OULUN AMMATTIKORKEAKOULU

Sampo Vuorento

HSL KAUPUNKIPYÖRIEN VUOKRAUSTEN ENNUSTAMINEN

KONEOPPIMINEN LINEAARISEN REGRESSION AVULLA

HSL KAUPUNKIPYÖRIEN VUOKRAUSTEN ENNUSTAMINEN

KONEOPPIMINEN LINEAARISEN REGRESSION AVULLA

Sampo Vuorento
Opinnäytetyö
Lukukausi syksy 2024
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan Tutkinto-ohjelma, Ohjelmistokehityksen suuntatumisvaihtoehto

---

Tekijä: Sampo Vuorento
Opinnäytetyön nimi: HSL kaupunkipyörien vuokrausten ennustaminen
Työn ohjaaja: Manne Hannula
Työn valmistumislukukausi ja -vuosi: Syksy 2024
Sivumäärä: 47 + 1 liite

Työn aiheena oli perehtyä koneoppimiseen ja tutkia sen avulla HSL kaupunkipyörien vuokrausten ennustamista. Lopullisena tavoitteena oli luoda malli, jonka avulla pystytään ennustamaan pyörien vuokrausten määriä perustuen olemassa olevaan dataan ja säätietoihin.

Työssä käytettiin HSL:n tarjoamia avoimia vuokrausdatoja touko- heinäkuulta 2021, sekä ilmatieteenlaitoksen säätietoja. Olemassa olevaa dataa tutkittiin data-analytiikan avulla hyödyntäen MySQL-tietokantaa, Node.js:ää ja Reactia. Ennustemallin rakentamiseen käytettiin Pythonia. Mallin testaamista suoritettiin erilaisilla sääennusteilla. Testaamisen tuloksena pystyttiin ennustamaan pyörien vuokrausten määrä melko tarkasti vilkkailla asemilla. Suoritettujen testausten perusteella pystyimme toteamaan, että mitä enemmän dataa on käytettävissä, niin sitä tarkempia ennusteita voidaan luoda. Pienen datamäärän vuoksi hiljaisten asemien ennusteiden luominen osoittautui haastavaksi.

Asiasanat: Koneoppiminen, ohjattu oppiminen, lineaarinen regressio.

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Option of Software Development

---

Author: Sampo Vuorento
Title of thesis: HSL kaupunkipyörien vuokrausten ennustaminen
Supervisor: Manne Hannula
Term and year when the thesis was submitted: fall 2024
Number of pages: 47 + 1 appendix

The thesis focused on predicting from HSL data the rental volumes of HSL city bikes using machine learning. The aim was to create a model that was capable of predicting bike rentals using the existing rental data provided by HSL. Open data from HSL bike rentals was from May to July 2021 and weather data was provided by Finnish Meteorological Institute. The application was built for data analysing by using MySQL for the database, Node.js and React for the interface. The predictive model was built with Python. Model was tested with various weather forecasts. Model then predicted rental volumes for high-traffic stations with good accuracy. Conclusion was that the more data available, the more precise would the predictions be, whilst forecasting for quiet low-traffic stations proved to be very challenging.

Keywords: Machine learning, supervised learning, linear regression

SISÄLLYS

1 JOHDANTO

Opinnäytetyössä perehdytään ensin siihen, mitä koneoppiminen on ja käydään läpi hieman teorioita. Koneoppimisen eri tyyppejä käydään läpi ja pureudutaan enemmän ohjattuun oppimiseen, mitä myös tämän opinnäytetyön tutkimuksessa käytetään. Koneoppiminen on varsin uusi tieteenala ja edistyy suurin harppauksin. Asiaan koitetaan perehtyä tällä hetkellä käytettävissä olevia tietoja hyödyntäen.

Opinnäytetyötä varten suunniteltiin projekti kaupunkipyörien vuokraamisesta. Projektiin saatiin suuri määrä dataa HSL:n (Helsingin Seudun Liikenne) kaupunkipyörien vuokraamisista 2021 touko-heinäkuun ajalta. Pyörien vuokrausasemia on satoja Helsingin ja Espoon alueella. Kaikki tämä data oli vapaasti ladattavissa ja käytettävissä. Taustatutkimuksena tutkittiin verkossa ollutta kilpailua, jossa pyrittiin tekemään ennustuksia pyörävuokrausten suhteen. Näistä projekteista saatiin ideoita oman projektin tekemiseen. Koneoppimisen kannalta on aina parempi, mitä enemmän dataa on käytettävissä. Suurella datamäärällä voidaan tehdä parempia malleja tulevaisuuden ennustamisen suhteen.

Aiheen valinta perustui siihen, että koneoppiminen on tällä hetkellä ajankohtainen asia ja kehitys on alalla nopeaa. Koneoppiminen on erittäin mielenkiintoinen ja monipuolinen aihe, jota voisi tutkia monelta eri kantilta erilaisin menetelmin. Aihe kiinnostaa erityisen paljon ja toivonkin, että tästä aiheen monipuolisesta tutkimisesta olisi hyötyä mahdollisesti työmarkkinoilla. Koska alalla kehitys on varsin nopeaa, niin se vaatii jatkuvaa opiskelua, jota on hyvä jatkaa tämän opinnäytetyön kirjottamisen jälkeen. Tässä työssä pyrittiin selvittämään, kuinka pystytään olemassa olevan datan perusteella ennustamaan kaupunkipyörien vuokrauksia. Data analytiikan avulla pystyy tekemään vuokrauksien määristä hakuja, näitä hyödyntämällä suoritetaan Python-kielellä koneoppimisen malleja, joiden avulla tehdään ennustuksia tulevista pyörien vuokrauksista. Käyttäjän antamien muuttujien avulla tehdään ennustus seuraavan päivän pyörien vuokrausten ja palautusten määrästä, käyttäjän valitsemalle pyöräasemalle. Ennustusten tekemiseen käytetään olemassa olevaa pyörien vuokrausten dataa sekä Ilmatieteen laitoksen säätietokantaa.

2 MITÄ ON KONEOPPIMINEN

Ensimmäisten koneiden rakentamisesta asti on koneiden tehtävä ollut suorittaa jokin työ. Alkuvaiheessa koneiden kehittäjät antoivat koneille hyvin helppoja tehtäviä suoritettavaksi. Koneet olivat alkuun vain mekaanisia eivätkä tehneet mitään muuta kuin sitä tehtävää, jota varten ne rakennettiin. Ajan kuluessa koneet ovat muuttuneet alati monimutkaisemmiksi. Koneiden alkuperäisestä yhdestä tehtävästä on koneet suorittamat tehtävät lisääntyneet, koneiden käyttökohteiden määrä on myös kasvanut räjähdysmäisesti.

Ensimmäinen elektroninen tietokone ENIAC (Electronic Numerical Integrator and Computer) otettiin käyttöön marraskuussa 1945. Kone oli valtava ja painoi 30 tonnia. Varhaisten tietokoneiden kehittämisen myötä tapahtui koneissa suuri vallankumous. Koneiden tehtävä ei ollutkaan enää vain mekaaninen vaan niille voitiin syöttää koodin kautta tehtäviä. Tässä vaiheessa koneet alkoivat ensimmäistä kertaa oppimaan jotakin. Kaikki oppiminen tapahtui ihmisen kirjoittaman koodin kautta. Automaatio oli syntynyt mikä mahdollisti koneiden toimimisen jopa ilman ihmisen valvontaa. (1.)

Koneiden kehitys on ollut todella nopeaa teollistumisen jälkeen. Deep Blue tietokone voitti tekoälyn avulla shakin hallitsevan maailmanmestarin 1996, tämä oli tekoälyn suurin saavutus siihen mennessä ja tästä kehitys on jatkunut hurjana eteenpäin. Koneille keksitään jatkuvasti uusia toimintatapoja ja uusia paikkoja missä niitä käyttää. Koneoppiminen on ottanut isoja harppauksia eteenpäin ja kehitys on nopeaa. Pitkään on puhuttu, että koneet saattavat syrjäyttää ihmiset työpaikoilla. Näin on joillakin aloilla jo tapahtunut ja uusia aluevaltauksia tulee jatkuvasti. Teolliset tuotantolinjat on ollut jo pitkään koneiden vastuulla. (2.)

Tekoälyn kehittämisen myötä on koneet alkaneet oppia itsenäisesti asioita. Koneet pystyvät korvaamaan ihmisiä useilla uusilla aloilla. Palvelunumerot ovat muuttuneet osittain automaattisiksi. Useilla verkkosivuilla kommunikoidaan verkkokeskustelussa robotin kanssa. Nämä robotit pystyvät tulkitsemaan ihmisten kysymykset ja etsivät tietokannastaan kysymyksiin vastaukset. Varsin vähän aikaa sitten tekoälyt vielä vaativat sen, että niiden muistiin syötetään tiedot käytettäväksi. Viimeisimmät koneet pystyvät jo pienen tiedon avulla kasvattamaan tietoisuuttaan. Koneelle voidaan kertoa mikä pyörä on ja tämän tiedon perusteella kone voi löytää verkosta äärettömän määrän kuvia missä esiintyy pyörä. Hakukoneet toimivat sillä perusteella mitä tietoa, vaikka kuvasta on kirjoitettu. Tekoälyn myötä koneet pystyvät tulkitsemaan itse kuvan sisältöä. (3, s. 10.)

Koneoppiminen on kiinnostanut ihmisiä suuresti ensimmäisten tietokoneiden valmistumisesta asti. Tietokoneiden ilmestyessä yhteiskuntaan yleisesti oletettiin tietokoneiden pystyvän tekemään päätöksiä kuten ihmisaivot. Koneoppiminen perustuu algoritmeihin ja tietyllä tavalla se onkin tieteellistä laskentaa. Koneoppiminen voidaan jakaa kolmeen pääkategoriaan: ohjattu oppiminen, ohjaamaton oppiminen ja vahvistettu oppiminen. (3, s. 10.)

2.1 Ohjattu oppiminen

Ohjatussa oppimisessa nimensä mukaisesti annetaan koneelle tietoa ihmisten syöttämänä, tiedoilla opetetaan algoritmi oppimaan. Koko ohjatussa oppimisessa luotu aineisto luo jokaiselle tapaukselle sekä syötteen että tuloksen. Tässä tapauksessa oppimaan ohjattu algoritmi etsii parasta funktiota kokeilemalla, miten kukin sen tuottama funktio vastaa sen omaa aineistoa. Samanaikaisesti aineisto tuottaa tuloksen ja toimii oppimisen ohjaajana. Tuloksen laatu riippuu paljon aineiston määrästä ja laadusta. Mitä enemmän syötettyä tietoa on, niin sitä parempi tulos saadaan. Syötteen laadun tulee myös olla hyvä. Huono syöte voi tuottaa heikkoja malleja ja koko algoritmi voi epäonnistua. Huono syöte yleensä johtuu datan puutteesta tai huonolaatuisesta datasta. (4, s. 100–103.)

Hyvänä esimerkkinä ohjatusta oppimisesta voidaan pitää IBM:n tietokonetta Deep Blue joka päihitti shakin hallitsevan maailmanmestarin Garry Kasparovin vuonna 1997. Jo vuonna 1985 Kasparov oli otellut varhaista tietokonetta vastaan. Tuolloin oli samaan tilaan tuotu 32 tietokonetta ja Kasparov pelasi pöytää vaihtaen samaan aikaan kaikkia koneita vastaan. Tuolloin Kasparov voitti 32-0. Deep Blue oli ensimmäinen kone, joka voitti shakin maailmanmestarin turnauspelissä. Deep Bluen voittoa Kasparovista pidetään merkittävänä virstanpylväänä tekoälyn kehittämisessä. Ohjattu oppiminen oli viety todella pitkälle, kone käytti monimutkaista hakualgoritmia ja siihen aikaan todella suurta laskentatehoa analysoimaan kaikkia mahdollisia siirtoja kulloisellekin pelitilanteelle. Deep Blue oli ohjattu ainoastaan pelaamaan shakkia, mutta osoitti että kone pystyy voittamaan ihmisen monimutkaisessakin pelissä. Nykyään kuka vain voi ladata älypuhelimeensa shakkisovelluksen, jonka tekoäly vastaa ketä tahansa shakin suurmestaria. (5, s. 9–10.)

Hyvänä esimerkkinä voidaan myös pitää Internetin hakukoneita. Hakukoneidenkin toiminta perustuu koneoppimiseen. Käyttäjien aikaisempien hakujen perusteella hakukoneet oppivat tarjoamaan käyttäjille mahdollisimman oikeita hakutuloksia. Roskapostisuodatin on toinen hyvä käytännön esimerkki. Suodattimelle voidaan opettaa tiettyjä fraaseja, joita sisältävät sähköpostit voidaan suoraan

siirtää roskapostiin. Roskapostisuodattimille opetettiin alkuun yksinkertaisia asioita, joilla löytää roskaposti. Nykyään suodattimet ovat jo varsin monimutkaisia. (6, s. 241.)

Laadukkaan datan pitää olla juuri semmoista kuin halutaan ja sitä täytyy olla riittävästi hyvän tuloksen saamiseksi. Datan tulee olla selvää, tarkkaa ja oikeellista. Erilaisten muuttujien avulla voidaan poistaa datasta selvästi vääriä arvoja, tämä tulee tehdä ennen kuin dataa käytetään. Dataa voidaan myös käyttää useammasta lähteestä, tällöin pitää ottaa huomioon datan samanlainen sisältö, jotta sitä voidaan käyttää samassa tehtävässä. Datan suhteen pitää olla perillä monista eri asioista, jotta sitä voidaan käyttää koneoppimisessa. Suosituimpia menetelmiä ohjattuun oppimiseen ovat päätöspuut, bayes-verkot sekä k-lähimmän naapurin algoritmit. (6, s. 243–244.)

2.1.1 Lineaarinen regressio

Ohjatun oppimisen yksi käytetyimmistä menetelmistä on lineaarinen regressio. Lineaarisen regression mallit ovat varsin yksinkertaisia ja helposti tulkittavia. Monessa tilanteessa onkin parasta pitää menetelmät mahdollisimman yksinkertaisina ja saada helposti tulkittava tulos nopeasti, jopa suhteellisen pienellä otannalla. Kaikista yksinkertaisin sovellus lineaarisesta regressiosta on sovelluksen mallintaminen kahden piirteen suhteen: syötepiirre X sekä kohdepiirre Y. Tässä regressiofunktio lineaariseen regression ongelmaan on:

$$Y = \omega_0 + \omega_1 X$$

Regressiofunktio on vain suoran yhtälö, funktion parametrit ovat ω_0 ja ω_1. Parametri ω_0 X:n arvolla nolla on y-akselin ja suoran leikkauspiste. Suoran kaltevuuden määrittää parametri ω_1. Parametrien arvot regressiofunktiossa ovat alunperintuntemattomat. Parametreille kun asetetaan arvoja, saadaan luotua aineistoa parhaiten kuvaava suora. Funktion aineiston kasvaessa saadaan pienennettyä kokonaisvirhettä. (4, s. 113–119.)

2.1.2 Logistinen regressio

Logistisessa regressiossa muuttuja voi saada vain kaksi arvoa. Logistista regressiota pidetään yksinkertaisen regressioanalyysin erityistyyppinä. Muuttujan kahta mahdollista arvoa pyritään selittämään sitä kautta, miten siihen vaikuttavat erilaiset tekijät. Logistisella regressiolla pyritään ennustamaan todennäköisyyksiä. Eri tekijöiden arvoja tarkasteltaessa lasketaan todennäköisyyttä millä

tarkasteltava asia tapahtuu. Millä todennäköisyydellä muuttujien arvot vaikuttavat tulokseen voidaan esimerkiksi arvioida ostavatko tuotetta ennemmin miehet vai naiset. (7.)

Pelkistettynä logistinen regressiomalli on kuin tavallinen regressiomalli. Mallissa on muuttujana tutkittavana olevan tapahtuman vedon logaritmi. Jotta logistinen regressioanalyysi ymmärretään paremmin, pitää ymmärtää, mitä tarkoitetaan vedolla (odds). Yleisesti vetosuhteita käytetään esimerkiksi vedonlyönnissä, voittosuhteita kuvaamaan käytetään vetolukuja. Esimerkiksi todennäköisyys, että nainen ostaa tuotteen on 0,8 ja todennäköisyys ettei hän osta tuotetta on 0,2. Naisen veto on täten 0,8 / 0,2 = 4. Todennäköisyys että mies ostaisi tuotteen on 0,5, näin ollen miesten veto on 0,5 / 0,5 = 1. Sukupuolten välinen vetosuhde saataisiin jakamalla naisten veto miesten vedolla 4 / 1 = 4. Vetosuhdekerroin on 4, tästä voi päätellä, että naisten todennäköisyys ostaa tuote on nelinkertainen miehiin verrattuna. Logistinen regressiomalli, jossa muuttujana on kyseisen tapahtuman vedon logaritmi, joka voidaan ilmaista kaavalla:

$$ln\left[\frac{P(Y=1)}{1-P(Y=1)}\right] = a + bx$$

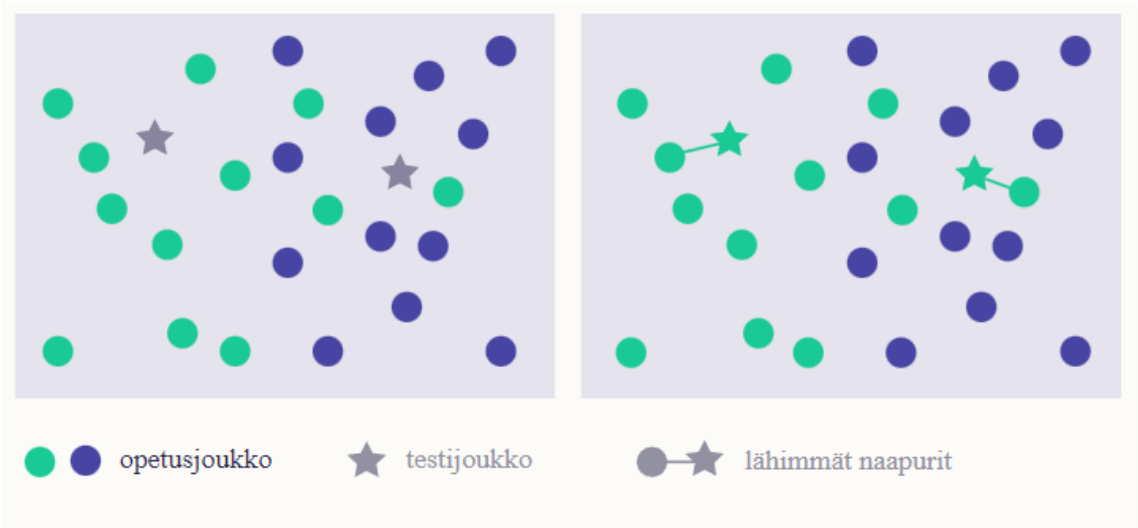
Tässä kaavassa $P(Y=1)$ on todennäköisyys, jolla muuttuja saa arvon yksi, a:n ollessa vakiotekijä, b regressiokerroin sekä x muuttujan arvo. Tämän logistisen regressiomallin kaavan lauseke $a + bx$ on täysin sama kuin lineaarisessa regressiomallissa. Tämän takia logistisen regressiomallin ongelmat sekä tulkinta ovat lähes identtiset kuin lineaarisessa regressioanalyysissä. (7.)

2.1.3 Lähimmän naapurin luokitin

Lähimmän naapurin luokitin, eli lyhyesti kutsuttuna 1-NN-luokitin (nearest neighbor classifier). 1-NN-luokitin on yksi yksinkertaisimpia luokittimia ja sitä voidaan kutsua paremmin luokittelusäännöksi, kuin luokittimeksi. Alun perin 1-NN-luokitin kehitettiin jo 50-luvulla, mutta iästään huolimatta se on hyvä perusmenetelmä, johon voidaan verrata muita luokittimia. Lyhyesti määriteltynä luokitellaan tuntematon näyte x, siihen luokkaan johon opetusaineistossa sen lähin osuma kuuluu. (3, s. 41–42.)

Kuvassa 1 on kuvattuna joukko opetusdataa (opetusjoukko). Opetusdatan jokainen arvo kuuluu omaan luokkaansa, vihreä tai sininen. Opetusdatan lisäksi on näkyvissä kaksi testiesimerkkiä, jotka on merkitty tähdellä. Luokitin määrittää kumpaan luokkaan nämä testiesimerkit kuuluvat.

Molemmat testiesimerkit luokitellaan vihreään luokkaan, koska vihreästä luokasta löytyy molempien lähin naapuri. (8.)



KUVA 1. Lähimmän naapurin dataa (8.)

2.1.4 Päätöspuut

Päätöspuut ovat yksi koneoppimisen vanhimpia menetelmiä. Päätöspuut ovat visuaalisesti helposti ymmärrettäviä ja selkeitä, ja tästä syystä päätöspuut ovat varsin suosittuja. Päätöspuut ovat tosin herkkiä muutokselle ja pienikin muutos datassa saattaa rakentaa täysin erilaisen puun. Päätöspuun rakenne koostuu kolmesta osasta. Koko datasetti löytyy juuresta ja sieltä alkaa datan jakaminen. Data jakaantuu jos-niin-säännöillä. Juuresta mennään sisäsolmuihin, joissa dataa jaetaan jos-niin-säännöillä aliryhmiin. Viimeisenä tulee lehtisolmut, jotka ovat puun loppusolmut ja antavat algoritmin ennusteen tai tuloksen. Lehtisolmut määrittävät luokan luokittelutehtävässä ja arvon regressiotehtävässä. Päätöspuu rakentuu jos-niin-säännöistä. Dataa jaetaan osajoukkoihin, kunnes jokin lopetusehto täyttyy. Yleisimpinä lopetusehtoina voidaan pitää sitä, että solmun kaikki datan havainnot ovat samaa luokkaa, solmun sisältämä data on pieni tai saavutetaan puun maksimisyvyys. (9, s. 96–99.)

2.1.5 Neuroverkot

Koneoppimisen ja tekoälyn menetelmät eli neuroverkot on johdettu biologisista hermoverkoista. Neuroverkot koostuvat neuroneista, jotka ovat yksinkertaisia laskennallisia yksiköitä, ja nämä yksiköt on järjestetty kerroksiksi. Neuroverkkojen avulla voidaan oppia tehokkaasti monimutkaisia malleja, siksi neuroverkkoja käytetäänkin varsin laajasti monissa erilaisissa sovelluksissa pelien tekoälystä kuvantunnistukseen. Neuroverkot rakentuvat kolmesta kerroksesta. Syötekerros vastaanottaa ensimmäisenä kaikki alkuperäiset syötteet. Toisena tulevat piilokerrokset, jotka voivat itsessään sisältää monia välikerroksia. Piilokerrokset suorittavat syötteen perusteella laskennallista prosessointia. Piilokerrokset eivät ole syötteelle suoraan näkyvissä. Viimeisenä kerroksena on ulostulokerros, tämä kerros tuottaa päätökset ja ennusteet neuroverkolta. Yksinkertaisesti voidaan sanoa koneen etsivän kuvasta tunnettuja muotoja. Ihminen voi piirtää tikku-ukon käyttämällä muutamaa viivaa ja yhtä ympyrää päänä. Aivot tunnistavat äkkiä mistä on kyse, neuroverkkojen periaate perustuu samaan tunnistamiseen. (10, s. 73–74.)

Neuroverkoille löytyy monia etuja. Neuroverkot pystyvät oppimaan monimutkaisia malleja, ne ovat myös hyvin skaalautuvia ja pystyvät käsittelemään suuria määriä dataa sekä monimutkaisia tehtäviä, monilla eri aloilla. Neuroverkoille löytyy myös haittoja. Suurimpana haittana voidaan pitää laskennallista vaativuutta. Neuroverkkojen laskennallinen vaativuus saattaa vaatia järeää laskentatehoa sekä paljon aikaa. Neuroverkkojen tulkinta saattaa olla vaikeaa ja niiden sisäistä toimintaa voi olla hankala tulkita. Yleensä neuroverkot vaativat myös suuren määrän dataa, jotta oppiminen on tehokasta. (3, s. 116–126.)

2.2 Ohjaamaton oppiminen

Ohjaamattomassa oppimisessa ihminen ei ole enää yksin syöttämässä dataa koneelle, vaan kone pystyy itsenäisestikin tekemään ratkaisuja. Ohjaamaton oppiminen ei ole välttämättä monimutkaisempaa kuin ohjattu oppiminen, mutta mahdollistaa monia uusia asioita. Molemmilla oppimistavoilla on omat vahvuutensa ja heikkoutensa. Siinä missä ohjattu oppiminen tarvitsee suuren määrän dataa, voidaan ohjaamatonta oppimista suorittaa pienemmällä määrällä dataa ja huomattavasti halvemmalla. (11, s. 130.)

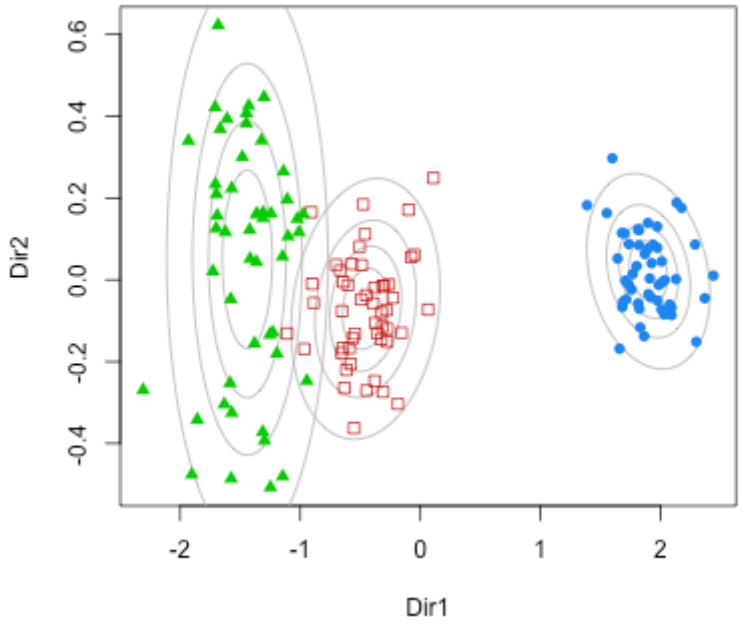
Luokiteltua dataa ei ole käytettävissä ohjaamattomassa oppimisessa. Jos käytössä on suuri määrä samankaltaisia kuvia, vaikka ihmisten kasvokuvia, voidaan ohjaamattoman oppimisen

menetelmillä luokitella saman henkilön kasvopiirteitä muistuttavat kuvat omiin ryhmiinsä. Täten saadaan aikaan ryhmiä, joissa samankaltaiset kuvat ovat omissa ryhmissään. (11, s. 130.)

2.2.1 Klusterointi

Klusteroinnissa data jaetaan klustereihin tai ryhmiin, siten että jokaisessa klusterissa kaikki data on mahdollisimman samankaltaista keskenään. Samalla eri klustereissa olevat datapisteet ovat mahdollisimman erilaisia. Täten saadaan monta eri klusteria, jotka ovat keskenään mahdollisimman erilaisia ja itsessään mahdollisimman samankaltaista dataa sisältäviä. Klusteroinnissa ei ole ennakkoon määriteltyjä luokkamerkintöjä, klusterointi on valvomattoman oppimisen menetelmä. (12.)

Klusterointimenetelmiä on useita erilaisia. K-keskiarvo on yksi yleisimmistä, tavoitteena on jakaa data k klusteriin, data jaetaan niin että jokainen datapiste on lähinnä klusterin keskipistettä. Hierarkkisessa klusteroinnissa rakennetaan klustereista hierarkia joko alaosat yhdistämällä tai yläosat jakamalla. Tiheyteen perustuva klusterointi etsii datasta tiheästi dataa sisältäviä alueita ja muodostaa klustereita näistä alueista. Tiheyteen perustuva klusterointi on käytännöllinen, kun klusterit sisältävät kohinaa tai ovat epäsäännöllisiä muodoltaan. Gaussian sekoitusmalli taas mallintaa dataa perustaen oletukseen, että data on peräisin useasta normaalijakaumasta. Gaussian sekoitusmalli käyttää odotusarvomaksimointia klusterien löytämiseen. Kuvassa 2 näkyy eri väreillä kuvattuna kolme eri klusteria. (12.)



KUVA 2. Kolme eri klusteria kuvattuna (13.)

Anomaliantunnistus on klusteroinnin sovellus, jolla tiedoista koitetaan etsiä poikkeuksia tai epänormaaleja havaintoja, jotka eroavat huomattavasti tietojen normaalista käyttäytymisestä. Tunnistettuja poikkeamia sanotaan anomalioiksi. Useilla eri sovellusalueilla voidaan käyttää anomaliantunnistusta kuten vikatilanteiden ennakoimisessa ja petosten havaitsemisessa. Keskeisinä käsitteinä voidaan mainita normaali havainto, joka vastaa datan normaalia käyttäytymistä. Anomalia on poikkeama datassa, joka eroaa huomattavasti datan normaalista käyttäytymisestä. Anomalian hyöty tulee siitä, että se saattaa viitata kiinnostavaan tai jopa huolestuttavaan tapahtumaan tietomassassa. Anomaliasta voidaan käyttää myös termiä poikkeama. (12.)

2.2.2 Pääkomponettianalyysi

Pääkomponenttianalyysi eli PCA on tilastollinen menetelmä, jonka avulla vähennetään korkean ulottuvuuden datan dimensioita, samalla säilytetään alkuperäisen datan varianssista mahdollisimman paljon. Datan visualisoinnin ja analysoinnin helpottamiseksi data projisoidaan pienempään dimensioon. Pääkomponenttianalyysin suurimmat hyödyt saadaan käsiteltäessä suuria määriä dataa, datan visualisoinnissa sekä mallien yksinkertaistamisessa. PCA:n eduiksi voidaankin mainita yksinkertaisuus ja tehokkuus, jota voidaan soveltaa suuriin datamääriin. Tietoa saadaan tiivistettyä

säilyttämällä suurin osa datan varianssista samalla vähentäen ulottuvuuksia. Haittoina voidaan pitää monimutkaisissa datarakenteissa pääkomponenttien lineaarisuutta alkuperäisiin muuttujiin. Alkuperäisten muuttujien kannalta pääkomponentit saattaa olla vaikeasti tulkittavissa. (9, s. 95–96.)

2.2.3 Assosiaatiosäännöt

Assosiaatiosäännöillä etsitään suurista tietomassoista suhteita sekä yhteyksiä, jotka liittyvät toisiinsa. Assosiaatiosääntöjä sovelletaan paljon markkina-analyyseissä. Ostoskorianalyysi on tästä hyvä esimerkki, voidaan ajatella henkilön ostaessa tuotteen X, niin hän todennäköisesti ostaa myös tuotteen Y. Sääntö määrittää sen, että suuri prosentti tuotteen X ostaneista osti myös tuotteen Y. Mikäli tuotteen X ostanut ei ole vielä ostanut tuotetta Y, voidaan olettaa hänen vielä ostavan tuotteen Y. Tuote X voisi olla vaikkapa leipä ja tuote Y margariini. Monet yhtiöt käyttävät kanta-asiakasohjelmia, joiden avulla voidaan luoda erilaisia sääntöjä, jotka saattavat pitää paikkansa, mutta niihin ei voi tietenkään täysin luottaa. Assosiaatiosäännöt voidaan käyttää suurissa tietokannoissa datan analysointiin ja erilaisten piilevien yhteyksien löytämiseen. Näin voidaan löytää käytännöllisiä ja käytännöllisiä säännönmukaisuuksia, joita voidaan käyttää hyväksi monilla eri aloilla kuten vähittäiskaupassa. (9, s. 149–150.)

2.3 Vahvistettu oppiminen

Yhtenä virstanpylväänä vahvistetussa oppimisessa voidaan pitää AlphaGo oppivaa ohjelmaa, joka voitti silloisen maailman parhaan Go pelaajan Lee Sedolin. 18-kertainen maailmanmestari Sedolin oli tappiosta niin pettynyt että lopetti kokonaan Gon pelaamisen. Siinä missä shakkilaudan koko on 8x8 on Go pelilauta 19x19, mahdollisten siirtojen määrä on dramaattisesti suurempi. Pelin monimutkaisuuden takia odotettiin, että menisi paljon pidempi aika, kunnes Go pelissä kone voittaisi ihmisen. AlphaGo pelasi itsenäisesti suuren määrän pelejä ja oppi niistä, määrä oli huomattavasti suurempi kuin mitä yksikään ihminen ehtisi elämänsä aikana pelaamaan. (9, s. 9.)

AlphaGon kehitti Googlen Deepmind-tiimi. Sovelluksesta teki vielä erikoisemman se, ettei sitä ohjelmoitu suoraan Gon pelaamiseen. Koodissa ei suoraan ohjattu toimimaan siirtojen mukaisesti, suoritettavia tehtäviä ei kohdistettu suoraan Go peliin. Ohjelman koodi perustui siihen, että kehitettiin kahta olemassa olevaa yleistä tekniikkaa. Lookahead-haulla luodaan hakualgometri joka ennakoi tulevia siirtoja monta askelta eteenpäin. Vahvistetulla oppimisella ohjelma arvioi pelin tilanteita.

Nämä tekniikat mahdollistivat sovelluksen pelaamisen yli-inhimillisellä tasolla. AlphaGosta kehitettiin myös uusi versio AlphaZero. Tämä uusi versio onnistui voittamaan AlphaGon Go-pelissä, sekä myös muita kehittyneitä ohjelmia kuten shakkia pelaavan Stockfishin. (14, s. 46–47.)

Vahvistettu oppiminen eroaa valvotusta oppimisesta siinä, että oppiminen perustuu palkitsemiseen ja rangaistuksiin. Sille ei anneta suoria vastauksia tai suoraa palautetta suoritetuista toiminnoista. Vahvistetussa oppimisessa agentti oppii toimimaan ympäristössään maksimoidakseen saamansa palkkiot pitkällä aikavälillä. (3, s. 152–171.)

Vahvistettu oppiminen koostuu peruskomponenteista, agentti on oppimisen oppiva yksikkö, joka suorittaa päätöksiä ja toimintoja ympäristössä. Ympäristö on se missä agentti suorittaa toimintojaan ja mistä agentti saa palautetta. Toimintatila edustaa kaikkia tilanteita, joissa agentti voi olla. Toiminto on agentin tekemät siirrot tai päätökset toimintatiloissa. Palkkio on palaute, jonka agentti saa suoritettuaan toiminnon. Se voi olla joko positiivinen eli palkinto tai negatiivinen eli rangaistus. Toimintapolitiikka on sääntöjoukko tai strategia mitä agentti noudattaa valitessaan toimintoja tietyissä tiloissa. Arvotoiminto on pitkällä aikavälillä toimintatilan arvon odotusarvo, sen avulla arvioidaan agentin tilaa. Q-funktion avulla ennakoidaan jokaisen toiminnon palkinnon määrä. Q-funktion avulla löydetään toiminnot, joiden avulla pyritään tavoittamaan suurin saavutettavissa oleva palkinto mahdollisimman nopeasti. (3, s. 152–171.)

3 TAUSTATUTKIMUS

Sivustolla https://www.kaggle.com/ on tarjolla paljon erilaisia projekteja lukuisista eri aiheista, joissa hyödynnetään tekoalyä. Sivustolta löytyy lukuisia projekteja koskien vuokrattavia polkupyöriä. Tutkimuksen ensimmäinen malliprojekti oli osallistunut kyseisellä sivulla ”Bike Sharing Demand” nimiseen kilpailuun. Kilpailussa annettiin kahdessa eri tiedostossa yhteensä 17 381 riviä dataa. Tämä on suhteellisen pieni määrä tarkkojen ennustusten tekemisen suhteen. Tätä materiaalia käytetään kilpailun tulosten tutkimiseen. Kilpailuun osallistui yhteensä 3 242 joukkuetta.

Tähän kilpailuun osallistui paljon erilaisia tiimejä ja lopputuloksina syntyi varsin erilaisia malleja. Useita kilpailusuorituksia tutkittiin ja niiden pohjalta pyrittiin löytämään mielenkiintoisia projekteja. Omaa tutkimusta varten perehdytään kahteen eri projektiin. Näistä projekteista otetaan mallia oman tutkimuksen tekemiseen ja mitä sen pitäisi mahdollisesti sisältää. Projekteissa on varsin erilaisia lähestymistapoja. Dataa on tarjolla ympäri vuoden ja ennustuksia tehdäänkin monen eri muuttujan suhteen. Eri muuttujia on viikonpäivät, vuodenajat sekä erilaiset säätiedot.

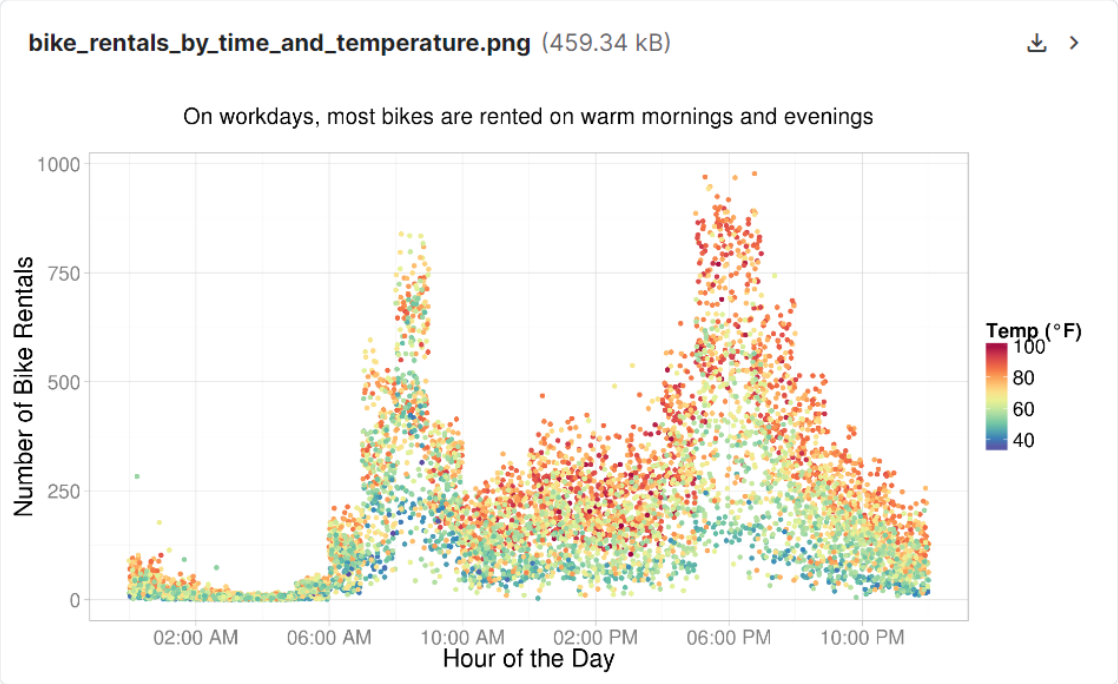
Toinen malliprojekti oli Kaggle sivulta projekti, jossa käytettiin hyödyksi ”Bike Rental Data Set – UCI” (https://www.kaggle.com/datasets/aguado/bike-rental-data-set-uci) tarjoamaa datasettiä. Datasetin käyttöön ei ollut määritelty vaatimuksia. Datasettiä sai hyödyntää parhaaksi katsomallaan tavalla. Tässä datasetissä oli 7 689 riviä dataa.

3.1 Ensimmäinen malliprojekti

Tutkimuksen ensimmäinen malliprojekti on Ben Hammerin tekemä ”Bike Rentals By Time And Temperature” (https://www.kaggle.com/code/benhamner/bike-rentals-by-time-and-temperature). Tässä projektissa on käytössä polkupyörien vuokraamisen kannalta ehkä tärkeimmät muuttujat. Kuten lähes kaikista muistakin projekteista tärkein muuttuja on aika. Tietyt kellonajat ovat selvästi ruuhkaisempia kuin muut. Tähän vaikuttaa työmatkaliikenne, joka tuottaa ruuhkapiikit arkisin aamuisin kello seitsemän ja kymmenen välille sekä iltapäivisin kello neljän ja seitsemän välille. Yöaikaan vuokrauksia on paljon vähemmän, eihän silloin ihmisiäkään ole paljoa liikenteessä.

Toinen tärkeä muuttuja polkupyörien vuokrauksen suhteen on vallitseva säätila. Lämpimässä aurinkoisessa päivässä on huomattavasti enemmän vuokrauksia kuin kylmässä ja sateisessa säässä.

Tutkimuksessa oli vuoden pyörävuokraukset. Materiaalia ei ollut suurta määrää mutta tälläkin datan määrällä saadaan aikaan hyvä hajonta vuokrauksien suhteen aikaan ja säähän suhteutettuna. Oheisesta graafista (Kuva 3) näkee hyvin, miten selvästi vuokraukset ajoittuvat työpäivien aikana aamupäivän sekä iltapäivän sekä alkuillan tunteihin. Yksi piste graafissa osoittaa aina yhtä päivää. Päivien värin mukaan on helposti nähtävissä, miten paljon lämpötila vaikuttaa polkupyörien vuokrauksen määrään. Projektissa ei kerrota mistä data on peräisin, lämpötilat ovat koko vuoden plussan puolella ja polkupyörien vuokrausta tarjotaan läpi vuoden.

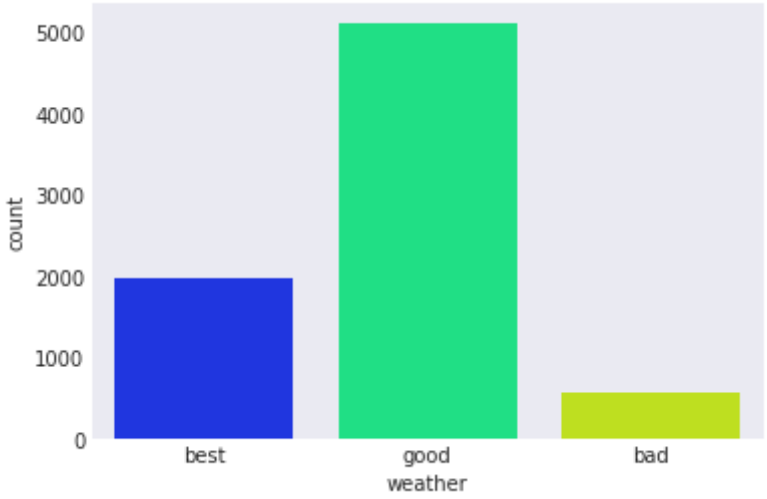


KUVA 3. Polkupyörien arkipäiväkohtainen vuokraus perustuen kellonaikaan ja lämpötilaan

3.2 Toinen malliprojekti

Toiseksi malliprojektiksi valittiin varsin erilainen projekti. Tämän projektin oli tehnyt Melike Dilekci, otsikolla ”Bike Rental Analysis | EDA Guidelines” (https://www.kaggle.com/code/melikedilekci/uci-bike-rental-data-set#Bike-Rental-Analysis). Tässä projektissa paneuduttiin enemmän datan käsittelyyn kuin ennustamiseen. Ohjatussa oppimisessa data on todella tärkeässä roolissa. Tämän projektin tehtävänä on ottaa selvää, miten datan eri arvoja voidaan käyttää. Olemassa olevasta

datasta saadaan tulostettua erilaisia graafeja. Näiden tulosten perusteella voidaan koittaa löytää mielenkiintoisia käyttökohteita omalle datalle, omassa projektissa.



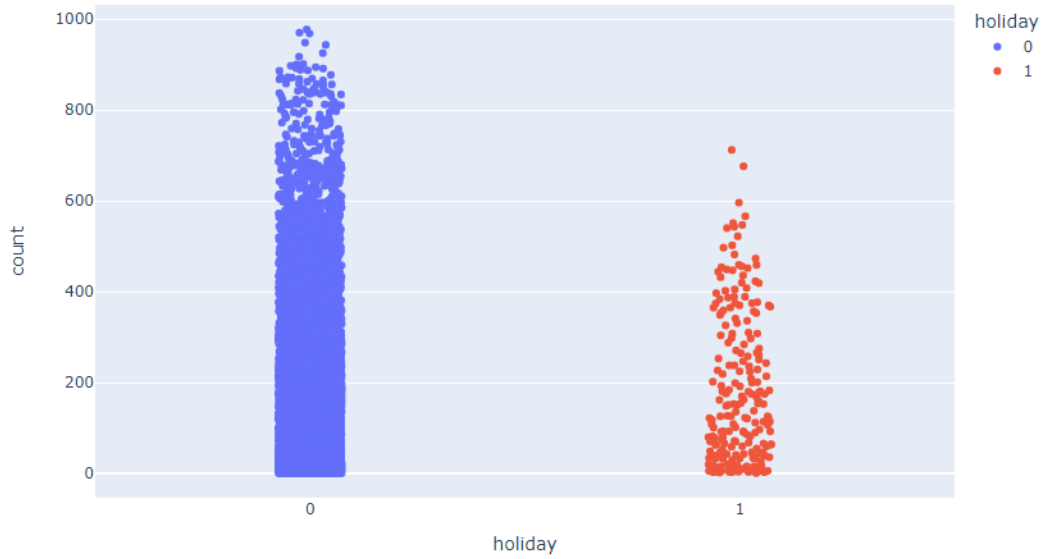
KUVA 4. Graafissa näkyy vuokrausten määrä sään mukaan

Projektissa oli keskitytty tutkimaan vuokrausten määrää vallitsevan sään ja vuodenajan mukaan. Kuva 4 osoittaa, miten data jakaantui sään perusteella. Sää oli määritelty kolmeen kategoriaan riippuen lämpötilasta, sateen määrästä ja tuulen nopeudesta.

KUVA 5. Vuokrausten määrä vuodenajan ja säätilan mukaan

Kuvassa 5 nähdään, miten eri vuodenaikoina pyörien vuokraukset osuvat vallitsevan säätilan mukaan. Näistä tuloksista on hyvin nähtävissä, kuinka paljon säätila vaikuttaa vuokrausten määrään. Aineistossa ei kerrota, mistä tämä data on peräisin. Data jakaantuu tasaisesti vuodenaikojen suhteen ja lämpötila pysyy samana läpi vuoden. Tämän perusteella data on peräisin mahdollisesti päiväntasaajan lähettyviltä.

Kuvassa 6 on vielä tulostettu graafiin vuokrausten määrä arkipäivinä ja pyhäpäivinä. Tästä graafista nähdään suoraan, että pyörien vuokrauksia on arkisin huomattavasti enemmän. Tästä voi päätellä, että suuri osa vuokrauksista on työmatkalaisten tekemiä.



KUVA 6. Vuokrausten määrä arkisin sekä viikonloppuisin tai pyhäpäivinä

4 TUTKIMUKSEN DATA

Tutkimuksessa käytettävä data on City Bike Finlandin omistamaa. Data on avoimesti saatavilla osoitteista:

- https://dev.hsl.fi/citybikes/od-trips-2021/2021-05.csv
- https://dev.hsl.fi/citybikes/od-trips-2021/2021-06.csv
- https://dev.hsl.fi/citybikes/od-trips-2021/2021-07.csv

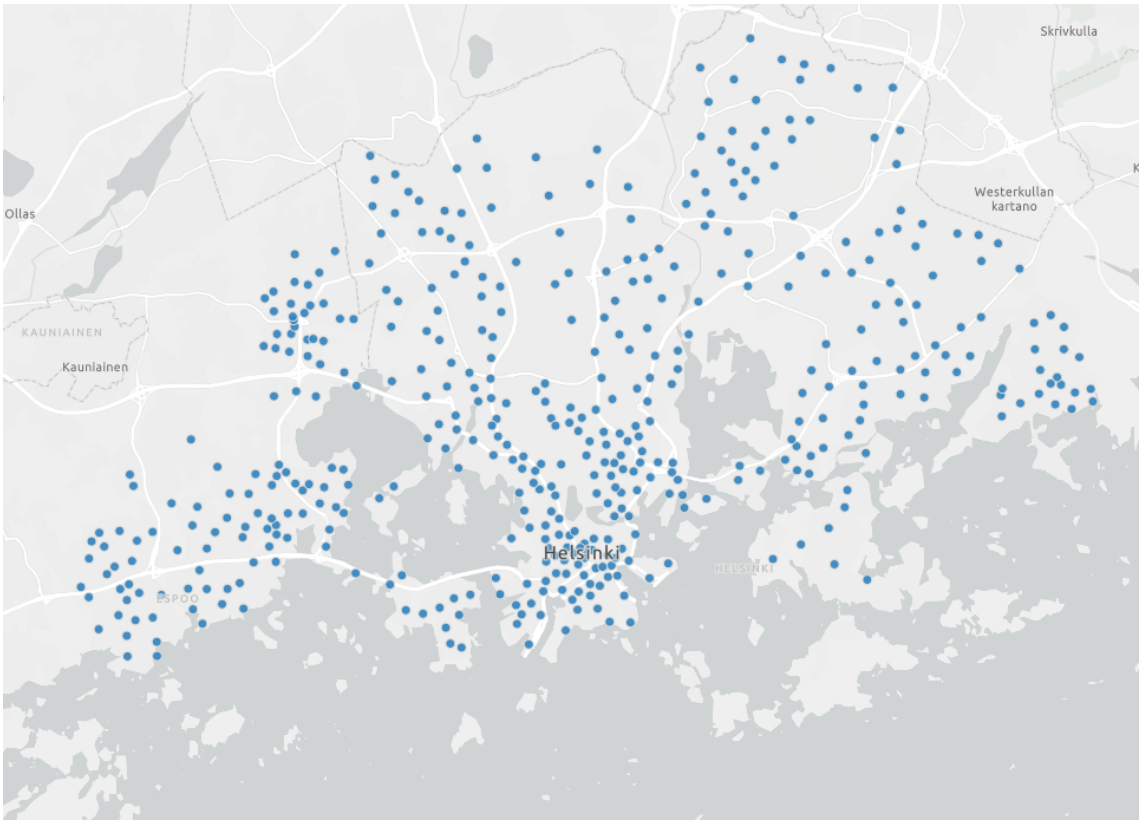
Näissä tiedostoissa on kaikki yhteiskäyttöpolkupyörien vuokraukset kuukausikohtaisesti. Jokainen vuokraus on oma rivinsä ja sisältää lähtöajan sekä lähtöaseman, kuljetun matkan ja vuokrauksen keston. Kuvassa 7 näkyy raakadata, ylärivillä muuttujat, jotka itse datassa erotetaan toisistaan pilkulla. Näitä arvoja käyttäen suoritetaan datan analysointi sekä ennustusten toteuttaminen.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Departure,Return,Departure station id,Departure station name,Return station id,Return station name,Covered distance (m),Duration (sec.)													
2	2021-05-31T23:57:25,2021-06-01T00:05:46,094,Laajalahden aukio,100,Teljäntie,2043,500													
3	2021-05-31T23:56:59,2021-06-01T00:07:14,082,Töölöntulli,113,Pasilan asema,1870,611													
4	2021-05-31T23:56:44,2021-06-01T00:03:26,123,Näkinsilta,121,Vilhonvuorenkatu,1025,399													
5	2021-05-31T23:56:23,2021-06-01T00:29:58,004,Viiskulma,065,Hernesaarenranta,4318,2009													
6	2021-05-31T23:56:11,2021-06-01T00:02:02,004,Viiskulma,065,Hernesaarenranta,1400,350													
7	2021-05-31T23:54:48,2021-06-01T00:00:57,292,Koskelan varikko,133,Paavalinpuisto,1713,366													
8	2021-05-31T23:54:11,2021-06-01T00:17:11,034,Kansallismuseo,081,Stenbäckinkatu,2550,1377													
9	2021-05-31T23:53:04,2021-06-01T00:14:52,240,Viikin normaalikoulu,281,Puotila (M),5366,1304													
10	2021-05-31T23:52:03,2021-06-01T00:15:16,116,Linnanmäki,117,Brahen puistikko,3344,1393													
11	2021-05-31T23:50:19,2021-06-01T00:05:58,116,Linnanmäki,145,Pohjolankatu,3248,935													
12	2021-05-31T23:50:05,2021-06-01T00:01:22,147,Käpylän asema,232,Oulunkylän asema,1633,672													
13	2021-05-31T23:50:00,2021-05-31T23:55:48,069,Kalevankatu,062,Välimerenkatu,1131,345													
14	2021-05-31T23:49:59,2021-05-31T23:59:49,147,Käpylän asema,232,Oulunkylän asema,1695,589													
15	2021-05-31T23:49:59,2021-05-31T23:55:38,069,Kalevankatu,062,Välimerenkatu,1125,336													
16	2021-05-31T23:49:36,2021-06-01T00:40:20,547,Jämeräntaival,547,Jämeräntaival,1227,3040													
17	2021-05-31T23:49:18,2021-06-01T00:05:09,201,Länsisatamankuja,041,Ympyrätalo,4245,948													
18	2021-05-31T23:48:53,2021-06-01T00:03:49,030,Itämerentori,050,Melkonkuja,2656,892													
19	2021-05-31T23:48:44,2021-05-31T23:56:06,235,Katariina Saksilaisen katu,239,Viikin tiedepuisto,2107,437													
20	2021-05-31T23:47:49,2021-05-31T23:51:11,727,Ratsutori,713,Upseerinkatu,549,198													
21	2021-05-31T23:46:14,2021-05-31T23:55:58,137,Arabian kauppakeskus,044,Sörnäinen (M),1970,582													
22	2021-05-31T23:46:13,2021-05-31T23:55:22,137,Arabian kauppakeskus,118,Fleminginkatu,1952,544													
23	2021-05-31T23:45:26,2021-06-01T00:12:04,264,Eränkävijäntori,267,Roihupelto,3134,1597													
24	2021-05-31T23:45:15,2021-05-31T23:51:54,049,Annankatu,162,Leppäsuonaukio,1186,394													
25	2021-05-31T23:45:09,2021-05-31T23:49:23,063,Jätkäsaarenlaituri,068,Albertinkatu,1841,249													
26	2021-05-31T23:44:27,2021-05-31T23:49:46,727,Ratsutori,711,Kirjurinkuja,974,314													

KUVA 7. Vuokrauksista talteen otetut tiedot raakadatan muodossa

Kaupunkipyörien asemien data on HSL:n omistamaa avointa dataa ja löytyy osoitteesta https://www.avoindata.fi/data/en_GB/dataset/hsl-n-kaupunkipyoraasemat/resource/a23eef3a-cc40-4608-8aa2-c730d17e8902

Linkki mistä näkyy kaikki pyöräasemat kartalla: https://public-transport-hslhrt.opendata.arcgis.com/datasets/helsingin-ja-espoon-kaupunkipy%C3%B6r%C3%A4asemat-avoin/explore. Kuvassa 8 näkyvät kartalla kaikki Helsingissä ja Espoossa sijaitsevat kaupunkipyörien asemat.



KUVA 8. Kaikki pyöräasemat kartalla, Espoossa sekä Helsingissä

Datan määrä on varsin suuri, yli kolme miljoonaa riviä kolmen kuukauden ajalta. Data on peräisin vuodelta 2021 touko-, kesä- sekä heinäkuulta. Vääristymien vuoksi dataa on hieman siivottu ja poistettu lyhyet vuokraukset, joissa pyörä on palautettu samalle asemalla. Yli kolmen tunnin vuokraukset myös poistettiin, joissakin tapauksissa ei ollut vuokraus katkennut ollenkaan koko kesän aikana.

Säätietojen data saatiin Ilmatieteen laitoksen avoimesta datasta: https://www.ilmatieteenlaitos.fi/avoin-data. Kaikki data kerättiin MySQL-tietokantaan.

5 TUTKIMUKSEN TOTEUTUS

Tutkimusta varten tehtiin sovellus datan tutkimista varten. Sovellus toteutettiin käyttäen back-endiin MySQL-tietokantaa sekä Node.js komentoja varten. Back-end tarkoittaa sovelluksen palvelinpuolta, joka vastaa tietojen tallennuksesta ja käsittelystä. Front-end toteutettiin käyttäen Reactia. Front-end puolestaan tarkoittaa sovelluksen käyttöliittymää, käyttäjälle selaimessa näkyvää näkymää. Tutkimuksessa oli käytössä suuri määrä dataa kaupunkipyörien vuokrauksista. Tarkoituksena oli tutkia dataa ja mahdollisesti pystyä tekemään ennusteita pyörien vuokrausmäärien suhteen. Vuokrauksia käsitellään sekä vuokrauksen lähtöpaikan että palautuspaikan suhteen. Vuokrauksia ja palautuksia on jokaisella asemalla eri määrä päivittäin. Ennusteella pyritään saamaan tuloksia, joiden perusteella tiedetään mistä asemilta pitää hakea pyöriä pois ja minne niitä pitää kuljettaa lisää.

5.1 Projektin back-end

Dataa oli käytössä todella paljon, yli kolme miljoonaa riviä. Suuren datamäärän ansiosta saadaan luotua tarkempia malleja. Sovellusta ajettaessa tosin huomattiin pian, että datan suuren määrän takia tulosten hakeminen saattoi kestää useita minuutteja. Tulosten nopeampaa läpikäymistä varten ynnättiin yhteen pyörävuokraukset, jotka lähtivät samalta asemalta ja palautettiin tietylle asemalle.

Data koostui miljoonista riveistä pyörävuokrauksia. Datan jokainen rivi sisälsi paljon mielenkiintoisia arvoja. Tärkeimpinä arvoina pidettiin vuokrauksen aikaa sekä sitä, mistä vuokraus alkoi ja mihin se päättyi. Tietokantaan lisättiin myöhemmin vielä ilmatieteenlaitoksen säätiedot. Nyt saatiin vuokraustietojen lisäksi jokaiselle vuokraukselle sinä päivänä olleet säätiedot. Tietokantoja pyrittiin muokkaamaan projektin kuluessa selkeämmäksi, jotta tuloksia saataisiin nopeammin ja tarkemmin näkyville.

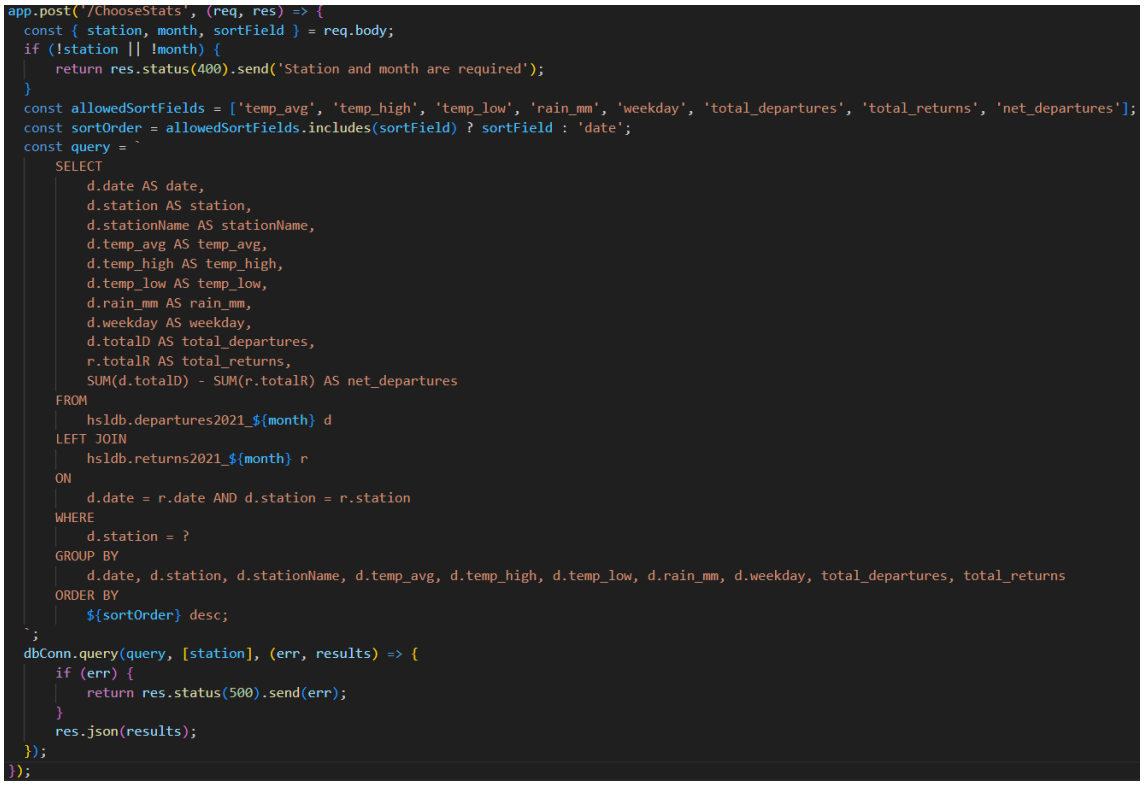
Alkuun MySQL-komennot olivat varsin lyhyitä ja näillä vain kaivettiin tietokannasta erilaisia tietoja. Data-analytiikan avulla pohdittiin tapoja, joilla tärkeimmät tiedot tuotaisiin esille sql-komentojen avulla. Näitä tuloksia voitaisiin käyttää jatkossa vuokrausten ennakoinnissa. Kuva 9 sisältää

käskyn, jolla haetaan tietokannasta kaikki vuokrausasemat ja tulostetaan aseman ID-numero, nimi ja osoite. Tietokannassa on HSL kaupunkipyörien kaikki 457 asemaa.

```
//Get all the stations from database table stations and print id, name, address
app.get('/Stations', function(req, res) {
    dbConn.getConnection(function() {
        dbConn.query('select distinct ID, Nimi, Osoite from stations ORDER BY ID ASC', function (error, results) {
            if (error) throw error;
            console.log("Stations fetched");
            res.send(results);
        })
    })
})
```

KUVA 9. Komento millä haetaan tietokannasta kaikki vuokrausasemat

Projektin edetessä pystyttiin tekemään monimutkaisempia hakuja tietokannasta ja koitettiin saada näillä tarkkoja tietoja tulostettua. Komennoista tuli koko ajan pidempiä, kun koitettiin tarkentaa hakuja täsmällisemmiksi. Erilaisia komentoja kokeiltiin tulostamaan jotain tiettyjä arvoja ja näistä pyrittiin löytämään mielenkiintoisia tietoja käytettäväksi. Lopulta komentoja yhdisteltiin ja saatiin aikaan kuvan 10 kaltainen komento. Tällä pystyttiin selaimesta valitsemaan asema, kuukausi ja tärkeimpien muuttujien joukosta haluttu arvo. Tämän komennon avulla pystyi nopeasti tutkimaan jatkuvuuksia tietyillä asemilla sekä sitä, miten erilaiset muuttujat vaikuttivat vuokrausten määrään. Näiden tulosten perusteella lähdettiin pohtimaan vuokrausten ennustusten toteuttamista.



```
app.post('/ChooseStats', (req, res) => {
  const { station, month, sortField } = req.body;
  if (!station || !month) {
      return res.status(400).send('Station and month are required');
  }
  const allowedSortFields = ['temp_avg', 'temp_high', 'temp_low', 'rain_mm', 'weekday', 'total_departures', 'total_returns', 'net_departures'];
  const sortOrder = allowedSortFields.includes(sortField) ? sortField : 'date';
  const query = `
      SELECT
          d.date AS date,
          d.station AS station,
          d.stationName AS stationName,
          d.temp_avg AS temp_avg,
          d.temp_high AS temp_high,
          d.temp_low AS temp_low,
          d.rain_mm AS rain_mm,
          d.weekday AS weekday,
          d.totalD AS total_departures,
          r.totalR AS total_returns,
          SUM(d.totalD) - SUM(r.totalR) AS net_departures
      FROM
          hsldb.departures2021_${month} d
      LEFT JOIN
          hsldb.returns2021_${month} r
      ON
          d.date = r.date AND d.station = r.station
      WHERE
          d.station = ?
      GROUP BY
          d.date, d.station, d.stationName, d.temp_avg, d.temp_high, d.temp_low, d.rain_mm, d.weekday, total_departures, total_returns
      ORDER BY
          ${sortOrder} desc;
  `;
  dbConn.query(query, [station], (err, results) => {
      if (err) {
          return res.status(500).send(err);
      }
      res.json(results);
  });
});
```

KUVA 10. Komento monimutkaisten tietojen hakemiseen tietokannasta

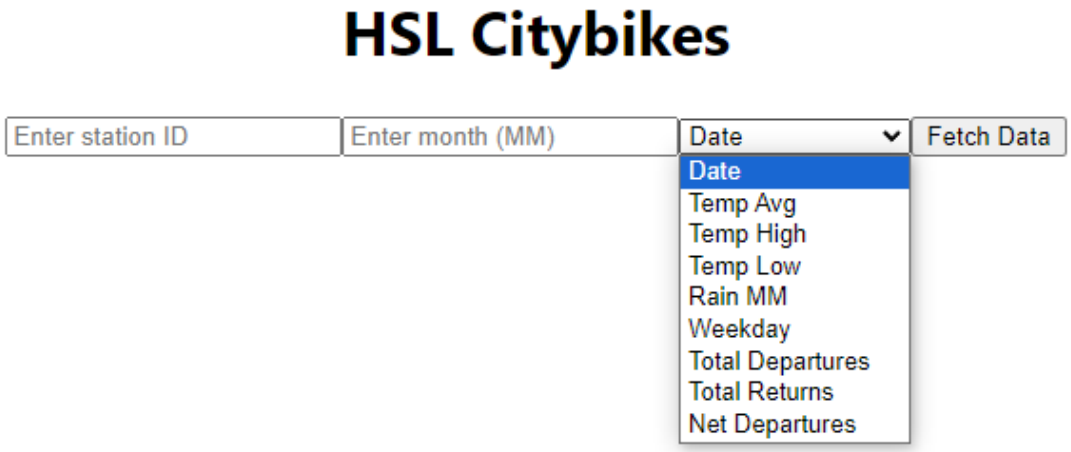
5.2 Projektin front-end Reactilla

Sovelluksessa saatiin tietokannasta tehdyt haut näkyviin Reactin avulla selaimeen. Monia eri komentoja käyttäen pyrittiin saamaan mahdollisimman käytännöllisiä tuloksia mallia varten. Alkuun keskityttiin tutkimaan yksittäisiä vuokraustapahtumia. Yksittäisistä vuokrauksista luotiin tuloksia vuokrausten keston suhteen ja miten pitkä matka pyörällä oli kuljettu vuokrauksen aikana. Asemia tutkittiin myös palautusten ja vuokrausten asemakohtaisten määrien suhteen. Saatiin paljon mielenkiintoisia tuloksia mutta ennustavan mallin suhteen yksittäisillä tuloksilla ei ole mitään merkitystä. Suuren datamäärän tutkiminen data analytiikan avulla on mielenkiintoista ja erilaisia hakuja voidaan suorittaa todella paljon, käyttäen erilaisia muuttujia.

Dataa tutkittaessa tarkemmin otettiin käsittelyyn samanaikaisesti useampia muuttujia, saatiin käytännöllisiä tuloksia mallia varten. Näitä tuloksia läpikäydessä todettiin monia erilaisia tapoja datan hyödyntämiseksi. Vuokrausten määrän perusteella voitiin huomata, että miltä asemalta vuokrataan enemmän pyöriä kuin sinne palautetaan tai palautetaan enemmän kuin vuokrataan. Tätä tietoa käyttämällä pystytään ennustamaan, minne pyöriä tulee kuljettaa asemilta mistä niitä vuokrataan

enemmän kuin palautetaan ja toisinpäin. Tilastoja katsomalla nähdään, että usealta asemalta vuokrataan huomattavasti enemmän pyöriä kuin mitä sinne palautetaan päivittäin. Tämän vuoksi pyöriä tulee päivittäin siirtää asemien väleillä suuriakin määriä.

Kun säätiedot oli lisätty tietokantaan niin hieman yllättäen vuokrausten määrä pysyi arkisin lähes samana kelistä riippumatta. Tietokannasta haettiin tietoja käyttäen Node.js:n kautta MySQL komentoja. Näitä komentoja tehtiin kymmeniä, jotta saatiin mahdollisimman kiinnostavia tuloksia. Kuvassa 11 näkyy sovelluksen hakusivu, jolla saatiin tehtyä asemakohtaisesti erilaisia hakuja eri muuttujilla. Kuvassa 12 näkyy tietyltä asemalta lähtöjen lukumäärä kuukauden eri päivinä. Tähän valittiin asema numero 123, Näkinsilta Helsingin Sörnäisissä, asema on varsin vilkas ja vuokrausten sekä palautusten määrä on suuri päivittäin.



KUVA 11. Selaimesta näkymä millä voitiin tehdä hakuja asemakohtaisesti

HSL Citybikes

123 | 06 | Total Departures | Fetch Data

Date	Station	Station Name	Temp Avg	Temp High	Temp Low	Rain MM	Weekday	Total Departures	Total Returns	Net Departures
09.06.2021	123	Näkinsilta	17.9	22.3	13.6	-1	Wednesday	400	352	48
15.06.2021	123	Näkinsilta	16.2	20.2	14	-1	Tuesday	376	358	18
22.06.2021	123	Näkinsilta	25.1	28.8	22.1	-1	Tuesday	360	380	-20
17.06.2021	123	Näkinsilta	16	20.5	11.5	-1	Thursday	336	328	8
18.06.2021	123	Näkinsilta	20	25.1	14.1	-1	Friday	334	282	52
08.06.2021	123	Näkinsilta	18	21.6	14.2	-1	Tuesday	322	354	-32
21.06.2021	123	Näkinsilta	25.6	29.8	19.7	-1	Monday	314	296	18
29.06.2021	123	Näkinsilta	22.4	26.9	17.1	-1	Tuesday	310	324	-14
16.06.2021	123	Näkinsilta	16.7	20.7	11.2	-1	Wednesday	300	294	6
10.06.2021	123	Näkinsilta	18.7	23	11.8	-1	Thursday	296	288	8
11.06.2021	123	Näkinsilta	19.6	23.8	13.4	-1	Friday	290	332	-42
30.06.2021	123	Näkinsilta	20.2	25.7	16.2	4	Wednesday	280	228	52
12.06.2021	123	Näkinsilta	18.1	21.5	15.8	7	Saturday	272	232	40
24.06.2021	123	Näkinsilta	20.1	22.9	17.7	-1	Thursday	264	268	-4
28.06.2021	123	Näkinsilta	21.5	26.1	16.1	-1	Monday	254	266	-12
23.06.2021	123	Näkinsilta	22.2	28	18.5	11	Wednesday	246	248	-2
14.06.2021	123	Näkinsilta	15.9	19.8	9.5	0	Monday	242	322	-80
20.06.2021	123	Näkinsilta	23.3	28.2	18.1	-1	Sunday	238	232	6
19.06.2021	123	Näkinsilta	21.9	27.8	17.7	-1	Saturday	236	264	-28
07.06.2021	123	Näkinsilta	19.8	25.5	14.5	-1	Monday	210	216	-6
25.06.2021	123	Näkinsilta	20.5	23.9	16.7	-1	Friday	192	156	36
13.06.2021	123	Näkinsilta	15.8	20.8	14.1	0	Sunday	170	162	8
03.06.2021	123	Näkinsilta	16.9	19.8	13.4	-1	Thursday	168	152	16
01.06.2021	123	Näkinsilta	13.4	17.9	9.7	-1	Tuesday	152	149	3
26.06.2021	123	Näkinsilta	20.8	23.9	18	-1	Saturday	150	136	14
02.06.2021	123	Näkinsilta	16.2	21.6	8.1	-1	Wednesday	143	157	-14
05.06.2021	123	Näkinsilta	19.1	23.4	11.7	-1	Saturday	141	143	-2
04.06.2021	123	Näkinsilta	17	21.6	11.9	-1	Friday	140	134	6
27.06.2021	123	Näkinsilta	21	26.2	17.9	-1	Sunday	138	160	-22
06.06.2021	123	Näkinsilta	18.7	26.7	13.4	18	Sunday	114	113	1

KUVA 12. Asemalta 123 lähteneet vuokraukset kesäkuussa 2021

Kuvasta 12 näkyy kaikki muuttujat ja niiden saamat arvot kesäkuussa 2021. Net Departures kertoo miten paljon pyöriä tulisi siirtää asemalle tai sieltä pois. Luvut vaihtelevat todella paljon eri asemien välillä. Joillakin asemilla vuokrausten ja palausten lukemat voivat heitellä suurestikin, kun taas toisilla kaava toistuu hyvin. Kuvassa 13 näemme aseman, jonka vuokrausten ja palautusten määrää voidaan käyttää tulosten perusteella hyväksi pyörien kuljetuksen suhteen asemien välillä.

HSL Citybikes

012 | 07 | Net Departures | Fetch Data

Date	Station	Station Name	Temp Avg	Temp High	Temp Low	Rain MM	Weekday	Total Departures	Total Returns	Net Departures
18.07.2021	12	Kanavaranta	22	26.3	18	0	Sunday	382	294	88
04.07.2021	12	Kanavaranta	22.7	26.4	17.3	-1	Sunday	253	206	47
11.07.2021	12	Kanavaranta	23.2	27.1	22.4	-1	Sunday	456	414	42
12.07.2021	12	Kanavaranta	22.4	26	17.7	-1	Monday	408	392	16
03.07.2021	12	Kanavaranta	21.5	25.9	14.9	-1	Saturday	336	321	15
25.07.2021	12	Kanavaranta	19.3	22.7	13.5	-1	Sunday	382	370	12
08.07.2021	12	Kanavaranta	22.8	25.4	20.7	1	Thursday	367	355	12
29.07.2021	12	Kanavaranta	20.3	23.1	18.8	-1	Thursday	312	300	12
20.07.2021	12	Kanavaranta	17.2	21.6	13.5	1	Tuesday	250	240	10
01.07.2021	12	Kanavaranta	21.8	27.2	17.8	-1	Thursday	238	232	6
10.07.2021	12	Kanavaranta	25.6	28.7	21.1	-1	Saturday	588	586	2
14.07.2021	12	Kanavaranta	25.6	30.1	20	-1	Wednesday	434	434	0
28.07.2021	12	Kanavaranta	20	23.7	18.3	9	Wednesday	216	216	0
13.07.2021	12	Kanavaranta	24.1	28.8	17.7	-1	Tuesday	496	498	-2
05.07.2021	12	Kanavaranta	23.6	27.1	19.6	-1	Monday	199	201	-2
07.07.2021	12	Kanavaranta	23.3	26.4	20.6	3	Wednesday	257	260	-3
23.07.2021	12	Kanavaranta	18.4	22.5	13.3	-1	Friday	350	360	-10
27.07.2021	12	Kanavaranta	23.3	28.7	17.6	11	Tuesday	384	398	-14
22.07.2021	12	Kanavaranta	18.9	25.1	12.6	-1	Thursday	338	354	-16
19.07.2021	12	Kanavaranta	18.1	22	14.5	-1	Monday	342	360	-18
06.07.2021	12	Kanavaranta	23.3	26.8	21.3	2	Tuesday	175	197	-22
26.07.2021	12	Kanavaranta	22.6	27.7	16	-1	Monday	360	386	-26
16.07.2021	12	Kanavaranta	25.1	29.8	21.7	-1	Friday	448	482	-34
30.07.2021	12	Kanavaranta	16.8	20.5	14.4	9	Friday	112	146	-34
21.07.2021	12	Kanavaranta	16.9	21.4	12	-1	Wednesday	342	388	-46
15.07.2021	12	Kanavaranta	26.4	30.2	20.3	-1	Thursday	426	478	-52
31.07.2021	12	Kanavaranta	17.4	21	14.2	3	Saturday	320	372	-52
02.07.2021	12	Kanavaranta	20.3	24.1	16.8	-1	Friday	248	300	-52
17.07.2021	12	Kanavaranta	22	26.6	16.2	-1	Saturday	548	604	-56
24.07.2021	12	Kanavaranta	17.1	20.4	12.4	-1	Saturday	448	518	-70
09.07.2021	12	Kanavaranta	21.7	25.5	17.6	-1	Friday	420	512	-92

KUVA 13. Heinäkuussa 2021 lajittelu vuokrausten ja palausten erotuksen suhteen asemalta 12

Saatavilla olevaa dataa hyödyntämällä ei ole mahdollista tietää onko asemalla vapaita pyöriä vuokrattavaksi. Jotkin asemat ovat tyhjinä ja tällöin vuokrausten määrään tulee poikkeuksia. Mitä enemmän dataa tietokannasta löytyy syötettynä ohjatulla oppimisella niin sitä tarkempia tuloksia voimme tehdä, sekä ennustuksia tulevaisuuden vuokrausten määrälle. Jo pelkästään näitä tuloksia katsomalla pystymme määrittelemään hyvin miltä asemilta ja mille asemille pyöriä tulisi minäkin viikonpäivänä kuljettaa.

Jotta saamme paremman kuvan vuokrausten määrästä asemakohtaisesti, loimme kaavan, jolla saimme vuokraukset ja palautukset valitulta viikonpäivältä. Kaava laski sitten yhteen jokaiselta kuukauden päivältä, jokaiselle asemalle vuokraukset ja palautukset. Pystyimme käymään läpi jokaisen viikonpäivän erikseen ja tutkimaan tilastoja koskien eri viikonpäiviä. Kuvassa 14 on tulostettuna asemat, joilta on 2021 toukokuussa tiistaisin vuokrattu eniten pyöriä suhteessa palautettuihin

pyöriin. Net Departures näyttää tässä positiivisen luvun, josta näemme, että asemalta lähti enemmän pyöriä, kuin mitä sinne palautettiin. Kuvassa 15 on nähtävillä 2021 toukokuun asemat, joille tiistaisin palautettiin eniten pyöriä suhteutettuna vuokrattuihin pyöriin. Negatiivinen Net Departures arvo tässä tarkoittaa, että asemalle palautettiin enemmän pyöriä, kuin mitä sieltä vuokrattiin. Näillä arvoilla on jo hyvin nähtävissä johdonmukaisuuksia vuokrauksien ja palautusten suhteen asemakohtaisesti. Arkipäivien ja viikonloppujen välisten vuokrausten ero on huomattava lähes kaikilla asemilla. Säätilan muutokset vaikuttavat vuokrausten määrään joillakin asemilla huomattavasti enemmän kuin toisilla.

HSL Citybikes

Desc order | May | Tuesday | Fetch Data

Date	Station	Station Name	Temp Avg	Rain MM	Weekday	Total Departures	Total Returns	Net Departures
18.05.2021	113	Pasilan asema	12.6	5	Tuesday	380	252	128
04.05.2021	113	Pasilan asema	7.1	-1	Tuesday	238	132	106
25.05.2021	113	Pasilan asema	11	0	Tuesday	394	312	82
18.05.2021	45	Brahen kenttä	12.6	5	Tuesday	206	124	82
11.05.2021	113	Pasilan asema	15.7	-1	Tuesday	388	324	64
25.05.2021	41	Ympyrätalo	11	0	Tuesday	452	398	54
18.05.2021	6	Hietalahdentori	12.6	5	Tuesday	202	150	52
04.05.2021	9	Erottajan aukio	7.1	-1	Tuesday	134	84	50
11.05.2021	116	Linnanmäki	15.7	-1	Tuesday	292	246	46
18.05.2021	44	Sörnäinen (M)	12.6	5	Tuesday	222	180	42
25.05.2021	129	Pernajantie	11	0	Tuesday	134	92	42
04.05.2021	44	Sörnäinen (M)	7.1	-1	Tuesday	170	132	38
25.05.2021	149	Toinen linja	11	0	Tuesday	148	110	38
25.05.2021	49	Annankatu	11	0	Tuesday	198	162	36
18.05.2021	86	Kuusitie	12.6	5	Tuesday	116	84	32

KUVA 14. Viikonpäivän mukaan nettopalautukset

HSL Citybikes

Asc order | May | Tuesday | Fetch Data

Date	Station	Station Name	Temp Avg	Rain MM	Weekday	Total Departures	Total Returns	Net Departures
18.05.2021	11	Unioninkatu	12.6	5	Tuesday	74	144	-70
25.05.2021	138	Arabiankatu	11	0	Tuesday	140	202	-62
11.05.2021	126	Kalasatama (M)	15.7	-1	Tuesday	476	530	-54
11.05.2021	65	Hernesaarenranta	15.7	-1	Tuesday	144	198	-54
25.05.2021	3	Kapteeninpuistikko	11	0	Tuesday	136	188	-52
18.05.2021	126	Kalasatama (M)	12.6	5	Tuesday	350	400	-50
25.05.2021	731	Leppävaarankäytävä	11	0	Tuesday	78	128	-50
11.05.2021	11	Unioninkatu	15.7	-1	Tuesday	202	248	-46
18.05.2021	39	Ooppera	12.6	5	Tuesday	112	158	-46
25.05.2021	5	Sepänkatu	11	0	Tuesday	194	238	-44
25.05.2021	71	Hietaniemenkatu	11	0	Tuesday	188	232	-44
25.05.2021	11	Unioninkatu	11	0	Tuesday	142	186	-44
11.05.2021	161	Eteläesplanadi	15.7	-1	Tuesday	144	186	-42
11.05.2021	591	Mellstenintie	15.7	-1	Tuesday	82	124	-42
25.05.2021	22	Rautatientori / länsi	11	0	Tuesday	294	332	-38

KUVA 15. Viikonpäivän mukaan nettopalautukset

Tämän perusteella oli luotu kaava, jolla pystyi tutkimaan vuokrauksia viikonpäivän suhteen. Tätä kaavaa hyödyntäen mallia kehitettiin eteenpäin ja laskettiin kuukaudelta kaikki vuokraukset ja palautukset yhteen jokaiselta kuukauden samalta viikonpäivältä. Mallin kehittämistä jatkettiin ja laskettiin keskiarvo kuukauden samoille viikonpäiville. Tällä luotiin lopputulos, jossa näkyi kuukauden halutun viikonpäivän keskiarvo. Kuvassa 16 näkyy tämän kaavan luoma lopputulos. Vasemmalla näkyy 2021 toukokuun tiistaiden keskiarvo asemille, joihin on palautettu enemmän pyöriä kuin mitä niiltä on vuokrattu (Bike Balance). Oikealla puolella on taas kuukauden keskiarvot asemille, joilta on vuokrattu enemmän pyöriä kuin mitä niille on palautettu. Tällä mallilla pystytään luomaan kuva, joka mahdollistaa eri viikonpäivinä ennakoimaan pyörien kuljetuksen täydeltä asemalta vajaalle asemalle.

Station	Station Name	Weekday	Bike Balance	Station	Station Name	Weekday	Bike Balance
11	Unioninkatu	Tuesday	39	113	Pasilan asema	Tuesday	-95
126	Kalasatama (M)	Tuesday	33.5	44	Sörnäinen (M)	Tuesday	-33
731	Leppävaarankäytävä	Tuesday	27.5	116	Linnanmäki	Tuesday	-29
94	Laajalahden aukio	Tuesday	24.5	109	Hertanmäenkatu	Tuesday	-24.5
161	Eteläesplanadi	Tuesday	21	45	Brahen kenttä	Tuesday	-18.5
64	Tyynenmerenkatu	Tuesday	17	86	Kuusitie	Tuesday	-18.5
65	Hernesaarenranta	Tuesday	15	129	Pernajantie	Tuesday	-17
201	Länsisatamankuja	Tuesday	14.5	202	Merihaka	Tuesday	-16.5
5	Sepänkatu	Tuesday	14	30	Itämerentori	Tuesday	-15
8	Vanha kirkkopuisto	Tuesday	14	763	Kalkkipellonmäki	Tuesday	-13.5
134	Haukilahdenkatu	Tuesday	13.5	66	Ehrenströmintie	Tuesday	-13
122	Lintulahdenkatu	Tuesday	13	9	Erottajan aukio	Tuesday	-11.5
95	Munkkiniemen aukio	Tuesday	13	529	Keilaniemi (M)	Tuesday	-11.5
77	Nordenskiöldinaukio	Tuesday	13	6	Hietalahdentori	Tuesday	-11
20	Kaisaniemenpuisto	Tuesday	12.5	749	Vallikatu	Tuesday	-11

KUVA 16. 2021 toukokuun tiistaiden keskiarvon asemakohtaiset nettopalautukset

5.3 Tutkimuksen ennustava malli

Tutkimuksen ennustava malli tehdään Python-kielellä. Data otetaan käyttöön omalla Linux-palvelimella pyörivästä MySQL-tietokannasta.

Ensimmäinen ennustava malli

Tutkimuksessa kun oli tutkittu kattavasti erilaisia tuloksia data analytiikan avulla, oli aika siirtyä ennustavan mallin tekemiseen. Ennustavaa mallia alettiin rakentamaan Python-kielen avulla. Kokemukset Python-kielen käytöstä olivat varsin heikot, joten tulosten saaminen oli suuren työn takana. Ensimmäinen askel oli saada toteutettua jonkunlainen haku, jolla saisimme yksinkertaisen tuloksen seuraavan päivän vuokrauksista valitulta asemalta. Ensimmäiset kokeilut tehtiin muokatulla tietokannalla. Data oli tiivistetty taulukkoon missä oli laskettu kaikki päivän vuokraukset yhteen jokaista asemaa kohti. Alkuperäisellä toukokuun 2021 taulukolla oli 767 894 riviä. Nyt kun data tiivistettiin, niin rivimäärä oli enää 11 738.

```
import pymysql
import pandas as pd
from sklearn.linear_model import LinearRegression
import numpy as np
connection = pymysql.connect(
    host='localhost',
    user='xxx',
    password='xxx',
    db='hsldb'
)
query = """
    SELECT
        d.date AS date,
        d.station AS station,
        d.stationName AS stationName,
        d.temp_avg AS temp_avg,
        d.temp_high AS temp_high,
        d.temp_low AS temp_low,
        d.rain_mm AS rain_mm,
        d.weekday AS weekday,
        d.daynumber AS daynumber,
        d.totalD AS total_departures
    FROM
        hsldb.departures2021_05 d
    WHERE
        d.station = %s
    ORDER BY
        d.date ASC
"""
station_id = '100'  # Tähän syötettiin suoraan koodiin haluttu asema
df = pd.read_sql(query, connection, params=(station_id,))
connection.close()
# Alustetaan data mallia varten
df['date'] = pd.to_datetime(df['date'])
df = df.set_index('date')
df['total_departures_next_day'] = df['total_departures'].shift(-1)
df = df.dropna()
X = df[['temp_avg', 'temp_high', 'temp_low', 'rain_mm']]
y = df['total_departures_next_day']

model = LinearRegression()
model.fit(X, y)

tomorrow_features = df[['temp_avg', 'temp_high', 'temp_low', 'rain_mm']].iloc[-1].values.reshape(1, -1)
predicted_departures = model.predict(tomorrow_features)
print(f"Predicted departures for tomorrow: {predicted_departures[0]:.2f}")
```

KUVA 17. Pythonilla tehty ensimmäinen ennuste

Kuvassa 17 on näkyvillä ensimmäisen koodin, jolla toteutetaan malli, joka ennusti halutun aseman seuraavan päivän vuokraukset olemassa olevilla vuokrausmäärillä. Tämä oli vielä varsin yksinkertainen haku, mutta tämän avulla pystyttiin luomaan ensimmäinen ennuste. Kuva 18 näyttää tuloksen käyttämällä tätä koodia ennusteen luomiseen. Malli näyttää olemassa olevan datan mukaan tehdyn ennustuksen huomisen vuokrausten määrälle valitulta asemalta 100. Käyttäjältä ei vielä tässä vaiheessa kysytä mitään ja aseman numerokin on syötetty suoraan koodiin mukaan.

```
Arvioidut pyörien vuokraukset huomenna asemalta 100: 43.89
>>>
```

KUVA 18. Asemalle 100 tehdyn ennusteen tulos

Malli perustui vaan olemassa olevan datan mukaan tehtyyn ennustukseen. Tässä vaiheessa on vielä kyseessä pääasiallisesti datan analysoiminen. Mallia pitää muokata pyytämään käyttäjältä muuttujia, joiden perusteella voimme tehdä tarkempia ennustuksia. Tietokannassa on valmiina jokaiselle päivälle säätiedot, näitä tietoja hyödyntämällä on mahdollista luoda tarkempi malli. Säätietoja hyödyntävässä mallissa pystytään kysymään käyttäjältä huomisen sääennuste. Näiden sääennusteiden avulla toteutetaan huomisen vuokrausten määrän ennustaminen.

Lopullisen mallin luominen

Ensimmäisen mallin luomiseen käytettiin pakattua dataa. Jotta saamme tarkempia tuloksia niin meidän tulee käyttää alkuperäistä dataa mahdollisimman tarkan mallin luomiseksi. Koodia ei tarvitse edes muuttaa hirveästi, jotta pääsemme näkemään tarkempia tuloksia. MySQL komentoa pitää muokata hakemaan tiedot pakkaamattomasta datasta. Seuraava malli hakee tiedot 2021 toukokuun koko datasta, käytössä on kaikki 767 894 ja tulosten pitäisi olla paljon tarkempia kuin pakatulla datalla.

Teemme haun samalle asemalle 100 kuin pakatulla datalla. Pakatun datan ennustus seuraavalle päivälle oli 43.89 vuokrausta ja pakkaamattomalla datalla ennuste on 42.74. Tulokset eivät eroa toisistaan kauhean paljoa, kun muuttujien määrää lisätään erotkin suurenevat.

Malli on vielä varsin kömpelö, asemakin on syötetty suoraan koodiin mukaan. Jotta mallista saadaan mitään hyötyä irti pitää käyttäjän pystyä syöttämään halutut tiedot mallille käsiteltäväksi. Ensimmäisenä lisäämme aseman numeron kysymisen koodiin. Kuvassa 19 näemme koodin, jolla pyydetään käyttäjää syöttämään aseman numero ja etsitään tietokannasta tämän aseman tiedot. Tämän jälkeen muuttujassa station_id kulkee mukana halutun aseman numero.

```
# Kysytään käyttäjältä halutun aseman numero
station_id = int(input("Syötä halutun aseman ID: "))

# Haetaan tietokannasta halutun aseman data
data = fetch_station_data(station_id)
```

KUVA 19. Kysytään käyttäjältä aseman numero

Nyt pystymme tulostamaan mallillamme käyttäjän valitseman aseman seuraavan päivän ennusteen olemassa olevien vuokrausten perusteella. Seuraavaksi pitää ottaa mukaan sään muuttujat. Tietokannassa on mukana menneiden kuukausien säätiedot ilmatieteenlaitoksen tietokannasta. Menneiden säätietojen lisäksi tarvitsemme myös seuraavan päivän säätietojen ennusteen. Käyttäjän pitää siis tietää mitä keliä on seuraavaksi päiväksi meteorologien mukaan luvattu.

Kuvassa 20 näemme koodin, jolla kysymme käyttäjältä seuraavan päivän sääennusteet. Tietokannassa on menneiden päivien säätiedot. Nyt käytämme päivän keskilämpötilaa, muuttuja temp_avg, sekä päivän sademäärää, muuttuja rain_mm, joka kertoo päivittäisen sademäärän millimetreissä. Näiden avulla saadaan luotua mallilla tarkempi arvio vuokrauksien ja palautusten määrästä halutulle asemalle.

```
# Pyydetään käyttäjältä huomisen sääennusteen tiedot
temp_avg = float(input("Syötä huomisen lämpötilan ennuste (ºC): "))
rain_mm = float(input("Syötä huomisen sademäärän ennuste (mm): "))
```

KUVA 20. Pyydetään käyttäjältä huomisen sään ennuste

Säätiedotuksissa ennustetaan päivän korkeimpia lämpötilan arvoja. Projektissa käytetään jatkossa mallin tekemisessä myös tietokannasta löytyvää temp_high muuttujaa, joka sisältää päivän korkeimman lämpötilan. Nyt haku voidaan suorittaa asemalle ja antaa säätiedotuksen mukaiset arvot mallille. Arvoja muuttamalla tulee hyvin erilaisia ennustuksia. Mallin toimivuus voidaan todentaa varsin erilaisilla ennustuksilla sään eri arvojen mukaan. Eri asemilla myös vaihtelut ovat myös suuria.

Kuvassa 21 näemme MySQL-komennon, jota käytämme eri asemien datan hakemiseen. Tässä station_id muuttuja saa arvon käyttäjän syötteestä. Tietokannasta 2021_07 taulusta haetaan kaikki vuokraukset sekä palautukset päivämäärän mukaan. Jokaiselle päivälle otetaan säätiedot 2021_07w taulusta. Tämä MySQL-komento toimii pohjana mallille, olemassa oleva data otetaan käyttöön tämän avulla.

```
SELECT
    d.DepartureDate,
    d.DepartureCount,
    COALESCE(r.ReturnCount, 0) AS ReturnCount,
    w.temp_high,
    w.rain_mm
FROM (
    SELECT
        DATE(Departure) AS DepartureDate,
        COUNT(*) AS DepartureCount
    FROM
        hsldb.2021_07
    WHERE
        DepartureStationId = {station_id}
    GROUP BY
        DATE(Departure)
) AS d
LEFT JOIN (
    SELECT
        DATE(`Return`) AS ReturnDate,
        COUNT(*) AS ReturnCount
    FROM
        hsldb.2021_07
    WHERE
        ReturnStationId = {station_id}
    GROUP BY
        DATE(`Return`)
) AS r ON d.DepartureDate = r.ReturnDate
LEFT JOIN
    hsldb.2021_07w w ON d.DepartureDate = STR_TO_DATE(w.date, '%%d.%%m.%%Y')
ORDER BY
    d.DepartureDate;
```

KUVA 21. MySQL koodi tietyn aseman datan hakemiseen tietokannasta

Liitteessä 1 näemme Python-koodin mitä on käytetty ennustuksen toteuttamiseen käyttäjän antamilla säätilan arvoilla huomiselle. Kuvassa 22 näemme asemalle 113, Pasilan juna-asema, tehdyn haun kaksilla eri sääennusteilla toukokuun 2021 dataa käyttämällä. Ensin on laitettu lämpötilaksi 15 astetta ja ei ollenkaan sadetta. Pyörien ennustetun vuokrausten määrä on noin 417 ja palautusten määrä asemalle noin 408. Nettovuokraukset tässä näyttää lukujen erotuksen, joka on noin 8 vuorokaudessa. Nettovuokrauksien ollessa positiivinen on pyörien vuokrauksia asemalta tehty enemmän kuin pyöriä on palautettu kyseiselle asemalle. Alemmassa kuvassa näemme haun, jossa sääennusteen arvot ovat käännetty toisinpäin, lämpötila nolla astetta ja sademäärä 15 millimetriä vuorokaudessa. Pyörien vuokrausten ennusteen määrä Pasilan juna-asemalta on nyt noin 496 ja

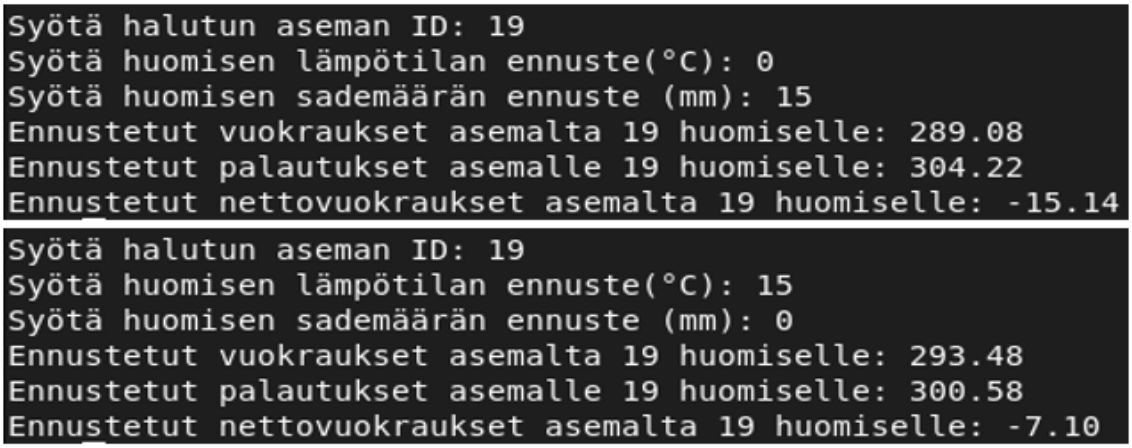
pyörien palautusten määrä asemalle noin 241. Nettovuokraukset ovat tälle ennusteelle 154 vuorokaudessa. Tällä asemalla näemme varsin suuren eron pyörien nettovuokrauksissa erilaisilla sääennusteilla.

```
Syötä halutun aseman ID: 113
Syötä huomisen lämpötilan ennuste(°C): 15
Syötä huomisen sademäärän ennuste (mm): 0
Ennustetut vuokraukset asemalta 113 huomiselle: 416.68
Ennustetut palautukset asemalle 113 huomiselle: 408.47
Ennustetut nettovuokraukset asemalta 113 huomiselle: 8.21
```

```
Syötä halutun aseman ID: 113
Syötä huomisen lämpötilan ennuste(°C): 0
Syötä huomisen sademäärän ennuste (mm): 15
Ennustetut vuokraukset asemalta 113 huomiselle: 495.74
Ennustetut palautukset asemalle 113 huomiselle: 341.50
Ennustetut nettovuokraukset asemalta 113 huomiselle: 154.24
```

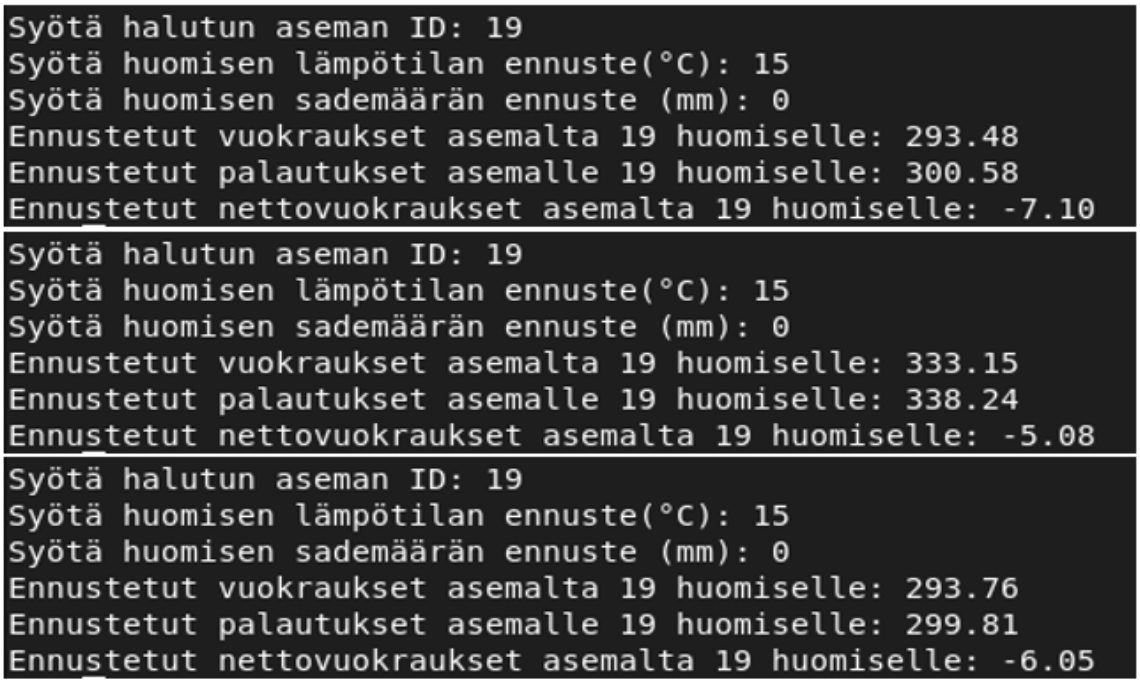
KUVA 22. Asemalle 113 tehty haku eri säätilan ennusteilla

Kun tarkastelemme enemmän pyörien vuokrausasemia niin löydämme varsin erilaisia tuloksia. Kuvassa 23 näemme samalla mallilla tuotetun ennusteen asemalle 19, Helsingin rautatientorin itäinen asema. Tässä haussa käytämme heinäkuun 2021 dataa tietokannastamme. Tällä asemalla on pyörien vuokrausten ja palautusten väliset luvut lähes samat. Ensimmäisessä kuvassa on säätilan ennusteen lämpötila 15 astetta ja sademäärä nolla millimetriä huomiselle. Toisessa kuvassa on säätilan ennusteen lämpötila nolla astetta ja sademäärän ennuste huomiselle 15 millimetriä. Nettopalautukset ovat ensimmäisillä arvoilla -15 ja toisilla ennusteen arvoilla -7. Kun vuokrausten ja palautusten määrä on lähes 300 päivittäin niin tällä asemalla malli ennustaa vuokrausten ja palautusten määrän varsin tarkasti kelistä riippumatta.



KUVA 23. Asemalle 19 tehty haku kaksilla eri säätilan ennusteilla

Nyt olemme tarkastelleet vuokrausten määrää yhden kuukauden dataa käyttämällä. Kun vertailemme kolmen eri kuukauden dataa keskenään, löydämme varsin mielenkiintoisia ennustuksia. Datarivejä oli vuoden 2021 toukokuulta 767 894, kesäkuulta 1 047 507 ja heinäkuulta 1 048 575 riviä. Kuvassa 24 näkyy samoilla sääennusteilla tehty vuokrausten ennuste samalle asemalle kolmen eri kuukauden dataa käyttämällä. Touko-, kesä- ja heinäkuun ennusteiden pyörien vuokrausten ja palautusten määrä on noin 300 paikkeilla joka kuukausi. Nettovuokrausten määrät ovat -7, -5 ja -6. Näemme että Helsingin Rautatientorin asemalla pyörien ennusteen nettovuokraukset ovat lähes samat joka kuukausi.



KUVA 24. Asemalle 19 tehty ennuste samoilla arvoilla kolmelle eri kuukaudelle

Tarkastelluilla asemilla on ollut suhteellisen paljon pyörien vuokrauksia ja palautuksia, näillä asemilla voimme tehdä suhteellisen hyviä ennustuksia. Kun alamme tarkastelemaan asemia, joilla on pyörien vuokrauksia ja palautuksia muutamasta muutamaan kymmeneen niin ennustukset heittelevät kuukausittain kymmenillä prosenteilla. Näille hiljaisille asemille ennustusten tekeminen on lähes mahdotonta. Näitä hiljaisia asemia on suhteellisen paljon, varsinkin kun lähdetään kauemmas Helsingin keskustasta. Isoissa asutuskeskuksissa ja varsinkin juna-asemien ja bussiterminaalien läheisyydessä pyörien vuokrauksia on paljon.

Esimerkkinä yhdestä hiljaisesta asemasta voimme ottaa Siltavoudintien aseman numero 231. Tältä asemalta pyörien vuokrausten määrä on yhdestä kolmeenkymmeneen. Pyörien vuokrausten ja palautusten määrä ei tunnu olevan mitenkään sidottu säätietoihin vuokrausten pienien määrien takia. Hiljaisten asemien nettovuokrauksilla ei sinänsä ole niin suurta merkitystä koska asemilta ei pienen vaihtuvuuden takia pyörät helposti lopu.

5.4 Mallin testaaminen

Mallin toimivuus pitää myös testata, jotta voimme tietää onko tuloksiin luottamista. Aivan ensimmäiseksi kokeiltiin ennustavuutta muokkaamalla tietokannan säätietoja. Valitaan tarkasteluun Kaivopuiston asema, mallilla saamme toukokuun arvoilla seuraavan päivän nettovuokrauksille arvon -27.18. 26.5.2021 on ollut hyvin sateinen päivä ja on satanut 40 millimetriä vettä. Vaihdetaan tähän arvoksi nolla ja katsotaan mitä meidän mallimme kertoo seuraavan päivän ennusteeksi. Sademäärän muutoksen jälkeen saamme mallilla seuraavan päivän ennusteen nettovuokrauksille arvon -29.69. Näemme että mallimme ottaa huomioon päivittäiset säätiedot.

Jotta pääsemme testaamaan malliamme vielä tarkemmin, muokkaamme MySQL komentoamme. Kuvan 25 MySQL-komentoa käyttämällä poistamme käytöstä sunnuntain vuokrausten datan, palautuksista poistamme tietenkin myös samat päivät. Muokatulla MySQL-komennolla saamme Kampin metroaseman pysäkillä mielenkiintoisia arvoja. Käytämme ennustuksen muuttujina 15 astetta ja nolla millimetriä sadetta. Koko tietokantaa käyttämällä saamme toukokuun nettovuokrauksien ennustuksen arvoksi 13.87. Kun poistamme vuokrauksista lauantait, on nettovuokrauksien arvo 13.7, sunnuntait poistamalla arvo on 14.62.

```
FROM
    hsldb.2021_05
WHERE
    DepartureStationId = {station_id}
    AND DATE(Departure) NOT IN ('2021-05-02', '2021-05-09', '2021-05-16', '2021-05-23', '2021-05-30')
```

KUVA 25. Muokattu MySQL komento datan määrän supistamiseksi

Mitä enemmän dataa on tarjolla, sitä tarkempia ennustuksia pystymme tekemään. Vilkkailla asemilla on helpompi tehdä ennustuksia, ne ovat myös paljon tarkempia kuin hiljaisilla asemilla. Kuvassa 23 näimme Helsingin Rautatientorin itäisen aseman vuokraukset kolmelta eri kuukaudelta, mallin tulokset olivat prosentin sisällä. Tätä voimme pitää erittäin hyvänä ennustuksena mallin pohjalta. Muiden asemien kohdalla ei näin pientä hajontaa löytynyt. Säätilat vaikuttivat muilla kokeilluilla asemilla paljon enemmän vuokrausten määrään. Hajonta vaikutti olevan pienintä juna- sekä metroasemien läheisyydessä olevilla pyörien vuokrausasemilla.

5.5 Graafisten tulosten luominen R-kielellä

Jotta dataa saatiin tulostettua myös graafisessa muodossa, otettiin käyttöön R-kieli. R-kieli on erittäin kätevä graafien tulostamiseen suhteellisen helpoilla koodeilla. Kieli on hyvin käytännöllinen ja pienillä muutoksilla saa aikaan näyttäviä erilaisia tuloksia. Kuvassa 26 on R-kielen koodi, jolla haetaan pyörien vuokraukset ja palautukset asemalle 30. Train-muuttujaan laitetaan vuokraukset ja test-muuttujaan palautukset. Näin saadaan tehtyä malli, jonka avulla voidaan tehdä käskyjä graafin piirtämiseen.

```r
library(ggplot2)
library(lubridate)
library(scales)
library(dplyr)
library(tidyr)
# Ladataan data
train <- read.csv("hsldb/Dep05.csv")
test  <- read.csv("hsldb/Ret05.csv")
# Perusarvot
cat("Number of training rows: ", nrow(train), "\n")
cat("Number of test rows: ", nrow(test), "\n")
# Sarakkeiden nimet
print(names(train))
print(names(test))
# Varmistetaan että tarvittavat sarakkeet on mukana
if (!all(c("totalD", "date", "station") %in% names(train))) {
    stop("Missing columns in Dep05.csv")
}
if (!all(c("totalR", "date", "station") %in% names(test))) {
    stop("Missing columns in Ret05.csv")
}
# Valitaan asema 30 tarkasteltavaksi
train <- train[train$station == 30, ]
test <- test[test$station == 30, ]
# Muutetaan sarakkeet numeroiksi
train$totalD <- as.numeric(train$totalD)
test$totalR <- as.numeric(test$totalR)
# Vaihdetaan päiväyksen formaatti
train$date <- as.Date(train$date, format = "%d.%m.%Y")
test$date <- as.Date(test$date, format = "%d.%m.%Y")
# Luodaan aika sarakkeet
train$datetime <- as.POSIXct(paste(train$date, "00:00:00"))
test$datetime <- as.POSIXct(paste(test$date, "00:00:00"))
train$hour <- hour(train$datetime)
train$temp_f <- train$temp_avg * 1
# Yhdistetään lähdöt ja palautukset
summary_data <- data.frame(
  date = train$date,
  total_departures = train$totalD,
  total_returns = test$totalR[match(train$date, test$date)]
)

# Käsitellään nolla-arvot palautuksissa
summary_data$total_returns[is.na(summary_data$total_returns)] <- 0
# Muokataan data plottaukseen
summary_data_long <- summary_data %>%
  pivot_longer(cols = c(total_departures, total_returns),
               names_to = "Type",
               values_to = "Count")
```

KUVA 26. R-koodilla tehty malli vuokrauksista ja palautuksista asemalle 30

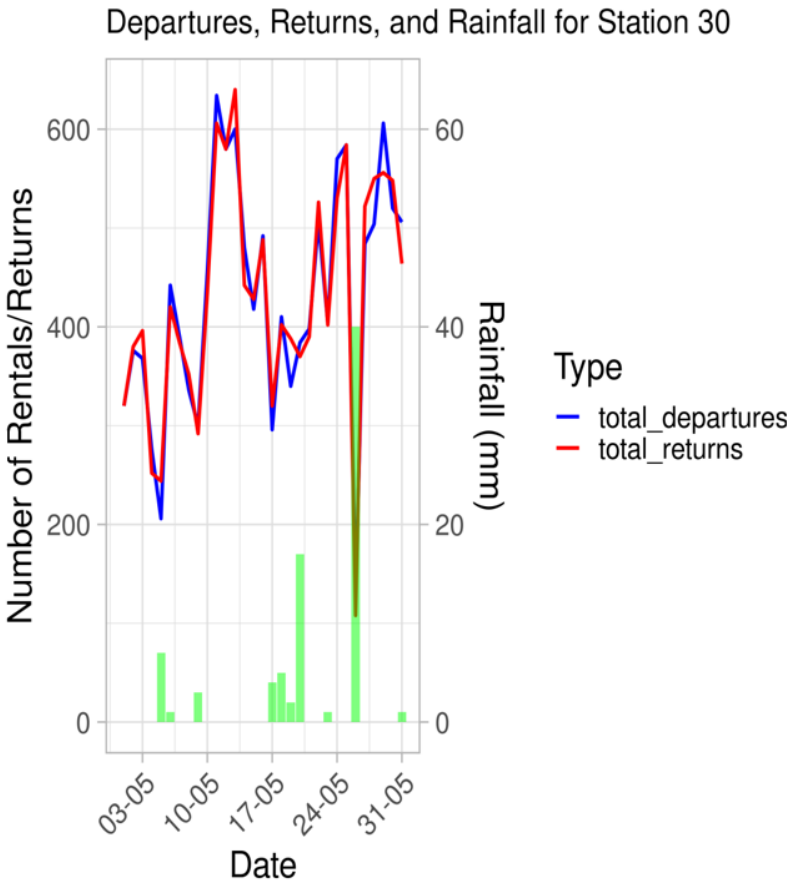
Kuvassa 27 näkyy koodi, joka ajetaan edellä esitettyyn malliin. Näin saadaan luotua tiedosto, joka sisältää graafin, jossa näkyy aseman 30 pyörien vuokraukset ja palautukset. Graafiin on lisätty

myös sademäärä millimetreissä. Muuttujia voi tässä vaiheessa lisätä suhteellisen helposti ja tuottaa erilaisia graafeja. Koodit syötetään RStudio ohjelmaan, joka suorittaa ne ja tallentaa lopputulokset.

```
library(ggplot2)
library(dplyr)

# Valmistellaan data sateeseen
rain_data <- train %>%
  filter(station == 30) %>%
  group_by(date) %>%
  summarise(total_rain = sum(rain_mm, na.rm = TRUE)) %>%
  mutate(total_rain = pmax(total_rain, 0))

# Yhdistetään sademäärä muuhun dataan
summary_data_combined <- summary_data_long %>%
  left_join(rain_data, by = "date")

# Luodaan graafi
p <- ggplot() +
  geom_line(data = summary_data_combined, aes(x = date, y = Count, color = Type), size = 1) +
  geom_bar(data = rain_data, aes(x = date, y = total_rain * 10), stat = "identity", fill = "green", alpha = 0.5) +
  theme_light(base_size = 20) +
  xlab("Date") +
  ylab("Number of Rentals/Returns") +
  scale_colour_manual(values = c("blue", "red")) +
  scale_x_date(date_breaks = "1 week", date_labels = "%d-%m") +
  ggtitle("Departures, Returns, and Rainfall for Station 30") +
  theme(plot.title = element_text(size = 18),
        axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_y_continuous(
    name = "Number of Rentals/Returns",
    sec.axis = sec_axis(~ . / 10, name = "Rainfall (mm)")
  )

ggsave("bike_departures_returns_and_rain_station_30.png", p)
```

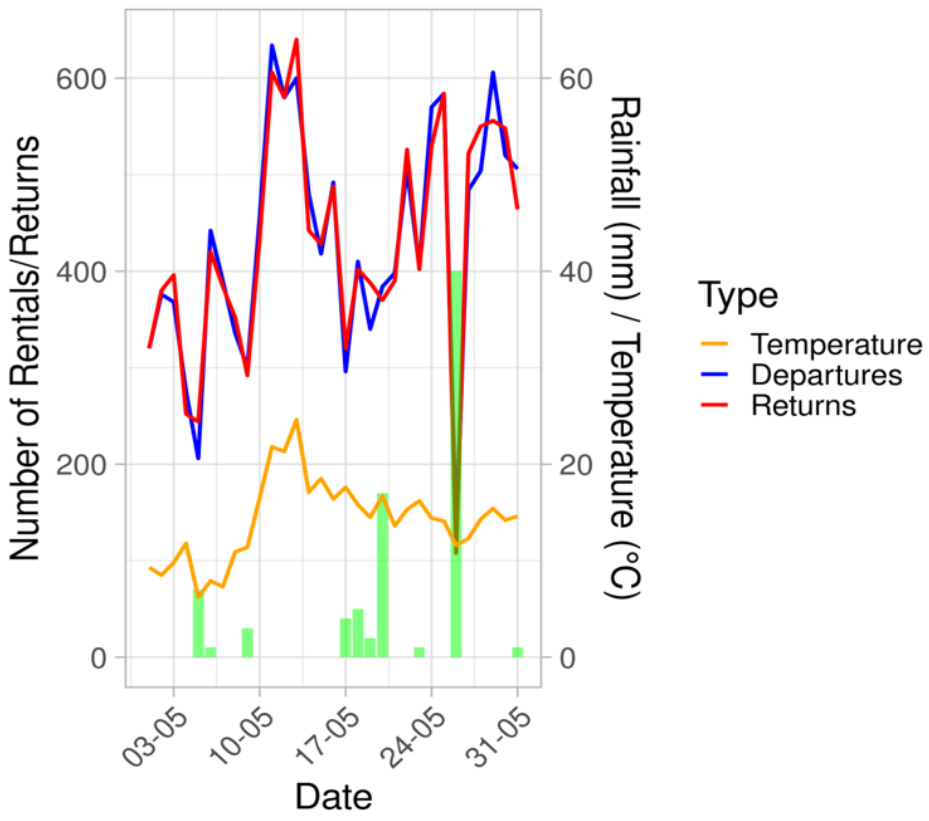
Kuva 27. Graafin tulostamiseen käytetty koodi

Kun plottauskoodi on ajettu RStudiossa niin saamme tiedoston, joka sisältää graafin. Edellisen koodin tuottama graafi on kuvassa 28. Kuvassa näemme asemalta 30 lähteneet vuokraukset kuvattuna sinisellä viivalla. Punaisella viivalla näkyy asemalle palautetut pyörät. Vihreät palkit kuvaavat päivän sademäärää. Graafissa on selvästi nähtävissä sateen vaikutus vuokrausten määrään. Yhtenä päivänä on satanut 40 millimetriä vettä ja pyörien vuokrausten, sekä palautusten, määrä asemalle on selvästi muita päiviä alhaisempi. Muidenkin sateisten päivien kohdalla on huomattavissa reilusti normaalia pienempää vuokrausten määrää.



KUVA 28. Aseman 30 vuokraukset sekä sademäärä

Kun kuvaan lisätään muuttujia, pystytään tutkimaan tuloksia vielä tarkemmin. Lisätään koodiin päivän korkein lämpötila ja katsotaan miten se vaikuttaa vuokrausten määrään. Kuvassa 29 on edellisten viivojen lisäksi mukana keltainen viiva, joka kuvaa lämpötilaa. Lämpötilan viiva seuraa aika hyvin mukana vuokrausten määrässä, näemme että silläkin on vaikutusta pyörien vuokrausten määrään. Kun kokeilemme muilla asemilla, niin saamme paljon samankaltaisia tuloksia. Joillakin asemilla heilunta on selvästi suurempaa kuin toisilla.



KUVA 29. Aseman 30 vuokraukset, sademäärä ja lämpötila

6 TULOKSET JA JOHTOPÄÄTÖKSET

Opinnäytetyön toteutuksessa perehdyttiin koneoppimisen perusteisiin koneoppimisen alkuajoista asti. Koneoppimisen teoriaa tutkittiin ja pyrittiin löytämään paras tapa suorittaa tutkimus kaupunkipyörien vuokrausten ennustamista varten. Päädyttiin käyttämään lineaarista regressiota ennustusten toteuttamiseen.

Ohjatun oppimisen avulla saatiin tutkimuksessa luotua erilaisia tuloksia, joiden perusteella pystyy luomaan malleja tulevien vuokrausten suhteen. Tulosten tarkkuuteen vaikuttaa monia eri muuttujia, joita ei aina datasta näe. Ennustusten tarkkuus on sitä parempi, mitä enemmän dataa on käytössä. Data-analytiikan avulla pystyy tekemään monia mielenkiintoisia hakuja. Näitä hakuja hyväksikäyttämällä voi toteuttaa Python-koodin avulla ennustavan mallin. Tätä mallia hyödyntämällä on mahdollista tehdä ennustuksia vuokrausten määrästä seuraavan päivän sääennusteiden mukaan. Ennustavan mallin toteutus oli varsin hankalaa, mutta lopullinen malli oli varsin hyvä tarkkojen ennustusten luomiseen.

Tutkimuksen ennustemalli pystyi ennustamaan vilkkaiden asemien vuokrausmäärät melko tarkasti, kuten Helsingin Rautatientorin itäisen aseman, jonka ennusteiden hajonta kolmen eri kuukauden datasta oli vain noin prosentin sisällä. Tämä osoitti, että malli toimii hyvin asemilla, joilla on paljon vuokrauksia ja palautuksia. Heikomman ennustetarkkuuden asemat, kuten Siltavoudintie, osoittivat kuitenkin vaihtelua, mikä vaikeutti tarkkojen ennusteiden luomista vähäisillä vuokrausmäärillä.

LÄHTEET

1. Bellis, Mary 2020. The History of the ENIAC Computer. Hakupäivä 25.11.2024. https://www.thoughtco.com/history-of-the-eniac-computer-1991601.
2. Glushenkov, Alex 2024. Deep Blue: The Chess Supercomputer That Changed AI and IBM Forever. Hakupäivä 25.11.2024. https://medium.com/@alexglushenkov/deep-blue-the-chess-supercomputer-that-changed-ai-and-ibm-forever-73cf98ce7b44.
3. Kämäräinen, Joni 2023. Koneoppimisen perusteet. Tallinna: Printon Trükikoda.
4. Kelleher, John D. & Tierney, Brendan 2018. Datatiede. Suom. Kimmo Pietiläinen. Helsinki: Libris / Painoliber Oy.
5. Kasparov, Garry & Greengard, Mig 2017 Syvä äly – Missä koneen älykkyys päättyy ja ihmisen luovuus alkaa. Suom. Mika Oksanen. Tallinna: Viisas Elämä.
6. Hänninen, Pasi 2022. Robotiikka ja tekoäly. Tammertekniikka / Amk-Kustannus Oy.
7. Ellonen, Noora & Kaakinen, Markus. Logistinen Regressio. Tampere: Yhteiskuntatieteellinen tietoarkisto. Hakupäivä 27.6.2024. https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvanti/regressio/logistinen/.
8. Elements of AI 2024. Tervetuloa opiskelemaan tekoälyn perusteita! MinnaLearn ja Helsingin Yliopisto. Hakupäivä 28.6.2024. https://course.elementsofai.com/fi/4/2.
9. Alpaydin, Ethem 2021. Koneoppiminen. Suom. Kimmo Pietiläinen. Helsinki: Terra Cognita.
10. Järvinen, Petteri 2023. Tekoäly ja minä – Ihmisenä tekoälyn aikakaudella. Helsinki: Tammi.
11. Kolari, Jukka & Kallio, Aleksi 2023. Tekoäly 123 – Matkaopas tulevaisuuteen. Jyväskylä: Docendo.
12. Priy, Surya 2024. Clustering in Machine Learning. Hakupäivä 15.8.2024. https://www.geeksforgeeks.org/clustering-in-machine-learning/.
13. Fraley, Chris & Raftery, Adrian. Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation 2012. Hakupäivä 26.7.2024. https://sites.stat.washington.edu/mclust/.
14. Russell, Stuart 2020. Human Compatible – AI and the Problem of Control. Great Britain: Penguin Books.

PYTHON-KOODI ENNUSTUSTEN TEKEMISEEN LIITE 1

```
import pandas as pd

from sqlalchemy import create_engine

from sklearn.linear_model import LinearRegression


# Funktio jolla haetaan data tietokannasta valitulle asemalle

def fetch_station_data(station_id):

# Luodaan yhteys MySQL tietokantaan

engine = create_engine('mysql+pymysql://xxx:xxx@localhost/hsldb')

# SQL käsky jolla haetaan data valitulle asemalle, mukaanlukien säätiedot

query = f"""

SELECT d.DepartureDate, d.DepartureCount,

COALESCE(r.ReturnCount, 0) AS ReturnCount, w.temp_high, w.rain_mm

FROM ( SELECT DATE(Departure) AS DepartureDate,

COUNT(*) AS DepartureCount

FROM hsldb.2021_05

WHERE DepartureStationId = {station_id}

GROUP BY DATE(Departure)

) AS d

LEFT JOIN (

SELECT DATE(`Return`) AS ReturnDate,

COUNT(*) AS ReturnCount
```

```
FROM hsldb.2021_05

WHERE ReturnStationId = {station_id}

GROUP BY DATE(`Return`)

) AS r ON d.DepartureDate = r.ReturnDate

LEFT JOIN

hsldb.2021_05w w ON d.DepartureDate = STR_TO_DATE(w.date, '%%d.%%m.%%Y')

ORDER BY d.DepartureDate;

"""

# Haetaan valitun aseman data

df = pd.read_sql(query, engine)

return df


# Funktio jolla treenataan lineaarisen regression malli vuokrauksille ja palautuksille

def train_models(df):

# Valmistellaan data vuokrauksille

X_departures = df[['ReturnCount', 'temp_high', 'rain_mm']]

y_departures = df['DepartureCount']

# Treenataan malli vuokrauksille

model_departures = LinearRegression()

model_departures.fit(X_departures, y_departures)

# Valmistellaan data palautuksille

X_returns = df[['DepartureCount', 'temp_high', 'rain_mm']]
```

```
y_returns = df['ReturnCount']

# Treenataan malli palautuksille

model_returns = LinearRegression()

model_returns.fit(X_returns, y_returns)

return model_departures, model_returns


# Funktio jolla ennustetaan huomisen vuokraukset sekä palautukset käyttäjän syötteen mukaan

def predict_departures_and_returns(model_departures, model_returns, return_count, depar-
ture_count, temp_high, rain_mm):

features_departures = pd.DataFrame([[return_count, temp_high, rain_mm]],

columns=['ReturnCount', 'temp_high', 'rain_mm'])

features_returns = pd.DataFrame([[departure_count, temp_high, rain_mm]],

columns=['DepartureCount', 'temp_high', 'rain_mm'])

# Tehdään ennustus

predicted_departures = model_departures.predict(features_departures)

predicted_returns = model_returns.predict(features_returns)

return predicted_departures[0], predicted_returns[0]


# Pääfunktio ennustuksen toteutukseen

if __name__ == "__main__":

try:

# Kysytään käyttäjältä halutun aseman ID

station_id = int(input("Syötä halutun aseman ID: "))
```

```
# Haetaan halutun aseman data
data = fetch_station_data(station_id)
# Varmistetaan että aseman ID:llä löytyy dataa
if data.empty:
print(f"Mitään dataa ei löynytyt asemalle ID: {station_id}")
else:
# Treenataan malli olemassa olevalla datalla
model_departures, model_returns = train_models(data)
# Kysytään käyttäjältä huomisen säätiedot
temp_high = float(input("Syötä huomisen lämpötilan ennuste(°C): "))
rain_mm = float(input("Syötä huomisen sademäärän ennuste (mm): "))
last_day = data.iloc[-1]
# Ennustetaan vuokraukset ja palautukset
predicted_departures, predicted_returns = predict_departures_and_returns(
model_departures, model_returns,
last_day['ReturnCount'], last_day['DepartureCount'],
temp_high, rain_mm
)
# Lasketaan nettovuokraukset
net_departures = predicted_departures - predicted_returns
# Tulostetaan ennustetut vuokraukset, palautukset ja nettovuokraukset valitulle asemalle
print(f"Ennustetut vuokraukset asemalta {station_id} huomiselle: {predicted_departures:.2f}")
```

```
print(f"Ennustetut palautukset asemalle {station_id} huomiselle: {predicted_returns:.2f}")

print(f"Ennustetut nettovuokraukset asemalta {station_id} huomiselle: {net_departures:.2f}")

except ValueError as e:

print(f"Invalid input: {e}")

except Exception as e:

print(f"An error occurred: {e}")
```