Antti Luopajärvi

Selitettävän tekoälyn käyttö lentoliikenteen luokittelussa koneoppimismenetelmillä

Tietotekniikan pro gradu -tutkielma

9. toukokuuta 2022

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Antti Luopajärvi

Yhteystiedot: antti.m.luopajarvi@student.jyu.fi

Ohjaajat: Ilkka Pölönen ja Petri Korpisaari (Insta Advance)

Työn nimi: Selitettävän tekoälyn käyttö lentoliikenteen luokittelussa koneoppimismenetelmillä

Title in English: Use of explainable artificial intelligence in classification of air traffic with machine learning methods

Työ: Pro gradu -tutkielma

Opintosuunta: Laskennalliset tieteet

Sivumäärä: 90+4

Tiivistelmä: Tässä pro gradu-tutkielmassa luokitellaan ilma-aluksia niiden lentoratojen perusteella, jotka muodostetaan käyttäen joukkoistettua havaintodataa. Alukset jäsennellään kategorioihin niiden ensisijaisen käyttötarkoituksen mukaan. Luokitteluun käytetään kolmea eri koneoppimismallia, joiden suorituskykyjä vertaillaan. Lisäksi mallien toimintaa ja ennustuksia tulkitaan selitettävän tekoälyn metodein, käyttäen sekä mallien sisäisiä lukemia että Shapley-arvoja.

Avainsanat: Luokittelu, lentoliikenne, koneoppiminen, selitettävä tekoäly

Abstract: In this Master's thesis aircraft are classified based on their trajectories, which are formed using crowdsourced observational data. The aircraft are organized according to their primary intented use. Classification is carried out by using three different machine learning algorithms, which are compared based on their performance. In addition, the algorithms' operations and predictions are interpreted with methods of explainable artificial intelligence, by using both the algorithms' inner values and Shapley values.

Keywords: Classification, air traffic, machine learning, explainable AI

Termiluettelo

ADS-B	Automatic Dependent Surveillance Broadcast
ANN	Keinotekoinen neuroverkko
BN	Eränormalisointi
dg	Suoruussuhdeluku
FN	Väärä negatiivinen
FP	Väärä positiivinen
GLU	Gated Linear Unit
ICAO	International Civil Aviation Organization
KNN	K:n lähimmän naapurin menetelmä
PCA	Pääkomponenttianalyysi
SSR	Secondary Surveillance Radar
TN	Oikea negatiivinen
TP	Oikea positiivinen
XAI	Selitettävä tekoäly

Kuviot

Taulukot

Sisällys

1 Johdanto

Viime vuosina joukkoistetun (eng. *crowdsourced*) datan määrä lentoliikenteestä on kasvanut merkittävästi. Varsinkin harrastelijoiden hankkimat ohjelmistoradiot ovat mahdollistaneet edullisen ja helpon tavan havaintodatan keräämiseen ilma-aluksista. Avoimesti saatavilla oleva data avaa monia uusia tutkimusmahdollisuuksia ja -kohteita. Lentoliikennettä tutkimalla voidaan saada tietoa esimerkiksi taloudesta[1] tai pandemioiden kehityksestä (Sun ym. 2022).

Lentoliikennettä on monenlaista, joten sen analysoimiseksi on usein tärkeää pystyä jollakin kriteerillä erottelemaan, mihin kategorioihin havainnoitavat alukset kuuluvat. Jaottelu voidaan tehdä esimerkiksi sillä perusteella, onko alus sotilas- vai siviililentokone, mikä on konetta operoivan yhtiön kotimaa, tai mihin tarkoitukseen konetta ensisijaisesti käytetään. Oli kriteeri mikä tahansa, kategoriatieto tarjoaa arvokasta lisätietoa aluksista ja mahdollistaa lentoliikenteen tutkimisen syvällisemmin kuin jos kaikkia aluksia kohdeltaisiin yhtenä massana. Esimerkiksi turismin historiallista kehitystä analysoidessa voidaan koko lentoliikenteen tarkastelun sijasta keskittyä vain matkustajalentokoneisiin.

Ilma-alukset eivät kuitenkaan itse lähetä tyyppi- ja kategoriatietoa, joten niiden päättelemiseksi on turvauduttava ulkoisiin datalähteisiin, joiden avulla tiedot on mahdollista johtaa. Ne koostuvat muutaman ilmailuviranomaisen tarjoaman datan lisäksi ilmailuharrastajien keräämästä aineistosta. Nämä datalähteet ovat kuitenkin puutteellisia ja sisältävät vanhentunutta tietoa, jonka lisäksi osa tiedosta on sellaista joka aluksen operaattorin on mahdollista väärentää tai muokata tunnistamisen vaikeuttamiseksi (Strohmeier ym. 2021). Siten näihin lähteisiin pohjaavat tyyppi- ja kategoriatiedot eivät välttämättä ole oikein.

Käyttämällä pelkästään ohjelmistoradioilla suoraan kerättyä havaintodataa ilma-aluksista, ja siitä laskettuja piirteitä on mahdollista väistää nämä dataan liittyvät epävarmuudet ja luokitella lentoliikennettä pelkästään lentoratojen perusteella. Luokitteluun voidaan käyttää koneoppimismallia, joka valjastetaan tunnistamaan eri kategorioihin kuuluvat alukset toisis-

1. https://www.bankofengland.co.uk/-/media/boe/files/monetary-policy-report/2020/may/monetary-policy-report-may-2020.pdf

taan. Malli oppii sille syötettyjen esimerkkien avulla riippuvuussuhteet datan ja kategorioiden välillä.

Vaikka malli onnistuisi luokittelemaan lentoja erittäin tarkasti, tosielämän sovelluksissa se ei välttämättä riitä. Jotta tuloksia voitaisiin hyödyntää käytännössä, nousee usein tarve ymmärtää perusteet, joilla alukset luokiteltiin tiettyihin kategorioihin. Luokittelijan toiminnan ymmärtäminen lisää ihmisten luottamusta sen antamiin tuloksiin. Myös mahdolliset viat ja vinoumat on helpompi tunnistaa, kun on mahdollisuus tarkastella logiikkaa päätösten takana.

Tässä pro gradussa tutkitaan lentoliikenteen luokittelua eri koneoppimismalleilla. Tavoitteena on selvittää miten lentoliikennettä voidaan luokitella eri kategorioihin, kuten kaupalliset matkustajalentokoneet, hävittäjät ja yksityiset pienlentokoneet, käyttäen dataa lentoradoista. Erona aikaisempiin tutkimuksiin lentoliikenteen luokittelusta, pyritään lisäksi selittämään mallien tuloksia siten, että tutkitaan mitkä piirteet datassa vaikuttavat tuloksiin vahvimmin.

Tutkimuskysymykset ovat:

1. Miten lentoliikennettä voidaan luokitella koneoppimismalleilla pelkästään lentoratojen perusteella?
2. Miten luokittelevien koneoppimismallien tuloksia voidaan selittää?

Pro gradun tutkimusmetodi on konstruktiivinen tutkimus. Tutkimuskysymyksiin pyritään vastaamaan kehittämällä toimintapa, joka alkaa datan keräämisestä ja tallentamisesta tietokantaan, siirtyen datan esiprosessointiin ja eksploratiiviseen data-analyysiin, jatkaen datan rikastamiseen lisäpiirteitä laskemalla, luokittelumallien kouluttamiseen sekä lopuksi tulosten sekä mallien toiminnan analysointiin ja selittämiseen. Toimintatapa liitetään aiempaan tieteelliseen tutkimukseen vertaamalla tuloksia ja käytettyjä menetelmiä.

2 Kirjallisuuskartoitus

Strohmeier ym. (2021) luokittelivat joukkoistetun ohjelmistoradiodatan avulla ilma-aluksia kategorioihin kuten kaupallinen, hävittäjä, tiedustelu ja harjoitus. He laskivat datasta neljän tyyppisiä piirteitä: paikkapiirteet (esim. korkeus), nopeuspiirteet (esim. kaksiulotteisen karteesisen koordinaatiston suuntaiset nopeudet), kiihtyvyyspiirteet (esim. kaksiulotteisen karteesisen koordinaatiston suuntaiset kiihtyvyydet), sekä lentopiirteet (esim. lennon ajallinen kesto). Luokittelumalleina käytettiin neljää eri mallia: tukivektorikone, k:n lähimmän naapurin menetelmä, päätöspuu ja satunnaismetsä. Parhaiten malleista suoriutui satunnaismetsä, jolla päästiin noin 87 % keskimääräiseen luokittelutarkkuuteen.

Kumar ym. (2021) käyttivät vastaavaa ilma-alusten lähettämää *Automatic Dependent Surveillance-Broadcast* (ADS-B) dataa tutkiessa lentokoneiden laskeutumisten keskeytyksiä (eng. *go-around*). He tunnistivat laskeutumisen kriittiset hetket ja laskivat piirteitä näiltä hetkiltä. Piirteitä olivat mm. koneen etäisyys kiitotien keskiviivasta, koneen kulma suhteessa kiitotiehen ja koneen etäisyys kiitotiehen. Tämän jälkeen lennot klusteroitiin käyttäen tiheysperusteista HDBSCAN-klusterointialgoritmia. Näin lennoista saatiin muodostettua kolme klusteria: nominaaliset, poikkeavat ja lennot, joiden laskeutumisen keskeytys tapahtui paljon normaalia myöhemmin. Klustereiden analysointivaiheessa saatiin selville, että poikkeavien lentojen laskuvaiheen manöövereissa oli enemmän vaihtelua verrattaen nominaalisiin. Lisäksi poikkeavien laskeutumisten keskeytysten kohdalla koneiden liike-energiatasot olivat huomattavasti korkeammat verrattuna nominaalisiin lentoihin.

Gingrass, Singham ja Atkinson (2021) tutkivat, onko mahdollista tunnistaa lentoratojen tyyppejä niiden geometrioiden perusteella. He kouluttivat keinotekoisen neuroverkon luokittelemaan erimuotoisia lentoratoja kategorioihin kuten suora, kaareva, yksisilmukkainen ja kahdeksikko. Mallille annettiin lentoradoista laskettuja piirteitä kuten havaintopisteiden määrä lentoradassa, kokonaismatkan pituus ja absoluuttinen menosuunnan vaihtelu lennon aikana. Malli toimi hyvin suorahkoille lentoradoille, mutta epätyyppilliset lentoradat tuottivat sille vaikeuksia. Luokittelija osasi erottaa hyvin myös yksisuuntaiset ja edestakaiset lentoradat toisistaan.

3 Teoria

3.1 Aineiston esikäsittely

3.1.1 Skaalaus

Monissa aineistoissa piirteiden arvojen skaala vaihtelee paljon. Esimerkiksi piirre, joka kertoo henkilön painon kilogrammoina, saa yleensä pienempiä arvoja kun henkilön palkkaa kuvaava piirre. Tästä syystä eri piirteiden perusteella laskettavissa määreissä, esimerkiksi euklidisessa etäisyydessä, suurempaa kokoluokkaa olevat piirteet dominoivat pienempiä (Aggarwal ym. 2015).

Piirteiden skaalauksella pyritään antamaan jokaiselle piirteelle yhtä suuri painoarvo ja siten parantamaan luokittelumallien suorituskykyä (Han, Kamber ja Pei 2012). Skaalausmenetelmiä on olemassa useita, eikä parhaan menetelmän löytämiseksi tietylle mallille ja käytettävälle aineistolle ole olemassa mitään tiettyä tapaa (Ahsan ym. 2021). Yksi skaalausmenetelmistä on normalisointi, jonka avulla piirteiden arvoalue muutetaan tietylle välille. Toinen tapa skaalata on logaritminen muunnos, jolla pyritään samaan vino jakauma symmetrisemmäksi (Hand, Mannila ja Smyth 2001).

Min-max-normalisoinnissa muunnetaan piirteiden arvoalue useimmiten joko välille $[0,1]$ tai $[-1,1]$. Tämän normalisointimenetelmän on havaittu nopeuttavan keinotekoisten neuroverkkomallien oppimisprosessia (Garca, Luengo ja Herrera 2014). Piirteen A kaikki arvot v saadaan muunnettua halutulle välille $[uusi\_min_A, uusi\_max_A]$ laskemalla:

$$v' = \frac{v - min_A}{max_A - min_A}(uusi\_max_A - uusi\_min_A) + uusi\_min_A, \tag{3.1}$$

jossa min_A ja max_A ovat piirteen A alkuperäiset minimi- ja maksimiarvot, tässä järjestyksessä.

Z-arvo-normalisoinnissa piirteiden jakaumat muunnetaan noudattamaan standardinormaalijakaumaa. Muunnoksen jälkeen arvojen keskiarvoksi tulee 0 ja keskihajonnaksi 1 (Garca, Luengo ja Herrera 2014).

Muunnos lasketaan piirteen A kaikille arvoille v seuraavasti:

$$v' = \frac{v - \overline{A}}{\sigma_A} \tag{3.2}$$

jossa $\overline{A}$ ja σ_A on piirteen A arvojen alkuperäinen keskiarvo sekä keskihajonta, tässä järjestyksessä.

Mikäli piirteen arvojen varianssi on hyvin suuri, voidaan sitä pienentää keinotekoisesti logaritmisella muunnoksella (Leydesdorff ja Bensman 2006). Vino jakauma saadaan näin muunnettua lähemmäs normaalijakaumaa (Benoit 2011). Muunnos voidaan tehdä esimerkiksi ottamalla piirteen arvoista luonnollinen tai kymmenkantainen logaritmi (R. M. West 2021).

3.1.2 Ulotteisuuden pienentäminen

Käsitteellä ”dimensioiden kirous” (Bellman 1961) tarkoitetaan ilmiötä, jossa luokittelumallin laskennallinen vaativuus ja luokitteluvirhe kasvavat radikaalisti aineiston ulottuvuuksien myötä. Aineiston uloitteisuutta pienentämällä pyritään muuntamaan moniuloitteinen data merkitykselliseen, pienempiuloitteiseen esitysmuotoon (Maaten, Postma ja Herik 2009). Yksi tähän käytetyistä menetelmistä on pääkomponenttianalyysi (PCA) (Pearson 1901).

PCA tähtää vähentämään aineiston ulotteisuutta luomalla uudet muuttujat, pääkomponentit, jotka säilyttävät mahdollisimman paljon alkuperäisen aineiston sisältämää vaihtelua eivätkä korreloi keskenään. Pääkomponentit järjestetään niin, että muutama ensimmäinen komponentti sisältää suurimman osan alkuperäisen aineiston muuttujien vaihtelusta (Jolliffe 2011).

PCA:ssa alkuperäinen joukko N kappaletta vektoreita $\{\mathbf{x}_i\}$, $\mathbf{x}_i \in \mathbb{R}^n$ tulee muuttaa uudeksi joukoksi vektoreita $\{\mathbf{y}_i\}$, $\mathbf{y}_i \in \mathbb{R}^m$, niin että $m < n$. Tavoitteena on löytää joukko ortogonaalisia kantavektoreita $[\mathbf{u}_1\mathbf{u}_2 \dots \mathbf{u}_n]$, joiden avulla jokainen vektori $\mathbf{x} \in \mathbb{R}^n$ voidaan esittää uudelleen muodossa $\mathbf{x} = \sum_{k=1}^{n} z_k \mathbf{u}_k$, jossa $z_k = \mathbf{u}_k^T \mathbf{x}$ (Kärkkäinen ja Saarela 2015).

Mahdollisimman lähelle tätä esitystä päästään vektorin $\tilde{\mathbf{x}} = \sum_{k=1}^{m} z_k \mathbf{u}_k + \sum_{k=m+1}^{n} b_k \mathbf{u}_k$ avulla, jossa jälkimmäinen termi on jäännösvirhe $\mathbf{x} - \tilde{\mathbf{x}} = \sum_{k=m+1}^{n} (z_k - b_k)\mathbf{u}_k$, kun jäännösvirhe on mahdollisimman pieni. Kyseessä on siis pienimmän neliösumman virheen minimointi:

$$\frac{1}{2}\sum_{i=1}^{N}\|\mathbf{x}_i-\tilde{\mathbf{x}}_i\|^2=\frac{1}{2}\sum_{i=1}^{N}\sum_{k=m+1}^{n}(z_{i,k}-b_k)^2=\frac{1}{2}\sum_{k=m+1}^{n}\mathbf{u}_k^T\mathbf{C}\mathbf{u}_k, \tag{3.3}$$

jossa $\mathbf{C}$ on aineiston otoskovarianssimatriisi

$$\mathbf{C}=\frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i-\bar{\mathbf{x}}_i)^T(\mathbf{x}_i-\bar{\mathbf{x}}_i), \tag{3.4}$$

jossa $\bar{\mathbf{x}}_i$ on otoskeskiarvovektori. Olkoon $\{\lambda_k,\mathbf{u}_k\}$ symmetrisen matriisin $\mathbf{C}$ k:nnes ominaisarvo ja ominaisvektori, toteuttaen

$$\mathbf{C}\mathbf{u}_k=\lambda_k\mathbf{u}_k, \qquad k=1,\ldots,n \tag{3.5}$$

Hyödyntäen yhtälöä 3.5 ja vektorien $\mathbf{u}_k$ ortogonaalisuutta voidaan yhtälö 3.3 kirjoittaa muodossa

$$\frac{1}{2}\sum_{k=m+1}^{n}\lambda_k. \tag{3.6}$$

Tällöin m kappaletta ominaisvektoreita, jotka vastaavat $\mathbf{C}$:n m kappaletta suurinta ominaisarvoa muodostavat kannan muunnetulle esitysmuodolle. Siten vektorit saadaan muunnettua muotoon $\mathbf{y}_i=\mathbf{u}_k(\mathbf{x}_i-\bar{\mathbf{x}})$, jossa $i=1,\ldots,m$ ja $\mathbf{u}_1$ on suurinta ominaisarvoa vastaava kantavektori, $\mathbf{u}_2$ toiseksi suurinta ja niin edelleen.

3.1.3 Ylinäytteistys

Aineiston kategoriajakauma on epätasainen, jos siinä esiintyvien havaintojen kategorioita ei ole suurin piirtein tasamäärä (Chawla 2005). Epätasainen kategoriajakauma voi vaikuttaa negatiivisesti luokittelumallien, kuten päätöspuiden (Japkowicz ja Stephen 2002) sekä keinotekoisten neuroverkkojen (Mazurowski ym. 2008), (Buda, Maki ja Mazurowski 2018) suorituskykyyn mallien oppimisen painottuessa yliedustettujen kategorioiden havaintoihin.

Yksi keino ongelman ratkomiseksi on ylinäytteistää harjoitusaineisto, lisäten keinotekoisesti

vähemmistökategorioiden havaintojen määrää, kunnes niiden lukumäärä vastaa enemmistökategorian havaintojen määrää (Japkowicz 2000). Eräs tähän käytetyistä menetelmistä on satunnainen vähemmistöylinäytteistys (eng. *random minority oversampling*) (Buda, Maki ja Mazurowski 2018), jossa poimitaan satunnaisesti valittuja otoksia vähemmistökategorioista ja monistetaan niitä, kunnes kategoriajakauma on tasainen.

Vaikka ylinäytteistyksen on havaittu toimivan (Japkowicz 2000), sen on myös havaittu johtavan ylisovittumiseen (Chawla ym. 2002), (Chawla, Japkowicz ja Kotcz 2004), (Wang ym. 2014). Ylisovittumisella tarkoitetaan ilmiötä, jossa luokittelumalli sovittuu harjoitusaineistoon liian hyvin, ja ei kykene ennustamaan kategorioita testiaineistolle, sillä testiaineiston havainnot eivät vastaa tarkalleen harjoitusaineiston havaintoja (Yanminsun, Wong ja Kamel 2011). Tämän lisäksi satunnainen vähemmistöylinäytteistys voi tuottaa vääristyneitä otoksia, sillä lukumäärältään pienissä kategorioissa poikkemat tai kohinaa sisältävät havainnot voivat olla yliedustettuina (Laurikkala 2001).

3.2 Piirrejalostus

Luokitteluaineisto voidaan myös esittää uudessa muodossa laskemalla sille lisäpiirteitä alkuperäisten piirteiden perusteella. Tätä toimintatapaa voidaan kutsua piirrejalostukseksi (eng. *feature engineering*). Menetelmän tarkoituksena on esittää aineisto sellaisessa muodossa, joka parantaa luokittelumallien suorituskykyä (Kuhn ja Johnson 2019). Tässä tutkielmassa jalostetaan piirteitä laskemalla yksittäisten lentoratojen piirteiden arvoista tilastollisia tunnuslukuja, kuten maksimi, sekä lentoratojen muotoa kuvaavia etäisyysgeometria-arvoja.

3.2.1 Etäisyysgeometria

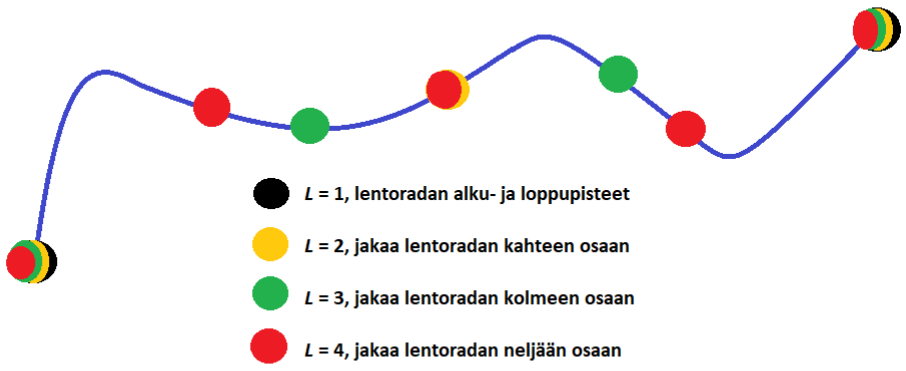
Etäisyysgeometrialla (Menger 1928) tarkoitetaan geometrian alaa, jossa koordinaattisysteeminä käytetään pisteiden sijaan pisteiden välisiä etäisyyksiä. Määritellään nämä etäisyydet seuraavasti (Liberti 2019): Olkoon R joukko $n+1$, $n \in \mathbb{Z}$ kappaletta pisteitä $\{\mathbf{p}_0, \ldots, \mathbf{p}_n\} \in \mathbb{R}^K$, $K \in \mathbb{Z}$. Tällöin kahden pisteen väliset etäisyydet voidaan määritellä joukkona arvoja $d_{\mathbf{p}_i\mathbf{p}_j} = ||\mathbf{p}_i - \mathbf{p}_j||$ kaikille $i, j \leq n$. Joukolle R voidaan määritellä metriikka $d : R \times R \rightarrow [0, \text{inf})$ siten, että

1. $\forall \mathbf{p}_i,\ \mathbf{p}_j \in R \quad d_{\mathbf{p}_i\mathbf{p}_j} = 0 \iff \mathbf{p}_i = \mathbf{p}_j$ (identiteetti);
2. $\forall \mathbf{p}_i,\ \mathbf{p}_j \in R \quad d_{\mathbf{p}_i\mathbf{p}_j} = d_{\mathbf{p}_j\mathbf{p}_i}$ (symmetrisyys);
3. $\forall \mathbf{p}_i,\ \mathbf{p}_j,\ \mathbf{p}_k \in R \quad d_{\mathbf{p}_i\mathbf{p}_j} + d_{\mathbf{p}_j\mathbf{p}_k} \geq d_{\mathbf{p}_i\mathbf{p}_k}$ (kolmioepäyhtälö).

$||\cdot||$ on eulkidinen normi, joka on metriikka $\mathbb{R}^K$:ssa.

Käytännön sovelluksissa etäisyysgeometriaa on käytetty muun muassa sensoriverkkojen paikantamiseen ja molekyylirakenteiden selvittämiseen (Liberti ym. 2012). Etäisyysgeometria mahdollistaa myös ilma-alusten lentoratojen muotojen tarkastelun jakamalla radan osiin ja mittaamalla sen suoruutta sekä kaarrosten kokoja (Neafus ym. 2020). Tällöin lentoradoille lasketaan etäisyysgeometriaa käyttäen ratojen muotoa kuvaavat tunnisteet, jonka avulla voidaan vertailla ratoja toisiinsa.

Käytännössä tämä tapahtuu seuraavasti (Rintoul ja Wilson 2015): Olkoon lentorata θ joukko pisteitä $\{x_0,\ldots,x_n\}, \quad n > 0$. Valitaan syvyys $k, \quad 0 < k \leq n-1$ jonka jälkeen valitaan $L+1, \quad L = 1,\ldots,k$ pistettä lentoradalta tasaisin välein joko etäisyyden tai ajan perusteella niin, että ensimmäinen piste on sama kuin radan alkupiste ja viimeinen piste sama kuin radan loppupiste. Jokainen L:n taso jakaa lentoradan L:ään osaan (kuvio 1).



Kuvio 1: Visualisointi kaksiulotteisen lentoradan jakamisesta osiin, kun $k = 4$

Tämän jälkeen lasketaan etäisyysgeometriatunniste lentoradalle. Tunniste on $k \times \frac{k+1}{2}$ pituinen vektori, jonka alkiot ovat lentoradan osien *suoruussuhdelukuja* (dg). Lentoradan **T** osalle, jonka alkupiste on x_a ja loppupiste x_b, voidaan laskea suoruussuhdeluku siten, että

$$dg(x_a, x_b) = \frac{d_e(x_a, x_b)}{d_t(x_a, x_b)}, \tag{3.7}$$

jossa

$$d_e(x_a, x_b) = ||x_b - x_a||, \tag{3.8}$$

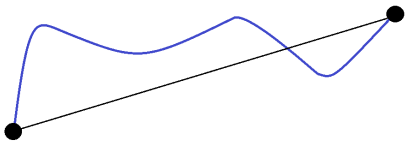
$$d_t(x_a, x_b) = \sum_{i=a}^{b-1} ||x_{i+1} - x_i||. \tag{3.9}$$

Suoruussuhdeluku on siis suhde kahden pisteen euklidisen etäisyyden ja lentorataa pitkin kuljetun matkan välillä. Suhdeluku on väliltä $[0, 1]$, tarkoittaen että suhdeluvun ollessa 1 mitattava lentoradan osa on täysin suora kahden pisteen välillä, ja sen ollessa 0 mitattavan osan alkupiste on sama kuin päätöspiste.

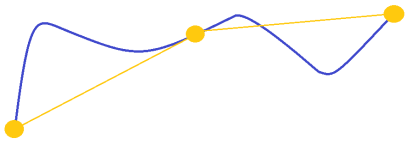
Suoruussuhdeluvut lasketaan jokaiselle L:n tasolle niin, että kun

1. $L = 1$, suoruussuhdeluvut lasketaan lentoradan ensimmäisen ja toisen pisteen väliltä
2. $L = 2$, suoruussuhdeluvut lasketaan lentoradan ensimmäisen- ja toisen, sekä toisen- ja kolmannen pisteen väliltä
3. $L = 3$, suoruussuhdeluvut lasketaan lentoradan ensimmäisen- ja toisen, toisen ja kolmannen, sekä kolmannen- ja neljännen pisteen väliltä

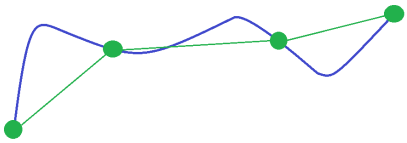
ja niin edelleen. Kuvio 2 havainnollistaa suoruuksien laskemista eri pisteiden välillä.



(a) $L = 1,\ dg_0 = \frac{\text{pisteiden välinen euklidinen etäisyys}}{\text{pisteiden välinen matka lentorataa pitkin}}$

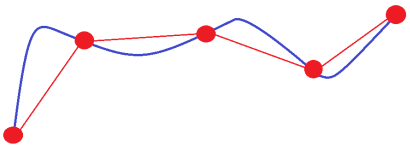


(b) $L = 2,\ dg_{1,2} = \frac{\text{pisteiden välinen euklidinen etäisyys}}{\text{pisteiden välinen matka lentorataa pitkin}}$



(c)

$L = 3,\ dg_{3,4,5} = \frac{\text{pisteiden välinen euklidinen etäisyys}}{\text{pisteiden välinen matka lentorataa pitkin}}$



(d)

$L = 4,\ dg_{6,7,8,9} = \frac{\text{pisteiden välinen euklidinen etäisyys}}{\text{pisteiden välinen matka lentorataa pitkin}}$

Kuvio 2: Suoruussuhdelukujen laskeminen kaksiuloitteiselle lentoradalle, kun $k = 4$

Kun kaikki suoruussuhdeluvut on laskettu tuloksena on etäisyysgeometriatunniste, jossa suoruussuhdeluvut on järjestettynä alkaen tason $L = 1$ suhdeluvuista, seuraten tason $L = 2$ suhdeluvut, ja niin edelleen, kaikille L:n arvoille. Esimerkiksi, kuvion 2 lentoradalle

$$\text{etäisyysgeometriatunniste} = [dg_0, dg_1, dg_2, dg_3, dg_4, dg_5, dg_6, dg_7, dg_8, dg_9].$$

Konveksi verho

Eulkidisessa avaruudessa äärelliselle joukolle Q pisteitä voidaan määritää konveksi verho niin, että se on pienin mahdollinen konveksi polygoni, jolla jokainen joukon Q piste on joko polygonin sisällä tai sen rajalla (Laurini 2017). Tämän avulla voidaan kuvata myös lentoratojen muotoa, esimerkiksi laskemalla radan ala ja sen konveksin verhon akselien suhdeluku.

Olkoon lentorata θ joukko $n+1$, $n \in \mathbb{Z}$ kappaletta pisteitä $\{\mathbf{x}_0, \ldots, \mathbf{x}_n\}$ (Rintoul ja Wilson 2015) . Tällöin joukolle θ voidaan määrittää konveksi verho H_θ, joka koostuu pisteistä $\{\mathbf{c}_1, \mathbf{c}_2, \ldots \mathbf{c}_m\} \subseteq \{\mathbf{x}_0, \ldots \mathbf{x}_n\}$. Tämän konveksin verhon keskipiste $H_{\text{keskipiste}}$ saadaan laskemalla

$$H_{\text{keskipiste}} = \frac{1}{n} \sum_{i=0}^{n} \mathbf{x}_i. \tag{3.10}$$

Lentoradan ala saadaan laskemalla sen konveksin verhon ala. Verhon akselien suhdeluku voidaan määrittää niin, että

$$\text{konveksin verhon akselien suhdeluku} = \frac{\text{pitkän akselin pituus}}{\text{lyhyen akselin pituus}}, \tag{3.11}$$

jossa lyhyen akselin pituudeksi arvioidaan

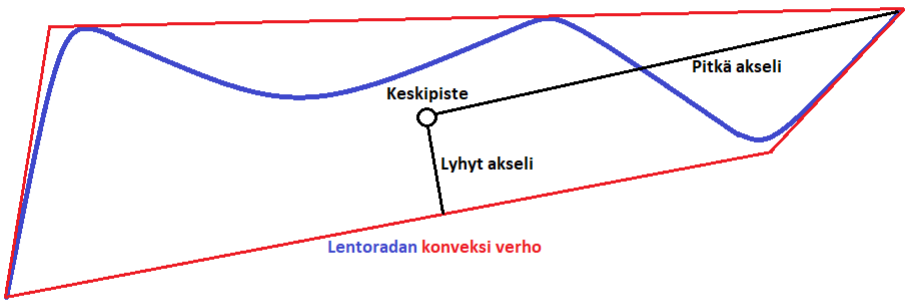
$$\min_{\mathbf{c} \in H_\theta} ||H_{\text{keskipiste}} - \mathbf{c}||. \tag{3.12}$$

Toisin sanoen, konveksin verhon keskipisteen ja lähimmän verhon pisteen välinen etäisyys.

Pitkän akselin pituus saadaan laskemalla

$$\max_{\mathbf{c}\in\theta} ||H_{\text{keskipiste}} - \mathbf{c}||. \tag{3.13}$$

Pitkän akselin tapauksessa suurin etäisyys lasketaan polygonin keskipisteestä kaukaisimpaan lentoradan pisteeseen. Johtuen verhon konveksisuudesta, kaukaisin piste on jokin konveksin verhon kärkipisteistä.



Kuvio 3: Visualisointi kaksiulotteisen lentoradan konveksista verhosta

3.3 Luokittelu

Uusien kohteiden luokittelu samankaltaisten havaintojen perusteella on yksi tyypillisimmistä tiedonlouhinnan tehtävistä (Menardi ja Torelli 2012). Luokittelussa tavoitteena kehittää malli, joka kertoo mihin k:sta luokasta jokin kohde kuuluu. Koneoppimisen kontekstissa luokittelun katsotaan olevan osa ohjattua oppimista (eng. *supervised learning*) (Goodfellow, Bengio ja Courville 2016). Nimellä viitataan siihen, että koneoppimismallia ohjataan tunnistamaan, miten kukin luokka on riippuvainen piirteiden arvoista antamalla sille opetusvaiheessa sekä piirteiden arvot, että luokkatiedon sisältäviä harjoitusesimerkkejä.

Luokittelussa aineisto siis jaetaan pareiksi $(\mathbf{x}_i, y_i)$, $i \in \mathbb{Z}$, jossa $\mathbf{x}_i$ on vektori havaintoa kuvaa-

via piirteitä, jotka voivat olla joko mitattuja, alkuperäisiä piirteitä tai niistä jalostettuja piirteitä. Näillä piirteillä oletetaan olevan vaikutus havainnon luokkatietoon y_i. Toisin sanoin, $\mathbf{x}_i$ on vektori selittäviä muuttujia ja y_i vastemuuttuja. Yleensä joukosta jatkuvia, diskreettejä tai kategorisia muuttujia koostuva $\mathbf{X}$ määritellään niin, että $\mathbf{X} = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathbb{R}^n \text{ kaikilla } i \in \mathbb{Z}\}$. Vastemuuttuja y taas saa arvoja kategorisesta määrittelyjoukosta $\mathbf{y} = \{y_0, \ldots, y_n\}$, $n \in \mathbb{Z}$.

Luokittelutehtävää ratkoessa aineistosta poimitaan otos $T_n = \{(\mathbf{x}_0, y_0), \ldots, (\mathbf{x}_n, y_n)\}$, $n \in \mathbb{Z}$ em. pareja harjoitusaineistoksi. Näitä pareja tarkastellen muodostetaan sääntö $\mathbf{R}_{T_n} : \mathbf{X} \mapsto \mathbf{y}$, jonka avulla voidaan ennustaa harjoitusaineiston ulkopuolisten parien $(\mathbf{x}_j, y_j) \ni T_n$ vastemuuttujan y arvot näkemättä niitä, käyttäen ainoastaan selittäviä muuttujia $\mathbf{x}$.

Luokittelutehtävän hoitamiseen on kehitetty useita malleja. Jokaisella menetelmällä on omat vahvuutensa, heikkoutensa ja implementointiin liittyvät ongelmansa (Phyu 2009), (Gorade, Deo ja Purohit 2017). Paras malli luokittelutehtävän ratkomiseksi riippuukin täysin tehtävän tyypistä (Soofi ja Awan 2017). Tässä tutkielmassa käytetään luokittelutehtävän ratkaisuun kolmea eri mallia, jotka eroavat lähestymistavoiltaan toisistaan, ja pyritään löytämään näistä paras malli tehtävän ratkaisemiseksi.

3.4 Luokittelumallit

3.4.1 K:n lähimmän naapurin menetelmä

Yksi tunnetuimmista koneoppimisluokittelijoista on k:n lähimmän naapurin menetelmä (KNN) (Cover ja Hart 1967). Luokittelu suoritetaan etsimällä datapisteiden k lähintä naapuria ja ennustaen luokka näiden naapurien luokkien perusteella. Menetelmä on helppokäyttöinen, intuitiivinen ja se sopii useisiin eri sovellusaloihin (Bhatia ja Vandana 2010), (Akhil, Deekshatulu ja Chandra 2013), (Suárez ym. 2016).

Käydään läpi KNN-menetelmän ydinidea (Cunningham ja Delany 2021): Olkoon D harjoitusaineisto havaintoja $(\mathbf{x}_i)_{i\in[1,n]}$, jossa $n = |D|$. Jokainen havainto koostuu joukosta F piirteitä, joissa kaikki numeeriset piirteet on normalisoitu välille $[0,1]$, sekä luokkatiedosta y_i. Tavoitteena on luokitella uusi havainto $\mathbf{q}$.

Jokaiselle harjoitusaineiston havainnon $\mathbf{x}_l \in D$ ja $\mathbf{q}$ välille voidaan laskea etäisyys

$$d(\mathbf{q}, \mathbf{x}_l) = \sum_{f \in F} \delta(\mathbf{q}_f, \mathbf{x}_{if}), \tag{3.14}$$

jossa δ on etäisyysmetriikka.

Käytettäviä etäisyysmetriikoita on useita, mutta yleinen tapa aineistolle jossa on sekä jatkuvia että diskreettejä muuttujia on

$$\delta(\mathbf{q}_f, \mathbf{x}_{if}) = \begin{cases} 0 & \text{, kun } f \text{ diskreetti ja } \mathbf{q}_f = \mathbf{x}_{if} \\ 1 & \text{, kun } f \text{ diskreetti ja } \mathbf{q}_f \neq \mathbf{x}_{if} \\ ||\mathbf{q}_f - \mathbf{x}_{if}|| & \text{, kun } f \text{ jatkuva.} \end{cases} \tag{3.15}$$

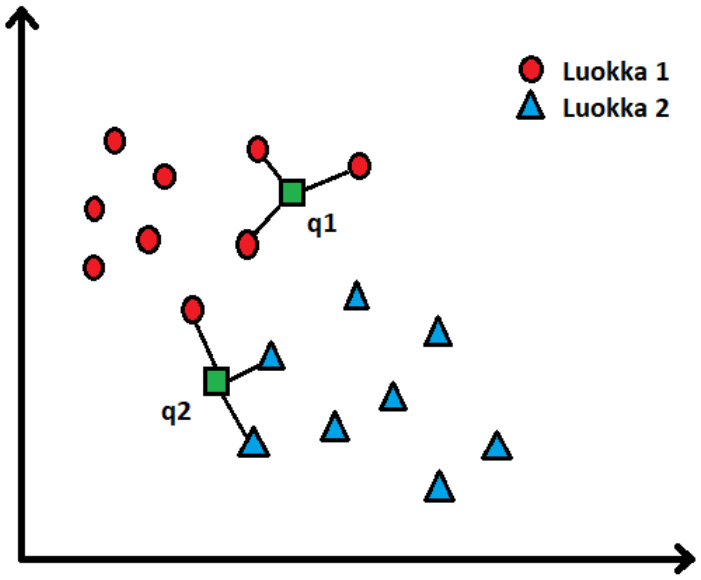
K kappaletta lähimpiä naapureita määritetään tällä metriikalla. Tämän jälkeen $\mathbf{q}$:n luokka päätellään näiden k:n lähimmän naapurin luokkien perusteella, esimerkiksi sen mukaan mitä luokkaa enemmistö naapureista edustaa.

Usein niiden naapurien, jotka ovat kaikista lähimpinä $\mathbf{q}$:ta halutaan saavat enemmän painoarvoa kuin kauempana olevien. Yksi tapa toteuttaa tämä on niis sanottu etäisyyspainotteinen äänestys, jossa naapureiden luokan vaikutus $\mathbf{q}$:n luokkaan painotetaan niiden etäisyyden käänteisluvulla

$$\text{Ääni}(y_j) = \sum_{c=1}^{n} \frac{1}{d(\mathbf{q}, \mathbf{x}_c)} 1(y_j, y_c). \tag{3.16}$$

Tällöin naapurin x_c ääni luokan y_j puolesta on yhtä kuin 1 jaettuna $\mathbf{q}$:n etäisyys naapuriin $\mathbf{x}_c$. $1(y_j, y_c)$ palauttaa 1 mikäli luokat täsmäävät, muuten 0.

Kuvio 4 havainnollistaa kahden uuden havainnon $\mathbf{q}_1$ ja $\mathbf{q}_2$ luokittelua kaksiulotteisessa avaruudessa, jossa naapureiden lukumäärä $k = 3$.



Kuvio 4: Visualisointi luokittelusta kolmen lähimmän naapurin menetelmällä kaksiulotteisessa avaruudessa. Havainto $\mathbf{q}_1$ luokitellaan kuuluvan luokkaan 1, havainto $\mathbf{q}_2$ taas luokkaan 2

3.4.2 Satunnaismetsä

Kokoelmaoppiminen (eng. *ensemble learning*) on ohjatun oppimisen kategoria, jossa koulutetaan useita itsenäisiä luokittelumalleja ja uudet datapisteet luokitellaan näiden yhteistuloksen perusteella (Dietterich 2000). Eräs tähän pohjautuvista menetelmistä on satunnaismetsä (eng. *Random Forest*) (Breiman 2001), jossa luokittelutulokseen päästään kouluttamalla joukko toisistaan riippumattomia päätöspuita ja valitsemalla tulokseksi enemmistön valitsema luokka. Satunnaismetsä on suosittu menetelmä sen helppokäyttöisyyden, suorituskykynsä sekä eri sovellusaloihin sopivuutensa vuoksi (Hastie, Tibshirani ja Friedman 2001), (Strobl ym. 2007), (Ahsan ym. 2021). Lisäksi useat ohjelmistoimplementaatiot satunnaismetsästä tarjoavat työkaluja mallin käyttämien piirteiden tärkeysjärjestyksen selvittämiseen (Lundberg, Erion ja Lee 2018).

Perehdytään satunnaismetsän toimintaan tarkemmin sen sisältämien päätöspuiden kautta. Luokitteleva päätöspuu voidaan määritellä funktiona, jonka tehtävänä on ennustaa havainnon **x** luokkatieto y mahdollisimman tarkasti (Louppe 2014). Käyttäen havaintojen luokkatiedoista ja piirteistä koostuvaa harjoitusaineistoa, päätöspuu koulutetaan erottelemaan havaintojen piirteiden arvojen perusteella luokat toisistaan.

Puun kouluttaminen tapahtuu rakentamalla se solmuista (eng. *nodes*) ja kaarista (eng. *edges*) (Rokach ja Maimon 2005). Aluksi puulle luodaan juurisolmu, jossa harjoitusaineisto jaetaan jollain kriteereillä kahteen tai useampaan osaan sen piirteiden arvojen perusteella. Jos harjoitusaineistossa olisi esimerkiksi piirre ”Ikä”, jakokriteeri voisi olla: ”Onko ikä alle 18 vuotta?”. Solmu pyrkii tekemään jaon niin, että se erottelee eri luokkiin kuuluvat havainnot mahdollisimman hyvin erilleen toisistaan. Yksi tässä käytetyistä mittareista on niin kutsuttu epäpuhtausmitta (eng. *impurity measure*).

Yksinkertaistaen epäpuhtausmitta on luku, joka kuvaa kuinka paljon jaosta syntyneissä osajoukoissa on samaan luokkaan kuuluvia havaintoja. Optimitapauksessa epäpuhtausmitta on 0, jolloin yhdessä osajoukossa on vain yhden luokan havaintoja. Muissa tapauksissa epäpuhtausmitta on välillä $]0, 1[$. Päätöspuun kouluttaminen perustuu siihen, että solmut pyrkivät jakamaan harjoitusaineistoa osiin epäpuhtausmittaa minimoiden. Epäpuhtausmittojen avulla voidaan myös laskea piirteiden tärkeydestä kertovia Gini-kertoimia, joilla mallin luokittelutuloksia voidaan selittää. Epäpuhtausmitan ja Gini-kertoimien laskemista käsitellään vielä tarkemmin kappaleessa 3.6.2.

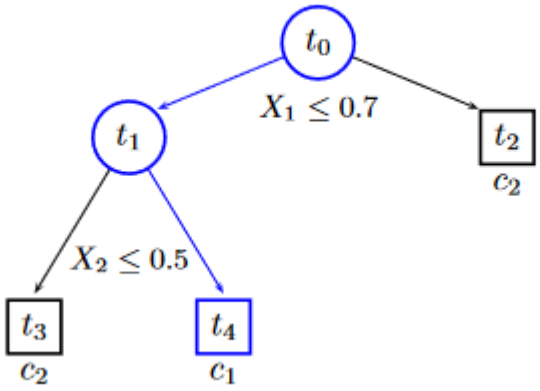
Solmun jaosta syntyneet osaharjoitusaineistot viedään kaaria pitkin seuraaville solmuille, joita kutsutaan lapsisolmuiksi. Näissä solmuissa ne jaetaan edelleen, ja kuljetetaan syntyneet osaharjoitusaineistot jälleen uusia kaaria pitkin uusille lapsisolmuille, joissa prosessi toistetaan. Päätöspuuta kasvatetaan lisäten solmuja ja kaaria, kunnes osaharjoitusainesto saapuu solmuun, jossa jakamista ei enää jonkin kriteerin perusteella jatketa. Näitä mahdollisia pysäytyskriteereitä ovat seuraavat:

- N_{min}: Lopetetaan jakaminen solmuun, jos solmun osaharjoitusaineisto sisältää vähemmän kuin N_{min} havaintoa.
- d_{max}: Lopetetaan jakaminen solmuun, jos sen syvyys (edeltävien jakojen lukumäärä)

$d_t \geq d_{max}$.

- β_{min}: Lopetetaan jakaminen solmuun, jos sen jaon aikaansaava epäpuhtausmitan väheneminen on pienempi kuin raja-arvo β_{min}.
- N_{leaf}: Lopetetaan jakaminen solmuun, jos sitä ei voida jakaa edelleen sellaisiin solmuihin, joissa molemmissa olisi vähintään N_{leaf} havaintoa (Louppe 2014).

Solmua, jossa jakamista ei enää jatketa, kutsutaan lehtisolmuksi (eng. *leaf node*). Lehtisolmuihin asetetaan tieto siitä, mihin luokkaan siihen päätyvät havainnot todennäköisemmin kuuluvat. Kun kaikki harjoitusaineiston havainnot ovat saapuneet johonkin lehtisolmuista, päätöspuu on koulutettu ja se on valmis ennustamaan luokkia uusille havainnoille. Kun puulle nyt annetaan syötteenä uusi havainto, se ”kuljettaa” havainnon sen piirteiden arvojen ja solmujen jakokriteerien perusteella johonkin lehtisolmuista, jonka arvo edustaa puun ennustusta havainnon luokaksi. Satunnaismetsässä tämä toistetaan usealla päätöspuulla, ja lopullinen ennustus valitaan enemmistön ennustaman luokan mukaan.



Kuvio 5: Viisisolmuinen päätöspuu luokittelutehtävän ratkaisuun, jossa solmu t_0 on juurisolmu ja solmut t_2, t_3, t_4 lehtisolmuja. Lehtisolmut edustavat mallin ennustuksia havainnon luokaksi c_i, $i \in \{1,2\}$. Esimerkiksi havainnolle $\mathbf{x} = [\mathbf{x}_1 = 0.2,\ \mathbf{x}_2 = 0.7]$, havainto päätyy lehtisolmuun t_4 ja luokaksi ennustetaan c_1. Kuvion lähde: Louppe (2014)

Satunnaismetsän rakenteeseen voidaan vaikuttaa valitsemalla kuinka monesta päätöspuusta malli halutaan rakentaa. Suurempi määrä puita johtaa yleensä parempiin ennustustuloksiin, mutta kasvattaa laskenta-aikaa (Louppe 2014). Lisäksi voidaan määrittää piirteiden lukumäärä, joiden perusteella solmujen jako suoritetaan. Mallin yleistettävyyden vuoksi päätöspui-

den solmujen jaot tehdään yleensä satunnaisesti valittujen piirteiden perusteella. Siten satunnaismetsän puut oppivat erottelemaan luokkia toisistaan vaihtelevin perustein, joka parantaa mallin suorituskykyä (Louppe 2014).

3.4.3 TabNet

TabNet (Arik ja Pfister 2020) on tässä tutkielmassa käytetyistä kolmesta koneoppimismalleista modernein. Se kuuluu ohjatussa oppimisessa syväoppimisen (eng. *deep learning*) kategoriaan, jonka menetelmillä on viime vuosina rakennettu hyvin tehokkaita, monissa tosielämän sovelluksissa käytössä olevia koneoppimisohjelmia. Syväoppiminen perustuu keinotekoisiksi neuroverkoiksi (eng. *artificial neural networks*) (ANN) kutsuttuihin malleihin. Näiden mallien rakenne mahdollistaa monimutkaisten riippuvuuksien ja konseptien oppimisen yhdistelemällä monia yksinkertaisempia palasia (Goodfellow, Bengio ja Courville 2016).

ANN-mallien tavoitteena luokittelutehtävissä on oppia sääntö $y = f(\mathbf{x}; \theta)$, jolla aineisto $\mathbf{x}$ parametreilla θ johtaa tarkimpaan approksimaatioon luokkatiedon y arvosta. Malli voidaan kuvata verkkona f, joka koostuu useasta eri toisiinsa liitetystä funktiosta (Goodfellow, Bengio ja Courville 2016). Esimerkiksi kuvion 6 verkko, joka sisältää kolme funktiota $f^{(1)}, f^{(2)}, f^{(3)}$, voidaan esittää seuraavasti:

$$f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x}))). \tag{3.17}$$

Verkon sisältämiä funktioita kutsutaan myös verkon kerroksiksi. Koska esimerkin verkko sisältää kolme funktiota, tai kerrosta, sen syvyyden sanotaan olevan kolme. Termillä ”syväoppiminen” viitataan tähän ANN-mallien monikerroksisuuteen. Verkon ensimmäistä kerrosta $(f^{(1)})$ kutsutaan ”syötekerrokseksi”, ja viimeistä $(f^{(3)})$ ”ulostulokerrokseksi”. Muita kun sisääntulo- tai ulostulokerroksia kutsutaan ANN-malleissa ”piilokerroksiksi” $(f^{(2)})$. Näitä piilokerroksia on yleensä syvissä ANN-malleissa useita.

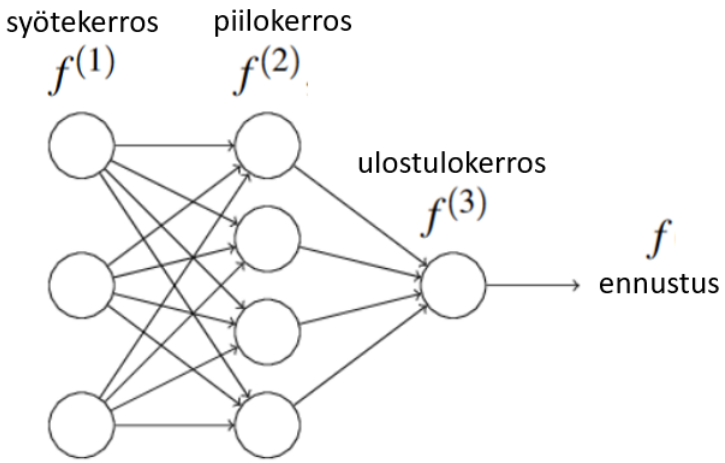
Jokaisen verkon kerroksen voidaan ajatella koostuvan edelleen yhdestä tai useammasta alifunktioista, joita kutsutaan myös ”neuroneiksi”, sillä niiden toiminta perustuu löyhästi ihmisaivojen hermosoluihin. Biologisten esikuviensa tavoin, neuronit siirtävät tietoa toisillensa

kerros kerrokselta verkon läpi. Kun verkolle annetaan syötteenä havainto $\mathbf{x}$, sen syötekerroksen neuronit aktivoituvat ja lähettävät syötteen eteenpäin jokaiselle seuraavan kerroksen neuronille. Tällaista verkkoa, jossa kerroksen neuronit vastaanottavat syötteen kaikilta edellisen kerroksen neuroneilta, kutsutaan ”kokonaan kytketyksi verkoksi” (eng. *fully connected network*) (FC). Ennen kun seuraavan kerroksen neuroni vastaanottaa syötteensä, se skaalaa sille lähetetyt syötteet painokertoimilla $\mathbf{w}$, summaa näin saadut arvot yhteen, ja lisää vielä lopuksi arvoon vakiotermin b. Tämän jälkeen neuroni laskee syötteen perusteella funktionsa arvon, jonka se lähettää edelleen seuraavan kerroksen neuronien syötteeksi. Neuronin laskemaa funktiota kutsutaan sen ”aktivointifunktioksi”. Aktivointifunktioita on olemassa useita, mutta yksi käytetyimmistä on nimeltään ”sigmoid-funktio”

$$\sigma(z) \equiv \frac{1}{1+e^{-z}}, \tag{3.18}$$

jossa $z = \mathbf{w} \cdot \mathbf{x} + b$ (Nielsen 2018).

Edellinen prosessi toistetaan verkon jokaisella kerroksella neuroni neuronilta, kunnes saavutetaan ulostulokerros. Tämän kerroksen neuronien arvoista saadaan mallin ennustus havainnon luokaksi.



Kuvio 6: Esimerkki keinotekoisesta neuroverkosta syvyydellä kolme. Kuvion lähde: Nielsen (2018)

Saatu ennustus on kuitenkin todennäköisesti väärin ensimmäisellä kerralla, sillä verkon neuronien parametrien $\mathbf{w}$ ja b arvot on yleensä valittu satunnaisesti (Nielsen 2018). Tämän takia määritellään virhefunktio (eng. *cost function*), joka laskee eron havainnon oikean luokan ja ennustuksen välillä. Virhefunktion arvo kertoo numeerisesti kuinka hyvin verkko osui ennustuksessaan maaliin. Kuten aktivointifunktioita, myös virhefunktioita on monia, esimerkkinä usein luokittelutehtävissä käytettävä ”ristientropia” (Ho ja Wookey 2019):

$$\text{Ristientropia} = -\sum_{k=1}^{K} y_k \times \log \hat{y}_k, \tag{3.19}$$

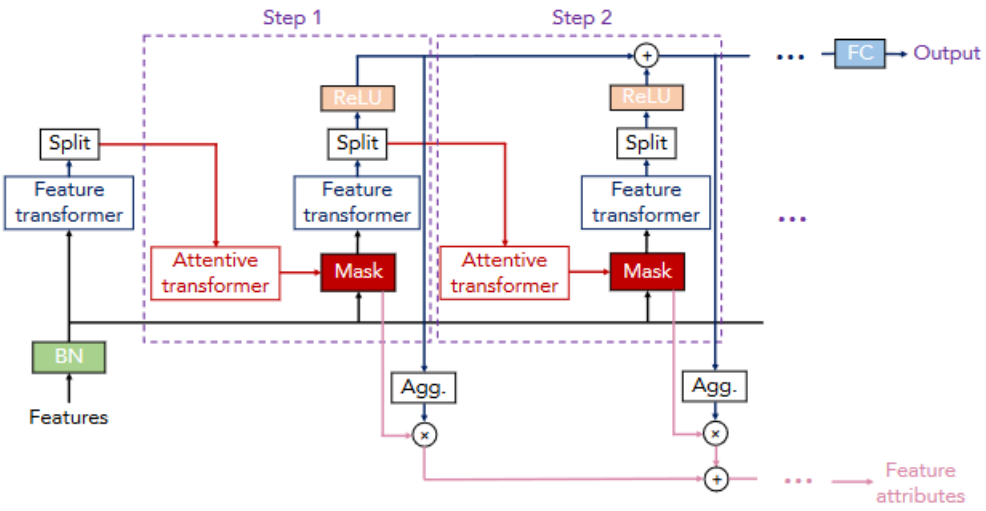
jossa K on luokkien kokonaislukumäärä tehtävässä, $\hat{y}_k$ on havainnon ennustus luokan k suhteen ja y_k arvo, joka kertoo suhteen havainnon oikean luokan ja luokan k välillä. Käytännössä usein havainnoilla on vain yksi oikea luokka, joten $y_k = 1$ yhden K luokista kohdalla, muuten 0.

Virhefunktion arvon ollessa tiedossa, käydään verkko läpi vastavirta-algoritmilla (eng. *back-propagation algorithm*) (Rumelhart, Hinton ja Williams 1986). Algoritmin aikaansaannoksena $\mathbf{w}$ ja b on päivitetty niin, että mikäli havainto syötettäisiin verkolle uudelleen, virhefunktion arvo olisi pienempi ja ennustus vastaisi paremmin oikeaa luokkatietoa. ANN-mallien ”oppiminen” pohjautuu tähän verkon parametrien säätämiseen vastavirta-algoritmin avulla, ja se toistetaan kaikilla verkolle syötetyillä opetushavainnoilla. Kun kaikki opetushavainnot on käyty läpi, on malli opetettu ja valmis ennustamaan luokkia uusille, harjoitusaineiston ulkopuolisille havainnoille.

Eri käyttökohteisiin on kehitetty useita eri ANN-malleja. Ylläkuvattua, yksinkertaista esimerkkiverkkoa kutsutaan ”eteenpäin kytketyksi verkoksi” (eng. *feed forward network*) (Goodfellow, Bengio ja Courville 2016). TabNet:n rakenne on huomattavasti monimutkaisempi; se koostuu useasta aliverkosta, jotka ovat kytkettynä toisiinsa. Pureudutaan seuraavaksi sen arkkitehtuuriin palanen kerrallaan.

TabNet:n rakenteen voi jakaa kahteen osaan; ”enkooderiin” ja ”dekooderiin”. Koska jälkimmäistä osaa tarvitaan vain ohjaamattoman oppimisen tehtävissä, keskitytään tässä tutkielmassa ainoastaan edelliseen. Enkooderi-osan voi ajatella olevan verkko, jonka kerrokset ovat

toisiinsa liitettyjä aliverkkoja. Näitä aliverkkoja on kahta tyyppiä: ”huomioiva transformeri” (eng. *attentive transformer*) sekä ”piirre-transformeri” (eng. *feature transformer*). Edellisen tehtävänä on oppia tunnistamaan oleellisimmat piirteet aineistosta, kun taas jälkimmäinen aliverkko prosessoi piirteet käyttökelpoisempaan muotoon. Enkooderi-osa (kuvio 7) käsittelee syötteensä niin kutsutuissa ”päätösaskelissa” (eng. *decision step*), jossa yhden askeleen aikana syöte kulkee yhden huomioiva transformeri- sekä yhden piirre-transformeri-aliverkon läpi. Päätösaskelien määrä on jokin kokonaisluku väliltä $[3..10]$ (Arik ja Pfister 2020).



Kuvio 7: Tabnet-mallin enkooderi-osan arkkitehtuuri. Kuvion lähde: Arik ja Pfister (2020)

Piirre-transformeri (kuvio 8, vasemmalla) koostuu niin kutsutuista ”piirrelohkoista”, joista jokainen sisältää aiemmin kuvatun kaltaisen kokonaan kytketyn verkon. Tämän verkon ulostulokerroksen neuronien arvot normalisoidaan ”eränormalisointi-menetelmällä” (eng. *batch normalization*) (BN) (Hoffer, Hubara ja Soudry 2017), jonka lisäksi ne syötetään vielä aktivointifunktiolle nimeltä ”gated linear unit” (GLU), joka määritellään seuraavasti:

$$h(\mathbf{x}) = (\mathbf{x} \cdot \mathbf{w} + b) \otimes \sigma(\mathbf{x} \cdot \mathbf{v} + c), \tag{3.20}$$

jossa $\mathbf{w}$:n ja b:n tapaan $\mathbf{v}$ ja c ovat aluksi satunnaisluvuilla alustettuja, myöhemmin vastavirta-algoritmilla säädettäviä parametreja, ja $\otimes$ on Hadamard-tulo (Dauphin ym. 2016).

Lopulta, GLU-aktivointifunktion arvo lähetetään eteenpäin alikerroksen seuraavalle piirre-

lohkolle. Yhteensä näitä piirrelohkoja on jokaisessa piirre-transformerissa neljä. Kaksi piirrelohkoista on niin kutsuttuja jaettuja lohkoja, eli niiden sisältämien verkkojen parametreja säädetään kaikille päätösaskelille yhteisesti. Kaksi lohkoa taas ovat jokaisen päätösaskeleen sisäisiä (Arik ja Pfister 2020).

Toinen aliverkkotyyppi, huomioiva transformeri (kuvio 8, oikealla), koostuu niin ikään kokonaan kytketystä verkosta, eränormalisointifunktiosta, sekä aktivointifunktiosta nimeltä ”Sparsemax”. Tämän funktion arvo lasketaan syötteelle $\mathbf{z}$ seuraavasti (Martins ja Astudillo 2016):

Algorithm 1: Sparsemax-aktivointifunktio

Input: z

Lajittele $\mathbf{z}$, niin että $z_{(1)} \geq \cdots \geq z_{(K)}$;

Etsi $k_{(\mathbf{z})} := \max\{k \in [K] \mid 1 + kz_{(k)} > \sum_{j \leq l} z_{(j)}\}$;

Määrittele $\tau(\mathbf{z}) = \frac{(\sum_{j \leq k(\mathbf{z})}) - 1}{k(\mathbf{z})}$;

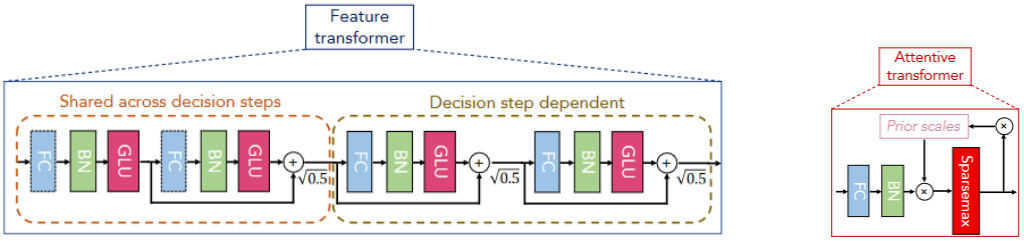
Output: p s.t. $p_i = [z_i - \tau(\mathbf{z})]_+$.

Syöte $\mathbf{z}$ saadaan skaalaamalla kokonaan kytketyn verkon ulostulokerroksen neuronien normalisoidut arvot edellisten päätösaskelten piirteiden tärkeydestä kertovalla termillä

$$\mathbf{P}[\mathrm{i}-1] = \prod_{j=1}^{i-1} (\gamma - M[\mathrm{j}]), \tag{3.21}$$

jossa i on nykyisen päätösaskeleen järjestysnumero, γ joustoparametri, joka määrää kuinka usealle päätösaskeleelle piirteet valikoituvat mukaan laskentaan, ja $M[\mathrm{j}]$ huomioivan transformerin arvo päätösaskeleella j.

Sparsemax-aktivointifunktion tarkoituksena on toimia piirteitä ”harventavana” suodattimena. Funktio prosessoi syötteensä niin, että sen pienimmät arvot pyöristetään nolliksi, jonka johdosta verkko ei siitä eteenpäin huomioi näitä arvoja vastaavia epäoleellisia piirteitä laskennassaan, Arik ja Pfister (2020) kirjoittaa. Laskemalla yhteen enkooderin päätösaskelien huomioivien transformereiden arvot, on mahdollista analysoida mitkä piirteet aineistossa malli katsoi oleelliseksi opetusprossessinsa aikana, ja mitkä se jätti huomiotta. Tämän ominaisuuden ansiosta TabNet:n luokitteluennustuksia on mahdollista selittää sille tärkeimpien piirteiden kautta.



(a) Piirre-transformeri

(b) Huomioiva transformeri

Kuvio 8: TabNet-mallin enkooderi-osan aliverkkojen arkkitehtuurit. Kuvion lähde: Arik ja Pfister (2020)

3.5 Evaluointi

Luokittelumallien evaluoinnilla tarkoitetaan niiden ennustusten osuvuuden mittaamista suhteessa oikeisiin luokkatietoihin. Se on olennainen osa luokitteluprosessia, sillä malleja kehittäessä tarvitaan tapa arvioida niiden toimivuutta käsillä olevan ongelman ratkaisemisessa. Evaluointitapoja on useita ja niiden valinnassa tulee kiinnittää huomiota luokittelutehtävän luonteeseen ja ratkaisutapaan (Novaković ym. 2017).

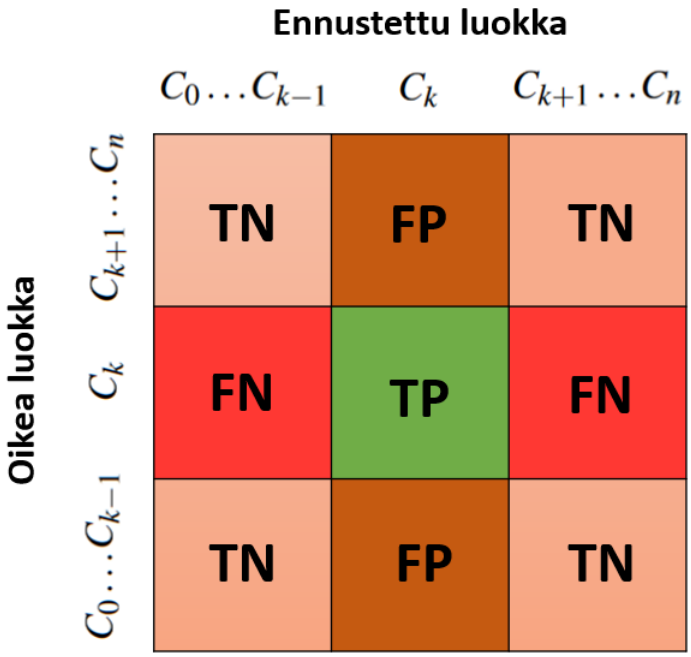
Tässä tutkielmassa luokittelumalleja evaluoidaan kahdella tavalla: sekaannusmatriiseilla (Kulkarni, Chong ja Batarseh 2020) sekä F-arvoilla (Chinchor 1992), joita käytetään varsinkin luokittelussa moneen kategoriaan (Lipton, Elkan ja Narayanaswamy 2014), (Pillai, Fumera ja Roli 2017) ja epätasaisten luokitteluaineistojen kanssa (Chicco ja Jurman 2020), (Hand, Christen ja Kirielle 2021).

3.5.1 Sekaannusmatriisi

Sekaannusmatriisi on $K \times K$ matriisi, jossa K on luokkien lukumäärä, ja jonka jokainen alkio (i, j) kertoo kuinka moni havainto kuuluu oikeasti luokkaan C_i mutta ennustettiin kuuluvan luokkaan C_j. Ideaalitapauksessa kaikki matriisin ei-diagonaalilla sijaitsevat alkiot ovat 0, jolloin kaikki havainnot ennustettiin oikein. Etenkin luokittelutehtävissä, joissa luokkia on useita sekaannusmatriisilla voidaan visualisoida selkeästi, mitkä luokat luokittelijamalli useimmiten sekoittaa keskenään (Alpaydin 2020).

Sekaannusmatriisin (kuvio 9) alkioista saadaan myös oikeiden negatiivisten (TN), väärien positiivisten (FP), väärien negatiivisten (FN), sekä oikeiden positiivisten (TP) ennustusten lukumäärät kategorioittain. Nämä neljä tunnuslukua kertovat ennustusten osuvuudesta seuraavasti:

- TN: Niiden havaintojen lukumäärä, joiden oikea luokka ei ole C_k ja joiden ennuste ei ollut C_k.
- FP: Niiden havaintojen lukumäärä, joiden oikea luokka ei ole C_k mutta joiden ennuste oli C_k.
- FN: Niiden havaintojen lukumäärä, joiden oikea luokka on C_k mutta joiden ennuste ei ollut C_k.
- TP: Niiden havaintojen lukumäärä, joiden oikea luokka on C_k ja joiden ennuste oli C_k.



Kuvio 9: Sekaannusmatriisi luokittelutehtävälle, jossa luokkien lukumäärä $= n$ ja ennustettu luokka $= C_k$. Kuvan lähde: Krüger (2016)

Mikäli kategoriajakauma on hyvin epätasainen ja havaintojen lukumäärä vaihtelee luokittain, voi olla vaikeaa vertailla ennustusten osuvuuksia luokkien välillä. Silloin sekaannusmatriisin tunnusluvuista voidaan laskea edelleen kaksi tunnuslukua: sisäinen tarkkuus (eng. *precision*)

$$P = \frac{TP}{TP+FP}, \tag{3.22}$$

ja herkkyys (eng. *recall*)

$$R = \frac{TP}{TP+FN}. \tag{3.23}$$

Sisäinen tarkkuus kuvaa kuinka moni havainnoista, joille ennustettiin ”kategoria on A” kuului oikeasti kategoriaan A (Cao, Chicco ja Hoffman 2020). Sisäistä tarkkuutta ei tule sekoittaa ulkoiseen tarkkuuteen (eng. *accuracy*), jossa otetaan huomioon myös oikeat ja väärät negatiiviset:

$$\text{Ulkoinen tarkkuus} = \frac{TP+TN}{TP+FN+TN+FP}. \tag{3.24}$$

Herkkyys taas kertoo kuinka monelle oikeasti kategoriaan A kuuluvista havainnoista ennustettiin ”kategoria on A”.

Luokittelijoita evaluoidessa voi joutua valitsemaan, pitääkö sisäistä tarkkuutta vai herkkyyttä tärkempänä (Kotsiantis, Kanellopoulos, Pintelas ym. 2006). Luokittelijan voidaan haluta olevan varmasti oikeassa, jos se ennustaa "kategoria on A", vaikka se tarkottaisi, että joka ikistä kategorian A havaintoa ei saadakaan tunnistettua. Tällöin painotettaisiin sisäistä tarkkuutta. Toisaalta mallin voidaan haluta tunnistavan mahdollisimman moni kategorian A havainnoista oikein, vaikka osa muiden kategorioiden havainnoista tulisikin samalla väärin luokitelluiksi. Tässä tapauksessa herkkyys olisi tärkempi.

Koska tässä tutkielmassa ei ole perusteita kummankaan arvon suosimiseen evaluoinnissa, käytetään luokittelumallien tarkkuuden mittaamiseen nämä molemmat arvot tasapuolisesti huomioon ottavaa painotettua F-arvoa.

3.5.2 Painotettu F-arvo

Käyttäen sekaannusmatriisin tunnuslukuja, painotettu F-arvo saadaan laskemalla yhteen kategorioiden sisäisen tarkkuuden ja herkkyyden harmoniset painotetut keskiarvot:

$$F_p = \sum_{i=1}^{n} w_i \frac{2P_i R_i}{P_i + R_i}, \tag{3.25}$$

jossa

$$n = \text{Kategorioiden lukumäärä}$$

ja

$$w_i = \frac{\text{Havaintojen lukumäärä kategoriassa } i}{\text{Kaikkien havaintojen lukumäärä}}.$$

3.6 Selitettävä tekoäly

Tekoälysovellukset ovat kasvavissa määrin osa yhteiskuntia ja niiden tekemät päätökset vaikuttavat ihmisten elämään muun muassa terveydenhuollon, lakien ja puolustusvoimien kautta (Barredo Arrieta ym. 2020). Nykyiset tekoälyteknologiat suorituvat haastavista tehtävistä jo niin hyvin, että ne tulevat olemaan yhä keskeisemmässä osassa tulevaisuuden ihmiskuntien kehitystä (D. M. West 2018). Sen lisäksi että tekoälyohjelmat itse ratkovat ongelmia, niiden antamaa opastusta käytetään ihmisten neuvomiseen ja päätösten tukemiseen (Doshi-Velez ja Kim 2017a).

Siinä missä ensimmäisten tekoälysovellusten toimintaa oli helppo ymmärtää, nykyisin yhä useampi niistä pohjautuu hankalammin selitettäviin syviin neuroverkkoihin (Barredo Arrieta ym. 2020). Näiden mallien suorituskyvyn takana on kasvava opetusdatan määrä sekä alati monimutkaistuvat arkkitehtuurit. Ne voivat koostua miljardeista parametreista (Brown ym. 2020), joka tekee niiden toiminnan hyvin haastavaksi ymmärtää ja niiden ratkaisut vaikeiksi tulkita. Sen vuoksi tällaisia malleja kutsutaan usein musta laatikko-malleiksi (Castelvecchi 2016).

Vaikkei kaikkien tekoälysovellusten toiminta tarvitsekaan olla ymmärrettävissä, kuten tapauksissa joissa tekoälyn virheellinen toiminta ei aiheuta merkittävää haittaa, monet ratkottavat ongelmat ovat niin moniselitteisiä ja laveasti määriteltyjä, että on tarpeen selvittää miksi ohjelma päätyi tiettyyn ratkaisuun. Kaikkia mahdollisia tilanteita, joissa ohjelma voisi toimia virheellisesti ei esimerkiksi ikinä voida testata tekoälyohjelmaa rakennettaessa; siksi sen tuloksia tulisi voida ymmärtää. Luokittelevan tekoälyohjelman ennuste havainnolle voi myös olla oikea, mutta perusteet eivät yleistettävissä muille havainnoille. Lisäksi kehitettävän tekoälyohjelman voidaan haluta olevan immuuni ihmisiä syrjiville stereotypioille, mutta koska tällainen vaatimus on vaikea ohjelmoida sovellukseen, se pystytään havaitsemaan vain tarkastelemalla syitä ohjelman tekemien ratkaisujen taustalla (Doshi-Velez ja Kim 2017b).

3.6.1 Terminologia

Käsitteellä selitettävä tekoäly (eng. *explainable artificial intelligence*) (XAI) tarkoitetaan tutkimusalaa, joka keskittyy kehittämään uusia metodeja koneoppimismallien selittämiseen (eng. *explain*) ja tulkitsemiseen (eng. *interpret*) (Linardatos, Papastefanopoulos ja Kotsiantis 2021). Nämä kaksi konseptia ovat lähellä toisiaansa; tulkittavat mallit ovat myös selitettäviä, mikäli ihminen voi ymmärtää niiden toiminnan (Adadi ja Berrada 2018). Niitä käytetään myös vuorotellen kirjallisuudessa (Molnar 2022). Termiä ”tulkittava” käytetään kuitenkin useammin kuin ”selitettävä” koneoppimisalan tutkimuksissa, Adadi ja Berrada (2018) kirjoittaa.

Linardatos, Papastefanopoulos ja Kotsiantis (2021) määrittelee selitettävyyden viittaavan mallien sisäisen logiikan, mekaniikan ja menettelytapojen ymmärtämiseen, tulkittavuuden taas mallien syötteen ja ulostulon syy-seuraussuhteiden ymmärtämiseen. Tämän perusteella tässä tutkielmassa keskitytään selitettävän tekoälyn osalta tulkittavuuteen ja tutkitaan miten luokitteluaineiston piirteiden ja mallien ennustusten välisiä yhteyksiä voidaan havainnoida. Mallien selitettävyyden ydin on kuitenkin niiden sisäisen toiminnan ymmärtäminen, jota tutkielman luku 3.4 käsittelee.

Doshi-Velez ja Kim (2017b) määrittelee tulkitsemisen ”kykynä selittää tai esittää koneoppimissysteemi ymmärrettävin termein ihmiselle”. Miller (2019) taas kuvaa seuraavasti: ”tul-

kittavuus mittaa kuinka paljon ihminen ymmärtää jonkin päätöksen syyn". Molnar (2022) toteaa, että malli on sitä paremmin tulkittava, mitä helpompi sen tekemät päätökset on ihmisen ymmärtää.

Tavat tulkittavuuden saavuttamiseksi voidaan jakaa kahteen kategoriaan. Yksinkertaisin tapa on käyttää vain sellaisia malleja, joiden toiminta on itsessään niin läpinäkyvää, että ne ovat tulkittavissa. Esimerkkejä tällaisista malleista on lineaarinen ja logistinen regressio sekä päätöspuu (Molnar 2022). Tällöin puhutaan mallien luontaisesta (eng. *intrinsic*) tulkittavuudesta (Carvalho, Pereira ja Cardoso 2019).

Toinen tapa on käyttää erillisiä analysointimenetelmiä mallin tulkitsemiseen. Tätä kutsutaan post hoc-tulkittavuudeksi. Monimutkaisempia malleja sekä musta laatikko-malleja voidaan tulkita esimerkiksi Gini-kertoimien (Menze ym. 2009) tai Shapley-arvojen avulla (Shapley 1952). Näillä pyritään saavuttamaan tulkittavuutta analysoimalla mitkä piirteet aineistossa ovat mallille tärkeimpiä sen toiminnan kannalta.

3.6.2 Gini-kerroin

Epäpuhtausmittaa, kriteeriä jonka perusteella päätöspuiden solmujen jaot suoritetaan, voidaan kutsua myös "Gini-epäpuhtaudeksi" (Menze ym. 2009). Laskemalla yksittäisten piirteiden vaikutus epäpuhtauden vähenemiseen solmuja jakaessa, voidaan tarkastella mihin piirteisiin päätöspuut nojaavat eniten erotellessaan luokkia toisistaan. Yksittäisten päätöspuiden Gini-epäpuhtaudet yhdistämällä saadaan laskettua niin kutsutut piirteiden "Gini-kertoimet" koko satunnaismetsälle. Näiden lukujen avulla on mahdollista arvioida piirteiden tärkeyttä mallille. Tärkeimpiä piirteitä tarkastelemalla voidaan tulkita mallin toimintaa ja ymmärtää ennustusten logiikkaa.

Määritellään Gini-kerroin seuraavasti (Menze ym. 2009): Olkoon $p_k = \frac{n_k}{n}$ se osajoukko kaikista n otoksesta, jotka kuuluvat luokkaan $k = \{0, 1\}$. Tällöin Gini-epäpuhtaus $i(t)$ saadaan laskemalla

$$i(t) = 1 - p_1^2 - p_0^2. \tag{3.26}$$

Gini-epäpuhtauden väheneminen, Δi, joka aiheutuu solmun jakamisesta lapsisolmuihin t_l ja t_r (ja vastaaviin osajoukkoihin $p_l = \frac{n_l}{n}$ ja $p_r = \frac{n_r}{n}$), raja-arvolla t_θ piirteen θ suhteen voidaan määrittää

$$\Delta i(t) = i(t) - p_l i(t_l) - p_r i(t_r). \tag{3.27}$$

Käymällä läpi kaikki solmun jakamisessa tarkasteltavat piirteet θ ja kaikki mahdolliset raja-arvot t_θ, saadaan muodostettua pari $\{\theta, t_\theta\}$, jolla maksimoidaan Δi. Tämän optimaalisen solmujaon aiheuttama Gini-epäpuhtauden väheneminen $\Delta i_\theta(t,T)$ toistetaan ja lasketaan yhteen jokaiselle piirteelle θ yksitellen. Prosessi toistetaan kaikille solmuille t kaikilla päätöspuilla P satunnaismetsässä:

$$I_G(\theta) = \sum_P \sum_t \Delta i_\theta(t,P) \tag{3.28}$$

Lukema on Gini-kerroin. Sen arvo indikoi kuinka usein piirre θ valittiin jakokriteeriksi solmuja jakaessa sekä kuinka suuri sen kokonaisvaikutus oli mallille luokkia eroteltaessa.

3.6.3 Shapley-arvot

Toinen tapa arvioida piirteiden tärkeyttä mallille on laskea kuinka paljon ne keskimäärin vaikuttivat kuhunkin ennustukseen. Näitä lukuja kutsutaan piirteiden Shapley-arvoiksi. Jos luokittelutehtävää ajalteltaisiin pelinä ja piirteitä pelaajina, Shapley-arvot kertovat kuinka suuri oli kunkin pelaajan kontribuutio pelin lopputulokseen.

Klassinen tapa laskea Shapley-arvot mallin piirteille on käyttää peliteoreettista lähestymistapaa. Menetelmässä malli koulutetaan uudelleen kaikilla piirteiden osajoukoilla $S \subseteq F$, jossa F on kaikkien piirteiden joukko. Näin voidaan havainnoida kunkin piirteen vaikutusta mallin ennustuksiin (Lundberg ja Lee 2017).

Piirteen i vaikutus saadaan laskettua kouluttamalla malli $f_{S\cup\{i\}}$ piirteen i kanssa, sekä toinen malli f_S ilman piirrettä i. Tämän jälkeen verrataan mallien ennustuksia toisiinsa: $f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)$, jossa x_S on osajoukon S piirteiden arvot. Koska piirteiden poisjät-

tämisen aikaansaama vaikutus riippuu muista toisista piirteistä, lasketaan ennustusten erotukset kaikille mahdollisille piirteiden osajoukoille $S \cup F \setminus \{i\}$.

Tämän jälkeen voidaan laskea piirteiden Shapley-arvot ϕ_i. Ne saadaan ottamalla painotettu keskiarvo kaikista mahdollisista ennustusten erotuksista:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]. \tag{3.29}$$

Koska Shapley-arvojen laskeminen kaikille mahdollisille kombinaatioille vaatii paljon laskentatehoa, yleensä tapana on estimoida niitä M:n otoksen perusteella. M:n arvolle ei ole olemassa mitään sääntöä, mutta sen pitäisi olla luku, jolla estimaatit saadaan laskettua tarkasti järjellisessä laskenta-ajassa (Molnar 2022).

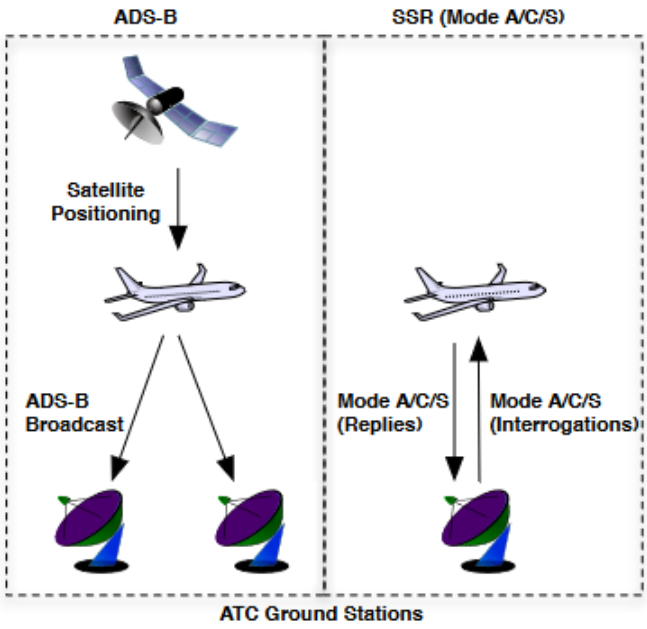
4 Aineisto

Aineistona käytettiin avointa joukkoistettua tilavektoridataa ilma-aluksista sekä niihin liitettävää metadataa. Aineisto hankittiin The OpenSky Network-tietokannan (Schäfer ym. 2014) kautta. The OpenSky Network on sveitsiläinen voittoa tavoittelematon yhdistys, jonka tavoitteena on edistää ilmatilan turvallisuutta, luotettavuutta ja tehokkuutta tarjoamalla kaikille avointa dataa ilmaliikenteestä.

4.1 Tilavektoridata

4.1.1 Teknologia

Joukkoistettu tilavektoridata siviililentoliikenteestä perustuu pääsääntöisesti kahteen teknologiaan. *Secondary Surveillance Radar* (SSR)- teknologia käyttää niin kutsuttuja transpoderitiloja A, C ja S ilma-alusten tietojen, kuten korkeuden ja aluksen transponderikoodin, kyselemiseen kohteilta. Uudempi *Automatic Dependent Surveillance-Broadcast* (ADS-B)-protokolla taas toimii niin, että ilma-alukset lähettävät itsestään säännöllisin väliajoin tietoja kuten oma identifioiva ICAO (*International Civil Aviation Organization*)-tunnus, paikkatieto sekä nopeus. Toisin kuin SSR-teknologiassa, ADB-S ei vaadi kyselyitä ilma-aluksille, vaan ne lähettävät tiedot automaattisesti (Strohmeier ym. 2021). Teknologioiden toimintaa on havainnollistettu kuviossa 10.



Kuvio 10: ADS-B- ja SSR-teknologioiden toiminta. Kuvion lähde: Strohmeier ym. (2021)

4.1.2 Kerääminen

The OpenSky Network tarjoaa eri tapoja tilavektoridatan lataamiselle:

1. Dataa voi ladata csv-tiedostoina päivämäärästä 25.5.2020 nykyhetkeen joka maanantailta.[1] Joka maanantailta data on edelleen järjestetty tunneittain omiin tiedostoihin (kuvio 11).
2. Dataa voi ladata ohjelmistorajapinnan kautta.[2] Rajapinta tarjoaa viimeisimmän tilavektoridatan ja ilman käyttäjätunnusta sillä voi tehdä uuden kyselyn 10 sekunnin välein.
3. Anomalla käyttöoikeutta koko datavarastoon, dataa voidaan hakea Shell-sovelluksen kautta.[3] Tämän kautta voidaan saada dataa useamman vuoden ajalta.

Näistä tavoista valittiin tapa 1. Näin saatiin ladattua valmiiksi ajallisesti järjestettyä historiallista dataa. Lataamista varten kirjoitettiin Python-ohjelma, joka haki ja purki tiedostot

1. https://opensky-network.org/datasets/states/
2. https://github.com/openskynetwork/opensky-api
3. https://opensky-network.org/data/impala

automaattisesti.

/datasets/states/2020-05-25/

Filename	Type	Size	Date Modified
.. (Parent Directory)	<System Dir>	<Directory>	Dec 7 2021 3:05 AM
00	<Directory>	<Directory>	May 26 2020 3:08 AM
01	<Directory>	<Directory>	May 26 2020 3:11 AM
02	<Directory>	<Directory>	May 26 2020 3:13 AM
03	<Directory>	<Directory>	May 26 2020 3:16 AM
04	<Directory>	<Directory>	May 26 2020 3:18 AM
05	<Directory>	<Directory>	May 26 2020 3:20 AM

Kuvio 11: Otos hakemiston rakenteesta yhdeltä päivältä ladattaessa tilavektoridataa tavalla 1

4.1.3 Kuvaus

Tilavektoridata on rakenteellista dataa, jossa yksi havainto kertoo tietoa yhdestä ilma-aluksesta tietyllä ajanhetkellä. Datassa on seuraavat piirteet:

1. time: Unix-aikaleima (kokonaisluku)
2. icao24: 24-bittinen ICAO-transponderitunniste (merkkijono)
3. lat: Leveyskoordinaatit (asteita)
4. lon: Pituuskoordinaatit (asteita)
5. velocity: Maanopeus (m/s)
6. heading: Kulkusuunta (asteita)
7. vertrate: Vertikaalinen nopeus, kertoo kohteen korkeuden muutoksesta (m/s)
8. callsign: Kohteen kutsutunnus (merkkijono)
9. onground: Onko kone ilmassa vai maassa (totuusarvo)
10. alert: Lennonjohdon käyttämä tieto (totuusarvo)
11. spi: Lennonjohdon käyttämä tieto (totuusarvo)
12. squawk: Toinen transponderitunniste (liukuluku)
13. baroaltitude: Korkeus barometrilla mitattuna (liukuluku)
14. geoaltitude: Korkeus GPS:llä mitattuna (liukuluku)
15. lastposupdate: Kulunut aika viimeisimmästä havainnosta (liukuluku)
16. lastcontact: Viimeisimmän havainnon Unix-aikaleima (liukuluku)

time	1590973200	1590973200	1590973200	1590973700	1590973900
icao24	ad7bdf	a2fb32	a87039	aacfbe	a97dcb
lat	41.813965	41.732173	42.150833	40.874913	40.733688
lon	-87.776354	87.663234	-87.799426	102.425849	-112.090210
velocity	100.863828	107.763432	58.702997	255.881984	105.268870
heading	245.918825	190.451633	138.908544	79.458183	175.515394
vertrate	17.23136	0.97536	-0.65024	-0.32512	-6.17728
callsign	SWA793	SWA1433	N64287	EJA796	DAL460
onground	False	False	False	False	False
alert	False	False	False	False	False
spi	False	False	False	False	False
squawk	1341.0	3110.0	5144.0	1074.0	1616.0
baroaltitude	967.74	815.34	586.74	12496.80	3154.68
geoaltitude	1013.46	891.54	617.22	13030.20	3261.36
lastposupdate	1.590973e+09	1.590973e+09	1.590973e+09	1.590974e+09	1.590974e+09
lastcontact	1.590973e+09	1.590973e+09	1.590973e+09	1.590974e+09	1.590974e+09

Taulukko 1: Otos tilavektoridatasta

4.2 Metadata

4.2.1 Kerääminen

The OpenSky Network tarjoaa myös metadataa lentoliikenteestä.[4] Aineisto on koostettu useasta eri lähteestä, kuten viranomaisien lentoliikennerekistereistä ja joukkoistetusti kerätystä datasta. Aineisto on ladattavissa csv-tiedostoina.

4.2.2 Kuvaus

Metadataa on erityyppistä. Tyypin 1 metadata on rakenteellista dataa, jossa yksi havainto kertoo tietoa ICAO-tunnisteeseen liitetystä ilma-aluksesta. Datassa on piirteitä kuten:

- icao24: 24-bittinen ICAO-transponderitunniste (merkkijono)
- manufacturername: Valmistaja (merkkijono)
- model: Malli (merkkijono)
- icaoaircrafttype: Tyyppi (merkkijono)
- categoryDescription: Kategorian kuvaus (merkkijono)

icao24	ae0321	4b1a39	aca3e6	4b4408	ae19ee
registration	74-2134	HB-JYF	N913XJ	HB-ZTW	82-0653
manufacturericao	ROCKWELL	AIRBUS	CANADAIR	KAMAN	FAIRCHILD (1)
manufacturername	Lockheed	Airbus	Bombardier Inc	Kaman Aerospace Corporation	Fairchild
model	C-130H Hercules	A319-111	CL-600-2D24	K-1200	OA-10A Thunderbolt II
typecode	C130	A319	CRJ9	KMAX	A10
serialnumber	4735	4778	15148	A94-0041	A10-0701
linenumber	NaN	NaN	NaN	NaN	NaN
icaoaircrafttype	L4T	L2J	L2J	H1T	L2J
operator	United States Air Force	Easyjet Switzerland Sa	Delta Air Lines	Rotex Helicopter Ag	United States Air Force
...	...	...	...	...	...
modes	False	False	False	False	False
adsb	False	False	False	False	False
acars	False	False	False	False	False
notes	NaN	NaN	NaN	NaN	NaN
categoryDescription	NaN	NaN	NaN	NaN	NaN

Taulukko 2: Otos tyypin 1 metadatasta

4. https://opensky-network.org/datasets/metadata/

Tyypin 2 metadata on myös rakenteellista, ja se sisältää lisätietoa eri tyypisistä ilma-aluksista. Datassa on piirteitä kuten:

- AircraftDescription: Aluksen tyyppi (esim. Landplane, Helicopter) (merkkijono)
- ManufacturerCode: Valmistajan koodi (merkkijono)
- ModelFullName: Aluksen malli (merkkijono)

AircraftDescription	LandPlane	LandPlane	LandPlane	LandPlane	LandPlane
Description	L2P	L1J	L1P	L1T	L2J
Designator	C402	SB35	QUAS	CA7T	F4
EngineCount	2	1	1	1	2
EngineType	Piston	Jet	Piston	Turboprop/Turboshaft	Jet
ManufacturerCode	AVIONES COLOMBIA	SAAB	AEROALCOOL	AEROCOMP	IAI
ModelFullName	402	RF-35 Draken	Quasar Lite	CA-7T Comp Air 7T	F-4 Kurnass 2000
WTC	L	L	M	L	M

Taulukko 3: Otos tyypin 2 metadatasta

5 Tietokanta

5.1 Tietokannan rakenne

Kun tilavektoridata ja metadata oli ladattu ja purettu csv-tiedostoiksi, luotiin PostgreSQL-tietokanta *trajectories*, jonne aineisto tallennettiin. Tietokannanhallintaohjelmana käytettiin pgAdmin 4-ohjelmaa. Tietokantaan luotiin kaksi skeemaa: *import* ja *reference*. *Import*-skeemaan luotiin yksi taulu: *state_vector*, johon tuotiin tilavektoridata. *Reference*-skeemaan luotiin kaksi taulua: *aircraft* ja *aircraft_types*. *Aircraft*-tauluun tuotiin tyypin 1 metadata, ja *aircraft_type*-tauluun tyypin 2 metadata.

Kun tilavektoridatasta oli muodostettu lennot ja lennoista edelleen piirrevektorit, luotiin tietokantaan uusi skeema *data*, ja tallennettiin ne tauluihin *flight* ja *flight_analytics*, tässä järjestyksessä. Tietokannan rakenne on visualisoituna liittessä A.

5.2 Aineiston tuonti tietokantaan

Tilavektoridata tuotiin tietokantaan tauluun *import.state_vector* yksi csv-tiedosto kerrallaan. Yhteensä tiedostoja oli 472 kappaletta ja tuomisen nopeuttamiseksi kirjoitettiin Bash-skripti (liite B). Koko aineiston tuomisen jälkeen taulussa oli 668 945 641 riviä. Ajallisesti aineisto oli väliltä 25.5.2020-16.11.2020.

Tyypin 1 ja 2 metadata tuotiin kumpikin yhdessä csv-tiedostossa tauluihin *reference.aircraft* ja *reference.aircraft_types*. Tuonnin jälkeen tauluissa oli 459999 ja 10020 riviä. Tilavektoriaineistosta luodut lennot sekä niistä muodostetut piirrevektorit tuotiin niin ikään kumpikin csv-tiedostona tauluihin *data.flight* ja *data.flight_analytics*. Molemmissa tauluissa oli tuonnin jälkeen 20232 riviä.

5.3 Aineiston hakeminen tietokannasta

Aineiston analyysivaiheen alussa (kappale 6.1) haettiin SQL-kyselyllä määrätty osajoukko tietokannan taulujen *import.state_vector* ja *reference.aircraft* riveistä (liite C). Myöhemmin

kappaleessa 6.4 tietokannasta haettiin SQL-kyselyllä (liite D) osajoukko taulujen *data.flight* ja *data.flight_analytics* riveistä.

6 Aineiston analyysi

Aineiston analyysi aloitettiin sen visualisoinnilla. Aineistoa kuvattiin muun muassa histogrammijakaumina ja korrelaatiomatriiseina. Seuraavaksi muodostettiin lennot tilavektori- ja metadataa yhdistäen. Tämän jälkeen leimattiin lennot kategorioihin käyttäen kerättyä metadataa sekä eri Internet-lähteitä. Tästä siirryttiin datan esiprosessoinnin kautta luokittelumallien hyperparametrien optimointiin harjoitusaineistolla. Seuraavaksi ennustettiin kategorioita testiaineistolle. Malleja evaluoitiin sekaannusmatriiseilla sekä eri tunnuslukuja käyttäen, jonka jälkeen analysoitiin tuloksia sekä tärkeimpiä piirteitä. Seuraavaksi kuvataan analyysivaihe tarkemmin kohta kohdalta.

6.1 Lentojen muodostaminen

Lennot muodostettiin raa'asta tilavektoridatasta sekä tyypin 1 metadatasta. Tässä käytettiin Python-kirjastoa nimeltä Tracktable[1], jonka on kehittänyt Yhdysvaltojen energiaministeriön tutkimuslaitos Sandia National Laboratories. Kirjasto tarjoaa lentojen muodostamisen lisäksi monia eri tapoja laskea geometriapiirteitä lentoradoille, klusterointialgoritmeja sekä visualisointimenetelmiä.

Kategorioiden leimaamisen helpottamiseksi päätettiin aluksi ottaa analyysiin vain osajoukko aineistosta niin, että valittiin aineisto vain Euroopan alueelta. Leimaamisen tuloksia tarkastellessa havaittiin että aineiston kategoriajakauma oli hyvin epätasainen; valtaosa lennoista, yli 60 %, kuului samaan kategoriaan. Kategoriajakauman tasapainottamiseksi päätettiin laajentaa analysoitavaa osajoukkoa aineistosta niin, että Euroopan alueen lisäksi haettiin tietokannasta aineistoa koko maailmasta tietyillä hakuehdoilla. Nämä hakuehdot valittiin niin, että tarkasteltiin kategoriajakaumaa ja luotiin avainsanoja niistä kategorioista, jotka olivat aliedustettuina aineistossa ja joista haluttiin saada muodostettua lisää lentoja.

Aineisto tallennettiin Pandas-kirjaston datakehys-tietorakenteeseen (DataFrame). Koska Tracktablelle tulee antaa data tiedostomuodossa, tallennettiin datakehys edelleen csv-tiedostoksi.

1. https://tracktable.sandia.gov/

Lentojen muodostaminen Tracktablella tehtiin seuraavasti:

1. Määriteltiin hakemistopolku luettavaan csv-tiedostoon.
2. Luotiin TrajectoryPointReader- olio, joka lukee lentoratapisteet tiedostosta.
3. Määriteltiin oliolle luettava tiedosto ja siinä käytettävä eroitin.
4. Määriteltiin lentoratapisteiden piirteet ja niitä vastaavat sarakkeet tiedostossa. Pakollisia piirteitä on ilma-aluksen identifioiva piirre (tässä icao24-tunniste), aikaleima ja sijaintipiirteet. Näiden lisäksi pisteille asetettiin tieto korkeudesta, nopeudesta, aluksen mallista, valmistajasta sekä operaattorista.
5. Luotiin AssembleTrajectoryFromPoints- olio, joka muodostaa lennot lentoratapisteistä.
6. Asetettiin AssembleTrajectoryFromPoints- olio käyttämään TrajectoryPointReader- olion lukemia lentoratapisteitä.
7. Määritettiin parametrien arvot lentojen luomiselle:
    - Seperation time (Maksimiaika kahden peräkkäisen havainnon välillä jotta ne lasketaan kuuluvan samaan lentoon): 30 minuuttia.
    - Seperation distance (Maksimietäisyys kahden peräkkäisen havainnon välillä jotta ne lasketaan kuuluvan samaan lentoon): 100 kilometriä.
    - Minimum length (Lennon minimipituus havaintojen lukumääränä): 10.
8. Luotiin lennot.

Näin saatiin muodostettua 20 228 lentoa. Taulukossa 4 on esimerkki yhdestä lennon rakenteesta, ja kuviossa 12 on esimerkkilento visualisoituna karttatasoon.

altitude	groundspeed	state_vector_id	manufacturername	model	operator	timestamp	longitude	latitude	icao24	flight_id
11551.92	252.178329	982053.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 04:22:10	30.920937	56.568832	789219	124
11551.92	252.178329	983066.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 04:22:20	30.880404	56.571808	789219	124
11551.92	252.111679	985558.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 04:22:30	30.840454	56.574692	789219	124
11551.92	252.178329	987233.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 04:22:40	30.798540	56.577681	789219	124
11551.92	252.178329	988432.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 04:22:50	30.758200	56.580614	789219	124
...	...	...	...	...	...	...	...	...	...	...
11620.50	262.183799	1738481.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 05:35:00	13.150635	55.045212	789219	124
11612.88	261.969210	1740667.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 05:35:10	13.113764	55.035499	789219	124
11620.50	262.223668	1742346.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 05:35:20	13.077366	55.025940	789219	124
11620.50	262.905045	1743360.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 05:35:30	13.040439	55.016273	789219	124
11620.50	262.905045	1744771.0	Airbus Industrie	A350 1041	Cathay Pacific	2020-05-25 05:35:400	13.002153	55.006310	789219	124

Taulukko 4: Esimerkkilento taulukossa

Kuvio 12: Esimerkkilento karttatasoon piirrettynä

6.2 Kategorioiden leimaaminen

Seuraavaksi muodostetut lennot leimattiin kategorioihin. Kategoriatyypiksi valittiin käyttötarkoitus, joka kuvaa kyseisen lennon pääasiallista tehtävää. Tämä kategoriatyyppi valittiin siksi, että siitä saataisiin useampi eri kategoria muodostettua ja että metadatan perusteella kategorioihin leimaaminen olisi mahdollista toteuttaa kohtuullisella työmäärällä. Toinen vaihtoehto kategoriatyypiksi jota harkittiin oli konetyyppi, joka kuvaa onko alus esimerkiksi suihkukone, potkurikone vai helikopteri. Tässä vaihtoehdossa kategorioiden määrä olisi jäänyt pienemmäksi kuin käyttötarkoitus-kategorialla, mutta toisaalta leimaamistyö olisi ollut nopeampaa, sillä konetyyppitieto löytyi suoraan tyypin 2 metadatasta piirteestä ”AircraftDescription”.

Ensisijaisena piirteenä käyttötarkoituksen selvittämiseksi käytettiin operaattori-kenttää. Tä-

mä kenttä sisälsi usein alusta operoivan lentoyhtiön tai viranomaisen nimen. Malli-kenttä sisälsi osille lennoista myös valmistajan nimen (esimerkiksi ”Boeing C-17 Globemaster III”) ja osille pelkän mallin (esimerkiksi ”45”). Jälkimmäisessä tapauksessa valmistajan nimi saatiin tarvittaessa omasta kentästään.

Käyttäen operaattorin Internet-sivuja selvitettiin minkä tyyppistä lentotoimintaa se harjoittaa. Suurimmalle osalle lennoista kategoria saatiin leimattua näin. Esimerkiksi operaattorin ollessa ”China Airlines” saatiin yhtiön verkkosivuilta[2] selville, että kyseinen lentoyhtiön pääasiallinen lentotoiminta on matkustajareittilennot. Mikäli lentotoimintaa oli useampaa kuin yhdentyyppistä, katsottiin lisäksi aluksen malli- ja valmistajatiedot. Tukena käytettiin muun muassa Wikipedia[3]- Planespotters[4]- ja JetPhotos[5]-sivustoja. Jos tämänkin jälkeen oli epäselvää, mikä kategoria lennolle tulisi leimata, poistettiin lento joukosta. Kun kaikki lennot oli leimattu, kategorioiksi oli saatu seuraavat:

- **Business**: Yksityis- ja liikelennot. Kaikki operaattorit, joiden pääasiallinen lentotoiminta on charter- tai businesslennot. Alukset ovat yleensä keskikokoisia suihkukoneita kuten Bombardier Learjet tai Cessna Citation Jet.
- **Commercial**: Matkustajareittilennot. Operaattorin pääasiallista lentotoimintaa on reittilennot. Alukset ovat suuria, paljon matkustajia kuljettavia suihkukoneita kuten Boeing 767 tai Airbus A350.
- **Cargo**: Rahtilennot. Alukset ovat suuria suihkukoneita kuten kategoriassa ”Commercial”, mutta operaattorin lentotoiminta on rahdin kuljetus. Esimerkkeinä tällaisista lentoyhtiöistä on Lufthansa Cargo ja Cargo Air.
- **Civil surveillance**: Siviilivalvontalennot. Operaattoreiden kuten poliisin ja rajavartiolaitoksen valvontatoimintaan liittyvät lennot kuuluvat tähän kategoriaan.
- **General**: Yleislennot. Tähän kategoriaan kuuluu lentokoulujen lennot, yksityishenkilöiden lennot sekä muihin kategorioihin sopimattomat lennot. Alukset ovat yleensä pieniä yksimoottorisia koneita, kuten Cessna 152.
- **Emergency**: Hälytys- ja pelastuslennot, esimerkiksi ilma-ambulanssilennot.

2. https://www.china-airlines.com/us/en
3. https://www.wikipedia.org/
4. https://www.planespotters.net/
5. https://www.jetphotos.com/

- **Military command and control**: Sotilaalliset koneet, joiden tarkoitus on muun muassa tutkatietoa käyttäen johtaa ja hallinnoida muiden alustyyppien toimintaa. Alukset voivat toimia myös komentokeskuksina maiden korkeimmalle johdolle. Esimerkkialuksia on Boeing E-7A Wedgetail ja Boeing E-4B.
- **Military EW**: Sotilaalliset elektronisen sodankäynnin koneet. Näitä koneita käytetään sekä tutkien että viestiliikenteen häirintään. Esimerkkialuksia on Lockheed EC-130H Hercules ja EC-130J Hercules.
- **Military fighter**: Hävittäjälennot. Tämän kategorian lentoja operoi jonkin maan sotilaallinen viranomainen ja koneet ovat hävittäjiä, kuten General Dynamics F-16 tai Eurofighter Typhoon.
- **Military surveillance**: Sotilaalliset tiedustelukoneet. Esimerkkialuksia on Boeing RC-135 ja Pilatus PC-XII U-28B.
- **Military tanker**: Ilmatankkauslennot. Kategorian lennoissa operaattori on sotilaallinen viranomainen ja koneiden käyttötarkoitus on ilmatankkausoperaatiot. Konetyyppeinä on esimerkiksi Airbusin A330 MRTT.
- **Military training**: Sotilaalliset harjoituslennot. Tämän kategoriaan kuuluu sotilaallisten operaattoreiden lentoharjoitustoiminta pienkoneilla kuten Cessna T-41.
- **Military transport**: Sotilaalliset miehistön- ja rahdinkuljetuslennot. Tähän kategoriaan kuuluu monia eri tyyppisiä koneita, esimerkiksi Lockheedin C-130 Hercules ja Boeingin C-32.
- **Research**: Tutkimus- ja kuvauslennot. Tämän kategorian operaattoreita ovat yritykset, jotka harjoittavat ilmakuvausta tutkimuskäyttöön, koneinaan esimerkiksi Cessna T310R.

Kuten taulukosta 5 nähdään, lentoaineistossa oli kolme suurta kategoriaa: ”Commercial”, ”Military training” sekä ”Military transport”. Nämä muodostivat yhdessä noin 72 % aineistosta. Kategoriajakauma oli kuitenkin tasaisempi kuin mitä se oli ollut pelkät Euroopan alueen lennot haettaessa.

Joidenkin kategorioiden osuus oli alle prosentin, vaikka niitä oli haettu koko maailman alueelta. Huomattava osuus sotilaallisista koneista ei lähetä ADB-S-dataa (Schäfer ym. 2017), joka voi selittää etenkin ”Military fighter”-kategorian lentojen alhaista lukumäärää.

Myös osa siviilikategorioista, esimerkiksi ”General” ja ”Research” olivat suhteellisen pieniä. Koska näihin kategorioihin kuului muun muassa lentokoulutusta sekä tutkimuslentoja tekevien operaattoreiden lennot, olisi niiden lukumäärää ollut mahdollista kasvattaa etsimällä Internetistä lisää tämänkaltaisia operaattoreita. Aineiston epätasainen kategoriajakauma otettiin kuitenkin myöhemmin huomioon aineiston esikäsittelyssä (luku 6.4.4) sekä evaluointitavan valinnassa (luku 3.5).

Vaikka ne lennot, joiden ensisijainen tarkoitus ei ollut mahdollista selvittää luotettavasti hylättiin joukosta, liittyy leimaustuloksiin silti epävarmuutta. Esimerkiksi kategorian ”Business” lennot sisältävät sekä yritysten että yksityishenkilöiden käyttämiä charterlentoja, joten ei voida varmasti tietää onko kyseinen lento tehty yksityis- vai liiketoiminta-asioissa. Huomioitavaa on myös että kategorian ”Commercial” lennot ovat ensisijaiselta toiminnaltaan matkustajareittilentoja, mutta osa niistä kuljettaa ruumassa myös rahtia (Boeing 2022a).

Myös sotilaskoneille voi olla useita eri käyttötapoja. Esimerkiksi kategorian ”Military tanker” aluksia voidaan käyttää useaan eri tarkoitukseen ilmatankkauksen lisäksi (Boeing 2022b). Myös osalla kategorian ”Military transport” koneista voidaan suorittaa erityyppisiä tehtäviä (U.S. Air Force 2022).

Kategoria	Lentojen lukumäärä	Osuus [%]
Business	467	2.3
Cargo	1199	6.0
Civil surveillance	308	1.5
Commercial	4866	24.1
Emergency	453	2.2
General	63	0.3
Military command and control	116	0.6
Military EW	105	0.5
Military fighter	43	0.2
Military surveillance	212	1.0
Military tanker	1150	5.7
Military training	5119	25.3
Military transport	4526	22.4
Research	180	0.9
Unknown	1421	7.0
Yhteensä	20228	100

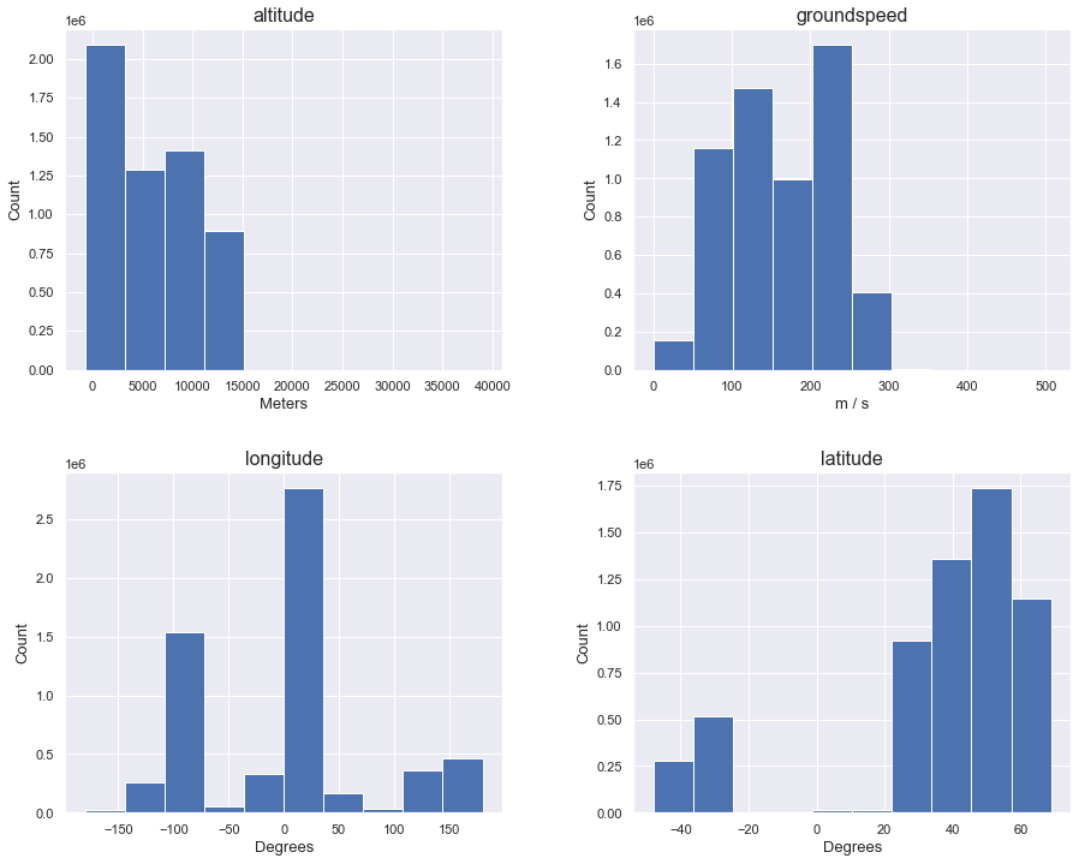
Taulukko 5: Lentoaineiston kategoriajakauma

6.3 Piirrejalostus

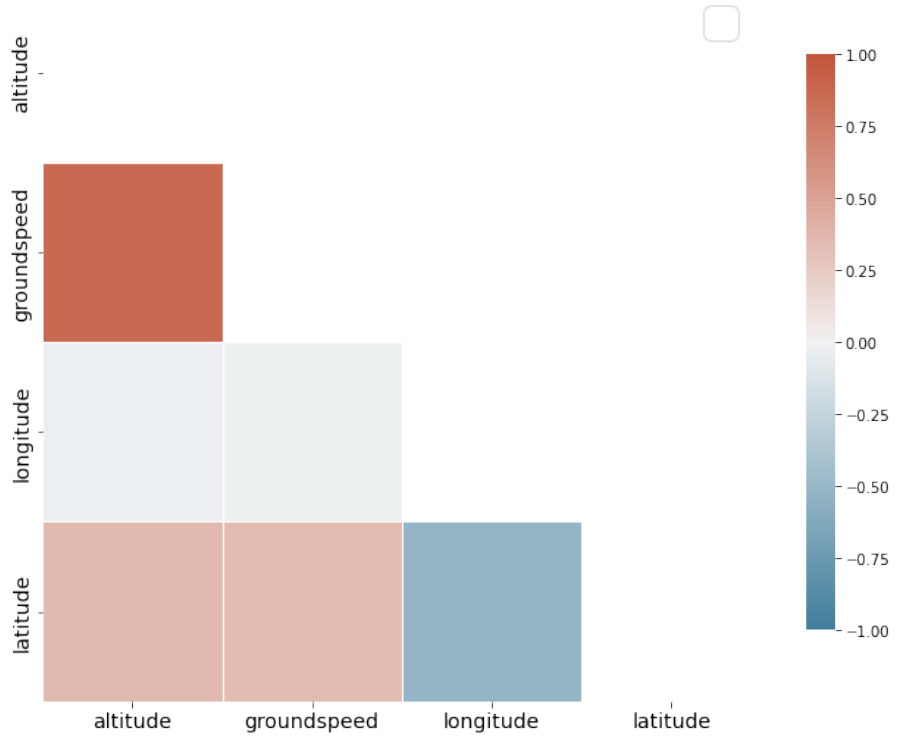
6.3.1 Esiprosessointi

Aluksi lentoaineistolle laskettiin tunnuslukuja, visualisoitiin numeeristen piirteiden jakaumia histogrammeina (kuvio 13), sekä laskettiin niiden väliset Pearsonin korrelaatiokertoimet (Kirch 2008) (kuvio 14). Tunnuslukuja tarkastelemalla selvitettiin onko aineistossa poikkeavia tai mahdollisesti virheellisiä datapisteitä. Esimerkiksi pituuskoordinaattien arvot tuli olla välillä $[-180, 180]$, leveyskoordinaattien välillä $[-90, 90]$ eikä nopeus voinut olla negatiivinen.

Puuttuvia arvoja oli piirteissä ”altitude” ja ”groundspeed”, 5 % ja 2 % kaikista arvoista, tässä järjestyksessä. Lennoille, joissa osa näistä piirteistä puuttui, korvattiin ne joko edellisellä tai seuraavalla arvolla. Jos kaikki arvot puuttuivat, asetettiin niiden arvoiksi 0.



Kuvio 13: Tilavektoriaineiston numeeristen piirteiden jakaumat



Kuvio 14: Tilavektoriaineiston numeeristen piirteiden korrelaatiomatriisi

6.3.2 Laskeminen

Käyttäen Python-kirjastoja Numpy ja Tracktable, lennoille laskettiin seuraavia piirteitä:

- **first_lat**: Leveyskoordinaattiarvo lennon alussa
- **first_lon**: Pituuskoordinaattiarvo lennon alussa
- **last_lat**: Leveyskoordinaattiarvo lennon lopussa
- **last_lon**: Pituuskoordinaattiarvo lennon alussa
- **points**: Tilavektorien lukumäärä yhteensä lennon aikana
- **timedelta**: Lennon kesto ajallisesti
- **hmax**: Maksimikorkeus lennon aikana
- **vmax**: Maksiminopeus lennon aikana
- **dg**$_n$: Etäisyysgeometria-arvot
- **cha**: Konveksin verhon ala
- **ch_ar**: Konveksin verhon akselien suhdeluku

6.4 Luokittelu

Luokitteluvaihe aloitettiin hakemalla aineisto tietokannasta. Kategorian ”Unknown” lennot jätettiin hakematta. Näin saatiin muodostettua luokitteluaineisto, kooltaan 20232 riviä ja 24 saraketta. Taulukossa 6 on viiden lennon otos luokitteluaineistosta.

first_lat	67.8766	-27.75	37.99	55.00	-37.76
first_lon	14.89	155.68	9.35	12.78	147.44
last_lat	67.86	-27.79	36.73	55.51	-38.10
last_lon	16.83	155.78	0.33	13.38	147.17
points	66	35	416	82	227
timedelta	650.0	340.0	4160.0	810.0	2260.0
hmax	11178.54	6286.50	10896.60	3246.12	4168.14
vmax	237.26	228.91	234.59	186.18	176.77
dg0	0.99	0.99	0.74	0.91	0.22
...	...	...	...	...	...
dg9	0.99	0.99	0.74	0.99	0.43
cha	0.80	0.88	24411.73	470.03	682.53
ch_ar	16484.79	37.80	23.07	8.81	3.35
category	Commercial	Military transport	Military transport	Cargo	Military training

Taulukko 6: Otos luokitteluaineistosta

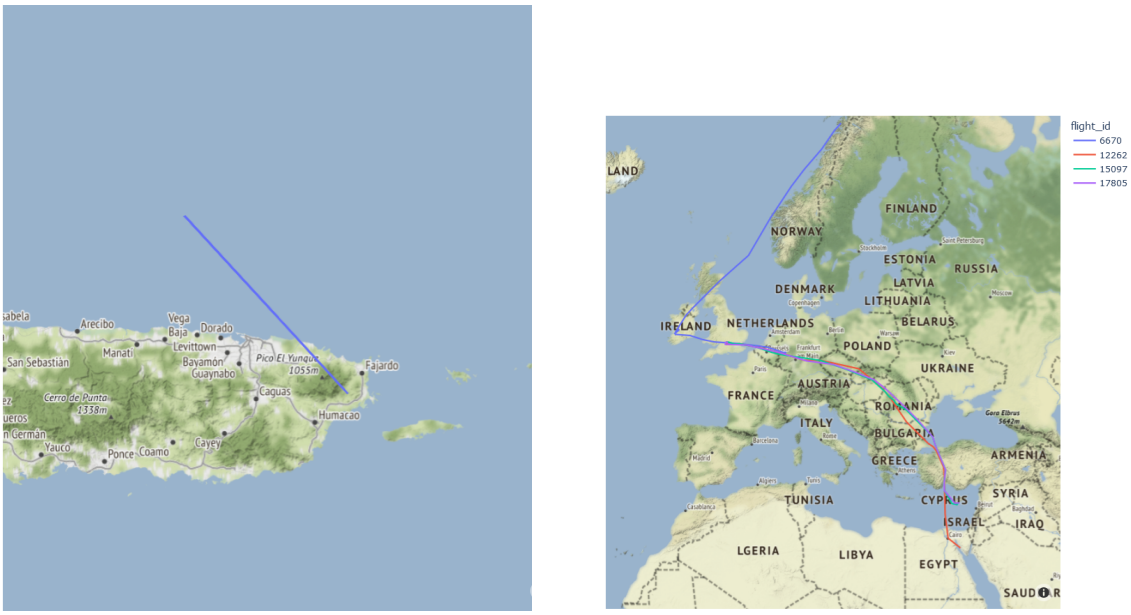
6.4.1 Esiprosessointi

Esiprosessointi aloitettiin jälleen aineistoon tutustumisella tunnuslukuja laskien ja jakaumia visualisoiden. Puuttuvia arvoja ei ollut, sillä ne oli käsitelty piirteitä laskettaessa.

Tunnusluvuista havaittiin piirteiden *cha* ja *ch_ar* kohdilla poikkeavia havaintoja: Maksimiarvot näille piirteille olivat 1 700 000 ja 1 300 000, kun 95% arvoista oli alle 51542 ja 529. Poikkeavia havaintoja ryhdyttiin tutkimaan niin, että havainnot joille näiden piirteiden arvot olivat yli 1 000 000, visualisoitiin karttapohjaan ja niiden muotoa tutkittiin. Piirteen *cha* suhteen näitä havaintoja oli 2 kappaletta, ja *ch_ar* suhteen 4 kpl. Havaittiin että poikkeamat piirteen *cha* suhteen olivat pitkiä lennetyltä matkaltaan ja joiden lentoradat sisälsivät

käännöksiä (kuvio 15, oikealla). Tämä johti suureen konveksin verhon alaan. Piirteen *ch_ar* suhteen poikkeavat havainnnot taas olivat täysin suoria lentoratoja (kuvio 15, vasemmalla). Näissä tapauksissa konveksin verhon lyhyimmän akselin pituus on hyvin pieni, ja siten piirteen arvo kasvaa hyvin suureksi.

Päädyttiin siihen ettei poikkeammissa ollut kyse virheellisistä arvoista, vaan suuret arvot olivat selitettävissä lentoradan muodoilla.



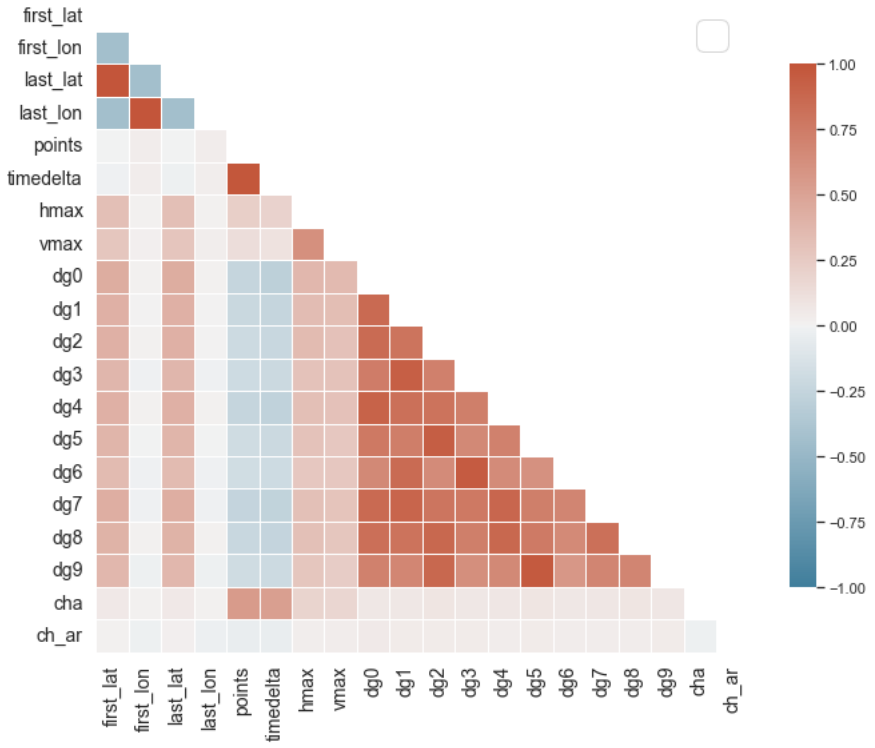
(a) Poikkeava lento piirteen *ch_ar* suhteen

(b) Poikkeavat lennot piirteen *cha* suhteen

Kuvio 15: Poikkeavia lentoja karttapohjaan visualisoituna

Seuraavaksi jaettiin aineisto *selittäviin muuttujiin* ja *vastemuuttujaan*. Vastemuuttujaksi asetettiin piirre *category* ja selittäviin muuttujiin kaikki muut piirteet. Tämän jälkeen visualisoitiin selittävien muuttujien riippuvuuksia korrelaatiomatriisilla. Pearsonin korrelaatiokertoimista (kuvio 16) huomattiin että etäisyysgeometria-arvojen sekä lennon pituuden välillä on heikko negatiivinen korrelaatio. Tämä tarkoittaa sitä että aineistossa kestoltaan lyhyet lennot ovat pitkiä lentoja enemmän yhteydessä suuriin etäisyysgeometria-arvoihin, jotka kertovat lentoradan suoruudesta.

Lopuksi aineisto jaettiin harjoitus- ja testiaineistoon suhteessa 4:1.



Kuvio 16: Luokitteluaineiston selittävien muuttujien korrelaatiomatriisi

6.4.2 Skaalaus

Luokittelumalleja varten selittävät muuttujat skaalattiin eri menetelmillä. Kuten luvussa 3.1.1 todettiin, parhaan menetelmän löytämiseksi ei ole olemassa mitään tiettyä keinoa, joten käytettyihin skaalausmetodeihin päädyttiin kokeilemalla eri vaihtoehtoja. KNN-mallia varten selittävät muuttujat skaalattiin z-arvo-normalisoinnilla, TabNet-neuroverkkomallille taas min-max-menetelmällä. Satunnaismetsä-mallille aineistoa ei skaalattu.

6.4.3 Pääkomponenttianalyysi

KNN-menetelmän selittävien muuttujien aineistolle suoritettiin pääkomponenttianalyysi ja komponenttien ominaisarvojen analysoinnin perusteella tämä aineisto muunnettiin kahdeksanulotteiseksi.

6.4.4 Ylinäytteistys

Aineisto ylinäytteistettiin TabNet:lle käyttäen satunnaista vähemmistöylinäytteistystä. Toteutuksena käytettiin Imbalanced-learn-kirjaston RandomOverSampler-luokkaa[6]. KNN:lle ja satunnaismetsälle alkuperäinen, kategoriajakaumaltaan epätasainen aineisto johti parempiin tuloksiin, joten niille aineistoa ei ylinäytteistetty.

6.4.5 Luokittelumallien sovitus

Kolme luokittelumallia sovitettiin edellä kuvattuun tapaan käsiteltyyn harjoitusaineistoon. KNN- ja satunnaismetsä-malleina käytettiin Python-kirjaston Scikit-learn toteutuksia[7,8] ja TabNet-mallina Python-kirjaston pytorch-tabnet toteutusta[9]. Mallien hyperparametreja optimoitiin ristiinvalidoinnilla (Refaeilzadeh, Tang ja Liu 2009), käyttäen kirjastojen Scikit-learn[10,11,12] sekä Optuna[13] työkaluja. Optimoidut hyperparametrit sekä ristiinvalidointitarkkuuksien keskiarvot näkyvät taulukossa 7.

6. https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html
7. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
8. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
9. https://github.com/dreamquark-ai/tabnet
10. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
11. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
12. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html
13. https://github.com/optuna/optuna

Luokittelumalli	Hyperparametrit	Tarkkuuden keskiarvo [%]
KNN	• Naapureiden lukumäärä: 8 • Metriikka: Minkowski	0.73
Satunnaismetsä	• Puiden lukumäärä: 600 • Otosten minimimäärä halkaisuun: 2 • Otosten minimimäärä lehtisolmuun: 2 • Maksimipiirteiden määrä: $\sqrt{\text{Piirteiden lukumäärä}}$ • Maksimisyvyys: 60 • Bootstrap-aggregointi: Ei	0.88
TabNet	• Maskin tyyppi : Sparsemax • Päätösennustekerroksen leveys: 64 • Askelten lukumäärä: 1 • Gamma: 1.4 • Jaettujen GLU-kerroksien määrä: 2 • Lambda: 0.0004 • Optimointialgoritmi: Adam • Oppimisnopeus: 0.02 • Virhefunktio: Ristientropia • Epokkien lukumäärä: 112	0.94

Taulukko 7: Luokittelumallien optimoidut hyperparametrit sekä tarkkuudet ristiinvalidoinnissa

6.4.6 Tulokset

Evaluointi testiaineistolla osoitti F-arvojen perusteella (taulukot 8, 9, 10) satunnaismetsän suoriutuneen malleista parhaiten. Sitä seurasi TabNet ja kolmantena KNN. Myös sekaannusmatriisien perusteella (kuviot 17, 18) kolmikon järjestys oli sama; satunnaismetsä oli KNN:ää parempi jokaisessa kategoriassa, kun taas vaikka TabNet oli satunnaismetsää tarkempi esimerkiksi ”Military tanker”- ja ”Research”-lentojen kohdilla, kokonaisuudessaan jälkimmäinen suoritui paremmin.

Sekaannusmatriiseista nähdään myös, että kaikki mallit oppivat tunnistamaan kategorioiden ”Civil surveillance”, ”Commercial”, ja ”Military training” lennot varsin hyvin. Tämän lisäksi satunnaismetsä ennusti kategorioiden ”Emergency”, ”General” sekä ”Military transport” melko tarkasti. TabNet taas pärjäsi satunnaismetsää paremmin kategorioiden ”Cargo” ja ”Research” kanssa. Kaikki kolme mallia ennustivat hyvin kolmen suuren kategorian (”Commercial”, ”Military training”, ”Military transport”) lennot.

Luokittelutulosten hajonta, eli kuinka moneen eri kategoriaan väärin luokitellut havainnot osuivat tietylle kategorialle, oli kaikilla kolmella mallilla suurinta kategorialle ’"Military transport”.

Kategoria	Sisäinen tarkkuus	Herkkyys	F-arvo	Lukumäärä
Business	0.61	0.35	0.45	93
Cargo	0.70	0.62	0.66	240
Civil surveillance	0.93	0.84	0.88	62
Commercial	0.83	0.94	0.88	973
Emergency	0.71	0.69	0.7	91
General	0.58	0.54	0.56	13
Military command and control	0.69	0.48	0.56	23
Military EW	0.67	0.29	0.40	21
Military fighter	0.40	0.22	0.29	9
Military surveillance	0.46	0.14	0.22	42
Military tanker	0.63	0.48	0.55	230
Military training	0.89	0.96	0.93	1024
Military transport	0.81	0.80	0.80	905
Research	0.86	0.53	0.66	36
Painotettu keskiarvo	0.81	0.82	0.81	3762

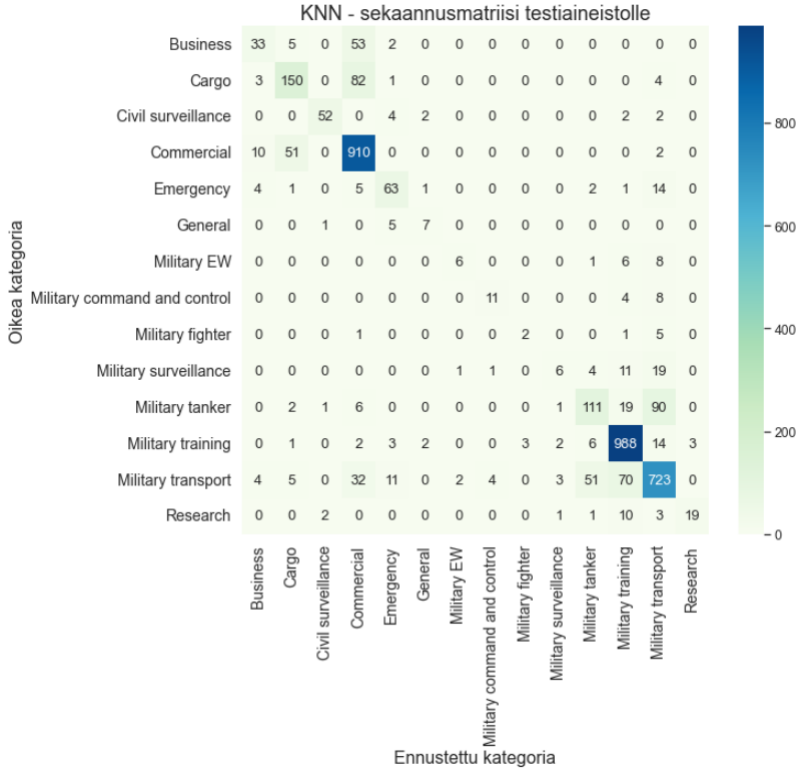
Taulukko 8: KNN-mallin luokittelutulokset testiaineistolle

Kategoria	Sisäinen tarkkuus	Herkkyys	F-arvo	Lukumäärä
Business	0.86	0.40	0.54	93
Cargo	0.85	0.72	0.78	240
Civil surveillance	1.00	0.95	0.98	62
Commercial	0.88	0.97	0.92	973
Emergency	0.91	0.80	0.85	91
General	1.00	0.92	0.96	13
Military command and control	0.81	0.74	0.77	23
Military EW	0.73	0.52	0.61	21
Military fighter	0.75	0.67	0.71	9
Military surveillance	0.61	0.26	0.37	42
Military tanker	0.85	0.60	0.71	230
Military training	0.97	0.98	0.97	1024
Military transport	0.85	0.94	0.89	905
Research	0.86	0.67	0.75	36
Painotettu keskiarvo	0.89	0.89	0.89	3762

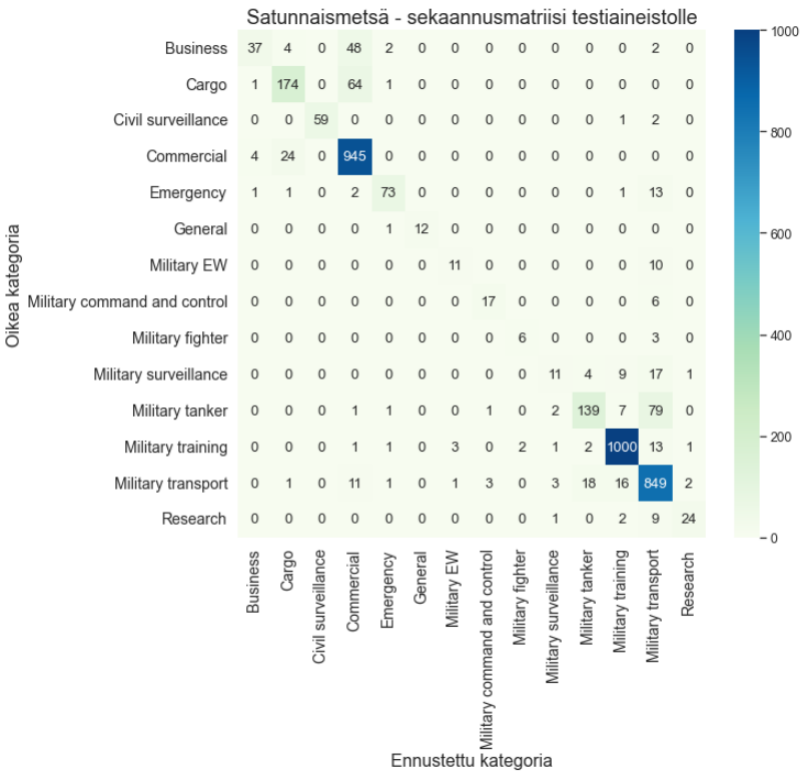
Taulukko 9: Satunnaismetsä-mallin luokittelutulokset testiaineistolle

Kategoria	Sisäinen tarkkuus	Herkkyys	F-arvo	Lukumäärä
Business	0.36	0.54	0.43	93
Cargo	0.61	0.85	0.71	240
Civil surveillance	0.93	0.85	0.89	62
Commercial	0.91	0.78	0.84	973
Emergency	0.72	0.80	0.76	91
General	0.85	0.85	0.85	13
Military command and control	0.62	0.70	0.65	23
Military EW	0.46	0.52	0.49	21
Military fighter	0.23	0.33	0.27	9
Military surveillance	0.31	0.40	0.35	42
Military tanker	0.57	0.65	0.61	230
Military training	0.93	0.94	0.94	1024
Military transport	0.86	0.78	0.82	905
Research	0.60	0.78	0.67	36
Painotettu keskiarvo	0.83	0.81	0.82	3762

Taulukko 10: TabNet-mallin luokittelutulokset testiaineistolle

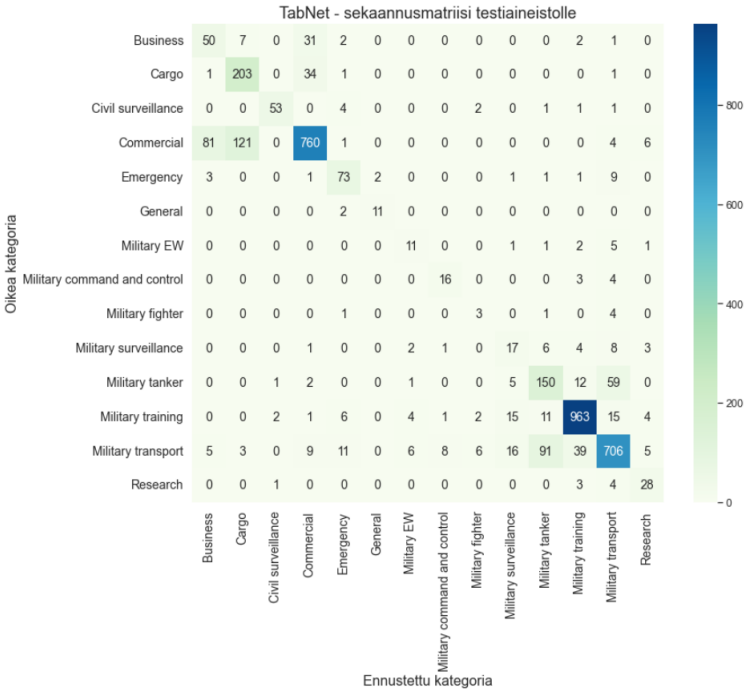


(a) KNN



(b) Satunnaismetsä

Kuvio 17: KNN- ja satunnaismetsä-mallien sekaannusmatriisit



Kuvio 18: TabNet-mallin sekaannusmatriisi

6.5 Selitettävyys

Luokittelumallien toimintaa selitettiin tärkeimpien piirteiden perusteella. Koska KNN-mallin aineisto oli käsitelty dimensioita pienentävällä pääkomponenttianalyysilla, alkuperäisten piirteiden tarkastelu ei enää ollut mahdollista. Pelkkien pääkomponenttien tärkeysjärjestys ei taas ole oleellista selittävyyden kannalta, mikäli halutaan tarkastella mitkä piirteet alkuperäisessä aineistossa vaikuttavat luokittelumallien toimintaan eniten. Siten KNN-malli jätettiin pois selitettävyysvaiheesta. Kahden muun mallin toiminnan selitettävyydeksi käytettiin kahta eri menetelmää.

6.5.1 Mallin arvot

Ensimmäinen menetelmä oli mallin sisällä laskettuja piirteiden tärkeysjärjestys. Satunnaismetsä laskee sisäisesti tärkeimmät piirteet Gini-kertoimen avulla, kun taas TabNet:n piirteiden tärkeysjärjestys saadaan tarkastelemalla huomioivien transformereiden arvoja. Piirteiden tärkeyttä kuvaavat normalisoidut painokertoimet ovat saatavilla molemmille malleille käytössä olleiden Scikit-learnin RandomForestClassifier- sekä Pytorchin TabNetClassifier-olioiden attribuuteista *feature_importances_*.

Satunnaismetsälle (kuvio 19) viisi tärkeintä piirrettä olivat maksiminopeus (*vmax*), loppupisteen koordinaatit (*last_lon*, *last_lat*) sekä alkupisteen koordinaatit (*first_lon*, *first_lat*). Yhdessä näiden viiden piirteen normalisoidut painokertoimet olivat noin 0.49. Vähiten oleellisia piirteitä oli etäisyysgeometriapiirteet dg_1, dg_6, dg_3, dg_9 ja dg_5.

Vastaavasti TabNet:lle (kuvio 19) viisi tärkeintä piirrettä olivat alkupisteen koordinaatit, maksiminopeus, loppupisteen pituuskoordinaattit, ja konveksin verhon ala (*cha*). Näiden piirteiden normalisoitujen painokertoimien summa oli noin 0.48. Vähiten tärkeimpiä piirteitä oli etäisyysgeometriapiirteet dg_9, dg_8, dg_5, dg_1 ja dg_6.

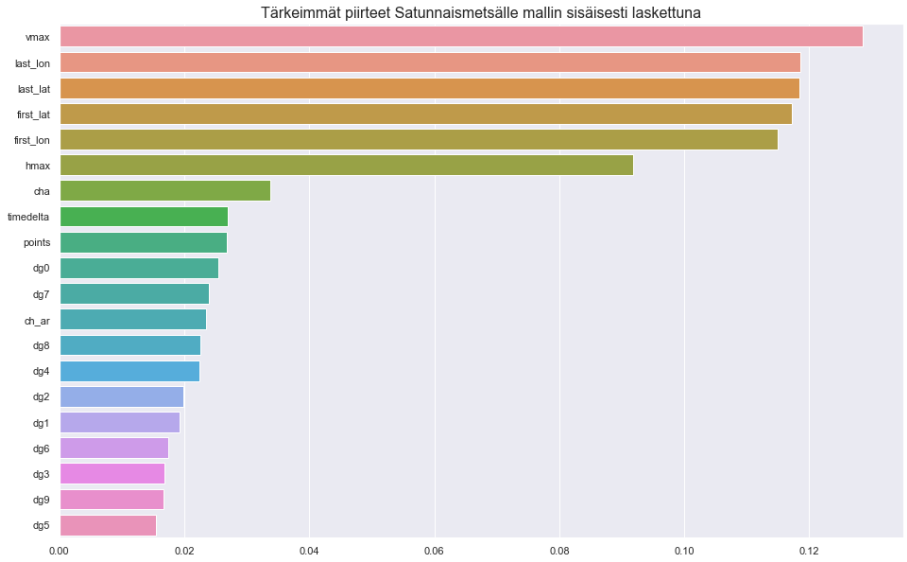
6.5.2 Shapley-arvot

Toinen menetelmä tärkeimpien piirteiden selvittämiseksi oli Shapley-arvojen laskeminen. Tähän käytettiin Python-kirjastoa shap[14]. Shapley-arvot laskettiin em. kirjaston avulla käyttäen satunnaismetsälle TreeExplainer-luokkaa[15] ja TabNet:lle KernelExplainer-luokkaa[16].

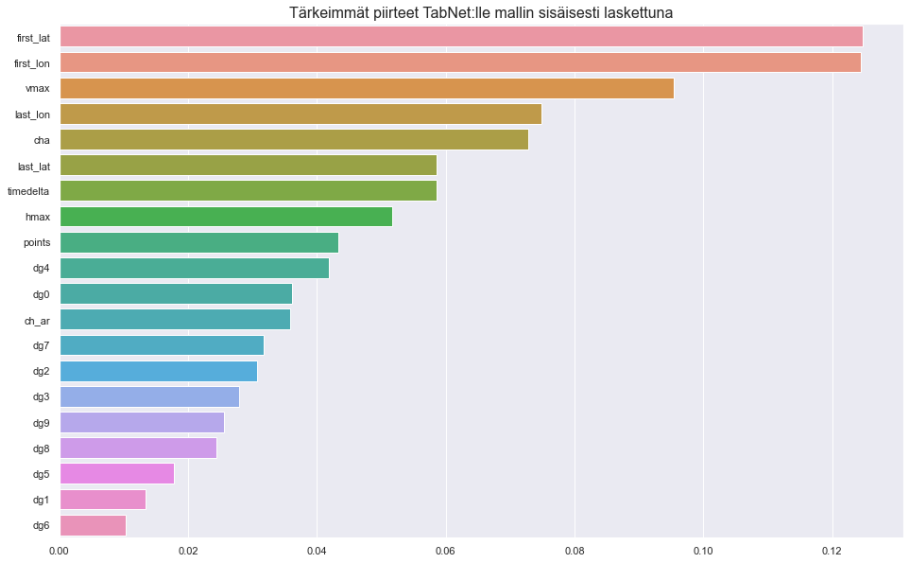
Viisi tärkeintä piirrettä satunnaismetsälle (kuvio 20) olivat Shapley-arvojen perusteella samat kuin mallin sisäisesti laskettuna, ainoastaan näiden keskenäisessä järjestyksessä oli eroja. Varsinkin kategorialle ”Commercial” olivat alku- ja loppupisteen leveyskoordinaatit oleellisia. Samoin kategorioille ”Military tanker”, ”Research” ja ”Civil surveillance” oli maksiminopeutta kuvaava piirre erityisen tärkeä.

TabNet:lle (kuvio 20) puolestaan tärkeimmät piirteet olivat konveksin verhon alaa lukuunottamatta samat kuin mallin sisäisesti lasketut. Sen tilalla oli loppupisteen leveyskoordinaatit. Alku- ja loppupisteen leveyskoordinaatit olivat huomattavan tärkeitä kategorialle ”Military transport”, kun taas kategorialle ”Commercial” nyt korostuivat alkupisteen sekä leveys- että pituuskoordinaatit. Samoin kuin satunnaismetsälle, Shapley-arvojen mukaan maksiminopeus oli TabNet:lle suhteellisen tärkeä kategorioissa ”Research” ja ”Civil surveillance”, sekä lisäksi kategorialle ”Emergency”. Piirre *hmax* oli kuudenneksi tärkein molemmille malleille.

14. https://github.com/slundberg/shap
15. https://shap-lrjball.readthedocs.io/en/latest/generated/shap.TreeExplainer.html
16. https://shap-lrjball.readthedocs.io/en/latest/generated/shap.KernelExplainer.html

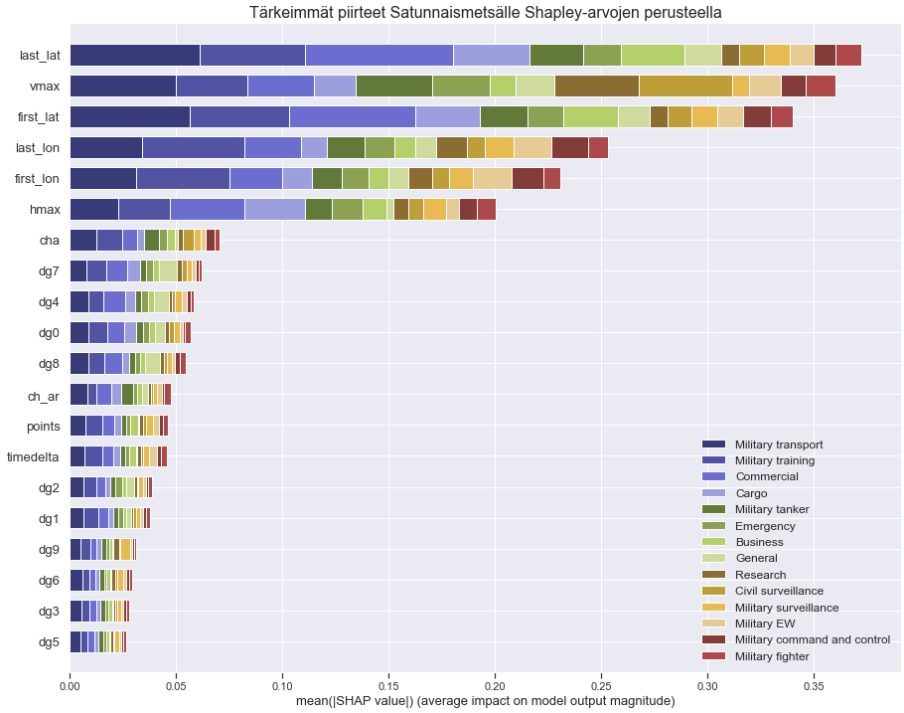


(a) Satunnaismetsä

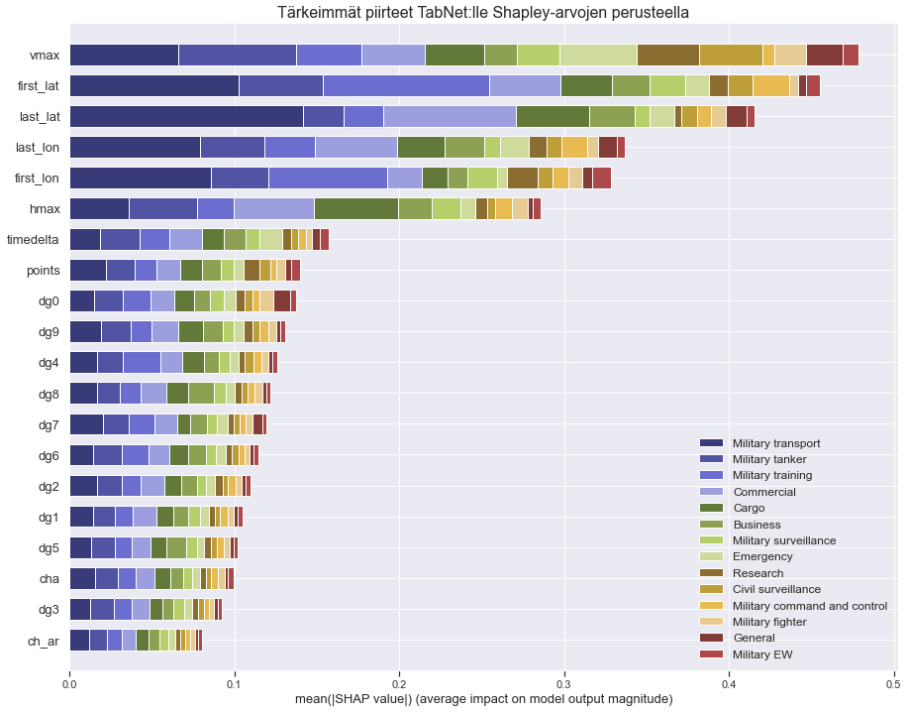


(b) TabNet

Kuvio 19: Piirteiden tärkeysjärjestys luokittelumalleille mallien sisäisesti laskettuna



(a) Satunnaismetsä



(b) TabNet

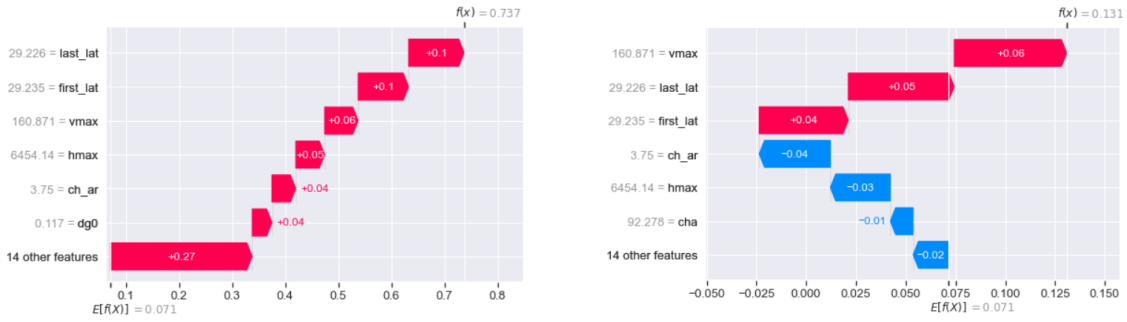
Kuvio 20: Piirteiden tärkeysjärjestys luokittelumalleille Shapley-arvojen perusteella

6.5.3 Yksittäisten havaintojen ennustusten analysointi

Shapley-arvojen avulla analysoitiin myös yksittäisten lentojen ennustuksia. Shap tarjoaa mahdollisuuden visualisoida tärkeimpiä piirteitä ja niiden vaikutusta ennustuksiin myös havaintokohtaisesti. Käydään seuraavaksi niitä läpi muutaman esimerkkitapauksen kautta. Kuvioissa 21, 22 ja 23 ennustukset ovat menneet pieleen, kun taas kuviossa 24 oikein.

Kuviosta 21 nähdään että leveyskoordinaattien ja maksiminopeuden arvoilla oli positiivinen vaikutus molempiin kategorioihin. Konveksin verhon asteluvulla (*ch_ar*) ja maksimikorkeudella (*hmax*) taas oli positiivinen vaikutus ennustetun kateogian, mutta negatiivinen todellisen kategorian suhteen. Toisessa esimerkkilennossa (kuvio 22) maksimikorkeus puski mallia ennustuksen suuntaan ja poispäin todellisesta kategoriasta. Leveyskoordinaateilla oli jälleen positiivinen vaikutus molempien kategorioiden suhteen.

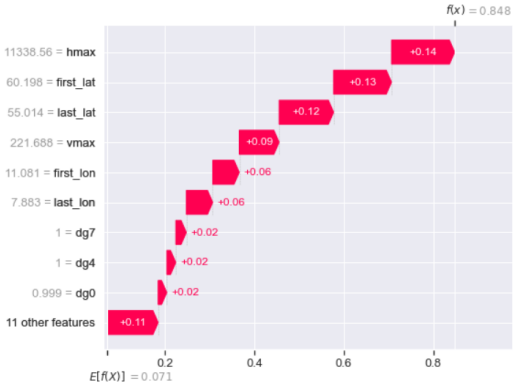
Kolmannessa esimerkissä (kuvio 23) nähdään miten sekä maksiminopeus että -korkeus painoivat ennustusta poispäin todellisesta kategoriasta. Myös lennon suoruutta kuvaavat etäisyysgeometria-arvot vaikuttivat negatiivisesti todellisen kategorian suhteen. Oikein ennustetun lennon (kuvio 24) Shapley-arvot kertovat että malli päätteli maksiminopeuden, etäisyysgeometria-arvo dg_2:n, ajallisen keston sekä alkupisteen pituuskoordinaattien perusteella lennon kuuluvan kategoriaan ”Military training”. Loppupisteen leveyskoordinaatit sekä toinen etäisyysgeometria-arvo dg_9 työnsivät ennustusta poispäin tästä kategoriasta.



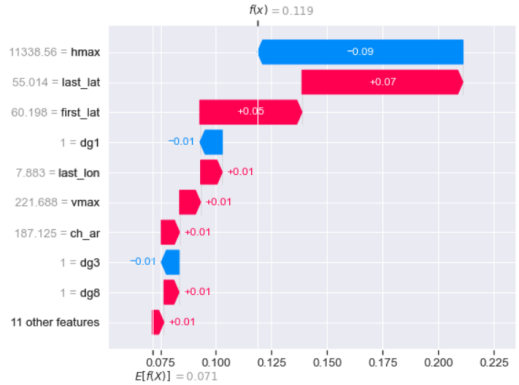
(a) Ennuste (Military transport)

(b) Oikea (Military tanker)

Kuvio 21: Esimerkki väärin ennustetusta lennosta satunnaismetsällä

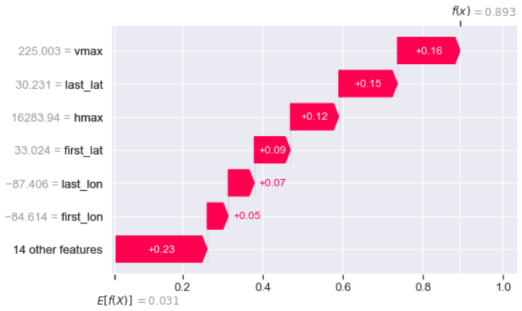


(a) Ennuste (Commercial)

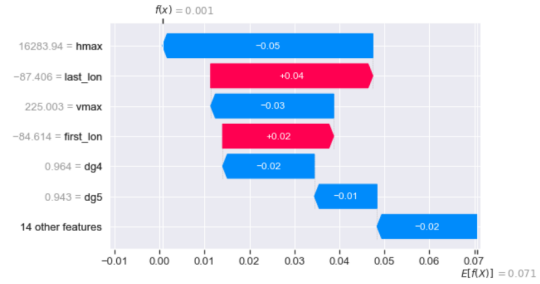


(b) Oikea (Cargo)

Kuvio 22: Esimerkki väärin ennustetusta lennosta satunnaismetsällä

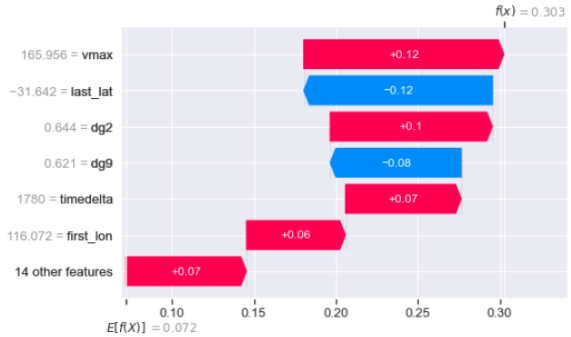


(a) Ennuste (Military transport)



(b) Oikea (Military training)

Kuvio 23: Esimerkki väärin ennustetusta lennosta TabNet:llä



(a) Ennuste ja oikea (Military training)

Kuvio 24: Esimerkki oikein ennustetusta lennosta TabNet:llä

7 Pohdinta

Tässä luvussa nostetaan muutamia huomioita aineistoon liittyvistä riskeistä, keskustellaan tutkimuksen aikaansaannoksista sekä verrataan tuloksia aikaisempiin tutkimuksiin. Luku on jaettu kolmeen osaan. Ensimmäisessä osassa puhutaan aineistosta, toisessa keskitytään luokitteluun, jonka jälkeen siirrytään tulkittavuuteen. Toisen ja kolmennen osan lopuissa vastataan tutkimuskysymyksiin.

7.1 Ainesto

Mitattuihin arvoihin liittyy aina tietty epävarmuus. Vaikka tilavektoridata esikäsiteltiin ja tarkastettiin virheellisten arvojen varalta (kappale 6.3.1), jokaista lentoa ei tarkastettu yksitellen, muutamia kymmeniä poikkeuksia lukuunottamatta. Niinpä aineistossa voi olla lentoja, joissa yksittäisten havaintopisteiden arvot ovat virheellisiä. Esimerkiksi nopeus voi hypätä yhtäkkiä 100 yksikköä korkeammaksi ja palata taas seuraavassa havaintopisteessä takaisin. Tällaiset arvot eivät erotu koko aineiston seasta virheellisinä, mutta aikasarjamuotoisessa lentoratadatassa ne olisi mahdollista havaita poikkeaviksi. Tällaiset arvot voivat vääristää lentoradoista laskettuja piirteitä ja siten haitata luokittelijoiden oppimista.

Myös joukkoistettuun metadataan ilma-aluksista sekä käsin leimattuihin pohjatotuuksiin sisältyy riski virheellisestä tiedosta. Leimaukset tukeutuivat täysin metadatan operaattoritietoihin, jotka on kerätty viranomaisten tietokannoista sekä ilmailuharrastajien toimesta, joten niissä voi olla virheitä. Kuten kappaleessa 6.2 todettiin, varsinkin niiden operaattoreiden tapauksissa, joiden konetyyppejä voidaan käyttää useaan eri tarkoitukseen, voi pohjatotuuksissa olla vääriä arvoja.

7.2 Luokittelu

Evaluoinnin perusteella voidaan sanoa kaikkien kolmen luokittelumallin onnistuneen melko hyvin lentojen luokittelussa. Mallien suorituskyvyissä oli kuitenkin eroja, ja tietyt kategoriat olivat kaikille malleille haastavimpia ennustaa. Kokonaisuudessaan parhaiten malleista suo-

riutui satunnaismetsä. Käydään seuraavaksi läpi muutamia havaintoja luokitteluprosessista ja vastataan ensimmäiseen tutkimuskysymykseen.

Hyperparametreja optimoitaessa harjoitusaineistolla parhaan ristiinvalidointitarkkuuden (0.94) saavutti TabNet. Samoin erotus (0.11) ristiinvalidointitarkkuuden ja testiaineiston painotetun F-arvon välillä oli kaikista suurin TabNet-mallilla. Tämän perusteella voidaan päätellä että TabNet ylisovittui harjoitusaineistoon huomattavasti. Koska myös ristiinvalidoinnissa tarkkuus lasketaan validoimalla ennustukset sillä osajoukolla aineistosta, jota ei ole käytetty mallin opetuksessa, näin suuri ero lukujen välillä herättää huomiota.

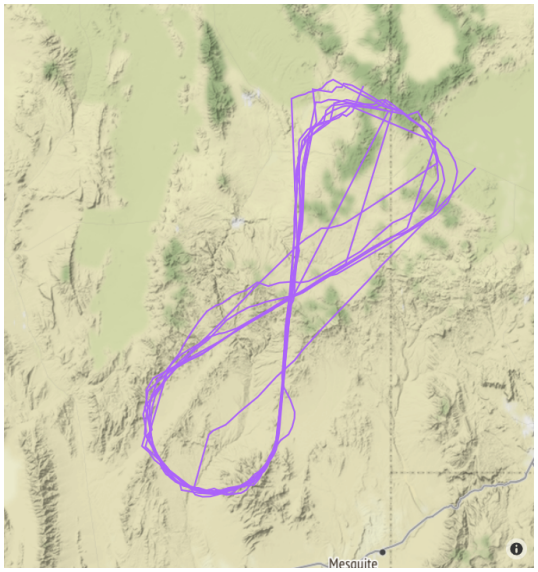
Ylisovittumisen voidaan päätellä johtuvan ainakin osittain siitä, että ylinäytteistyksen johdosta pienimpien kategorioiden havaintoja monistettiin niin monta kertaa, että harjoitusaineisto sisälsi duplikaattihavaintoja ristiinvalidoinnin harjoitus- ja validointiosioissa. TabNet oppi siis hyvin harjoitusaineiston piirteet, ja koska osa validointiosan havainnoista oli täysin samoja kuin harjoitusosassa, oli niiden ennustaminen mallille triviaali tehtävä. Koska validointiosan tarkoituksena on testata mallin ennustuskykyä uusilla havainnoilla, tällainen ilmiö aineistossa vääristää sen tarkkuutta. KNN:llä ja satunnaismetsällä ko. lukujen erotus oli toisaalta paljon pienempi (-0.06, 0.01), vaikka niiden ristiinvalidointiaineisto oli täysin sama. Nämä kaksi mallia eivät siis ylisovittuneet harjoitusaineistoon yhtä paljoa kuin TabNet.

Kaikki kolme mallia onnistuivat varsin hyvin kolmen lukumäärältään suurimpien kategorioiden (”Commercial”, ”Military training”, Military transport”) luokitteluissa. Näiden kolmen kategorian lentojen osuus koko aineistosta olivat 24,1 %, 25,3 % ja 22,4 %. Tämä herättää pohtimaan olisiko ennustusten osuvuudet olleet muillekin kategorioille tarkemmat, mikäli havaintoja olisi ollut enemmän. Esimerkiksi malleille haastavien ”Business”- ja ”Military EW”- tyyppisten lentojen osuus oli hyvin pieni; vain 2,3 % ja 0,5 %. Voi olla, että haastavat kategoriat olisivat edelleen olleet haastavia, vaikka ne olisivat olleet paremmin edustettuina harjoitusaineistossa. Näiden kategorioiden lennot saattavat olla vaikeammin tunnistettavissa esimerkiksi siitä syystä, että niiden lentoradat muistuttavat liikaa muiden kategorioiden lentoratoja tutkimuksessa käytettyjen piirteiden perusteella. Joka tapauksessa, kuten kappaleessa 3.1.3 todettiin, epätasaisen kategoriajakauman on havaittu vaikuttavan negatiivisesti luokittelumallien suorituskykyyn vähemmistökategorioiden osalta.

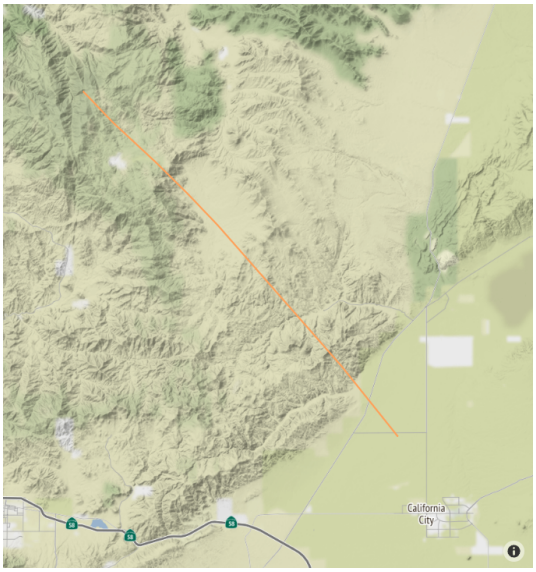
Toisaalta, vaikka lukumäärältään suurimpien kategorioiden luokittelu onnistui kaikilta luokittelijoilta mallikkaasti, suoraa yhteyttä havaintojen määrän ja ennustusten osuvuuden välille ei voida osoittaa. Pienimmissä kategorioissa oli huomattavaa vaihtelua F-arvoissa, esimerkiksi kategorian ”Research” (0.9 % havainnoista) F-arvot olivat 0.66, 0.75, 0.67 KNN:lle, satunnaismetsälle ja TabNet:lle, tässä järjestyksessä. Toisaalta lukumäärältään tätä vielä pienemmälle kategorialle ”General” (0.3 % havainnoista) vastaavat lukemat olivat 0.56, 0.96 ja 0.85. Vaikka havaintojen määrä ja tasainen kategoriajakauma auttaakin luokittelumallien oppimista (kappale 3.1.3), tämän tutkimuksen tulosten perusteella mallit voivat oppia luokittelemaan tarkasti myös sellaisien kategorioiden ilma-aluksia, joista on vain vähän havaintoja aineistossa.

Yllättävää oli kaikkien mallien huonot F-arvot (0.22, 0.37, 0.35) kategorialle ”Military surveillance”. Satunnaismetsälle tämä oli alhaisin lukema kaikista kategorioista. Nämä koneet kiertävät usein tiedusteltavan alueen yllä useita kertoja jotain tiettyä, kaartelevaa lentorataa (Strohmeier ym. 2021), joten niiden luokittelun ei olisi olettanut olevan näin haastavaa. Selitys saattaa piillä siinä, että käytössä ollut tilavektoriaineisto ei välttämättä sisällä kaikkia havaintopisteitä lennoista. Tilavektoridataan voi tallentua vain lennon alkupätkä, loppupätkä, tai jotain tältä väliltä. Tämä voi johtua esimerkiksi dataa keräävien ohjelmistoradioiden heikosta peitosta alueella (Strohmeier ym. 2021), tai sotilaallisten lentojen tapauksissa siitä, ettei koneiden ole aina pakko lähettää tietoa itsestään (Federal Aviation Administration 2019).

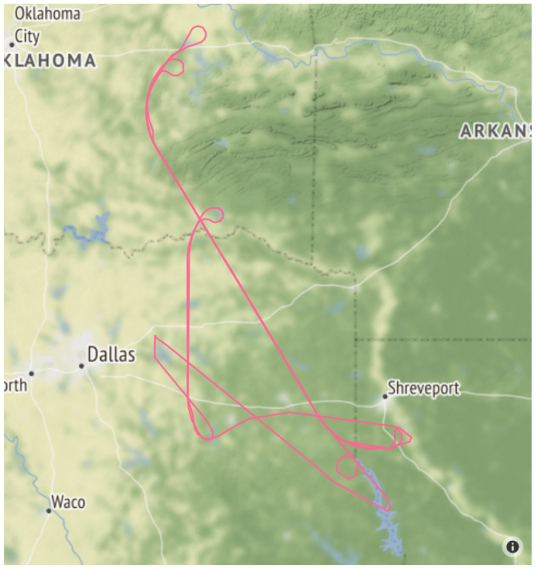
Tiedustelukoneiden (”Military surveillance”) tapauksessa voi esimerkiksi olla, että tilavektoriaineistoon tallentuu vain se osa lennosta, jossa kone on vasta siirtymässä alueelle, jolla se suorittaa tiedustelutehtäväänsä. Tällöin sen lentoradan perusteella on vaikea erottaa sitä mistä tahansa muusta suoraan lentävästä ilma-aluksesta. Sekaannusmatriisien perusteella luokittelumallit sekoittivatkin kategorian ”Military surveillance” koneita usein ”Military transport”-tyypin koneisiin. Kuviossa 25 on visualisoitu eri tiedustelukoneiden lentoratoja.



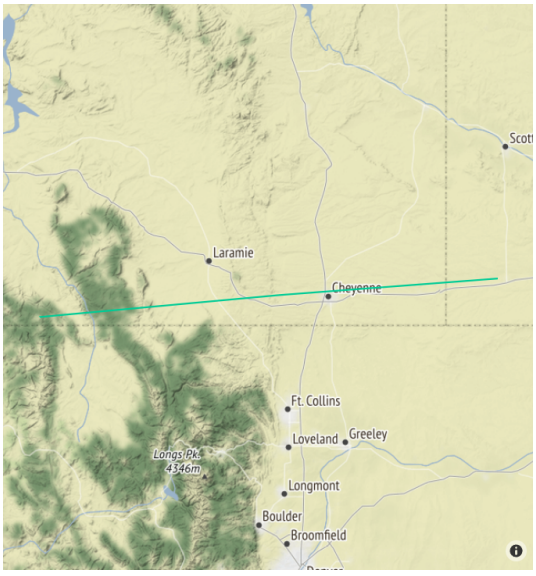
(a) Lentorata 1



(b) Lentorata 2



(c) Lentorata 3



(d) Lentorata 4

Kuvio 25: Esimerkkejä kategorian ”Military surveillance” lentoradoista karttatasoon piirrettynä

Lentoja muodostaessa yhden lennon minimipituudeksi asetettiin kymmenen havaintopistettä. Tätä lukua kasvattamalla olisi saatu eliminoitua osa tällaisista pätkälennoista pois. Toisaalta siirtymä voi olla hyvinkin pitkä matka, jolloin minimipituus olisi pitänyt asettaa hyvin korkeaksi, jos olisi haluttu varmistaa ettei aineistossa ole pelkkiä siirtymäosia. Silloin myös muiden kategorioiden kokonaisia lentoja olisi mahdollisesti jäänyt pois aineistosta, sillä lentojen pituus vaihtelee paljon. Osa kategorioista oli jo valmiiksi hyvin pieniä lukumääriltään, joten tässä tapauksessa aineistoa olisi pitänyt ladata enemmän. Tilavektoridatan lataaminen oli kuitenkin suhteellisen hidasta, jonka lisäksi olemassa oleva data vei jo niin paljon tilaa levyllä, ettei lisää dataa olisi juurikaan mahtunut. Tämän ongelman olisi toki voinut ratkaista lisätilaa järjestelemällä, mutta ajankäytön takia päätettiin olla lataamatta lisää dataa.

Myös artikkelin Strohmeier ym. (2021) tutkimuksessa satunnaismetsä suoritui KNN:ää paremmin, ollen neljästä mallista kaikista tarkin. Heidän käyttämänsä evaluointimetriikka oli kategorioiden keskimääräinen ulkoinen tarkkuus, jonka arvo satunnaismetsälle oli 0.87 ja KNN:lle 0.84. Vaikka tässä tutkimuksessa käytettiin evaluointimetriikkana epätasaisille luokitteluaineistoille paremmin sopivaa F-arvoa (kappale 3.5), vertailun vuoksi malleille laskettiin myös ulkoiset tarkkuudet. Ne olivat 0.89 satunnaismetsälle ja 0.82 KNN:lle. Tässä tutkimuksessa saavutettu paras ulkoinen tarkkuus oli siis hieman korkeampi kuin heillä, vaikka kategorioiden määrä oli heillä pienempi (8) kuin tässä tutkimuksessa (14).

Kategorian ”Military transport” hajonta oli myös Strohmeierin ym. tutkimuksessa suurin. He veikkasivat sen johtuvan ko. kategorian havaintojen pienestä määrästä aineistossa, kuljetuskoneiden käyttämisestä moniin eri tarkoituksiin sekä toisten kategorioiden havaintojen lentoratojen samankaltaisuudesta. Tämän tutkimuksen aineistossa ko. kategoria oli hyvin edustettuna, mutta kahden muun syyn voidaan ajatella selittävän hajontaa. Intuition pohjalta voidaan päätellä siviilitavarankuljetuskoneiden (”Cargo”) ja sotilaskuljetuskoneiden (”Military transport”) lentoratojen muistuttavan toisiaan niissä tapauksissa, joissa koneiden tehtävänä on siirtää kuormaa paikasta A paikkaan B mahdollisimman lyhintä reittiä pitkin. Lisäksi kuljetuskoneita käytetään useassa erityyppisessä roolissa, kuten kappaleessa 6.2 mainittiin.

TabNet taas suoritui sen kehittäjien tutkimuksessa luokittelutehtävistä paremmin kuin useat päätöspuu-pohjaiset mallit (Arik ja Pfister 2020). Tässä tutkimuksessa malli ylisovittui harjoitusaineistoon vahvemmin kuin muut mallit, jääden toiselle sijalle. TabNet:n kehittäjät pe-

rustelevat, että syväoppivan mallin etuna aikaisempiin menetelmiin verrattuna on sen parempi suorituskyky suurilla luokitteluaineistoilla. Artikkelin luokittelutehtävien aineistojen[1,2] koot, 580 000 ja 1 000 000 havaintoa, ovat huomattavasti suuremmat kuin tämän tutkimuksen (20 228). Voi olla että TabNet olisi suoriutunut paremmin suhteessa satunnaismetsään suuremmalla luokitteluaineistolla.

Ensimmäiseen tutkimuskysymykseen ”Miten lentoliikennettä voidaan luokitella koneoppimismalleilla pelkästään lentoratojen perusteella?” voidaan tutkimusten tulosten perusteella vastata seuraavasti: Käyttäen joukkoistettua tilavektori- sekä metadataa ilma-aluksista on mahdollista muodostaa lentoja, joille lasketaan niiden lentoradoista johdettuja piirteitä. Lentoja voidaan luokitella näiden piirteiden perusteella koneoppimismalleilla kategorioihin, jotka kertovat lentojen ensisijaiset käyttötarkoitukset. Luokittelun osuvuus riippuu käytettävästä mallista sekä lennon käyttötarkoituksesta. Parhaalla mallilla (satunnaismetsä) lentojen keskimääräistä luokittelutarkkuutta kuvaava lukema oli 0.89 ja huonoimmalla (k:n lähimmän naapurin menetelmä) 0.81.

7.3 Ennustusten tulkittavuus

Satunnaismetsän ja TabNet:n ennustuksia tulkittiin tärkeimpien piirteiden avulla. Sekä mallien sisäisesti tarkasteltuna että Shapley-arvojen perusteella oleellisimmat piirteet vaikuttivat liittyvän sijaintitietoihin (*first_lat*, *first_lon*, *last_lat*, *last_lon*) sekä maksiminopeuteen *vmax*.

Sijaintitietojen tärkeydestä voi päätellä, että samaan tarkoitukseen käytettävät koneet lentävät samoja reittejä. Maksiminopeuden voidaan taas ajatella näkyvän eri tarkoituksessa lennettävien koneiden lennossa niin, että esimerkiksi harjoituskoneita lennettäessä maksiminopeus pysyy alhaisempana kuin matkustajalentokoneen kohdalla, sillä edellisessä on tarkoitus harjoitella koneen hallintaa, kun taas jälkimmäisessä lentää kaupungista toiseen suhteellisen nopeasti.

Myös konetyypin rajoitteet nopeuden suhteen johtavat siihen, ettei kaikkien kategorioiden koneilla edes pääse yhtä korkeisiin nopeuksiin kuin toisilla. Esimerkiksi klassisen harjoi-

1. https://archive.ics.uci.edu/ml/datasets/covertype
2. https://archive.ics.uci.edu/ml/datasets/Poker+Hand

tuskoneen Cessna T-41 A:n[3] maksiminopeus on noin 224 km/h, kun taas matkustajakone Boeing 767:n[4] yli 900 km/h.

Viisi tärkeintä piirrettä olivat pääsääntöisesti samat mallin sisäisten- ja Shapley-arvojen perusteilla. Muiden piirteiden järjestys vaihteli mallin ja laskutavan myötä. Artikkelin Arik ja Pfister (2020) tutkimuksessa kaikista tärkein piirre oli sama TabNet:n sisäisten ja Shapley-arvojen perusteella, mutta muiden piirteiden järjestys oli eri. Satunnaismetsälle taas Shapley-arvojen kautta saadut tärkeysjärjestykset havaittiin artikkelin Orlenko ja Moore (2021) tutkimuksessa olevan lähempänä oikeaa kuin mallin sisäisten arvojen perusteella lasketut. Oikean tärkeysjärjestyksen he kertovat saaneensa herkkyysanalyysin kautta, jota tässä tutkimuksessa ei suoritettu.

Voidaan myös sanoa, että etäisyysgeometriatunnisteen arvot (dg_n) olivat vähiten tärkeitä malleille. Lentoradan muotoa mittaavista piirteistä ainoastaan konveksin verhon alan (*cha*) voidaan sanoa olleen tärkeä molemmille malleille sisäisten arvojen perusteella. Piirre oli seitsemänneksi tärkein satunnaismetsälle ja viidenneksi TabNet:lle. Kuten edellisessä kappaleessa todettiin, osa luokitteluaineiston lennoista koostuu vain osasta todellisesta lennosta. Voi olla, että lentoradan muotoa kuvaavat piirteet olisivat osoittautuneet tärkeämmiksi, jos kaikki luokitteluaineiston lennot olisivat olleet kokonaisia, jolloin mallit olisivat pystyneet paremmin tunnistamaan eri kategorioiden lentoratojen eroavaisuudet etäisyysgeometriapiirteitä käyttäen.

Toiseen tutkimuskysymykseen ”Miten luokittelevien koneoppimismallien tuloksia voidaan selittää?” voidaan vastata näin: Tuloksia voidaan selittää tulkitsemalla koneoppimismalleja. Yksi tapa tulkita on selvittää, mitkä piirteet luokitteluaineistossa olivat tärkeimpiä malleille niiden toiminnan kannalta. Piirteiden tärkeysjärjestyksen kautta on mahdollista ymmärtää miksi mallit päätyivät tiettyihin tuloksiin. Tulkitsemisessa voidaan käyttää joko mallien sisäisiä piirteiden tärkeydestä kertovia arvoja tai Shapley-arvoja. Piirteiden tärkeyttä voidaan tarkastella joko yksittäisen ennustuksen kohdalla tai kaikille ennustuksille yhteisesti.

3. https://www.nationalmuseum.af.mil/Visit/Museum-Exhibits/Fact-Sheets/Display/Article/198032/cessna-t-41a-mescalero/

4. https://www.united.com/ual/en/us/fly/travel/inflight/united-airlines-fleet.html

8 Yhteenveto

Tässä pro gradussa tutkittiin, miten lentoliikennettä voidaan luokitella koneoppimismenetelmillä pelkkien lentoratojen perusteella. Lisäksi tutkimuksen kohteena oli selvittää, miten luokittelijoiden tuloksia voidaan selittää. Kolme erityyppistä koneoppimismallia opetettiin luokittelemaan ilma-aluksia kategorioihin, jotka kertovat niiden ensisijaiset käyttötarkoitukset. Mallien luokittelutuloksia selitettiin niille tärkeimpien piirteiden avulla.

Tulosten pohjalta voidaan sanoa, että ilma-aluksia voidaan luokitella koneoppimismalleilla lentoradoista johdettujen piirteiden perusteella. Luokittelun tarkkuus riippuu käytetystä mallista ja ilma-aluksen käyttötarkoituksesta, mutta havaintojen määrään suhteutettuna kaikki mallit onnistuivat ennustamaan lentojen kategoriat melko tarkasti.

Tutkimuksessa käytetyillä selitettävän tekoälyn menetelmillä saatiin hyviä tuloksia. Havaittiin, että tuloksia voidaan selittää tärkeimpien piirteiden kautta joko hyödyntäen mallien sisäisiä arvoja, tai vaihtoehtoisesti käyttäen Shapley-arvoja. Piirteiden tärkeyttä voidaan tarkastella joko koko mallille yleisesti tai havaintokohtaisesti. Tärkeimmissä piirteissä tutkituille malleille korostuivat aluksen lentoradan alku- ja loppupisteen sijannit sekä maksiminopeus.

Tutkimus tuo tulosten selittämisen kautta merkittävän lisän aikaisempaan lentoliikenteen luokitteluun keskittyvään tutkimukseen. Tärkeimpien piirteiden avulla voidaan ymmärtää luokittelumallien mekanismeja ja analysoida ennustusten perusteita. Tutkimuksessa käytetyt menetelmät mahdollistavat koneoppivien mallien sisään- ja ulostulodatan välisten syy-seuraussuhteiden tarkastelun. Selitettävä tekoäly on suhteellisen uusi tutkimusala, ja se tarjoaa tulevaisuudessa uusia mahdollisuuksia datan, mallien sekä käsiteltävän ongelman kokonaisvaltaisempaan ymmärtämiseen sekä uusia sovellusmahdollisuuksia erilaisten päätöksentekojärjestelmien kehitystyöhön.

Tutkimuksen heikkouksia ovat käytetyn harjoitusaineiston epätasainen kategoriajakauma sekä aineiston laatuun liittyvät epävarmuudet. Vaikka lukumäärältään pienimpien kategorioiden kokoa kasvatettiin harjoitusaineistossa ylinäytteistämällä, johti alkuperäisten havaintojen vähäinen määrä näissä kategoriossa luultavasti mallien heikentyneisiin suorituskykyihin.

Aineiston laatuun liittyvät kysymykset koskevat varsinkin pohjatotuuksia. Niissä voi olla virheellisiä arvoja, sillä osa konetyypeistä on sellaisia, joita lentoja operoivat tahot voivat käyttää useampaan eri käyttötarkoitukseen.

Tutkimuksessa käytettyä lentoliikennedataa ei ole ollut avoimesti saatavilla kovin kauaa, joten siihen pohjaavien tutkimusten määrä on vähäinen. Tämä vaikeuttaa tulosten vertailua muihin tutkimuksiin, mutta toisaalta luo mahdollisuuden toimia pohjana lisätutkimukselle. Lopuksi esitetään muutamia mielenkiintoisia jatkotutkimuskohteita.

- Miten luokittelutulokset muuttuvat, jos lentojen minimipituutta havaintopisteinä nostetaan suuremmaksi?
- Miten luokittelumallien tulkittavuutta voidaan käyttää lisäpiirteiden jalostuksessa?
- Miten luokittelumallien tulkittavuutta voidaan käyttää mallien kehitystyössä?

Lähteet

Adadi, Amina, ja Mohammed Berrada. 2018. “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)”. *IEEE Access* 6:52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052.

Aggarwal, Charu C, ym. 2015. *Data mining: the textbook.* Nide 1. Springer.

Ahsan, Md Manjurul, M. A. Parvez Mahmud, Pritom Kumar Saha, Kishor Datta Gupta ja Zahed Siddique. 2021. “Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance”. *Technologies* 9 (3). ISSN: 2227-7080. https://doi.org/10.3390/technologies9030052. https://www.mdpi.com/2227-7080/9/3/52.

Akhil, Jabbar, Bulusu Deekshatulu ja Priti Chandra. 2013. “Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm”. *Procedia Technology* 10 (joulukuu): 85–94. https://doi.org/10.1016/j.protcy.2013.12.340.

Alpaydin, Ethem. 2020. *Introduction to machine learning.* MIT press.

Arik, Sercan O., ja Tomas Pfister. 2020. *TabNet: Attentive Interpretable Tabular Learning.* arXiv: 1908.07442 [cs.LG].

Barredo Arrieta, Alejandro, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia ym. 2020. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”. *Information Fusion* 58:82–115. ISSN: 1566-2535. https://doi.org/https://doi.org/10.1016/j.inffus.2019.12.012. https://www.sciencedirect.com/science/article/pii/S1566253519308103.

Bellman, Richard E. 1961. *Adaptive Control Processes: A Guided Tour.* Princeton University Press. ISBN: 9781400874668. https://doi.org/doi:10.1515/9781400874668. https://doi.org/10.1515/9781400874668.

Benoit, Kenneth. 2011. “Linear regression models with logarithmic transformations”. *London School of Economics, London* 22 (1): 23–36.

Bhatia, Nitin, ja Vandana. 2010. *Survey of Nearest Neighbor Techniques.* arXiv: 1007.0085 [cs.CV].

Boeing. 2022a. “Importance of freighters in the air cargo industry”. Viitattu 26. tammikuuta 2022. https://www.boeing.com/commercial/market/cargo-forecast/importance-of-freighters/.

———. 2022b. “KC-46A Pegasus”. Viitattu 26. tammikuuta 2022. https://www.boeing.com/defense/kc-46a-pegasus-tanker/.

Breiman, Leo. 2001. “Random forests”. *Machine learning* 45 (1): 5–32.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan ym. 2020. *Language Models are Few-Shot Learners.* https://doi.org/10.48550/ARXIV.2005.14165. https://arxiv.org/abs/2005.14165.

Buda, Mateusz, Atsuto Maki ja Maciej A. Mazurowski. 2018. “A systematic study of the class imbalance problem in convolutional neural networks”. *Neural Networks* 106 (lokakuu): 249–259. ISSN: 0893-6080. https://doi.org/10.1016/j.neunet.2018.07.011. http://dx.doi.org/10.1016/j.neunet.2018.07.011.

Cao, Chang, Davide Chicco ja Michael M. Hoffman. 2020. *The MCC-F1 curve: a performance evaluation technique for binary classification.* arXiv: 2006.11278 [stat.ML].

Carvalho, Diogo V., Eduardo M. Pereira ja Jaime S. Cardoso. 2019. “Machine Learning Interpretability: A Survey on Methods and Metrics”. *Electronics* 8 (8). ISSN: 2079-9292. https://doi.org/10.3390/electronics8080832. https://www.mdpi.com/2079-9292/8/8/832.

Castelvecchi, Davide. 2016. “Can we open the black box of AI?” *Nature* 538 (lokakuu): 20–23. https://doi.org/10.1038/538020a.

Chawla, N. 2005. “Data Mining for Imbalanced Datasets: An Overview”. Teoksessa *The Data Mining and Knowledge Discovery Handbook.*

Chawla, N. V., K. W. Bowyer, L. O. Hall ja W. P. Kegelmeyer. 2002. “SMOTE: Synthetic Minority Over-sampling Technique”. *Journal of Artificial Intelligence Research* 16 (kesäkuu): 321–357. ISSN: 1076-9757. https://doi.org/10.1613/jair.953. http://dx.doi.org/10.1613/jair.953.

Chawla, Nitesh V., Nathalie Japkowicz ja Aleksander Kotcz. 2004. “Editorial: Special Issue on Learning from Imbalanced Data Sets”. *SIGKDD Explor. Newsl.* (New York, NY, USA) 6, numero 1 (kesäkuu): 1–6. ISSN: 1931-0145. https://doi.org/10.1145/1007730.1007733. https://doi.org/10.1145/1007730.1007733.

Chicco, Davide, ja Giuseppe Jurman. 2020. “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”. *BMC Genomics* 21 (tammikuu). https://doi.org/10.1186/s12864-019-6413-7.

Chinchor, Nancy. 1992. “MUC-4 Evaluation Metrics”. Teoksessa *Proceedings of the 4th Conference on Message Understanding,* 22–29. MUC4 ’92. McLean, Virginia: Association for Computational Linguistics. ISBN: 1558602739. https://doi.org/10.3115/1072064.1072067. https://doi.org/10.3115/1072064.1072067.

Cover, T., ja P. Hart. 1967. “Nearest neighbor pattern classification”. *IEEE Transactions on Information Theory* 13 (1): 21–27. https://doi.org/10.1109/TIT.1967.1053964.

Cunningham, Pádraig, ja Sarah Jane Delany. 2021. “k-Nearest Neighbour Classifiers - A Tutorial”. *ACM Computing Surveys* 54, numero 6 (heinäkuu): 1–25. ISSN: 1557-7341. https://doi.org/10.1145/3459665. http://dx.doi.org/10.1145/3459665.

Dauphin, Yann N., Angela Fan, Michael Auli ja David Grangier. 2016. *Language Modeling with Gated Convolutional Networks.* https://doi.org/10.48550/ARXIV.1612.08083. https://arxiv.org/abs/1612.08083.

Dietterich, Thomas G. 2000. “Ensemble Methods in Machine Learning”. Teoksessa *MULTIPLE CLASSIFIER SYSTEMS, LBCS-1857,* 1–15. Springer.

Doshi-Velez, Finale, ja Been Kim. 2017a. *Towards A Rigorous Science of Interpretable Machine Learning.* https://doi.org/10.48550/ARXIV.1702.08608. https://arxiv.org/abs/1702.08608.

———. 2017b. “Towards A Rigorous Science of Interpretable Machine Learning”. *arXiv: Machine Learning.*

Federal Aviation Administration. 2019. *Revision to Automatic Dependent Surveillance-Broadcast (ADS–B) Out Equipment and Use Requirements.* misc. https://www.govinfo.gov/content/pkg/FR-2019-07-18/pdf/2019-15248.pdf.

Garca, Salvador, Julin Luengo ja Francisco Herrera. 2014. *Data Preprocessing in Data Mining*. Springer Publishing Company, Incorporated. ISBN: 331910246X.

Gingrass, Colton, Dashi I. Singham ja Michael P. Atkinson. 2021. "Shape Analysis of Flight Trajectories Using Neural Networks". *Journal of Aerospace Information Systems* 18 (11): 762–773. https://doi.org/10.2514/1.I010923.

Goodfellow, Ian, Yoshua Bengio ja Aaron Courville. 2016. *Deep Learning*. Http://www.deeplearningbook.org. MIT Press.

Gorade, Sudhir M, Ankit Deo ja Preetesh Purohit. 2017. "A study of some data mining classification techniques". *International Research Journal of Engineering and Technology* 4 (4): 3112–3115.

Han, Jiawei, Micheline Kamber ja Jian Pei. 2012. "3 - Data Preprocessing". Teoksessa *Data Mining (Third Edition),* Third Edition, toimittanut Jiawei Han, Micheline Kamber ja Jian Pei, 83–124. The Morgan Kaufmann Series in Data Management Systems. Boston: Morgan Kaufmann. ISBN: 978-0-12-381479-1. https://doi.org/https://doi.org/10.1016/B978-0-12-381479-1.00003-4. https://www.sciencedirect.com/science/article/pii/B9780123814791000034.

Hand, D. J. (David J.), Heikki Mannila ja Padhraic Smyth. 2001. *Principles of data mining / David Hand, Heikki Mannila, Padhraic Smyth.* [kielellä eng]. Adaptive computation and machine learning. Cambridge, Mass.: MIT Press. ISBN: 026208290X.

Hand, David J., Peter Christen ja Nishadi Kirielle. 2021. *F*: An Interpretable Transformation of the F-measure.* arXiv: 2008.00103 [cs.LG].

Hastie, Trevor, Robert Tibshirani ja Jerome Friedman. 2001. *The Elements of Statistical Learning.* Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Ho, Yaoshiang, ja Samuel Wookey. 2019. "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling". *IEEE Access* 8:4806–4813.

Hoffer, Elad, Itay Hubara ja Daniel Soudry. 2017. “Train longer, generalize better: closing the generalization gap in large batch training of neural networks”, https://doi.org/10.48550/ARXIV.1705.08741. https://arxiv.org/abs/1705.08741.

Japkowicz, Nathalie. 2000. “Learning from Imbalanced Data Sets: A Comparison of Various Strategies”, 10–15. AAAI Press.

Japkowicz, Nathalie, ja Shaju Stephen. 2002. “The class imbalance problem: A systematic study”. *Intelligent Data Analysis,* 429–449.

Jolliffe, Ian. 2011. “Principal Component Analysis”. Teoksessa *International Encyclopedia of Statistical Science,* toimittanut Miodrag Lovric, 1094–1096. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-04898-2. https://doi.org/10.1007/978-3-642-04898-2_455. https://doi.org/10.1007/978-3-642-04898-2_455.

Kirch, Wilhelm, toimittanut. 2008. “Pearson’s Correlation Coefficient”. Teoksessa *Encyclopedia of Public Health,* 1090–1091. Dordrecht: Springer Netherlands. ISBN: 978-1-4020-5614-7. https://doi.org/10.1007/978-1-4020-5614-7_2569. https://doi.org/10.1007/978-1-4020-5614-7_2569.

Kotsiantis, Sotiris, Dimitris Kanellopoulos, Panayiotis Pintelas ym. 2006. “Handling imbalanced datasets: A review”. *GESTS international transactions on computer science and engineering* 30 (1): 25–36.

Krüger, Frank. 2016. “Activity, Context, and Plan Recognition with Computational Causal Behaviour Models”. Tohtorinväitöskirja, joulukuu.

Kuhn, M., ja K. Johnson. 2019. *Feature Engineering and Selection: A Practical Approach for Predictive Models.* Chapman & Hall/CRC Data Science Series. CRC Press. ISBN: 9781351609470. https://books.google.fi/books?id=xy73DwAAQBAJ.

Kulkarni, Ajay, Deri Chong ja Feras A. Batarseh. 2020. “5 - Foundations of data imbalance and solutions for a data democracy”. Teoksessa *Data Democracy,* toimittanut Feras A. Batarseh ja Ruixin Yang, 83–106. Academic Press. ISBN: 978-0-12-818366-3. https://doi.org/https://doi.org/10.1016/B978-0-12-818366-3.00005-8. https://www.sciencedirect.com/science/article/pii/B9780128183663000058.

Kumar, Satvik G., Samantha J. Corrado, Tejas G. Puranik ja Dimitri N. Mavris. 2021. “Classification and Analysis of Go-Arounds in Commercial Aviation Using ADS-B Data”. *Aerospace* 8, numero 10 (lokakuu). https://www.mdpi.com/2226-4310/8/10/291.

Kärkkäinen, Tommi, ja Mirka Saarela. 2015. “Robust Principal Component Analysis of Data with Missing Values”. Teoksessa *Machine Learning and Data Mining in Pattern Recognition,* toimittanut Petra Perner, 140–154. Cham: Springer International Publishing. ISBN: 978-3-319-21024-7.

Laurikkala, Jorma. 2001. “Improving Identification of Difficult Small Classes by Balancing Class Distribution”, 63–66. Kesäkuu. ISBN: 978-3-540-42294-5. https://doi.org/10.1007/3-540-48229-6_9.

Laurini, Robert. 2017. “5 - Geographic Relations”. Teoksessa *Geographic Knowledge Infrastructure,* toimittanut Robert Laurini, 83–109. Elsevier. ISBN: 978-1-78548-243-4. https://doi.org/https://doi.org/10.1016/B978-1-78548-243-4.50005-0. https://www.sciencedirect.com/science/article/pii/B9781785482434500050.

Leydesdorff, Loet, ja Stephen Bensman. 2006. “Classification and powerlaws: The logarithmic transformation”. *Journal of the American Society for Information Science and Technology* 57 (11): 1470–1486. https://doi.org/https://doi.org/10.1002/asi.20467. eprint: https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.20467. https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.20467.

Liberti, Leo. 2019. “Distance Geometry and Data Science” (syyskuu).

Liberti, Leo, Carlile Lavor, Nelson Maculan ja Antonio Mucherino. 2012. *Euclidean distance geometry and applications.* arXiv: 1205.0349 [q-bio.QM].

Linardatos, Pantelis, Vasilis Papastefanopoulos ja Sotiris Kotsiantis. 2021. “Explainable AI: A Review of Machine Learning Interpretability Methods”. *Entropy* 23 (1). ISSN: 1099-4300. https://doi.org/10.3390/e23010018. https://www.mdpi.com/1099-4300/23/1/18.

Lipton, Zachary Chase, Charles Elkan ja Balakrishnan Narayanaswamy. 2014. *Thresholding Classifiers to Maximize F1 Score.* arXiv: 1402.1892 [stat.ML].

Louppe, Gilles. 2014. *Understanding Random Forests: From Theory to Practice.* https://doi.org/10.48550/ARXIV.1407.7502. https://arxiv.org/abs/1407.7502.

Lundberg, Scott M, ja Su-In Lee. 2017. "A Unified Approach to Interpreting Model Predictions". Teoksessa *Advances in Neural Information Processing Systems,* toimittanut I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan ja R. Garnett, nide 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Lundberg, Scott M., Gabriel G. Erion ja Su-In Lee. 2018. "Consistent Individualized Feature Attribution for Tree Ensembles". *ArXiv* abs/1802.03888.

Maaten, Laurens van der, Eric O. Postma ja Jaap van den Herik. 2009. "Dimensionality Reduction: A Comparative Review".

Martins, André F. T., ja Ramón Fernandez Astudillo. 2016. *From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification.* https://doi.org/10.48550/ARXIV.1602.02068. https://arxiv.org/abs/1602.02068.

Mazurowski, Maciej, Piotr Habas, Jacek Zurada, Joseph Lo, Jay Baker ja Georgia Tourassi. 2008. "Training Neural Network Classifiers for Medical Decision Making: The Effects of Imbalanced Datasets on Classification Performance". *Neural networks : the official journal of the International Neural Network Society* 21 (maaliskuu): 427–36. https://doi.org/10.1016/j.neunet.2007.12.031.

Menardi, Giovanna, ja Nicola Torelli. 2012. "Training and assessing classification rules with unbalanced data". *Data Mining and Knowledge Discovery* (tammikuu). https://doi.org/10.1007/s10618-012-0295-5.

Menger, K. 1928. "Untersuchungen über allgemeine Metrik". *Mathematische Annalen* 100:75–163. http://eudml.org/doc/159284.

Menze, Bjoern, Bernd Kelm, Ralf Masuch, Uwe Himmelreich, Peter Bachert, Wolfgang Petrich ja Fred Hamprecht. 2009. "A comparison of Random Forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data". *BMC bioinformatics* 10 (elokuu): 213. https://doi.org/10.1186/1471-2105-10-213.

Miller, Tim. 2019. “Explanation in artificial intelligence: Insights from the social sciences”. *Artificial Intelligence* 267:1–38. ISSN: 0004-3702. https://doi.org/https://doi.org/10.1016/j.artint.2018.07.007. https://www.sciencedirect.com/science/article/pii/S0004370218305988.

Molnar, Christoph. 2022. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.* 2. painos. christophm.github.io/interpretable-ml-book/.

Neafus, Tom, Nathan Lieu, Chris Igleasias, Tarun Vellanki, Yoav Dekel ja Abhishek Jani. 2020. “Trajectory Characterization Using Tracktable.” (huhtikuu). https://www.osti.gov/biblio/1776675.

Nielsen, Michael A. 2018. *Neural Networks and Deep Learning.* misc. http://neuralnetworksanddeeplearning.com/.

Novaković, Jasmina Dj, Alempije Veljović, Siniša S Ilić, Željko Papić ja Tomović Milica. 2017. “Evaluation of classification models in machine learning”. *Theory and Applications of Mathematics & Computer Science* 7 (1): 39–46.

Orlenko, Alena, ja Jason H Moore. 2021. “A comparison of methods for interpreting random forest models of genetic association in the presence of non-additive interactions”. *BioData mining* 14 (1): 1–17.

Pearson, Karl. 1901. “LIII. On lines and planes of closest fit to systems of points in space”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (11): 559–572. https://doi.org/10.1080/14786440109462720. eprint: https://doi.org/10.1080/14786440109462720. https://doi.org/10.1080/14786440109462720.

Phyu, Thair Nu. 2009. “Survey of Classification Techniques in Data Mining”.

Pillai, Ignazio, Giorgio Fumera ja Fabio Roli. 2017. “Designing multi-label classifiers that maximize F measures: State of the art”. *Pattern Recognition* 61:394–404. ISSN: 0031-3203. https://doi.org/https://doi.org/10.1016/j.patcog.2016.08.008. https://www.sciencedirect.com/science/article/pii/S0031320316302217.

Refaeilzadeh, Payam, Lei Tang ja Huan Liu. 2009. “Cross-Validation”. Teoksessa *Encyclopedia of Database Systems,* toimittanut LING LIU ja M. TAMER ÖZSU, 532–538. Boston, MA: Springer US. ISBN: 978-0-387-39940-9. https://doi.org/10.1007/978-0-387-39940-9_565. https://doi.org/10.1007/978-0-387-39940-9_565.

Rintoul, Mark, ja Andrew Wilson. 2015. “Trajectory analysis via a geometric feature space approach”. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 8 (syyskuu). https://doi.org/10.1002/sam.11287.

Rokach, Lior, ja Oded Maimon. 2005. “Top-Down Induction of Decision Trees Classifiers—A Survey”. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 35 (joulukuu): 476–487. https://doi.org/10.1109/TSMCC.2004.843247.

Rumelhart, David E, Geoffrey E Hinton ja Ronald J Williams. 1986. “Learning representations by back-propagating errors”. *nature* 323 (6088): 533–536.

Schäfer, Matthias, Martin Strohmeier, Vincent Lenders, Ivan Martinovic ja Matthias Wilhelm. 2014. “Bringing up OpenSky: A large-scale ADS-B sensor network for research”. Teoksessa *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks,* 83–94. https://doi.org/10.1109/IPSN.2014.6846743.

Schäfer, Matthias, Martin Strohmeier, Matthew Smith, Markus Fuchs, Vincent Lenders, Marc Liechti ja Ivan Martinovic. 2017. “OpenSky Report 2017: Mode S and ADS-B Usage of Military and other State Aircraft”. Teoksessa *IEEE/AIAA 36th Digital Avionics Systems Conference.* DASC. Syyskuu. https://opensky-network.org/files/publications/dasc17.pdf.

Shapley, Lloyd S. 1952. *A Value for N-Person Games.* Santa Monica, CA: RAND Corporation. https://doi.org/10.7249/P0295.

Soofi, Aized Amin, ja Arshad Awan. 2017. “Classification techniques in machine learning: applications and issues”. *Journal of Basic and Applied Sciences* 13:459–465.

Strobl, Carolin, Anne-Laure Boulesteix, Achim Zeileis ja Torsten Hothorn. 2007. “Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution.” BMC Bioinformatics, 8(1), 25”. *BMC bioinformatics* 8 (helmikuu): 25. https://doi.org/10.1186/1471-2105-8-25.

Strohmeier, Martin, Matthew Smith, Vincent Lenders ja Ivan Martinovic. 2021. “Classi-Fly: Inferring Aircraft Categories from Open Data”. *ACM Transactions on Intelligent Systems and Technology* (New York, NY, USA) 12, numero 6 (marraskuu). ISSN: 2157-6904. https://doi.org/10.1145/3480969. https://doi.org/10.1145/3480969.

Suárez, A., Sánchez, F.J. Iglesias-Rodríguez, P. Riesgo Fernández ja F.J. de Cos Juez. 2016. “Applying the K-nearest neighbor technique to the classification of workers according to their risk of suffering musculoskeletal disorders” [kielellä English]. *International Journal of Industrial Ergonomics* 52 (Complete): 92–99. https://doi.org/10.1016/j.ergon.2015.09.012.

Sun, Connie, Vijayalakshmi K. Kumarasamy, Yu Liang, Dalei Wu ja Yingfeng Wang. 2022. “Using a Layered Ensemble of Physics-Guided Graph Attention Networks to Predict COVID-19 Trends”. *Applied Artificial Intelligence* 0 (0): 1–24. https://doi.org/10.1080/08839514.2022.2055989. eprint: https://doi.org/10.1080/08839514.2022.2055989. https://doi.org/10.1080/08839514.2022.2055989.

U.S. Air Force. 2022. “C-130 Hercules”. Viitattu 19. huhtikuuta 2022. https://www.af.mil/About-Us/Fact-Sheets/Display/Article/1555054/c-130-hercules/.

Wang, Kung-Jeng, Bunjira Makond, Kun-Huang Chen ja Kung-Min Wang. 2014. “A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients”. *Applied Soft Computing* 20 (heinäkuu): 15–24. https://doi.org/10.1016/j.asoc.2013.09.014.

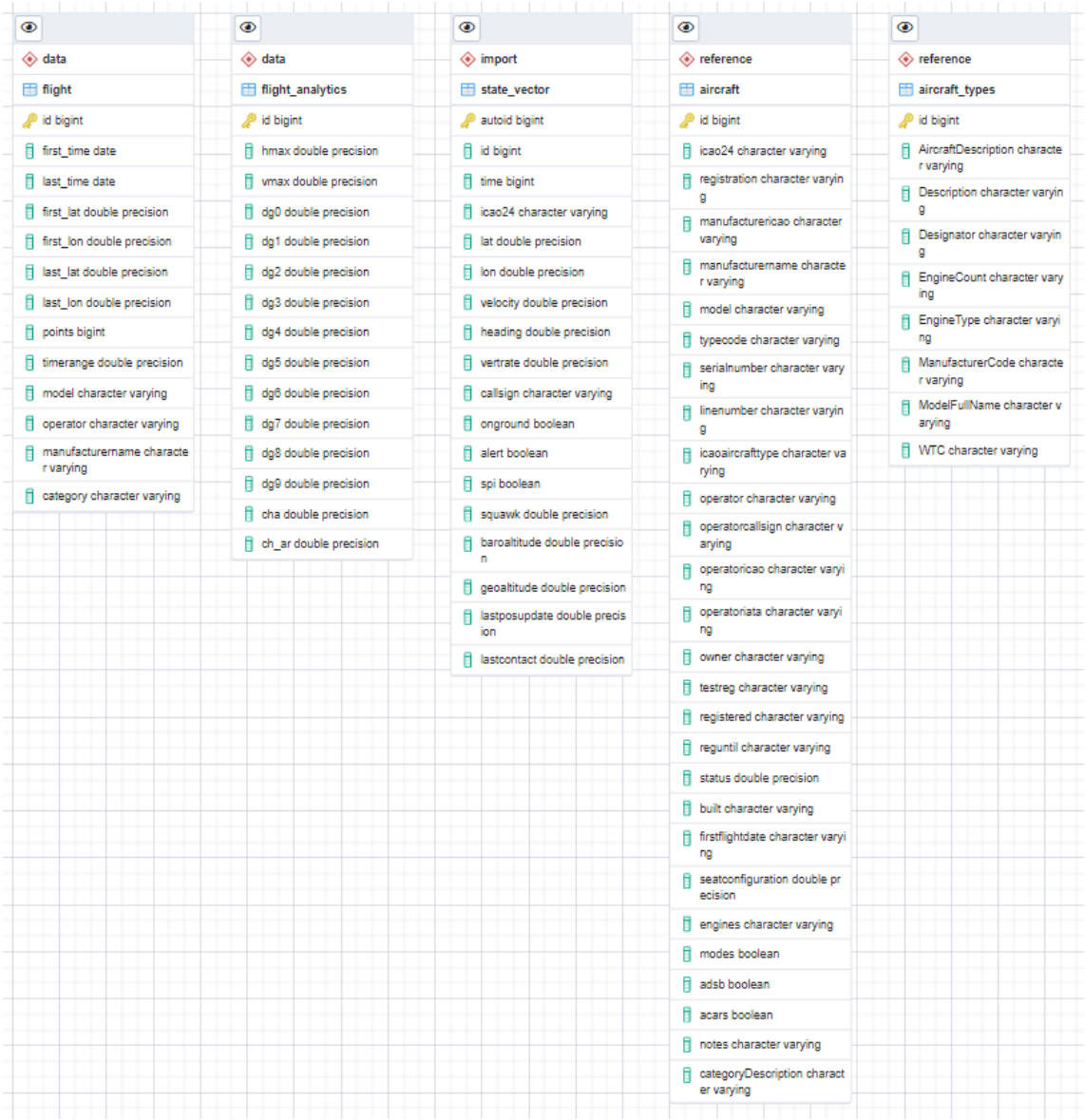
West, Darrell M. 2018. *The future of work: Robots, AI, and automation.* Brookings Institution Press.

West, Robert M. 2021. “Best practice in statistics: The use of log transformation”. PMID: 34666549, *Annals of Clinical Biochemistry* 0 (0): 00045632211050531. https://doi.org/10.1177/00045632211050531. eprint: https://doi.org/10.1177/00045632211050531. https://doi.org/10.1177/00045632211050531.

Yanminsun, Andrew Wong ja Mohamed S. Kamel. 2011. “Classification of imbalanced data: a review”. *International Journal of Pattern Recognition and Artificial Intelligence* 23 (marraskuu). https://doi.org/10.1142/S0218001409007326.

Liitteet

A Tietokannan rakenne



B Bash-skripti yhden kuukauden tilavektoriaineiston tuontiin tietokantaan

```
#!/bin/bash
# Import state vector data into database

export PGPASSWORD=""
for j in 02 09 16
do
    for i in $(seq -w 01 23)
      do
        psql -d trajectories -U postgres -c "\\copy import.
            state_vector (time, icao24, lat, lon, velocity,
            heading, vertrate, callsign, onground, alert, spi,
             squawk, baroaltitude, geoaltitude, lastposupdate,
             lastcontact)
          FROM 'C:/Users/antti.luopajarvi/gradu/data/
            state_vectors/csv/states_2020-11-$j-$i.csv/
            states_2020-11-$j-$i.csv'
          DELIMITER ',' ''
          CSV
          HEADER
          ENCODING 'UTF8'
          QUOTE '\"'
          ESCAPE '''';"
      done
done
```

C SQL-kysely tilavektoriaineiston hakuun

```
SELECT s.autoid as id, TO_TIMESTAMP(s.time) AS timestamp, s.
    icao24, s.lat, s.lon, s.velocity, s.heading, s.geoaltitude
     as altitude, ac.model, ac.manufacturername, ac.operator
FROM import.state_vector s
INNER JOIN reference.aircraft ac ON s.icao24 = ac.icao24
WHERE (((s.lat BETWEEN 55.0 AND 68.91) AND (s.lon BETWEEN 5
    AND 30.94))
  OR (LOWER(ac.operator) LIKE '%%air force%%')
  OR (LOWER(ac.operator) LIKE '%%keystone aerial surveys%%')
      // kuvauslentoja
  OR (LOWER(ac.operator) LIKE '%%school of aviation%%')
  OR (LOWER(ac.operator) LIKE '%%aviation academy%%')
  OR (LOWER(ac.operator) LIKE '%%air ambulance%%')
  OR (LOWER(ac.operator) LIKE '%%private owner%%')
  OR (LOWER(ac.operator) LIKE '%%police%%')
  OR (LOWER(ac.operator) LIKE '%%border guard%%')
  OR (LOWER(ac.operator) LIKE '%%coast guard%%')
  OR (LOWER(ac.operator) LIKE 'jet linx') // businesslentoja
  OR (LOWER(ac.operator) LIKE 'netjets')) // businesslentoja
```

D SQL-kysely luokitteluaineiston hakuun

```
SELECT f.id as flight_id, f.first_lat, f.first_lon, f.
    last_lat, f.last_lon, f.points, f.timerange as timedelta,
    fa.hmax, fa.vmax, fa.dg0, fa.dg1, fa.dg2, fa.dg3, fa.dg4,
    fa.dg5, fa.dg6, fa.dg7, fa.dg8, fa.dg9, fa.cha, fa.ch_ar,
    f.category
FROM data.flight f
INNER JOIN data.flight_analytics fa ON f.id = fa.id
WHERE f.category <> 'Unknown')
```