Viitattu: 12.5.2020]. Saatavissa: <https://mxnet.apache.org/>
- Apache Spark. (2020). MLlib | Apache Spark. [verkkodokumentti]. [Viitattu: 12.5.2020]. Saatavissa: <https://spark.apache.org/mllib/>
- Caffe2. (2018). Caffe2 and PyTorch join forces to create a Research + Production platform PyTorch 1.0. Caffe2. [verkkodokumentti]. [Viitattu: 8.6.2020]. Saatavissa: [http://caffe2.ai/blog/2018/05/02/Caffe2\\_PyTorch\\_1\\_0.html/](http://caffe2.ai/blog/2018/05/02/Caffe2_PyTorch_1_0.html/)
- Chainer. (2020). Chainer: A flexible framework for neural networks. Chainer. [verkkodokumentti]. [Viitattu: 14.5.2020]. Saatavissa: <https://chainer.org/>
- Deep Learning. (2017). Software links « Deep Learning. [verkkodokumentti]. [Viitattu: 3.6.2020]. Saatavissa: [http://deeplearning.net/software\\_links/](http://deeplearning.net/software_links/)
- Deeplearning4j. (2020). Deeplearning4j. [verkkodokumentti]. [Viitattu: 13.5.2020]. Saatavissa: <https://deeplearning4j.org/>
- Deng, L., & Liu, Y. (Eds.). (2018). Deep Learning in Natural Language Processing. Springer Singapore. <https://doi.org/10.1007/978-981-10-5209-5>
- Ertel, W. (2011). Introduction to Artificial Intelligence. Springer London. <https://doi.org/10.1007/978-0-85729-299-5>

- Google. (2020a). AI Platform. Google Cloud. [verkkodokumentti]. [Viitattu: 23.5.2020]. Saatavissa: <https://cloud.google.com/ai-platform?hl=fi>
- Google. (2020b). All Pricing | Compute Engine Documentation. [verkkodokumentti]. [Viitattu: 9.6.2020]. Saatavissa: <https://cloud.google.com/compute/all-pricing/>
- Google. (2020c). Cloud AI Building Blocks. Google Cloud. [verkkodokumentti]. [Viitattu: 24.5.2020]. Saatavissa: <https://cloud.google.com/products/ai/building-blocks?hl=fi>
- Haff, Gordon. (2018). How Open Source Ate Software: Understand the Open Source Movement and So Much More. Berkeley, CA: Apress. <https://doi.org/10.1007/978-1-4842-3894-3>.
- Horovod. (2019). Horovod documentation — Horovod documentation. [verkkodokumentti]. [Viitattu: 16.5.2020]. Saatavissa: <https://horovod.readthedocs.io/en/latest/>
- IBM. (2020a). IBM Watson products and solutions. [verkkodokumentti]. [Viitattu: 25.5.2020]. Saatavissa: <https://www.ibm.com/watson/products-services>
- IBM. (2020b). Watson Studio - Overview. [verkkodokumentti]. [Viitattu: 24.5.2020]. Saatavissa: <https://www.ibm.com/cloud/watson-studio>
- IBM. (2020c, March 12). 2019 IBM Annual Report. [verkkodokumentti]. [Viitattu: 24.5.2020]. Saatavissa: <https://www.ibm.com/annualreport/index.html>
- ITCandor. (2020). Global IaaS/PaaS market share by vendor 2019. Statista. [verkkodokumentti]. [Viitattu: 10.4.2020]. Saatavissa: <http://www.statista.com/statistics/1052864/worldwide-iaas-paas-vendor-share/>
- Keras. (2020). Keras: the Python deep learning API. [verkkodokumentti]. [Viitattu: 18.5.2020]. Saatavissa: <https://keras.io/>
- Kinnunen, S.-K., Happonen, A., Marttonen-Arola, S., & Kärri, T. (2019). Traditional and extended fleets in literature and practice: Definition and untapped potential. *International Journal of Strategic Engineering Asset Management (IJSEAM)*, 20.

- Kortelainen, H., Happonen, A., & Hanski, J. (2019). From Asset Provider to Knowledge Company—Transformation in the Digital Era. In J. Mathew, C. W. Lim, L. Ma, D. Sands, M. E. Cholette, & P. Borghesani (Eds.), *Asset Intelligence through Integration and Interoperability and Contemporary Vibration Engineering Technologies* (pp. 333–341). Springer International Publishing. [https://doi.org/10.1007/978-3-319-95711-1\\_33](https://doi.org/10.1007/978-3-319-95711-1_33)
- Kortelainen, H., Happonen, A., & Kinnunen, S.-K. (2016). Fleet Service Generation—Challenges in Corporate Asset Management. In K. T. Koskinen, H. Kortelainen, J. Aaltonen, T. Uusitalo, K. Komonen, J. Mathew, & J. Laitinen (Eds.), *Proceedings of the 10th World Congress on Engineering Asset Management (WCEAM 2015)* (pp. 373–380). Springer International Publishing. [https://doi.org/10.1007/978-3-319-27064-7\\_35](https://doi.org/10.1007/978-3-319-27064-7_35)
- Kuznetsov, V. (2018). Machine Learning as a Service for HEP. ArXiv:1811.04492 [Hep-Ex, Physics:Physics]. <http://arxiv.org/abs/1811.04492>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- LexisNexis PatentSight. (2019, July). Leading owners of machine learning & AI patents 2010-2019. Statista. [verkkodokumentti]. [Viitattu: 4.5.2020]. Saatavissa: <http://www.statista.com/statistics/1032627/worldwide-machine-learning-and-ai-patent-owners-trend/>
- Li, D., & Du, Y. (2007). *Artificial intelligence with uncertainty*. CRC press.
- Louridas, P., & Ebert, C. (2016). Machine Learning. *IEEE Software*, 33(5), 110–115.
- Masood, A., & Hashmi, A. (2019). *Cognitive Computing Recipes: Artificial Intelligence Solutions Using Microsoft Cognitive Services and TensorFlow*. Apress. <https://doi.org/10.1007/978-1-4842-4106-6>
- Microsoft. (2019). microsoft/CNTK [C++]. Microsoft. [verkkodokumentti]. [Viitattu: 12.5.2020]. Saatavissa: <https://github.com/microsoft/CNTK>
- Microsoft. (2020a). Azure Bot Service | Microsoft Azure. [verkkodokumentti]. [Viitattu: 26.5.2020]. Saatavissa: <https://azure.microsoft.com/en-us/services/bot-service/>

Microsoft. (2020b). Azure Machine Learning | Microsoft Azure. [verkkodokumentti]. [Viitattu: 26.5.2020]. Saatavissa: <https://azure.microsoft.com/en-us/services/machine-learning/>

Microsoft. (2020c). Cognitive Services—APIs for AI Developers | Microsoft Azure. [verkkodokumentti]. [Viitattu: 26.5.2020]. Saatavissa: <https://azure.microsoft.com/en-us/services/cognitive-services/>

Microsoft Docs. (2017). The Microsoft Cognitive Toolkit - Cognitive Toolkit - CNTK. [verkkodokumentti]. [Viitattu: 3.6.2020]. Saatavissa: <https://docs.microsoft.com/en-us/cognitive-toolkit/>

ONNX. (2020). ONNX | Home. [verkkodokumentti]. [Viitattu: 18.5.2020]. Saatavissa: <https://onnx.ai/index.html>

Perrault, R., Shoham, Y., Brynjolfsson, E., Clark, J., Etchemendy, J., Grosz, B., Lyons, T., Manyika, J., Mishra, S., & Niebles, J. C. (2019, December). The AI Index 2019 Annual Report. Stanford HAI. <https://hai.stanford.edu/research/ai-index-2019>

PyTorch. (2020). PyTorch. [verkkodokumentti]. [Viitattu: 12.5.2020]. Saatavissa: <https://www.pytorch.org>

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>

scikit-learn. (2020). scikit-learn: machine learning in Python — scikit-learn 0.22.2 documentation. [verkkodokumentti]. [Viitattu: 12.5.2020]. Saatavissa: <https://scikit-learn.org/stable/>

Sewak, M. (2019). *Deep Reinforcement Learning: Frontiers of Artificial Intelligence*. Springer Singapore. <https://doi.org/10.1007/978-981-13-8285-7>

Srinivasan, S. (2014). *Cloud Computing Basics*. Springer New York. <https://doi.org/10.1007/978-1-4614-7699-3>

- Statista. (2018). AI frameworks popularity/usage 2018. Statista. [verkkodokumentti]. [Viitattu: 2.4.2020]. Saatavissa: <http://www.statista.com/statistics/943058/ai-frameworks-popularity-usage-ranking/>
- Szeliski, R. (2011). Computer Vision: Algorithms and Applications. Springer-Verlag. <https://doi.org/10.1007/978-1-84882-935-0>
- TensorFlow. (2020). Why TensorFlow. [verkkodokumentti]. [Viitattu: 12.5.2020]. Saatavissa: <https://www.tensorflow.org/about>
- Theano. (2017). Welcome — Theano 1.0.0 documentation. [verkkodokumentti]. [Viitattu: 12.5.2020]. Saatavissa: <http://deeplearning.net/software/theano/>
- Turner, J. (2019). Robot Rules: Regulating Artificial Intelligence. Springer International Publishing. <https://doi.org/10.1007/978-3-319-96235-1>
- XGBoost. (2020). XGBoost Documentation — xgboost 1.1.0-SNAPSHOT documentation. [verkkodokumentti]. [Viitattu: 16.5.2020]. Saatavissa: <https://xgboost.readthedocs.io/en/latest/>