Tuomas Moisio

Tekoälyn soveltaminen jalkapallo vedonlyöntiin

Tietotekniikan pro gradu -tutkielma

2. huhtikuuta 2020

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Tuomas Moisio

Yhteystiedot: tusamois@student.jyu.fi

Ohjaajat: Timo Hämäläinen

Työn nimi: Tekoälyn soveltaminen jalkapallo vedonlyöntiin

Title in English: Artificial Intelligence in football betting

Työ: Pro gradu -tutkielma

Opintosuunta: Ohjelmisto- ja tietoliikennetekniikka

Sivumäärä: 79+15

Tiivistelmä: Jalkapallovedonlyönti on nykyään miljardibisnes. Vetoa voi lyödä melkein mistä tahansa maan sarjasta ja joukkueesta. Vedonlyöjälle tärkeää on ymmärtää joukkueiden väliset voimasuhteet, jonka perusteella hän voi tehdä todennäköisyysarvion ottelun lopputuloksesta. Tätä todennäköisyysarviota vedonlyöjä vertaa vedonvälittäjien tekemään todennäköisyysarvioon ja mikäli se poikkeaa vedonvälittäjän todennäköisyysarviosta, hän voi lyödä vetoa kohteesta. Todennäköisyysarvion luominen perustuu usein tilastoituun dataan, minkä analysoimisessa tietokone on ihmistä huomattavasti parempi. Tässä tutkielmassa tutkitaan, miten tätä analysointia voidaan tehdä paremmin hyödyntäen hyväksi koneoppimisen menetelmiä. Aihetta lähestytään tekoälyn ja urheiluvedonlyönnin teorian näkökulmasta, jonka jälkeen vertaillaan kolmea eri algoritmia käytännössä: tukivektorikone, k:n lähimmän naapurin menetelmä ja naiivi Bayesin luokitin.

Avainsanat: koneoppiminen, jalkapallo, vedonlyönti, tekoäly, tukivektorikone, lähimmän naapurin menetelmä, naiivi bayes

Abstract: Football betting today is a billion dollar business. You can place bets on almost any league and team in the world. For the gambler it is important to understand the power relations between the teams, so that he can make a probability estimate of the outcome of the match. The bettor compares this probability estimate to the bookmaker's own probability estimate, and if it differs, he can place a bet. Creating probability estimate is often

based on statistical data, which the computer can analyze better than human. This paper explores how this analysis can be better accomplished by utilizing machine learning. The topic is approached from the perspective of artificial intelligence and sports betting theory, followed by a comparison of three different algorithms in practice: support vector machine, k-nearest neighbor and naïve Bayes classifier.

Keywords: machine learning, football, betting, artificial intelligence, support vector machine, nearest neighbor, naïve bayes,

Termiluettelo

Alikerroin	Kerroin, joka on pienempi kuin sitä vastaavan todennäköisyyden käänteisluku.
AUC	*Area under curve*, ROC-käyrän alle jäävä pinta-ala.
C-SVC	*C-support vector classification,* joustavan marginaalin luokitin tukivektorikoneessa.
FPR	*False positive rate,* virheellisesti luokiteltujen lopputulosten osuus.
H2H	*Head to head* –pelimuoto, jossa veikataan kahden eri joukkueen tai tulosvaihtoehdon lopputulosta tai tulosvaihtoehtoa.
Hyperparametri	*Hyperparameter,* parametri, jonka arvo määritetään ennen opetusvaiheen alkamista.
kNN	*K-Nearest neighbour,* k:n lähimmän naapurin menetelmä
ROC	*Receiver operating characteristic,* luokittelijan toimivuutta kuvaava käyrä
SVM	*Support vector machine*, tukivektorikone.
TNR	*True negative rate*, oikein luokiteltujen lopputulosten osuus.
Todennäköisyysarvio	Arvio, millä todennäköisyydellä joukkue A voittaa joukkue B:n. Annetaan yleensä muodossa (xx-xx-xx), missä ensimmäinen luku kuvaa kotijoukkueen voiton todennäköisyyttä, keskimmäinen tasapelin todennäköisyyttä ja viimeinen vierasvoiton todennäköisyyttä. Todennäköisyysarvion käänteisluku on vedonvälittäjän tarjoama kerroin.
Ylikerroin	Kerroin, joka on suurempi kuin sitä vastaavan todennäköisyyden käänteisluku.

Kuviot

Taulukot

Sisältö

1 Johdanto

Vedonlyöntimarkkinat ovat vuosien saatossa kasvaneet miljarditeollisuudeksi. Vuonna 2017 vedonlyöntimarkkinan koon arvioitiin olevan 45,8 miljardia dollaria, ja sen odotetaan kasvavan vuoteen 2024 mennessä yli kaksinkertaiseksi. Tästä markkinan koosta urheiluvedonlyönnin osuus on arvioitu olevan noin 30-40%, loput 60-70% koostuvat pokeripeleistä, casinopeleistä sekä raha-automaattipeleistä. (Statista, 2019a), (Statista, 2019b)

Jopa Suomessa rahapeliyhtiö Veikkaus Oy:n liikevaihto on kohonnut vuosien saatossa yli 3 miljardiin euroon. Tästä liikevaihdosta n. 25 % muodostuu "harrastuspeleistä" kuten urheiluvedonlyönti ja totopelit. (Veikkaus, 2018)

Suomeen ensimmäiset urheiluvedonlyöntikohteet tulivat pelattavaksi vakioveikkauksen muodossa vuonna 1940. Vakioveikkauksessa vedonlyöjä veikkaa 13 kohteen lopputuloksen, 1=kotivoitto, X=tasapeli ja 2=vierasvoitto. Voittoluokkia vakiossa on neljä: 13, 12, 11 tai 10 oikein, eli voittaakseen rahaa, pelaajan täytyy veikata oikein minimissään 10 kohdetta 13:sta mahdollisesta. Pelaajan voiton määrä määräytyy pelin liikevaihdon ja osumien lukumäärän mukaan. (Veikkaus 2017; Veikkaus 2019a)

Yksittäisiä kohteita pystyi pelaamaan vasta vuonna 1993 pitkävedon muodossa, jossa veikataan yksittäisen ottelun lopputulosta tai yksittäistä ottelun tapahtumaa kuten koti- tai vierasjoukkueen maalimäärää tai keltaisten korttien määrää. Pitkävedossa voiton määrä määräytyy pelaajan panoksen sekä vedonlyöntiyhtiön määrittämän kertoimen mukaan, kerroin x panos. Vuotta myöhemmin, Veikkaus lanseerasi Tulosvedon sekä Voittajavedon, jolloin Suomalainen vedonlyönti käynnistyi kunnolla. (Veikkaus 2017; Veikkaus 2008)

Urheiluvedonlyönnissä liikkuvan rahan takia, erilaiset koneelliset menetelmät ovat yleistyneet pelaajien keskuudessa, kuten datan analysointi, robottipelaaminen ja aivan viimeisimpänä tekoäly. Näillä menetelmillä pyritään saamaan etua vedonlyöntitoimistoon sekä muihin kanssapelaajiin nähden, joko etsimällä vähän pelattuja, mutta todennäköisiä rivejä tai luomalla todennäköisyysarvioita ja otteluiden ennusteita perustuen tarjolla olevaan dataan.

Tässä pro gradu-tutkielmassa tarkastellaan yhden tekoälyn osa-alueen, koneoppimisen, metodien käyttöä ja soveltamista jalkapallo vedonlyöntiin. Tutkimus pyrkii vastaamaan kysymykseen, voiko koneoppimisen metodeja käyttämällä ennustaa jalkapallo-otteluiden tuloksia ja näin ollen tehdä voittavaa vedonlyöntiä (vedonlyöntivoitot > käytetty raha vedonlyöntiin). Tutkimus on rajattu koskemaan vain Englannin Valioliigaa, sillä maailman suurimpana sarjamuotoisena kilpailuna, siitä löytyy tarkkaa ottelukohtaista ja historiallista tietoa vuodesta 2004 asti, kun tilastopalvelu Opta alkoi tilastoimaan sen ottelutapahtumia. Tässä tutkimuksessa käytetty data on saatu avoimesta lähteestä, sivulta football-data.co.uk, ja se on yhdistetty yhdeksi kokonaisuudeksi eri kausien osalta koneellisesti tutkimusvaiheessa.

Tutkimuksessa vertaillaan kolmea eri koneoppimisen algoritmia ja niiden soveltuvuutta ennustamaan datasta ottelun lopputulos. Tutkimus toteutettiin konstruktiivisena tutkimuksena, eli toteutetaan sovellus, joka pyrkii ennustamaan sille annetun datan perusteella lopputulos. Tätä algoritmien tuottamaa ennustusta lopputuloksesta verrataan oikeaan lopputulokseen. Jokaisen ottelun H2H-kertoimen avulla saadaan selville tarkkaan, onko voittava vedonlyönti mahdollista. Algoritmit on toteutettu Python ohjelmointikielellä, käyttäen avuksi koneoppimisen kirjastoa scikit-learn. Tutkimuksessa käytettävät koneoppimisen menetelmät ovat tukivektori luokitin, k:n lähimmän naapurin menetelmä sekä naiivi Bayesin luokitin.

Tutkielman luvussa kaksi käsitellään tekoälyä ja koneoppimista yleisellä tasolla. Luvussa kolme käsitellään urheiluvedonlyönnin teoriaa ja eri pelimuotoja. Luvussa neljä käsitellään käytettävissä olevaa dataa, ja sen perusteella luotua uutta datamassaa, josta koneoppimisalgoritmit oppivat. Luvussa käsitellään myös algoritmien arvioimiseen käytettyjä eri mittareita. Luvuissa viisi, kuusi ja seitsemän käsitellään yllä lueteltujen algoritmien toimintaa, testaus tutkimuksessa käytettävällä datalla sekä kyseisten algoritmien tulokset. Luvussa kahdeksan tehdään yhteenveto tutkimuksesta.

2 Tekoäly

Tässä luvussa käsitellään tekoälyn kehitystä sen alkuajoista tähän päivään, määritelmää ja jakoa eri sovellusalueisiin. Lisäksi luvussa käsitellään tekoälyn tärkeimpien sovellusalueiden koneoppimisen, syväoppimisen ja neuroverkkojen toimintaa.

2.1 Tekoäly

Tekoälyn (eng. Artificial Intelligence) juuret ulottuvat kaukaisuudessaan jo antiikin Kreikkaan saakka. Noin 850 vuotta ennen ajanlaskun alkua, kreikkalainen filosofi ja kirjailija Homeros kuvaili kirjassaan Ilias, Hefaistos Jumalan luomia mekaanisia palvelijoita tarjoilemassa Jumalille ruokaa. Jotkut näistä "roboteista" olivat ihmisen kaltaisia ja toiset vain koneita. Tämän kuitenkin katsotaan olevan ensimmäinen kirjallinen esitys koneista, jotka osaavat ajatella ja toimia itsenäisesti. Vaikka ajatus mekaanisesta koneesta esitettiin jo antiikin Kreikassa, ei sen toteutus ollut mahdollista ennen tietokoneiden kehitystä 1940-luvulla, ja kaikki tekoälyä koskevat teokset olivat tätä ennen olleet lähinnä tieteisfiktiota, fantasiaa tai tieteellistä spekulaatiota. (McCorduck, Minsky & Simon 1977; Buchanan 2006; McCorduck 2004)

"Ajattelevia koneita" pyöriteltiin käsitteenä paljon 1940-luvulla mm. Alan Turingin ja Konrad Zusen toimesta. Varsinkin Zuse oli kiinnostunut ajattelevista koneista ja hän pyöritteli mielessään ajatusta koneesta, joka pelaisi shakkia suurmestarin (GM) tasolla. Alan Turingin kehittämän Turingin koneen myötä näistä ajatuksista ja ideoista tuli mahdollisia. 1950 Turing julkaisi paperin Computing Machinery and Intelligence, jossa hän tarkasteli hypoteettisesti mahdollisuutta luoda ajattelevia koneita. Tässä paperissa esiteltiin myös Turingin testi, eräänlainen konsepti, jolla voidaan testata koneen osoittamaa älyllistä käyttäytymistä. Turingin testissä tietokone laitetaan keskustelemaan testin tekijöiden kanssa. Jos tarkkailijat eivät kykene huomaamaan eroa tietokoneen ja ihmisen välillä, katsotaan että tietokone on älykäs. (McCorduck ym. 1977; Buchanan 2006)

Ensimmäisen kerran kone "läpäisi" Turingin testin vasta vuonna 2014, kun tietokonebotti Eugene Goostman huijasi n. 30 % Turingin testin tuomareista luulemaan, että he olivat

keskustelleet ihmisen kanssa. Tämäkään tapahtuma ei kuitenkaan todistanut tietokoneen olevan älykäs, sillä tietokonebotti Eugene Goostmanin ainoa tarkoitus oli huijata testin tuomareita esittämällä 13-vuotiasta ukrainalaista poikaa. Eugene huijasi tuomareita vastaamalla kysymyksiin kierrellen ja vastaamalla vitsein tuomareiden kysymyksiin. (Jääskeläinen, 2019)

Ensimmäisen kerran käsitteen tekoäly mainitsi yhdysvaltalainen professori John McCarthy vuonna 1956 järjestetyssä kesäseminaarissa. Seminaarissa kymmenen hengen tutkijaryhmä tutki Dartmouthin yliopistossa otaksumaa, että jokainen oppimisen tai mikä tahansa älykkyyden muoto voidaan määritellä niin tarkkaan, että koneet voivat simuloida sitä. Tutkijaryhmän mukaan, ensimmäinen ajatteleva kone oli vain yhden sukupolven päässä. (McCorduck ym. 1977; Buchanan 2006)

Pääosin tämän ennustuksen ja sen syntymiseen vaikuttaneiden tutkimusten takia, John McCarthy palkattiin professoriksi Massachusettsin teknilliseen korkeakouluun (Massachusetts institute of technology, MIT). MIT:ssa, McCarthy perusti maailman ensimmäisen tekoälylaboratorion yhdessä toisen tekoälyn pioneerin Marvin Minskyn kanssa. Kaksikko tutkijaryhmineen sai miljardien dollarien apurahan kyseisen tekoälytutkimusta varten. Tutkijaryhmän ennustus ensimmäisestä ajattelevasta koneesta sukupolven päästä osoittautuikin liian optimistiseksi. Tarvittavaa läpimurtoa ei onnistuttu tekemään ja tekoälylaboratorion rahoitus lopetettiin 1973. Tähän osasyynä oli Sir James Lighthillin tekemä raportti tekoälyn tilasta Iso-Britanniassa, jonka perusteella sekä Iso-Britannia, että Yhdysvallat lopettivat tekoälytutkimuksen rahoituksia useilta eri tutkimuskeskuksilta. Tämä käynnisti ajanjakson, jota kutsutaan tekoälytalveksi. Tällöin tekoälyn kehitystä ei juurikaan rahoitettu ja mielenkiinto tutkimusaluetta kohtaan hiipui lähes olemattomiin. (Jääskeläinen, 2019; Buchanan 2006; McCorduck 2004)

1980-luvun alussa tekoälytalvi kuitenkin loppui ja rahoitusta alettiin lisäämään jälleen tekoälyn kehitykseen. Tehtaisiin ja yrityksiin asennettiin ensimmäisiä “älykkäitä robotteja”, luonnollisen kielen käsittely koneellisesti otti harppauksia eteenpäin ensimmäisillä oikean maailman sovelluksilla ja tietokoneet menivät tasaisesti eteenpäin shakin pelaamisessa, jota pidettiin ehdottomana edellytyksenä älykkäälle käyttäytymiselle. Tärkeimpänä kuiten-

kin olivat asiantuntijajärjestelmät, jotka esiteltiin usealle eri tosielämän tilanteelle päätöksenteon tueksi kuten lääketieteeseen, osakemarkkinoille ja suurille rakennusprojekteille. Syynä tähän voidaan pitää tekoälyn osa-alueen koneoppimisen kukoistuksen alkamista. (McCorduck, 2004)

Tietokoneiden kehityttyä lisää 1990-luvulla, tekoälyn soveltamisesta eri aloille tuli helpompaa, nopeampaa ja halvempaa. Tämän seurauksena huomattiin, että tekoälyn avulla voidaan ratkaista useita erilaisia käytännön ongelmia lääketieteestä logistiikkaan. Suurena saavutuksena voidaan pitää vuotta 1997 ja sen vuoden toukokuussa tapahtunutta tapahtumaa, kun IBM:n shakkitietokone Deep Blue voitti ensimmäisenä koneena shakin hallitsevan maailmanmestarin Garry Kasparovin. (McCorduck, 2004)

Seuraavan sysäyksen tekoälyn kehitys sai 2010-luvun alussa, kun syväoppimisessa tapahtuneet läpimurrot ajoivat tekoälyn kehitystä "tekoälybuumiksi" asti. Vaikka syväoppiminen keksittiin jo 1950-luvulla samoihin aikoihin termin tekoäly kanssa, ei sitä oltu aktiivisesti tutkittu ennen 1980-lukua, johtuen tutkijoiden vähäisestä tiedosta neuroverkkojen osalta. Syväoppiminen on koneoppimisen yksi osa-alueista, joka mahdollistaa laitteen itseoppimisen sille annettujen tehtävien suorittamiseksi. (Copeland, 2016)

Vaikka tekoälyä on kehitetty jo yli 60 vuotta, ei sille ole kyetty antamaan yksiselitteistä määritelmää. Tämä johtuu siitä, että tekoälyyn vahvasti liittyvän ja sen yläluokan kategoriana toimivan älykkyyden määrittely on ollut ihmisille vaikeaa jo monta kymmentä vuotta. Koska tekoälyn määritelmä liittyy vahvasti älykkyyden määritelmään, se perii älykkyyden määrittelyssä vallitsevat ongelmat. Toinen syy tekoälyn määrittelyn vaikeudelle on sen laajuus ja alati muuttuvat tekniikka. Tekoälyä tutkimusaiheena määritellään jatkuvasti uudelleen, kun tietyt aihepiirit kasvavat omaksi tutkimusalueekseen ja näin ollen niiden ei enää katsota kuuluvaksi tekoälyn aihealueeseen. Esimerkiksi viisikymmentä vuotta sitten automaattiset reitinoptimointimenetelmät laskettiin tekoälyksi, mutta nykyään ne luokitellaan osaksi tietojenkäsittelytieteiden perusteita. (Helsingin yliopisto & Reaktor, 2019)

Koska tekoälyn määrittely on vaikeaa, tässä tutkielmassa esitellään monta erilaista määritelmää tekoälylle. Kansainvälisen ICT-alan tutkimus- ja konsultointiyritys Gartner, Inc:n luoman tietotekniikan termien sanaston mukaan, tekoäly soveltaa edistynyttä analyysiä ja

logiikkapohjaisia tekniikoita, kuten koneoppiminen, tapahtumien tulkitsemiseen sekä päätöksen teon tueksi. Toinen tekniikan alan järjestö IEEE (Institute of Electirical and Electronics) ei edes käytä käsitettä tekoäly, vaan se korvaa käsitteen autonomisilla ja älykkäillä järjestelmillä. (Gartner Inc, 2019)

Suomen Valtionvarainministeriön selonteossa "Eettistä tietopolitiikkaa tekoälyn aikakaudella" tekoäly määriteltiin tietokoneohjelmiksi, "joiden avulla koneet, laitteet, ohjelmat, järjestelmät ja palvelut voivat toimia älykkäästi, eli joustavasti ja tarkoituksenmukaisesti monimutkaisessa ja osin ennustamattomassa ympäristössä". (Raskulla, 2019)

Samoilla linjoilla on Lappeenrannan-Lahden teknillisen yliopiston professori Atte Jääskeläinen, joka kuvaa tekoälyä seuraavasti: "Tietokoneet pystyvät toimintaan, jonka on perinteisesti ajateltu vaativan ihmisälyä, ja kykenevät itsenäisesti mukauttamaan toimintaansa niille annetun datan perusteella." Tekoälyn keskeisimpiä sovelluksia ovat ennustaminen datan perusteella, datan luokittelu, datan ryhmittely ja poikkeamien tunnistaminen. (Jääskeläinen, 2019)

Siukkosen ja Neittaanmäen kirjassa "Mitä tulisi tietää tekoälystä" on kuvattu osuvasti tekoälyn toimintaa seuraavasti: "Tekoäly on yksi tietokoneen toimintojen jatke, mittaviin laskentoihin kykenevä ohjelma tai järjestelmä. Sen englanninkielinen nimike artificial intelligence (AI) tarkoittaa tietokoneen tuottamaa keinotekoista älykkyyttä luonnollisen älykkyyden (*engl. natural intelligence*, NI) tai luonnollisen oppimisen (*engl. organic learning*, OL) vastaparina". (Siukkonen & Neittaanmäki, 2019)

Yhteistä näille määritelmille ovat käsitteet autonomisuus ja adaptiivisuus. Tekoälyn autonomisuus tarkoittaa sitä, että tekoäly kykenee suorittamaan tehtäviä mahdollisesti monimutkaisessa ympäristössä ilman jatkuvaa ohjaamista. Adaptiivisuus puolestaan kuvaa tekoälyn kykyä "parantaa suorituskykyä oppimalla kokemuksesta". (Helsingin yliopisto ym., 2019)

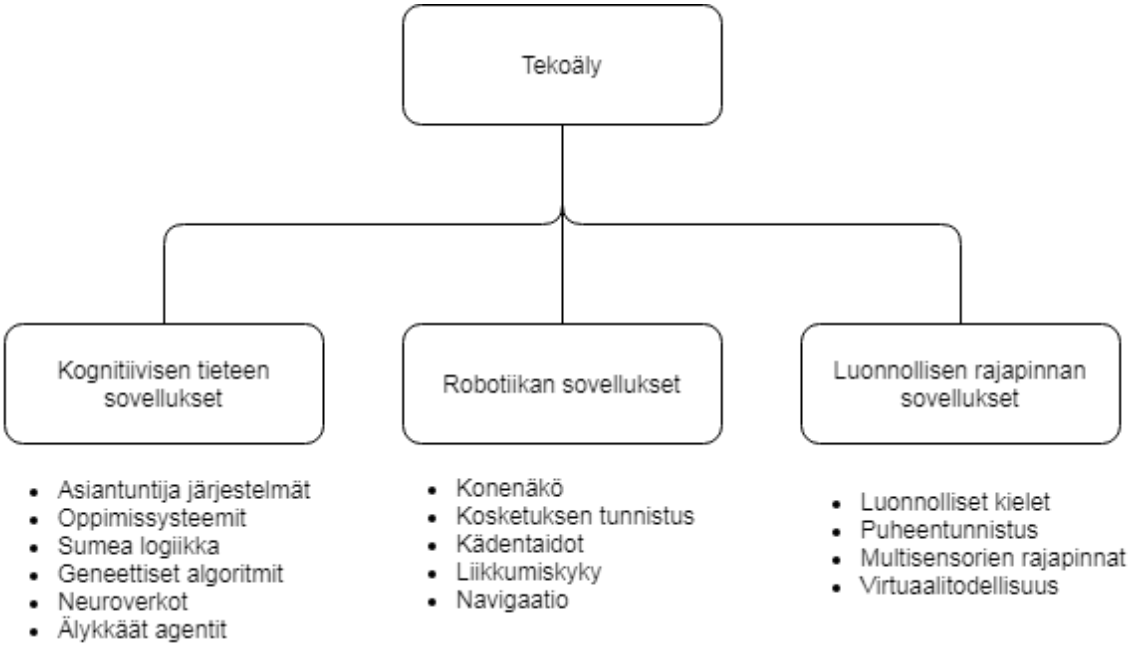
Täytyy ottaa huomioon, että kaikki esitellyt määritelmät kuvaavat kapeaa tekoälyä (*engl. narrow AI*), eli järjestelmää tai ohjelmaa, joka osaa suorittaa tiettyjä älykkäiltä vaikuttavia toimenpiteitä yksi tehtävä kerrallaan. Kapean tekoälyn vastakohtana toimii yleinen tekoäly (*engl. general AI, artificial general intelligence*), joka puolestaan viittaa järjestelmään tai

ohjelmaan, joka osaa ratkaista minkä tahansa älyllisesti ratkaistavissa olevan ongelman tai ongelmat ja pystyy suorittamaan useita tehtäviä useissa eri konteksteissa ja useissa eri ympäristöissä. Yleinen tekoäly on tuttu tieteiskirjallisuudesta, mutta oikeassa elämässä tutkijat eivät ole saavuttaneet edistystä yli 50 vuoteen. (Helsingin yliopisto ym., 2019; Goertzel, 2014)

Vastaava kahtiajako voidaan tehdä ”vahvan” ja ”heikon” tekoälyn välillä. Tämä jako perustuu erotteluun älykkyyden ja älykkään toiminnan välillä. Heikko tekoäly tarkoittaa järjestelmiä tai ohjelmia, jotka koneoppimista ja algoritmeja hyödyntämällä voivat suoriutua tehtävästä tekemällä älykkäiltä vaikuttavia toimintoja, olematta kuitenkaan sanan varsinaisessa merkityksessä älykkäitä esimerkiksi shakkia pelaava tietokone tai robotti-imuri. (Helsingin yliopisto ym., 2019; Siukkonen ym., 2019)

Vahvalla tekoälyllä tarkoitetaan aidosti älyllistä ja tietoista olentoa tai konetta. Vahvan tekoälyn tutkijoiden mukaan tietokoneeseen voidaan ohjelmoida ihmisviisauden kaltaista tietoisuutta olemassaolosta. Tämän tilan tietokone voisi saavuttaa matkimalla ihmisen aivotoimintaa, ja sitten ylittämällä sen älylliset kyvyt. Vahvan tekoälyn luomiseksi, tutkijoiden pitäisi kuitenkin pystyä vastaamaan kysymyksiin voiko kone olla tietoinen tilastaan tai saavuttaa olotilan, jossa kone määrittää sen omat pyrkimykset ja tavoitteet. (Helsingin yliopisto ym., 2019; Siukkonen ym., 2019)

Perinteisen kahtiajaon lisäksi, tekoäly voidaan jakaa osa-alueisiin sen sovellusalueiden perusteella (Kuva 1); 1. kognitiivisen tieteen sovellukset, 2. Robotiikan sovellukset ja 3. Luonnollisen kielen sovellukset. Kuvassa 1 esitetään sovellusalueiden lisäksi myös niiden käyttämät tärkeimmät tekoälyn eri menetelmät. (Vähäkainu & Neittaanmäki, 2018)



Kuva 1 Tekoälyn jako osa-alueisiin

2.2 Koneoppiminen

Yksi hyväksytyistä älykkyyden määritelmistä on peräisin Yhdysvaltalaiselta psykologilta ja professorilta Linda Gottfredsonilta, joka määrittelee älykkyyden olevan “yleinen henkinen kyky, johon sisältyy muun muassa kyky perustella, suunnitella, ratkaista ongelmia, ajatella abstraktisti, ymmärtää monimutkaisia ideoita, oppia nopeasti ja oppia kokemuksesta. Se ei ole pelkästään kirjoista oppimista, kapeaa akateemista taitoa tai testien tekemistä. Pikemminkin se heijastaa laajempaa ja syvempää kykyä ymmärtää ympäristöämme ja selvittää mitä tehdä.” (Gottfredson, 1997)

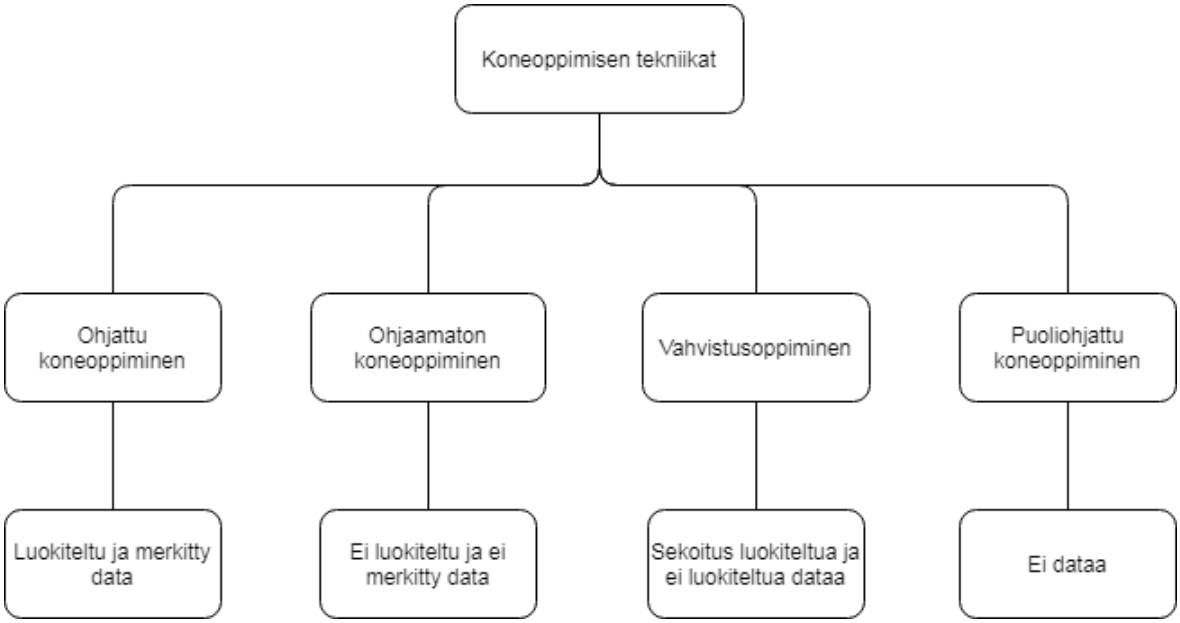
Kuten aikaisemmin mainittiin, Gottfredsonin määrittelemän älykkyyden eli toisin sanoen vahvan tekoälyn ohjelmoiminen tietokoneeseen on vielä tänä päivänä mahdotonta, mutta osa yllä mainitusta älykkyyden määrittelystä voidaan siirtää koneeseen tai ohjelmaan, kuten ongelmanratkaisu, oppiminen nopeasti ja oppiminen kokemuksesta. (Siukkonen ym., 2019)

Oppimisen siirtäminen tietokoneeseen tehdään tilastotieteen ja datan avulla. Tätä tietokoneen opettamista datan avulla kutsutaan koneoppimiseksi. Terminä koneoppiminen tarkoittaa datasta löytyvien merkitsevien kaavojen ja mallien automatisoitua tunnistamista. Sen avulla pyritään luomaan koneelle älykkyyttä, käyttämällä hyväksi tilastollisia oppimismenetelmiä. Tällaisia tilastollisia menetelmiä ovat esimerkiksi lineaarinen regressio ja bayesiläisen tilastotiede, jotka on keksitty jo yli 200 vuotta sitten. (Helsingin yliopisto ym. 2019; Shalev-Shwartz & Ben-David 2014)

Koneoppimisella katsotaan olevan kaksi eri tehtävää: ensimmäisenä tehtävänä on antaa selkeyttä ongelmille, joihin ihminen ei kykene muodostamaan itse toimivaa algoritmia. On olemassa tehtäviä, joita ihmiset suorittavat vaivattomasti ja rutiininomaisesti, mutta me emme osaa sanoa miten teemme sen. Esimerkiksi tunnistamme ystäviemme äänen väkijoukosta, mutta emme osaa selittää miten se käytännössä tapahtuu. Koska emme ymmärrä tätä ilmiötä kunnolla, emme osaa luoda ongelman pohjalta koneellista algoritmia, joka osaisi puheentunnistusta samassa määrin kuin ihminen. Tätä ymmärryksen tuottamaa ongelmaa koneoppiminen yrittää korjata. (Mohammed, Khan & Bashier 2017).

Koneoppimisen toisena tehtävä on ratkaista ongelmia, joita ihmiset eivät kykene itse suorittamaan. Tällaiset tehtävät liittyvät lähinnä erittäin suurten ja monimutkaisten datajoukkojen analysointiin: tähtitieteellinen data, lääketieteellisten arkistojen muuntaminen lääketieteelliseksi tiedoksi, sään ennustaminen ja esimerkiksi genomitietojen analyysi. Koska digitaalisesti tallennettua dataa on yhä enemmän ja enemmän ihmisten saatavilla, on selvää, että datan joukossa on piilotettuna merkitsevää informaatiota ja malleja, jotka ovat liian suuria ja monimutkaisia ihmisille ymmärtää. Tietokoneiden muistin ja tehokkuuden lisääntymisen ansiosta, koneoppiminen avaa vuosi vuodelta uusia mahdollisuuksia datan mallien havaitsemiselle. (Shalev-Shwartz ym., 2014)

Koneoppiminen jaetaan neljään eri osa-alueeseen ratkaistavien ongelmien luonteesta ja koneoppimisalgoritmille annetun datan luonteesta riippuen: ohjattu koneoppiminen (*engl. supervised learning*), ohjaamaton koneoppiminen (*engl. unsupervised learning*), puoliohjattu koneoppiminen (*engl. semi-supervised learning*) ja vahvistusoppiminen (*engl. reinforcement learning*) (Kuva 2). (Mohammed ym., 2017)



Kuva 2 Koneoppimisen tekniikkojen jako eri osa-alueisiin.

Ohjatun koneoppimisen menetelmissä tarkoituksena on päätellä funktio tai kartoitus saatavilla olevasta opetusdatasta (*engl. training data*), jonka lopputulos tiedetään (*engl. labeled data*). Tämä opetusdata koostuu tunnisteiden syötevektorista X ja tulosvektorista Y. Vektorin Y:n tunniste on vektori X:n syötteiden selite. Yhdessä nämä muodostavat opetusesimerkin (*engl. training example)*. Ohjatussa koneoppimisessa käytetty opetusdata siis muodostuu erilaisista opetusesimerkeistä. Datan tunnisteiden tai luokkien avulla, ohjatun koneoppimisen algoritmille kerrotaan muuttuja datasta, joka algoritmin pitää ennustaa. (Mohammed ym., 2017)

Tulosvektori Y koostuu jokaisen opetusesimerkin luokista opetusdatassa. Nämä luokat tulosvektorille on antaa ohjaaja *(engl. supervisor).* Yleensä, nämä ohjaajat ovat ihmisiä, sillä ihmisten arviointikyky sisältää vähemmän virheitä kuin koneen arviointikyky, vaikka ovatkin kalliimpia käyttää. Esimerkki tällaisesta datasta olisi tietokanta kuvista, joihin ihminen määrittää, sisältävätkö kuvat auton.

Ohjatun koneoppimisen menetelmät jaetaan kahteen eri data-analyysin osa-alueeseen: regressiomenetelmiin ja luokittelumenetelmiin. Molempia käytetään tärkeiden tietojen kuvaamiseen mallien avulla tai tulevaisuuden ennustamiseen. Luokittelumenetelmät ennustavat kategorisia (diskreettejä, järjestämättömiä) luokkia, kun taas regressiomenetelmät ar-

vioivat jatkuvia luokkia. Bhavsarin ja Ganatran mukaan luokittelumenetelmät pystyvät käsittelemään dataa laajemmin kuin regressiomenetelmät, mikä on johtanut luokittelumenetelmien suosion nousemiseen. (Bhavsar & Ganatra, 2012)

Ohjaamattomassa koneoppimisessa, data ei sisällä ohjaajia eikä näin ollen tunnisteita. Ohjaamattoman koneoppimisen tarkoituksena on löytää tällaisesta datasta piilotettuja rakenteita ja säännönmukaisuuksia. Datan tunnisteiden puuttumiseen voi olla useita eri syitä. Joskus datalle ei ole mahdollista luoda tunnisteita sen kustannusten takia tai datan luontaisten ominaisuuksien takia se ei yksinkertaisesti ole mahdollista. Tällaista dataa on esimerkiksi sensoreista kerättävä aikasarjadata, joka voi kasvaa sekunnissa miljoonia rivejä (Big Data). (Mohammed ym., 2017)

Ohjaamattomassa koneoppimisessa menetelmät jaetaan myös kahteen osaan: klusterointiin (*engl. clustering)* ja dimensionaalisuuden vähentämiseen *(engl. dimensionality reduction).* Klusteroinnin tehtävänä on ryhmitellä joukko objekteja siten, että samanlaiset objektit päätyvät samaan ryhmään ja erilaiset objektit eri ryhmään. Dimensionaalisella vähentämisellä tarkoitetaan puolestaan prosessia, jossa korkean dimensionaalisuuden data kartoitetaan uuteen tila-avaruuteen, jonka dimensionaalinen tila on pienempi. Tällä pyritään laskemaan korkean dimensionaalisuuden aiheuttamaa laskennallista haastetta. Lisäksi korkean dimensionaalisuuden data voi johtaa huonoon lopputulokseen algoritmien osalta. Datan dimensionaalinen alentaminen voi auttaa myös datan klusteroinnin osalta. (Shalev-Shwartz ym., 2014)

Puoliohjattu koneoppiminen sisältää ominaisuuksia sekä ohjatusta että ohjaamattomasta koneoppimisesta. Puoliohjatussa koneoppimisessa data on luokatonta (*engl. unlabeled*), mutta algoritmille annetaan jotain ohjaajan tuottamaa informaatiota. Yleensä tämä tieto liittyy datasta saatuihin esimerkkitapauksiin. (Chapelle, Schölkopf & Zien, 2006).

Vahvistusoppiminen pyrkii käyttämään datana vuorovaikutuksesta ympäristön kanssa kerättyjä havaintoja toimenpiteisiin, jotka maksimoivat palkkion tai minimoivat riskin. Vahvistusoppimisen metodi on seuraavanlainen:

1. Syötteen lähtötila observoidaan.
2. Ohjelma käyttää päätöksenteko-toimintoa suorittaakseen toiminnon.

3. Toiminnon jälkeen, ohjelma saa palkinnon tai vahvistuksen seurattavalta ympäristöltä.
4. Palkkiota koskeva tila-toiminta pari tallennetaan.

Tallennettujen tietojen avulla, tietyn tilan toimintatapaa voidaan hienosäätää ja sen kautta auttaa ohjelmaa optimaaliseen päätöksentekoon. (Mohammed ym., 2017)

Tässä tutkielmassa käytetyt algoritmit kuuluvat ohjatun koneoppimisen luokittelumenetelmiin, koska jalkapallo-ottelun lopputulos edustaa kolmea mahdollista luokkaa (kotivoitto, tasapeli ja vierasvoitto). Näiden kolmen tuloksen lisäksi, luokittelun katsotaan olevan moniluokkaista luokittelua, binäärisen eli kaksiluokkaisen luokittelun sijaan.

3 Urheiluvedonlyönti

Tässä luvussa käsitellään urheiluvedonlyönnin pelimuotoja, teoriaa ja urheiluvedonlyöntiin liittyviä käsitteitä. Urheiluvedonlyönniksi määritellään kaikki rahapelit, joissa yritetään arvioida urheilutapahtumien tuloksia, kuten lopputuloksia, puoliaikatuloksia, maalimääriä, erikoistilanteiden lukumääriä tms. Ulkopuolelta katsottuna se vaikuttaa helpolta, täytyy veikata oikein vain ottelun tulos, mutta tilastollisesti vain pieni osa kaikista pelaajista jää plussalle vedonlyönnissä eli harrastaa voittavaa vedonlyöntiä. (Buchdal, 2003)

3.1 Pelimuodot

Vuonna 2019 urheiluvedonlyönnin pelimuotoja on pilvin pimein, ja niiden määrä kasvaa tasaiseen tahtiin. Tässä alaluvussa käsitellään tunnetuimmat ja suosituimmat pelimuodot maailmalla keskittyen vedonlyöntiin Suomessa. Pelimuotoja on vaikea jaotella, sillä vedonlyöntiyhtiöt käyttävät hieman eri nimiä eri pelimuodoista.

3.1.1 Pitkäveto

Pitkävedossa pelaat kohteen voittajaa tai tulosyhdistelmää 1 – 10 kohteelle (1 = kotivoitto tai ensin mainitun tulosvaihtoehdon toteutuminen, X = tasapeli tai keskimmäisenä mainitun tulosvaihtoehdon toteutuminen, 2 = vierasvoitto tai viimeisenä mainitun tulosvaihtoehdon toteutuminen). Jos tulosvaihtoehtoja on useampi kuin kolme, määritellään kohdeottelun/kohdekilpailun kaikki tulosvaihtoehdot erikseen. Voittosumma määräytyy peliin sisältyvien tulosvaihtoehtojen yhteiskertoimesta ja valitusta panoksesta. Pelin kokonaishinnan minimi on 1 €. Samalla panoksella voi vetoon valita 1 - 10 kohdetta. Kohteiden määrä ei vaikuta pelin hintaan. (Veikkaus, 2019b)

Kaikki kohteet ovat pelattavissa yksittäisvetoina (Single). Järjestelmäpelaaminen mahdollistaa useiden yhdistelmien pelaamisen samalla kupongilla. Pelaaja voi valita peliin kaikki yhdistelmät, jotka valitsemien merkkien sisällä on muodostettavissa. Esimerkiksi triplassa pelataan kolmen (3) ja nelosissa neljän (4) ottelun tulosyhdistelmää. Saman ottelun eri pe-

likohteita ei voi yhdistää keskenään samaan peliin (esim. jääkiekko-ottelun perus- ja tasoituskohdetta). (Veikkaus, 2019b)

Kolmen vaihtoehdon (1X2) kohteessa lyödään vetoa varsinaisen peliajan tuloksesta ja kahden vaihtoehdon (12) kohteessa koko ottelun (mukaan lukien jatkoaika, rangaistuspotkukilpailu, tms.) tuloksesta, ellei kohteessa toisin mainita. Poikkeuksena ovat kahden vaihtoehdon tasoitus- ja maalimääräkohteet (esim. yli/alle -kohteet), joissa lyödään vetoa varsinaisen peliajan tuloksesta. (Veikkaus, 2019b)

Tasoituskohteessa kertoimiin vaikuttavat joukkueille annetut tasoitukset. Kohteissa lyödään vetoa ottelun lopputuloksesta varsinaisella peliajalla, joka perustuu koti- tai vierasjoukkueelle annettuun tasoitukseen. Tasoituksen määrä on merkitty sen kohdejoukkueen kohdalle, jolle tasoitus annetaan. Kertoimet saattavat muuttua kohteen ollessa auki. Voimassa on pelaamishetkellä järjestelmässä oleva kerroin, joka taltioituu pelitositteeseen. (Veikkaus, 2019b)

3.1.2 Tulosveto

Tulosvedossa pelataan kohteena olevan ottelun kummankin joukkueen maalimääriä varsinaisella peliajalla tai muita tulosperusteena olevien suoritusten lukumäärän mukaisia oikeita tuloksia. Tulosvedon suosituimpia kohdelajeja ovat jalkapallo, jääkiekko. koripallo ja salibandy. Pelin panos vaihtelee 1,00:sta 100,00 euroon. Tulosvedon kertoimet voivat olla joko kiinteitä tai muuttuvia. Veikkaus voi muuttaa kiinteää kerrointa pelin ollessa avoin. Pelaaja saa peliinsä sen kertoimen, joka on voimassa pelitietojen tallentumishetkellä. Muuttuva kerroin lasketaan pelin päätyttyä ja siihen vaikuttaa tulosvaihtoehtoon pelattujen panosten kokonaissumma. (Veikkaus, 2019c)

Tulosvedon pelimuotoja ovat hajarivi, säästöjärjestelmä ja järjestelmä. Hajarivissä pelataan yksi maalimäärä kummallekin kohdepelin joukkueelle. Järjestelmäpelaamisella tarkoitetaan vähintään kahden vaihtoehtoisen maalimäärän pelaamista samalle joukkueelle. Peli sisältää tuolloin kaikki ne yksittäiset tulosyhdistelmät, jotka valittujen maalimäärien avulla voidaan muodostaa. Säästöjärjestelmä tarkoittaa järjestelmää, johon sisältyvistä tulosvaih-

toehdoista on poistettu kotivoitot, tasapelit tai vierasvoitot pelaajan valinnan mukaisesti. (Veikkaus, 2019d)

Veikkauksen peleissä tulosvedon palautusprosentti on 80 %. Kertoimet määräytyvät siten, että palautettava summa ($0{,}8 * vaihto$) jaetaan tulosta/tulosyhdistelmää pelatulla euromäärällä. Esimerkiksi jos kohteen vaihto on 45 000 €, ja jalkapallo-ottelun tulosta 1-1 on pelattu 7 500 eurolla, kertoimeksi muodostuu $0{,}8 * 45\,000 / 7\,500 = 4{,}80$. (Veikkaus, 2019d)

Tämä tekee tulosvedosta totalisaattoripelin, eli vedonlyöjä ei tiedä vetoa ostaessaan, minkä kertoimen hän lopulta saa. Jos pelivaihto on pieni, saattavat viime minuuttien liikkeet olla suuria ja yllätyksellisiä ja vaikuttaa tulosvedon kertoimeen huomattavasti. (Vuoksenmaa ym., 1999)

3.1.3 Moniveto

Moniveto on Tulosvedosta kehitetty muuttuvakertoiminen vedonlyöntipeli, jossa pelataan 2 – 6 ottelussa kohdejoukkueiden tekemien maalien määriä tai muita tulosperusteena olevien suoritusten lukumäärän mukaisia oikeita tuloksia. Kohdeottelu voi olla myös kohdelistassa määritelty lajin sääntöjen mukainen pelijakso tai muu tapahtuma. Neljän, viiden tai kuuden kohdeottelun tapauksessa pelistä käytetään nimeä SuperMoniveto. (Veikkaus, 2019e)

Monivedon minimipanos on kohteesta riippuen 0,05 ja 0,20 euron välillä. Maksimipanos on 100 euroa. Voittosumma määräytyy panoksen ja pelitapahtumista laskettavan muuttuvan kertoimen mukaan. Monivedon kohteena on lähinnä jalkapalloa ja jääkiekkoa. (Veikkaus, 2019e)

Monivedon pelimuotoja on peruspeli, säästöjärjestelmä ja järjestelmä. Pelitapoja ovat itse valitut tai pelijärjestelmän arpomat maalimäärät tai tulokset. Peruspelissä pelataan yksi maalimäärä jokaiselle kohdepelin joukkueelle. Järjestelmäpelaamisella tarkoitetaan vähintään kahden vaihtoehtoisen maalimäärän pelaamista samalle joukkueelle. Peli sisältää tuolloin kaikki ne yksittäiset tulosyhdistelmät, jotka valittujen maalimäärien avulla voidaan

muodostaa. Säästöjärjestelmä tarkoittaa järjestelmää, johon sisältyvistä tulosvaihtoehdoista on poistettu kotivoitot, tasapelit tai vierasvoitot pelaajan valinnan mukaisesti. Monivedon palautusprosentti on 70. Kahden voittoluokan Monivedossa ylemmälle voittoluokalle jaetaan 50 % ja alemmalle voittoluokalle 20 % vaihdosta. Kertoimet määräytyvät siten, että voittoluokalle osoitettu osuus vaihdosta jaetaan tulosta/tulosyhdistelmää pelatulla euromäärällä, samoin kuin tulosvedossa. (Veikkaus, 2019f)

Monivedossa pelitositteella miinusmerkki maali- tai pistemäärän perässä (esim. 16-, 70-) tarkoittaa vaihtoehtoa maali-/pistemäärä tai sen alle ja plusmerkki (esim. 8+, 15+, 100+) vaihtoehtoa maali-/juoksu-/pistemäärä tai sen yli. (Veikkaus, 2019f)

Monivedossa voittojen koot vaihtelevat suuresti. Suurin kerroin monivedossa on ollut yli kaksi miljoonaa. Tämän takia monivedossa pyritään suurvoittoihin samoin kuin lotossa tai vakioveikkauksessa. Kuten tulosvetokin, myös moniveto on totalisaattoripeli, eli pelaajan vedon lopullinen kerroin selviää vasta kohteen suljettua. (Vuoksenmaa ym., 1999)

3.1.4 Vakioveikkaus

Vakiossa veikataan 6 – 18 kohteen voittajia varsinaisella peliajalla (1=kotivoitto, X=tasapeli tai 2=vierasvoitto) tai kahden taikka kolmen kilpailijan keskinäisen kilpailun tulosta taikka tulosvaihtoehdon toteutumista. Vakiossa valitaan voittajat jokaiseen kohteeseen. Vakio 1:ssä on aina 13 pelikohdetta, muissa Vakioissa 6 – 18. Vakioissa on 1-4 voittoluokkaa, joissa voitto-osuus määräytyy pelin liikevaihdon ja osumien lukumäärän mukaan. Vakion rivihinta vaihtelee kohteittain 0,10 - 0,25 euron välillä. Vakion kohdelajeina on lähinnä jalkapalloa, jääkiekkoa, formulaa ja yksilölajeja. (Veikkaus, 2019g)

Vakion pelimuotoja ovat hajarivi, järjestelmä ja haravajärjestelmä. Hajarivejä voit pelata joko omavalintaisilla numeroilla tai pelijärjestelmän arpomina pikapeliriveinä. Vakiota voi pelata myös osittaisena pikapelinä itse asetettuun hintarajaan asti. Osittaisessa pikapelissä valitaan osa voittajista itse ja annetaan pelijärjestelmän arpoa loput voittajista. Pelijärjestelmä tuottaa asetettuun hintarajaan sopivan peliehdotuksen, johon voittajat on arvottu senhetkisen suosituimmuuslistan mukaan. (Veikkaus, 2019h)

Järjestelmäpelissä haluttuihin otteluihin voi merkitä useamman kuin yhden tulosvaihtoehdon, jolloin peli sisältää samalla kertaa useita hajarivejä. Täydellinen järjestelmä pitää sisällään kaikki ne hajarivit, jotka valituilla merkeillä voidaan muodostaa. Haravajärjestelmä eroaa täydellisestä järjestelmästä siten, että siihen on poimittu vain ns. avainrivit. Haravajärjestelmät on jaettu kolmeen ryhmään minimitakuun mukaan. Minimitakuu kertoo, mikä on haravan taattu vähimmäistulos, kun riviin osuu ”kaikki oikein” tulos. Niin järjestelmien kuin haravajärjestelmienkin kokonaishinta on järjestelmän sisältämä rivimäärä kerrottuna rivihinnalla. (Veikkaus, 2019h)

Järjestelmäpelit ovat täydellisiä järjestelmiä. Järjestelmissä haluttuihin kohteisiin voi merkitä useampia tulosvaihtoehtoja. Peli sisältää kaikki ne yksittäisrivit, jotka valituista numeroista voidaan muodostaa. Tositteesta näkee heti arvonnan jälkeen, mitä vähintään voittaa. Tämän lisäksi järjestelmällä saa alavoittoja eli pienempien voittoluokkien voittoja. (Veikkaus, 2019h)

Vakion voittoluokat vaihtelevat kohdemäärän mukaan; 13 kohteen Vakiossa voittoluokkia ovat 13, 12, 11 ja 10 oikein, 12 kohteen Vakiossa voittoluokkia ovat 12, 11 ja 10 oikein, 11 kohteen Vakiossa voittoluokkia ovat 11 ja 10 oikein ja 10 kohteen Vakiossa voittoluokkia ovat 10 ja 9 oikein.

3.1.5 Voittajavedot

Voittajavedossa pelataan kohteena olevan tapahtuman voittajia, määriteltyjä oikeita tulosyhdistelmiä tai tulosvaihtoehtoja. Suosituimpia kohdelajeja ovat jalkapallo, jääkiekko ja formulat. Pelin panos vaihtelee 0,20:sta 100,00 euroon ja kertoimet voivat olla kiinteitä tai muuttuvia. (Veikkaus, 2019i)

Voittajavedon muita pelimuotoja ovat Superkaksari, Supertripla, Päivän Pari ja Päivän Trio. Superkaksarissa vedonlyönnin kohteena on kilpailun voittaja ja toiseksi sijoittuva kilpailija paremmuusjärjestyksessä. Supertriplassa vedonlyönnin kohteena on kilpailun voittaja, toiseksi sijoittuva ja kolmanneksi sijoittuva kilpailija paremmuusjärjestyksessä. Päivän Parissa vedonlyönnin kohteena ovat kahden eri kilpailun voittajat tai oikeat määritellyt tulosyhdistelmät tai -vaihtoehdot. Päivän Triossa vedonlyönnin kohteena ovat kol-

men eri kilpailun voittajat tai oikeat määritellyt tulosyhdistelmät tai -vaihtoehdot. (Veikkaus, 2019i)

Toisin kuin pitkäveto, tulosveto, moniveto tai vakioveikkaus, eri voittajavedot voivat olla pitkäaikaisvetoja, jotka saattavat kestää koko sarjan ajan (kuten vaikka Valioliigan maalipörssin voittaja). Tämän takia ne ovat erittäin suosittuja pelikansan keskuudessa, sillä pienellä rahalla saa jännitystä jopa vuoden ajaksi. (Vuoksenmaa ym., 1999)

3.1.6 Live-veto

Live-vedossa lyödään vetoa jo käynnissä olevasta ottelusta tai kilpailusta. Live-vedossa voi pelata esimerkiksi ensimmäistä maalintekijää, maalin syntymisaikaa, tuleeko tietyllä aikavälillä jäähyä tai kulmapotkujen lukumäärää. Pelikohde sisältää yhden tai useampia tulosvaihtoehtoja, joille on määritelty kiinteät kertoimet. Kaikki vedot ovat yksittäisvetoja. Live-vedossa voitto muodostuu pelaajan panoksesta ja vedonvälittäjän antamasta kertoimesta, panos * kerroin. Live-vedon etuna on pelaajalle se, että hän voi korjata omia todennäköisyysarvioitaan kisan/ottelun edetessä. Mikäli tämä tapahtuu nopeammin kuin pelinjärjestäjän luoma todennäköisyysarvio, voi pelaaja saavuttaa hyviä ylikertoimia live-vedosta. Veikkaus on arvioinut oman live-vetonsa palautusprosentikseen n. 90 %. (Veikkaus, 2019j), (Veikkaus, 2019k)

3.2 Urheiluvedonlyönnin teoria

Urheiluvedonlyönnin voidaan katsoa olevan yksi kaupankäynnin muoto, jossa raha vaihtaa omistajaa. Urheiluvedonlyönnissä kaupankäynnin osapuolet ovat vedonvälittäjä, joka myy todennäköisyysarvioita ottelun tapahtumista ja vedonlyöjä, joka voi halutessaan ostaa vedonvälittäjän todennäköisyysarvion tietyllä rahasummalla. Käytännössä, kuka tahansa voi olla vedonvälittäjä tai pelintarjoaja, mutta varsinaisesti termiä käytetään ihmisistä tai yrityksistä jotka tarjoavat todennäköisyysarvioita useasta eri kohteesta tarkoituksena tehdä voittoa. Jotta vedonlyöjä voi ostaa vedon, vedonvälittäjän täytyy määrätä todennäköisyysarviolle kerroin eli vedon hinta. (Vuoksenmaa, Kuronen & Nåls, 1999).

3.2.1 Kerroin ja todennäköisyysarvio

Kerroin ilmaisee, kuinka moninkertaisena pelaaja saa rahansa takaisin voittaessaan vedon. Kerroin voidaan määritellä kolmella eri tavalla, riippuen maasta ja sen kulttuurista. Lähes kaikkialla Euroopassa kerroin annetaan desimaalilukuna kahden desimaalin tarkkuudella esimerkiksi 3,00. Tämä tarkoittaa, että pelaaja voittaa kolminkertaisena rahansa takaisin. Esimerkiksi lyömällä vedonlyöntikohdetta 10 eurolla, pelaaja voittaa 30 euroa ($10 € * 3{,}00 = 30 €$). Tähän voittoon sisältyy pelaajan oma panos. Iso-Britanniassa kerroin ilmaistaan murtolukuna muodossa 2/1 tai 2-1. Tällöin veikkaamalla yhden yksikön rahaa, pelaaja saa puhdasta voittoa 2 yksikköä, eli 3 kertaa oman panostuksensa. Kolmas tapa ilmoittaa kerroin on yhdysvaltalaisten käyttämässä muodossa +150 tai -200. Kerroin +150 tarkoittaa, että pelaamalla 100 yksikköä rahaa, pelaaja saa voittaessaan 150 yksikköä takaisin. +150 vastaa eurooppalaista kerrointa 2,50. -200 puolestaan osoittaa sen, kuinka monta yksikköä rahaa pelaajan on riskeerattava voittaakseen 100 yksikköä. Eli pelaamalla 200 yksikköä pelaaja voi saada puhdasta voittoa 100 yksikköä, mikä vastaa eurooppalaista kerrointa 1,50. Tässä tutkimuksessa käytetään eurooppalaista, desimaaliluvun esitysmuotoa. (Buchdal, 2003; Vuoksenmaa ym., 1999)

Urheilutapahtuman kertoimen vedonvälittäjä määrittää tapahtuman todennäköisyysarvion perusteella, laskemalla sen käänteisluvun. Esimerkiksi jalkapallo-ottelu, jossa vastakkain ovat kotijoukkue A ja vierasjoukkue B. Tällöin ottelun lopputulokselle on kolme mahdollista tapahtumaa $\{K, T, V\}$, missä K on kotivoitto, T on tasapeli ja V on vierasvoitto. Vedonvälittäjä arvioi, että $P(K) = 0{,}5, (T) = 0{,}3\ ja\ P(V) = 0{,}2$. Nämä todennäköisyydet vedonvälittäjä muuttaa kertoimiksi ja myy ne vedonlyöntimarkkinoilla. Tässä esimerkissä kotijoukkueen A:n voiton kertoimeksi määräytyisi sen käänteisluvun mukaan 2,00, sillä $1 / 0{,}5 = 2{,}00$, joukkueiden tasapelin kertoimeksi 3,33, $sill$ä $1 / 0{,}3 = 3{,}33$ ja vierasjoukkueen B kertoimeksi 5,00, sillä $1 / 0{,}2 = 5{,}00$. Vedonlyöjät voivat joko ostaa vedonvälittäjän todennäköisyysarviot tai hylätä ne. Kertoimia vertaamalla vedonlyöjä tietää mitä hänelle tarjotaan, ja onko vetoa kohteesta järkevä lyödä. (Buchdal, 2003; Vuoksenmaa ym., 1999)

Yleensä vedonvälittäjä ei anna kertoimeksi täysin arvioimaansa todennäköisyyden käänteislukua, vaan sisällyttää kertoimeensa etumarginaalin eli voittoprosentin, turvaten sillä toimintansa jatkuvuuden. Tämä vedonvälittäjän etumarginaali on yleensä 3 - 20 %. Jos vedonlyönnin kohteena olisi esimerkiksi kolikonheitto, jossa on kaksi mahdollista tapahtumaa KR = kruuna ja KL = klaava. Todennäköisyys näille tapahtumille on $P(KL) = P(KR) = 0{,}5$, jolloin molempien tapahtumien todelliseksi kertoimeksi saataisiin 2,00. Mikäli vedonvälittäjä haluaa saada viisi prosenttia voittoa vedon järjestämisestä, hän tarjoaa molempiin tapahtumiin kertoimen 1,90, sillä $2{,}00 * 0.95 = 1{,}90$. Mikäli vedonvälittäjä haluaa saada 10 % voittoa, tarjoaa hän kerrointa 1,80, sillä $2{,}00 * 0.90 = 1{,}80$. Jos henkilö A veikkaa 10 eurolla klaavaa kertoimella 1,90 ja henkilö B veikkaa 10 eurolla kruunaa kertoimella 1,90, klaavan tullessa, vedonvälittäjä maksaa 19 euroa ($1{,}90 * 10\,€$) henkilölle A, tehden voittoa yhden euron henkilöiden A ja B vedoista ($10\,€ + 10\,€ - 19\,€ = 1\,€$). Tässä tapauksessa henkilö B menettää veikkaamansa 10 € veikatessaan väärää tulosta. Koska vedonlyöjien pelaama rahamäärä oli tässä tapauksessa yhteensä 20 €, voittoa tule vedonvälittäjälle $1 / 20 = 5$ %. (Buchdal, 2003; Vuoksenmaa ym., 1999)

Vedonvälittäjän etu eli voittomarginaali on aina pois vedonlyöjältä. Jos yllä olevaa kolikonheitto esimerkkiä toistettaisiin äärettömän monta kertaa, häviäisi vedonlyöjä odotusarvolta viisi prosenttia käyttämästään panostuksesta. Jotta vedonlyöjän olisi järkevää pelata kyseistä peliä ja voittaa pitkällä tähtäimellä rahaa pelissä, pitäisi vedonvälittäjän antaa hänelle vähintään kerroin 2,01 kruunalle tai klaavalle. Tämä olisi tietenkin haitallista vedonvälittäjälle, sillä odotusarvolta häviäisi vedonvälittäjä rahaa vedonlyöjälle. (Buchdal, 2003; Vuoksenmaa ym., 1999)

Pelin palautusprosentti ilmaisee kokonaisvaihdosta pelaajille voittoina palautettavan rahamäärän prosentteina. Palautusprosentti saadaan, kun vedonvälittäjän etu vähennetään sadasta prosentista. Palautusprosentti voidaan johtaa myös pelin kertoimesta, jolloin saadaan pelin teoreettinen palautusprosentti. Kolikonheitto pelissä kertoimella 1,90 palautusprosentti on 95 % ($100\,\% - 5\,\% = 95\,\%$) ja kertoimella 1,80 kolikonheiton palautusprosentti olisi 90 %. Tilanteessa, jossa vedonvälittäjällä ei ole etua puolellaan tai etu on vedonlyöjällä, palautusprosentti on suurempaa tai yhtä suurta kuin 100 %, eli pelaaja tulee voittamaan pelatessaan tarpeeksi monta kertaa. Esimerkiksi kolikonheitto kertoimella 2,01

antaa palautusprosentiksi 100,5 %. Tätä kutsutaan myös vedon tuoton odotusarvoksi, jonka kaava on diskreetin satunnaismuuttujan odotusarvon kaavaa $\mathrm{E(X)} = \sum_{i=i}^{n} x_i\, p_i$, missä x_i on vetokohteen todennäköisyys ja p_i vetokohteen arvo eli kerroin. Tässä tapauksessa siis 0,5 * 2,01. Mikäli lyöt vetoa 100 eurolla, voit odottaa saavasi voittoa 50 senttiä. (Buchdal, 2003; Vuoksenmaa ym., 1999; Emet, 2014)

Vedonvälittäjän etu vaihtelee suuresti riippuen pelimuodosta ja vedonvälittäjästä. On vedonlyöjän tehtävä selvittää, kuinka suuri etu pelinjärjestäjällä on suhteessa pelaajaan. Vedonvälittäjän palautusprosentti saadaan selville summaamalla yhteen kaikille eri lopputuloksille tarjottavien kertoimien käänteisluvut. Esimerkiksi vuonna 2019 Mestareiden liigan kamppailussa Real Madrid - PSG vedonlyöntitoimistot Veikkaus ja Unibet tarjosivat alla olevan taulukon mukaiset kertoimet ottelun lopputulokselle:

Taulukko 1 Unibetin ja Veikkauksen kertoimet ja todennäköisyysarviot ottelulle Real Madrid – PSG

Real Madrid - PSG	**1**	**X**	**2**
Veikkauksen kertoimet	2,15	3,85	2,85
Käänteisluvut eli todennäköisyysarvio	0,465	0,260	0,351
Unibetin kertoimet	2,25	4,00	3,05
Käänteisluvut eli todennäköisyysarvio	0,444	0,250	0,328

Veikkauksen kertoimien käänteislukujen summaksi muodostuu $0{,}465 + 0{,}260 + 0{,}351 = 1{,}076$ ja palautusprosentiksi saadaan $1/1{,}076 = 93$ %. Veikkaus laskee itselleen tästä vedonlyöntikohteesta noin seitsemän prosentin voittomarginaalin. Unibetin kertoimien käänteislukujen summa on puolestaan 1,022 ja pelin palautusprosentti n. 98 %, jolloin voittomarginaaliksi muodostuu noin kaksi prosenttia. (Cortis, 2015)

Yksi voittavan vedonlyönnin peruselementeistä on vertailla eri vedonlyöntitoimistojen tarjoamia kertoimia ja laskea mistä vedonlyöjä saa parhaimman hinnan eli kertoimen vedolleen. Pelatessa useampaa kuin yhtä kohdetta, pelinjärjestäjän voittomarginaali kertautuu. Kolmessa kohteessa joissa on sama Unibetin voittomarginaali 2 %, vedon yhteiseksi voittomarginaaliksi tulee noin kuusi prosenttia. Veikkauksen seitsemän prosentin voittomarginaalilla kolmen kohteen yhteisvoittomarginaali kasvaisi jo melkein 20 prosenttiin. (Cortis, 2015)

Urheiluvedonlyönnissä todennäköisyydet ovat arvioita joukkueiden tai pelaajien voimasuhteista, eivätkä ehdottomia todennäköisyyksiä kuten vaikka kolikonheitto tai ässän saaminen korttipakasta. jotka perustuvat klassiseen todennäköisyyteen ja ovat näin ollen symmetrisiä tapahtumia toisiinsa nähden. Näissä tapauksissa voidaan laskea tarkka todennäköisyys laskemalla suotuisten tapausten lukumäärän suhde kaikkien mahdollisten tapauksien lukumäärään, kunhan voidaan olettaa, ettei mikään tapaus ole yleisempi kuin toinen, eli että kaikki tapaukset ovat yhtä yleisiä. Nämä todennäköisyydet ovat myös kaikkien tiedossa ja laskettavissa. Urheiluvedonlyönnissä ei voida laskea tarkkaa, ehdotonta todennäköisyyttä joukkueen tai urheilijan voitolle. Tämän takia vedonvälittäjä voi tehdä virheitä arvioidessaan ottelun voimasuhteita. (Vuoksenmaa ym., 1999; Koskenoja, 2002)

Jos vedonvälittäjä on tehnyt virheen arvioidessaan joukkueen voiton todennäköisyyden alakanttiin ja tarjotessaan siksi liian suurta kerrointa, kutsutaan sitä ylikertoimeksi. Jos vedonvälittäjä arvioi joukkueen voiton todennäköisyyden yläkanttiin ja tarjoaa sen takia liian pientä kerrointa, kutsutaan sitä alikertoimeksi. Voittavan vedonlyönnin kannalta ylikertoimet ovat järkeviä pelikohteita. Esimerkiksi erittäin tasainen jalkapallo-ottelu, jonka kaikki tulosvaihtoehdot K = kotivoitto, T = tasapeli, V = vierasvoitto ovat yhtä todennäköisiä $P(K) = P(T) = P(V) = 0{,}333$. Todellisiksi kertoimiksi muodostuisivat 3,00-3,00-3,00. Vedonvälittäjä kuitenkin arvioi tulosvaihtoehtojen todennäköisyydeksi $P(K) = 0{,}45, P(T) = 0{,}3\ ja\ P(V) = 0{,}25$, jolloin kertoimiksi muodostuisi 2,22-3,33-4,00. Tämä tarkoittaisi kotivoiton olevan selkeä alikerroin, sillä vedonlyöntitoimiston kerroin 2,22 on selkeästi pienempi kuin ”todellinen kerroin” 3,00. Vierasvoitto sen sijaan osoittautuu ylikertoimeksi, ja on todennäköisyyksiin nähden järkevä pelikohde, sillä todellinen kerroin on pienempi kuin vedonlyöntitoimiston muodostama kerroin. (Vuoksenmaa, 1999)

Koska vedonlyöjä arvioi vierasvoiton todennäköisyydeksi 33 %, ei hän voi olettaa saavansa voittoa jokaisella vedolla, vaan oletuksena on vierasjoukkueen voittavan yhden kerran kolmesta. Jos oletus toteutuu ja vedonlyöjä häviää kaksi kertaa kolmesta, on veto silti kannattava, sillä kertoimella 4,00 panoksen saa takaisin nelinkertaisena. Tästä vedonlyöntivoitosta yksi yksikkö on vedon panos, kaksi yksikköä on aikaisempia hävittyjä panoksia ja yksi yksikkö puhdasta voittoa. Tämä on toinen voittavan vedonlyönnin peruselementeistä, sillä sen tarkoituksena ei ole voittaa joka ikistä vetoa, vaan löytää tilastollisesti kannattava veto. (Vuoksenmaa, 1999)

Yllä olevan esimerkin mukaiset hinnoitteluvirheet jaetaan kahteen osa-alueeseen: vedonvälittäjän tiedostamattomiin ja tietoisiin virhearvioihin. Tiedostamattomat virhearviot johtuvat pelinjärjestäjän puutteellisesta kertoimenlaskennasta. Joko hän ei ole tiennyt jotain urheilutapahtumaan vaikuttavaa tekijää (esimerkiksi maalivahdin loukkaantuminen juuri ennen ottelua), tai hän on arvioinut tekijöiden vaikutukset väärin (esimerkiksi sisäisen motivaatiotekijän väheksyminen). Tietoiset hinnoitteluvirheet perustuvat joukkopsykologiaan, jossa vedonvälittäjä käyttää hyväkseen massojen epäloogista käyttäytymistä. Tällaista toimintaa on esimerkiksi ”kansan varmojen”, kuten Manchester United tai Barcelona, kertoimien määrittäminen pienemmäksi, mitä todennäköisyysarvio antaa olettaa. Vedonvälittäjä tietää, että massat silti pelaavat näitä alikertoimia johtuen joukkueen tunnettuudesta ja suosiosta. (Vuoksenmaa, 1999)

3.2.2 Todennäköisyysarvioihin vaikuttavat tekijät

Todennäköisyysarvion määrittämisellä eli kertoimenlaskennalla tarkoitetaan kaikkia niitä menetelmiä, joilla jotakin urheilutapahtumaa koskevaa tieto muutetaan todennäköisyysarvioksi ja kertoimiksi. Parhaat kertoimenlaskijat kykenevät arvioimaan tapahtuman todennäköisyydet noin kahden prosentin tarkkuudella. Menetelmiä todennäköisyysarvion määrittämiseen on useita, vedonvälittäjillä on omansa ja vedonlyöjällä omansa. Yhteistä näille on menetelmien käyttötarkoitus, niillä pyritään arvioimaan joukkueiden tai urheilijoiden välisiä voimasuhteita. (Vuoksenmaa, 1999)

Yksi suosituimmista menetelmistä on luoda joukkueille voimasuhdeluku tai voimaluku. Tässä menetelmässä jokaiselle sarjan joukkueelle luodaan oma voimalukunsa kauden alussa kuvaamaan joukkueen pelivoimaa. Näitä voimalukuja päivitetään ja korjataan jokaisen ottelun jälkeen, mikäli vedonlyöjä on arvioinut joukkueen pelivoiman väärin. Kauden alussa voimaluvut muodostetaan lähinnä perustuen edellisvuoden sarjataulukkoon siten, että voitosta *w* joukkue saa 1 pisteen ja tasapelistä *d* 0,5 pistettä voimalukuun, joka jaetaan kaikilla otteluilla *n* ja kerrotaan sadalla silmämääräisen vertailun helpottamiseksi.

$$\frac{w * 1 + d * 0{,}5}{n} * 100$$

Tätä lukua voidaan pitää lähtökohtana joukkueen pelivoiman arviolle. Sitä kuitenkin muutetaan ottamalla huomioon joukkueeseen vaikuttaneet loukkaantumiset, satunnaistekijät, joukkueen pelaajasiirrot, joukkueen ikärakenne sekä joukkueen takana olevan organisaation taso ja taloudellinen tila. Esimerkiksi Valioliigan kauden 2018-2019 mestarijoukkueen Manchester Cityn voimaluvuksi voitaisiin määrittää sarjataulukon perusteella $32 * 1 + 2 * 0{,}5 / 38 * 100 = 86{,}8$. Tätä lukua korjattaisiin kaudelle 2019-2020. Lähteneet avainpelaajat tiputtaisivat voimalukua kahdella yksiköllä ja kesällä pitkäaikaisloukkaantumisen saaneet pelaajat tiputtaisivat vielä kolmella yksiköllä, jolloin voimaluvuksi saataisiin 81,8. Samanlainen analyysi tehtäisiin valioliigan jokaiselle joukkueelle. Jokaisen ottelun jälkeen, joukkueen voimalukua muutettaisiin nähdyn ottelun perusteella. Mikäli joukkue oli erittäin huono häviten ottelun 7-0, voimalukua voidaan pienentää. Mikäli joukkue pelasi hyvin, mutta sattuman ja huonon tuurin seurauksena hävisi silti, ei voimalukua ole järkevä muuttaa. (Vuoksenmaa, 1999; Livetulokset, 2019)

Tärkeä asia, joka täytyy ottaa huomioon voimalukuja laskiessa, on joukkueen kotietu. Kotikenttäedun suuruus vaihtelee urheilulajista ja joukkueesta riippuen. Jotkut joukkueet pelaavat yhtä hyvin vieraissa kuin kotona, jotkut eivät kykene voittamaan otteluakaan ilman kotiyleisön tukea ja ääritapauksissa joukkueet saattavat pelata paremmin vieraissa kuin kotona, mikäli kotijoukkueen fanit buuaavat kriisissä olevalla joukkueelle. Kotikenttäetuun on monia syitä, vaikka pääasiassa siihen vaikuttavat vierasjoukkueeseen ja erotuomariin kohdistuva henkinen paine katsojien osalta. Voimalukujen suhteen tämä tarkoittaa sitä, että voimaluvut olisi hyvä erottaa koti- ja vierasvoimaluvuiksi. Hyvä tapa mitata kotikenttä-

edun suuruutta on kotipisteet/kaikki pisteet -tunnusluku. Manchester Cityn tunnusluvuksi tulisi kauden 2018-2019 osalta $54/98 = 0{,}551$ eli vain noin 55 prosenttia kaikista pisteistä tuli kotiotteluista. Puolestaan hyvin kotona pelanneen Arsenalin vastaava luku oli $45/70 = 0{,}642$ eli kaikista pisteistä kotikentältä saatiin 64,2 prosenttia. (Pollard & Gómez, 2009; Vuoksenmaa, 2009; Livetulokset 2019)

Voimaluvut perustuvat ottelutilastoiden numeeriseen tilastotietoon, eli tietoon mitä voidaan mitata. Ottelun lopputulokseen vaikuttavat kuitenkin myös tekijät, joita ei voi mitata. Kertoimenlaskijan on selvitettävä, mitkä asiat vaikuttavat ottelun lopputulokseen ja kuinka paljon. Lisäksi hänen on päätettävä, mitä asioita hän ottaa huomioon kerroin arviossaan. Koska ottelukohtaista tietoa on saatavilla erittäin paljon, päätös ei ole helppoa. (Vuoksenmaa, 1999)

Tällaisia otteluun vaikuttavia tekijöitä ovat esimerkiksi henkiset tekijät, kuten joukkueiden ja pelaajien tahtotaso. Henkisiä tekijöitä ei voi mitata mittareilla, vaikka lainalaisuuksia tältäkin saralta löytyy. Tilastollisesti hyvän joukkueen ”huono päivä” ajoittuu tärkeän pelin jälkeiseen vierasotteluun, esimerkiksi kahden Mestareiden liiga ottelun väliin jäävä sarjaottelu. Syy tähän on se, etteivät edes ammattijalkapalloilijat jaksa ottaa kaikkia otteluita yhtä vakavasti. Euro-pelien rasitukset, henkinen tyhjentyminen ja alitajuinen vammojen välttäminen huonontavat suoritusta. Henkisiin tekijöihin liittyvät myös joukkueen ja pelaajien motivaatiotasot, jotka korostuvat kauden lopussa. Osa joukkueista ei enää kamppaile sarjan voitosta, mutta he eivät voi myöskään enää pudota sarjaporrasta alemmas. Tällöin heillä ei ole voitettavaa eikä hävittävää, joten heidän motivaatiotasonsa voi olla alhaalla. Putoamista vastaan taistelevat joukkueet puolestaan ovat pakkovoittojen edessä, jolloin todennäköisyysarviota on nostettava huomattavasti. Valioliigassa on tilastoitu jopa kymmenen prosenttia korkeampi voittotodennäköisyys putoajajoukkueen viimeisissä peleissä kuin sarjan keskivaiheella tapahtuvassa pelissä. (Vuoksenmaa, 1999)

Toinen vaikeasti mitattava tekijä on pelaajien loukkaantumisten vaikutus joukkueen suorituskykyyn. Jääkiekossa loistavan yksittäisen pelaajan puuttuminen ei välttämättä vaikuta joukkueen peliin yhtä paljoa kuin Amerikkalaisessa jalkapallossa pelinrakentajan puuttu-

minen. Jalkapallossa keskuspuolustajan puuttuminen voi vaikuttaa joukkueen pelaamiseen huomattavan paljon enemmän, kuin parhaimman maalintekijän puuttuminen.

Viimeinen vaikeasti mitattava tekijä on sattuma, joka vaikuttaa jokaiseen urheilutapahtumaan riippumatta lajista. Joskus suosikin sauva katkeaa hiihtokilpailun loppukirissä, joskus jalkapallo joukkue osuu tolppaan kahdeksan kertaa ottelussa ja häviää 1-0. Satunnaistekijöiden ja sattuman vaikutus vaihtelee kuitenkin lajikohtaisesti. Yksilölajeissa satunnaistekijöiden vaikutus on pienempi kuin joukkuelajeissa kuten jalkapallossa, jossa yksittäinen ottelu koostuu jo itsessään pienistä satunnaistapahtumista. Jalkapallossa taasen sattuman vaikutus lopputulokseen on suurempi kuin muissa joukkuelajeissa kuten jääkiekossa. Tämä johtuu siitä, että maalintekotilanteita ei tule yhtä paljoa kuin jääkiekossa tai koripallossa, jolloin yksi epäonnistunut laukaus tai tuomarivirhe voi muuttaa ottelun kulkua ja lopputuloksen. (Vuoksenmaa, 1999)

Kun vedonlyöjä on käsitellyt kaikki hänen mielestään tärkeät otteluun vaikuttavat tekijät, hän vertaa lopullista kerrointa vedonvälittäjien tarjoamiin lukemiin. Jos vedonlyöjä löytää etumarginaalin, ylikertoimen, hänen on järkevää panostaa otteluun.

3.2.3 Panostus

Miten pelaajan kuuluisi panostaa? Voittavaan vedonlyöntiin kuuluu kertoimenlaskentataitojen lisäksi kyky käsitellä rahaa ja panostaa oikealla tavalla ja määrällä oikeisiin kohteisiin. Panostuksen päättäminen lähtee liikkeelle pelikassan määrittämisestä. Pelikassalla tarkoitetaan sitä rahamäärää, joka on varattu vain ja ainoastaan vedonlyöntiin, eikä muuhun elämään kuten vuokraan tai ruokaan. Tämä on ensimmäinen kohta vastuullisesta pelaamisesta, mikä pitää huolen, ettei pelaamisesta muodostu vedonlyöjälle ongelma. Pelikassan määräksi suositellaan valittavan määrä, jonka voi huoletta hävitä. (Buchdal, 2003)

Tämän jälkeen päätetään miten panostetaan vedonlyöntikohteeseen. Vedonlyöjien ja sijoittajien suosiossa panoskoon määrittämiseen on amerikkalaisen matemaatikko John Larry Kelly Juniorin mukaan nimetty kaava, Kellyn kaava $\theta = \frac{p*k-1}{k-1}$, missä θ on paras mahdollinen panoskoko osuutena pelikassasta, p on pelaajan todennäköisyysarvio ja k on vedonvä-

littäjän tarjoama kerroin. Kellyn kaava kertoo, kuinka suuri osuus pelikassasta vedonlyönnissä kannattaa panostaa seuraavalle kierrokselle, jotta rahavarat karttuisivat keskimäärin mahdollisimman nopeasti. Esimerkiksi mikäli pelikassan arvo on 1000 €, pelaajan arvioima todennäköisyys p = 0,6 ja vedonvälittäjän tarjoama kerroin k = 1,8, suosittelee Kellyn kaava panokseksi $0{,}6 * 1{,}8 - 1/1{,}8 - 1 = 0.1$ eli 10 % prosenttia pelikassasta, mikä on 100 €. (Liukkonen, 2019)

Kellyn kaavassa nimittäjä $k - 1$ on voiton suhde panokseen eli tuotto pelaajan voittaessa vedon. Osoittaja on puolestaan jo aikaisemmin mainittu odotusarvon suhde panokseen eli odotettu tuotto. Kellyn kaavan mukaisella panostuksella riskiä pyritään pienentämään niin, että pelaaja vaurastuisi mahdollisimman tehokkaasti pitkällä aikavälillä, kun samaa peliä ajatellaan toistettavan loputtomiin. (Liukkonen, 2019)

Kellyn kaavaa käytettäessä kannattaa kiinnittää huomiota kahteen asiaan: Koska todennäköisyys on pelaajan itsensä arvioima, on inhimillistä, että voiton todennäköisyyttä yliarvioidaan ja sen seurauksena panokseksi tulee liian suuri osuus pelikassasta. Toinen asia mikä pelaajan täytyy muistaa, on se, että vaikka pelaaja käyttää Kellyn kaavaa oikein, ei se takaa pitkään tappioputkeen joutumista ja pelikassan huomattavaa hupenemista. Tämän takia kaavaan voidaan lisätä vakio d, jota kutsutaan Kellyn jakajaksi. Tällöin kaavaksi muodostuu $\theta = \frac{1}{d} * \frac{p*k-1}{k-1}$. Kukin pelaaja säätää jakajan d omaan riskinottoonsa sopivaksi. Suurentamalla arvoa riski pienenee, ja pienentämällä arvoa, riski kasvaa. Täytyy kuitenkin muistaa, että pienentämällä riskiä odotettu tuotto pienenee samassa suhteessa. Kellyn kaavalle ei löydy järkevää vaihtoehtoa, joka ottaisi pelaajan pelikassan ja vedon todennäköisyyden huomioon samalla tavalla. Vedonlyöjä voi kuitenkin käyttää muita strategioita kuten ruletista tutut martingaalistrategia ja fibonaccistrategia tai pelata jokaisen vedon samalla panoksella. (Liukkonen, 2019)

Martingaalistrategiassa tarkoituksena on aina häviön sattuessa tuplata panos, kunnes pelaaja voittaa, näin ollen kuitaten aikaisemmat tappiot. Strategia vaatii pelinkohteen kertoimeksi vähintään 2,00, sillä se on raja-arvo, jolla peluri kattaa aikaisemmat tappiot. Se on myös syy, minkä takia strategiaa on käytetty pitkään ruletissa. Strategian heikkoutena voidaan pitää erittäin nopeasti nousevaa panoskokoa. Häviöputken sattuessa, panos nousee kaavalla

$2^n - 1$, missä n on perättäisten häviöiden lukumäärä. Mikäli peluri häviää 15 kertaa ruletissa tai urheiluvedonlyönnissä peräjälkeen, olisi peluri hävinnyt jo siihen mennessä 32 767 euroa, minkä takia seuraavan panoksen koko täytyisi olla 32 768 euroa, jotta hän saisi kuitattua tappionsa. Fibonaccin lukujonoon perustuva strategia noudattaa samaa periaatetta, mutta asetettavat panokset noudattavat $2^n - 1$ kaavan sijasta Fibonaccin lukujonoa:

$$F_1 = 1$$
$$F_2 = 1$$
$$F_n = F_{n-2} + F_{n-1}, n \geq 2, \text{missä } n \in N$$

Fibonaccin lukujonoon perustuvassa strategiassa panoskoko ei nouse läheskään yhtä nopeasti kuin martingaalistrategiassa, mutta silti se voi syödä vedonlyöjän pelikassan lähes kokonaan. (Archontakis & Osborne, 2007)

4 Menetelmien valinta ja käsiteltävä data

Tässä kappaleessa esitellään käytettyjen menetelmien valinta sekä käsiteltävä data ja sille tehdyt toimenpiteet ennen varsinaisen tutkimuksen suorittamista. Lopuksi kappaleessa kerrotaan miten valittuja algoritmeja arvioidaan.

4.1 Datajoukon kuvaus

Tässä tutkielmassa päädyttiin käyttämään julkisesta lähteestä (football-data.co.uk) saatua dataa Englannin Valioliigasta. Kaudesta 2003/2004 lähtien, data on yhdenmukaista, sisältäen otteluiden H2H-kertoimet, joten aikaisempia kausia ei oteta mukaan käsiteltävään datajoukkoon.

Football-data.co.uk –sivusto on ilmainen jalkapallo vedonlyöntiportaali, joka tarjoaa historiallisia tuloksia ja kertoimia auttaakseen jalkapallovedonlyöntiä harrastavia henkilöitä analysoimaan monien vuosien tietoja nopeasti ja tehokkaasti. Vaikka muitakin jalkapallotietokantoja on olemassa, valikoitui football-data.co.uk:n keräämä data tutkimuksen dataksi, sillä heidän datansa on valmiiksi csv-muodossa ja ilmaiseksi saatavilla. Datajoukko sisältää myös otteluiden H2H-kertoimet, mikä helpottaa käytettävien menetelmien vedonlyöntiin soveltuvuuden tutkimusta. Football-data.co.uk-sivustolle on tilastoitu myös muiden maiden sarjojen historiadataa. Dataa löytyy yhteensä 27 eri maasta, mutta tässä tutkimuksessa keskitytään vain Englannin Valioliigaan, sillä se on maailman suurin sarjamuotoinen kilpailu.

Data on sivustolla jaoteltu yhden sarjakauden mittaisiksi kokonaisuuksiksi. Yksi csv-tiedosto sisältää siis yhden kauden (380 ottelua) ottelukohtaiset tilastot. Tähän tutkimukseen sisältyy 16 kautta, joten tutkimuksen datajoukko sisältää 6080 eri ottelun tiedot. Datajoukon ominaisuudet voidaan jakaa kolmeen eri osa-alueeseen: ottelun tunnistetiedot, ottelukohtaiset tilastot ja vedonlyöntitilastot. Yhdestä valioliiga-ottelusta on tilastoitu seuraavat tiedot:

Taulukko 2 http://football-data.co.uk/ -sivustolta saadun datan sisältämät sarakkeet ja niiden selite.

Ominaisuuden nimi	**Ominaisuuden selite**
Div	Maa ja sarjataso
Date	Päivämäärä
Time	Ajankohta
HomeTeam	Kotijoukkue
AwayTeam	Vierasjoukkue
FTHG	Kotijoukkueen maalimäärä ottelussa
FTAG	Vierasjoukkueen maalimäärä ottelussa
FTR	Ottelunlopputulos
HTHG	Kotijoukkueen maalimäärä puoliajalla
HTAG	Vierasjoukkueen maalimäärä puoliajalla
HTR	Puoliaikatulos
Referee	Tuomari
HS	Kotijoukkueen laukaukset
AS	Vierasjoukkueen laukaukset
HST	Kotijoukkueen laukaukset maalia kohti
AST	Vierasjoukkueen laukaukset maalia kohti
HF	Kotijoukkueen rikkeet

AF	Vierasjoukkueen rikkeet
HC	Kotijoukkueen kulmapotkut
AC	Vierasjoukkueen kulmapotkut
HY	Kotijoukkueen keltaiset kortit
AY	Vierasjoukkueen keltaiset kortit
HR	Kotijoukkueen punaiset kortit
AR	Vierasjoukkueen punaiset korit
B365H	Bet365:n määrittämä kotijoukkueen voiton kerroin
B365D	Bet365:n määrittämä tasapelin kerroin
B365A	Bet365:n määrittämä vierasjoukkueen voiton kerroin.
B365>2.5	Bet365:n määrittämä kerroin yli 2.5 maalia ottelussa
B365<2.5	Bet365:n määrittämä kerroin alle 2.5 maalia ottelussa

Jokaisesta ottelusta on tilastoitu myös muiden yhtiöiden määrittämiä kertoimia otteluille, mutta tässä tutkimuksessa käytetään Bet365:n määrittämiä kertoimia otteluille, jatkuvuuden ja yhdenmukaisuuden takia.

Koska dataa ei voi sellaisenaan käyttää koneoppimismenetelmiin, täytyy sille tehdä paljon muokkauksia. Sen avulla luodaan koneellisesti uusi datajoukko kuvaamaan jo pelattujen otteluiden tilastoja, jotta yhden ottelun lopputuloksen ennustaminen on helpompaa. Tämä datajoukko koostuu seuraavista sarakkeista:

Taulukko 3 Tutkimuksessa käytetyn datajoukon kuvaus.

Ominaisuuden nimi:	**Ominaisuuden selite:**
HomeTeam	Kotijoukkue
AwayTeam	Vierasjoukkue
HTP	Kotijoukkueen kauden kokonaispisteet
ATP	Vierasjoukkueen kauden kokonaispisteet
HHP	Kotijoukkueen kauden kotipisteet
AHP	Vierasjoukkueen kauden kotipisteet
HAP	Kotijoukkueen kauden vieraspisteet
AAP	Vierasjoukkueen kauden vieraspisteet
HHGS	Kotijoukkueen tehdyt maalit kotiotteluissa
AHGS	Vierasjoukkueen tehdyt maalit kotiotteluissa
HAGS	Kotijoukkueen tehdyt maalit vierasotteluissa
AAGS	Vierasjoukkueen tehdyt maalit vierasotteluissa
HHAS	Kotijoukkueen keskimääräinen maalimäärä kotiotte-lussa
AHAS	Vierasjoukkueen keskimääräinen maalimäärä kotiot-telussa
HAAS	Kotijoukkueen keskimääräinen maalimäärä vierasot-telussa

AAAS	Vierasjoukkueen keskimääräinen maalimäärä vierasottelussa
HHGC	Kotijoukkueen päästetyt maalit kotiotteluissa
AHGC	Vierasjoukkueen päästetyt maalit kotiotteluissa
HAGC	Kotijoukkueen päästetyt maalit vierasotteluissa
AAGC	Vierasjoukkueen päästetyt maalit vierasotteluissa
HHDS	Kotijoukkueen päästettyjen maalien keskiarvo kotiottelua kohden
AHDS	Vierasjoukkueen päästettyjen maalien keskiarvo kotiottelua kohden
HADS	Kotijoukkueen päästettyjen maalien keskiarvo vierasottelua kohden
AADS	Vierasjoukkueen päästettyjen maalien keskiarvo vierasottelua kohden
HHS	Kotijoukkueen laukaukset keskimäärin kotiotteluissa
AHS	Vierasjoukkueen laukaukset keskimäärin kotiotteluissa
HAS	Kotijoukkueen laukaukset keskimäärin vierasottelussa
AAS	Vierasjoukkueen laukaukset keskimäärin vierasottelussa
HHST	Kotijoukkueen laukaukset keskimäärin maalia kohti kotiottelussa

AHST	Vierasjoukkueen laukaukset keskimäärin maalia kohti kotiottelussa
HAST	Kotijoukkueen laukaukset keskimäärin maalia kohti vierasottelussa
AAST	Vierasjoukkueen laukaukset keskimäärin maalia kohti vierasottelussa
FTR	Kyseisen ottelun lopputulos
B365H	Vedonlyöntiyhtiön Bet365:n antama H2H-kerroin kotijoukkueen voitolle
B365D	Vedonlyöntiyhtiön Bet365:n antama H2H-kerroin tasapelille
B365A	Vedonlyöntiyhtiön Bet365:n antama H2H-kerroin vierasjoukkueen voitolle

Datajoukko luotiin lukemalla football-data.co.uk -sivustolta saadusta yhden kauden datasta 10 riviä (1 ottelukierros) kerrallaan sarjataulukkoon. Tämän jälkeen luotiin uusi datajoukko, johon sijoitettiin joukkueiden nimet, lopputulos ja kertoimet football-data.co.uk:n tiedoista, ja loput tiedot itse luodusta sarjataulukosta. Tämä toimenpide tehtiin kaikille yhden kauden 38 kierroksesta, ja kaikille kausille kaudesta 2003 lähtien. Jokaisen kauden 10 ensimmäisen ottelun tiedot luotuun datajoukkoon merkittiin arvoksi 0, sillä joukkueiden sarjatasojen muuttumisen takia, ei edelliskausien sarjataulukon siirtäminen seuraavaan ollut mielekästä.

Uuden datajoukon luomisessa pyrittiin käyttämään hyväksi urheiluvedonlyönnin teorian hyväksi katsottuja periaatteita. Kokonaispistesarakkeeseen joukkueelle annettiin voitosta yksi (1) piste ja tasapelistä 0,5 pistettä voimalukujen määrityksessä käytettyjen periaatteiden mukaan. Lisäksi pisteet ja pelitapahtumien tilastoinnit jaettiin erikseen koti- ja vie-

rasotteluihin, jotta vahvan koti- ja vierasvoiman joukkueet olisi mahdollista erottaa toisistaan.

Koska käytettävät koneoppimisenmenetelmät eivät osaa käsitellä muuta kuin numeerista tietoa, muutettiin joukkueiden nimet ja lopputulos (kotivoitto=H, vierasvoitto=A, tasapeli=D) numeeriseksi tiedoksi. Jokainen joukkue sai yksilöllisen id:n väliltä 1-40, jolla nimi korvattiin. Lopputulos muunnettiin siten, että vierasvoiton arvoksi annettiin kaksi (2), kotivoiton arvoksi yksi (1) ja tasapelin arvoksi nolla (0).

4.2 Toteuttavien menetelmien valinta ja käytettävät työkalut

Toteuttavat menetelmät valikoituivat pitkälti saatavilla olevan datan sekä vallitsevan ongelman perusteella. Koska datan määrä on rajallinen (otoskoko 6080), ja data on luokallista, luokitteluongelmiin hyvin sopivat algoritmit olivat itsestään selviä vaihtoehtoja. Näistä algoritmeista valittiin tilastollisen luokittelun kategoriasta yksi algoritmi (Bayesin luokittelu), instanssipohjaisesta oppimisesta yksi algoritmi (k:n lähimmän naapurin menetelmä) sekä uusin koneoppimisen tekniikka tukivektorikone, joka on lineaaripohjainen luokitinmalli. Lähtökohtana on tutkia, mikä yllä esitellyistä algoritmeista soveltuu parhaiten paljon satunnaisuutta sisältävän jalkapallo-ottelun tuloksen ennustamiseen. (Scikit-learn, 2019b)

Toinen valintaan vaikuttava tekijä oli tutkimuksessa käytetty koneoppimisen kirjasto scikit-learn, joka antoi valmiit työkalut ja ohjeet kaikkien yllä olevien algoritmien toteuttamiselle.

Menetelmät toteutettiin ja niihin liittyvät laskennat suoritettiin JetBrains PyCharm 2019.1.2-versiolla. Tietokoneen käyttöjärjestelmänä oli Windows 7 Professional, koneessa oli Intel® Core™ i7-3632QM CPU @2.20GHz sekä 8 GB keskusmuistia.

4.3 Menetelmien toimivuuden arviointi

Tässä tutkimuksessa menetelmien toimivuutta sekä luotettavuutta arvioidaan neljän eri luokan avulla: *TN (*true negative), *TP* (true positive, *FN* (false negative) ja *FP* (false positive). Näiden avulla voidaan määrittää järjestelmän käyttökelpoisuus, sillä mitä vähemmän

virheellisiä luokitteluja algoritmit tekevät, sitä parempi se on. Näiden luokkien avulla voidaan laskea varsinaiset mittarit oikeiden negatiivisten osuus (*engl. true negative rate, TNR),* oikeiden positiivisten osuus (*engl. true positive rate, TPR, recall),* väärien positiivisten osuus (*engl. false positive rate, FPR*), tarkkuus (*engl. precision),* osuvuus (*engl. accuracy)* ja F-mitta (*engl. F-measure*) seuraavilla tavoilla. (Witten & Frank, 2005; Tharwat, 2018)

$$TNR = \frac{TN}{TN + FP}$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

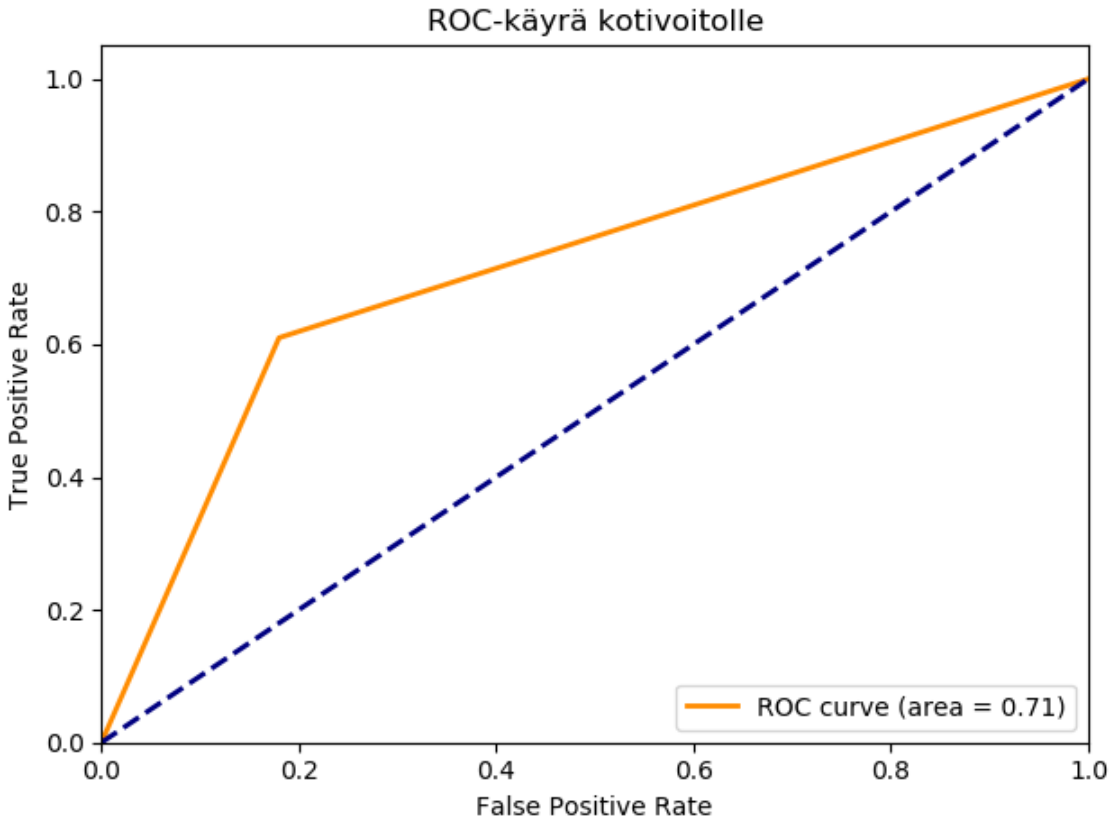
$$precision = \frac{TP}{TP + FP}$$

$$accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

$$F - measure = 2 * \frac{TPR * precision}{TPR + precision}$$

Jatkossa näihin mittareihin viitataan niiden lyhenteillä ja englannin kielisillä nimillä.

Viimeinen mittari, jota tutkimuksessa käytetään, on ROC-käyrä (*engl. Receiver operating characteristic*), jonka avulla saadaan graafinen esitystapa FPR- ja TPR-mittareille. Tähän kaavioon sijoitetaan x-akselille FPR ja y-akselille TPR (Kuva 3). ROC-käyrä skaalataan koordinaatistoon koordinaattipisteiden [0,0] ja [1,1] välille. Tärkeää tietoa ROC-käyrässä on sen alle jäävä osuus (AUC, *engl. Area Under the Curve*). Mikäli AUC on 0.5, on algoritmilla yhtä suuri tarkkuus kuin arvaamalla. Mikäli AUC on 1, algoritmin tarkkuus on täydellinen.



Kuva 3 MNB menetelmän ROC-käyrä kotivoitolle.

5 Tukivektorikoneet

Tässä kappaleessa tarkastellaan ohjatun koneoppimisen algoritmin, tukivektorikoneen *(engl. support vector machines, SVM)* toimintaa. Sitä käytetään lineaaristen ennustajien arvioimiseen korkean dimensionaalisuuden tila-avaruudessa. Korkea dimensionaalisuus kasvattaa sekä datan monimutkaisuutta, että laskennallista monimutkaisuutta. SVM algoritmi hoitaa datan monimutkaisuuden etsimällä ”suuren marginaalin” erottimia. (Shalev-Shwartz ym., 2014)

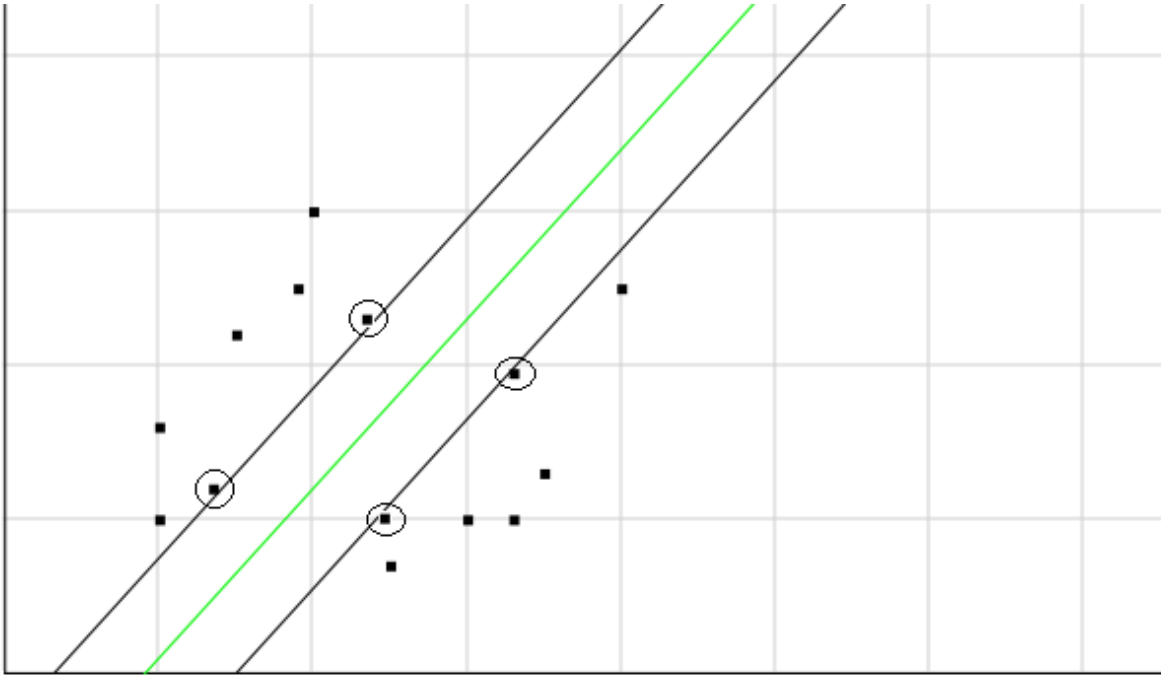
5.1 Menetelmän kuvaus

Tukivektorikoneita on kahdenlaisia, riippuen käytössä olevasta datasta: kova ja pehmeä SVM. Kovaa eli lineaarista SVM algoritmia käytetään silloin, kun käytössä oleva datassa on kaksi luokkaa, jotka ovat lineaarisesti separoituvia. Tällöin SVM pyrkii löytämään separoituvalle datajoukolle hypertason *(engl. hyperplane)*, jossa luokat toisistaan erottava marginaali on mahdollisimman suuri. Suuren marginaalin valitseminen datan hypertasolle maksimoi SVM:n kyvyn ennustaa luokka aikaisemmin tuntemattomille esimerkeille. Esimerkiksi olkoon opetusdata $S = (x_1, y_1), \ldots, (x_m, y_m)$, missä jokainen $x_i \in \mathrm{R}^d$ ja $y_i \in \{\pm 1\}$. Jos data lineaarisesti separoituva, on olemassa välitaso (w, b), siten että

$$\forall i \in [m], y_i\big((w,\, x_i) + b\big) > 0.$$

Kaikki välitasot, jotka täyttävät tämän ehdon, ovat empiirisen riskin minimoijia (ERM-hypoteesi), eli niiden 0-1 virhe on nolla, mikä on pienin mahdollinen virhe. Kuvassa 4 on esitetty SVM algoritmin toiminta lineaarisesti separoituvalle datalle. Siinä hypertasoa kuvaa vihreä suora $W * X + b = 0$, missä vektori $W = \{w_1, w_2, \ldots, w_n\}$ ilmaisee n:n painotukset, ja b on korjauskerroin. Suurinta mahdollista marginaalia kahden luokan välillä mustat suorat $W * X + b = -1$ ja $W * X + b = 1$. Kun optimaalinen hypertaso on löydetty (lineaarisesti separoituvan datan tapauksessa), datapisteet, jotka sijaitsevat marginaalin reunalla, muodostavat lineaarisen kombinaation marginaalista. Näitä datapisteitä kutsutaan tukivektoripisteiksi (Kuva 4). Tämän takia SVM:n mallin monimutkaisuuteen ei vaikuta harjoitusdatassa olevien ominaisuuksien määrä (SVM-algoritmin valitsemien tukivektori-

pisteiden lukumäärä on yleensä pieni). Tästä syystä tukivektorikoneet soveltuvat hyvin tehtäviin, joissa ominaisuuksien määrä on suuri suhteessa harjoitusinstanssien määrään. (Noble, 2006; Shalev-Shwartz ym., 2014; Han, J. & Kamber, M., 2006; Kotsiantis, 2007)



Kuva 4 SVM:n hypertaso ja marginaalit lineaarisesti separoituvalle datalle

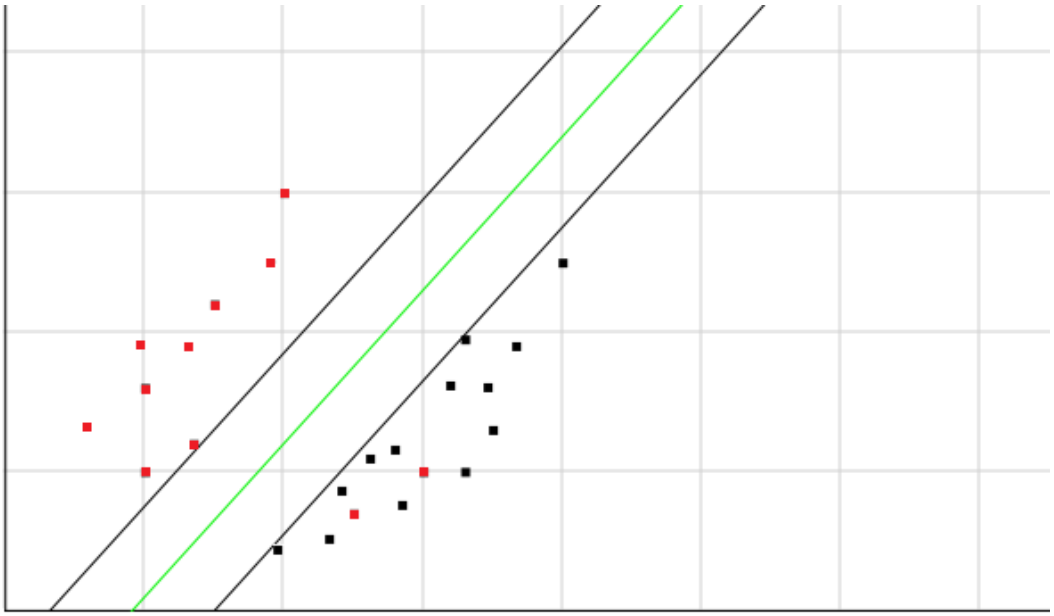
Kuitenkin usein reaalimaailmassa, käytössä oleva data ei ole lineaarisesti separoituvaa. Tällöin täytyy ottaa käyttöön virheen käsittely, joka mahdollistaa algoritmin toimimisen myös ei-separoituvalla datalla, jossa luokan datapisteitä voi olla myös toisella puolella hypertasoa (Kuva 5). Virheen korjauksen myötä, algoritmi jättää huomiotta osan datapisteistä, ”virhearviot”, määrittääkseen hypertason. Tätä tasoa kutsutaan pehmeäksi marginaaliksi. Koska SVM:n luokittelu ei kuitenkaan saa sisältää liikaa vääriä luokitteluja, pehmeään marginaaliin täytyy sisällyttää parametri, joka hallitsee karkeasti, kuinka monen datapisteen sallitaan rikkovan hypertasoa ja kuinka kaukana tasosta ne saavat olla. Parametrin määrittäminen on monimutkaista, koska marginaalien täytyy silti olla mahdollisimman suuria, jotta luokittelu onnistuu tarkasti. Parametrin ξ lisäämisen jälkeen, jolloin SVM toimii seuraavalla tavalla: Algoritmi etsii $W \in R^2, b \in R$ ja $\xi_i, i = 1,2, \dots, l$, minimoidakseen $\left(\frac{1}{l}\right)\left(\sum_{i=1}^{l} \xi_i\right)^q + \lambda\|W\|^2$, seuraavin rajoittein

$$W * X + b \geq 1 - \xi_i, \text{ missä } y_i = 1;$$

$$W * X + b \leq 1 - \xi_i, \text{ missä } y_i = -1;$$

$$\xi_i \geq 0, \forall i.$$

Tässä λ on käyttäjän valitsema parametri ja q on positiivinen kokonaisluku. (Noble, 2006; Lin, Lee, & Wahba, 2002)



Kuva 5 SVM:n hypertaso ja marginaalit ei-separoituvalle datalle.

Toinen ratkaisu ei-separoituvan datan ongelmaan on kuvata data korkeamman dimension ulottuvuudessa ja määrittää hypertaso siellä. Tätä korkeamman dimensionaalisuuden avaruutta kutsutaan muunnetuksi tila-avaruudeksi (engl. *transformed feature space*) ja koko toimenpidettä pehmeäksi tai epälineaariseksi tukivektorikoneeksi. Oikein valittu muunnettu tila-avaruus, jolla on riittävän korkea dimensionaalisuus, mahdollistaa minkä tahansa harjoitusdatan lineaarisen separoitumisen. Näin ollen lineaarinen datan separoituminen muunnetussa tila-avaruudessa vastaa ei-separoituvaa dataa alkuperäisessä tila-avaruudessa. (Kotsiantis, 2007)

Laskennallisesti datapisteiden kuvaaminen korkean dimensionaalisuuden omaavassa tila-avaruudessa ei olisi mahdollista ilman ydinfunktioita (*engl. kernel function*). Kun data ku-

vataan toisessa Hilbertin tila-avaruudessa H, kun $\Phi : R^d \rightarrow H$, tarvitsee harjoitusalgoritmi vain H tila-avaruudessa kuvatut datan pisteet, toisin sanoen funktion $\Phi(x_i) * \Phi(x_j)$. Mikäli on olemassa ydinfunktio K, niin että $K(x_i, x_j) = \Phi(x_i) * \Phi(x_j)$, tarvittaisiin vain käyttää funktiota K harjoitusalgoritmissa, eikä Φ tarvittaisiin ikinä määrittää. Ydinfunktion tarkoituksena on siis laskea sisätuotteet suoraan tila-avaruudessa, suorittamatta yllä kuvattua kartoitusta. Kun hypertaso on määritetty, ydinfunktiota käytetään kuvaamaan jälleen uudet pisteet alkuperäiseen tila-avaruuteen luokittelua varten. (Kotsiantis, 2007)

Oikean ydinfunktion valitseminen on erittäin tärkeää, sillä ydinfunktio määrittää muunnetun tila-avaruuden, jossa harjoitusdatan instanssit luokitellaan. Kotsiantisin mukaan, yleinen käytäntö on arvioida potentiaalisesti parhaimpia asetuksia, jonka jälkeen harjoitusjoukkoon käytetään ristiinvalidointia parhaan ydinfunktion löytämiseksi. Tämän takia tukivektorikoneet ovat hitaampia harjoitteluvaiheessa kuin monet muut algoritmit. Hyvää tukivektorikoneessa on se, että niin kauan kuin ydinfunktio on määritelty oikein, toimii myös SVM oikein, vaikka käyttäjä ei tietäisikään tarkalleen mitä ominaisuuksia harjoitusdatasta käytetään ydinfunktion käyttämässä muunnellussa tila-avaruudessa. (Kotsiantis, 2007)

Tukivektorikoneille määriteltyjä ydinfunktioita on useita kuten polynomi ydinfunktio $K(x, x') = (x * x' + 1)^P$, hyperbolinen tangenttiydinfunktio $K(x, x') = \tanh(kx * x' - \delta)^P$ ja radiaalipohjainen ydinfunktio (*RBF*) eli Gaussianin ydinfunktio $K(x, x') = e^{-\gamma\|x-x'\|^2}$, missä γ on käyttäjän määrittämä parametri $\gamma > 0$. (Shalev-Shwartz ym., 2014; skicit-learn, 2019a; Kotsiantis, 2007)

5.2 Hyperparametrien etsintä

Jotta joustavan marginaalin tukivektoriluokitinta (C-SVC) voidaan käyttää, täytyy määritellä hyperparametrien arvot C ja γ sekä päättää käytettävä ydinfunktio. Hyperparametrien etsintä tehtiin käyttäen hyväksi scikit-learn kirjaston *GridSearchCV*-objektia, jonka avulla parametrit optimoidaan ristiinvalidoidulla ruudukkohaulla parametririvien yli. Optimointi

tehtiin etsimällä paras testitulos eri C:n arvoilla, kolmelle eri ydinfunktiolle. Taulukossa 4 on kuvattu C:n arvot, käytetty ydinfunktio sekä testitulos, jonka pohjalta valinta tehdään.

Taulukko 4 GridSearchCV:n tulokset hyperparametrin C ja ydinfunktion etsimiselle

C	**Kernel function**	**Mean test score**
10	Linear	0.56155735
10	RBF	0.46474921
10	Polynomial	0.47106278
1	Linear	0.5634865
1	RBF	0.4640477
1	Polynomial	0.46895826
0.1	Linear	0.56611715
0.1	RBF	0.46229393
0.1	Polynomial	0.47737636
0.01	Linear	0.57348299
0.01	RBF	0.46229393
0.01	Polynomial	0.4847422
0.001	Linear	0.57962119
0.001	RBF	0.4622939
0.001	Polynomial	0.51841459
0.0001	Linear	0.57471063

0.0001	RBF	0.46229393
0.0001	Polynomial	0.53016485

GridsearchCV -ominaisuuden käyttäminen vei erittäin kauan aikaa, sillä hyperparametrien optimointiin kului aikaa 13 tuntia ja 42 minuuttia. Kuten yllä olevasta taulukosta nähdään, paras tulos saatiin lineaarisella ydinfunktiolla, silloin kun C:n arvo on 0.001 tuloksella 0.579. Gamman γ parasta arvoa ei tarvitse määritellä, koska parhaaksi ydinfunktioksi saatiin lineaarinen ydinfunktio ja gamman γ määrittäminen ei kuulu sen ominaisuuksiin. Testissä huomattiin, että mitä suuremman arvon C saa, sitä kauemmin algoritmilla kestää.

5.3 Menetelmän arviointi testausdatalla

Parametrien muodostamisen jälkeen, muodostettiin algoritmi testausdatan instansseille käyttämällä Scikit-learn kirjaston *svm.SVC*-algoritmia, joka tarkoittaa joustavan marginaalin käyttöä tukivektorikoneessa (*engl. C-support vector classification*). Tämän algoritmin hyperparametriksi C annettiin yllä saatu tulos 0.001 ja ydinfunktioksi määrättiin lineaarinen ydinfunktio. Ajankäytöllisesti, SVC-algoritmi suoriutui opetusdatan käsittelystä verrattain hitaasti. *SVC*:n *fit*-funktion käyttämä aika algoritmin harjoittamiseen mitattiin olevan 1.661 sekuntia ja ennustamiseen käytetyn *predict*-funktion käytetty aika 0.06 sekuntia.

Taulukossa 5 esitetään matriisimuodossa datajoukon instanssien ennustetut ja todelliset ryhmät. Taulukon riveillä on esitetty todelliset ryhmät ja sarakkeilla ennustetut ryhmät. Taulukossa diagonaalisesti esitetyt tummennetut arvot ovat oikeiden luokittelujen lukumääriä. Tulokset saatiin käyttämällä scikit-learn kirjaston *confusion_matrix* –ominaisuutta.

Taulukko 5 SVM:n Instanssien ennustetut ja todelliset ryhmät.

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**154**	26	0
	Vierasvoitto	33	**96**	0
	Tasapeli	51	19	**1**

Algoritmin tarkkuutta arvioitiin kappaleessa neljä esiteltyjen mittareiden mukaan. Tulokset näkyvät taulukossa 6.

Taulukko 6 SVM-menetelmän arviointi testausdatalla.

	FPR	**TNR**	**Support**	**TPR**	**Precision**	**F-measure**	**AUC**
Kotivoitto	0.42	0.58	180	0.86	0.65	0.74	0.72
Vierasvoitto	0.18	0.82	129	0.74	0.68	0.71	0.78
Tasapeli	0.00	1.00	71	0.01	1.00	0.03	0.51

Algoritmin osasi ennustaa siis 66.053 % kaikista otteluista. Mikäli pelaaja olisi panostanut euron jokaiseen kohteeseen, olisi hän saanut puhdasta voittoa kaudesta yhteensä 321.43€.

6 k:n lähimmän naapurin menetelmä

Tässä kappaleessa käsitellään k:n lähimmän naapurin menetelmää (*engl. k Nearest Neighbor, k-NN*) ja sen käyttötarkoitusta. K-NN on insanssipohjainen menetelmä eli laiska oppimisalgoritmi (*engl. lazy learning algorithm*). Ne siis viivästyttävät induktio- tai yleistämisprosessia, kunnes kohteen luokittelu on suoritettu. Laiskat koneoppimisen algoritmit tarvitsevat vähemmän laskennallista tehoa harjoitusdatan käsittelyyn kuin innokkaat algoritmit kuten päätöspuut tai neuroverkot, mutta ne tarvitsevat enemmän laskennallista aikaa, eli ne ovat hitaampia. (Kotsiantis, 2007)

6.1 Algoritmin kuvaus

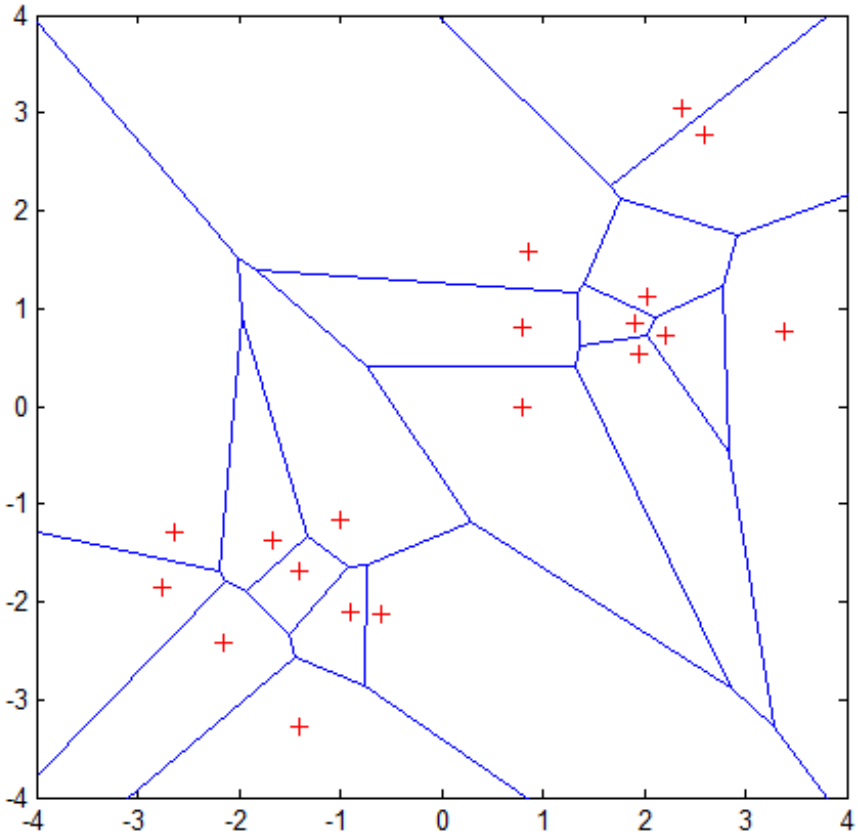
Lähimmän naapurin menetelmät ovat yksinkertaisia algoritmeja, joiden ideana on muistaa harjoitusdata, ja sitä hyödyntämällä ennustaa uuden instanssin kohdeluokka, perustuen lähimpiin naapureihin harjoitusdatassa. Lähimmän naapurin menetelmä perustuu oletukseen, että ominaisuudet, joiden avulla kuvataan datapisteitä, vaikuttavat ennustettavaan luokkaan siten, että lähimpänä olevilla datapisteillä olisi sama luokka kuin ennustettavalla datapisteellä. K-NN-menetelmää käytetään pääasiassa luokittelu- ja regressio-ongelmien ratkaisemiseen. Mikäli avaruus K on diskreetti ja rajallinen, kyseessä on luokitteluongelma ja mikäli kyseessä on jatkuva avaruus K, kyseessä on regressio-ongelma. (Shalev-Shwartz ym., 2014; Mohammed, ym., 2017)

K-NN luokittelu perustuu pääasiassa testidatan ja harjoitteludatan väliseen euklidiseen etäisyyteen. Olkoon x_i syöte data, jolla on ominaisuudet $p\ (x_{i1}, x_{i2}, \ldots, x_{ip})$, n on syöte datan lukumäärä $(i = 1, 2, \ldots, n)$ ja p ominaisuuksien lukumäärä $(j = 1, 2, \ldots, p)$. Euklidinen etäisyys x_i:n ja x_l:n $(l = 1, 2, \ldots, n)$ välillä määritellään

$$d(x_i, x_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \ldots + (x_{ip} - x_{lp})^2}$$

Graafinen kuvaus lähimmän naapurin menetelmästä on kuvattu kuvassa 6 Voronoin diagrammina. Diagrammiin on merkitty 19 otosta (punaiset ”+” –merkit), sekä Voronoin solu, R, joka ympyröi jokaisen otoksen. Voronoin solu koteloi kaikki naapurin pisteet, jotka ovat

lähimpänä otosta. $R_i = \{x \in R^p : d(x, x_i) \leq d(x, x_m), \forall\, i \neq m\}$, missä R_i on Voronoin solu otokselle x_i ja x edustaa kaikkia pisteitä Voronoin solun R_i sisällä. Voronoin diagrammi kuvaa hyvin kahta kNN-ominaisuutta koordinaatistossa: 1) kaikki mahdolliset pisteet Voronoin solun sisällä ovat kyseisen otoksen lähimpiä naapureita, 2) jokaiselle otokselle, lähin naapuri määritetään Voronoin solun reunan etäisyyden perusteella. (Peterson, 2009)



Kuva 6 Voronoin diagrammi kNN-menetelmästä 19 syötetapauksella, kun $k = 1$ (Peterson, 2009).

Yleistettynä, instanssit voidaan ajatella pisteinä n-dimension instanssiavaruudessa, missä jokainen n:n dimensio vastaa yhtä n:n ominaisuutta, joka kuvaa instanssia. Pisteiden absoluuttinen sijainti avaruudessa ei ole yhtä merkittävää kuin pisteiden välinen suhteellinen etäisyys. Tämä suhteellinen etäisyys määritetään käyttämällä etäisyysmittaria. Ideaalisesti

etäisyysmittari minimoi kahden samanlaisesti luokitellun instanssin, ja maksimoi kaksi eritavalla luokiteltua instanssia. (Kotsiantis, 2007)

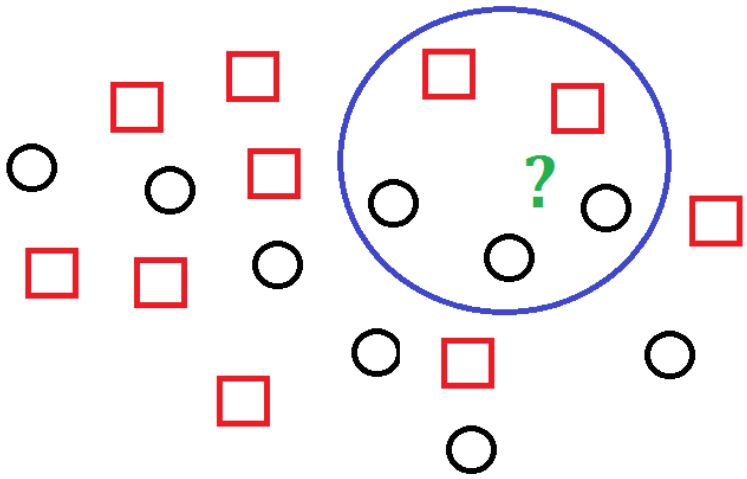
Yllä olevassa esimerkissä etäisyyden mittaamiseen käytettiin Euklidista etäisyyttä, sillä se on suosituin kahden pisteen välisen etäisyyden mittari, mutta myös muita etäisyysmittareita voidaan käyttää. Eri etäisyysmittarit on lueteltu taulukossa 7. (Kotsiantis, 2007)

Taulukko 7 Eri etäisyysmittarit kNN-algoritmille

Minkowskin etäisyys:	$D(x,y) = (\sum_{i=1}^{m} \lvert x_i - y_i \rvert^r)^{1/r}$
Manhattanin etäisyys:	$D(x,y) = \sum_{i=1}^{m} \lvert x_i - y_i \rvert$
Chebychevin etäisyys:	$D(x,y) = \max_{i=1}^{m} \lvert x_i - y_i \rvert$
Camberran etäisyys:	$D(x,y) = \sum_{i=1}^{m} \frac{\lvert x_i - y_i \rvert}{\lvert x_i + y_i \rvert}$
Kendallin järjestyskorrelaatiokerroin:	$D(x,y) = 1 - \frac{2}{m(m-1)} * \sum_{i=j}^{m} \sum_{j=1}^{i-1} sign(x_i - x_j) sign(y_i - y_j)$

Kun uuden datapisteen etäisyys muihin datajoukon naapureihin on mitattu halutulla tavalla, täytyy uusi datapiste luokitella, mikäli kyseessä on luokitteluongelma, tai sen arvo ennustaa, mikäli kyseessä on regressio-ongelma. Luokittelu tapahtuu määrittämällä k. Algoritmin k, kuvaa sitä, kuinka monta lähintä solua otetaan mukaan ennustettavan muuttujan arvioon, katso kuva 7. Näistä syötedatan lähimmistä arvoista, algoritmi laskee keski-arvon ennustettavalle muuttujalle, ja määrää arvon sen perusteella. Sekä regression, että luokitte-

lun tapauksessa, voi olla hyödyllistä käyttää painotettuja muuttujia keskiarvon laskemisessa, niin että lähimmät naapurit kontribuoivat enemmän keskiarvoon kuin kaukana sijaitsevat naapurit. Yksi yleisimmistä painotuksista on tutkitun datapisteen ja naapurin välisen etäisyyden d käänteisluvun $\frac{1}{d}$ käyttäminen keskiarvossa pelkän etäisyyden d tilalla. (Peterson, 2009; Mohammed ym., 2017)



Kuva 7 k:n lähimmän naapurin menetelmä, kun $k = 5$.

K:n optimaalinen valinta riippuu suuriltaosin saatavilla olevasta datasta. Suuret k:n arvot vähentävät luokittelussa syntyvää kohinaa, mutta tekevät samalla luokkien välisistä rajoista epäselkeät. K voidaan määrittää käyttämällä erilaisia heuristisia menetelmiä, mutta käytännössä, k on yleensä pariton luku, jotta tasatuloksia rajojen pituuksien välille ei synny. (Peterson, 2009)

Yksinkertainen versio kNN menetelmästä saadaan kun asetetaan $K = 1$. Tätä sääntöä kutsutaan yleisesti lähimmän naapurin luokittelusäännöksi tai 1-NN säännöksi: $h_S(x) = y_{\pi 1(x)}$. Toisin sanoen, vektori x luokitellaan sen lähimpään naapuriin. Mikäli harjoitusdata on riittävän suuri, 1-NN säännön avulla saadaan hyvä tarkkuus luokittelulle. Theodoridisin ja Konstantinoksen mukaan, kun $N \to \infty$, luokitteluvirheen todennäköisyys 1-NN säännölle, P_{NN}, rajoittuu $P_B \leq P_{NN} \leq P_B\left(2 - \frac{M}{M-1}P_B\right) \leq 2P_B$, missä P_B on optimaalinen Bayesin virhe. 1-NN luokittelun tekemä virhe on siten asymptoottisesti korkeintaan kaksi kertaa

suurempi kuin optimaalisen luokittelijan virhe. (Peterson, 2009; Theodoridis & Konstantinos, 2006)

Koska k-NN ei ole parametrinen algoritmi, se ei tee johtopäätöksiä taustalla olevista datajakaumista, vaan datan mallin määrää pelkästään saatavilla oleva data (toisin kuin esimerkiksi lineaarisessa regressiossa). Tämän takia k-NN on erityisen hyvä epälineaarisen datan mallintamisessa, eikä sen käyttämisellä tule niin suuria ongelmia ylisovittamisen kanssa. Ongelmia muodostuu kuitenkin laskentatehon ja aikavaativuuden osalta. K-NN on tehoa ja aikaa vaativa algoritmi, sillä naapurin etsimiseen käytetyn ajan aikavaativuudeksi saadaan Theodoridisin ja Konstantinosin mukaan $kN(O(kN))^2$. (Theodoridis, ym., 2006)

6.2 Menetelmän arviointi testausdatalla

K-NN menetelmä toteutettiin scikit-learn kirjaston funktiolla *KNeighborsClassifier*. Tällä algoritmille annettiin opetusdataa 5700 instanssia, ja 380 instanssia testausdataa, kuten *Datan esikäsittely* -kappaleessa kuvattiin.

Algoritmin käyttämien hyperparametrien määrittämisessä käytettiin algoritmin oletusarvoja tai datan pohjalta optimoituja arvoja. Algoritmin etäisyysmittarina käytettiin kirjaston oletusarvoa eli Eukliidista etäisyyttä ja jokaiselle datapisteelle määriteltiin sama painotus riippumatta datapisteiden ja testauspisteiden välisestä etäisyydestä. Pisteiden väliseen etäisyyteen käytetyn algoritmin kirjasto määritti automaattisesti parhaimman tuloksen mukaan kolmikosta *BallTree, KDTree* ja raa'an voiman algoritmi (*engl. Brute Force*).

Algoritmin valintaan vaikuttavat monet eri asiat. Tällä hetkellä algoritmi valitsee raa'an voiman algoritmin, jos $k \geq N/2$, syötetty data on harvaa eli sisältää paljon puuttuvia arvoja tai käytetty etäisyyden mittari ei sovi kahdelle muulle algoritmille. Muutoin se käyttää joko *BallTree*- tai *KDTree*-algoritmia, riippuen algoritmin hyväksytystä etäisyyden mittarista. (Sklearn, 2019d)

Optimaalinen k:n arvo määritettiin välille 1-50 käyttäen hyväksi scikit-learn kirjaston *GridSearchCV*-ominaisuutta, jonka avulla k:n arvo optimoidaan ristiinvalidoidulla ruuduk-

kohaulla parametriirivin yli. Taulukossa 8 on listattu testissä käytetyt k:n arvot sekä *GridSearchCV*:n tulokset. Parhaimmaksi tulokseksi saatiin $k = 47$ testituloksella 0.5507.

Taulukko 8 tulokset k:n eri arvoilla.

k	Tulos	k	Tulos	k	Tulos	k	Tulos
1	0.42603643	14	0.5254397	27	0.54082915	40	0.54663945
2	0.47707286	15	0.52873744	28	0.54334171	41	0.54679648
3	0.46451005	16	0.53282035	29	0.54491206	42	0.5450691
4	0.48900754	17	0.53517588	30	0.54569724	43	0.54381281
5	0.48979271	18	0.53815955	31	0.54444095	44	0.54648241
6	0.50125628	19	0.53674623	32	0.54648241	45	0.54805276
7	0.50738065	20	0.53753141	33	0.54601131	46	0.54962312
8	0.51256281	21	0.53800251	34	0.54444095	47	0.55072236
9	0.50957915	22	0.53941583	35	0.54679648	48	0.55072236
10	0.51460427	23	0.54098618	36	0.54585427	49	0.55009422
11	0.51680276	24	0.54271357	37	0.54711055		
12	0.52198492	25	0.5428706	38	0.54632538		
13	0.52465452	26	0.54585427	39	0.54569724		

Parametrien muodostamisen jälkeen, muodostettiin algoritmi testausdatan instansseille käyttämällä Scikit-learn kirjaston *KNeighborsClassifieria*. Ajankäytöllisesti, kNN-algoritmi suoriutuu opetusdatan käsittelystä erittäin nopeasti. *KNeighborsClassifierin fit-*

funktion käyttämä aika algoritmin harjoitukseen mitattiin olevan vain 0.140 sekuntia ja ennustamiseen käytetyn *predict*-funktion käytetty aika vain 0.252 sekuntia.

Taulukossa 9 esitetään matriisimuodossa datajoukon instanssien ennustetut ja todelliset ryhmät. Taulukon riveillä on esitetty todelliset ryhmät ja sarakkeilla ennustetut ryhmät. Taulukossa diagonaalisesti esitetyt tummennetut arvot ovat oikeiden luokittelujen lukumääriä. Tulokset saatiin käyttämällä scikit-learn kirjaston *confusion_matrix* –ominaisuutta.

Taulukko 9 kNN: Instanssien ennustetut ja todelliset ryhmät.

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**153**	26	1
	Vierasvoitto	40	**87**	2
	Tasapeli	47	20	**4**

Algoritmin tarkkuutta arvioitiin kappaleessa neljä esiteltyjen mittareiden mukaan. Tulokset näkyvät taulukossa 10.

Taulukko 10 kNN-menetelmän arviointi testausdatalla.

	FPR	**TNR**	**Support**	**TPR**	**Precision**	**F-measure**	**AUC**
Kotivoitto	0.44	0.57	180	0.85	0.64	0.73	0.71
Vierasvoitto	0.18	0.82	129	0.68	0.65	0.67	0.75
Tasapeli	0.01	0.99	71	0.06	0.57	0.10	0.52

Algoritmin osasi ennustaa siis 64.211 % kaikista otteluista. Mikäli pelaaja olisi panostanut euron jokaiseen kohteeseen, olisi hän saanut puhdasta voittoa kaudesta yhteensä 303.51€.

7 Naivi Bayesin luokitin

Tässä kappaleessa käsitellään ohjatun koneoppimisen menetelmää: naiivia Bayesin luokitinta. Se on yksinkertainen, todennäköisyyspohjainen luokitin, joka perustuu Bayesin teoreemaan, olettaen että ominaisuuksien joukossa on vahvaa (naiivia) riippumattomuutta. Huolimatta epärealistisesta lähtöoletuksesta, naiivi Bayesin luokitin on osoittautunut tehokkaaksi luokittimeksi käytännön sovelluksiin, kuten tekstin luokitteluun, sairauden diagnosointiin ja järjestelmän suorituskyvyn hallintaan. (Mohammed, ym., 2017; Rish, 2001)

7.1 Menetelmän kuvaus

Bayesin ehdollisen todennäköisyyden kaava, johon menetelmä perustuu, on muotoa:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)},$$

missä A ja B ovat joitain tapahtumia, $P(A)$ ja $P(B)$ ovat tapahtumien riippumattomat priori-todennäköisyydet, $P(A|B)$ on A:n todennäköisyys ehdolla B (posteritodennäköisyys, *engl. posterior probability*) ja todennäköisyys $P(B|A)$ on B:n todennäköisyys ehdolla A, jota kutsutaan uskottavuudeksi (*engl. likelihood*). Olkoon vektori $X = (x_1, x_2, \dots, x_n)$ instanssi (riippumattomilla ominaisuuksilla n) joka luokitellaan, ja c_j on yksi K:n luokista. Bayesin teoreemaa käyttämällä voidaan laskea posteritodennäköisyys $P(c_j|X)$ käyttäen hyväksi todennäköisyyksiä $P(c_j), P(X)$ ja $P(X|c_j)$. (Mohammed, ym., 2017)

Todennäköisyyden $P(c_j|X)$ laskeminen on kuitenkin vaikeaa, jos harjoitusdata on korkeadimensionaalista. Tämän takia naiivin Bayesin luokittelijan täytyy tehdä yksinkertaisen (naiivi) oletus, jota kutsutaan luokan ehdolliseksi riippumattomuudeksi, että ennustajan x_i arvon vaikutus annettuun luokkaan c_j on riippumaton muiden ennustajien arvoista. Tällöin voidaan laskea jokaiselle K:n luokalle todennäköisyys $P(c_j|X)$, missä $\prod_{j=1}^{K}$. X:n instanssi määrätään luokalle c_j, jos ja vain jos $P(c_k|X) > P(c_j|X)$, kun $1 \leq j \leq K, j \neq k$. Toisin sanoen tämä maksimoi arvon $P(c_k|X)$. Arvoa c_k, joka on maksimoitu, kutsutaan maksi-

maaliseksi posteriori hypoteesiksi (*engl. maximum posteriori hypothesis*). (Mohammed, ym., 2017; Rish, 2001; Han, 2006)

Otetaan esimerkiksi taulukossa 11 kuvattu aineisto säätilalle ja golfin pelaamiselle:

Taulukko 11 Data-aineisto kuvaamaan milloin pelataan golfia.

Näkymä	**Lämpötila**	**Kosteus**	**Tuulisuus**	**Pelataan**
Aurinkoinen	Kuuma	Korkea	Epätosi	Ei
Aurinkoinen	Kuuma	Korkea	Tosi	Ei
Pilvinen	Kuuma	Korkea	Epätosi	Kyllä
Sateinen	Lämmin	Korkea	Epätosi	Kyllä
Sateinen	Viileä	Normaali	Epätosi	Kyllä
Sateinen	Viileä	Normaali	Tosi	Ei
Pilvinen	Viileä	Normaali	Tosi	Kyllä
Aurinkoinen	Lämmin	Korkea	Epätosi	Ei
Aurinkoinen	Viileä	Normaali	Epätosi	Kyllä
Sateinen	Lämmin	Normaali	Epätosi	Kyllä
Aurinkoinen	Lämmin	Normaali	Tosi	Kyllä
Pilvinen	Lämmin	Korkea	Tosi	Kyllä
Pilvinen	Kuuma	Normaali	Epätosi	Kyllä
Sateinen	Lämmin	Korkea	Tosi	Ei

Tehtävänä on ennustaa datan eri ominaisuuksien avulla, pelataanko golfia vai ei. Merkitään golfin pelaamisen todennäköisyyttä $P(G) = 9 / 14$ ja todennäköisyyttä, että golfia ei pe-

lata $P(K) = 5 / 14$. Olkoon X (näkymä = pilvinen, lämpötila = lämmin, kosteus = normaali, tuulisuus = epätosi) uusi tapahtuma. Tällöin kaavan mukaan täytyy laskea ehdolliset todennäköisyydet, jotka on esitelty alla olevassa taulukossa.

Taulukko 12 Ehdolliset todennäköisyydet (uskottavuus) säätiloille, kun golfin pelaaminen on tosi tai epätosi.

Näkymä	
$P(Aurinkoinen\|Pelataan = Kyllä) = 2/9$	$P(Aurinkoinen\|Pelataan = Ei) = 3/5$
$P(Pilvinen\|Pelataan = Kyllä) = 4/9$	$P(Pilvinen\|Pelataan = Ei) = 0$
$P(Sateinen\|Pelataan = Kyllä) = 3/9$	$P(Sateinen\|Pelataan = Ei) = 2/5$
Lämpötila	
$P(Kuuma\|Pelataan = Kyllä) = 2/9$	$P(Kuuma\|Pelataan = Ei) = 2/5$
$P(Lämmin\|Pelataan = Kyllä) = 4/9$	$P(Lämmin\|Pelataan = Ei) = 2/5$
$P(Viileä\|Pelataan = Kyllä) = 3/9$	$P(Viileä\|Pelataan = Ei) = 1/5$
Kosteus	
$P(Korkea\|Pelataan = Kyllä) = 3/9$	$P(Korkea\|Pelataan = Ei) = 4/5$
$P(Normaali\|Pelataan = Kyllä) = 6/9$	$P(Normaali\|Pelataan = Ei) = 1/5$
Tuulisuus	
$P(Tosi\|Pelataan = Kyllä) = 3/9$	$P(Tosi\|Pelataan = Ei) = 3/5$
$P(Epätosi\|Pelataan = Kyllä) = 6/9$	$P(Epätosi\|Pelataan = Ei) = 2/5$

Yllä olevan taulukon todennäköisyyksien mukaan, voidaan määrittää uuden tapahtuman X kuuluminen jompaankumpaan luokkaan:

1. $P(X|Pelataan = Kyllä)$
2. $P(X|Pelataan = Ei)$

Todennäköisyydet saadaan määritettyä seuraavalla laskutoimituksella:

$P(X \mid Pelataan = Kyllä) = P(Näkymä = Pilvinen \mid Pelataan = Kyllä)$
$* P(Lämpötila = Lämmin \mid Pelataan = Kyllä)$
$* P(Kosteus = Normaali \mid Pelataan = Kyllä)$
$* P(Tuulisuus = Epätosi \mid Pelataan = Kyllä)$

$P(X \mid Pelataan = Ei) = P(Näkymä = Pilvinen \mid Pelataan = Ei)$
$* P(Lämpötila = Lämmin \mid Pelataan = Ei)$
$* P(Kosteus = Normaali \mid Pelataan = Ei)$
$* P(Tuulisuus = Epätosi \mid Pelataan = Ei)$

Naiivin Bayesin luokittelua tehdessä ja todennäköisyyksiä laskiessa voi tulla vastaan tilanne, jossa prioritodennäköisyyden osoittajaksi tulee arvoksi 0 (eli dataan ei ole merkitty tilannetta, vaikka se olisi todennäköinen), mikä määrittää koko tapahtuman todennäköisyyden nollaksi. Ylläolevassa esimerkissä tilanne tulee vastaan kohdassa $P(Näkymä = Pilvinen \mid Pelataan = Ei)$, minkä takia todennäköisyydeksi tulee 0. Tätä varten luokittelussa käytetään apuna Laplacen estimointia (*engl. Laplace estimator*), joka korjaa todennäköisyyden. Oletetaan käytössä olevan harjoitusdata olevan niin suuri, että lisäämällä yhden arvon jokaiseen prioritodennäköisyyteen, olisi muutos merkityksetön arvioitavien prioritodennäköisyyksien kannalta ja se silti korjaisi nolla-todennäköisyyden. Mikäli jokaiseen k luokkaan lisätään 1, täytyy muistaa lisätä arvo k vastaavaan nimittäjään, jota käytetään todennäköisyyden laskemisessa. (Han, 2006)

Uudet prioritodennäköisyydet Laplacen estimaattorin kanssa olisivat:

$$P(Pelataan = Kyllä) = (9 + 1) / (14 + 2) = 10 / 16$$

$$P(Pelataan = Ei) = (5 + 1) / (14 + 2) = 6 / 16$$

Uudet ehdolliset todennäköisyydet Laplacen estimaattorin kanssa olisivat alla olevan taulukon mukaiset:

Taulukko 13 Uudet ehdolliset todennäköisyydet Laplacen estimoinnin jälkeen.

Näkymä	
$P(Aurinkoinen \mid Pelataan = Kyllä) = 3/12$	$P(Aurinkoinen \mid Pelataan = Ei) = 4/8$
$P(Pilvinen \mid Pelataan = Kyllä) = 5/12$	$P(Pilvinen \mid Pelataan = Ei) = 1/8$
$P(Sateinen \mid Pelataan = Kyllä) = 4/12$	$P(Sateinen \mid Pelataan = Ei) = 3/8$
Lämpötila	
$P(Kuuma \mid Pelataan = Kyllä) = 3/12$	$P(Kuuma \mid Pelataan = Ei) = 3/8$
$P(Lämmin \mid Pelataan = Kyllä) = 5/12$	$P(Lämmin \mid Pelataan = Ei) = 3/8$
$P(Viileä \mid Pelataan = Kyllä) = 4/12$	$P(Viileä \mid Pelataan = Ei) = 2/8$
Kosteus	
$P(Korkea \mid Pelataan = Kyllä) = 4/11$	$P(Korkea \mid Pelataan = Ei) = 5/7$
$P(Normaali \mid Pelataan = Kyllä) = 7/11$	$P(Normaali \mid Pelataan = Ei) = 2/7$
Tuulisuus	
$P(Tosi \mid Pelataan = Kyllä) = 4/11$	$P(Tosi \mid Pelataan = Ei) = 4/7$
$P(Epätosi \mid Pelataan = Kyllä) = 7/11$	$P(Epätosi \mid Pelataan = Ei) = 3/7$

Ehdolliset todennäköisyydet $P(X \mid Pelataan = Kyllä)$ ja $P(X \mid Pelataan = Ei)$ voidaan nyt laskea helposti:

$P(X \mid Pelataan = Kyllä) = P(Näkymä = Pilvinen \mid Pelataan = Kyllä)$
$* P(Lämpötila = Lämmin \mid Pelataan = Kyllä)$

$* P(Kosteus = Normaali \,|Pelataan = Kyllä)$
$* P(Tuulisuus = Epätosi \mid Pelataan = Kyllä)$
$= 5/12 * 5/12 * 7/11 * 7/11 = 0.070305$

$P(X \mid Pelataan = Ei) = P(Näkymä = Pilvinen \mid Pelataan = Ei)$
$* P(Lämpötila = Lämmin \mid Pelataan = Ei)$
$* P(Kosteus = Normaali \,|Pelataan = Ei)$
$* P(Tuulisuus = Epätosi \mid Pelataan = Ei)$
$= 1/8 * 3/8 * 2/7 * 3/7 = 0.00574$

Ehdollisen todennäköisyyden laskemisen jälkeen, voidaan laskea luokille posterioritodennäköisyys Bayesin kaavaa noudattamalla. Tässä esimerkissä kyseessä on luokitteluongelmasta, vertaillaan Bayesin kaavasta saatuja todennäköisyyksiä toisiinsa, ja suuremman todennäköisyyden saanut arvo merkitään tapahtuman X arvoksi. Koska molempien posterioritodennäköisyyksien nimittäjä on sama, riittää vertailu tuloille:

$P(Pelataan = Kyllä \mid X) = P(Pelataan = Kyllä) * P(X|Pelataan = Kyllä) = 10/16 * 0.070305 = 0.043941.$

$P(Pelataan = Ei \mid X) = P(Pelataan = Ei) * P(X|Pelataan = Ei) = 6/16 * 0.00574 = 0.002152.$

Koska golfin pelaamisen todennäköisyys on suurempi kuin golfin pelaamattomuus, merkitään uuden tapauksen X luokan *Pelataan* arvoksi ”Kyllä”.

Naiivin Bayesin luokittelussa, pyritään määrittämään myös luokitteluvirhe. Luokitteluvirheen todennäköisyys eli luokittelijan h riski määritellään seuraavasti:

$R(h) = P\big(h(X) \neq g(X)\big) = \sum_{x\in\Omega} P\big(h(x) \neq g(x)\big)P(X = x) = E_x\{P\big(h(x) \neq g(x)\big)\},$
missä $E_x$odotus x:n suhteen ja $R^* = R(h^*)$ ilmaisee Bayesin virheen (Bayesin riski). Luokittelija h on optimaalinen silloin kun sen riski vastaa Bayesin riskiä. Koko naiivin Bayesin luokittelijan tarkoituksena on minimoida luokittelun odotettu riski ja maksimoida uskottavuusfunktion suurin uskottavuus (*engl. maximun likelihood estimation*). (Rish, 2001; Shalev-Shwartz ym., 2014)

7.2 Menetelmän arviointi testausdatalla

Naiivi Bayesin menetelmä toteutettiin eri tavalla verrattuna kNN-algoritmiin, sillä hyperparametrien optimoinnin suorittaa algoritmi itse, käyttäen hyväksi suurimman todennäköisyyden menetelmää. Käyttäjän päätettäväksi jää, mitä Bayesin algoritmia käytetään: Gaussin Naiivi Bayesin luokitin, Multinomi Naiivi Bayesin luokitin (*engl. Multinomial Naive Bayes)*, Komplementti Naiivi Bayesin luokitin (*engl. Complement Naive Bayes)* vai Bernoullin Naiivi Bayesin luokitin. Tässä tutkielmassa ei käsitellä Bernoullin luokittimen soveltuvuutta, sillä sitä käytetään dataan, joka jakautuu Bernoullin monimuuttuja jakaumien mukaan, toisin sanoen, voi olla useita muuttujia, mutta ne ovat jakautuneet binäärisesti. (Scikit-learn, 2019c)

7.2.1 Gaussian Naiivi Bayes luokitin

Gaussin naiivi Bayesin (GNB) luokitin olettaa ominaisuuksien todennäköisyyksien olevan Gaussianaalisia:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}}\exp(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2})$$

missä parametrit $\sigma_y$ja μ_y optimoidaan suurimman todennäköisyyden mukaan. GNB luokitinta käytetään yleisesti sen yksinkertaisuuden ja nopeuden takia. (Scikit-learn, 2019c; Lou, W., Wang, X., Chen, F., Chen, Y., Jiang, B. & Zhang, H., 2014)

GNB algoritmi muodostettiin käyttäen hyväksi scikit-learn kirjaston *GaussianNB()*-ominaisuutta. Ajankäytöllisesti algoritmi oli nopeampi kuin lähimmän naapurin luokitin tai tukivektorikoneet, sillä GNB:llä kesti vain 0.006 sekuntia opetukseen ja 0.001 sekuntia datasta ennustamiseen. Taulukossa 14 esitetään GNB:n luokittimen tulokset sekaannusmatriisina.

Taulukko 14 Gaussin Naiivi Bayesin luokitin: instanssit ja todelliset ryhmät.

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**95**	28	57
	Vierasvoitto	12	**81**	36
	Tasapeli	21	16	**34**

Algoritmin tarkkuutta arvioitiin kappaleessa neljä esiteltyjen mittareiden mukaan. Tulokset näkyvät taulukossa 15.

Taulukko 15 GNB-menetelmän arviointi testausdatalla.

	FPR	**TNR**	**Support**	**TPR**	**Precision**	**F-measure**	**AUC**
Kotivoitto	0.17	0.84	180	0.53	0.74	0.62	0.68
Vierasvoitto	0.18	0.82	129	0.63	0.65	0.64	0.73
Tasapeli	0.30	0.70	71	0.48	0.27	0.34	0.59

Algoritmin osasi ennustaa siis 55.263 % kaikista otteluista. Mikäli pelaaja olisi panostanut euron jokaiseen kohteeseen, olisi hän saanut puhdasta voittoa kaudesta yhteensä 245.77€.

7.2.2 Multinomi Naiivi Bayesin luokitin

Multinomi naiivi Bayesin (MNB) luokitin käyttää hyväkseen multinomisesti jakautunutta dataa. Jakauma parametrisoidaan vektoreiden $\theta_y = (\theta_{yi}, \dots, \theta_{yn})$ jokaiselle luokalle y, missä n on ominaisuuksien lukumäärä ja θ_{yi} on ominaisuuden i näkymisen todennäköisyys $P(x_i|y)$ luokassa y. θ_y parametrit estimoidaan siloitetulla suurimman todennäköisyyden periaatteella, relatiivisella frekvenssilaskennalla $\hat{\theta}_{yi} = \frac{N_{yi}+\alpha}{N_y+\alpha n}$, missä $N_{yi} = \sum_{x \in T} x_i$ kuvaa lukumäärää, jona ominaisuus i ilmaantuu luokassa y, harjoitusdatassa T ja $N_y = \sum_{i=1}^{n} N_{yi}$ on y:n ominaisuuksien summa. Määrittämällä tasoituspriorin α:n arvoksi > 0, estetään nollalla jakamisen mahdollisuus. Asettamalla $\alpha = 1$, saadaan edellisessä luvussa esitetty *Laplacen estimaattori*. (Scikit-learn, 2019c)

MNB algoritmi muodostettiin käyttäen hyväksi scikit-learn kirjaston *MultinomialNB()*-ominaisuutta. Ajankäytöllisesti MNB oli tähänastisista menetelmistä nopein, sillä MNB:llä kesti vain 0.005 sekuntia opetukseen ja 0.0 sekuntia datasta ennustamiseen. Taulukossa 16 esitetään MNB:n luokittimen tulokset sekaannusmatriisina.

Taulukko 16 MNB-luokitin: instanssit ja todelliset ryhmät.

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**121**	43	16
	Vierasvoitto	19	**92**	18
	Tasapeli	30	27	**14**

Algoritmin tarkkuutta arvioitiin kappaleessa neljä esiteltyjen mittareiden mukaan. Tulokset näkyvät taulukossa 17. Verrattuna GNB:hen, MNB vaikuttaa sopivammalta vaihtoehdolta tarkasteltavien tunnuslukujen perusteella.

Taulukko 17 MNB-menetelmän arviointi testausdatalla.

	FPR	TNR	Support	TPR	Precision	F-measure	AUC
Kotivoitto	0.25	0.75	180	0.67	0.71	0.69	0.71
Vierasvoitto	0.28	0.72	129	0.71	0.57	0.63	0.72
Tasapeli	0.11	0.89	71	0.20	0.29	0.24	0.54

Algoritmin osasi ennustaa siis 59.737 % kaikista otteluista. Mikäli pelaaja olisi panostanut euron jokaiseen kohteeseen, olisi hän saanut puhdasta voittoa kaudesta yhteensä 263.69€.

7.2.3 Komplementti Naiivi Bayesin luokitin

Komplementti naiivi Bayesin luokitin (CNB) on standardin MNB algoritmin muunnos. Se sopii erityisen hyvin epäsymmetriselle datalle, sillä CNB:n ideana on käyttää kunkin luokan komplementin tilastoja laskemaan mallin parametrien painotukset. CNB algoritmin luojat Jason Rennie, Lawrence Shih, Jaime Teevan ja David Karger osoittavat empiirisellä tutkimuksella, että parametrien estimointi CNB:lle on vakaampaa kuin MNB:lle ja että CNB voittaa MNB:n kaikissa tekstin luokittelu tehtävissä. (Rennie, J., Shih, L., Teevan, J. & Karger, D, 2003)

Painotuksen laskeminen tapahtuu seuraavalla tavalla:

$$\hat{\theta}_{ci} = \frac{\alpha_i + \sum_{j:y_j \neq c} d_{ij}}{\alpha + \sum_{j:y_j \neq c} \sum_k d_{kj}}$$

$$w_{ci} = \log \theta_{ci}$$

$$w_{ci} = \frac{w_{ci}}{\sum_j |w_{cj}|}$$

missä summat ovat kaikki dokumentit j, jotka eivät ole luokassa c, d_{ij} on joko i:n lukumäärä tai tf-idf arvo dokumentissa j, a_i on tasoituksen hyperparametri kuten MNB:ssä ja $\alpha = \sum_i a_i$. Toinen normalisointi kohdistuu ongelmaan, joka muodostuu pidempien asiakirjojen hallitsemaan parametri-estimointiin MNB:ssä. Luokittelusääntö CNB:ssä on: $\hat{c} = arg\ \min_c \sum_i t_i w_{ci}$, eli käytännössä dokumentti määritellään kuuluvaksi luokkaan, jolla on kehnoin komplementti yhteensopivuus. (Rennie, J. & ym., 2003)

CNB algoritmi muodostettiin käyttäen hyväksi scikit-learn kirjaston *ComplementNB()*-ominaisuutta. Ajankäytöllisesti CNB yhtä nopea kuin MNB, sillä kesti yhtä kauan opetuksessa (0.005 sekuntia) ja yhtä kauan ennustamisessa (0.0 sekuntia). Se oli kuitenkin tarkempi kuin tähän mennessä käsitellyistä naiivi Bayesin luokittimista tarkin, kuten taulukko 18 esittää.

Taulukko 18 CNB-luokitin: instanssit ja todelliset ryhmät

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**132**	47	1
	Vierasvoitto	25	**104**	0
	Tasapeli	37	33	**1**

CNB saavutti parhaimman tuloksen sillä, että se keskittyi ennustamaan vain koti- ja vierasvoittoja. Sillä kuten taulukosta 18 nähdään, ei malli ennustanut pelattavan kuin yhden tasapelin. Mallin tarkkuutta arvioitiin kappaleessa neljä esiteltyjen mittareiden mukaan. Tulokset näkyvät taulukossa 19.

Taulukko 19 CNB-menetelmän arviointi testausdatalla.

	FPR	**TNR**	**Support**	**TPR**	**Precision**	**F-measure**	**AUC**
Kotivoitto	0.31	0.69	180	0.73	0.68	0.71	0.71
Vierasvoitto	0.32	0.68	129	0.81	0.57	0.66	0.74
Tasapeli	0.00	1.00	71	0.01	0.50	0.03	0.51

Algoritmin osasi ennustaa siis 62.368 % kaikista otteluista. Mikäli pelaaja olisi panostanut euron jokaiseen kohteeseen, olisi hän voittanut kauden aikana yhteensä 285.67€.

8 Yhteenveto

Tässä tutkielmassa vertailtiin kolmea eri koneoppimisen menetelmää jalkapallovedonlyönnin lopputuloksen ennustamiseen. Parhaiten luokittelussa onnistui (tarkkuutta arvioidessa) joustavan marginaalin tukivektoriluokitin. Sillä saavutettiin korkein tarkkuus 66.053 % ja sillä olisi tehty eniten puhdasta voittoa. Se ei kuitenkaan ennustanut kuin yhden ottelun päättyvän tasapeliin, mikä kertoo tasapelien satunnaisuuden määrästä ja niiden vaikeasta ennustettavuudesta. Käyttäessä tarkkuutta arvioinnin mittarina, toiseksi sijoittui k:n lähimmän naapurin menetelmä 64.211 %:n tarkkuudella ja kolmanneksi komplementti naiivi Bayesin menetelmä 62.368 %:n tarkkuudella.

Jos arvioidaan algoritmeja väärien positiivisten (FPR) ja oikeiden negatiivisten (TNR) osuuksilla, ennusti Gaussin naiivi Bayesin luokitin parhaiten kotivoittoja. Sillä oli pienin FPR 0.17 ja suurin TNR 0.84. Parhaiten vierasvoittoja ennustivat SVM, kNN ja GNB, joilla kaikilla FPR oli 0.18 ja TNR 0.82. Tasapelejä suurimmalla todennäköisyydellä oikein ennusti SVM ja CNB, jotka molemmat ennustivat yhden tasapelin oikein, antaen FPR arvoksi näin ollen 0.00 ja TNR arvoksi tasan 1.00.

Mielestäni tulokset olivat odotetun kaltaisia. Jalkapallo-ottelu sisältää niin paljon satunnaisuutta ja inhimillisiä virheitä, että täydellisesti ennustavan koneen luominen on mahdotonta. Algoritmeja pitäisi testata pienemmän satunnaisuuden joukkueurheilulajeissa, kuten koripallossa, missä ”maalinteko” tilanteita tulee paljon enemmän kuin jalkapallossa, eikä yksi virhe välttämättä ratkaise koko ottelua. Aihe vaatii myös jalkapallovedonlyönnin sisällä lisää tutkimista. Mikäli käytössä on paremmin joukkueen hyökkäys- ja puolustusvoimaa kuvaavaa dataa, olisi mahdollista saada parempia tuloksia aikaan, kuin mitä tässä tutkimuksessa on saatu. Myös irrallisen ”kuntopuntarin” lisäämisellä dataan, voisi olla positiivista vaikutusta algoritmien toimintaan. Toisen lajin testaamisen lisäksi, pitäisi algoritmeja testata ennustamaan joukkueen yhden ottelun maalimäärää, koska niiden avulla saisi testattua muitakin pelimuotoja, kuin pelkät joukkueiden H2H-pelimuodot.

Ohjatun koneoppimisen algoritmit kuitenkin osoittivat käsittelevänsä dataa paremmin kuin ihmiset. Yhdistettynä tämä ihmisen kykyyn käsitellä ja arvioida ei-mitattavaa tietoa kuten motivaatiota, uskon, että koneoppimismenetelmistä on erittäin paljon hyötyä vedonlyöjälle.

Lähteet

Statista. (2019a). Sports Betting and Gambling Market/Industry - Statistics & Facts. Haettu 10.11.2019 osoitteesta https://www.statista.com/topics/1740/sports-betting/

Statista. (2019b). Size of the online gambling market in 2017 and 2024. Haettu 10.11.2019 osoitteesta https://www.statista.com/statistics/270728/market-volume-of-online-gaming-worldwide/

Veikkaus. (2018). Veikkaus Oy Vuosiraportti 2018. Haettu 10.11.2019 osoitteesta https://cms.veikkaus.fi/site/binaries/content/assets/dokumentit/vuosikertomus/2018/veikkaus_vuosiraportti_2018.pdf

Veikkaus. (2019a). Tiesitkö tämän vakiosta. Haettu 10.11.2019 osoitteesta https://www.veikkaus.fi/fi/vakio#!/ohjeet/tiesitko-taman-vakiosta

Veikkaus. (2008, 16. lokakuuta). UUTISKIRJE 6/2008 - 16.10.2008. Haettu 10.11.2019 osoitteesta https://www.veikkaus.fi/info/viestit/uutiskirje08_06.html

Veikkaus. (2017). Historia. Haettu 10.11.2019 osoitteesta https://www.veikkaus.fi/fi/yritys#!/yritystietoa/historia

McCorduck, P., Minsky, M. & Simon, H. A. (1977). History of Artificial intelligence. Teoksessa Proceedings of the Fifth International Joint Conference on Artificial Intelligence (II)

Buchanan, B. (2006). A (Very) Brief History of Artificial Intelligence. *AI Magazine Volume 26*.

McCorduk, P. (2004). Machines Who Think A Personal Inquiry into the History and Prospects of Artificial Intelligence. Natick, Massachusetts: A K Peters, Ltd.

Jääskeläinen, A. (2019). Mitä tapahtuu huomenna, kun tekoäly poistaa järjettömyydet? WSOY.

Copeland, M. (2016, 29. Heinäkuuta). What's the Difference Between Artificial Intelligence, Machine Learning and Deep Learning. Haettu 10.11.2019 osoitteesta https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Helsingin yliopisto & Reaktor (2019). Elements of AI. https://course.elementsofai.com/fi/1/1

Gartner, Inc. (2019). Information Technology Gartner Glossary Artificial Intelligence (ai). Haettu 10.11.2019 osoitteesta https://www.gartner.com/en/information-technology/glossary/artificial-intelligence

Raskulla, S. (2019). Suomen tekoälyohjelman 2017–2019 eettiset ulottuvuudet. Politiikka-Lehti, 61(3), 247-259.

Siukkonen, T. & Neittaanmäki, P. (2019) *Mitä tulisi tietää tekoälystä.* Jyväskylä: Docendo.

Goertzel, B. (2014). Artificial General Intelligence: Concept, State of the Art, and Future Prospects. *Journal of Artificial General Intelligence 5(1) 1-46, 2014.*

Vähäkainu, P. & Neittaanmäki, P. (2018). Tekoäly terveydenhuollossa. *Informaatioteknologian tiedekunnan julkaisuja No. 45/2018.*

Gottfredson, L. (1997) Mainstream Science on Intelligence: An Editorial With 52 Signatories, History, and Bibliography. *Wall Street Journal, INTELLIGENCE 24(I) 13-23.*

Shalev-Shwartz, S. & Ben-David, S. (2014) Understanding Machine Learning: From Theory To Algorithms. New York: Cambridge University Press.

Mohammed, M., Khan, M. B. & Bashier, E. B. M. (2017). Machine Learning: Algorithms and Applications. London: CRC Press.

Buchdal, J. (2003) Fixed Odds Sports Betting: The Essential Guide. London: High Stakes Publishing.

Vuoksenmaa, J., Kuronen, A. & Nåls, J. (1999). Urheiluvedonlyönti: Voittajan opas. Jyväskylä: Gummerus Kirjapaino Oy.

Kaarakka, T. (2003). Sattuman matematiikkaa II – todennäköisyyslaskennan aksioomat. Matematiikkalehti Solmu, 1/2003.

Koskenoja, M. (2002). Sattuman matematiikkaa I – klassinen todennäköisyys. Matematiikkalehti Solmu, 2/2002.

Veikkaus. (2019b). Pitkäveto lyhyesti. Haettu 3.12.2019 kohteesta https://www.veikkaus.fi/fi/pitkaveto?sportId=1&selectedLeagues=1-all#!/ohjeet/pitkaveto-lyhyesti

Veikkaus. (2019c). Tulosveto lyhyesti. Haettu 3.2.2019 kohteesta https://www.veikkaus.fi/fi/tulosveto#!/ohjeet/tulosveto-lyhyesti

Veikkaus. (2019d). Tulosveto peliohjeet. Haettu 3.12.2019 kohteesta https://www.veikkaus.fi/fi/tulosveto#!/ohjeet/peliohjeet

Veikkaus. (2019e). Moniveto lyhyesti. Haettu 3.12.2019 kohteesta https://www.veikkaus.fi/fi/moniveto#!/ohjeet

Veikkaus. (2019f). Moniveto peliohjeet. Haettu 3.12.2019 kohteesta https://www.veikkaus.fi/fi/moniveto#!/ohjeet/peliohjeet

Veikkaus. (2019g). Vakio lyhyesti. Haettu 3.12.2019 kohteesta https://www.veikkaus.fi/fi/vakio#!/ohjeet/vakio-lyhyesti

Veikkaus. (2019h). Vakio peliohjeet. Haettu 3.12.2019 kohteesta https://www.veikkaus.fi/fi/vakio#!/ohjeet/peliohjeet

Veikkaus. (2019i). Voittajavedot lyhyesti. Haettu 3.12.2019 kohteesta https://www.veikkaus.fi/fi/voittajaveto#!/ohjeet

Veikkaus. (2019j). Live-veto lyhyesti. Haettu 3.12.2019 kohteesta https://www.veikkaus.fi/fi/live-veto?sportId=1&selectedLeagues=1-all#!/ohjeet

Veikkaus. (2019k). Tiesitkö tämän Live-vedosta. Haettu 3.12.2019 kohteesta https://www.veikkaus.fi/fi/live-veto?sportId=1&selectedLeagues=1-all#!/ohjeet/tiesitko-taman-live-vedosta

Cortis, D. (2015). Expected Values And Variances In Bookmaker Payouts: A Theoretical Approach Towards Setting Limits On Odds. The Journal of Prediction Markets 9-1

Pollard, R. & Gómez, M. (2009). Home advantage in football in South-West Europe: Long-term trends, regional variation, and team differences. *European Journal of Sport Science 9/6*.

Livetulokset. (2019). Valioliiga 2018/2019 tulokset, sarjataulukot. Haettu 11.12.2019 kohteesta https://www.livetulokset.com/jalkapallo/englanti/valioliiga-2018-2019/.

Emet, S. (2014). Johdatus todennäköisyyslaskentaan ja tilastotieteeseen. *Matematiikan ja tilastotieteen lts Turun yliopist.*

Liukkonen, J. (2019) Kellyn kaava. *Matematiikkalehti Solmu, 3/2019.*

Archontakis, F. & Osborne, E. (2007). Playing It Safe? A Fibonacci Strategy for Soccer Betting. *JOURNAL OF SPORTS ECONOMICS, 8/3*

Chapelle, O.,Schölkopf, B. & Zien, A. (2006). Semi-Supervised Learning. *The MIT Press Cambridge, Massachusetts, London*

Bhavsar, H. & Ganatra, A. (2012). A Comparative Study of Training Algorithms for Supervised Machine Learning. *International Journal of Soft Computing and Engineering, 2/4.*

Noble, W. (2006). What is a support vector machine? *Nature Biotechnology 24/12.*

Han, J. & Kamber, M. (2006). Data mining: Concepts and techniques. *Morgan Kaufmann Publishers, San Francisco.*

Lin, Y. Lee, Y. & Wahba, G. (2002) Support Vector Machines for Classification in Non-standard Situations. *Kluwer Academic Publishers*.

Scikit-learn. (2019a). sklearn.svm.LinearSVC. Haettu 20.1.2020 kohteesta https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC

Peterson, L. (2009) K-nearest neighbor. *Scholarpedia, 4/2.*

Theodoridis, S. & Konstantinos, K. (2006). Pattern Recognition. *Academic press, London*.

Rish, I. (2001). An empirical study of the naive Bayes classifier. *TJ Watson Research Ceter.*

Kotsiantis, S. (2007). Supervised Machine Learning: A Review of Classification Techniques. *University of Peloponnese, Greece.*

Scikit-learn. (2019b) API Reference. Haettu 2.3.2020 kohteesta https://scikit-learn.org/stable/modules/classes.html

Witten, I. H., & Eibe, F. (2005). Data mining: Practical machine learning tools and techniques. Morgan Kaufmann Series in data management systems.

Tharwat, A. (2018). Classification assessment methods. *Faculty of Computer Science and Engineering, Frankfurt University of Applied Sciences.*

Scikit-learn. (2019c). Naive Bayes. Haettu 2.3.2020. Kohteesta https://scikit-learn.org/stable/modules/naive_bayes.html

Lou, W., Wang, X., Chen, F., Chen, Y., Jiang, B. & Zhang, H. (2014). Sequence Based Prediction of DNA-Binding Proteins Based on Hybrid Feature Selection Using Random Forest and Gaussian Naive Bayes. *School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, PR China.*

Rennie, J., Shih, L., Teevan, J. & Karger, D. (2003) Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *Artificial Intelligence Laboratory; Massachusetts Institute of Technology; Cambridge*

Scikit-learn. (2019d). Nearest Neighbors. Haettu 2.3.2020. Kohteesta https://scikit-learn.org/stable/modules/neighbors.html#classification

Liitteet

A Käsiteltävä data

Taulukko 20 sisältää http://football-data.co.uk/ -sivustolta ladatun datan sarakkeiden otsikot sekä niiden selitteen. Tästä datasta luotiin uusi datajoukko, jota käytettiin tutkimuksessa.

Taulukko 20 http://football-data.co.uk/ -sivustolta saadun datan sisältämät sarakkeet ja niiden selite

Ominaisuuden nimi	**Ominaisuuden selite**
Div	Maa ja sarjataso
Date	Päivämäärä
Time	Ajankohta
HomeTeam	Kotijoukkue
AwayTeam	Vierasjoukkue
FTHG	Kotijoukkueen maalimäärä ottelussa
FTAG	Vierasjoukkueen maalimäärä ottelussa
FTR	Ottelunlopputulos
HTHG	Kotijoukkueen maalimäärä puoliajalla
HTAG	Vierasjoukkueen maalimäärä puoliajalla
HTR	Puoliaikatulos
Referee	Tuomari

HS	Kotijoukkueen laukaukset
AS	Vierasjoukkueen laukaukset
HST	Kotijoukkueen laukaukset maalia kohti
AST	Vierasjoukkueen laukaukset maalia kohti
HF	Kotijoukkueen rikkeet
AF	Vierasjoukkueen rikkeet
HC	Kotijoukkueen kulmapotkut
AC	Vierasjoukkueen kulmapotkut
HY	Kotijoukkueen keltaiset kortit
AY	Vierasjoukkueen keltaiset kortit
HR	Kotijoukkueen punaiset kortit
AR	Vierasjoukkueen punaiset korit
B365H	Bet365:n määrittämä kotijoukkueen voiton kerroin
B365D	Bet365:n määrittämä tasapelin kerroin
B365A	Bet365:n määrittämä vierasjoukkueen voiton kerroin.
B365>2.5	Bet365:n määrittämä kerroin yli 2.5 maalia ottelussa
B365<2.5	Bet365:n määrittämä kerroin alle 2.5 maalia ottelussa

Taulukko 21 Tutkimuksessa käytetyn datajoukon kuvaus.

Ominaisuuden nimi:	**Ominaisuuden selite:**
HomeTeam	Kotijoukkue
AwayTeam	Vierasjoukkue
HTP	Kotijoukkueen kauden kokonaispisteet
ATP	Vierasjoukkueen kauden kokonaispisteet
HHP	Kotijoukkueen kauden kotipisteet
AHP	Vierasjoukkueen kauden kotipisteet
HAP	Kotijoukkueen kauden vieraspisteet
AAP	Vierasjoukkueen kauden vieraspisteet
HHGS	Kotijoukkueen tehdyt maalit kotiotteluissa
AHGS	Vierasjoukkueen tehdyt maalit kotiotteluissa
HAGS	Kotijoukkueen tehdyt maalit vierasotteluissa
AAGS	Vierasjoukkueen tehdyt maalit vierasotteluissa
HHAS	Kotijoukkueen keskimääräinen maalimäärä kotiotte-lussa
AHAS	Vierasjoukkueen keskimääräinen maalimäärä kotiot-telussa
HAAS	Kotijoukkueen keskimääräinen maalimäärä vierasot-telussa

AAAS	Vierasjoukkueen keskimääräinen maalimäärä vierasottelussa
HHGC	Kotijoukkueen päästetyt maalit kotiotteluissa
AHGC	Vierasjoukkueen päästetyt maalit kotiotteluissa
HAGC	Kotijoukkueen päästetyt maalit vierasotteluissa
AAGC	Vierasjoukkueen päästetyt maalit vierasotteluissa
HHDS	Kotijoukkueen päästettyjen maalien keskiarvo kotiottelua kohden
AHDS	Vierasjoukkueen päästettyjen maalien keskiarvo kotiottelua kohden
HADS	Kotijoukkueen päästettyjen maalien keskiarvo vierasottelua kohden
AADS	Vierasjoukkueen päästettyjen maalien keskiarvo vierasottelua kohden
HHS	Kotijoukkueen laukaukset keskimäärin kotiotteluissa
AHS	Vierasjoukkueen laukaukset keskimäärin kotiotteluissa
HAS	Kotijoukkueen laukaukset keskimäärin vierasottelussa
AAS	Vierasjoukkueen laukaukset keskimäärin vierasottelussa
HHST	Kotijoukkueen laukaukset keskimäärin maalia kohti kotiottelussa

AHST	Vierasjoukkueen laukaukset keskimäärin maalia kohti kotiottelussa
HAST	Kotijoukkueen laukaukset keskimäärin maalia kohti vierasottelussa
AAST	Vierasjoukkueen laukaukset keskimäärin maalia kohti vierasottelussa
FTR	Kyseisen ottelun lopputulos
B365H	Vedonlyöntiyhtiön Bet365:n antama H2H-kerroin kotijoukkueen voitolle
B365D	Vedonlyöntiyhtiön Bet365:n antama H2H-kerroin tasapelille
B365A	Vedonlyöntiyhtiön Bet365:n antama H2H-kerroin vierasjoukkueen voitolle

B Urheiluvedonlyönnin teoria

Taulukko 22 Unibetin ja Veikkauksen kertoimet ja todennäköisyysarviot ottelulle Real Madrid – PSG

Real Madrid - PSG	**1**	**X**	**2**
Veikkauksen kertoimet	2,15	3,85	2,85
Käänteisluvut eli todennäköisyysarvio	0,465	0,260	0,351

Unibetin kertoimet	2,25	4,00	3,05
Käänteisluvut eli todennäköisyysarvio	0,444	0,250	0,328

C SVM: hyperparametrien optimointi GridSearchCV-funktiolla

Taulukko 23 GridSearchCV:n tulokset hyperparametrin C ja ydinfunktion etsimiselle

C	**Kernel function**	**Mean test score**
10	Linear	0.56155735
10	RBF	0.46474921
10	Polynomial	0.47106278
1	Linear	0.5634865
1	RBF	0.4640477
1	Polynomial	0.46895826
0.1	Linear	0.56611715
0.1	RBF	0.46229393
0.1	Polynomial	0.47737636
0.01	Linear	0.57348299
0.01	RBF	0.46229393
0.01	Polynomial	0.4847422

0.001	Linear	0.57962119
0.001	RBF	0.4622939
0.001	Polynomial	0.51841459
0.0001	Linear	0.57471063
0.0001	RBF	0.46229393
0.0001	Polynomial	0.53016485

D SVM: Menetelmän arviointi testausdatalla

Taulukko 24 SVM:n Instanssien ennustetut ja todelliset ryhmät

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**154**	26	0
	Vierasvoitto	33	**96**	0
	Tasapeli	51	19	**1**

Taulukko 25 SVM-menetelmän arviointi testausdatalla

	FPR	**TNR**	**Support**	**TPR**	**Precision**	**F-measure**	**AUC**
Kotivoitto	0.42	0.58	180	0.86	0.65	0.74	0.72
Vierasvoitto	0.18	0.82	129	0.74	0.68	0.71	0.78
Tasapeli	0.00	1.00	71	0.01	1.00	0.03	0.51

E KNN: Etäisyysmittarit ja k:n optimointi

Taulukko 26 Eri etäisyysmittarit kNN-algoritmille

Minkowskin etäisyys:	$D(x,y) = (\sum_{i=1}^{m} \lvert x_i - y_i \rvert^r)^{1/r}$
Manhattanin etäisyys:	$D(x,y) = \sum_{i=1}^{m} \lvert x_i - y_i \rvert$
Chebychevin etäisyys:	$D(x,y) = \max_{i=1}^{m} \lvert x_i - y_i \rvert$
Camberran etäisyys:	$D(x,y) = \sum_{i=1}^{m} \frac{\lvert x_i - y_i \rvert}{\lvert x_i + y_i \rvert}$
Kendallin järjestyskorrelaatiokerroin:	$D(x,y) = 1 - \frac{2}{m(m-1)} * \sum_{i=j}^{m} \sum_{j=1}^{i-1} sign(x_i - x_j) sign(y_i - y_j)$

Taulukko 27 tulokset k:n eri arvoilla.

k	**Tulos**	***k***	**Tulos**	***k***	**Tulos**	***k***	**Tulos**
1	0.42603643	14	0.5254397	27	0.54082915	40	0.54663945
2	0.47707286	15	0.52873744	28	0.54334171	41	0.54679648
3	0.46451005	16	0.53282035	29	0.54491206	42	0.5450691
4	0.48900754	17	0.53517588	30	0.54569724	43	0.54381281
5	0.48979271	18	0.53815955	31	0.54444095	44	0.54648241
6	0.50125628	19	0.53674623	32	0.54648241	45	0.54805276
7	0.50738065	20	0.53753141	33	0.54601131	46	0.54962312
8	0.51256281	21	0.53800251	34	0.54444095	47	0.55072236
9	0.50957915	22	0.53941583	35	0.54679648	48	0.55072236
10	0.51460427	23	0.54098618	36	0.54585427	49	0.55009422
11	0.51680276	24	0.54271357	37	0.54711055		
12	0.52198492	25	0.5428706	38	0.54632538		
13	0.52465452	26	0.54585427	39	0.54569724		

F KNN: Menetelmän arviointi testausdatalla

Taulukko 28 kNN: Instanssien ennustetut ja todelliset ryhmät

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**153**	26	1
	Vierasvoitto	40	**87**	2
	Tasapeli	47	20	**4**

Taulukko 29 kNN-menetelmän arviointi testausdatalla

	FPR	**TNR**	**Support**	**TPR**	**Precision**	**F-measure**	**AUC**
Kotivoitto	0.44	0.57	180	0.85	0.64	0.73	0.71
Vierasvoitto	0.18	0.82	129	0.68	0.65	0.67	0.75
Tasapeli	0.01	0.99	71	0.06	0.57	0.10	0.52

G Naiivin Bayesin luokittimen esimerkki

Taulukko 30 Data-aineisto kuvaamaan milloin pelataan golfia.

Näkymä	**Lämpötila**	**Kosteus**	**Tuulisuus**	**Pelataan**
Aurinkoinen	Kuuma	Korkea	Epätosi	Ei
Aurinkoinen	Kuuma	Korkea	Tosi	Ei
Pilvinen	Kuuma	Korkea	Epätosi	Kyllä
Sateinen	Lämmin	Korkea	Epätosi	Kyllä
Sateinen	Viileä	Normaali	Epätosi	Kyllä
Sateinen	Viileä	Normaali	Tosi	Ei
Pilvinen	Viileä	Normaali	Tosi	Kyllä
Aurinkoinen	Lämmin	Korkea	Epätosi	Ei
Aurinkoinen	Viileä	Normaali	Epätosi	Kyllä
Sateinen	Lämmin	Normaali	Epätosi	Kyllä
Aurinkoinen	Lämmin	Normaali	Tosi	Kyllä
Pilvinen	Lämmin	Korkea	Tosi	Kyllä
Pilvinen	Kuuma	Normaali	Epätosi	Kyllä
Sateinen	Lämmin	Korkea	Tosi	Ei

Taulukko 31 Ehdolliset todennäköisyydet (uskottavuus) säätiloille, kun golfin pelaaminen on tosi tai epätosi.

Näkymä	
$P(Aurinkoinen\|Pelataan = Kyllä) = 2/9$	$P(Aurinkoinen\|Pelataan = Ei) = 3/5$
$P(Pilvinen\|Pelataan = Kyllä) = 4/9$	$P(Pilvinen\|Pelataan = Ei) = 0$
$P(Sateinen\|Pelataan = Kyllä) = 3/9$	$P(Sateinen\|Pelataan = Ei) = 2/5$
Lämpötila	
$P(Kuuma\|Pelataan = Kyllä) = 2/9$	$P(Kuuma\|Pelataan = Ei) = 2/5$
$P(Lämmin\|Pelataan = Kyllä) = 4/9$	$P(Lämmin\|Pelataan = Ei) = 2/5$
$P(Viileä\|Pelataan = Kyllä) = 3/9$	$P(Viileä\|Pelataan = Ei) = 1/5$
Kosteus	
$P(Korkea\|Pelataan = Kyllä) = 3/9$	$P(Korkea\|Pelataan = Ei) = 4/5$
$P(Normaali\|Pelataan = Kyllä) = 6/9$	$P(Normaali\|Pelataan = Ei) = 1/5$
Tuulisuus	
$P(Tosi\|Pelataan = Kyllä) = 3/9$	$P(Tosi\|Pelataan = Ei) = 3/5$
$P(Epätosi\|Pelataan = Kyllä) = 6/9$	$P(Epätosi\|Pelataan = Ei) = 2/5$

Taulukko 32 Uudet ehdolliset todennäköisyydet Laplacen estimoinnin jälkeen.

Näkymä	
$P(Aurinkoinen\|Pelataan = Kyllä) = 3/12$	$P(Aurinkoinen\|Pelataan = Ei) = 4/8$
$P(Pilvinen\|Pelataan = Kyllä) = 5/12$	$P(Pilvinen\|Pelataan = Ei) = 1/8$
$P(Sateinen\|Pelataan = Kyllä) = 4/12$	$P(Sateinen\|Pelataan = Ei) = 3/8$
Lämpötila	
$P(Kuuma\|Pelataan = Kyllä) = 3/12$	$P(Kuuma\|Pelataan = Ei) = 3/8$
$P(Lämmin\|Pelataan = Kyllä) = 5/12$	$P(Lämmin\|Pelataan = Ei) = 3/8$
$P(Viileä\|Pelataan = Kyllä) = 4/12$	$P(Viileä\|Pelataan = Ei) = 2/8$
Kosteus	
$P(Korkea\|Pelataan = Kyllä) = 4/11$	$P(Korkea\|Pelataan = Ei) = 5/7$
$P(Normaali\|Pelataan = Kyllä) = 7/11$	$P(Normaali\|Pelataan = Ei) = 2/7$
Tuulisuus	
$P(Tosi\|Pelataan = Kyllä) = 4/11$	$P(Tosi\|Pelataan = Ei) = 4/7$
$P(Epätosi\|Pelataan = Kyllä) = 7/11$	$P(Epätosi\|Pelataan = Ei) = 3/7$

H Naiivin Bayesin luokittimen arviointi testausdatalla

Taulukko 33 GNB luokitin: instanssit ja todelliset ryhmät.

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**95**	28	57
	Vierasvoitto	12	**81**	36
	Tasapeli	21	16	**34**

Taulukko 34 GNB-menetelmän arviointi testausdatalla.

	FPR	**TNR**	**Support**	**TPR**	**Precision**	**F-measure**	**AUC**
Kotivoitto	0.17	0.84	180	0.53	0.74	0.62	0.68
Vierasvoitto	0.18	0.82	129	0.63	0.65	0.64	0.73
Tasapeli	0.30	0.70	71	0.48	0.27	0.34	0.59

Taulukko 35 MNB-luokitin: instanssit ja todelliset ryhmät

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**121**	43	16
	Vierasvoitto	19	**92**	18
	Tasapeli	30	27	**14**

Taulukko 36 MNB-menetelmän arviointi testausdatalla.

	FPR	**TNR**	**Support**	**TPR**	**Precision**	**F-measure**	**AUC**
Kotivoitto	0.25	0.75	180	0.67	0.71	0.69	0.71
Vierasvoitto	0.28	0.72	129	0.71	0.57	0.63	0.72
Tasapeli	0.11	0.89	71	0.20	0.29	0.24	0.54

Taulukko 37 CNB-luokitin: instanssit ja todelliset ryhmät.

		Ennustettu ryhmä		
		Kotivoitto	**Vierasvoitto**	**Tasapeli**
Todellinen ryhmä	**Kotivoitto**	**132**	47	1
	Vierasvoitto	25	**104**	0
	Tasapeli	37	33	**1**

Taulukko 38 CNB-luokitin: instanssit ja todelliset ryhmät.

	FPR	**TNR**	**Support**	**TPR**	**Precision**	**F-measure**	**AUC**
Kotivoitto	0.31	0.69	180	0.73	0.68	0.71	0.71
Vierasvoitto	0.32	0.68	129	0.81	0.57	0.66	0.74
Tasapeli	0.00	1.00	71	0.01	0.50	0.03	0.51