Tampereen ammattikorkeakoulu

BigQuery ML: Koneoppimismallien luominen standardisoidulla SQL-syntaksilla

Peetu Juuti

OPINNÄYTETYÖ
Huhtikuu 2020

Tieto- ja viestintätekniikka
Tietoliikennetekniikka ja tietoverkot

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintätekniikka
Tietoliikennetekniikka ja tietoverkot

JUUTI PEETU:
BigQuery ML: Koneoppimismallien luominen standardisoidulla SQL-syntaksilla

Opinnäytetyö 40 sivua, joista liitteitä 6 sivua
Huhtikuu 2020

Tämän opinnäytetyön tavoitteena oli perehtyä BigQuery ML -alustan käyttöön ja tutkia sen toimivuutta ja ominaisuuksia koneoppimistyökaluna. Työssä pohditaan myös alustan hyödyllisyyttä ja sen käyttömahdollisuuksia liiketoiminnan näkökulmasta.

Koneoppiminen vaatii tekijältä ohjelmointitaitoa ja koneoppimistyökalujen osaamista. Tämä rajoittaa ratkaisukehityksen yrityksessä vain pienelle henkilömäärälle, mikä ei välttämättä ole kovin tehokasta tai tuotteliasta liiketoiminnan kannalta. BigQuery ML -koneoppimistyökalun tarkoituksena on antaa data-analytiikoille tai muille SQL-ammatinharjoittajille mahdollisuus tehdä koneoppimismalleja valmiiksi olemassa olevalla tietotaidolla.

Opinnäytetyössä tutkitaan alustan kokonaisuutta ja sen koneoppimisympäristöä sekä niihin liittyvää dokumentaatiota ja algoritmien selityksiä. Työssä käsitellään koneoppimistyöprosessin eri vaiheet tutkimalla, analysoimalla ja mallintamalla dataa käyttämällä prosessissa BigQueryä keskitettynä datavarastona ja koneoppimismallien luomisalustana. Työssä tutustutaan myös yleisesti datatieteen, koneoppimisen ja muiden aiheeseen liittyvien osa-alueiden käsitteisiin, työprosesseihin ja käyttötapoihin. Työprosesseja ja käyttötapoja havainnollistetaan työssä visualisoivilla kuvilla ja esimerkeillä.

Työn tuloksissa tuodaan esille BigQueryn hyötyjä ja esitetään sen koneoppimisympäristön etuja sekä puutteita. Kehitysehdotuksena koneoppimistyöprosessissa voitaisiin käyttää lähtökohtaisesti BigQueryä, mutta soveltaa muita koneoppimistyökaluja tarpeen mukaan.

Asiasanat: BigQuery ML, koneoppimistyökalu, SQL, datavarasto

ABSTRACT

Tampere University of Applied Sciences
Information and Communications Technology
Telecommunications and Networks

JUUTI PEETU:
BigQuery ML: Creating Machine Learning Models with Standardized SQL syntax

Bachelor's thesis 40 pages, appendices 6 pages
April 2020

The purpose of this thesis was to study the functionality and features of BigQuery ML as a machine learning tool. The thesis also discusses the usefulness and application of BigQuery from a business perspective.

Machine learning requires extensive programming skills and knowledge of machine learning tools. This limits the solution development in a company to only a small group of people, which may not be efficient or productive for a business. The purpose of BigQuery ML as a machine learning tool is to enable data analysts or other SQL practitioners to build machine learning models with already existing skills.

The thesis examines the platform as a whole and its machine learning environment, as well as related documentation and explanations of algorithms. It also covers all the different stages of machine learning process by exploring, analyzing and modeling data using BigQuery as a centralized data warehouse and as a platform for creating machine learning models. The thesis also explores concepts, work processes and use cases of data science, machine learning and other related areas. Work processes and use cases are illustrated in the thesis with visualizing images and examples.

The results of the thesis highlight the benefits of BigQuery in its entirety and the advantages and disadvantages of its machine learning environment. As a development proposal, BigQuery could be used primarily for machine learning, but when needed, utilizing other machine learning tools to get more desirable results.

Key words: BigQuery ML, machine learning tool, SQL, data warehouse

SISÄLLYS

LYHENTEET JA TERMIT

API	Application Programming Interface, ohjelmointirajapinta
BI	Business Intelligence, liiketoimintatiedon hallinta
BQML	BigQuery Machine Learning
dataskaalautuvuus	järjestelmän riittävä kapasiteetti datan kasvaessa
ennustemuuttuja	muuttuja, jonka arvo ei riipu toisesta muuttujasta
framework	järjestelmän perustana oleva rakenne
hajonta	mittojen eroavaisuus toisistaan
harjoittaminen	koneoppimismallin opettamisen prosessi
klusterointi	datan ryhmittely eri osiin yhtäläisyyksien perusteella
kysely	query, tietokannasta hakeminen
lineaarinen regressio	menettelytapa, jossa tutkitaan vaste- ja ennustemuuttujan suhdetta toisiinsa
logistinen regressio	tietyn luokan todennäköisyyden mallintaminen
luokittelu	menettelytapa, jossa data asetetaan luokkaan johon se kuuluu
MAE	Mean Absolute Error, absoluuttinen keskivirhe
mallintaminen	matemaattinen esitys jostain oikean elämän prosessista
ML	Machine Learning, koneoppiminen
MSE	Mean Squared Error, keskineliövirhe
NoOps	No Operations, automatisoitu ympäristö
piirreoppiminen	joukko tekniikoita, jossa järjestelmä automaattisesti löytää esitystavat piirteiden havaitsemiseen tai luokitteluun datasta
REST API	HTTP-pyyntöjä käyttävä sovellusohjelmointirajapinta
SQL	Structured Query Language, standardisoitu kyselykieli
syntaksi	ohjelmointikielen kielioppisäännöt
tiheyden ennakointi	arvio muuttujan mahdollisten arvojen jakaumasta.
UI	User Interface, käyttöliittymä
validointi	toimenpide, jossa varmistetaan jonkin asian tarkkuus
vastemuuttuja	muuttuja, jonka arvo riippuu toisesta muuttujasta

1 JOHDANTO

Teknologian ja internetin kasvaessa, dataa kerätään nykypäivänä valtavia määriä. Yritykset käyttävät tätä dataa analytiikkaan ja koneoppimiseen, joilla voidaan edistää liiketoimintaa. Resurssit eivät usein kuitenkaan riitä kaiken datan hyödyntämiseen, koska koneoppiminen valtuutetaan yrityksessä tyypillisesti vain pienelle määrälle henkilöitä.

Tässä opinnäytetyössä tutkitaan Googlen BigQuery-analytiikkadatavarastoa ja sen BigQuery ML -koneoppimisympäristöä, jonka tavoitteena on mahdollistaa koneoppimisen käyttö suuremmalle joukolle ihmisiä ilman syvempää koneoppimisymmärrystä. Opinnäytetyön tarkoituksena on tutustua BigQueryyn ja sen koneoppimisympäristön käyttöön sekä tutkia sen toimivuutta ja käyttömahdollisuuksia.

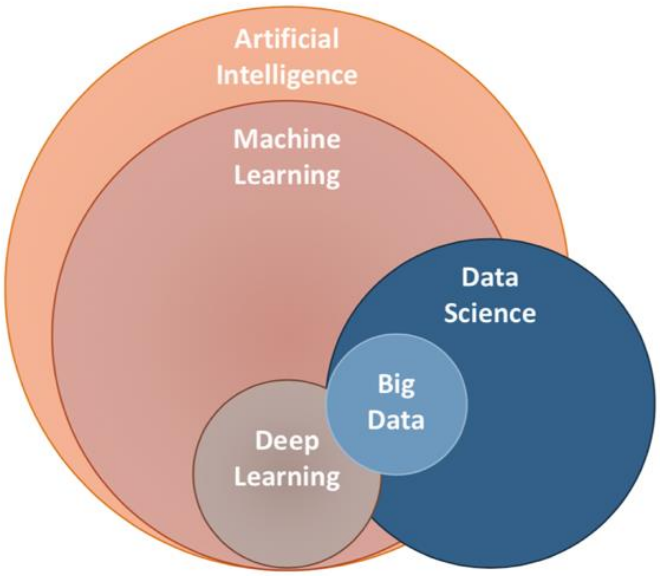
Opinnäytetyön alussa luvussa kaksi käydään läpi teoriaosuutta, jossa perehdytään datatieteen, koneoppimisen ja muiden olennaisten aihealueiden käsitteisiin. Teoriaosuudessa selitetään myös aihealueiden työvaiheita ja käyttötarkoituksia. Kolmannessa luvussa tutustutaan BigQueryyn analytiikkadatavarastona ja koneoppimistyökaluna. Luvussa opetellaan myös käyttämään BigQueryä ja tutustutaan sen perustoimintoihin. Luvuissa neljä ja viisi käydään läpi koneoppimistyöprosessiin sisältyvät vaiheet käyttämällä BigQueryä keskitettynä datavarastona ja koneoppimistyökaluna. Opinnäytetyön lopussa pohditaan vielä alustan toimivuutta, hyödyllisyyttä sekä mahdollisia integrointiratkaisuja.

2 DATATIEDE, KONEOPPIMINEN JA NIIDEN OSAJOUKOT

2.1 Yleistä

Big data, datatiede, koneoppiminen, neuroverkot ja syväoppiminen ovat kaikki termejä, jotka ovat kasvattaneet suosiota lähivuosina nopeaa vauhtia. Yhä useampi yritys on alkanut tajuamaan, kuinka arvokasta big data on, ja miten sitä voidaan hyödyntää datatieteen sekä koneoppimisen avulla. Tätä prosessia kutsutaan usein nimellä Business Intelligence eli liiketoimintatiedon hallinta. Nykyään kaikki hyvin suunnitellut bisnesmallit sisältävät tarkasti määritellyn suunnitelman datan käytöstä ja hyödyntämisestä joillain tavalla organisaation sisällä. Yrityksien uudesta halusta hyödyntää dataa on aiheutunut myös suuri tarve datatieteilijöille ja data-analyytikoille työmarkkinoilla. (Srinidhi Sunny 2019.)

Usein näiden termien merkitys voi olla kuitenkin hieman epäselvää ja ne menevät helposti sekaisin. Vaikka big data, datatiede, koneoppiminen, neuroverkot ja syväoppiminen ovat kaikki linkitetty jollain tavalla toisiinsa (kuva 1), niillä on kaikilla omat toiminnot ja tarkoituksensa. Tässä luvussa käydään läpi kyseiset termit, jotta voidaan saada selkeämpi kuva, mitä datatiede, koneoppiminen ja muut niihin osajoukot tarkoittavat ja oikeasti pitävät sisällään.



KUVA 1. Diagrammi miten tekoäly, big data, datatiede, koneoppiminen ja syväoppiminen ovat yhdistettyinä toisiinsa (GMG Group n.d.)

2.2 Big data

Big data on termi, jota käytetään laajoista datajoukoista, jotka vaativat skaalautuvan arkkitehtuurin, tehokasta tallennustilaa, manipulaatiota ja analysointia. Big data voidaan määritellä myös datan määräksi, joka on suurempi kuin yksi petatavu eli miljoona gigatavua. Tämän kokoinen datan määrä saadaan kerättyä esimerkiksi sosiaalisen median, älypuhelimien tai muiden kuluttajatuotteiden kautta. Laajoissa datajoukoissa on valtava määrä mahdollisuuksia organisaatioille, joilla on riittävää tietotaitoa ja teknologiaa muuntaa dataa uusien asioiden ymmärtämiseen ja päätösten tekoon. (University of Wisconsin n.d.)

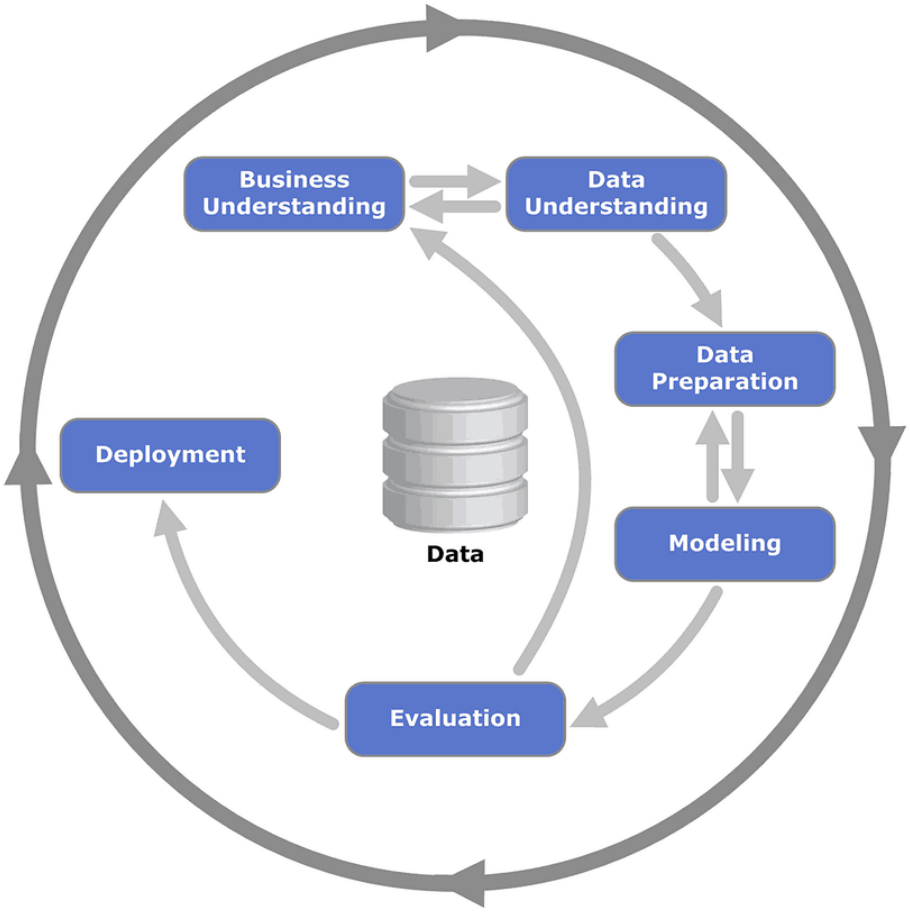
Massadataa voidaan kuvailla viidellä asialla, jotka ovat volyymi, nopeus, tyyppi, arvo ja vaihtelevuus. Nykyaikana datan määrä ei ole kooltaan ongelma, mutta suurin haaste on identifioida oleellisimmat asiat valtavan kokoisista datajoukoista ja muodostaa niistä jotain hyödyllistä. Dataa myös generoidaan todella nopeaa vauhtia ja datatieteilijöiden haaste on löytää tapa kerätä, prosessoida ja hyödyntää suuria määriä dataa, kun sitä tuodaan tietokantoihin. Dataa voidaan kerätä kolmessa eri muodossa: jäsenneltynä, epäjäsenneltynä tai osittain jäsenneltynä. Jäsennelty data voidaan järjestää siististi sarakkeiden kanssa tietokannassa. Tämän tyyppinen data on helppo syöttää, varastoida ja analysoida. Epäjäsennelty data on paljon hankalampi järjestää ja määritellä, koska data voi olla esimerkiksi sähköpostien tai sosiaalisen media viestien muodossa. Datan täytyy olla myös arvokasta eli sen on oltava jotenkin hyödyllistä liiketoiminnan kannalta. Vaikka dataa kerättäisiin valtavia määriä, jos siitä ei saada mitään hyötyä irti niin se on arvotonta. Big dataan investoiminen voi oikein toteutettuna pidemmän ajan jälkeen tuottaa runsaita tuloksia. (BBVA 2017; University of Wisconsin n.d.)

Viimeisenä ja kaikista tärkeimpänä on datan vaihtelevuus, jolla tarkoitetaan datan laatua ja sen luotettavuutta eli kuinka tarkkaa data on. Jos lähdedata on epätarkkaa tai virheellistä, sen analysointi ei ole hyödyllistä. Poikkeamat, epävakaisuudet ja kaksoiskappaleet ovat esimerkiksi joitain asioita, jotka vaikuttavat datan tarkkuuteen. Tämän takia täytyy varmistaa, että itse data ja sen

prosessointitapa ovat kummatkin järkeviä, jotta sitä on helppo tulkita ja kerätä olennaista informaatiota. (GutCheck 2019.)

2.3 Datatiede

Datatiede on laaja aihealue, jossa käytetään algoritmeja ja koneoppimismalleja datan analysointiin ja prosessointiin. Datatieteeseen sisältyy myös datan integrointia, visualisointia ja liiketoimintapäätösten tekemistä sekä niiden käyttöönottoa. (Shankar Ramya 2019.) Käyttämällä suuria määriä dataa ja analysoimalla sitä, voidaan saada selville uusia näkökulmia ja syvempää ymmärrystä datan aihealueesta. Yritykset voivat tämän avulla antaa parempia suosituksia, ennustaa tulevaisuuden trendejä ja rakentaa uusia hyödyllisiä ominaisuuksia liiketoiminnan parantamiseksi. (Srinidhi Sunny 2019.)



KUVA 2. Datatiedeprojektin vaiheet (Silipo Rosaria 2018)

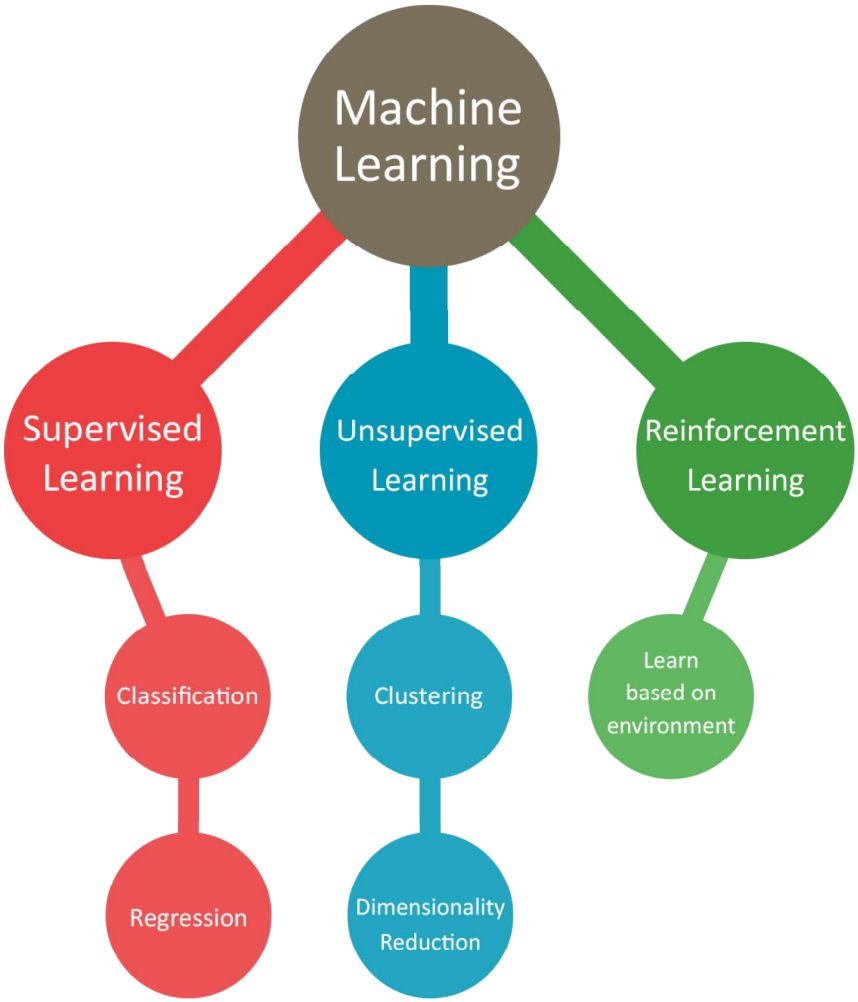
Datatiedeprojekteissa koko prosessi voidaan jakaa omiin lohkoihin tai vaiheisiinsa (kuva 2). Ennen datan keräämistä täytyy olla selvä tavoite ja ymmärrys siitä, mitä datalla halutaan tehdä liiketoiminnan parantamiseksi. Dataa voidaan tämän jälkeen kerätä monella eri tavalla, joista yleisimpiä ovat esimerkiksi tietokannat, asiakaskyselyt, tutkimukset ja internet-sivujen evästeet. Keräyksen jälkeen data puhdistetaan ja prosessoidaan, jotta sitä voidaan tutkia ja analysoida. Tämä vaihe vie usein eniten tai jopa valtaosan ajasta, koska raaka kerätty data on aina jollain tavalla virheellistä tai hieman puutteellista. Ennen mallintamista dataa tutkitaan ja tehdään erilaisia visualisointeja, jotta voidaan löytää ja havaita hyödyllisiä trendejä, poikkeamia sekä kuvioita datassa. Mallintaminen on usein datatiedeprojektien tärkein ja myös kiinnostavin vaihe, koska sen avulla voidaan käyttää tutkittua dataa ja saada uutta ymmärrystä sekä ennusteita tulevaisuuden trendeistä. Ennen käyttöönottoa, luotu malli arvioidaan ja varmistetaan olevan ennusteissaan tarkka, toiminnallinen ja liiketoimintatavoitteen mukainen. Luotuja koneoppimismalleja voidaan käyttöönoton jälkeen vielä mahdollisesti parantaa uudella datalla ja testaamisella. (Smith Alivia 2019.)

2.4 Koneoppiminen

Koneoppiminen on aihealue, joka menee usein tekoälyn kanssa sekaisin. Molemmat ovat kuitenkin eri asioita, vaikka ne liittyvät toisiinsa. Lyhyesti sanottuna koneoppiminen tarkoittaa jotakin ohjelmaa tai sovellusta, joka pystyy oppimaan ja mukautumaan uuteen informaatioon ilman manuaalista kanssakäymistä. Koneoppiminen on tekoälyn osa-alue, jossa ohjelman algoritmit pysyvät ajan tasalla syöttämällä itseensä jatkuvasti uutta dataa. Yleisiä asioita, missä koneoppimista voidaan soveltaa ovat esimerkiksi kasvojen tunnistus älypuhelimissa ja tableteissa sekä pankkien koneoppimisalgoritmit rahansiirtopetoksien estämiseen reaaliajassa. (Srinidhi Sunny 2019.)

Koneoppiminen on ollut olemassa jo vuosikymmeniä, mutta vasta viime vuosina sitä on voitu hyödyntää tehokkaasti. Teknologia on edistynyt huomattavaa vauhtia ja internet on yleistynyt joka puolella maailmaa. Dataa siis kertyy paljon enemmän kuin ennen, sen kerääminen on helpompaa ja datan varastoiminen on

huomattavasti halvempaa. Tämä ei ollut ennen mahdollista, vaikka koneoppimistekniikoita ja algoritmeja oli jo olemassa. Koneoppimisen aihealueeseen kuuluu pääasiassa kolme eri luokkaa, jotka ovat ohjattu oppiminen, ohjaamaton oppiminen ja vahvistusoppiminen (kuva 3). Nämä ovat kaikki erilaisia tapoja soveltaa koneoppimista samankaltaisiin, mutta silti erilaisiin käyttötarkoituksiin. (Srinidhi Sunny 2019.)

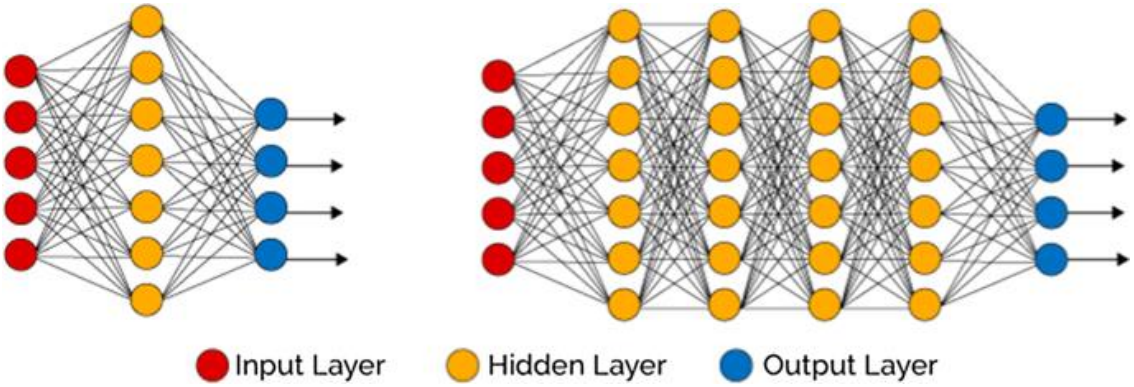


KUVA 3. Koneoppimisen luokat ja niihin perustuvat algoritmit

2.4.1 Neuroverkot ja syväoppiminen

Ihmisaivot ovat rakenteeltaan yhdistettyjä verkkoja hermoista, jotka prosessoivat informaatiota eri tavoilla. Neuroverkko on konsepti, joka koittaa simuloida näitä verkkoja ja saada tietokoneet käyttäytymään niin kuin yhdistetyt aivosolut, jotta ne pystyvät oppimaan ja tekemään päätöksiä samalla tavalla kuin ihmiset.

Yksinkertaisimmassa muodossaan neuroverkko sisältää vain kolme hermokerrosta (kuva 4). Nämä ovat syöttökerros (data syötetään järjestelmään), piilokerros (data prosessoidaan) ja lähtökerros (järjestelmä päättää, mitä datan perusteella tehdään).



KUVA 4. Neuroverkko ja syväoppimisjärjestelmä (S. Emmanuel 2019)

Hermokerroksia voi olla kuitenkin enemmän kuin kolme ja tässä vaiheessa voidaan ottaa käyttöön termi syväoppiminen. Syväoppimisessa neuroverkko on monimutkaisempi ja data kulkee useamman hermokerroksen läpi. Neuroverkkojen ja syväoppimisjärjestelmien oppimisalgoritmi voi olla joko ohjattu tai ohjaamaton, joista kerrotaan tarkemmin seuraavissa alaluvuissa. (Marr Bernard n.d.)

2.4.2 Ohjattu oppiminen

Ohjattu oppiminen on koneoppimisluokista kaikista yleisin ja yksinkertaisin. Se voidaan jakaa kahteen ryhmään, luokittelu ja regressio. Luokittelussa yhdistetään sopivat syöttölähtö-parit keskenään ja regressiossa syöttöpisteet osoitetaan jatkuvaan lähtöön (kuva 5). Luokittelussa ja regressiossa kummassakin koitetaan saada selville suhteita ja rakenteita syöttödatasta ja tästä tuottaa oikeanmukaista lähtödataa. Täytyy ottaa kuitenkin huomioon, että oikeanmukaisuus riippuu täysin harjoitusdatasta ja vaikka tämä antaa usein melko tarkan kuvan todellisuudesta, niin mallin ennustama data ei välttämättä ole täysin virheetöntä oikean elämän tilanteissa. (BrainStation 2017; Soni Devin 2018.)

KUVA 5. Luokittelun ja regression toimintaperiaate (Seebo n.d, muokattu)

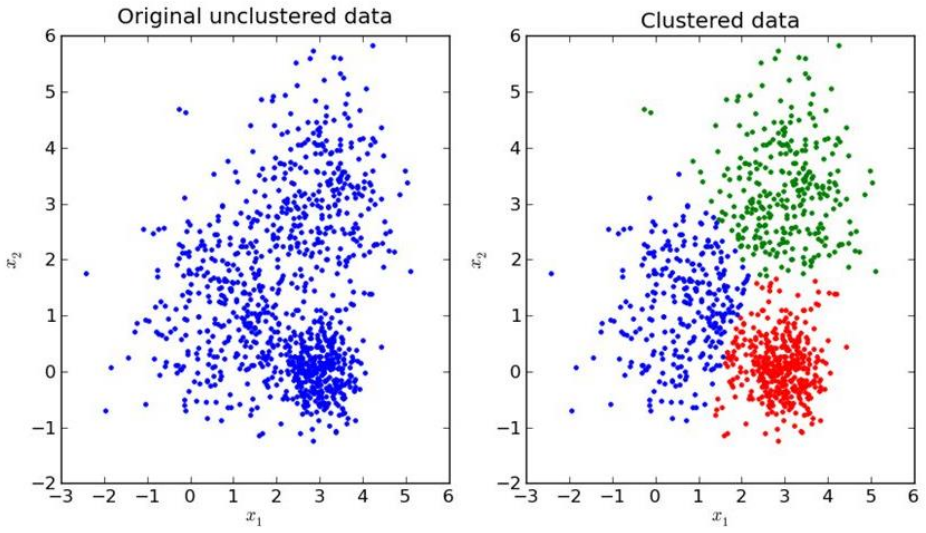
Tärkeimmät asiat, mitä ohjatussa oppimisessa täytyy ottaa huomioon, on mallin monimutkaisuus ja bias-variance tradeoff. Kummatkin liittyvät toisiinsa, joista mallin monimutkaisuus tarkoittaa kuinka monimutkaista tehtävää halutaan oppia, ja tämä riippuu täysin syöttödatasta. Liian pieni datan määrä monimutkaisessa mallissa voi ylisovittua ja samoin liian suuri data yksinkertaisessa mallissa voi alisovittua. Tämän takia mallille täytyy valita juuri sopivat asetukset, ettei malli ole liian monimutkainen tai yksinkertainen.

Bias-variance tradeoff tarkoittaa taas minkälainen poikkeama ja varianssi mallissa on. Suurella poikkeamalla ja matalalla varianssilla malli voisi olla esimerkiksi 15% väärässä suurimman osan ajasta, kun taas pienellä poikkeamalla ja suurella varianssilla mallin ennustus voi olla väärässä 2 – 30 % ajasta. Mallia tehtäessä täytyy siis päättää datan perusteella, kuinka poikkeavan ja vaihtelevan mallista haluaa tehdä. (Soni Devin 2018.)

2.4.3 Ohjaamaton oppiminen

Yleisimmät tehtävät ohjaamattomassa oppimisessa ovat klusterointi, piirreoppiminen ja tiheyden ennakointi. Kaikissa näissä tehtävissä tarkoitus on

saada selville datan luontainen rakenne käyttämättä mitään tarkkoja nimikkeitä (kuva 6). Koska ohjaamaton oppiminen ei käytä datassa nimikkeitä, ei ole usein mitään mahdollisuutta verrata mallin toimintaa alkuperäisen datan kanssa. (Soni Devin 2018.)



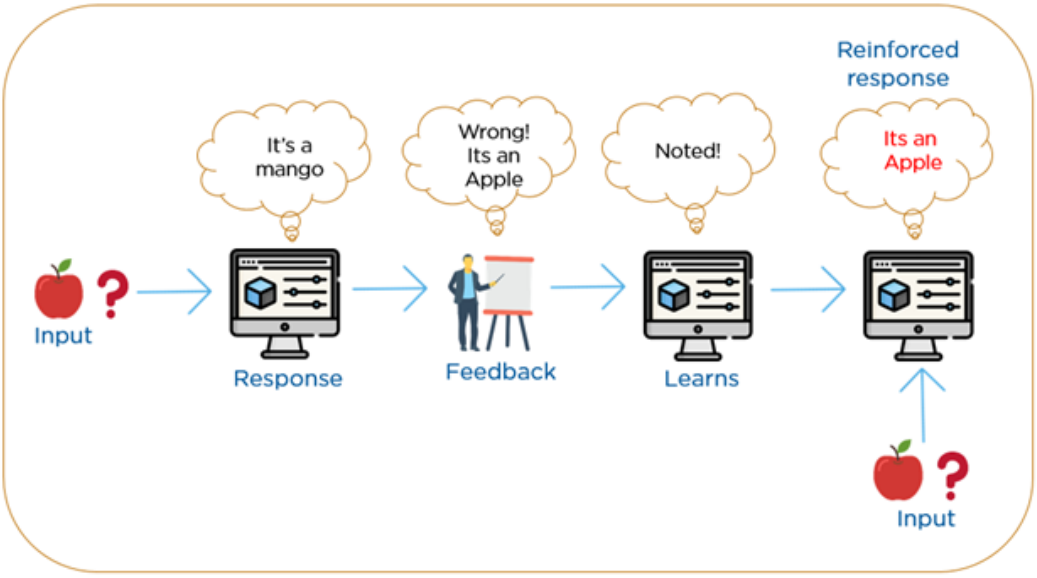
KUVA 6. Ohjaamattoman oppimisen toimintaperiaate (Abdullah Hamza 2018)

Käyttökohteina yleisimpinä ovat tutkiva analysointi ja ulottuvuuden pienentäminen. Ohjaamaton oppiminen on hyvin hyödyllistä tutkivassa analysoinnissa, koska sillä voidaan automaattisesti havaita datan rakennetta. Ulottuvuuden pienentämisellä tarkoitetaan tapoja esittää dataa käyttämällä vähemmän sarakkeita tai ominaisuuksia. Tarkoitus on esittää dataa käyttäen piileviä piirteitä datassa, jotka vastaavat datan alkuperäisiä piirteitä. Tämä piilevä rakenne esitetään usein paljon vähemmällä määrällä ominaisuuksia kuin alkuperäinen data, joten datan prosessointi on paljon kevyempää ja toistuvat ominaisuudet ovat poistettu. (Soni Devin 2018.)

2.4.4 Vahvistusoppiminen

Vahvistusoppiminen on melko uusi oppimistapa, joka on yleistynyt lähiaikoina. Tässä oppimistavassa mallille ei anneta valmiiksi oikeita syöttölähtö-pareja, mutta sille annetaan niin sanottu palkintotoiminto. Tällä tavalla malli päätyy eri lopputuloksiin, jotka eivät välttämättä ole aina oikein. Malli kuitenkin oppii

virheistään, kunnes se lopulta päätyy oikeaan vastaukseen (kuva 7). Se siis imitoi tapaa, miten ihmiset ja eläimet oppivat, testaamalla eri asioita ja saamalla jonkinlaisen palkinnon, kun vastaus on oikein. (BrainStation 2017.)



KUVA 7. Vahvistusoppimisen toimintaperiaate (Perera Shanika 2019)

Vahvistusoppimisen tarkoituksena on rakentaa matemaattinen framework ongelmien ratkaisuun. Oppimistapaa voidaan hyödyntää useissa eri aihealueissa esimerkiksi robotiikassa tai peleissä. Kummassakin aihealueessa kone opetetaan automatisoimaan jokin prosessi ja tekemään siitä täysin virheettömän. Kaikki asiat eivät ole kuitenkaan ratkaistavissa vahvistusoppimisen avulla. Oikean elämän tilanteissa voi olla useita komplekseja tekijöitä ja ympäristöllisiä muuttujia, jotka vaikuttavat lopputulokseen. (garychl 2018.)

3 BIGQUERY ANALYTIIKKADATAVARASTONA

3.1 BigQuery kokonaisuudessaan

BigQuery on Googlen ylläpitämä petatavumittakaavassa oleva matalakustanteinen pilvessä toimiva analytiikkadatavarasto. Nykypäivänä dataa on niin paljon, että se vaatii valtavan määrän laitteistoa, dataskaalautuvuuden ennakoimista ja järjestelmän arkkitehtuurin hallintaa. Tämä vaatii suuren määrän ihmisiä hoitamaan työtaakkaa, jotta kaikki saadaan toimimaan sujuvasti. BigQuery antaa mahdollisuuden keskittyä enemmän datan analysointiin käyttäen valmiiksi tuttua SQL-syntaksia, joka tekee työprosessista huomattavasti helpompaa. Alusta on tarkoitettu suurikokoisille datajoukoille ja se on NoOps eli hallittavaa infrastruktuuria ei ole ja tietokannan ylläpitoa ei tarvita. (Google Cloud n.d.)

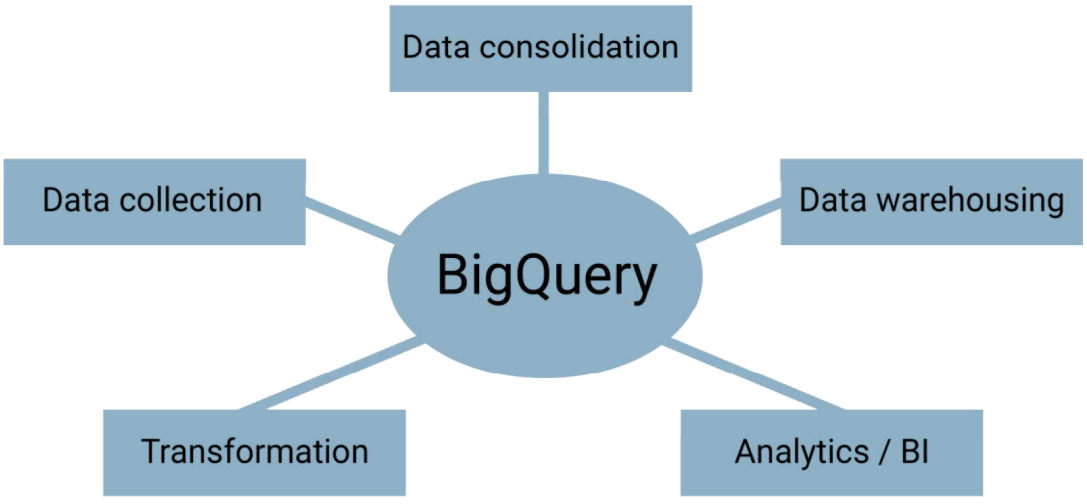
BigQuery on palveliton eli data voidaan varastoida alustalle matalalla hinnalla ja sitä voidaan skaalata nopeasti. Se toimii Googlen valmiiksi olemassa olevalla pilviarkkitehtuurilla sekä erilaisten dataprosessointimallien avulla. Algoritmit suorittavat alustalla tallennustilan optimointia tietokannassa oleviin datajoukkoihin ja muuttavat datarakenteita, jotta tulosten laatu voidaan maksimoida. Joustava dataskaalautuvuus takaa yrityksillä analytiikan ja tietokantahaun käyttämisen vuokraamalla palvelintilaa vain tarpeen mukaan. Reaaliaikainen sopeutuminen ja nopea kyselykapasiteetti tekevät BigQuerystä ideaalisen moneen eri käyttötarkoitukseen. (Blitz Shelby 2019; InfoTrust n.d.; Sisense n.d.)

Yhdistämällä BigQueryn pilvidatavaraston Google Cloud -yhteistyökumppanin järjestelmän/työkalun kanssa, voidaan saada suurin hyöty alustan datavarastoinnista. Tämä mahdollistaa alan johtavien dataintegraatio- ja BI-työkalujen käyttämisen datan purkamiseen, muokkaamiseen, lataamiseen ja visualisointiin. Sopiva integrointiratkaisu BigQueryn kanssa voi pienentää data-analyytikkojen ja datatieteilijöiden työtaakkaa ja tehdä työprosessista paljon sulavampaa työntekijöiden välillä. (Google Cloud n.d.; Blitz Shelby 2019.)

3.2 Pääominaisuudet ja toiminnot

Dataskaalautuvuuden ansiosta alustalla voidaan prosessoida ulkopuolisia datalähteitä pilvitallennustilan kautta, vaikka data ei olisi varastoitu BigQueryssä. BigQuery ei kuitenkaan takaa datan yhtenäisyyttä alkuperäisen lähteen kanssa, jos siihen tulee muutoksia kesken kyselyiden. Datan siirtäminen tehdään *Data Transfer Servicen* avulla, jolla voidaan ottaa dataa aikataulutetusti useasta lähteestä kerralla. Alustalla voidaan myös prosessoida tilitapahtuma -tietokantoja tai laskentataulukkoja Google Driven kautta. Tallennettu data voidaan jakaa järkevällä tavalla osiin ja useaan eri sijaintiin, joka voi helpottaa työprosessia työntekijöiden välillä. Data on alustalla automaattisesti varmuuskopioitu ja voidaan mahdollisesti palauttaa. (Google Cloud n.d.)

BigQueryn koneoppiminen antaa mahdollisuuden reaaliaikaiseen analytiikkaan. Reaaliaikaisen analytiikan avulla dataa voidaan syöttää ja analysoida välittömästi. Alustalla voidaan myös ohjata maantieteellistä dataa, jolla voidaan kokonaan ohittaa suurien dataryhmien hankala manipulointi. Datan analysointiin ja visualisointiin tarvitaan jokin visualisointityökalu, jolla havaintoja ja ennustuksia voidaan näyttää yksinkertaisemmassa ja helpommin katseltavassa muodossa. BigQueryn taulukoihin voidaan helposti ottaa yhteys useasta eri visualisointityökalusta. Suosituimpia tuettuja työkaluja ovat esimerkiksi Tableau, Google Data Studio, Looker ja Power BI. (Google Cloud n.d.)



KUVA 8. BigQueryn pääominaisuudet ja -toiminnot

3.3 BigQuery ML

BigQuery ML tai vain lyhyesti BQML on koneoppimistyökalu, jolla voidaan luoda ja ajaa koneoppimismalleja käyttämällä vain standardisoitua SQL-syntaksia. Laajoihin datajoukkoihin liittyvä koneoppiminen vaatii tekijältä laajaa ohjelmointitaitoa ja ymmärrystä koneoppimisesta. Tämä rajoittaa usein työn vain pienelle määrälle henkilöitä yrityksessä. BQML:n tarkoitus on antaa data analyytikoille tai muille SQL-ammatinharjoittajille mahdollisuus luoda koneoppimismalleja valmiiksi osatulla tietotaidolla ilman syvempää ymmärrystä. BQML eliminoi myös tarpeen siirtää dataa toiseen kohteeseen, jos tietokanta on valmiiksi jo integroitu alustalle. (Google Cloud n.d.)

BQML tukee tällä hetkellä seuraavia koneoppimismalleja (Google Cloud n.d.):

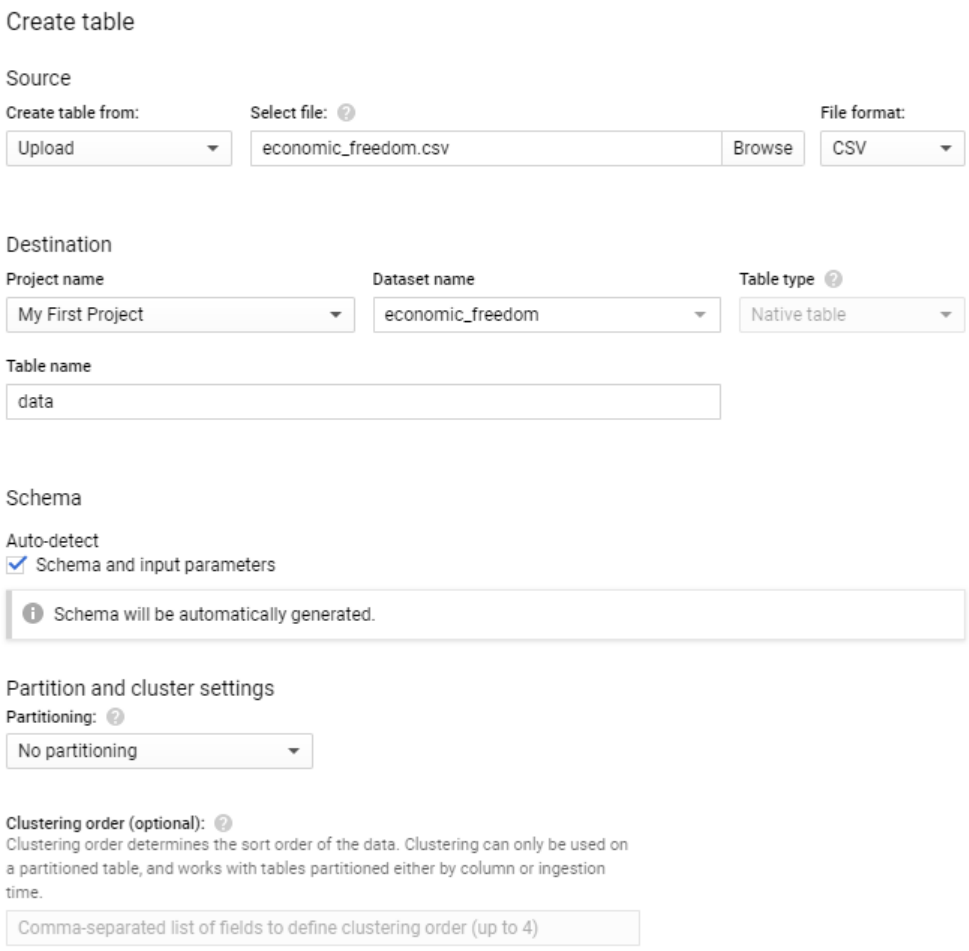
- Lineaarinen regressio (ennakointi)
- Binäärinen logistinen regressio (luokittelu)
- Moniluokkainen logistinen regressio (luokittelu)
- K-means klusterointi (segmentaatio)
- TensorFlow -mallin tuonti (mallin muuttaminen SQL-syntaksiseen muotoon)

BQML:n ansiosta koneoppimisen käyttö voidaan valtuuttaa yrityksessä suuremmalle määrälle henkilöitä. Pelkän SQL:n osaamistarve koneoppimiseen poistaa tarpeen muiden ohjelmointikielien ja työkalujen käyttöön. Mallinnus ja datan testaus on myös helpompaa ja nopeampaa, koska dataa ei tarvitse tuoda ulkopuolisesta tietokannasta. (Google Cloud n.d.)

3.4 BigQueryn käyttöönotto

Kolme yleisintä käyttötapaa alustalla on datan syöttäminen ja vienti, datan kysely ja katselu sekä datan hallinta. Nämä kaikki onnistuvat käyttämällä joko alustan web-käyttöliittymää, komentoriviä, REST API:a tai klienttikirjastoja. BigQueryn web-käyttöliittymää voidaan käyttää siirtymällä selaimella Google Cloud Platform -alustan konsoliin ja kirjoittamalla hakupalkkiin BigQuery. Ennen käyttöä, täytyy

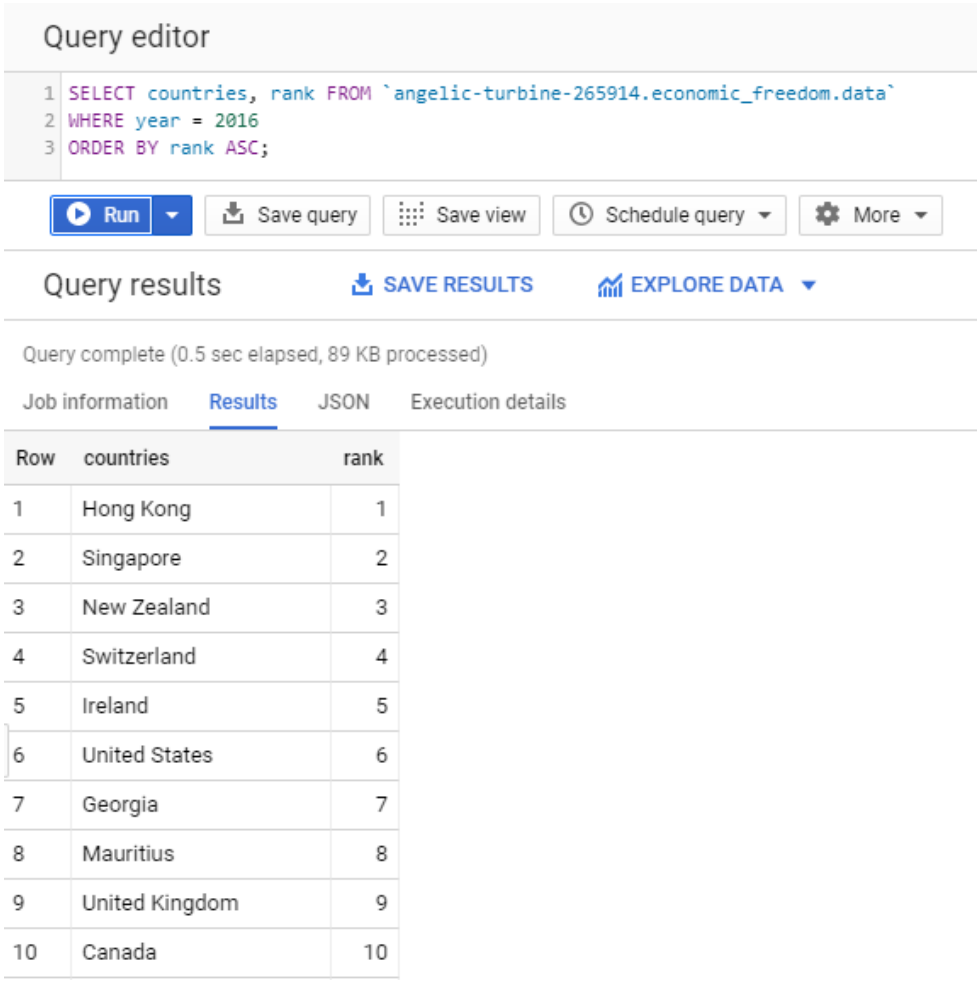
asettaa BigQueryn API päälle, jotta alustan ominaisuuksia voidaan käyttää. Kun alusta on saatu auki, syötetään ensin jotain dataa, jota voidaan tarkastella. Tämä onnistuu valitsemalla projekti, joka on automaattisesti generoitu ensimmäisellä käyttökerralla. Valitaan *Create dataset* ja annetaan sille nimeksi *economic_freedom*. Esimerkkidatana käytetään Kagglesta ladattua *Economic Freedom of the World* -datajoukkoa (Schneider Guillermina 2018). Datajoukossa on kaikkien tutkimuksessa olleiden maiden taloudelliseen vapauteen viittaavaa tietoa. Kun datajoukko on luotu, ladataan siihen taulukko *Create table*-painikkeesta. Käytetään taulukon tekemiseen kuvan 9 mukaisia asetuksia.



KUVA 9. Taulukon luominen BigQueryllä

Datajoukko ja siihen tehty taulukko on nyt luotu. Ladatusta datasta voidaan nähdä *Details*-ikkunassa dataan sisältyvät sarakkeet ja mitä tyyppiä ne ovat. Itse dataa voidaan tarkastella *Preview*-ikkunasta, joka näyttää kaikki 3726 datajoukossa olevaa riviä. Yleensä halutaan kuitenkin valita jokin tietty osa ja määrä datasta, jotta sitä on helpompi tarkastella. Tämä onnistuu käyttämällä kyselyeditoria ja

asettamalla halutut kyselykriteerit. Kyselyeditorin saa avattua painamalla *Compose new query*. Kyselyeditori on jo auki, jos teksti näkyy harmaana. Jos haluttaisiin tietää esimerkiksi kaikkien maiden taloudellisen vapauden sijoitukset vuonna 2016 järjestettynä sijoituksen mukaan ensimmäisestä viimeiseen, asetettaisiin kyselykriteerit kuvan 10 mukaisesti. Kyselyeditorin alla nähdään, miltä ajettu kysely näyttää. Kyselyn prosessointiin käytetty datan määrä on myös näkyvissä, joka on tärkeää tietää, koska yleisesti prosessoidaan paljon suurempia datamääriä ja kyselyiden prosessoinnista kertyy kuluja. Käyttöliittymä on muuten samankaltainen ja toimii kuin muutkin tietokannan hallintaohjelmat.



KUVA 10. Hakueditorin käyttö BigQueryllä

4 DATAN ANALYSOINTI JA TUTKIMINEN

4.1 Yleistä

Tutkivalla datan analysoinnilla tarkoitetaan prosessia, jossa datasta koitetaan etsiä samankaltaisia kuvioita, poikkeamia ja testata hypoteesejä graafisten visualisointien avulla. Tärkeintä on ymmärtää dataa ja kerätä siitä mahdollisimman useaa näkökulmaa ennen sen käyttämistä. Tutkivan datan analysoinnin kolme tärkeintä asiaa on ymmärtää kaikki datassa olevat muuttujat, puhdistaa data ja analysoida muuttujien suhteita toisiinsa. Kaikki nämä kolme asiaa menevät usein käsi kädessä. Muuttujien lukeminen ja ymmärtäminen tekevät datan puhdistamisesta helpompaa. Datan puhdistaminen taas tekevät muuttujien suhteiden analysoinnista ja visualisoinnista paljon selkeämpää ja helpommin ymmärrettävää. (Shin Terence 2019.)

4.2 Datan tarkastelu ja puhdistaminen

Käytetään Kagglesta ladattua *World Happiness Report 2019* -datajoukkoa (PromptCloud 2019). Dataa olisi tarkoitus käyttää ennustamaan terveellistä elämänajanodotetta datajoukon muiden muuttujien avulla. Käytetään datan analysointiin ja sen tutkimiseen Jupyter Notebook nimistä alustaa. Jupyter Notebook on pilvessä toimiva open source web-sovellus, jolla voidaan luoda koodia, visualisointeja ja laskutoimituksia sisältäviä dokumentteja. Avataan Jupyterissa datajoukko ja tarkastellaan sen sisältöä kuvasta 11.

	Country (region)	Ladder	SD of Ladder	Positive affect	Negative affect	Social support	Freedom	Corruption	Generosity	Log of GDP\nper capita	Healthy life\nexpectancy
0	Finland	1	4	41.0	10.0	2.0	5.0	4.0	47.0	22.0	27.0
1	Denmark	2	13	24.0	26.0	4.0	6.0	3.0	22.0	14.0	23.0
2	Norway	3	8	16.0	29.0	3.0	3.0	8.0	11.0	7.0	12.0
3	Iceland	4	9	3.0	3.0	1.0	7.0	45.0	3.0	15.0	13.0
4	Netherlands	5	1	12.0	25.0	15.0	19.0	12.0	7.0	12.0	18.0
5	Switzerland	6	11	44.0	21.0	13.0	11.0	7.0	16.0	8.0	4.0
6	Sweden	7	18	34.0	8.0	25.0	10.0	6.0	17.0	13.0	17.0
7	New Zealand	8	15	22.0	12.0	5.0	8.0	5.0	8.0	26.0	14.0
8	Canada	9	23	18.0	49.0	20.0	9.0	11.0	14.0	19.0	8.0
9	Austria	10	10	64.0	24.0	31.0	26.0	19.0	25.0	16.0	15.0

KUVA 11. Maailman onnellisuus 2019 -datajoukko

Datajoukon sarakkeiden määritelmät:

- Country (region): Maan nimi
- Ladder: Elämän onnellisuuden sijoitus
- SD of Ladder: Standardi deviaatio elämän onnellisuuden sijoituksesta
- Positive affect: Positiivisten tunteiden vaikutus onnellisuuteen
- Negative affect: Negatiivisten tunteiden vaikutus onnellisuuteen
- Social support: Yhteiskunnallisen tuen vaikutus onnellisuuteen
- Freedom: Vapauden vaikutus onnellisuuteen
- Corruption: Korruption vaikutus onnellisuuteen
- Generosity: Anteliaisuuden vaikutus onnellisuuteen
- Log of GDP per capita: Bruttokansantuotteen vaikutus onnellisuuteen
- Healthy life expectancy: Terveellisen elämänajanodotteen vaikutus onnellisuuteen

Kaikki datajoukon sarakkeet paitsi maa ja elämän onnellisuuden sijoitus vaikuttavat jotenkin elämän onnellisuuteen. Sarakkeiden arvot ovat kaikki maiden sijoituksia muuttujista eikä oikeita arvoja. Esimerkiksi *Corruption*-sarake ei ole mitta korruptiosta vaan kyseisen maan sijoitus korruption määrästä. Katsotaan seuraavaksi tarkemmin sarakkeiden tietoja ja tarvittaessa puhdistetaan data (kuva 12).

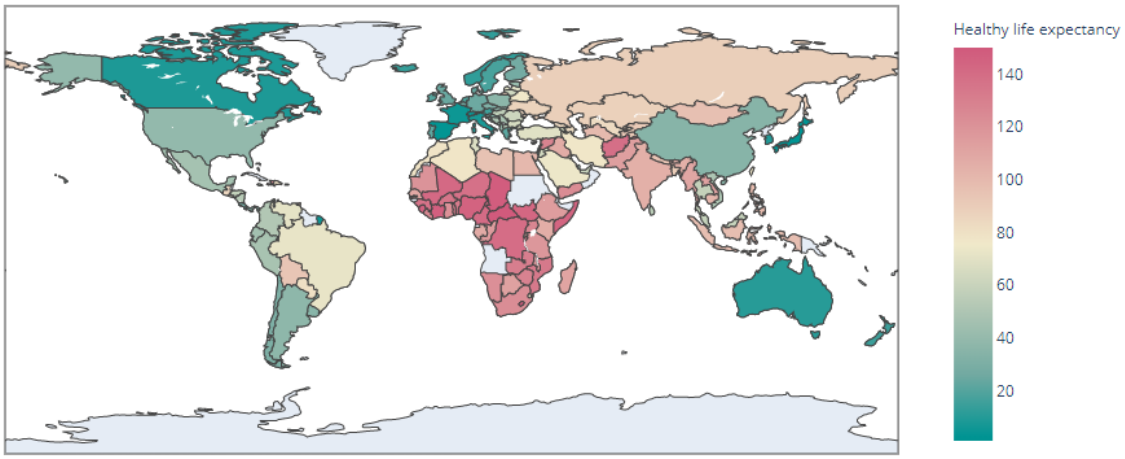
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
Country (region)          156 non-null object
Ladder                    156 non-null int64
SD of Ladder              156 non-null int64
Positive affect           155 non-null float64
Negative affect           155 non-null float64
Social support            155 non-null float64
Freedom                   155 non-null float64
Corruption                148 non-null float64
Generosity                155 non-null float64
Log of GDP
per capita      152 non-null float64
Healthy life
expectancy    150 non-null float64
dtypes: float64(8), int64(2), object(1)
memory usage: 13.5+ KB
```

KUVA 12. Maailman onnellisuus 2019 -datajoukon tiedot

Datassa on rivejä yhteensä 156, jotka ovat tutkimuksessa osallistujana olleet maat. Kaikki ennustuksiin tarvittavat sarakkeet ovat jo numeerisessa muodossa eli datan formaatteja ei tarvitse muokata. Datajoukko näyttäisi kuitenkin sisältävän hieman puuttuvia arvoja. Jotta datajoukkoa voidaan käyttää mallintamisessa, täytyy puuttuvat arvot täyttää tai poistaa. Tässä tapauksessa otetaan jokaisen sarakkeen arvoista keskiarvot ja asetetaan ne puuttuvien arvojen tilalle. Tämä ei ole optimaalisin tapa ratkaista ongelmaa, mutta se toimii riittävän hyvin, ettei se tule vaikuttamaan kovin paljon mallin tarkkuuteen.

4.3 Datan analysointi

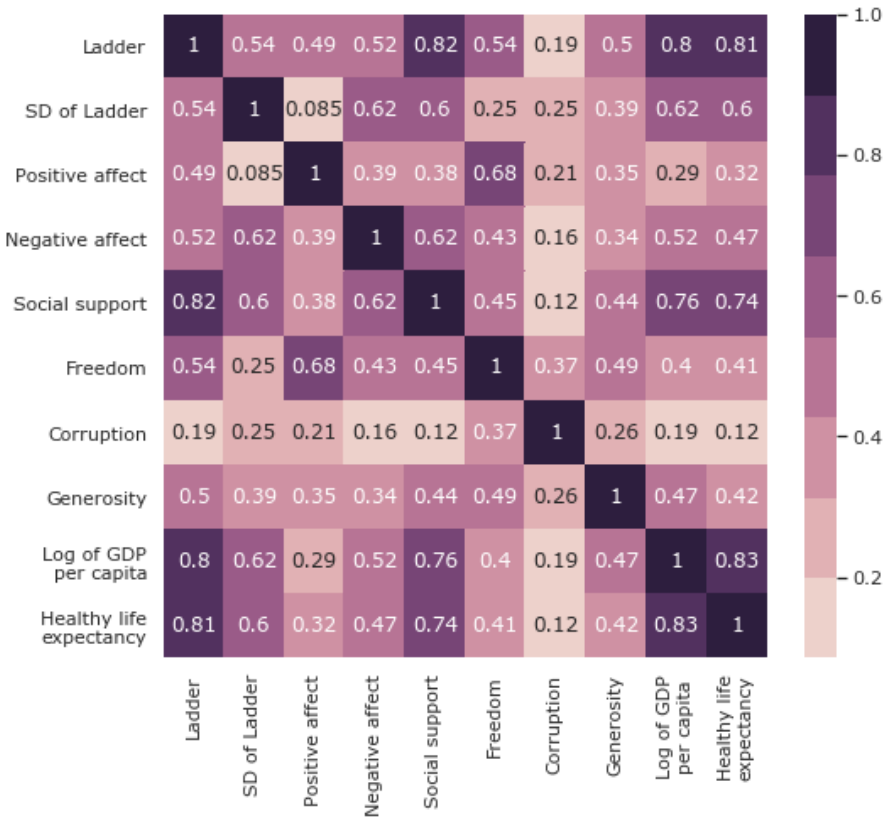
Nyt kun ymmärretään datajoukossa olevat muuttujat paremmin, voidaan tehdä hieman visualisointeja ja tutkia muuttujia tarkemmin. Koska tarkoitus olisi myöhemmin ennustaa terveellistä elämänajanodotetta, sijoitetaan ensin muuttujan arvot maailmankartalle ja katsotaan, miten ne sijoittuvat (kuva 13).



KUVA 13. Terveellinen elämänajanodote maailmankartalla

Maailmankartta on värikoodattu sijoituksen pudotessa vihreästä punaiseksi. Kartasta nähdään, että maan kehittyneisyys vaikuttaisi olevan suurin tekijä terveellisen elämänajanodotuksen sijoituksessa. Kehittyneimmillä mailla on pääosin hyvä sijoitus, kun taas esimerkiksi Afrikan ja Lähi-Idän mailla sijoitus on hyvin huono.

Katsotaan seuraavaksi muuttujien yhtäläisyyksiä tekemällä niistä korrelaatiomatriisi. Korrelaatiomatriisi on taulukko, joka vertaa muuttujia keskenään ja laskee sen perusteella suhdekertoimet. Suhdekertoimet muodostavat korrelaatiomatriisiin värikartan, jonka avulla voidaan helposti nähdä muuttujien tärkeys ja yhtäläisyys. Katsotaan datajoukosta tehtyä korrelaatiomatriisia kuvasta 14.



KUVA 14. Korrelaatiomatriisi Maailman onnellisuus 2019 -datajoukosta

Korrelaatiomatriisista nähdään, että jokainen muuttuja korreloi positiivisesti toisiinsa. Tämä on loogista, koska maailman maiden onnellisuuden laskemiseen on todennäköisesti käytetty onnellisuuteen positiivisesti vaikuttavia tekijöitä. Tärkeintä on kuitenkin katsoa Healthy life expectancy -muuttujan yhtäläisyyttä muihin tekijöihin. Huomataan, että bruttokansantuote, yhteiskunnallinen tuki, elämän onnellisuuden sijoitus ja sen deviaatio vaikuttavat tähän eniten. Sijoitus ja sen deviaatio ovat kuitenkin datajoukon muista muuttujista laskettuja arvoja eli näitä muuttujia ei tarvitse huomioida. Tehdään bruttokansantuotteesta ja yhteiskunnallisesta tuesta hajontakaaviot ja katsotaan, miten pisteet sijoittuvat (kuva 15).

KUVA 15. Bruttokansantuotteen ja yhteiskunnallisen tuen vaikutus terveelliseen elämänajanodotteeseen.

Hajontakaavio on koordinaatisto, joka sisältää datajoukon kaikki datapisteet. Pisteet sijoittuvat akseleilla muuttujien arvojen paikoille ja tämän avulla voidaan tarkemmin tarkastella niiden suhdetta. Bruttokansantuotteella ja yhteiskunnallisella tuella näyttää olevan kummallakin positiivinen lineaarinen suhde terveelliseen elämänajanodotteeseen. Muuttujien suhteet ovat kohtuullisen kokoiset, mutta ei kuitenkaan merkittävän suuret.

5 KONEOPPIMISMALLIEN LUOMINEN BQML:SSÄ

5.1 Koneoppimismallin luominen

Ennen mallintamista, tallennetaan ensin datajoukko Jupyterista ja siirretään se BigQueryyn. Tehdään BigQueryssä uusi datajoukko nimeltä *world_happiness*. Lisätään siihen vielä taulukko ja muokattu datajoukko samoilla asetuksilla kuin aikaisemmin. BigQueryssä koneoppimismallin luominen onnistuu samalla kyselyeditorilla, jota aikaisemmin käytettiin datajoukon tutkimiseen. Tarkastellaan kuvassa 16 olevaa kyselyä. Malli saadaan luotua käyttämällä komentoa *CREATE MODEL* ja sen perään annetaan mallille nimi sekä viitataan samalla käytettävään datajoukkoon. Annetaan mallille nimeksi vaikka *life_expectancy_model*. Asetuksiin määritellään käytettävä algoritmi ja ennustettava tekijä datajoukossa. Käytetään yksinkertaista lineaarista regressiota ja asetetaan ennustavaksi tekijäksi *Healthy_life_expectancy*. Mallin käyttämät sarakkeet ennustamista varten voidaan valita *SELECT*-lausekkeella. Jätetään pois maa-, sijoitus- ja sijoituksen deviaatio -sarakkeet, koska niistä ei ole mitään hyötyä ennustuksen kannalta. Lopuksi valitaan datajoukko, josta data tuodaan mallille. Riippuen datajoukon koosta, mallin luomisprosessissa voi kestää useita minuutteja, mutta käytetty datajoukko on melko pieni niin aikaa ei mene kovin kauan.

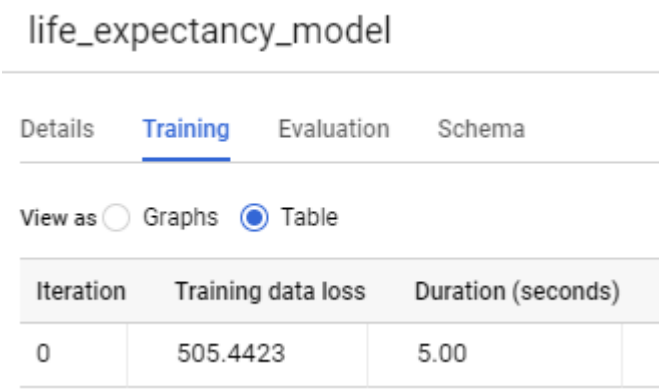
Query editor

```
1  CREATE OR REPLACE MODEL world_happiness.life_expectancy_model
2  OPTIONS
3    (model_type='linear_reg', input_label_cols=['Healthy_life_expectancy']) AS
4  SELECT
5    Positive_affect,
6    Negative_affect,
7    Social_support,
8    Freedom,
9    Corruption,
10   Generosity,
11   Log_of_GDP_per_capita,
12   Healthy_life_expectancy,
13 FROM `angelic-turbine-265914.world_happiness.data`
```

KUVA 16. Koneoppimismallin luominen kyselyeditorilla maailman onnellisuusdatasta.

5.2 Harjoittaminen

Mallin harjoittaminen, validointi ja arviointi tapahtuvat BigQueryssä mallin luomisprosessissa. Prosessissa BQML määrittelee automaattisesti harjoittamis-, validointi- ja arviointidatan määrän, jotta malli ei yli- tai alisovitu. Normaalissa tapauksessa BQML:n koneoppimisalgoritmit toimivat ottamalla useita malleja, jossa ennuste jo tiedetään ja muuttamalla eri tekijöiden painotuksia mallissa. Algoritmit tekevät siis iterointeja, jotta mallin ennusteet vastaavat mahdollisimman hyvin oikeanmukaisia arvoja. Tarkastellaan seuraavaksi harjoituksesta saatuja tuloksia kuvasta 17.



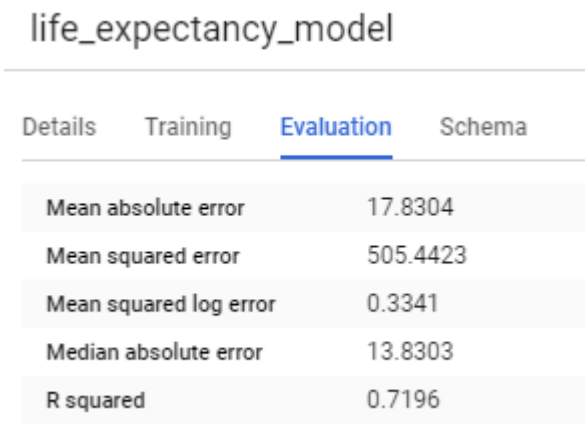
KUVA 17. Koneoppimismallin harjoitustulokset.

Kun datajoukko sisältää vain pienen määrän dataa ja käytössä on lineaarinen regressio, BQML voi käyttää nopeampaa Pseudo-käänteistä algoritmia. Algoritmi tuottaa yhtä hyvät tulokset kuin tekemällä useita iteraatioita, mutta prosessi on nopeampi. (Lakshmanan Lak 2019.) Tämän takia harjoitustuloksissa näkyy vain yksi iteraatio. Tuloksissa harjoitusdatahäviö esittää käytetyssä mallissa keskimääräistä neliövirhettä, jolla voidaan määritellä mallin ennustettujen arvojen virhe. Arvosta voidaan päätellä, kuinka hyvin malli on sovitettu vertaamalla sitä validointidatavirheeseen.

5.3 Arviointi

Arviointi on olennainen osa koneoppimismallien kehitysprosessia, koska on tärkeää tietää toimiiko malli oikein ja onko se riittävän luotettava ennusteiden

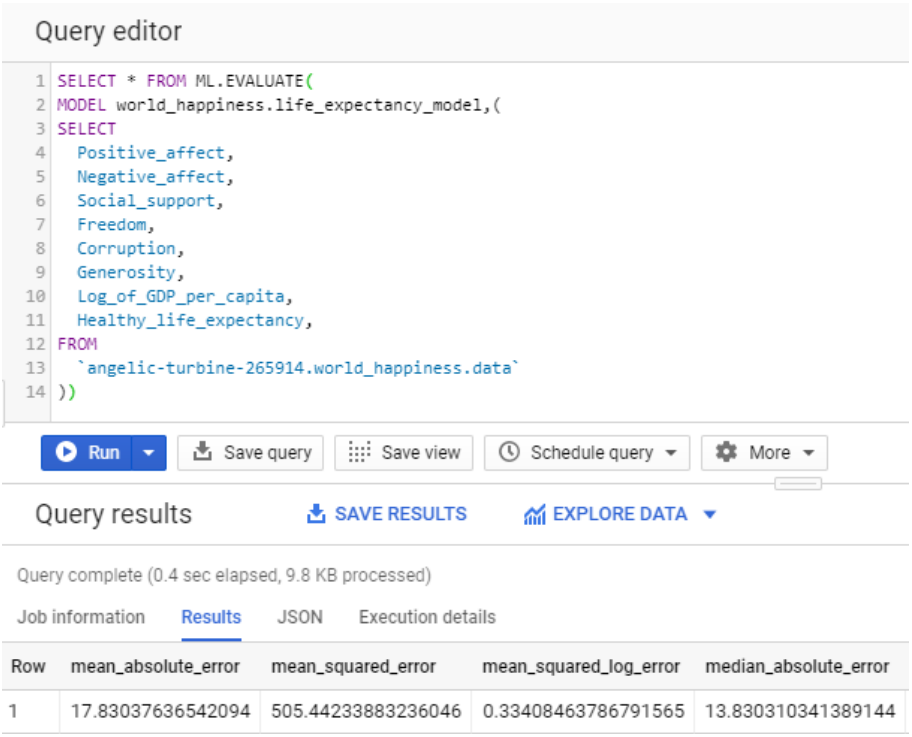
tekemisessä. BQML suorittaa arvioinnin samalla mallin luomisvaiheessa käyttämällä ns. pidättämistekniikkaa, jossa käytettävä data jaetaan osiin harjoitusta, validointia ja arviointia varten. Pidättämistekniikan tarkoitus on saada mahdollisimman tarkka arvio mallin toiminnallisuudesta. (Mutuvi Steve 2019.) Katsotaan seuraavaksi saatuja arviointituloksia kuvasta 18.



KUVA 18. Koneoppimismallin arviointitulokset.

Arviointituloksissa MSE eli *Mean squared error* esittää validointidatavirhettä. Virhe näyttäisi olevan sama kuin harjoitusdatahäviö (kuva 13) eli malli on siis hyvin sovittunut. Tuloksissa toinen olennainen arvo on MAE eli *mean absolute error*, joka mittaa kahden jatkuvan tekijän eroa. MAE on keskiarvo mitattujen tekijöiden absoluuttisista virheistä, jotka ovat siis ennuste ja alkuperäinen arvo. Arvioinnissa MAE-arvoksi saatiin noin 17,83 eli toisin sanoen malli voi olla väärässä ennustettaessa terveellistä elämänajanodotetta keskimääräisesti 17,83 sijoituksen verran.

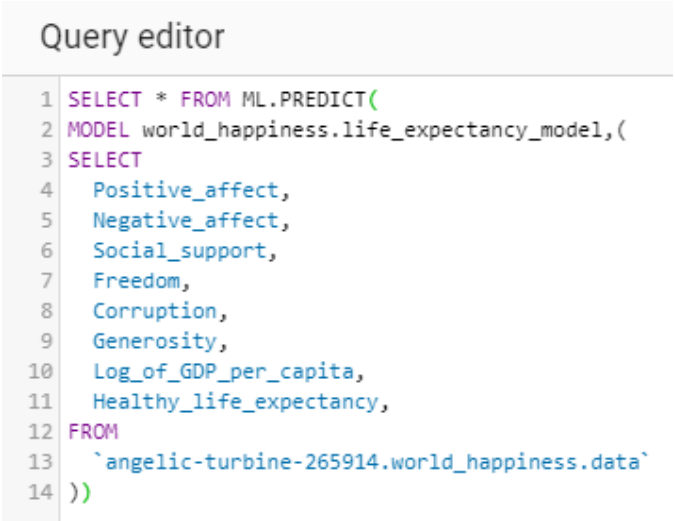
Mallin arvioiminen voidaan vaihtoehtoisesti suorittaa myös erillisellä datajoukolla (kuva 19). Tätä tekniikkaa kutsutaan nimellä ristivalidointi. Käyttämällä erikseen toista datajoukkoa mallin arvioimiseen, voidaan mahdollisesti saada paremmat tulokset mallin tarkkuudesta kuin käyttämällä pidätystekniikkaa. (Mutuvi Steve 2019.) Esimerkissä on käytetty samaa datajoukkoa mitä aikaisemmin, koska erillistä datajoukkoa ei ollut käytettävissä. Tulokset ovat sen takia samat kuin pidätystekniikalla.



KUVA 19. Ristivalidointitekniikka ja siitä saadut tulokset.

5.4 Ennusteiden tekeminen

Nyt kun malli on arvioitu ja ollaan varmistettu sen olevan riittävän tarkka, voidaan käyttää harjoitettua mallia ennusteiden tekemiseen. Ennustuksessa algoritmi koittaa määritellä mahdollisimman tarkan lopputuloksen ennustettavasta tekijästä. Algoritmi vertaa harjoitettua mallia ja siihen varattua testidataa keskenään, ja tekee sen perusteella oikeanmukaisen päätöksen. BQML:ssä ennusteet voidaan tehdä kuvan 20 mukaisella kyselyllä.



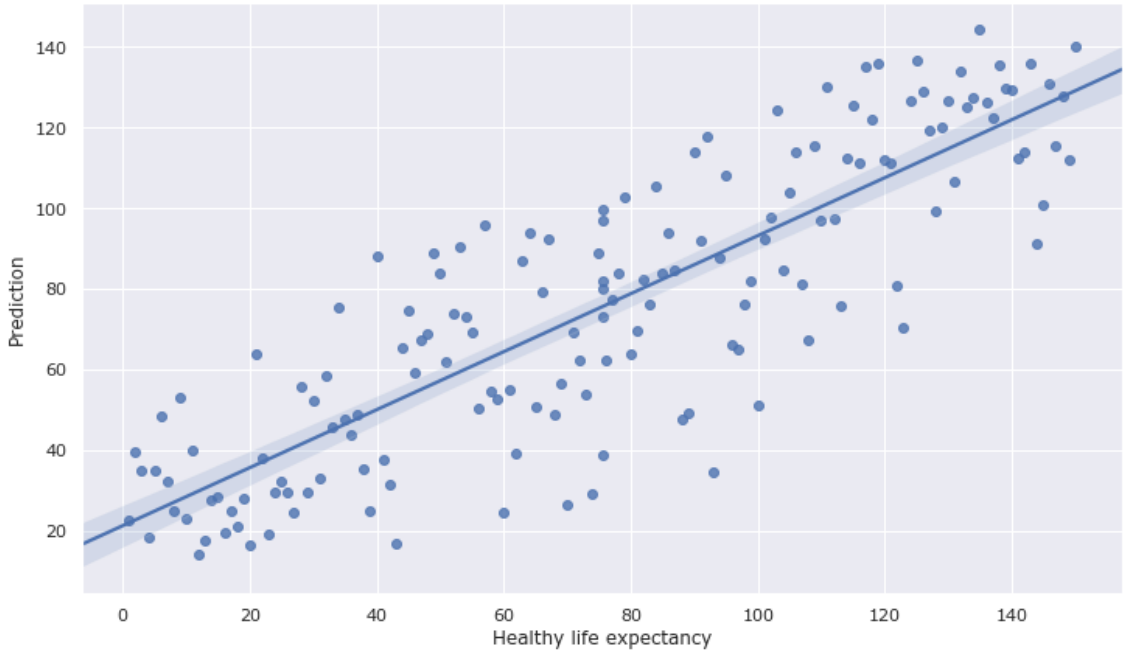
KUVA 20. Ennusteiden tekeminen BQML:llä.

Kysely on muuten samanlainen kuin aikaisemmin tehdyt kyselyt, mutta käytetään funktiota *ML.PREDICT*, joka osoittaa mallilla ennustamista. Funktio antaa lopputulokseksi kaikki syöttödatassa olleet sarakkeet ja mallin ennustettavan tekijän sarakkeen. Riippuen käytetystä koneoppimisalgoritmista ja asetuksista, malli voi myös ennusteissaan antaa enemmän kuin vain yhden sarakkeen ulostuloksi. Tarkastellaan seuraavaksi ennustettuja tuloksia ja alkuperäistä dataa kuvasta 19.

predicted_Healthy_life_expectancy	Positive_affect	Negative_affect	Social_support	Freedom	Corruption	Generosity	Log_of_GDP_per_capita	Healthy_life_expectancy
124.36920956842937	54.0	102.0	144.0	21.0	2.0	90.0	132.0	103.0
19.22636791307714	24.0	26.0	4.0	6.0	3.0	22.0	14.0	23.0
24.534172880386265	41.0	10.0	2.0	5.0	4.0	47.0	22.0	27.0
27.64733786853806	22.0	12.0	5.0	8.0	5.0	8.0	26.0	14.0
24.760405246627975	34.0	8.0	25.0	10.0	6.0	17.0	13.0	17.0
18.470712419379623	44.0	21.0	13.0	11.0	7.0	16.0	8.0	4.0
13.904025835705625	16.0	29.0	3.0	3.0	8.0	11.0	7.0	12.0
19.506334253127548	62.0	19.0	27.0	28.0	9.0	30.0	2.0	16.0
16.1966953620803	33.0	32.0	6.0	33.0	10.0	9.0	6.0	20.0
24.866724425786103	18.0	49.0	20.0	9.0	11.0	14.0	19.0	8.0
20.86264070064366	12.0	25.0	15.0	19.0	12.0	7.0	12.0	18.0
22.82684795823385	47.0	37.0	7.0	17.0	13.0	6.0	18.0	10.0
38.56759897693542	105.0	28.0	76.0	66.0	14.0	18.0	9.0	75.5
29.345171794509923	52.0	42.0	9.0	63.0	15.0	4.0	23.0	24.0
91.04434143523844	2.0	18.0	145.0	14.0	16.0	96.0	76.5	144.0

KUVA 21. Terveellisen elämänajanodotteen ennusteet ja syöttödata.

Verrataan taulukosta terveellisen elämänajanodotteen arvoja ennustettuihin arvoihin, jotka löytyvät ensimmäisestä *predicted_Health_life_expectancy*-sarakkeesta. Ennustetuissa arvoissa vaikuttaa olevan vain vähän heittoa alkuperäisiin arvoihin lukuun ottamatta muutamaa poikkeamaa. Arvoista voidaan kuitenkin päätellä, että aikaisemmin tehdyssä mallin arvioimisessa 17,83 sijoituksen ennustevirhe vaikuttaisi olevan melko tarkka. Tallennetaan seuraavaksi kyselyllä tehty taulukko, jotta voidaan visualisoida ennuste. Tämä onnistuu *Save results* -painikkeella. Taulukon voi tallentaa pilveen tai lokaalisti joko CSV- tai JSON-formaatissa. Avataan tallennettu taulukko Jupyterissa ja tehdään ennustetuista arvoista regressiokäyrä (kuva 22).



KUVA 22. Regressiokäyrä ennustetuista arvoista verrattuna alkuperäisiin arvoihin.

Regressiokäyrä toimii samalla tapaa kuin hajontakaavio, mutta datapisteiden lisäksi koordinaatistoon on piirretty käyrä, joka esittää vastemuuttujan riippuvuutta ennustemuuttujasta. Normaalisti regressiokäyrässä verrattaisiin ennustuksessa käytettyä testidataa ennustettuihin arvoihin, mutta valitettavasti BigQueryllä ei pysty tallentamaan testidataa erikseen. Tämän takia jouduttiin käyttämään koko datajoukon arvoja käyrän tekemiseen.

Regressiokäyrän pisteet ovat kohtuullisen hyvin sijoittuneet, käyrä näyttää selvästi olevan positiivinen ja suuria poikkeamia ei ole paljon. Malli voisi olla siis melko hyvä terveellisen elämänajanodotteen ennustamiseen, jos datajoukon arvot olisivat olleet eri muodossa. Malli antaa datajoukon muodon takia arvoksi sijoituksen eikä ikää, mikä olisi paljon ideaalisempi mitta. Ennustaminen on tietenkin mallilla mahdollista, mutta sijoitus ei ole mittana kovin hyödyllinen, jos mallia haluttaisiin käyttää työelämän tilanteissa.

6 POHDINTA

BigQuery ja sen koneoppimisympäristö osoittautuivat tuovan datatieteen ja koneoppimisen keskitettyyn ympäristöön, jotka tekevät ratkaisukehityksen työprosessista tehokkaampaa ja sulavampaa työntekijöiden välillä. Koska hallittavaa infrastruktuuria ei BigQueryssä ole ja alustalla on riittävä kapasiteetti skaalautua datan kasvaessa, se tekee siitä ideaalisen yrityksille, jotka keräävät dataa useasta eri tietokannasta tai jostain muusta lähteestä. Yritykset voivat käyttää BigQueryä keskitettynä datavarastona, jossa tätä dataa voidaan analysoida ja mallintaa tehokkaasti.

Koneoppiminen on integroitu alustalle, niin että tietokannan hallitseminen ja koneoppimisen kehitys tehdään samalla käyttöliittymällä. Mallien luominen tapahtuu yksinkertaisesti BigQueryn koneoppimifunktioiden ja SQL-syntaksin avulla. BigQuery ML tukee yleisimpiä koneoppimisalgoritmeja, jotka kattavat valtaosan koneoppimisen käytöstä maailmanlaajuisesti. Manuaalinen asetuksien säätäminen koneoppimismalleissa on alustalla rajattu melko pieneksi, joka tekee prosessista yksinkertaisempaa, mutta samalla tekijä ei saa paljon valtaa koneoppimismallia tehdessä. Prosessointiaika vaihtelee, mutta lähtökohtaisesti se on hyvin nopeaa suurempienkin datajoukkojen kanssa.

Koneoppimistyökaluna BigQuery ML sopii parhaiten nopealle kokeilulle ja testaamiselle, jotta voidaan nähdä, onko koneoppiminen mahdollista kyseisestä datasta. Alustan pilvilaskennan kustannukset ovat hyvin alhaiset, niin toistuva testaaminen ei tuota mallinnuksesta paljon kustannuksia. Mallien tarkkuus voi olla joskus riittävää ja syvempää tutkimista ei tarvitse tehdä, mutta jonkin ongelman löytyessä BigQuery ML ei anna mahdollisuutta parantaa mallia riittävästi. Muista koneoppimistyökaluista ei siis voida tai kannata kokonaan vielä luopua. Optimaalisin ratkaisu olisi käyttää siis lähtökohtaisesti BigQueryä, mutta soveltaa muita koneoppimistyökaluja tarpeen mukaan.

LÄHTEET

Abdullah Hamza. 07.07.2018. Machine learning: A strategy to learn and understand (Chapter 3) Part 3: Unsupervised Learning. Luettu 27.01.2020. https://medium.com/the-21st-century/machine-learning-a-strategy-to-learn-and-understand-chapter-3-9daaad4afc55

BBVA. 08.05.2017. The five V's of big data. Luettu 10.01.2020. https://www.bbva.com/en/five-vs-big-data/

Blitz Shelby. 14.01.2019. The Benefits of Combining Google BigQuery and BI. Luettu 04.02.2020. https://dzone.com/articles/the-benefits-of-combining-google-bigquery-and-bi

BrainStation. 29.08.2017. Machine Learning 101 | Supervised, Unsupervised, Reinforcement & Beyond. Luettu 23.01.2020. https://towardsdatascience.com/machine-learning-101-supervised-unsupervised-reinforcement-beyond-f18e722069bc

garychl. 02.08.2018. Applications of Reinforcement Learning in Real World. Luettu 31.01.2020. https://towardsdatascience.com/applications-of-reinforcement-learning-in-real-world-1a94955bcd12

GMG Group. N.d. How are AI, machine learning, big data, deep learning and data science interconnected?. Luettu 07.01.2020. https://gmggroup.org/how-are-ai-machine-learning-big-data-deep-learning-and-data-science-interconnected/

Google Cloud. N.d. BigQuery documentation. Luettu 02.02.2020. https://cloud.google.com/bigquery/docs

GutCheck. 29.08.2019. Veracity: The most important "V" of Big Data. Luettu 11.01.2020. https://www.gutcheckit.com/blog/veracity-big-data-v/

InfoTrust. N.d. What Is BigQuery and Why Is It Such a Hot Topic?. Luettu 02.02.2020. https://infotrust.com/articles/what-is-bigquery/

Lakshmanan Lak. 05.02.2019. BigQuery ML gets faster by computing a closed-form solution (sometimes). Luettu 08.02.2020. https://medium.com/google-cloud/bigquery-ml-gets-faster-by-computing-a-closed-form-solution-sometimes-1baa5a838eb6

Marr Bernard. N.d. Deep Learning Vs Neural Networks - What's The Difference?. Luettu 20.01.2020. https://bernardmarr.com/default.asp?contentID=1789

Mutuvi Steve. 16.04.2019. Introduction to Machine Learning Model Evaluation. Luettu 17.02.2020. https://heartbeat.fritz.ai/introduction-to-machine-learning-model-evaluation-fa859e1b2d7f

Perera Shanika. 20.08.2019. An introduction to Reinforcement Learning. Luettu 28.01.2020. https://towardsdatascience.com/an-introduction-to-reinforcement-learning-1e7825c60bbe

PromptCloud. 20.03.2019. World Happiness Report 2019. Luettu 13.02.2020. https://www.kaggle.com/PromptCloudHQ/world-happiness-report-2019

S. Emmanuel. 20.02.2019. Multilayer Perceptron (MLP), Artificial Neural Network (ANN), and Deep Learning. Luettu 21.01.2020. https://www.meetup.com/en-AU/Deep-Learning-for-Sciences-Engineering-and-Arts/events/257483663/

Schneider Guillermina. 19.10.2018. Economic Freedom of the World. Luettu 10.02.2020. https://www.kaggle.com/gsutters/economic-freedom

Seebo. N.d. Machine learning and AI in manufacturing. Luettu 24.01.2020. https://www.seebo.com/machine-learning-ai-manufacturing/

Shankar Ramya. 17.12.2019. Data Science vs. Machine learning. Luettu 16.01.2020. https://hackr.io/blog/data-science-vs-machine-learning

Shin Terence. 12.01.2019. An Extensive Step by Step Guide to Exploratory Data Analysis. Luettu 16.01.2020. https://towardsdatascience.com/an-extensive-guide-to-exploratory-data-analysis-ddd99a03199e

Silipo Rosaria. 19.11.2018. Practicing Data Science. Luettu 13.01.2020. https://www.dataversity.net/practicing-data-science/

Sisense. N.d. Google Big Query Data Warehouse. Luettu 02.02.2020. https://www.sisense.com/glossary/google-big-query-data-warehouse/

Smith Alivia. 04.07.2019. 7 Fundamental Steps to Complete a Data Project. Luettu 13.01.2020. https://blog.dataiku.com/2019/07/04/fundamental-steps-data-project-success

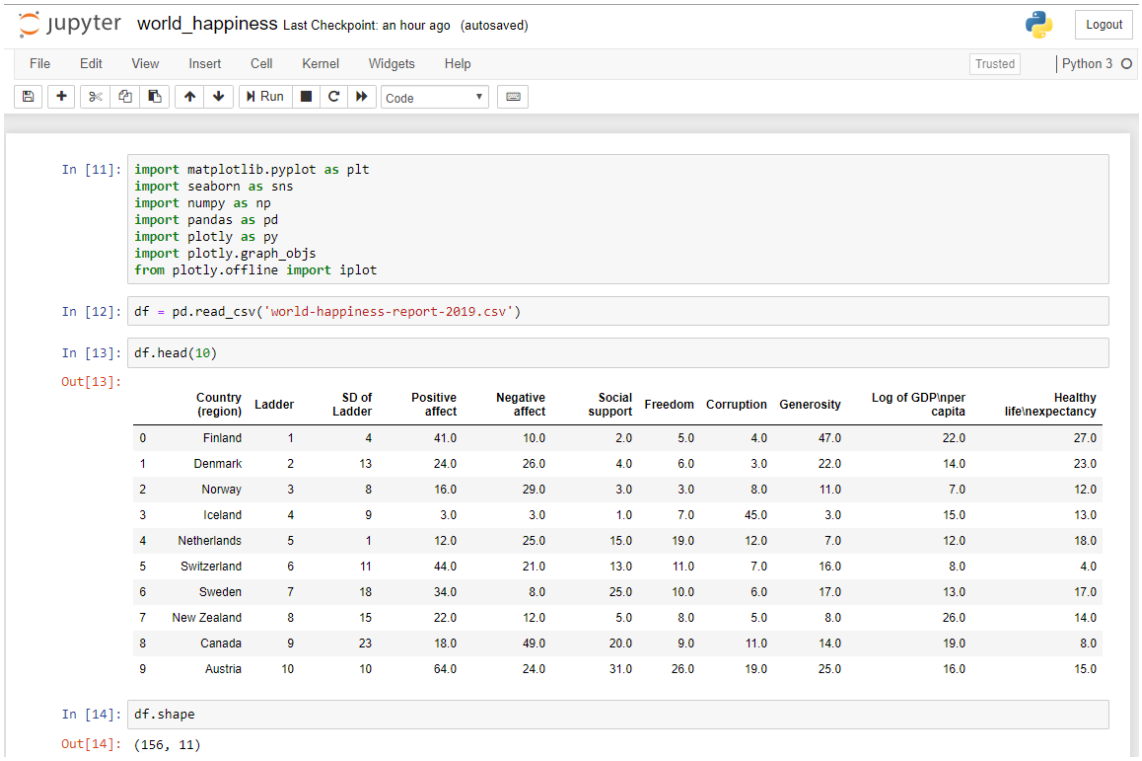
Soni Devin. 22.03.2018. Supervised vs Unsupervised Learning. Luettu 23.01.2020. https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Srinidhi Sunny. 19.11.2019. Data Science vs. Artificial Intelligence vs. Machine Learning vs. Deep Learning. Luettu 07.01.2020. https://towardsdatascience.com/data-science-vs-artificial-intelligence-vs-machine-learning-vs-deep-learning-9fadd8bda583

University of Wisconsin. N.d. What is Big Data?. Luettu 10.01.2020. https://datasciencedegree.wisconsin.edu/data-science/what-is-big-data/

LIITTEET

Liite 1. Tuodut kirjastot ja datajoukon lukemiseen käytetty koodi



jupyter world_happiness Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

In [11]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
import plotly as py
import plotly.graph_objs
from plotly.offline import iplot
```

In [12]:
```
df = pd.read_csv('world-happiness-report-2019.csv')
```

In [13]:
```
df.head(10)
```

Out[13]:

	Country (region)	Ladder	SD of Ladder	Positive affect	Negative affect	Social support	Freedom	Corruption	Generosity	Log of GDP\nper capita	Healthy life\nexpectancy
0	Finland	1	4	41.0	10.0	2.0	5.0	4.0	47.0	22.0	27.0
1	Denmark	2	13	24.0	26.0	4.0	6.0	3.0	22.0	14.0	23.0
2	Norway	3	8	16.0	29.0	3.0	3.0	8.0	11.0	7.0	12.0
3	Iceland	4	9	3.0	3.0	1.0	7.0	45.0	3.0	15.0	13.0
4	Netherlands	5	1	12.0	25.0	15.0	19.0	12.0	7.0	12.0	18.0
5	Switzerland	6	11	44.0	21.0	13.0	11.0	7.0	16.0	8.0	4.0
6	Sweden	7	18	34.0	8.0	25.0	10.0	6.0	17.0	13.0	17.0
7	New Zealand	8	15	22.0	12.0	5.0	8.0	5.0	8.0	26.0	14.0
8	Canada	9	23	18.0	49.0	20.0	9.0	11.0	14.0	19.0	8.0
9	Austria	10	10	64.0	24.0	31.0	26.0	19.0	25.0	16.0	15.0

In [14]:
```
df.shape
```

Out[14]: (156, 11)

Liite 2. Puutteellisten arvojen täyttämiseen käytetty koodi

In [38]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 11 columns):
Country (region)            156 non-null object
Ladder                      156 non-null int64
SD of Ladder                156 non-null int64
Positive affect             155 non-null float64
Negative affect             155 non-null float64
Social support              155 non-null float64
Freedom                     155 non-null float64
Corruption                  148 non-null float64
Generosity                  155 non-null float64
Log of GDP
per capita      152 non-null float64
Healthy life
expectancy    150 non-null float64
dtypes: float64(8), int64(2), object(1)
memory usage: 13.5+ KB
```

In [39]:
```
df['Positive affect'].fillna(df['Positive affect'].mean(),inplace=True)
df['Negative affect'].fillna(df['Negative affect'].mean(),inplace=True)
df['Social support'].fillna(df['Social support'].mode(),inplace=True)
df['Freedom'].fillna(df['Freedom'].median(),inplace=True)
df['Corruption'].fillna(df['Corruption'].mean(),inplace=True)
df['Generosity'].fillna(df['Generosity'].median(),inplace=True)
df['Log of GDP\nper capita'].fillna(df['Log of GDP\nper capita'].mean(),inplace=True)
df['Healthy life\nexpectancy'].fillna(df['Healthy life\nexpectancy'].mean(),inplace=True)
```

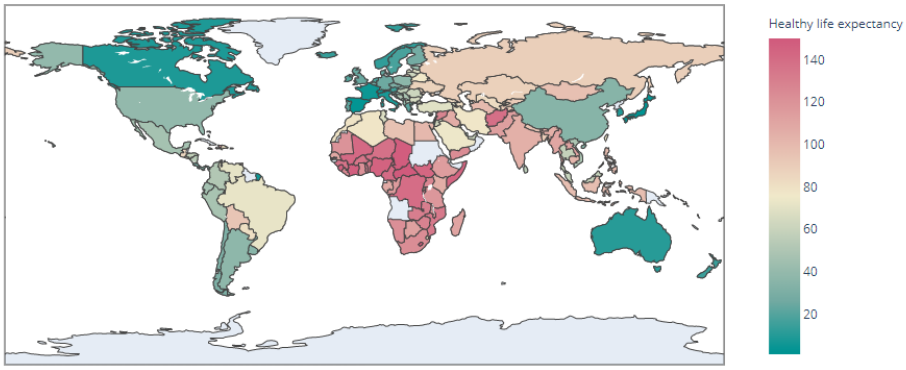
In [40]:
```
df.isnull().sum()
```

Out[40]:
```
Country (region)            0
Ladder                      0
SD of Ladder                0
Positive affect             0
Negative affect             0
Social support              0
Freedom                     0
Corruption                  0
Generosity                  0
Log of GDP\nper capita      0
Healthy life\nexpectancy    0
dtype: int64
```

In [41]:
```
df.to_csv('world_happiness_2019.csv', index=False)
```

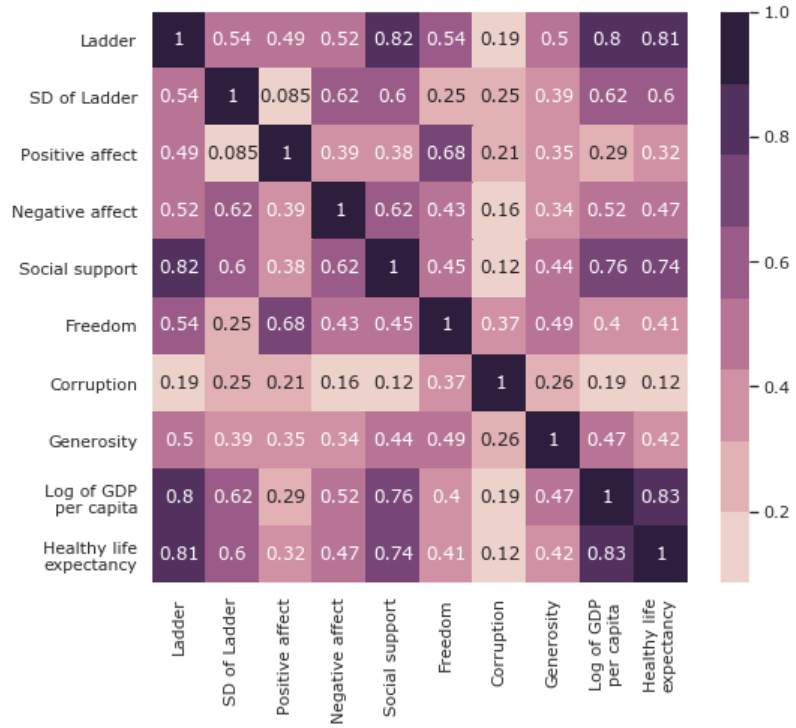
Liite 3. Maailmankartan tekemiseen käytetty koodi

In [20]:
```
data = dict(type = 'choropleth',
            locations = df['Country (region)'],
            locationmode = 'country names',
            z = df['Healthy life\nexpectancy'],
            text = df['Country (region)'],
            colorbar = {'title':'Healthy life expectancy'},
            reversescale = False,
            colorscale='tealrose')
layout = dict(geo = dict(showframe = True, projection = {'type': 'equirectangular'}))
choroplethmap = py.graph_objs.Figure(data = data, layout=layout)
iplot(choroplethmap)
```



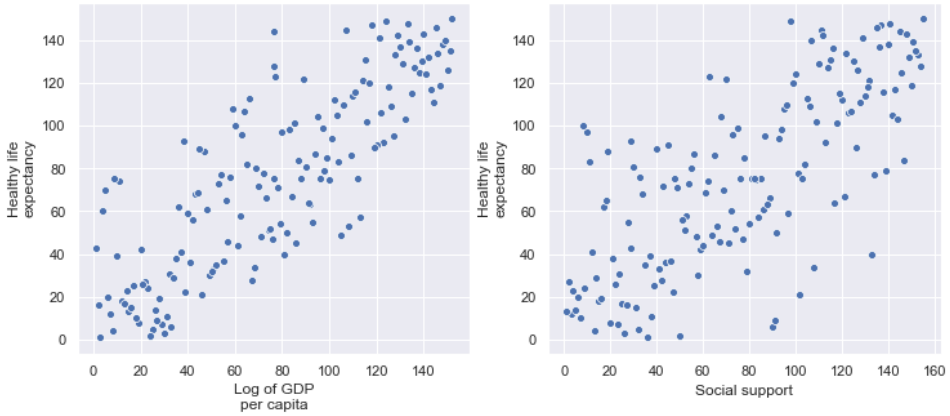
Liite 4. Korrelaatiomatriisin tekemiseen käytetty koodi

```
In [42]: plt.figure(figsize=(8,7))
         plt.rcParams["font.family"] = "Verdana"
         cmap = sns.cubehelix_palette(8)
         ax = sns.heatmap(df.corr(), annot=True, cmap=cmap)
         bottom, top = ax.get_ylim()
         ax.set_ylim(bottom + 0.5, top - 0.5)
         plt.show()
```



Liite 5. Hajontakaavioiden tekemiseen käytetty koodi

In [48]:
```
sns.set()
fig, ax = plt.subplots(1, 2, figsize=(12,5))
sns.scatterplot(data=df, y='Healthy life\nexpectancy', x='Log of GDP\nper capita', ax=ax[0])
sns.scatterplot(data=df, y='Healthy life\nexpectancy', x='Social support', ax=ax[1])
plt.show()
```



Liite 6. Mallin ennusteiden regressiokuvaajaan käytetty koodi

In [3]:
```
df.head()
```

Out[3]:

	predicted_Healthy_life_expectancy	Positive_affect	Negative_affect	Social_support	Freedom	Corruption	Generosity	Log_of_GDP_per_capita	Healthy_life_exp
0	124.369210	54.0	102.0	144.0	21.0	2.0	90.0	132.0	
1	19.226368	24.0	26.0	4.0	6.0	3.0	22.0	14.0	
2	24.534173	41.0	10.0	2.0	5.0	4.0	47.0	22.0	
3	27.647338	22.0	12.0	5.0	8.0	5.0	8.0	26.0	
4	24.760405	34.0	8.0	25.0	10.0	6.0	17.0	13.0	

In [12]:
```
sns.set()
plt.figure(figsize=(12,7))
plt.rcParams["font.family"] = "Verdana"
sns.regplot(data = df, x='Healthy_life_expectancy', y='predicted_Healthy_life_expectancy')
plt.xlabel('Healthy life expectancy')
plt.ylabel('Prediction')
plt.show()
```