



Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

SESILIA MERIMAA

Koneoppiminen pilvipalveluna

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN
KOULUTUSOHJELMA

2021

Tekijä(t) Merimaa, Sesilia	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Maaliskuu 2021
	Sivumäärä 36	Julkaisun kieli Suomi
Julkaisun nimi Koneoppiminen pilvipalveluna		
Tutkinto-ohjelma Sähkö- ja automaatiotekniikka		
<p>Tiivistelmä</p> <p>Pilvipalvelut ja koneoppiminen ovat viimevuosien aikana kasvattaneet valtavasti suosiotaan. Opinnäytetyössäni tutkittiin, miten koneoppiminen voidaan toteuttaa pilvipalveluna.</p> <p>Työni teoriaosuudessa kerrottiin pilvipalveluista ja koneoppimisesta käsitteenä ja niiden keskeisimmistä ominaisuuksista. Lisäksi työssäni esiteltiin koneoppimisen käytön mahdollistavia yleisimmin käytössä olevia välineitä. Käytännönsuudessa työssäni selvitettiin tunnettujen pilvipalvelujen tarjoajien ratkaisut koneoppimiseen pilvipalveluna. Siinä käytiin läpi kunkin pilvipalveluntarjoajan vaatimat vaiheet ympäristön käyttöönotosta valmiin opetetun koneoppimismallin aikaansaamiseksi.</p> <p>Tutkimuksen lopputuloksena saatiin selville, että vertailtavat pilvipalveluntarjoajat ovat perusominaisuuksiltaan hyvin samankaltaisia. Eroavaisuuksia löytyy kuitenkin kustannuksista, tarjottavien palvelujen määrästä ja niiden käytännöllisyydestä. Vertailuista alustoista ainoastaan Google Cloud tarjoaa ilmaisen vaihtoehdon koneoppimiselle. Microsoft Azure tarjoaa mielipiteitä jakavan täysin koodittoman tavan suorittaa koneoppimista. Amazon Web Services tarjoaa laajimman valikoiman koneoppimispalveluita, mutta suurimmasta osasta saatu hyöty ei käytännössä kuitenkaan ylitä muiden pilvipalveluntarjoajien ratkaisuilla saatua hyötyä.</p>		
Avainsanat: Pilvipalvelut, koneoppiminen		

Author(s) Merimaa, Sesilia	Type of Publication Bachelor's thesis	Date March 2021
	Number of pages 36	Language of publication Finnish
Title of publication Machine learning as a cloud service		
Degree programme Electrial and Automation Engineering		
Abstract Cloud services and machine learning have grown enormously in popularity in recent years. My Bachelor's thesis investigated how machine learning can be implemented as a cloud service. The theoretical part of my work was about cloud services and machine learning as a concept and their key features. In addition, my work introduced the most commonly used tools for machine learning. In the practical part of my work, the solutions of well-known cloud service providers for machine learning as a cloud service were investigated. It included the steps required by each cloud provider to deploy the environment to achieve a ready taught machine learning model. As a result of the study, it was found that the compared cloud service providers have very similar basic features. However, there are differences in costs, in the amount and the practicality of different services. Of the comparisons, only Google Cloud offers a free alternative to machine learning. Microsoft Azure provides a completely code-free way to perform machine learning that shares opinions. Amazon Web Services offers the widest range of machine learning services, but the practical benefits of most do not outweigh the benefits of other cloud provider's solutions.		
Key words: Cloud services, machine learning		

SISÄLLYS

1 JOHDANTO	5
2 PILVIPALVELUT	6
2.1 Pilvipalvelut käsitteenä.....	6
2.2 Palvelumallit.....	6
2.3 Palvelutyypit.....	8
2.4 Käyttömahdollisuudet	9
2.5 Turvallisuus	10
3 KONEOPPIMINEN	10
3.1 Koneoppiminen käsitteenä	10
3.2 Ohjattu oppiminen.....	11
3.3 Ohjaamaton oppiminen	12
3.4 Vahvistusoppiminen.....	14
3.5 Käytettävät välineet	15
3.5.1 Python	15
3.5.2 Analytiikkakirjastot.....	16
4 TUTKITUT PALVELUT	19
4.1 Koneoppiminen palveluna.....	19
4.1.1 Microsoft Azure	19
4.1.2 Google Cloud	23
4.1.3 Amazon Web Services.....	27
4.2 Toiminnallisten kokeilujen tulokset	29
4.2.1 Käyttöönotto	30
4.2.2 Koneoppimisen suorittaminen	33
4.2.3 Koneoppimismallin tallentaminen	34
5 POHDINTA	35

LÄHTEET

LIITTEET

1 JOHDANTO

Pilvipalvelu on IT-palvelun toimittamista Internetin kautta. Nykyisen pilvipalveluinfrastruktuurin syntyyn ja kehitykseen on vaikuttanut virtuaalikoneiden käytön yleistyminen 1990-luvulta lähtien. Pilvipalvelujen suosio onkin viimevuosien aikana kasvanut merkittävästi. Yhä useampi organisaatio pyrkii siirtämään toimintaansa pilveen. Tämä edellyttää kuitenkin laajaa tutkimusta eri pilvipalvelumalleista ja -tyypeistä. Lisäksi on tärkeää myös vertailla eri pilvipalvelutarjoajia.

Koneoppimisessa koneen tarkoitus on oppia itsenäisesti päättämään oikeaan lopputulokseen ilman, että käyttäjä sitä opettaa. Ajatus oppimisesta esitettiin jo 1950-luvulla, jolloin termiä ”koneoppiminen” käytettiin ensimmäisen kerran. Samoihin aikoihin ilmestyivät myös ensimmäiset koneoppimisen toteutukset. Tässä työssä tutkinta kohdennetaan koneoppimiseen pilvipalveluna, sillä koneoppiminen on noussut kysytyksi osa-alueeksi erilaisten IT-ratkaisujen parissa.

Tämän työn tehtävänä on ensin selvittää, mitä pilvipalvelu käsitteenä tarkoittaa ja käydä läpi yleisimpiä tapoja hyödyntää pilvipalvelujen tarjoamia ratkaisuja tietotekniisiin tarpeisiin. Sen jälkeen tarkastellaan, mitä koneoppiminen käsitteenä tarkoittaa. Sitten esitellään yleisimmin käytettyjä koneoppimismenetelmiä ja käydään läpi muutamia olennaisia koneoppimisen käyttöön tarvittavia ohjelmoinnin välineitä, joita ovat Python-ohjelmointikieli ja sen analytiikkakirjastot.

Opinnäytetyön päätavoitteena on tutustua Googlen, Microsoftin ja Amazonin pilvipalvelualustaan, jotka tarjoavat erilaisia ratkaisuja koneoppimismenetelmien käyttöön. Tavoitteena on selvittää, miten näiden palvelujen käyttöönotto ja hyödyntäminen käytännössä tapahtuu ja mitä eroavaisuuksia pilvipalveluntarjoajilla on. Tässä työssä keskitytään erityisesti palvelujen käyttäjäystävällisyyteen ja eri käyttömahdollisuuksiin.

2 PILVIPALVELUT

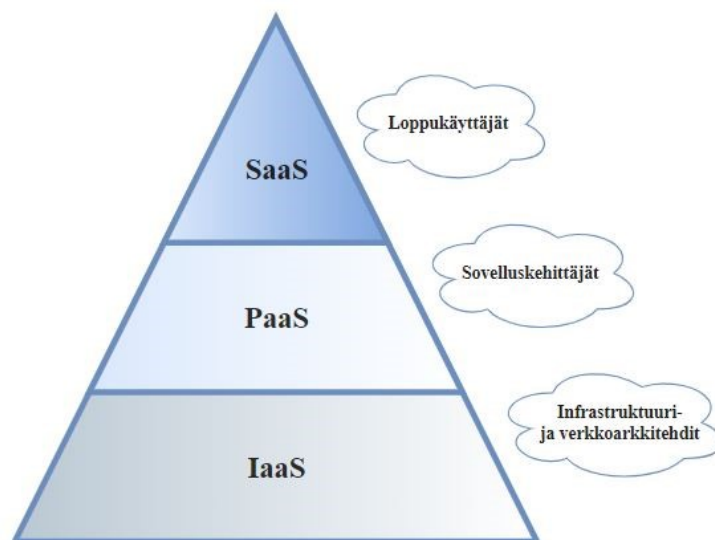
2.1 Pilvipalvelut käsitteenä

Pilvipalvelu tarkoittaa IT-palvelua, jota käytetään Internetin kautta. Palveluita on mahdollista käyttää esimerkiksi avaamalla verkkoselain laitteella, jolla on toimiva verkkoyhteys. Pilvipalveluille olennaista on se, että palvelusta maksetaan käytön mukaan, ne ovat nopeasti saatavilla ja niiden teho määräytyy tarpeen mukaan. Eli käyttäjä maksaa vain siitä, mitä käyttää. Pilvipalvelu on tietokonepalvelun toimittamista Internetin eli ”pilven” kautta. Termiä pilvi käytetään usein Internetin kielikuvana, ja se voidaan määritellä uudelleenlaiseksi apuohjelmaksi, joka käyttää kolmansien osapuolten Internetin kautta asennettuja virtuaalipalvelimia (Harkut 2019). Tietokonepalveluna voivat toimia esimerkiksi palvelimet, tallennustila, tietokannat, verkko, ohjelmistot sekä datan käsittelyssä käytettävä analytiikka, tekoäly ja koneoppiminen.

2.2 Palvelumallit

Pilvipalvelut jaetaan yleensä kolmeen palvelumalliin eli IaaS-, PaaS- ja SaaS-palveluihin. Oikean palvelumallin tai palvelumallien yhdistelmän valitseminen on kriittinen menestystekijä pilvipohjaisten ratkaisujen toimittamisessa. Oikean valinnan tekemiseksi on ymmärrettävä, mitä palvelumalleja on ja mitkä ovat pilvipalvelujen tarjoajien vastuut verrattuna pilvipalvelun kuluttajan vastuuseen. (Kavis 2014.)

Kuviossa 1 havainnollistetaan yleisimmät palvelumallit ja niistä hyötyvät käyttäjäryhmät.



Kuvio 1. Pilvipalvelujen palvelumallit

IaaS (Infrastructure-as-a-service) eli infrastruktuuri palveluna tarkoittaa, että konesalin kapasiteettia käytetään tai vuokrataan. Tämä on houkutteleva vaihtoehto organisaatioille, jotka haluavat rakentaa sovelluksia alusta alkaen ja haluavat hallita lähes kaikkia elementtejä itse. Se kuitenkin vaatii organisaatiolta laajan teknisen osaamisen. Parhaimmillaan tämä malli helpottaa innovointia ja lyhentää uusien sovellusten ja palveluiden käyttöönottoaikaa sekä ylläpitokustannuksia, mutta puutteita on huomattu turvallisuudessa liittyen kriittisimpiin tietoihin. IaaS-mallin käyttöönotto on usein organisaation ensimmäinen vaihe pilveen siirtymisessä. (Ranger 2018.)

PaaS (Platform-as-a-service) eli alusta palveluna tarkoittaa ohjelmistojen ja laitteistojen vuokraamista ja käyttämistä. PaaS-palvelumalli pitää sisällään tallennus-, verkko- ja virtuaalipalvelimien lisäksi työkalut ja ohjelmistot. Näitä ohjelmistokehittäjät tarvitsevat rakentaakseen ja kehittääkseen sovelluksia, joiden tulee sisältää esimerkiksi väliohjelmistoja, tietokannan hallintaa tai toimiva käyttöjärjestelmä. (Ranger 2018.)

PaaS-palvelumallin etuna on, että nämä alustat integroituvat useisiin kolmansien osapuolten ohjelmistoratkaisuihin, jotka voivat olla esimerkiksi laajennuksia tai lisäosia. Lisäksi ohjelmistokehittäjien ei enää tarvitse hallinnoida ja ylläpitää sovellusliittymien tekniikkaa, joten PaaS antaa organisaatioille mahdollisuuden keskittyä ydinosaamiseensa. (Kavis 2014, 41.)

SaaS-palvelumalli (Software-as-a-service) on palvelumalli, johon organisaatioiden tulisi pyrkiä, jos sen toiminta halutaan siirtää kokonaan pilveen. SaaS-palvelumalli tarkoittaa sovelluksen toimittamista palveluna. Taustalla olevalla laitteistolla ja käyttöjärjestelmällä ei ole merkitystä loppukäyttäjälle, joka käyttää palvelua verkkoselaimen tai sovelluksen kautta. SaaS-mallin menot koostuvat sovelluksista, etenkin sovellushankinnoista ja järjestelmäinfrastruktuuri-ohjelmistoista. (Ranger 2018.)

SaaS-palvelumallissa pilvipalvelun tarjoaja siis hoitaa kaiken infrastruktuurin, sovelluslogiikan, käyttöönnotot ja kaiken, mikä liittyy tuotteen tai palvelun toimittamiseen. Organisaation ei itse tarvitse tarjota ylläpitoa tai palkata henkilöstöä hallinointiin. (Kavis 2014, 42.)

2.3 Palvelutyypit

Pilvipalvelut jaetaan usein kolmeen yleisimpään palvelutyyppiin eli yksityiseen pilveen, julkiseen pilveen ja hybridipilveen. Oikean palvelutyyppin valinnassa organisaation tulee ottaa huomioon laskenta-, verkko- ja tallennustarpeet, käytettävissä olevat resurssit ja liiketoimintatavoitteet sekä palvelutyyppien edut ja haitat. Palvelutyypit vaihtelevat sen mukaan, kuka hallitsee infrastruktuuria ja missä se sijaitsee. (Shaptonova, 2020.)

Kuviossa 2 havainnollistetaan yleisimmät pilvipalvelutyypit.



Kuvio 2. Pilvipalvelujen palvelutyypit.

Yksityinen pilvi on omaan käyttöön varattu palvelu, jota ylläpitää vain yksi organisaatio. Sitä hallinnoidaan ja isännöidään sisäisesti tai jonkin kolmannen osapuolen toi-

mesta. Koska ylläpito toteutetaan itse, tämä pilvityyppi vaatii yleensä paljon resursseja. Yksityinen pilvi vaatii tilaa, laitteistoja ja ympäristön valvontaa, mikä johtaa myös määrääjain lisäinvestointeihin. (Chopra 2017, 51.)

Pilven sanotaan olevan julkinen, kun palvelut suoritetaan verkon kautta, joka on avoin julkiseen käyttöön. Teknisesti katsottuna julkisen ja yksityisen pilven pilviarkkitehtuurin erot voivat olla hyvin pieniä, mutta tietoturvassa on usein eroavaisuuksia. Yleensä julkiset pilvipalvelujen tarjoajat, kuten Amazon AWS, Microsoft Azure ja Google Cloud, omistavat ja ylläpitävät infrastruktuuria ja tarjoavat pääsyn vain Internetin kautta, toisin sanoen suoraa yhteyttä ei tarjota. (Chopra 2017, 51.)

Hybridipilvi on kahden aikaisemman tyypin yhdistelmä, jotka pysyvät kumpikin erillisinä kokonaisuuksina, mutta ovat sidottuja yhteen. Hybridipilvi tarjoaa siis molempien käyttöönottomallien edut, mutta myös haitat. Organisaatiolla on käytössä hybridipilvi esimerkiksi silloin, kun yritys käyttää julkista kehitysympäristöä, joka lähettää tietoja yksityiseen pilveen tai palvelinkeskukseen perustuvaan sovellukseen (Hurwitz, 2012, 13).

2.4 Käyttömahdollisuudet

Pilvipalveluilla voidaan nykypäivänä kattaa lähes kaikki palvelut, jotka eivät vaadi käyttäjää olemaan fyysisesti lähellä käyttämäänsä laitetta. Pilvipalveluilla voidaan tarjota esimerkiksi laaja valikoima vaihtoehtoja tallennukseen tai prosessointitehoa niin yritysten kuin yksityiskäyttöön tarkoitettuihin IT-ratkaisuihin. Keskeiset mahdollistavat tekniikat ovat nopeat laaja-alaiset verkot, tehokkaat ja edulliset palvelintietokoneet sekä korkean suorituskyvyn virtualisointi laitteistolle. (Ranger 2018; National Institute of Standards and Technology 2019.)

Asiakaskunta vaihtelee suuryrityksistä pieniin ja keskisuuriin yrityksiin. Eri liikeyritykset voivat siis käyttää pilvipalveluja eri tavoin. Jotkut käyttäjät ylläpitävät kaikkia sovelluksia ja tietoja pilvessä, kun taas toiset käyttävät hybridimallia pitämällä tietyt sovellukset ja tiedot yksityisissä palvelimissa ja toiset pilvessä. (Harkut 2019.)

2.5 Turvallisuus

Pilvipalveluiden turvallisuus on yksi tärkeimmistä huomioitavista asioista organisaation siirtyessä pilveen. Pilvipalvelun on oltava sekä helposti saatavilla että samalla mahdollisimman turvallinen. Vaikka tietojen tallentaminen pilveen on usein turvallisempaa kuin omalle kiintolevyllä, tulee palveluntarjoajilla olla tarkat menetelmät pilvien suojaamiseksi.

Useimpien organisaatioiden pilvipalvelujen riskit saadaan muutamalla keskeisellä asialla minimoitua. Ensimmäisenä tulisi uusien turvallisuuskäytäntöjen määrittelyt tehdä uudestaan, sillä todennäköisesti ne eivät täysin sovellu uuteen pilvi-infrastruktuuriin. Pilviympäristön tulisi olla automatisoitu ja konfiguroitu huomaamaan kaikki poikkeamat niin suorituskyvyyssä kuin muissakin valvontakohteissa. Infrastruktuuria tulisi myös kohdella sovelluskoodin tapaan, joka tarkistetaan ennen käyttöönottoa. Tällöin vältytään väärien määritysten ja vaatimustenvastaisuuksien tuontia palveluun. (Accenture 2020.)

3 KONEOPPIMINEN

3.1 Koneoppiminen käsitteenä

Koneoppiminen on tekoälyn osa-alue. Se on kokoelma algoritmeja ja tekniikoita, joita käytetään datasta oppivien järjestelmien toteuttamiseen. Koneoppimisessa keskitytään kehittämään tietokoneohjelmia, jotka voivat käyttää annettuja tietoja itsenäiseen oppimiseen sekä suorituskyvyn optimoimiseen esimerkkietojen tai aiemman kokemuksen avulla. (Alpaydin 2014; Seikku 2018; Lee 2019, 2.)

Kaikkien koneoppimismallien perusta on tilastotieteessä. Lisäksi koneoppimisessa on vahvasti läsnä matematiikka ja tietojenkäsittelytiede. Termiä koneoppiminen on kuitenkin vaikeaa yksiselitteisesti määritellä, sillä osa koneoppimisen menetelmistä käyttää itse ”oppimisen” sijaan tilastollisia ja/tai matemaattisia menetelmiä. Oppimisella tarkoitetaan tässä tilanteessa sitä, että kone oppii toistuvista tapahtumista ilman, että

käyttäjää erikseen sitä opettaa. Aikaisemmin koneoppimisen menetelmiä käytettiin lähinnä vain tilastotieteessä, mutta tänä päivänä koneoppimisen käsite on kuitenkin laajentunut ja se eroaa tilastotieteestä mm. käyttötarkoituksensa takia.

Koneoppimisalgoritmit luokitellaan usein ohjatuksi oppimiseksi, ohjaamattomaksi oppimiseksi ja vahvistusoppimiseksi. Koneoppimisalgoritmien luokittelua on havainnollistettu kuviossa 3.



Kuvio 3. Koneoppimisalgoritmit.

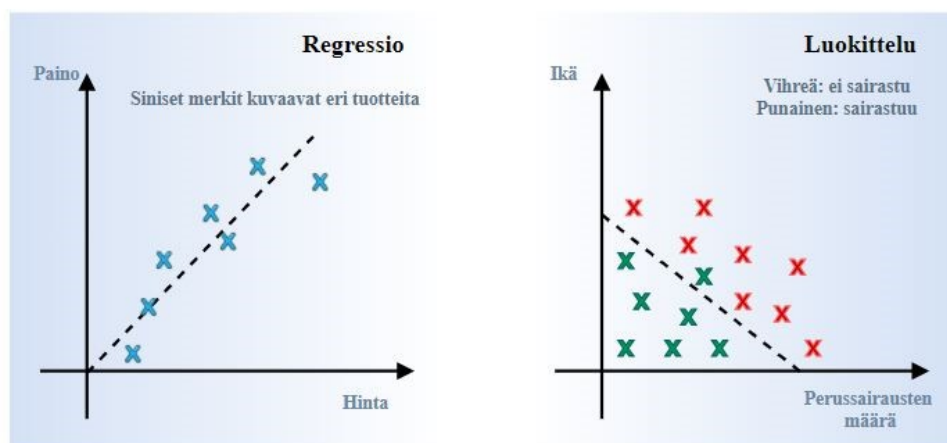
3.2 Ohjattu oppiminen

Suurin osa käytännön koneoppimisesta käyttää ohjattua oppimista. Ohjatut koneoppimisalgoritmit voivat soveltaa aiemmin opittua tulevien tapahtumien ennustamiseksi. Ohjatussa oppimisessa datan rakenne on tiedossa. Siinä ennustetaan kohdemuuttujan arvoa tunnettujen syötemuuttujien avulla. Oppimisalgoritmi tuottaa päätellyn funktion ennusteiden tuottamiseksi syötetyistä lähtöarvoista. Ohjatun oppimisen algoritmin avulla on mahdollista myös verrata tulosta oikeaan, todelliseen tulokseen. Mahdollisen virheen avulla saadaan algoritmia muokattua entistä tarkemmaksi malliksi. Oppiminen loppuu, kun algoritmi saavuttaa hyväksytyyn suorituskäyvyn. (Brownlee 2020; Expert.ai 2020.)

Ohjattu oppiminen voidaan vielä jakaa regressio- ja luokittelualgoritmeihin. Regressiossa muiden tiedossa olevien arvojen perusteella pyritään ennustamaan tietyn jatkuvan kohdemuuttujan todellinen arvo, kuten ”hinta” tai ”paino”. Tunnetuimpia regression

algoritmeja ovat lineaariregressio, päätöspuu, satunnaismetsä ja neuroverkko. Luokittelussa kohdemuuttuja on arvoiltaan diskreetti. Tällaisen muuttujan arvoina voisi olla esimerkiksi ”tartunta” tai ”ei tartuntaa”. (Brownlee 2020.)

Regressiota ja luokittelua on havainnollistettu kuviossa 4. Kohdemuuttuja pyritään ennustamaan tiedettäessä muiden muuttujien arvot. Tähän voidaan käyttää luokittelun algoritmeja, joista tunnetuimpia ovat logistinen regressio, tukivektorikone, Bayesilainen päättely ja k:n lähimmän naapurin luokitin. Regression lisäksi myös luokittelussa voidaan käyttää päätöspuu- ja satunnaismetsäalgoritmeja.



Kuvio 4. Esimerkki ohjatun oppimisen regressiosta ja luokittelusta.

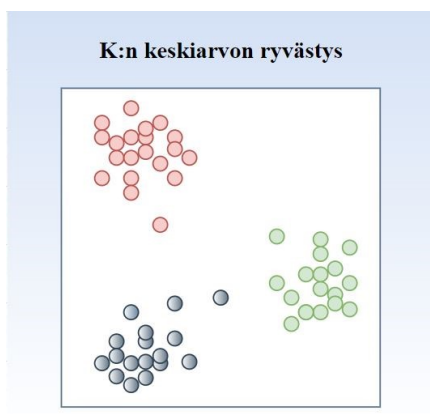
Kuvion 4 regressiossa ennustetaan tuotteen hinnan perusteella sen painoa. Ennustuksen virhe mitataan todellisten arvojen ja ennustukseen käytettävän katkoviivasuoran välisten etäisyyksien summana. Kuvion 4 luokittelussa ennustetaan henkilön sairastumista iän ja perussairauksien määrän avulla. Siinä ennustuksen virhe mitataan väärin ennustettujen arvojen määränä.

3.3 Ohjaamaton oppiminen

Ohjaamattomassa oppimisessä vain syötemuuttujat ovat tiedossa, eli datan rakenne ja oikea todellinen kohdemuuttujan arvo ovat etukäteen tuntemattomia. Ohjaamattomassa oppimisessä pyritään löytämään tutkittavasta datasta suhteita. Ohjaamattoman oppimisen avulla voidaan esimerkiksi ryhmitellä asiakkaat ennalta tuntemattomiin ryhmiin. Tiettyyn ryhmään kuuluvilla asiakkailla voisi olla esimerkiksi samanlaiset

ostotavat. Tätä menetelmää kutsutaan ryvästyksiksi eli klusteroinniksi. Klusteroinnin algoritmeja ovat esimerkiksi k :n keskiarvon ryvästys ja hierarkkinen ryvästys. (Lee 2019, 6; Brownlee 2020.)

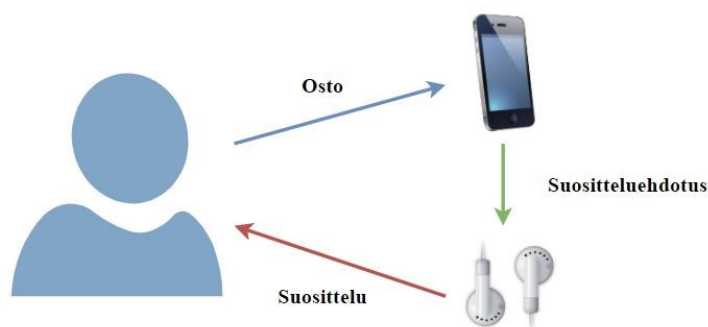
Kuviossa 5 on havainnollistettu k :n keskiarvon ryvästystä, jossa k on kolme. Eli noin 50 datahavaintoa on ryvästetty kolmeen ryhmään, jotka muodostuvat klusterien havaintojen ja niiden keskiarvojen etäisyyksien perusteella.



Kuvio 5. K :n keskiarvon ryvästys.

Yksi tunnetuista ohjaamattomaan oppimiseen perustuvista neuroverkkomalleista on suomalaisen Teuvo Kohosen 1980-luvun alussa kehittämä SOM (Self-Organizing Map), eli itseohjautuva kartta, jota tunnetusti kutsutaan myös Kohosen kartaksi. Tämän neuroverkkomallin avulla pystytään järjestelemään ja luokittelemaan isoja tietomääriä siten, että samankaltaiset asiat hakeutuvat lähelle toisiaan. Siten muodostuu ryppäiden kartta, jolla kuvataan data-aineistossa olevia luokkia. (Kohonen & Honkela 2007.)

Ohjaamatonta oppimista voidaan käyttää mm. suositteluun, jossa tuotteita ja henkilöitä sisältävässä datassa pyritään löytämään henkilöille suositeltavia tuotteita. Kuviossa 6 havainnollistetaan suosittelujärjestelmän mahdollista toimintatapaa. Kuvion henkilö on ostanut suosittelujärjestelmän sisältävästä verkkokaupasta itselleen uuden puhelimen. Tämän jälkeen kyseiselle henkilölle suositellaan henkilön ostokäyttäytymisen perusteella muodostuneen profiloinnin mukaan muita tuotteita, tässä tapauksessa uusia kuulokkeita.

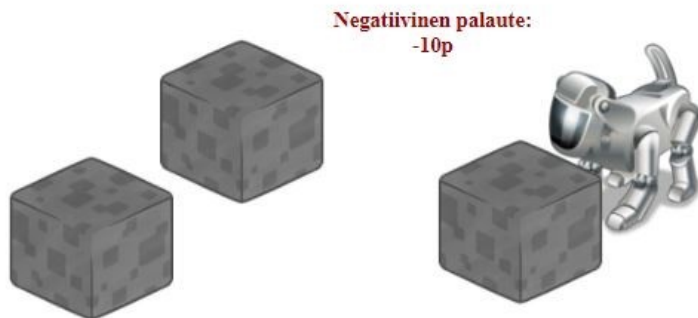


Kuvio 6. Esimerkki suosittelujärjestelmästä.

3.4 Vahvistusoppiminen

Vahvistusoppiminen on koneoppimisen haara, jossa oppiminen tapahtuu vuorovaikutuksessa ympäristön kanssa. Se on tavoitteellista oppimista. Siinä oppijalle ei opeteta, mitä toimia tulisi tehdä, vaan oppija oppii toimintansa seurauksista. Se on yksi tekoälyn aktiivisimmista tutkimusalueista ja se kasvaa nopeasti monenlaisten algoritmien avulla. Vahvistusoppimisen avulla koneet ja ohjelmistoagentit voivat automaattisesti määrittää ihanteellisen käyttäytymisen tietyssä asiayhteydessä maksimoidakseen sen suorituskyvyn. Jotta agentti voi oppia, vaaditaan yksinkertainen ympäristön antama positiivinen tai negatiivinen palaute. Tämä tunnetaan vahvistussignaalina. Algoritmi pyrkii löytämään eniten positiivista vahvistussignaalia tuottaneen mallin. (Ravichandiran 2018, 6.)

Esimerkkinä vahvistusoppimisesta voidaan kuvitella, että halutaan opettaa robottikoira kävelemään ilman esteisiin törmäämistä. Mikäli robottikoira törmää esteeseen, annetaan negatiivinen vahvistussignaali. Jos robottikoira ymmärtää, että osuminen esteeseen johtaa negatiiviseen palautteeseen ja kiertämällä esteen, sille annetaan kaksinkertainen määrä positiivista palautetta. Tällä tavalla robottikoira oppii oikean polun ja yrittää maksimoida positiivisen vahvistussignaalin määrän. Tätä esimerkkiä on havainnollistettu kuvassa 1 ja kuvassa 2.



Kuva 1. Vahvistusoppimisen esimerkki negatiivisesta signaalista.



Kuva 2. Vahvistusoppimisen esimerkki positiivisesta signaalista.

3.5 Käytettävät välineet

Yleisin koneoppimisessa käytettävä väline on Python-ohjelmointikieli ja sen sisältämät analytiikkakirjastot.

3.5.1 Python

Python on selkeä ja suosittu ohjelmointikieli. Pythonilla voidaan toteuttaa olio-ohjelmointia, funktionaalista ohjelmointia ja rakenteista ohjelmointia. Pythoniin sisältyy suuri standardikirjasto, joka tukee monia yleisiä ohjelmoinnin tarpeita ja tehtäviä. Lisäksi sen ominaisuuksia pystytään helposti laajentamaan lisäämällä uusia kirjastoja ja moduuleja. Pythonin lataaminen, käyttäminen tai sovellukseen sisällyttäminen on ilmaista. Se on saatavana avoimen lähdekoodin lisenssillä, joten Pythonia voidaan myös vapaasti muokata ja edistää. (Python 2021.)

Pythonin yksinkertainen ja helposti opittava syntaksi korostaa luettavuutta, mikä helpottaa koneoppimismallien luomista ja kehittämistä. Python soveltuu hyvin koneoppimiseen, sillä sen etuihin kuuluu myös joustavuus, alustojen riippumattomuus, laaja käyttäjäyhteisö sekä pääsy suuriin kirjastoihin ja koneoppimisen kehyksiin. (Be-klemysheva 2021.)

3.5.2 Analytiikkakirjastot

Pythonin analytiikkakirjastot koostuvat moduuleista, jotka sisältävät funktioita. Yleisimmät koneoppimisessa käytettävät kirjastot ovat NumPy, Pandas, Matplotlib, Scikit-learn ja TensorFlow sekä sen sisältämä Keras.

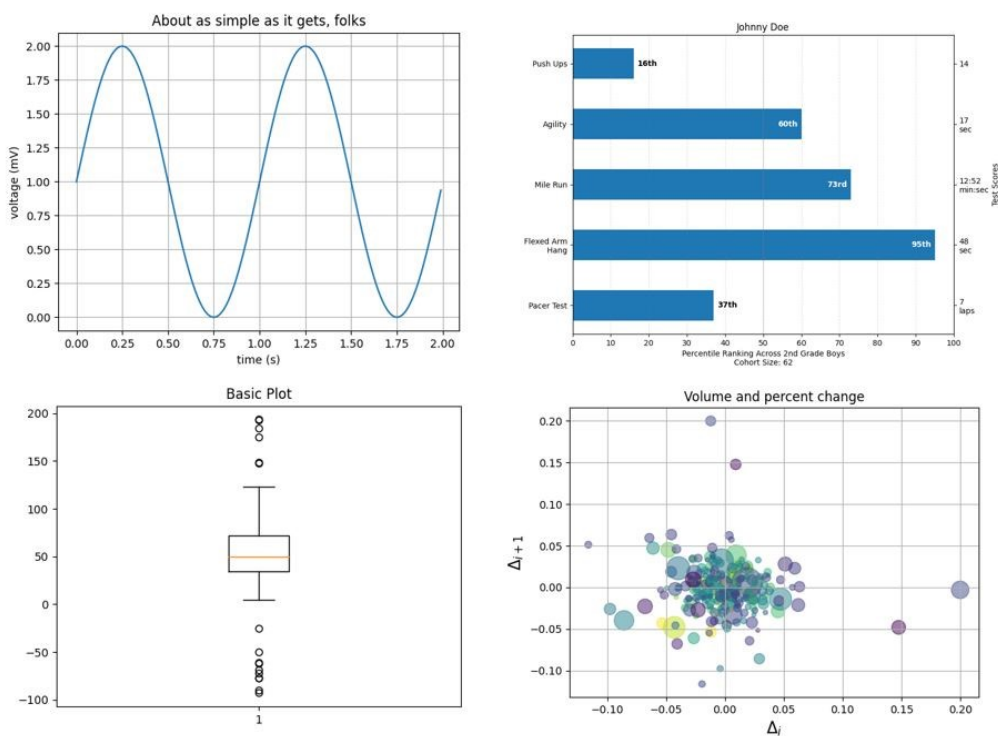
NumPy muodostaa perustan monille muille kirjastoille. Näitä ovat mm. Pandas, Matplotlib, Scikit-learn ja TensorFlow. NumPy-kirjasto on toimiva paketti tieteelliseen laskentaan. NumPy mahdollistaa suurten N-ulotteisten taulukoiden nopean käsittelyn. Se antaa mahdollisuuden myös käsitellä aineistoja, joita ei pystyttäisi käsittelemään alkuperäisellä Pythonilla. NumPy-kirjaston numeeriset laskentatyökalut kattavat mm. matemaattiset toiminnot, satunnaislukugeneraattorit ja lineaarialgebran operaatiot. NumPy on helppokäyttöinen ja sopii kaiken tasoisille ohjelmoijille sen korkean tason syntaksin ansiosta. (NumPy 2021.)

Pandas on datakehikkojen käsittelyyn tarkoitettu kirjasto, jota käytetään usein yhdessä NumPy-kirjaston kanssa. Helppokäyttöinen ja joustava Pandas-kirjasto sisältää nopean ja tehokkaan DataFrame-objektin datan käsittelyyn. Se sisältää myös työkalut datan lukemiseen ja kirjoittamiseen ohjelman tietorakenteiden ja eri ulkoisten formaattien välillä. Näitä ovat esimerkiksi tekstitiedostot, binääriset tiedostot ja SQL-tietokannat. Muita Pandas-kirjaston käyttökohteita ovat puuttuvan tiedon käsittely ja datakehikkojen uudelleen muotoilu. (Pandas 2021.)

Matplotlib on datan visualisointiin tarkoitettu kirjasto. Se mahdollistaa erilaisten korkealaatuisten 2D-kaavioiden, kuten viiva-, palkki-, laatikko- ja pistekaavioiden piirtämisen. Säännöllistä käyttöä varten Matplotlib-kirjasto tarjoaa yksinkertaisen olioraja-

pinnan ja pyplot-moduulin piirtämisen helpottamiseksi. Matplotlib tukee myös vuorovaikutteista käyttöliittymää, joka on erittäin hyödyllinen web-pohjaisten sovellusten luomisessa. Matplotlib-kirjaston suurimpia etuja on sen helppokäyttöisyys. Sen sisältämät moduulirakenteet mahdollistavat yksinkertaisen käyttöönoton. Se on myös tiiviisti integroitu Numpy- ja Pandas-kirjastoihin. Esimerkiksi Pandas-kirjaston avulla tehdyn DataFrame-objektin saa helposti visualisoitua Matplotlib-syntaksilla. Lisäksi Matplotlib tarjoaa valmiita oletustyyli- ja taulukoita, joilla visualisoinnin toteuttaminen on yksinkertaista. (Yim, Chung & Yu 2018, 7-8.)

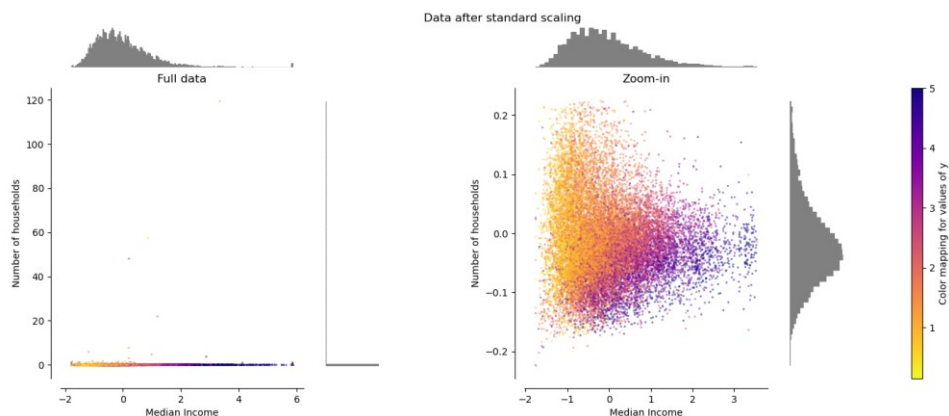
Kuvassa 3 on esimerkkejä erilaisista Matplotlib-kirjaston kaaviomalleista.



Kuva 3. Esimerkkikaaviot viiva-, palkki-, laatikko- ja pistekaavioista. (Matplotlib 2021)

Scikit-learn on avoimen lähdekoodin kirjasto, joka tukee koneoppimisen ohjattua ja ohjaamatonta oppimista. Se tarjoaa erilaisia työkaluja datan esiprosessointiin ja koneoppimismallin sovittamiseen, valintaan sekä arviointiin. Scikit-learn-kirjasto on rakennettu NumPy-, SciPy- ja Matplotlib-kirjastojen päälle. Datan esiprosessointi on yleensä koko koneoppimisen tärkein ja työläin vaihe. Olennaista on myös datan visualisointi ennen ja jälkeen sen muuntamista käyttötarkoitukseensa sopivaksi. Tätä on

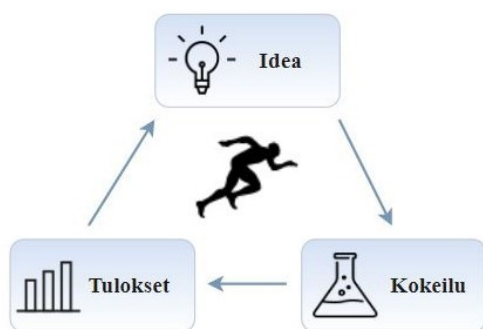
havainnollistettu kuvassa 4. Siinä vasemmalla puolella on visualisoitu koko käytössä oleva data-aineisto sellaisenaan eli täysin ilman käsittelyä. Oikealla puolella on kuvattuna esiprosessoitu ja skaalattu data-aineisto. Eli tässä data on puhdistettu ja skaalattu käyttötarkoitukseen sopivaksi. (Scikit-learn 2021a; Tutorialspoint 2021.)



Kuva 4. Esimerkkikuva datan esiprosessoinnin visualisoinnista. (Scikit-learn 2021)

TensorFlow on Googlen kehittämä Python-perustainen avoimen lähdekoodin koneoppimisen alusta. Se on ensisijaisesti kehitetty syväoppimisen työkaluksi, mutta tukee myös perinteistä koneoppimista. TensorFlow mahdollistaa suurten numeeristen laske-
misten ja tietomäärien käsittelyn, joita vaaditaan syväoppimisessä. (Simplilearn 2020; TensorFlow 2021a.)

TensorFlow sisältää Keras-kirjaston, joka on helppokäyttöinen syväoppimisen ohjelmointirajapinta. Keras mahdollistaa syväoppimisessä käytettävien neuroverkkojen nopean kokeilun ja prototyyppien tekemisen. Eli ajatuksena on pystyä siirtymään ideasta mahdollisimman nopeasti lopulliseen tulokseen. Nopeudellaan Keras helpottaa uusien kokeilujen suorittamista. Tätä ajatusta on havainnollistettu kuviossa 7. (Keras 2021.)



Kuvio 7. Keras-ohjelmistorajapinnan käyttämisajatus havainnollistettuna

4 TUTKITUT PALVELUT

4.1 Koneoppiminen palveluna

MLaaS (Machine learning as a service), eli koneoppiminen palveluna viittaa useisiin palveluihin, jotka tarjoavat koneoppimisen työkaluja pilvipalvelujen osana. MLaaS-palveluntarjoajat tarjoavat kehittäjille palveluja, jotka kattavat esimerkiksi datan esiprosessoinnin, mallien opettamisen ja arvioinnin sekä niiden avulla ennustamisen. (Wolhuter 2018.)

Tässä työssä tutustutaan muutamaa suureen pilvipalvelujen tarjoajaan, joiden palveluihin sisältyy myös MLaaS-palvelut. Näitä ovat Microsoft, Google ja Amazon.

4.1.1 Microsoft Azure

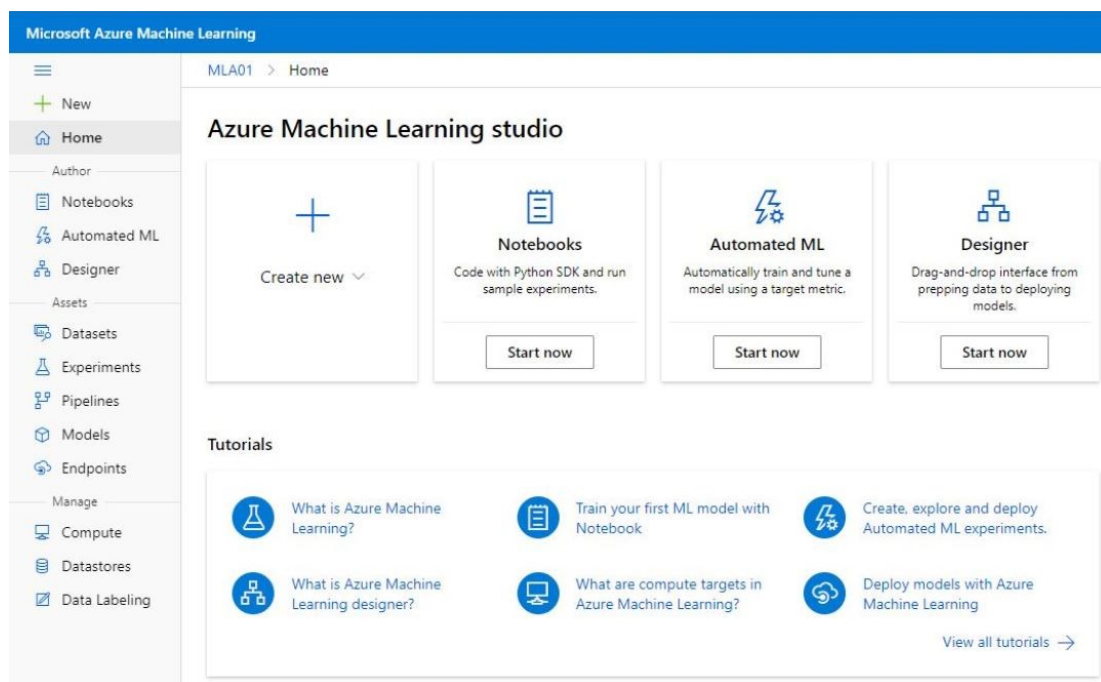
Microsoft Azure on Microsoftin luoma julkinen pilvipalvelualusta. Azuren avulla voi käyttää ja hallita Microsoftin tarjoamia pilvipalveluita ja resursseja. Azure tarjoaa paljon erilaisia pilvipalveluja eri käyttötarkoituksiin ja ne voivat vaihdella yksinkertaisista verkkopalveluista koko organisaation ylläpitoon pilvessä. Näitä ovat etätallennustilan, tietokantapalvelujen lisäksi mm. koneoppimisen palvelut, joita tässä työssä tutkitaan. Azuren resurssien ja palveluiden käyttöönottoon tarvitaan vain yhteys Internetin kautta Azure-portaaliin. Azure-portaali on verkkopohjainen konsoli, joka tarjoaa vaihtoehdon komentorivityökaluille. Sen graafisen käyttöliittymän avulla käyttäjä voi hallita Azure-tilausta sekä suorittaa, hallita ja seurata käytössä olevia palveluja ja resursseja. Tilaajana käyttäjälle kertyy kulutetuista Azure-resursseista aiheutuvat kustannukset, kuten esimerkiksi laskenta- ja tallennuskustannukset. (Microsoft 2021a.)

Microsoftin tarjoama Azure Machine Learning on valmis pilvipalvelukomponentti koneoppimiseen. Sitä voidaan käyttää koneoppimismallien opettamiseen, kehittämiseen, automatisointiin, hallintaan ja seurantaan (Microsoft 2021b). Tätä komponenttia voidaan pitää PaaS-palvelumallina. Microsoft tarjoaa myös valmiiksi konfiguroituja virtuaalikoneita (Data Science Virtual Machines), jotka on tarkoitettu koneoppimisen lisäksi data-analytiikkaan ja muuhun datan käsittelyyn. Eli Microsoft Azurella on myös

mahdollisuus IaaS-palvelumallin käyttöön. Tässä työssä keskitymme kuitenkin Azure Machine Learning -palveluun.

Microsoft Azuren tilaajana pääsee kirjautumaan Azure-portaaliin, jonka jälkeen luodaan uusi Azure Machine Learning -palvelun työtila (Workspace). Tämä työtila on pilven perusresurssi, jota käytetään koneoppimismallien kokeilemiseen, opettamiseen ja käyttöönottoon. Kun työtila on luotu, avataan Azure Machine Learning studio (myöh. AMLS). Se on verkkoportaali, joka yhdistää yksinkertaisen koodillisen sekä täysin koodittoman tavan koneoppimismallien käytölle (Microsoft 2021c).

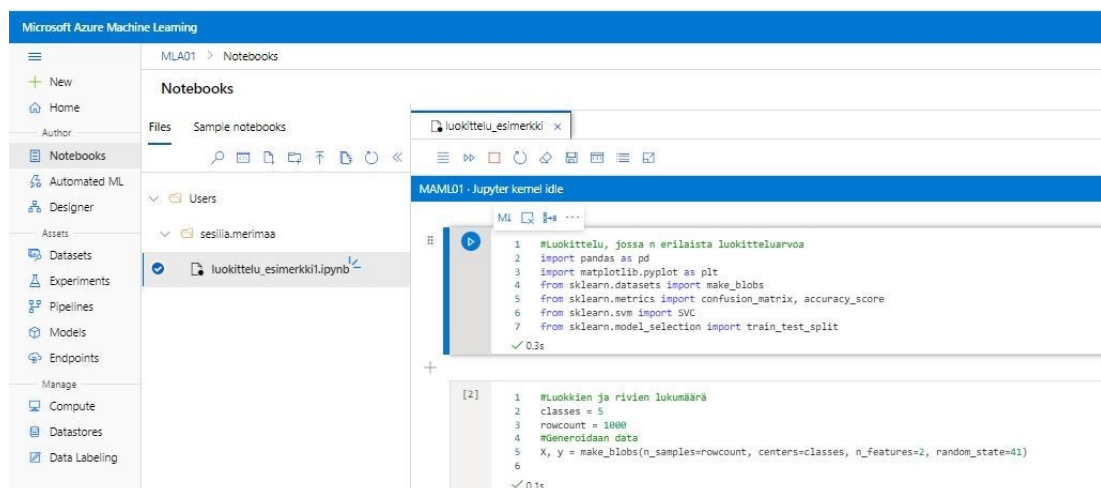
Kuvassa 6 on AMLS:n aloitusnäky, jonka avulla käyttäjä pääsee helposti kokeilemaan kaikkia studion koneoppimisen palveluita. Lisäksi aloitussivu takaa nopean pääsyn tutoriaaleihin, mikäli käyttäjä niitä tarvitsee.



Kuva 6. Azure Machine Learning studion aloitusnäky.

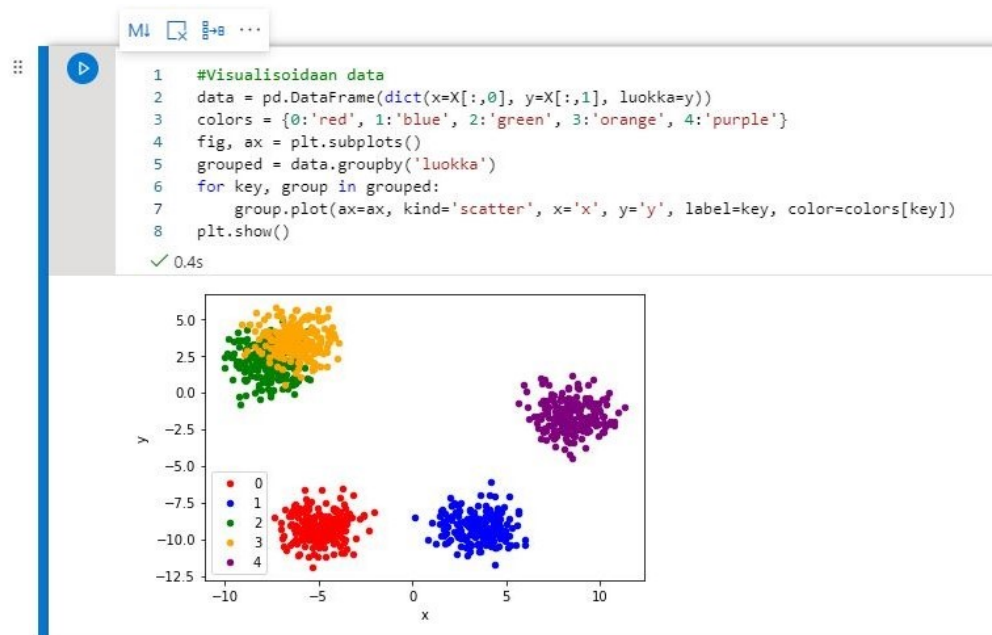
Tässä työssä tutustumme vain tavanomaiseen koneoppimisen toteuttamiseen ohjelmointikielen koodin avulla. Kuvassa 7 on havainnollistettu AMLS:ta, jossa koneoppimisalgoritmeja pystytään ajamaan Python-ohjelmointikielellä Jupyter Notebookin avulla. Jupyter Notebook on avoimen lähdekoodin verkkosovellus, jota käytetään mm.

datan esiprosessointiin, tilastolliseen mallinnukseen, datan visualisointiin ja koneoppimiseen (Jupyter 2021). AMLS:ssa Jupyter Notebook mahdollistaa koodin suorittamisen osissa. Eli käyttäjä voi halutessaan esimerkiksi ajaa uudelleen vain tarvittavat osat koodista. Näiden lisäksi Jupyter Notebookia voi hyödyntää tavallisen muistion tapaan.



Kuva 7. Azure Machine Learning studion Jupyter Notebook -instanssi.

AMLS:ssa kokeiltavana koneoppimisen menetelmänä käytetään luokittelua tukivektorikoneen avulla. Tukivektorikone on luokitteluun käytettävä koneoppimisalgoritmi. Käsiteltävänä oleva data on luotu Skicit-learnin koneoppimisen kokeiluun tarkoitettulla valmisdatan generointifunktiolla. Funktiolla `make_blobs()` voidaan luoda pisteitä, joilla on Gaussin jakauma (Skicit-learn 2021b). Skicit-learnin lisäksi AMLS tukee myös monia muita yleisimmin käytettyjä analytiikkakirjastoja. Datan esiprosessoinnin jälkeen käytetään Matplotlib-analytiikkakirjaston avulla tuotettua kaaviota datan visualisointiin. Sen jälkeen data jaetaan opetus- ja testidataan, jonka jälkeen tehdään luokittelumalliksi tukivektorikone. Viimeiseksi kokeillaan luokittelumallin ennustustarkkuutta, joka saa arvon 0,94. Koodi on kokonaisuudessaan liitteessä 1. Tätä prosessia datan visualisoinnista mallin opettamiseen ja sen avulla ennustamiseen on kuvattu kuvissa 8 ja 9.



Kuva 8. Datan visualisointi Matplotlib-kirjaston avulla.

```

[4] 1 #Jaetaan data opetus- ja testidataan
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state = 0)
3
✓ 0.1s

```

```

[5] 1 #Tehdään luokittelumalli
2 model = SVC(kernel='linear', C=1.0)
3 model.fit(X_train, y_train)
4 predictions = model.predict(X_test)
✓ 0.1s

```

```

[6] 1 #Tehdään luokittelumatriisi
2 cm = confusion_matrix(y_test, predictions)
3 print('Ennustustarkkuus testidatalle =', round(accuracy_score(y_test,predictions),2))
4
✓ 0.1s

```

Ennustustarkkuus testidatalle = 0.94

Kuva 9. Datan opetus- ja testidataan jako, luokittelumallin opetus ja sen avulla ennustaminen.

Edellä esitellyn tavan lisäksi koneoppimisalgoritmi on mahdollista suorittaa myös automatisoidusti tai täysin koodittomasti ”vedä ja pudota” suunnittelutyökalulla. Siinä toteutuksen eri osa-alueet rakennetaan valmiista paloista, eikä käyttäjän tarvitse kirjoittaa riviäkään koodia. Se sisältää myös valmiita malleja, joiden avulla pystyy koekilemaan erilaisia koneoppimisen ratkaisuja.

Azure Machine Learning on kokonaisuudessaan selkeä ja tarpeeksi tehokas käsittelemään suurempiakin datamääriä nopeasti. Se toimii muiden Azure-alustan palveluiden kanssa ja integroituu myös useiden avoimen lähdekoodin työkalujen ja kehysten kanssa. Tällaisia ovat esimerkiksi Visual Studio Code ja Jupyter Notebook. Tämä mahdollistaa datan käsittelyn aloittamisen lokaalisti omalla tietokoneella ja vasta myöhemmässä vaiheessa käyttäjä voi siirtää työnsä pilveen. AMLS tukee Pythonin lisäksi myös toista yleisesti käytettyä R-ohjelmointikieltä.

4.1.2 Google Cloud

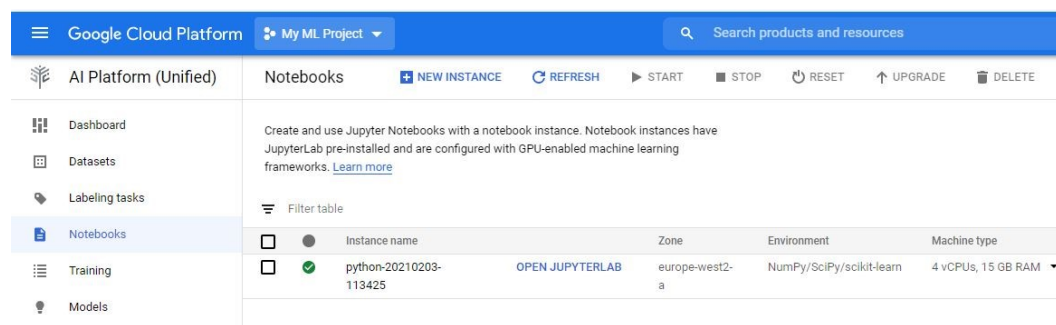
Google Cloud on Googlen luoma julkinen pilvipalvelualusta. Microsoft Azuren tapaan myös Google Cloud tarjoaa verkkopohjaisen konsolin resurssien luomiseen, hallintaan ja seurantaan. Google Cloud Platform (myöh. GCP) on osa Google Cloudia ja se tarjoaa hallintatyökalujen lisäksi datan käsittelyn, tietojen tallennuksen, data-analytiikan ja koneoppimisen palvelut. (Google Cloud 2021a.)

Seuraavissa alaluvuissa käydään tarkemmin läpi vaihtoehtoiset Googlen tarjoamat tavat koneoppimisen toteuttamiseksi pilvipalveluna. Nämä ovat Google Cloud AI Platform, joka vastaa PaaS-palvelumallin toteutusta ja Google Colaboratory, joka vastaa SaaS-palvelumallia. Lisäksi Google tarjoaa Azuren lailla mahdollisuuden valmiiksi konfiguroituihin virtuaalikoneisiin, jotka vastaavat puolestaan IaaS-palvelumallia, mutta siihen ei tässä työssä tarkemmin tutustuta.

GCP:n sisältämä Google Cloud AI Platform on valmiiksi konfiguroitu alusta datan käsittelyyn ja koneoppimiseen. Tämä alusta sisältää AI Platform Notebooksin, joka mahdollistaa virtuaalikoneinstanssien luonnin ja hallinnoinnin valmiiksi asennetuilla koneoppimiseen soveltuvilla paketeilla. (Google Cloud 2021b.)

Aloittaakseen AI Platform Notebooks -käytön on käyttäjän kirjauduttava Google-käyttäjätunnuksellaan GCP-konsoliin. Sen kautta luodaan uusi projekti ja Notebooks-instanssi. Käyttäjää maksaa käytöstä Notebooks-instanssien käyttämien pilvipalveluresurssien mukaan. Eli vähimmäishintoja tai ennakkomaksuja ei ole.

Tässä työssä käytetyssä Python-instanssissa on valmiiksi asennettuna JupyterLab, joka mahdollistaa koneoppimisen suorittamisen. JupyterLab on verkkopohjainen käyttöliittymä ja kehitysympäristö Jupyter-muistikirjoille, datalle ja koodille. Sitä käytetään samalta palvelimelta ja se käyttää samaa muistikirjan tiedostomuotoa kuin Jupyter Notebook (Jupyter 2021). Ohjelmointikielenä voi Pythonin lisäksi käyttää myös R-kieltä, aivan kuin Azurenkin tarjoamassa palvelussa. Kuvassa 10 on näkymä Python-instanssista GCP-konsolissa.



Kuva 10. AI Platform Notebooks -näkökulma instansseista.

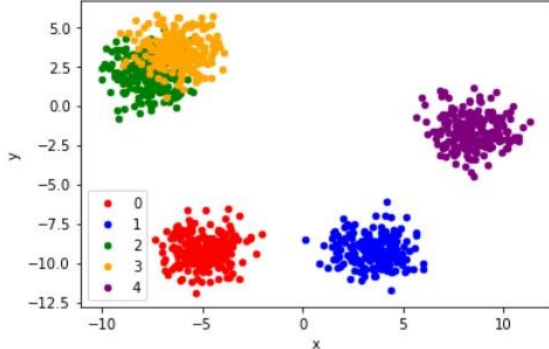
Kun avataan JupyterLab-käyttöliittymä, päästään kokeilemaan Python-koodin ajoa. AI Platform Notebookissa kokeiltavana koneoppimisen menetelmänä käytetään samaa tu-kivektorikoneluokittelua kuin aikaisemmin AMLS:ssa. Tätä on havainnollistettu kuvassa 11.

```
Tukivektorikone01.ipynb
```

```

data = pd.DataFrame(dict(x=X[:,0], y=X[:,1], luokka=y))
colors = {0: 'red', 1: 'blue', 2: 'green', 3: 'orange', 4: 'purple'}
fig, ax = plt.subplots()
grouped = data.groupby('luokka')
for key, group in grouped:
    group.plot(ax=ax, kind='scatter', x='x', y='y', label=key, color=colors[key])
plt.show()

```



```

[4]: #Jaetaan data opetus- ja testidataan
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state = 0)

[5]: #Tehdään Luokittelumalli
model = SVC(kernel='linear', C=1.0)
model.fit(X_train, y_train)
predictions = model.predict(X_test)

[6]: #Tehdään Luokittelumatriisi
cm = confusion_matrix(y_test, predictions)
print('Ennustustarkkuus testidatalle =', round(accuracy_score(y_test,predictions),2))

Ennustustarkkuus testidatalle = 0.94

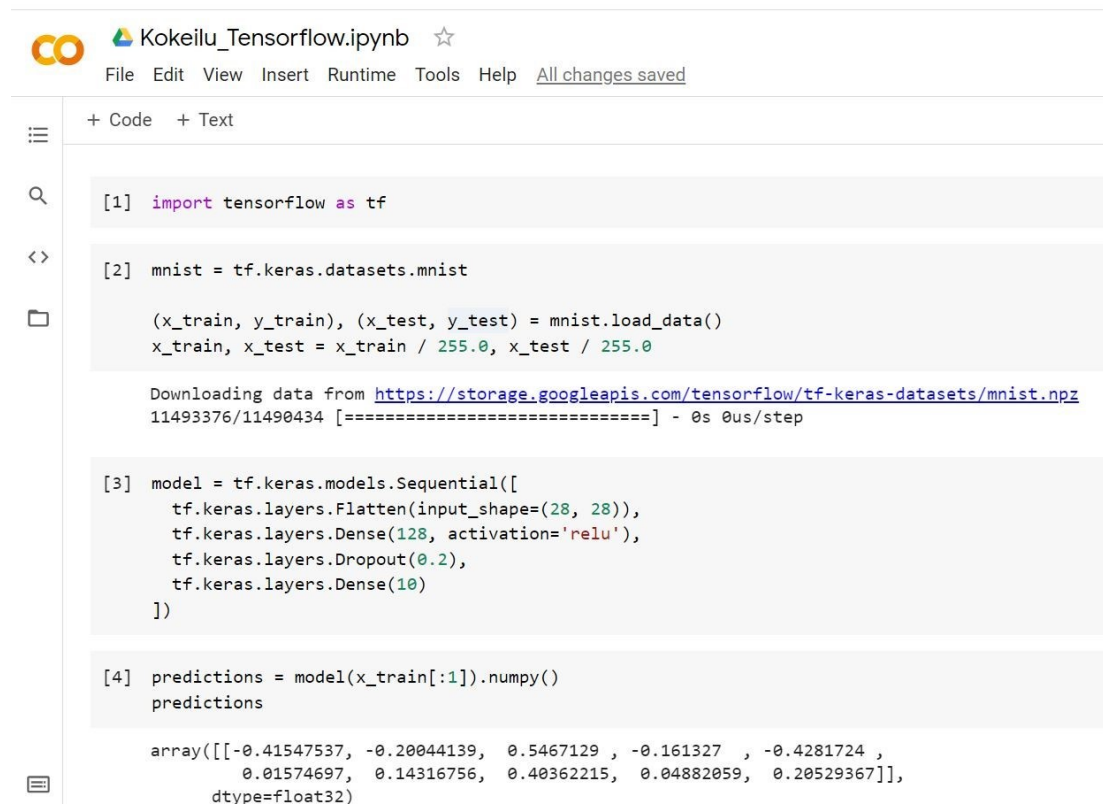
```

Kuva 11. Luokittelumallin kokeilu AI Platform Notebooks -instanssin sisältämässä JupyterLab-palvelussa.

Seuraavaksi tutustutaan Googlen toiseen koneoppimisen palveluun. Google Colabory eli Colab on Python-kehitysympäristö, joka toimii selaimessa käyttäen Google Cloudia. Colab on käyttäjälle täysin ilmainen kokonaan pilvessä toimiva palvelu. Colab perustuu Jupyter Notebook -verkkosovellukseen, joten sen käyttöä varten käyttäjän ei tarvitse ladata, asentaa tai suorittaa mitään. Colab tarjoaa mahdollisuuden tallentaa Jupyter Notebook -tiedostoja mm. Google Drive -palveluun. Se myös tukee monia suosittuja koneoppimisen kirjastoja. (Colab 2021).

Tässä työssä tutustutaan Colabin käyttöön valmiin tutoriaaliv avulla, jonka nimi on “TensorFlow 2 quickstart for beginners” (TensorFlow 2021b). Jupyter Notebookin taktia Colabin käyttö toimii hyvin samankaltaisesti kuin aiemmin tässä työssä käsitellyt pilvipalvelut. Käyttäjä voi siis suorittaa koodia osissa ja lisätä tiedostoon tavalliselle

muistiolle tyypillisiä muistiinpano-osioita. Kuvassa 12 havainnollistetaan Colabin Notebook-näkymää ja sen avulla suoritettua kuvien luokittelua. Käytössä olevana datana toimii käsin kirjoitettuja numeroita sisältävä MNIST-tietokanta (Burges, Cortes, & LeCun 2021). Se sisältyy TensorFlow-kirjaston Datasets-kokoelmaan, joka tarjoaa koneoppimisen opetteluun hyödynnettäviä käyttövalmiita dataa.



The screenshot shows a Colab notebook titled "Kokeilu_Tensorflow.ipynb". The code is as follows:

```
[1] import tensorflow as tf

[2] mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [=====] - 0s 0us/step

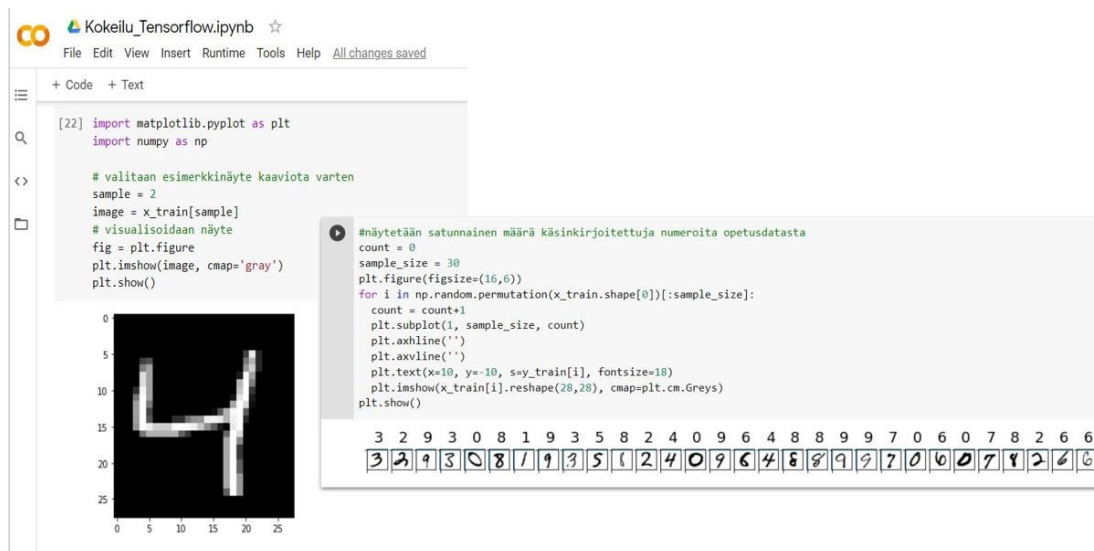
[3] model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10)
])

[4] predictions = model(x_train[:1]).numpy()
predictions

array([[ -0.41547537, -0.20044139,  0.5467129 , -0.161327 , -0.4281724 ,
         0.01574697,  0.14316756,  0.40362215,  0.04882059,  0.20529367]],
      dtype=float32)
```

Kuva 12. Colabin käytännön kokeilua. (TensorFlow 2021b)

TensorFlow-tutoriaalın tarkoitus on luokitella käsin kirjoitetut numerot vastaamaan todellisia numeroita. Kuvassa 13 on havainnollistettu tutoriaalissa käytettävän datan esimerkkejä.



Kuva 13. Datan esimerkkejä visualisoituna.

Googlen palveluissa on muitakin eroja kuin hinta. AI Platform Notebooks ei sisällä rajoituksia suorituksen keston, eikä palvelun skaalautuvuuteen. Sen sijaan Colab in suorituksen kesto aika rajoittuu 12 tuntiin ja se tarjoaa rajoitetusti keskusmuistia eli maksimissaan 13 gigatavun verran. Lisäksi käytettävissä on korkeintaan kaksi ydintä sekä yksi grafiikkaprosessori. (Agrawal 2020.)

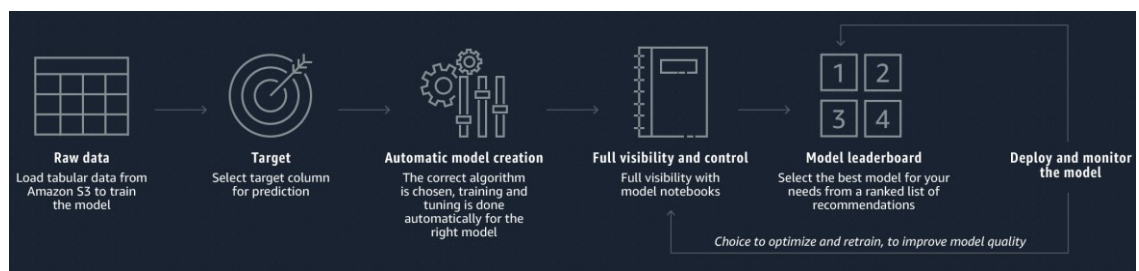
4.1.3 Amazon Web Services

Amazon Web Services eli AWS on julkinen pilvipalvelualusta, joka vastaa toteutukseltaan IaaS-palvelumallia. AWS tarjoaa huomattavasti enemmän palveluja ja ominaisuuksia kuin muut pilvipalvelujen tarjoajat. Laskennan, tallennustilan, tietokantojen ja muiden lisäksi se tarjoaa kattavan määrän koneoppimisen palveluita.

AWS tarjoaa muun muassa Amazon SageMaker -palveluita, jotka mahdollistavat käyttäjän toteuttaa koko prosessi datan käsittelystä koneoppimismallin opettamiseen ja sen avulla ennustamiseen. Lisäksi AWS tarjoaa esimerkiksi AI Services -palvelua, jonka avulla käyttäjä pystyy rakentamaan oppivia sovelluksia ilman minkäänlaista taitoa koneoppimismenetelmien käyttämisestä. Tässä työssä keskitytään kuitenkin vain merkittävimpiin Amazon SageMaker -palveluihin. (AWS 2021a.)

Amazon SageMaker tarjoaa useita koneoppimisen palveluita niihin käyttötarkoituksiin, joissa käyttäjällä itsellään on hyvin vähän tai ei lainakaan kokemusta koneoppimisesta. Yksi näistä on SageMaker JumpStart, joka mahdollistaa koneoppimisen aloittamisen nopeasti ja helposti. Se tarjoaa joukon yksinkertaisesti käyttöönotettavia valmiita ratkaisuja yleisimpiin käyttötapauksiin. Valmiita avoimen lähdekoodin malleja on yli 150 ja käyttäjä voi vapaasti muokata niitä tarpeen vaatiessa.

Toinen helpon aloittamisen mahdollistama palvelu on SageMaker Autopilot. Se rakentaa, opettaa ja sovittaa automaattisesti parhaat koneoppimismallit käyttäjän antamien tietojen ja datan perusteella. Samalla se mahdollistaa käyttäjälle läpinäkyvän ja täyden hallinnan ylläpidon. Kuvassa 14 on kuvattu SageMaker Autopilot -palvelun toimintaa. (AWS 2021b, AWS 2021c.)

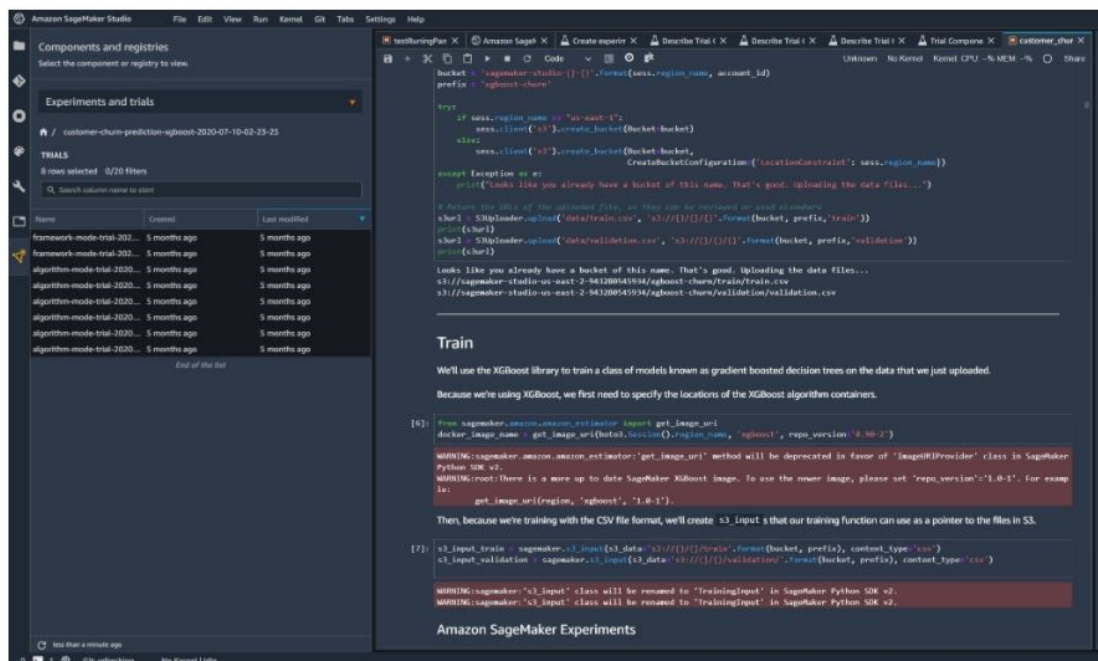


Kuva 14. Amazon SageMaker Autopilot -palvelun toiminta. (AWS 2021c)

Seuraavaksi tutustutaan SageMaker Studioon, joka on tämän työn kannalta olennaisin Amazon SageMakerin tarjoama koneoppimisen palvelu. Se on täysin integroitu kehitysympäristö, joka tuo yhteen laajan joukon koneoppimistarkoituksiin rakennettuja ominaisuuksia (AWS 2017). Sen avulla pystytään valmistelemaan, rakentamaan, opettamaan ja ottamaan käyttöön koneoppimismalleja.

Koneoppimismallin luominen tapahtuu samanlaisen Notebooks-instanssin avulla kuin Microsoft Azuren ja Googlen Cloudin ratkaisuihin. Eli kyseessä on Jupyter Notebook, joka mahdollistaa yksinkertaisen datan analysoinnin ja koodisolujen hallinnan. SageMaker Studioon Notebooks tukee aikaisemmin esiteltyjen välineiden tavoin kaikkia yleisesti käytössä olevia koneoppimisen analytiikkakirjastoja. Lisäksi käytössä on valmiita koneoppimisalgoritmeja, jotka on optimoitu Amazonin alustalle takaamaan muita alustoja tehokkaamman suorituskyvyn.

Amazon SageMaker -palvelun käyttö aloitetaan luomalla AWS-käyttäjätili, jonka jälkeen käyttäjä pääsee kirjautumaan Amazon AWS -konsoliin. Myös Amazonin pilvipalveluissa kaikki resurssien hallinta tapahtuu konsolin kautta. Seuraavaksi avataan Amazon SageMaker -palvelun konsoli koneoppimispalvelujen valikosta ja luodaan uusi Notebooks-instanssi, jonka kautta avataan SageMaker Studio. Koska SageMaker Studio on Jupyter Notebookin isännöimä, toteutus on käyttöliittymältään ja toiminnaltaan hyvin samankaltainen kuin Microsoftin AMLS ja Googlen AI Platform Notebooks. AWS:n tarjoama ratkaisu on kuitenkin hieman modernisoidumpi ja persoonisempi versio Jupyter Notebookin oletuskäyttöliittymästä. Tätä on havainnollistettu kuvassa 15.



Kuva 15. Amazon SageMaker Studio (AWS 2017)

4.2 Toiminnallisten kokeilujen tulokset

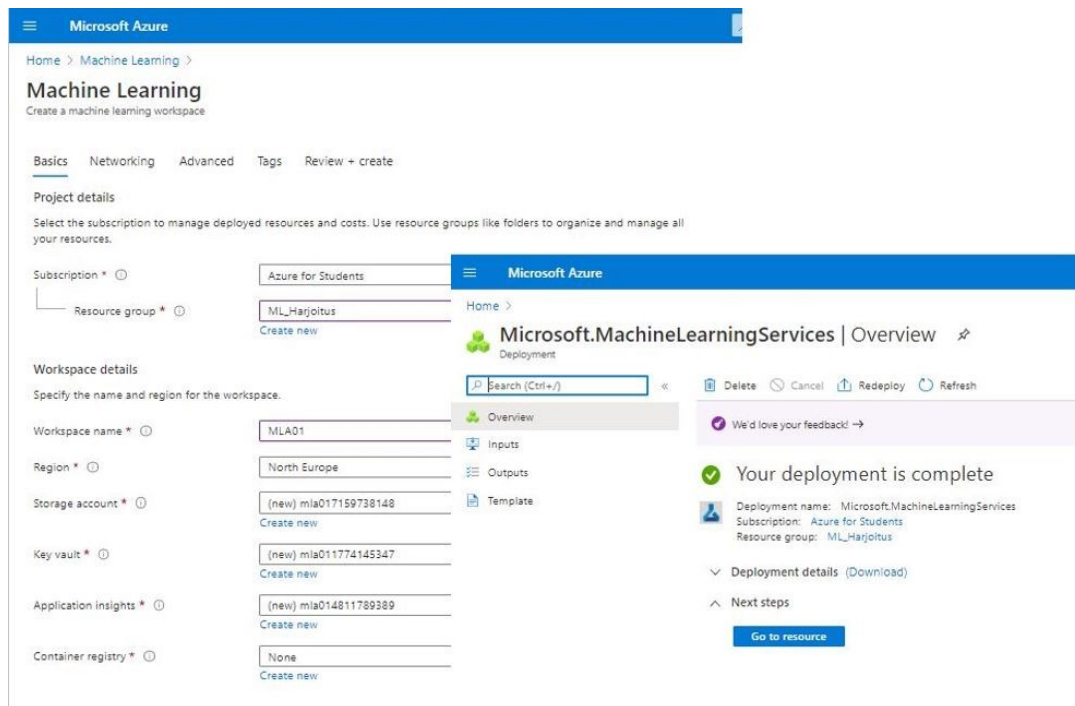
Seuraavaksi tässä työssä käydään läpi tuloksia, joita saatiin tutkimalla kolmen suuren pilvipalveluntarjoajan MLaaS-palveluja. Työssä käytettiin Samkin opiskelijoiden käyttöön tarkoitettuja palvelupaketteja ja käyttölisenssejä, joten työssä ei oteta tarkkaa kantaa eri toimijoiden kustannuksiin. Taulukossa 1 on havainnollistettu pilvipalvelu- alustojen teknisiä eroja ja yhtäläisyyksiä koneoppimisen palveluissa.

Taulukko 1. Eri pilvipalvelualustojen vertailua

	Microsoft Azure Machine Learning	Google Cloud Machine Learning	Amazon Web Services Machine Learning
Käyttöliittymä	Azure Machine Learning	Google Cloud AI Platform (ei huomioida Google Colaboratorya)	Amazon Machine Learning Console
Sisäänrakennetut valmiit algoritmit	Ei	Kyllä	Kyllä
Automaattinen algoritmien ehdotus	Ei	Kyllä	Kyllä
Tuetut kehikset	TensorFlow, Scikit-learn, Microsoft Cognitive Toolkit, Spark ML	TensorFlow, Scikit-learn, Keras, XGBoost	TensorFlow, MXNet, Keras, Gluon, Pytorch, Caffee2, Chainer, Torch
Datan sijaintivaatimukset	Pienten datamäärien tuonti onnistuu kolmannelta osapuolelta, mutta isommat datamäärien tulee olla tallennettuna Azureen.	Data täytyy olla tallennettuna ja mallien alustettuna Google Cloud Storage -palvelussa.	Datan täytyy olla Amazon Web Services -alustalla tallennettuna, jotta sitä voidaan käyttää koneoppimisen palveluissa.
Hinnoittelu	AML:n käytön kuukausimaksu 9,99\$. Lisäksi jokaisesta käytetystä tunnista 1\$ lisäveloitus.	Mallien opettamiseen tuntihinta noin 0,54\$ riippuen koneoppimisen opetus yksiköistä. Tämän lisäksi ennakointimaksuja noin 0,10\$-0,44\$/h.	Datan analysointiin ja mallien suorittamiseen tuntihinta noin 0,42\$. Tämän lisäksi ennakointimaksuja noin 0,001\$-0,10\$/h.
Muut veloitteet	Vaaditaan Azure-tilin maksullinen versio.	Vaaditaan Google Cloud Platform -tili.	Vaaditaan AWS-tilin maksullinen versio. Datan säilyttämisestä AWS-palvelussa laskutetaan erikseen.

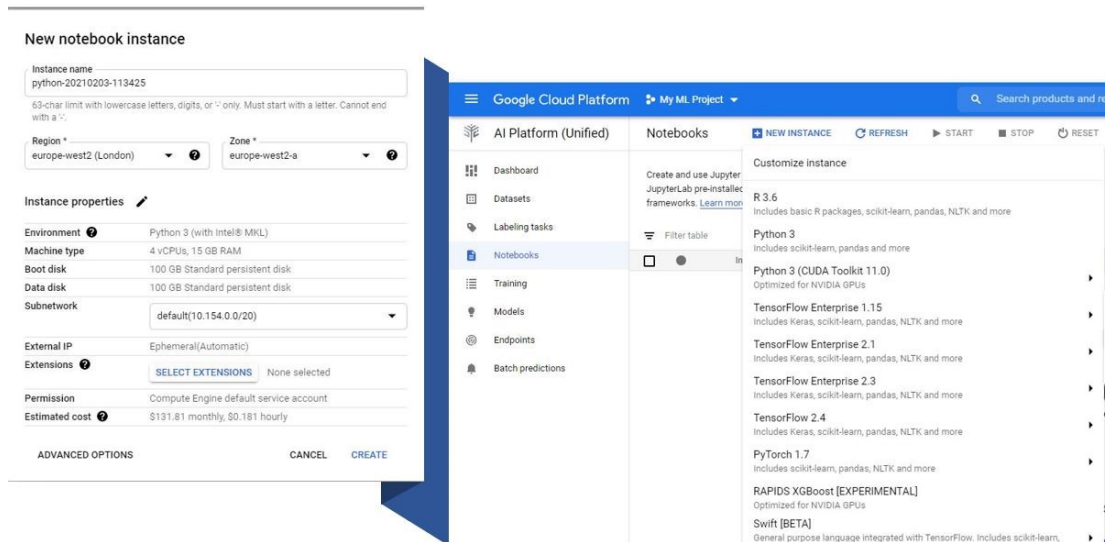
4.2.1 Käyttöönotto

Azure Machine Learning -pilvipalvelun käyttöönotto vaatii Microsoft käyttäjätilin perustamisen. Tämän jälkeen käyttäjän tulee tehdä Azure-tilaus perustamalleen tilille. Käyttäjän on myös mahdollista hyödyntää Microsoftin tarjoamaa ilmaista kokeilujaksoa. Kun tilaus on kunnossa, käyttäjän tulee luoda uusi työtila (Machine Learning Workspace) ja varata sille tarvittava määrä tallennustilaa Azureen. Työtilan luominen vaatii hieman perehtymistä, sillä käyttäjän tulee osata määritellä tarvittavat tekniset tiedot oikein. Näitä ovat esimerkiksi "Subscription", "Recourse group" ja "Region". Microsoftin dokumentaatio kattaa kuitenkin jokaisen kohdan selityksen siltä osin, että käyttäjä saa luotua työtilan onnistuneesti. Kuva 16 kuvaa näitä vaihteita.



Kuva 16. Machine Learning Workspace -työtilan luominen.

Google Cloudin molempien aikaisemmin työssä esiteltyjen palveluvaihtoehtojen käyttöönotto vaatii Google-käyttäjätilin. Colab:n käyttö onnistuu heti käyttäjän kirjaututtua tunnuksillaan selaimessa toimivaan palveluun. Mikäli käyttäjä haluaa suorittaa koneoppimista GCP:n kautta, tulee konsoliin kirjautumisen jälkeen luoda tarvittaessa uusi projekti. Tähän asti palvelu on käyttäjälle ilmainen, mutta seuraavaksi käyttäjän tulee luoda Notebooks-instanssi, jonka käytöstä maksetaan kuluneiden resurssien mukaan. Myös Google tarjoaa ilmaisen kokeilujakson, jota käyttäjä voi halutessaan hyödyntää. Seuraavaksi käyttäjä siis valitsee, minkälaisen instanssin hän haluaa luoda. Kuvassa 17 näkyy muutamia instanssivaihtoehtoja. Valikoimasta löytyy muitakin kuin tässä työssä käytetty Python-instanssi. Instanssin luomisen yhteydessä tulee Azuren tapaan määrittellä lähes vastaavat tekniset tiedot.



Kuva 17. Notebooks-instanssin luonti Google Cloud Platformissa.

Amazon Web Services -pilvipalvelujen käyttöönotto vaatii AWS-käyttäjätilin luomisen. AWS-käyttäjätilin luominen on huomattavasti työläämpää kuin Azure- tai Google Cloud -tilin luominen. Laskutus sopimuksen lisäksi tarvitaan vahvistus puhelimitse, jotta AWS pystyy turvallisesti todentamaan käyttäjän tilin luomisen yhteydessä. Käyttäjä saa uniikin ID-tunnuksen, jota tarvitaan käyttöönoton seuraavassa vaiheessa. Tässä vaiheessa käyttäjä on saanut kertakirjautumistunnuksen, jonka avulla pystyy käyttämään kaikkia AWS:n tarjoamia palveluita, kuten SageMaker Studiota. Eli käyttäjä on kirjautuneena niin sanottuna ”root”-käyttäjänä. AWS kuitenkin suosittelee, ettei tätä käyttäjää käytettäisi päivittäisiin pilvipalvelunkäyttötarkoituksiin, vaan siihen luotaisiin yksilöllinen IAM (Identity and Access Management) eli järjestelmävalvojan käyttäjätunnus. Tässä työssä sen luomista ei käydä läpi, mutta siihen löytyy tarkat ohjeet AWS-dokumentoinnista. Käyttöönoton viimeinen vaihe eli instanssin luominen vaatii jälleen käyttäjältä tutustumista teknisiin tietoihin, jotka tulee täyttää oikein. Kuvassa 18 on esitetty instanssin luomiseen tarvittavia tietoja. Käyttöönoton jälkeen käyttäjä pääsee SageMaker Studioon, jonka käytön kustannukset perustuvat käytettyihin sekunteihin. AWS tosin tarjoaa Microsoftin ja Googlen tapaan ilmaisen ko-keilujakson, mutta alkaa sen jälkeen laskuttaa käyttösekuntien mukaan.

The screenshot shows the 'Create notebook instance' page in the Amazon SageMaker console. The left sidebar contains navigation options: Dashboard, Notebook instances (selected), Lifecycle configuration, Jobs, Resources, Models, Endpoint configuration, and Endpoints. The main content area is titled 'Create notebook instance' and includes a brief description of SageMaker's pre-built notebook instances. Below this is the 'Notebook instance settings' section, which contains several configuration options:

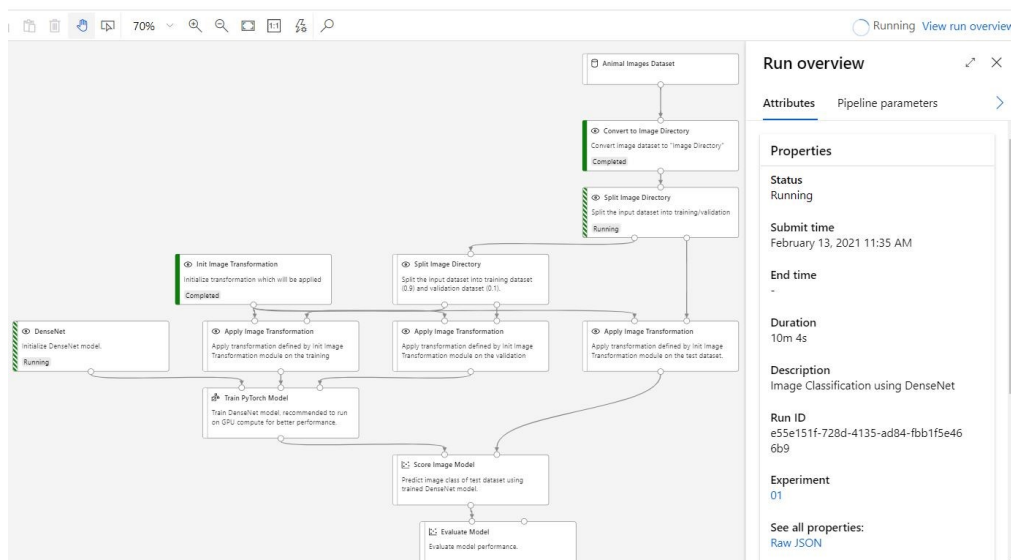
- Notebook instance name:** A text input field with a note: 'Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.'
- Notebook instance type:** A dropdown menu currently set to 'ml.t2.medium'.
- IAM role:** A dropdown menu currently set to 'AmazonSageMaker-ExecutionRole-20180309T130111'. A note states: 'Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the AmazonSageMakerFullAccess IAM policy attached.'
- VPC - optional:** A dropdown menu currently set to 'No VPC'. A note states: 'Your notebook instance will be provided with SageMaker provided internet access because a VPC setting is not specified.'
- Lifecycle configuration - optional:** A dropdown menu currently set to 'No configuration'. A note states: 'Customize your notebook environment with default scripts and plugins.'
- Encryption key - optional:** A dropdown menu currently set to 'No Encryption'. A note states: 'Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN.'

Kuva 18. Notebook-instanssin luominen Amazon SageMaker -palvelussa (Fan 2018)

4.2.2 Koneoppimisen suorittaminen

Kaikilla tässä työssä tutkituilla pilvipalveluntarjoajilla oli käytössä Jupyter Notebooksin isännöimä käyttöliittymä, jonka kautta päästiin kokeilemaan koneoppimisen menetelmiä Python-ohjelmointikielellä. Koodin suoritus oli kaikissa todella nopeaa ja tehokasta, eikä työkalujen käyttäminen ollut kovin vaativaa. Kaikki kolme palveluntarjoajaa mahdollistavat pääsyn kokeilemaan koneoppimista pilvessä.

Vaikka Microsoft Azure ja Amazon tarjoavat useita eri tapoja suorittaa koneoppimista, ilman minkäänlaista aihealueen osaamista ja tietämystä koneoppimisesta. Yleensä on kuitenkin välttämätöntä ymmärtää teknisesti prosessin jokainen vaihe datan esiprosessoinnista opettamiseen, jotta koneoppimismallia voidaan pitää luotettavana. Lisäksi en itse pidä kovin hyödyllisenä Azuren “vedä ja pudota” ratkaisua. Koko prosessi on myös melko hidas toteuttaa raahattavien osien välillä, eikä suoritusnopeus ole mitenkään verrattavissa Notebooks-palvelun nopeuteen ajaa annettuja algoritmeja. Kuvassa 19 on esimerkki Azure Machine Learning studion suunnittelutyökalun valmiista luokittelumallista.



Kuva 19. Azure Machine Learning studion suunnittelutyökalu.

4.2.3 Koneoppimismallin tallentaminen

Jokainen tässä työssä läpi käyty pilvipalveluntarjoaja tukee Skicit-learn-kirjastoa, joten opetettu koneoppimismallin tallentaminen tiedostoon ja tiedostosta lataaminen on mahdollista. Tämä mahdollistaa siis kolmannen osapuolen hyödyntämisen. Käyttäjä voi ensin opettaa mallin esimerkiksi Googlen tarjoamassa palvelussa ja myöhemmin hyödyntää samaa opetettua mallia Microsoftin tarjoamassa palvelussa. Kuvassa 20 on esimerkki mallin tallentamisesta Colabissa. Opetusmallin lataaminen tiedostosta on havainnollistettu kuvassa 21. Tässä toiminto on toteutettu Pickle-kirjaston avulla.

```

#Jaetaan data opetus- ja testidataan
[6] X_train, X_test, y_train, y_test = train_test_split(X, y

#Tehdään luokittelumalli
#model = DecisionTreeClassifier(max_depth = 2).fit(X_train, y_train)
model = SVC(kernel='linear', C=1.0)
model.fit(X_train, y_train)

#Tallennetaan malli
#Tallennetaan malli
del.predict(X_test)

#Tallennetaan malli
#Tallennetaan malli
delumatriisi
matrix(y_test, predictions)
arkkuus testidatalle =', round(accuracy_

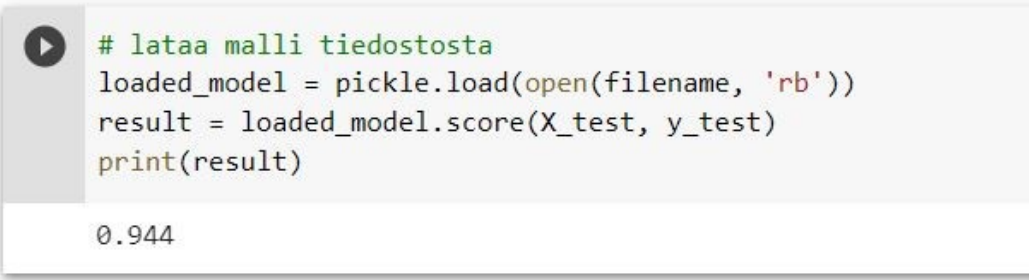
testidatalle = 0.94

import pickle

# tallenna malli
filename = 'finalized_model.sav'
pickle.dump(model, open(filename, 'wb'))

```

Kuva 20. Opetusmallin tallentaminen tiedostoon ja syntyneen tiedoston sijainti.



```
# lataa malli tiedostosta
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.score(X_test, y_test)
print(result)
```

0.944

Kuva 21 Opetusmallin lataaminen tiedostosta.

Lisäksi pilvipalvelussa tehdyn koodin tallentaminen tiedostoon onnistuu kaikissa aikaisemmin käsitellyissä pilvipalveluissa. Koodin saa ladattua esimerkiksi suoraan käyttäjän omalle koneelle Python-tiedostona (.py) tai Jupyter Notebook -tiedostona (.ipynb). Tämän saa tehtyä suoraan käyttöliittymän valikoista ilman koodin ajamista.

5 POHDINTA

Opinnäytetyöni keskeisimpänä tavoitteena oli saada riittävän kattava kuva siitä, miten koneoppimista voidaan tuottaa pilvipalveluna. Olennaista on kuitenkin ensin käydä läpi, mitä pilvipalvelut ja koneoppiminen tarkoittavat. Molempien aihealueiden määrittely lyhyesti ja yksiselitteisesti on usein ongelmallista, minkä takia keskityin antamaan hyvän ja selkeän yleiskuvan kustakin osa-alueesta.

Pilvipalveluihin tutustuessani opin, että pilvipalvelujen käyttöönotto on usein iso ja monimutkainen prosessi, jossa organisaation tulee tarkasti miettiä eri mahdollisuuksien etuja ja mahdollisia haittoja. Yrityksen on tarkkaan kartoitettava, onko toiminnan siirtäminen pilveen edes kannattavaa. Monesti kustannusten lisäksi suurin ongelma on datan sijainti, ja siihen liittyvät turvallisuusrajoitteet. Esimerkiksi koneoppimisessa datan sijainnin tulee lähes aina olla pilvipalveluntarjoajalla. Vaikka Microsoft Azure mahdollistaa pienten datamäärien tuonnin kolmannelta osapuolelta, todellisuudessa usein on tarve käyttää suuria datamääriä, jolloin datan on oltava tallennettuna Azuren resursseihin.

Haastavimmaksi koin ”helppona ja yksinkertaisena” markkinoidun Azuren ”vedä ja pudota” ratkaisun. Sitä voisi hyödyntää demoamistarkoituksessa tai koneoppimisen toimintamallin visualisoinnissa, mutta muussa tapauksessa koen sen melko tarpeettomaksi koneoppimisen työkaluksi. Mietityttämään jäi myös AWS:n tarjoamat SageMaker JumpStart sekä SageMaker Autopilot. Näissä tapauksissa palvelu sovitaa automaattisesti käyttäjän tilanteeseen parhaiten sopivan valmismallin. On kuitenkin selvää, että mikäli käyttäjä haluaa saada varmistuksen koneoppimismallin luetettavuudesta, on hänen käytävä valmismallin vaiheet perusteellisesti läpi. Näiden kahden palvelun käytöstä saatu ajallinen ja laadullinen hyöty saattaa monesti jäädä hyvinkin pieneksi.

Colabin käyttö oli muihin työssä tutustuttuihin pilvipalveluihin verrattuna huomattavasti yksinkertaisempaa ja nopeampaa. Colab soveltuu erinomaisesti itsenäiseen koneoppimisen harjoittamiseen, mutta mm. rajoitustensa takia se ei todennäköisesti ole IT-yrityksille paras vaihtoehto. Mikäli tarkoituksena on harjoittaa yrityksen toimintaa pilvessä, hyvä valinta on ottaa käyttöön jokin tässä työssä esitellyistä PaaS-palvelumallin ratkaisuista.

Tulevaisuutta ajatellen näyttää siltä, että pilvipalvelujen käyttö on tullut jäädäkseen. Samalla uskon, että koneoppimista hyödyntävien ratkaisujen kysyntä jatkaa kasvuaan yhä useammilla toimialoilla. Tästä syystä koneoppimisen toteuttaminen pilvipalveluna tuntuu toimivalta ratkaisulta, jolle löytyy kysyntää jatkossa yhä enemmän.

LÄHTEET

Accenture 2020. Cloud Computing. Viitattu 21.11.2020.

<https://www.accenture.com/us-en/insights/cloud-computing-index>

Alpaydin, E. 2014. Introduction to machine learning. Third edition. Cambridge, Massachusetts: MIT Press. Viitattu 26.12.2020. <https://ebookcentral.proquest.com/lib/samk/reader.action?docID=3339851>

Agrawal, A. 2020. Google Colaboratory vs Google Cloud AI Platform Notebooks. Viitattu 3.2.2021. <http://gcp.live/google-colaboratory-vs-google-cloud-ai-platform-notebooks/>

AWS 2021. Machine Learning on AWS. Viitattu 13.2.2021. https://aws.amazon.com/machine-learning/#Explore_AWS_Machine_Learning_services

AWS 2021. Getting Started with Amazon SageMaker. Viitattu 13.2.2021. <https://aws.amazon.com/sagemaker/getting-started/>

AWS 2021. Amazon SageMaker Autopilot. Viitattu 13.2.2021. <https://aws.amazon.com/sagemaker/autopilot/>

AWS 2017. Introduction Amazon SageMaker. Viitattu 13.2.2021. <https://aws.amazon.com/about-aws/whats-new/2017/11/introducing-amazon-sagemaker/>

Beklemysheva, A. 2021. Why Use Python for AI and Machine Learning? Viitattu 15.3.2021. <https://steelkiwi.com/blog/python-for-ai-and-machine-learning/>

Brownlee, J. 2020. Supervised and Unsupervised Machine Learning Algorithms. Viitattu 1.1.2021. <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>

Burges, C., Cortes, C. & LeCun, Y. 2021. The MNIST Database of handwritten digits. Viitattu 3.2.2021. <http://yann.lecun.com/exdb/mnist>

Chopra, R. 2017. Cloud Computing: An Introduction. Bloomfield: Mercury Learning & Information. Viitattu 21.11.2020. <https://ebookcentral.proquest.com/lib/samk/reader.action?docID=4895092>

Colab 202. Welcome to Colaboratory. Viitattu 3.2.2021. <https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=OwuxHmxllTwN>

Fan, L. 2018. Viitattu.25.2.2021. <https://aws.amazon.com/blogs/machine-learning/customize-your-amazon-sagemaker-notebook-instances-with-lifecycle-configurations-and-the-option-to-disable-internet-access/>

Google Cloud 2021. About Google Cloud services. Viitattu 3.2.2021. <https://cloud.google.com/docs/overview/cloud-platform-services>

Google Cloud 2021. Introduction to AI Platform Notebooks. Viitattu 3.2.2021. <https://cloud.google.com/ai-platform/notebooks/docs/introduction>

Harkut D, 2019. Cloud Computing - Technology and Practices. IntechOpen. Viitattu 3.1.2021. <https://www.intechopen.com/books/cloud-computing-technology-and-practices/introductory-chapter-cloud-computing>

Hurwitz, J. 2012. Hybrid Cloud for Dummies. Hoboken, New Jersey: Wiley. Viitattu 19.12.2020. <http://ebookcentral.proquest.com/lib/samk/detail.action?docID=818070>

Jupyter 2021. Jupyter Notebook. Viitattu 3.2.2021. <https://jupyter.org/>

Kavis, M. 2014. Architecting the cloud: Desing decisions for cloud computing service models (SaaS, PaaS and IaaS). Hoboken, New Jersey: Wiley. Viitattu 19.12.2020. <http://ebookcentral.proquest.com/lib/samk/detail.action?docID=1605593>

Keras 2021. Keras Simple.Flexible. Powerful. Viitattu 8.1.2021. <https://keras.io/>

Kohonen, T. & Honkela, T. 2007. Kohonen network. Teoksessa Scholarpedia journal. Viitattu 12.2.2021. http://www.scholarpedia.org/article/Kohonen_network

Lee, W. 2019. Python Machine Learning. Hoboken, New Jersey: Wiley. Viitattu 2.1.2021. <https://ebookcentral.proquest.com/lib/samk/reader.action?docID=5747364>

Matplotlib 2021. Gallery. Viitattu 7.1.2021. <https://matplotlib.org/gallery/index.html>

Microsoft 2021. What is Azure? Viitattu 26.1.2021. <https://docs.microsoft.com/en-us/learn/modules/intro-to-azure-fundamentals/what-is-microsoft-azure>

Microsoft 2021. What is Azure Machine Learning? Viitattu 26.1.2021. <https://docs.microsoft.com/en-us/azure/machine-learning/overview-what-is-azure-ml>

Microsoft 2021. What is Azure Machine Learning Studio? Viitattu.28.1.2021. <https://docs.microsoft.com/en-us/azure/machine-learning/overview-what-is-machine-learning-studio>

National Institute of Standards and Technology 2019. NIST Cloud Computing Program – NCCP. Viitattu 21.11.2020. <https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp>

NumPy 2021. NumPy. Viitattu 7.1.2021. <https://numpy.org/>

Pandas 2021. About pandas. Viitattu 7.1.2021. <https://pandas.pydata.org/about/index.html>

Python 2021. What is Python? Executive Summary. Viitattu 6.1.2021. <https://www.python.org/doc/essays/blurb/>

Ranger, S. 2018. What is cloud computing? Everything you need to know about the cloud explained. Viitattu 21.11.2020. <https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud/>

Ravichandiran, S. 2018. Hands-on reinforcement learning with python: Master reinforcement learning and deep reinforcement learning by building intelligent app. Birmingham, London; Mumbai: Packt. Viitattu 3.1.2021. <https://ebookcentral.proquest.com/lib/samk/reader.action?docID=5439844&query=>

Scikit-learn 2021. scikit-learn Machine learning in Python. Viitattu 8.1.2021. <https://scikit-learn.org/stable/>

Scikit-learn 2021. sklearn.datasets.make_blobs. Viitattu 30.1.2021. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html

Seikku, E. 2018. Mikä ihmeen tekoäly, koneoppiminen ja ennakoiva analytiikka? Viitattu 26.12.2020. https://www.tivi.fi/kumppaniblogit/hewlett_packard_enterprise/mika-ihmeen-tekoaly-koneoppiminen-ja-ennakoiva-analytiikka/

Shaptunova, Y. 2020. 4 Best Cloud Deployment Models Overview. Viitattu 19.12.2020. <https://www.sam-solutions.com/blog/four-best-cloud-deployment-models-you-need-to-know/>

Simplilearn 2020. What is Tensorflow: Deep Learning Libraries and Program Elements Explained. Viitattu 12.1.2021. <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow>.

TensorFlow 2021. TensorFlow is an end-to-end open source platform for machine learning. Viitattu 12.1.2021. <https://www.tensorflow.org/>

TensorFlow 2021. TensorFlow 2 quickstart for beginners. Viitattu 3.2.2021. <https://www.tensorflow.org/tutorials/quickstart/beginner>

Tutorialspoint 2021. Scikit Learn – Introduction. Viitattu 8.1.2021. https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm

Yim, A., Chung, C. & Yu, A. 2018. Matplotlib for Python Developers. Packt Publishing. Viitattu 7.1.2021. <https://ebookcentral.proquest.com/lib/samk/reader.action?docID=5371691&query=>

Wolhuter, S. 2018. Machine Learning as a Service — what is it and how can it help your business? Viitattu 12.2.2021. <https://thebrainfiles.wearebrain.com/machine-learning-as-a-service-what-is-it-and-how-can-it-help-your-business-3310ac4f0b25>

KOODIESIMERKKI LUOKITTELUSTA

```
#Luokittelu, jossa n erilaista luokitteluarvoa
#Otetaan tarvittavat analytiikkakirjastot käyttöön
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split

#Luokkien ja rivien lukumäärä
classes = 5
rowcount = 1000

#Generoidaan data
X, y = make_blobs(n_samples=rowcount, centers=classes, n_features=2, random_state=41)

#Visualisoidaan data
data = pd.DataFrame(dict(x=X[:,0], y=X[:,1], luokka=y))
colors = {0:'red', 1:'blue', 2:'green', 3:'orange', 4:'purple'}
fig, ax = plt.subplots()
grouped = data.groupby('luokka')
for key, group in grouped:
    group.plot(ax=ax, kind='scatter', x='x', y='y', label=key, color=colors[key])
plt.show()

#Jaetaan data opetus- ja testidataan
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)

#Tehdään luokittelumalli, käyttäen tukivektorikonetta
model = SVC(kernel='linear', C=1.0)
model.fit(X_train, y_train)
predictions = model.predict(X_test)

#Tehdään luokittelumatriisi
cm = confusion_matrix(y_test, predictions)
print('Ennustustarkkuus testidatalle =', round(accuracy_score(y_test, predictions), 2))
```