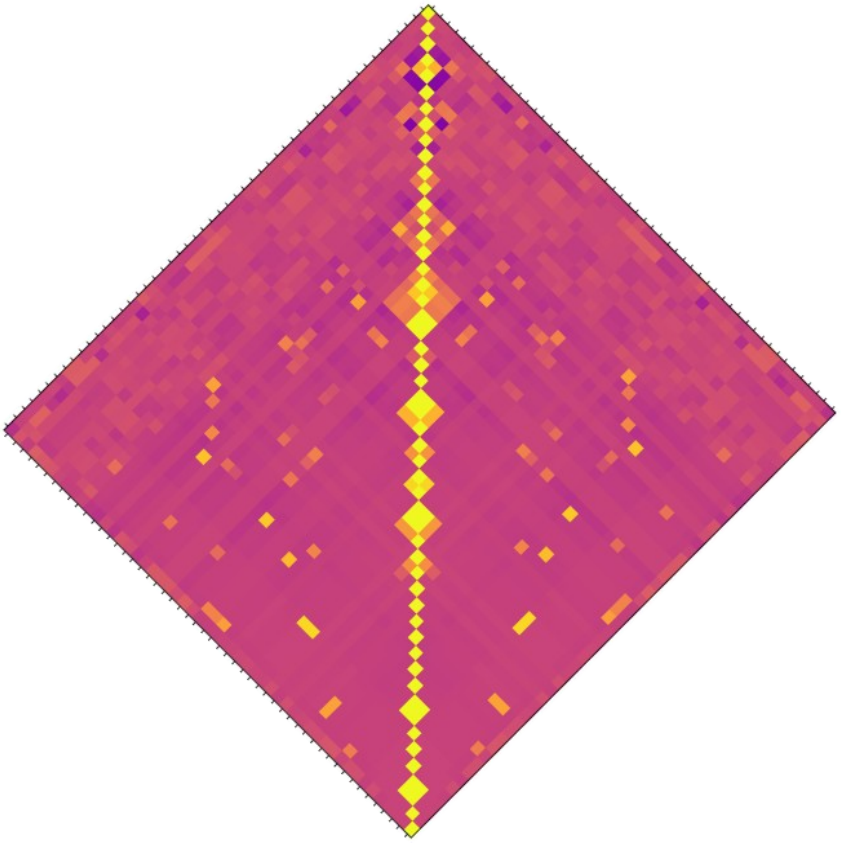
Raportti: Wasteless-hankkeen esiselvitysvaiheen tuloksia

Aistico Oy, Jussi Rasku, 10.4.2020



Sisällysluettelo

Johdanto

Wasteless-hankkeen esiselvitysvaiheen tarkoituksena oli tehdä data-aineiston laadullisen ja määrällisen laadun analysointia, koneoppimiseen perustuvan ennustemallin prototyypin rakentamista ja opettamista, sekä antaa suosituksia tulevaan aineiston keräämiseen ja varsinaisen toteutusvaiheen valmisteluun.

Toteutustekniikaksi valittiin IPython-työkirjat, joita kirjoitettiin useita – yksi kunkin datalähteen esiprosessointiin, yksi suoritteiden ennustamiseen ja yksi hävikkien ennustamiseen. Lisäksi toteutettiin työkirja, joka mahdollistaa tuotetuilla koneoppimismalleilla ennustamisen. Python on data-analytiikassa viime aikoina voimakkaasti yleistynyt ohjelmointikieli ja sen valinta mahdollisti nopean iteroinnin esiselvitysvaiheen aikana. Kieli myös mahdollistaa tuotetun ohjelmakoodin käytön hankkeen seuraavissa vaiheissa. Kaikki raportin laatimiseksi käytetyt ohjelmistot olivat avointa lähdekoodia eikä niiden käyttämiseen tarvita erillisiä lisenssejä tai hankalasti saatavissa olevia työkaluja.

Tärkeimmät valitun lähestymistavan edut ovat toistettavuus ja joustavuus. Mikäli halutaan myöhemmin lisätä täydentäviä tietoaineistoja (esim. uusien kuukausien tietoja), voidaan se tehdä päivittämällä aineistojen lähdetiedostoja tai -kansioita ja suorittamalla Python-työkirjojen ohjelmakoodit uudestaan. Mikäli esikäsittely olisi tehty käsin, tämä ei olisi mahdollista - tai se olisi erittäin työlästä. Esiprosessointiohjelmistot voi siis ajatella hyvin perinpohjaisiksi dokumenteiksi siitä, mitä käsin tehtäviä operaatioita olisi tarvittu aineiston saattamiseksi oikeaan muotoon. Lisäksi työkirjoja on mahdollista muuttaa, mikäli tulevaisuudessa ilmenee tarpeita esiprosessointikoodin laajentamiseen.

Koneoppimislähestymistavan sopivuuden todentamiseksi esiprosessoitua ja siivottua aineistoa yritettiin mallintaa lukuisilla erilaisilla koneoppivilla algoritmeilla. Keskeinen tulos tästä kokeellisesta työstä on, että nykyinen käytettävissä oleva aineisto ei salli luotettavan ennustemallin rakentamista suoritemäärien ennustamiseen. Käytettävissä oli esikäsittelyn, avoimilla datalähteillä täydentämisen ja aineiston puhdistuksen jälkeen 377 päivän aineisto, jossa oli mukana suoritemäärät, eväät, säätiedot, alueelliset hengitystieinfektiotiedot, ja päivittäiset ruokalajitiedot. Edes kokeilluista menetelmistä kehittyneemmät, kuten satunnaismetsät tai neuroverkot, eivät pystyneet oppimaan eri tekijöiden välisiä yhteyksiä käyttökelpoisella tarkkuudella. Näyttääkin siltä, että ennustemallin syötteeksi tarvittaisiin sellaisia uusia tietoja, jotka ovat koulun ja ruokalan järjestelmissä ja jotka eivät vielä olleet käytettävissä: ilmoitetut poissaolot, koulun tapahtumat, ruokaloiden teemapäivät, lomakaudet jne. Lisäksi suoritekirjausten analyysi paljasti merkkejä siitä, että suoriteaineistossa saattaa olla laadullisia puutteita ja suositus onkin, että aineistoa kerätään lisää ja jo saatavilla olevasta aineistosta tunnistetaan poikkeamat ja tarkastetaan kirjaukset tehneiltä asiantuntijoilta, onko kyseessä anomalia vai todellinen poikkeustilanne. Näin voidaan parantaa sekä aineiston laatua, että määrää, jotta ennustustarkkuutta voidaan toteutusvaiheessa nostaa lähemmäs käyttökelpoista tasoa.

Hävikkiaineiston laatu oli suoriteaineistoa parempaa. Aineisto on toistaiseksi liian pieni, että kokonaishävikkiä oltaisiin voitu ennustaa luotettavasti, mutta erityisesti lautashävikin ennustaminen vaikuttaa olevan hyvä kohde koneoppimista hyödyntävälle lähestymistavalle.

Aineiston esiprosessointi

Esiprosessointivaihe sisältää aineiston muuttamisen koneoppimisalgoritmien kannalta oikeaan muotoon. Samalla voidaan tehdä niiden laadun, yhdenmukaisuuden ja oikeellisuuden tarkastelua. Tämä vaihe on usein työläs, mutta samalla välttämätön, jotta tulokset olisivat luotettavia. Tyypillisenä haasteena on, että aineiston keräyksen suorittavat eri henkilöt kuin he, jotka ovat vastuussa aineiston analysoinnista ja mallien rakentamisesta. Aineiston laatuun ei usein tämän vuoksi kiinnitetä riittävästi huomiota, sillä puutteelliseen laatuun liittyvät murheet jäävät jonkun muun kuin aineiston kerääjän vastuulla. Siksi onkin tärkeää, että puutteet tunnistetaan ja niistä keskustellaan aineistoa keräävien tahojen kanssa. Tavallisimpia puutteita aineistossa ovat:

- Kokonaan virheellisiä kirjauksia ja numeroita (ts. kirjoitus-/merkintävirheitä) tai niiden yhdistelmiä.
- Kohinaa tai harhaa mittaustuloksissa (esim. mittausinstrumenttien arvioissa).
- Epätarkkoja tai unohtuneita kirjauksia (esim. vakioarvo tai muistinvarainen arvio usealle jälkikäteen täytetylle mittaustapahtumalle).
- Kokonaan puuttuvia sarakkeita, jotka olisivat olleet tarpeellisia aineiston hyödynnettävyyden kannalta.

Esiprosessoinnilla pyritään tunnistamaan edellä listattuja ongelmia ja muokkaamaan aineistot muotoon, jossa ne ovat valmiimmin hyödynnettävissä mallin opettamisvaiheessa. Esiprosessointivaiheen tärkeyttä ja työläyttä ei kannata aliarvioida. Tätäkin raporttia laatiessa kenties 80 % kokonaistyömäärästä käytettiin aineiston laadun varmistamiseen ja sen muuttamiseen oikeaan muotoon. Tämä on tunnettu totuus data-analytiikka työssä, mihin tässä hankkeessa toi vielä oman mausteensa data-aineistojen kirjauskäytänteiden kirjavuus ja aineistojen monet eri lähteet. Ilman työlästä esiprosessointia on kuitenkin hyvin vaikea lausua mitään aineiston laadusta tai sen soveltuvuudesta mallien rakentamiseen.

Alla on kuvattu kunkin aineistolajin esiprosessointivaiheet perusteluineen. Tarkemmat tiedot ja yksityiskodat on luettavissa raportin mukana toimitetuista IPython-työkirjoista. Aineiston käsittelyssä hyödynnettiin paljon Pandas-taulukkoja ja Matplotlib-visualisointikirjastoa. Osassa tehtävistä jouduttiin myös käyttämään Scikit-learn -kirjaston ennustemalleja, jotta aineistoa pystyttiin täydentämään. Aineiston osien täydentäminen aina kun se oli mahdollista, oli välttämätöntä, sillä aineiston määrä oli kohtalaisen pieni.

Suoritekirjausten esiprosessointi

Suoritekirjaukset pitivät sisällään keittiön tarjoamat lounasannokset, eritysruokavalioannokset, kahvitukset, eväät jne. Ne oli kirjattu kunkin kuukauden merkinnät ja summat sisältyviin `xlsx` -tiedostoihin. Rivikohtaiset kirjaukset oli tehty kullekin päivälle. Osa kirjauksista oli merkitty PDF-tiedostoihin, mistä ne tuotiin käsin XLSX-muotoon. Kuitenkin, koska PDF-tiedostoissa rivitietoja (eli päiväystietoja) ei ollut toistettu joka sivulle, oli kirjausten rivi joissain tapauksissa vaikea päätellä. Tällä saattaa olla vaikutusta joidenkin vuoden 2020 XLSX-tiedostojen kahvitusmerkintöjen luotettavuuteen.

Analyysin aikana tehtiin seuraavia havaintoja seuranta-aineiston laadusta:

1. Sarakkeiden nimet, järjestys ja käyttö ovat jonkin verran muuttuneet seurantajakson aikana. Esimerkiksi vuoden 2016 ensimmäisissä tiedostoissa kouluruokailut on merkitty `Lounas`-sarakkeeseen kun myöhemmin on siirrytty merkitsemään nämä tiedot `Lounas koulut`-sarakkeeseen.

2. Viikolle ~49/2017 asti merkinnät vaikuttavat olevan epätarkkoja. Koko viikko on saatettu määrätä samaa suoritemäärää. 24.11.2017 kirjauskäytännössä on tapahtunut selkeä muutos, mutta vielä tästä päivästäkin edespäin suoritteiden kirjaustarkkuus vaihtelee.

3. 24.11.2017–2018-08-31 suoritusmerkintöjen kymmentasaluvut ovat huomattavan paljon todennäköisempiä, mitä normaalijakauma antaisi olettaa. Tämä viittaa siihen, että suoritteet oli edelleen jossain määrin arvioitu.

4. Sarakkeeseen “Erityisruokavalio koulut” on kirjattu vakiomäärä ruokailijoita (erityisruokavalioiden kokonaismäärä) aina 28.01.2020 asti.

5. XLSX-tiedostoista löytyi muutamia selkeästi virheellisiä tietoja, ja niihin on tehty käsin seuraavat korvaukset: `to 15.9.2016 1640,00->451`.

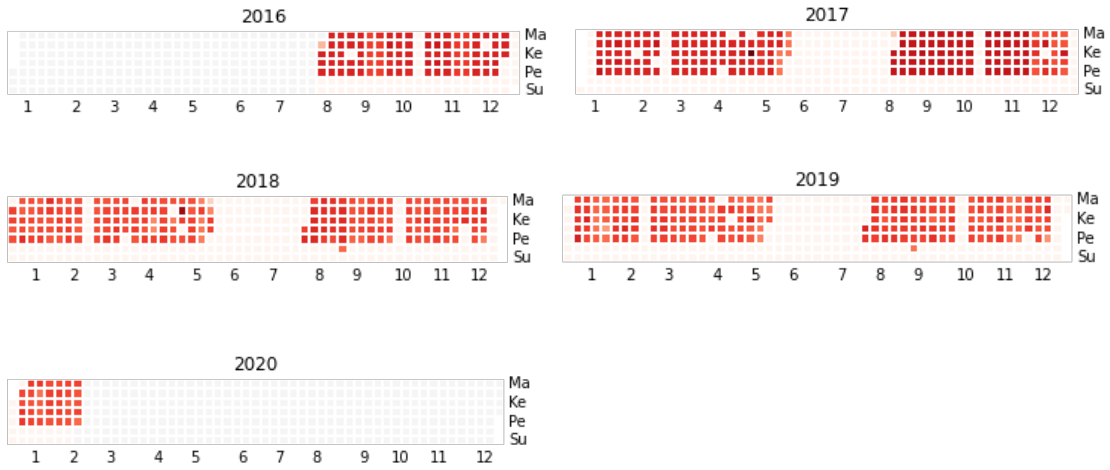
Havaintojen aiheuttamiin potentiaalisiin ongelmiin varauduttiin esiprosessointiohjelmassa `YhdistaSuoritteet.ipynb` yhdenmukaistamalla sarakkeiden käyttöä ja lukemalla vain lounassuoritteiden ennustamisen kannalta olennaiset sarakkeet. Mikäli jokin sarake puuttuu, ohjelma antaa tästä varoituksen. Käytettävissä olevalle aineistolle nämä varoitukset liittyivät lähinnä esim. puuttuvaan “Jäätelö”-sarakkeeseen.

Suoritteet luetaan `src/suoritteet`-kansiossa olevista `.xlsx`-tiedostoista. Sieltä poimittiin vain edellä alla listatut sarakkeet ja jätetään esim. yhteenvetorivi huomiotta. Lisäksi poistettiin nollasuoritteiset rivit (eli viikonloput ja lomat). Kaikkia suoritesarakkeita ei pidetty, vaan vain tehtävän kannalta olennaisimmat (katso Taulukko 1, jossa Vierasateria/ilmainen on yhdistetty sarakkeen Vierasateria kanssa). Näin sisään luetulla seurantajaksolla syksy 2016 – alkukevät 2020 valituissa sarakkeissa oli suoritteita yhteensä Taulukkoon 1 kirjatut määrät.

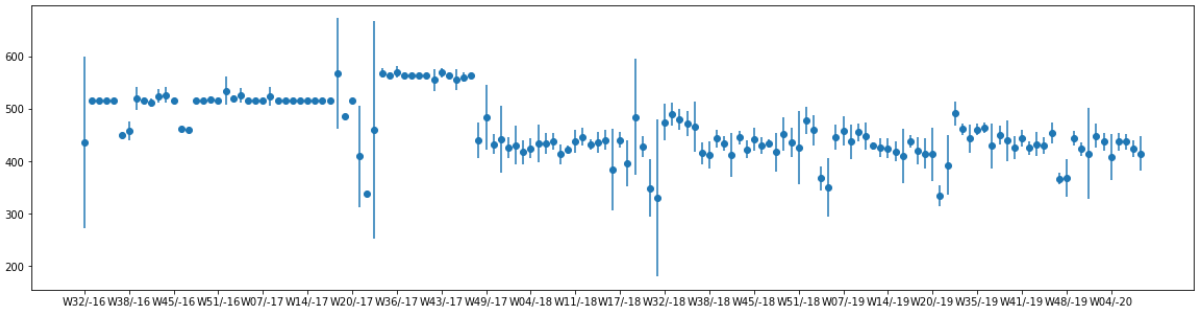
Taulukko 1: Luetut suoritteet yhteensä.

Suoritetyyppi	Yhteensä
Lounas koulut	315914
Erityisruokavalio koulut	29065
Eväät	1720
Vierasateria	122
Makkarat	371
Jäätelö	287

Voimme visualisoida päivittäiset lounassuoritteiden määrät kalenterissa (Kuva 1). Mitä tummempi neliö, sitä suurempi suoritteiden määrä. Kuvasta näemme hyvin myös juhlapyhät ja lomaviikot. Samoin voimme huomata muutaman lauantaipäivän, jolloin on ollut koulua ja koululounas on ollut tarjolla. Analyysiä tehtäessä ei ollut tietoa, minkä päivän ruokalistaa lauantaisin noudatettiin, joten kyseiset lauantaipäivät suodatettiin pois suoriteaineistosta.



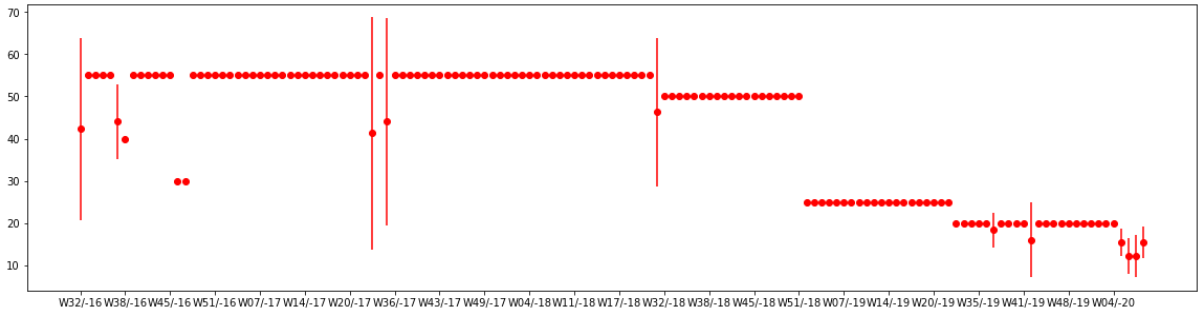
Kuva 1: Kalenterivisualisaatiot lounaiden suoritemerkinnöistä.



Kuva 2: Viikoittainen lounassuoritteiden keskiarvo ja keskihajonta.

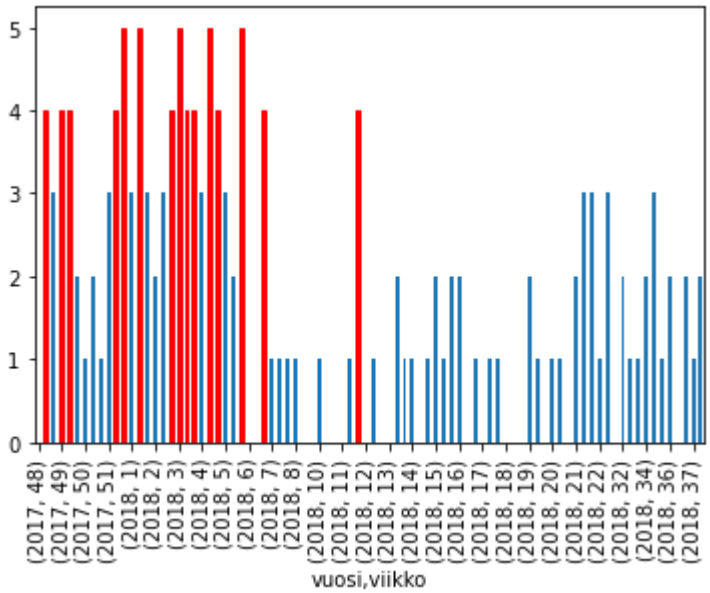
Yllä oleva kuvaaja havainnollistaa, miten ruokailijasuoritemäärä on kehittynyt seurantajakson aikana. Kuvaajasta näkee suoraan, että syksyllä 2017 kirjauskäytäntö on radikaalisti muuttunut ja aiempi kirjauskäytäntö on ollut huomattavan suuripiirteinen. Tämä valitettavasti tarkoittaa, että suoritemerkinnät 24.11.2017 asti ovat ennustemallin osalta käyttökelvottomia, sillä koko viikko on saatettu määrätä samaa suoritemäärää. Aineistosta voi havaita, että 24.11.2017 kirjauskäytännössä on tapahtunut selkeä muutos ja tästä päivästä edespäin päivittäin kirjattujen suoritteiden määrä vaihtelee, ja täten sen jälkeiset suoritemäärät vaikuttavat uskottavammilta.

Myös erittäin poikkeukselliset suoritemääräkirjaukset näkyvät Kuvassa 2 suurena viikon keskihajontana (korkeat pylväät). Poikkeamien poistaminen aineistosta on yleensä viisasta. Päiväkohtaisten suoritekirjausten osalta tämä tehtiin poistamalla 2,5 % pienintä ja 5 % suurinta suoritemäärää. Syyt poikkeaviin päiviin varmasti vaihtelevat, ja ne voidaan jäljittää vain erikoistapauksissa. Esimerkiksi elokuun ensimmäinen merkintä oli lähes joka vuonna huomattavasti pienemmälle ruokailijamäärälle ja ilman erityisruokavalioita ja tällöin luultavasti vain henkilökunta on ruokaillut koululla. Nämä, ja vastaavat poikkeavat havainnot ovat kuitenkin niin suuria ja harvinaisia, että niillä ei todennäköisesti ole myöhemmin opetettavan mallin kannalta tilastollista arvoa. Ne kannattaa siis suodattaa pois aineistosta.



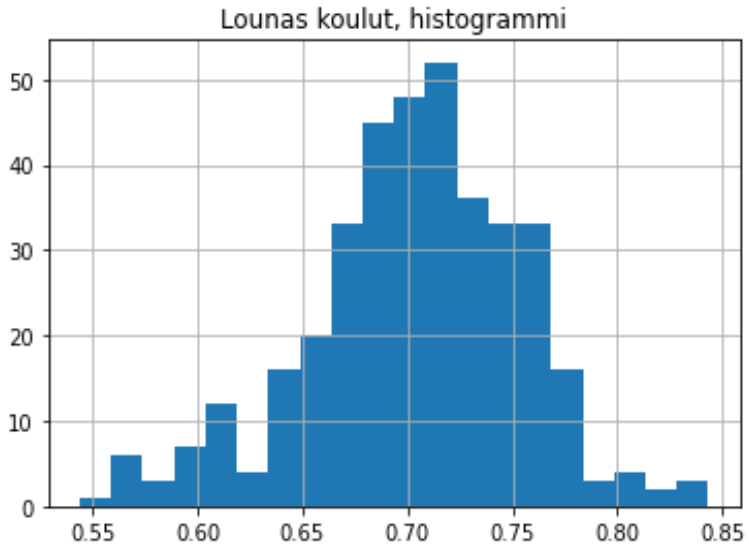
Kuva 3: Viikoittainen erityisruokavaliosuoritteiden keskiarvo ja keskihajonta.

Myös erityisruokavaliosuoritteiden tarkastelu on tarpeen. Sarakkeeseen *"Erityisruokavalio koulut"* on kirjattu muutamia poikkeuspäiviä lukuun ottamatta vakiomäärä ruokailijoita (erityisruokavalioiden kokonaismäärä) aina 28.01.2020 asti. Itse asiassa, kuten Kuvasta 5 näemme, erityisruokavaliolounaiden osalta kirjauskäytäntö vastaa todellisuutta vasta keväällä 2020. Sitä ennen kirjattu on lähinnä merkittävät poikkeustilanteet. Aineiston laadun vuoksi erityisruokavaliota noudattaville asiakkaille ei kannata yrittää laatia ennustemallia.



Kuva 4: Viikon suoritekirjausten määrä (kpl), jotka ovat kymmenellä tai viidellä jaollisia.

Kun aineistosta suodatetaan pois poikkeamat, lauantait ja pahimmat epäluotettavat kirjaukset, lopulliseen suoriteaineistoon jää 377 päivää. Valitettavasti aineistoon jää kuitenkin edelleen muutamia epäilyttäviä kirjauksia. Esimerkiksi loppuvuoden 2017 ja alkuvuoden 2018 suoritekirjaukset on todennäköisesti arvioitu jälkikäteen, sillä ne ovat epäilyttävän usein jaollisia viidellä tai kymmenellä (katso Kuva 4) ja suoritusmerkintöjen kymmentasaluvut ovat huomattavan paljon todennäköisempiä, mitä normaalijakauma antaisi olettaa. Tämä viittaa siihen, että suoritteet on edelleen jossain määrin arvioitu ajanjaksolla 24.11.2017 - 2018-08-31. Kuten kuvasta näemme, alkuvuonna 2018 suoritemäärät on yleisesti arvioitu viiden suoritteen tarkkuudella. Nämä aineistorivit on tästä huolimatta otettu mukaan seuraavaan vaiheeseen, eli mallin rakentamisvaiheeseen, sillä datan määrä oli jo valmiiksi alhainen käytettäväksi koneoppimisperusteisen mallin rakentamiseen. Alla Kuvassa 5 on annettu normalisoitujen suoritemäärien jakauma, joka näyttää jakautuvan aika hyvin normaalijakauman mukaisesti.



Kuva 5: Suoritearvojen jakauma

Viimeisenä esiprosessointiaskeleena suoritemääristä on tehty suhteellisia. Tämä tarkoittaa, että ne on normalisoitu jakamalla koululounaiden suoritemäärä oletusruokavaliota noudattavien ruokailijoiden määrällä kyseisenä lukuvuonna. Tässä on käytetty lisätaulukkoa tiedostosta `"Ruokailijamäärät Lapuan yläkoulu 2016-2020.xlsx"`. Valitettavasti tässä taulukossa ei ollut annettu erityisruokavaliota noudattavien ruokailijoiden määrää. Erityisruokavaliosuoritteiden osalta käytettyjen vakiokirjausten (Kuva 3) ansiosta, nämä puute voitiin korjata ottamalla lukuvuosikohtainen mediaani suoriteaineiston "*Erityisruokavalio koulut*" -sarakkeesta.

Huomasimme, että retket on jo kirjattu suoriteaineistoon "Eväät"-sarakkeeseen, eikä niitä täten tarvinnut käsitellä tai syöttää erikseen. Tämä sarake tosin otettiin mallin rakentamisvaiheessa omaksi piirteekseen, jotta se voidaan huomioida syötteenä suoritemääriä ennustettaessa. Muita ruokailija- ja suoritemäärään vaikuttavia poissaoloja ei mitä ilmeisimmin ole kirjattu, eikä niitä täten ollut käytettävissä esiselvitysvaiheessa.

Tekstimuotoisen ruokalistojen esiprosessointi

Ruokalistojen esiprosessoinnin tarkoituksena oli tehdä luetulle datalle joitain tarkastuksia ja korjauksia sekä yhdenmukaistaa eri lukuvuosien ruokalistat. Tekstimuotoisen syötteen antaminen suoraan koneoppiville algoritmeille ei useinkaan ollut mahdollista. Siksi tämän vaiheen tavoitteena oli muodostaa ruokalistojen perusteella päiväkohtaiset piirrevektorit, jotka kuvailevat kyseisenä päivänä tarjolla ollutta ruokaa.

Ruokalistat toimitettiin Word- ja PDF-dokumentteina, josta ne käsin siirrettiin XLSX-tiedostoihin. Jokaisen tiedoston muoto oli sama:

- Ensimmäisellä rivillä on otsikko *"Viikko"* ja käytössä olevan viikkoruokalistan numero.
- Seuraavat rivit ovat viikonpäivät siten, että ensimmäisessä sarakkeessa lukee viikonpäivä (esim. *"Maanantai"*), jota seuraa kunkin toistuvan viikon ruokalajit.
- Viimeisellä rivillä on rivitieto *"Viikot"* ja seuraavissa sarakkeissa mitä kyseisten lukuvuoden viikkonumeroita kyseisen sarakkeen ruokalajit koskevat.
- Lukuvuosi käy ilmi XLSX-tiedoston nimestä.

Nämä luettiin ja samalla ruokalajikuvauksia yhtenäistettiin siten, että kauttaviivan (“/”) ja muiden välimerkkien käyttö oli yhdenmukaista. Samalla suluissa olevat sivuhuomautukset poistettiin. Käyttämällä vuotta, viikkonumeroa ja viikonpäivänumerointia, ruokalistat muutettiin päiväkohtaiseen muotoon, jossa jokaisella rivillä oli päiväys ja kyseisenä päivänä tarjottu ruoka tekstimuodossa. Aineiston esikäsittelyssä on helppo tehdä virheitä, etenkin indeksien ja numerointien kanssa. Aineiston tähänastinen esikäsittely validoitiin tarkistamalla joka lukuvuoden ensimmäisen ja viimeisen päivän ruokalajin oikeellisuus, sekä muutamia satunnaisia hajapäiviä keskeltä.

Kuva 6: Sanapilvi piirteytetyistä ruokalistoista.

Kuten aiemmin totesimme, tekstimuotoiset kuvaukset eivät sellaisenaan ole käyttökelpoisia koneoppimiselle. Siksi tekstikuvaukset piti vielä piirteyttää. Tässä käytettiin ns. *bag-of-words* menetelmää, jossa kukin ruokalistan kuvauksessa käytetty sana koodataan omaksi binääripiirteekseen (joka siis on 0, jos sanaa ei ole kyseisen päivän ruuan kuvauksessa, ja 1 jos se on). Sanojen perusmuotoon muuttamiseen käytettiin `libvoikko`-modulia ja yhdyssanojen pilkkomiseen `finnsyll`-modulia. Valitettavasti näiden toiminta osoittautui osin epäluotettavaksi (erityisesti yhdyssanojen osalta), joten esikäsittelyä varten jouduttiin laatimaan suuri joukko aineistokohtaisia sanankäsittely- ja katkomissääntöjä. Säännöstö on kuitenkin suhteellisen yleiskäyttöinen ja sitä voidaan täydentää, mikäli uutta aineistoa tulee saataville. Siksi käsin laaditut säännöt ja ruokalistojen piirteytys siirrettiin omaan tiedostoonsa (`src/menu_stemmer.py`). Hajotetut yhdyssanat ja muut sanat palautettiin perusmuotoonsa `libvoikko`-stemmauksen avulla, jonka jälkeen Scikit-learn `CountVectorizer` tuotti piirrevektorin. Koska tekstimuodosta johdettujen piirteiden määrä kasvaa nopeasti, aineistosta tuotettiin kolme erikokoista piirrevektoria (41 / 63 / 92 sanaa eli piirrettä).

Piirredataa voidaan havainnollistaa sanapilven avulla. Keräämällä summat kullekin suurimmassa piirrevektorissa olevalle sanalle kaikkien aineistossa olleiden päivien yli voimme tuottaa Kuvan 6 mukaisen sanapilven, jossa sanan koko kuvastaa sen esiintyneisyyttä päivittäisissä ruokalistoissa. Näin yhdellä silmäyksellä saa käsityksen siitä, mitä Lapuan yläkoululla oli tarjolla lukuvuosina 2016-2020. Sanapilvistä on raportin mukana toimitetussa `docs`-kansiossa painokelpoiset suuriresoluutioiset kuvat.

Hengitystieinfektiodata ja sen esiprosessointi

THL kerää ja tarjoaa avoimia tietoaineistoja, joista voidaan päätellä maakunnassa, tai joskus jopa kunnassa, riehuvan kausi-influenssan ja muiden hengitystieinfektioepidemioiden laajuus. Tilanne on olennaista myös ruokaloiden kannalta, sillä se heijastuu kouluihin poissaoloina ja sitä kautta vaikuttaa ruokailijoiden määrään.

Käyttökelpoisimmassa muodossa olivat Hoitoilmoitusrekisteri AvoHILMOn tiedot. Ne on mm. ladattavissa THL:n tiiviste- ja kuutiokäyttöliittymän (TIKU) kautta. Palvelussa on myös rajapinta koneluettavalle datalle, mutta siihen ei esiselvitysvaiheessa tutustuttu aikataulusyistä. Tiedot näyttävät päivittyvän palveluun reaaliaikaisesti (ts. edellisen viikon hengitystieinfektion vuoksi tehtyjen käyntien määrä on aina tiedossa). Lisäksi tiedossa kuluvan viikon kertymä voidaan saada näkyviin, mutta valitettavasti tätä päiväkohtaista kertymää ei voi jälkikäteen palauttaa - ainakaan TIKU käytön avulla. Tämän vuoksi myös esiselvitysvaiheessa käytettiin vain viikkokohtaisia summia, vaikka pidempään tehtävässä aineisonkeruussa myös ajantasaisemmat viikkokertymät voisivat olla tarjolla.

Tässä yhteydessä tarkastellaan THL:n seurantatietoja hengitystieinfektioiden (ARI) ja influenssan kaltaisen taudin (ILI) aiheuttamista käynneistä. Tiedot on haettu terveyspalveluntuottajalle "Lapuan terveyskeskus (90308)" ja tiedot on käsin siirretty XLSX-tiedostossa uudelle välilehdelle rivitiedoiksi vaikeammin hyödynnettävien saraketietojen sijaan. Aineisto on lisensoitu hyvin sallivalla CC BY 4.0 -lisenssillä, mikä mahdollistaa sen käytön osana toteutettavaa palvelua.

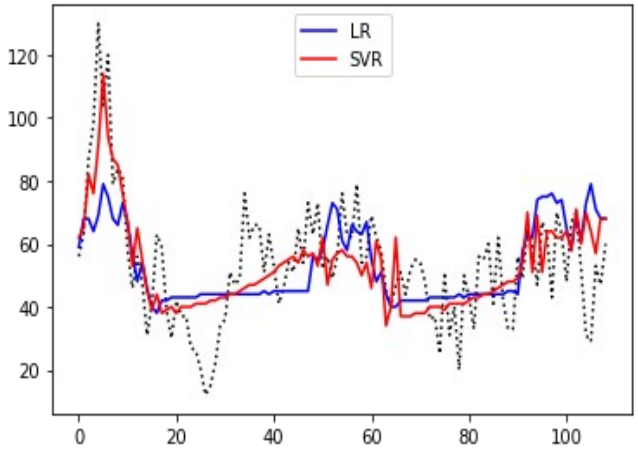
Hoitoilmoitusrekisterin TIKU-käytön suurin rajoite oli, että käyntitiedot olivat saatavilla vain vuoden 2018 alkuun. Koska ruokalan seuranta-aineisto ulottui syksyyn 2016, tuli käyntitietoaineisto täydentää jotenkin. Onneksi THL on laatinut vuosittaisia “Influenssakausi Suomessa” -raportteja (Ikonen et al 2015-2019). Niissä on kuvan muodossa raportoitu influenssaepidemian viikoittainen eteneminen sairaanhoitopiireittäin. Kullekin maakunnalle ja viikolle on annettu luokitus: *'hyvin korkea', ‘kohtalainen’, ‘normaali’, 'matala'*.

Esiselvitysvaihetta varten tilaaja toimitti raporttien maakunta- ja viikkokohtaiset seurantatiedot XLSX-tiedostoon kerättynä. Luokitustietoa oli saatavilla verkkoaineistossa myös viime kuukausilta (THL 2020) joten sitä käytettiin täydentämään olemassa olevia luokitustietoja. Ensin luokitustieto kuitenkin uudelleenjäsenneltiin sarakkeisiin. Influenssa-aktiivisuusluokitus koodattiin numeroarvoiksi 0-3, jossa 0 on matala ja 3 hyvin korkea. Koska kesän ajalta (viikot 23-39) luokitustietoa ei ollut influenssa-aktiivisuus- raportoinnissa mukana, merkittiin kunkin kesän tilanteeksi *‘matala*’.* Kouluruokalan toiminnan kannalta tällä päätöksellä ei pitäisi olla suurta merkitystä, mutta se antoi meille yhtenäisen tietoaineiston. Koodauksen jälkeen THL:n influenssaraporttien ja AvoHILMOn aineiston päällekkäiset tiedot yhdistettiin ja laadittiin regressioennustemalli Lapuan terveyskeskuksen infektiokäyntien määrälle. Opetusaineiston koko oli 109 viikkoa/riviä.

Pitääksemme käyntimäärät yhteismitallisina eri paikkakuntien välillä, se normalisoitiin. Tässä käytettiin jakajana Lapuan tämänhetkistä asukasmäärää (14280 asukasta). Lisäksi influenssaepidemian viikoittaisen etenemisen sairaanhoitopiireittäin jaetun aineiston tiedoista hyödynnettiin kullekin datapisteelle kahden edeltävän, kyseisen viikon ja kahden seuraavan viikon influenssa-aktiivisuusluokitusta. Ennen syöttämistä koneoppimisalgoritmeille kaikki arvot, mukaan lukien terveyskeskuksessa hengitystieinfektioiden vuoksi käyneiden vierailujen suhteellinen osuus Lapuan asukkaista, vielä skaalattiin välille [0.0, 1.0].

Regressiomenetelmien testaus tehtiin jättämällä vuorotellen yksi opetusaineiston arvo pois (ns. leave-one-out cross-validation), jolla saadaan hyvä arvio menetelmän suorituskyvylle. Aineistoon sovitettiin kaksi malliperhettä: lineaariregressio (LR), ja tukivektorikoneregressio (SVR). SVR on vahvasti parametrisoitu, joten sille tehtiin hilapohjainen parametrien haku käyttäen Scikit-learn moduulin `GridSearchCV` -työkalua. Optimoiduilla parametreilla se tuotti keskiarvoltaan hieman tarkemmat ennusteet kuin LR. Tätä on havainnollistettu Kuvassa 7, jossa katkoviiva on todellinen hengitystieinfektion vuoksi terveyskeskuksessa kyseisellä viikolla asioineiden määrä ja sininen ja punainen viiva ovat mallien ennustamat käyntimäärät. Huomaa, että mallit saavat syötteekseen ainoastaan viikkonumeron ja THL:n raporttien mukaiset Etelä-Pohjanmaan sairaanhoitopiirin viikkoluokituksen kausi-influenssan raportoiduille etenemiselle viiden viikon ajalta. Näiden perusteella ennustemalli antaa ulostulona montako asiakasta Lapuan terveyskeskuksessa ennustettiin olevan kyseisellä viikolla.

Tukivektoriregressioon (SVR) perustuva malli näyttää (odotetusti) mallintavan lineaarimallia tarkemmin poikkeustilanteet, mutta niiden tarkkuus on lukujen valossa ja silmämääräisesti tarkastellen samantasoinen. Huomaa, miten kesän ajalta puuttuva tieto influenssakausi-luokituksista (viikot n. 22-40, Kuva 7) selkeästi vähentää mallin herkkyyttä.

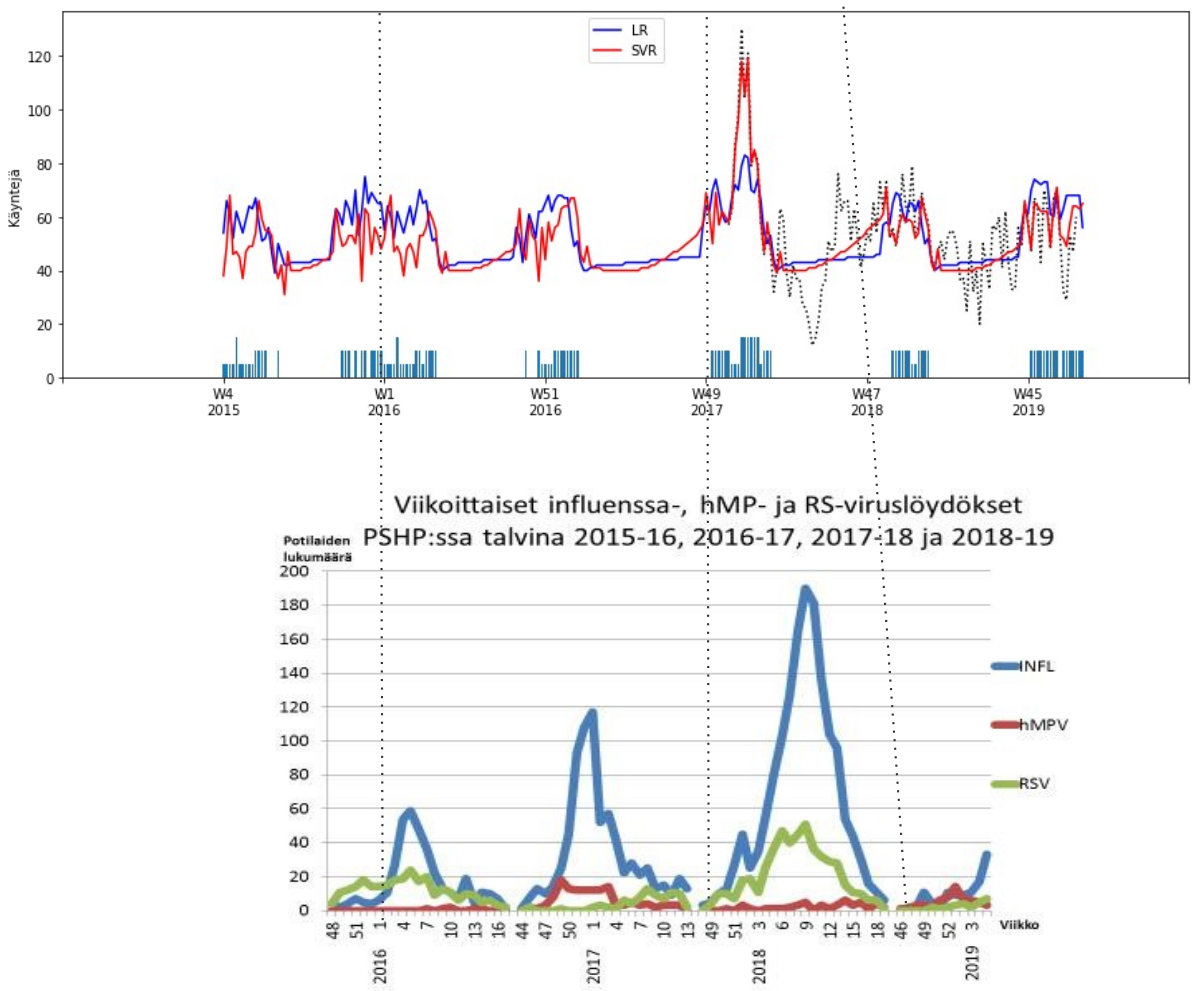


Kuva 7: Lineaariregressiomalli verrattuna metaoptimoituun tukivektorikoneeseen ja opetusaineistoon (musta katkoviiva).

Tällä ei ole vaikutusta käyttökohteessamme, eli ruokailijamäärien ennustaminen lukuvuoden aikana, mutta ilmiö on hyvä tunnistaa. Lisäksi, ennen mallin valintaa kannattaa varmistua niiden suorituskyvystä koko ajanjaksolle 1/2015-13/2020, jolle on saatavissa influenssa-aktiivisuusluokitukset viikoittaisesta etenemisestä sairaanhoitopiireittäin. Ennustemallien tulokset on esitetty alla (palkin korkeus kuvaa influenssaepidemialuokitusta Etelä-Pohjanmaan sairaanhoitopiirissä).

Lineaariregressiomalli (LR) näyttää olevan tukivektorikoneregressiota (SVR) vakaampi. Itse asiassa, SVR:n ennusteet, ehkä sen parametrien hakemiseen käytetystä vaivan vuoksi, näyttävät yleistyvän huonommin. Sen sijaan lineaariregressiomallin ulostulo näyttää suhteellisen järkevältä, joskin ylikonservatiiviselta Voimme vielä validoida tuloksia vertaamalla ennusteita Pirkanmaan sairaanhoitopiirin seurantatiedon kanssa. Jos otamme huomioon, että huiput ajoittuvat Pirkanmaalla ja Lapualla eri tavoin, näyttävät niiden suhteelliset koot ja ajankohdat järkeviltä. Esim. Ikonen et al. (2016, Kuva 6) “Influenssakausi Suomessa” seurantaraportti vuosille 2015-2016 vahvistaa alkuvuoden 2016 osalta, että Etelä-Pohjanmaa oli kyseisenä vuonna influenssaepidemiassa Pirkanmaata useita viikkoja edellä.

Päädyimme siis käyttämään lineaariregressiomallia ja täydentämään sillä ne viikot, joille ei ollut saatavilla THL:n käyntiseurantatietoja luokitteluista hengitystieinfektio (ARI) ja influenssan kaltainen tauti (ILI). Tämä ennustettu tieto ei ole täydellisen tarkkaa, etenkään koska kesän osalta influenssatasotieto puuttuu, mutta se silti mahdollisti käyntimäärien täydentämisen ja käyttämisen datalähteenä. Käyntitietojen käyttämisestä on monta etua: Terveyskeskuksen hengitystieinfektioihin liittyvä käyntimäärä on raportteja ajantasaisempi, tarkempi ja alueellisempi mittari influenssa-aktiivisuudelle.



Kuva 8: Etelä-Pohjanmaan influenssakauden perusteella ennustetut vierailut Lapuan terveyskeskuksella verrattuna Pirkanmaan sairaanhoitopiirin hengitystieinfektiolöydösten seurantatietoihin 2016-2019: https://www.tays.fi/fi-FI/Pirkanmaan_influenssaepidemia_on_kaynnis(83125)

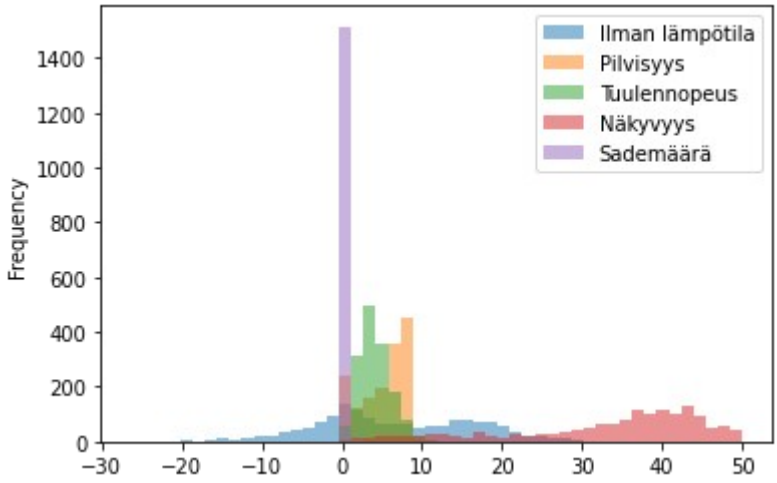
Säätietojen esiprosessointi

Ilmatieteenlaitos ylläpitää palvelua, josta sääasemien historiatiedot on ladattavissa CSV tai XLSX-muodoissa: https://www.ilmatieteenlaitos.fi/havaintojen-lataus#!/

Tietojen lisenssi on CC BY 4.0. Käyttöehtona siis on, että säätietojen alkuperä on ilmoitettava ellei Ilmatieteenlaitos erikseen vaadi maininnan poistamista. Lisäksi lentohavaintosääasemien tietoihin liittyy erityisehtoja, jotka tosin liittyvät enemmän vapautumiseen väärien tietojen aiheuttamista ongelmista FMI:n taholta.

Kohdekoulua lähimpänä oleva sääasema oli **Kauhavan lentoasema**, joka on linnuntietä n. 16,5 kilometrin päässä. Tuntikohtaiset tiedot kaikista mitatuista kohteista (mm. ilman lämpötila, näkyvyys, tuulen nopeus) ladattiin palvelun kautta CSV-muodossa alkaen

vuodesta 2016 ja päättyen vuoden 2020 tietoihin 30.3.2020 asti. Palvelusta ladattavien tietojen määrää on rajoitettu, joten lataukset tehtiin vuosi kerrallaan. CSV-muoto mahdollisti tietojen yhdistämisen tekstieditorissa tiedostoksi `FMI_Kauhava_säädata_2016-2020.csv`.



Kuva 9: Yhteenveto säätiedoista

Tuntikohtaisten tietojen sijaan olimme kiinnostuneita koulupäivän säästä. Päivä asetettiin alkamaan kahdeksalta ja päättymään kello 14:00 iltapäivällä. Vakioita `DAY_STARTS_HOURS` ja `DAY_ENDS_HOURS` muuttamalla ajanjaksoa voisi esim. kaventaa koskemaan vain ruokailuaikaa. Koska ladattu aineisto oli kohtalaisen suuri (yli 36587 riviä) päätettiin aineiston lukeminen tehdä käyttäen Python-kieleen sisäänrakennettua `csv`-modulia, joka suodattaa pois käyttämättömät tuntikirjaukset ilman lämpötilalle (degC), pilvien määrälle (1/8), tuulen nopeudelle (m/s), näkyvyydelle (km) ja sateen intensiteetille (mm/h). Päiväkohtaiset keskiarvot laskettiin käyttäen Pandas-taulukkoa ja tiedot tallennettiin prosessoitujen tiedostojen kansioon CSV-muodossa. Myöhempi tarkastelu paljasti, että puuttuvia arvoja oli etenkin tuulen nopeuden mittaukselle jonkin verran, mutta ne korvattiin lukemalla 3.5 m/s, joka on Kauhavan lentoaseman keskimääräinen tuulennopeus. Keskimääräistä arvoa käytettiin, sillä se vääristää aineistoa vähiten.

Hävikkiseurannan esiprosessointi

Hävikkiseurantataulukko (tiedosto `Hävikkiseuranta 20200317.xlsx`) näyttää rakentuneen jossain määrin orgaanisesti. Se ei siksi sellaisenaan ollut valmis koneluettavaksi ja erityisesti *“Selite”* sarakkeen tiedot piti muuttaa rakenteiseen muotoon, jotta niitä voitaisiin hyödyntää ennustusmallin rakentamisessa. Näin olikin tehty tilaajan toimittamassa päivitetyssä `Hävikkiseuranta 20200317_V2.xlsx`-tiedostossa. Vaikka rivitietoja olikin nyt paremmin, oli taulukko luonteeltaan muita aineistolähteitä moniulotteisempi. Siksi tiedosto osoittautui hankalasti koneluettavaksi ja siihen tehtiin käsin seuraavat muutokset kuitenkin niin, että tiedoston perusrakenne pysyi samana ja että kyseistä tiedostoa voitaisiin käyttää tulevien kirjausten pohjana:

- Otsikoita yhdenmukaistettiin.
- Ylimääräiset päivän summarivit poistettiin (summat olivat 4 kertaa).
- Erityisruokavalioruuat (niille päiville kuin tieto oli kirjattu selitykseen) siirrettiin ruoka-aineiden jatkeeksi omalle rivilleen. Näin voitiin kirjata niiden valmistusmäärät, hävikit ja uudelleenkäytöt kuten muidenkin ruoka-aineiden kohdalla. Samassa yhteydessä eritysruokavalioihin liittyvä erillinen määräsarake poistettiin. Huomioitavaa on, että erityisruokavalioiden valmistusmäärät ja hävikit on kirjattu vain osittain, mikä heikentää tämän tiedon hyödynnettävyyttä.
- Erityisruokavalioruokien lisääminen rivitiedoiksi vaikutti myös summariviin. Kaikki em. muutokset on korostettu keltaisella.
- Tiistain 10.03.2020 ruokailijamäärä oli epähuomiossa laskettu väärin. Korjattu selitteen mukaiseksi.
- Poistettu sarake *“Keittiö”* (aina sama).
- Saraketta F, *“Hävikki (kg)”*, näytetään käytetyn kahdella tavalla. Ruoka-ainekohtaisten hävikkien ilmoittamiseen ja 2019 osalta kokonaishävikin laskemiseen. Kokonaishävikkien laskemiset päiväkohtaiselle riville on otettu pois.
- Epäselvät, epävarmat ja puuttuvat hävikkitietosolut on korostettu punaisella, seuraavana päivänä tarjotut hyödynnetyt ruoka-aineet on merkitty oranssilla,
- Maito, näkkileipä ja levite oli merkitty rivitietoihin 2019, mutta niille ei ollut kirjauksia ja 2020 kyseisenkaltaisia rivejä ei enää ole kirjattu. Poistettu.
- Yhdenmukaistettiin ruoka-ainerivien järjestystä niin, että ne ovat aina:

    1. Pääruoka
    2. Lisuke (jos on)
    3. Kastike (jos on)
    4. Salaatti (jos on)
    5. Juures tai hedelmä (jos on)
    6. Erikoisleivät ja leivän päälliset tms. (jos on)
    7. Täydentävät tai korvaavat ruoka-annokset (jos on)
    8. Edellisen päivän uudelleen tarjotut (jos on)
    9. Erityisruokavalio (jos kirjattu)

Lisäksi aineistoa käsiteltäessä tehtiin lukuisia huomioita ja havaittiin potentiaalisia ongelmia. Alle on kirjattu se, miten ne on ratkaistu käsintehtävässä validoinnissa:

- Sitä, miten hävikkiä on hyödynnetty ei ole kirjattu jokaiselle päivälle. Esim. seuraavana päivänä tarjolla oleva salaatti olisi ollut hyvä merkitä selkeästi *seuraavan* päivän ruoka-aineiden joukkoon. Lisäksi, luultavasti yli jääneestä ja uudelleen-käytetystä ruuasta osa saattaa mennä jätteeksi, mikä tekee myös hävikkilukemista potentiaalisesti epätarkkoja. Nyt jää epäselväksi miten ja milloin hävikin uudelleenkäyttö on tehty. Vain jos uudelleenkäyttö on eksplisiittisesti mainittu, se on lisätty päivän ruoka-aineiden listaan.

- Selitekirjaukset ruokailijoiden osalta voidaan tulkita kahdella tavalla. Esim. "*406 henkilöä 15 erityisruokavaliota"* voi tarkoittaa sitä, että tavallisia ruokailijoita oli 406 ja lisäksi 15 erityisruokavaliota. Koska ruokailijamäärien kirjauksia ei ole tehty yhdenmukaisesti, on vaarana, että selitteen ruokailijamäärän tulkinnassa on tehty virheitä. Version 3 laatimisessa on käytetty tulkintaa, että erityisruokavalioruokailijat on aina merkitty edellisten lisäksi ellei toisin ole mainittu (ts. *"406 henkilöä 15 erityisruokavaliota"* tarkoittaa, että 406 lounasruokailijaa ja 15 erityisruokavalioruokailijaa). Toinen vastaava on kirjaus tiistailta 18.02.2020: "*Asiakkaita 431 / erityisruokavalioita 21 henk."* Onko erityisruokavaliollinen asiakas vai ei? Tulkinnanvaraiset kirjaukset on merkitty oranssilla.

- Viimeinen kirjauspäivä, eli tiistain 17.03.2020 näyttää olevan keskeneräinen. Se poistettiin versiosta 3.

- Joinain päivinä on selitteeseen merkitty, että eritysruokavaliosta hyödynnetään yli jääneet ruuat. Mikäli samana päivänä on merkitty, että myös linjastohävikistä hyödynnetään osa, on epäselvää pitääkö tämä sisällään myös eritysruokavaliohävikin tai muut erikseen mainitut hyödynnettävät hävikit. Versiota 3 tehdessä oletettiin, että hyödynnetty eritysruokavaliohävikki ja muut hävikit on jo laskettu selitteessä ilmoitettuun linjastohävikkiin mukaan.

Käsin tehtyjen muutosten seurauksena syntyi muokattu aineisto `Hävikkiseuranta 20200317_V3.xlsx`, joka luettiin ja muunnettiin rivimuotoon Jupyter-työkirjassa `TaydennaHavikit.ipynb`. Jälleen, perustelu koodissa tehtävässä muunnoksessa oli, että jatkossa täydentyvän aineiston lukemiseen voidaan käyttää samaa työkalua, mikäli tulevat merkinnät pidetään V3-muodossa.

Varsinainen työkirjan tekemä esiprosessointi koostuu kokonaan tyhjien rivien, viikkorivien, kuukausirivien ym. ylimääräisten rivien poistamisesta. Myös erityisruokavaliota koskevat erilliset hävikki ja valmistusmäärät sisältävät rivit poistettiin, sillä aineiston arvioitiin olevan erityisruokavaliota noudattavien niin pieni, että sille ei ollut vielä esiselvitysvaiheessa järkeä rakentaa omaa mallia. Koska erityisruokavalioiden hävikit ovat kuitenkin osa kokonais-hävikkiä (katso muokkauslista yllä), pidettiin sarake, jossa on ilmoitettu erityisruokavaliota noudattavien ruokailijoiden lukumäärä. Myös ruokalajikohtaiset valmistus-, hävikki-, ja menekkitiedot poistettiin. Vaikka nämä ovat rikasta tietoutta esimerkiksi toteutusvaiheessa toteutettavan palvelun käyttäjille, ennustusmalleille tämä tieto oli liian yksityiskohtaista.

Ruokalajitiedot, jotka olivat päiväysrivin ja yhteenvetorivin (jolla on ensimmäisessä sarakkeessa solu *'yht'*) välissä luettiin ja muutettiin ruokalistojen esiprosessoinnin yhteydessä kuvatulla *"Bag-of-words"*-piirteytysmenetelmällä täydentämään päiväriviä. Kuten aiemmin ruokalistojen esiprosessoinnin yhteydessä mainittiin, tämä piirteytyskoodi on erillisessä `menu_stemming.py` -tiedostossa, josta se ladataan tekstimuotoisten ruokalistasisältöjen piirteyttämistä tekevään Jupyter-työkirjaan.

Myös yhteenvetorivin yhteenvetotiedot otettiin talteen. Erityispäiväsarakkeen arvot muutettiin muodosta *'K'/'E'* kokonaislukumuotoon, eli 0/1 arvoiseksi muuttujaksi. Loput päivätietorivin saraketiedot, kuten ruokailijamäärä, erityisruokailijamäärä, lautashävikki, valmistushävikki, linjastohävikki, se kuinka paljon hävikistä hyödynnettiin, ja se oliko kyseessä erityispäivä, otettiin myös luonnollisesti mukaan rivitietoihin. Lopuksi kaikki nämä tiedot yhdistettiin takaisin yhteen tauluun, jossa on 36 riviä ja 99 muuttujaa ja jossa rivin indeksinä on päiväys.

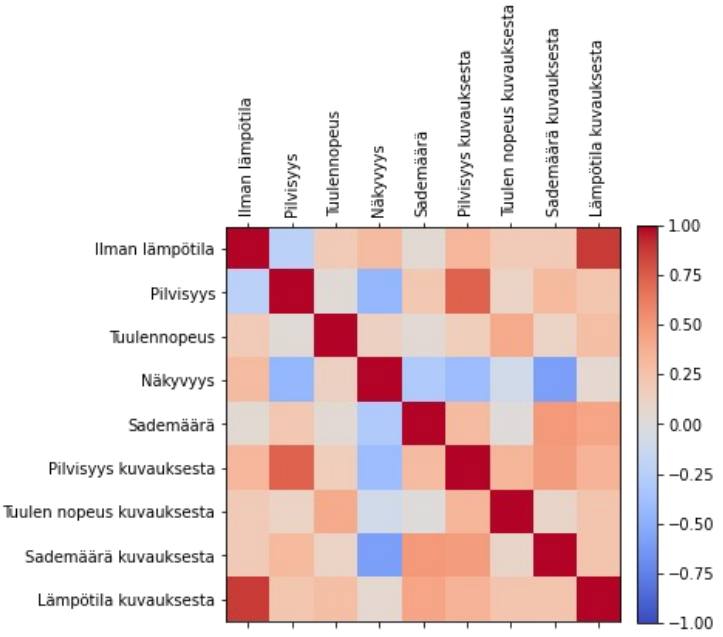
Taulukossa oli kuitenkin lukuisia yksittäisiä puuttuvia arvoja kuten:

- menekki, hävikki ja ruuan valmistusmäärä (kg) viideltä vuoden 2019 päivältä,
- ruokailijamäärät päiviltä 30.01.2020, 03.02.2020, 20.02.2020 ja 06.03.2020, mutta myös erityisruokavalioruokailijoissa oli puutteita (ei listattu),
- lautashävikki ja linjastohävikki puuttuvat päivältä 04.02.2020 ja
- valmistushävikki päiviltä 27.-29.01.2020, 04.02.2020, 12.02.2020, 13.03.2020.
- se, kuinka suuri osuus hävikistä hyödynnetään on annettu vain n. puolelle päivistä.

Koska rivimäärä on niin pieni ja puutteita paljon, päätimme käyttää aineiston täydentämiseen imputointia. Muut vaihtoehdot puuttuvan aineiston aiheuttamaan ongelmaan olisi ollut rivien ja sarakkeiden poisto. Valitettavasti tällöin aineistoa olisi jäänyt jäljelle vain muutamia rivejä, joten tämä ei ollut realistinen vaihtoehto. Käyttämämme lähinaapuri-imputointi täyttää puuttuvat arvot kahden samankaltaisimman rivin arvojen keskiarvolla. Epäluotettavimmat sarakkeet puuttuvien tietojen vuoksi ovat erityisruokavalioruokailijoiden lukumäärä ja hävikin hyödyntämismäärä (kg), mikä tulee huomioida ennustemalleja rakennettaessa. Täydennetyt tiedot tallennettiin CSV-muodossa täydennetyt_`havikit.csv` -tiedostoon.

Osana hävikkitietoja oli myös kirjaukset vallitsevasta säästä. Koska meillä oli jo esiprosessoituna Ilmatieteenlaitoksen säätiedot, päätimme vertailla näiden kahden kirjauksen samankaltaisuutta. Sanalliset kuvaukset hävikkiaineistossa muutettiin suuripiirteisesti numeerisiksi arvoiksi pilvisyydestä, tuulen nopeudesta ja sademäärästä. Sen jälkeen säätietoja voitiin vertailla niiden korrelaatioiden avulla (Kuva 10).

Erityisesti lämpötilakirjaukset ja pilvisyys näyttävät olleen hyvin samankaltaiset. Sademäärä ja näkyvyys myös korreloivat hyvin vahvasti. Tekstimuotoisesta kuvauksesta asetetut ja Kauhavan säähavaintoaseman datasta peräisin olevat sademäärät sen sijaan eivät korreloi kovin hyvin. Tämä saattaa johtua mittauspaikkojen etäisyydestä (16,5 km), sateiden kuurottaisuudesta, tai sitten tavasta, jolla kuvausmuotoiset sademäärät muutettiin numeerisiksi arvoiksi.



Kuva 10: Kahdesta eri lähteestä peräisin olevien säätietojen väliset korrelaatiot.

Epätarkkuudesta huolimatta päivittäiset säätiedot ovat riittävän lähellä, jotta voimme tehdä päätöksen käyttää mallien opettamisessa Ilmatieteenlaitoksen monipuolisempaa dataa. Pudotimme siis hävikkitiedostoon kirjatut säätiedot pois. Suosituksena onkin, että säätietoja ei enää kerätä käsin, sillä ne ovat suuremmalla tarkkuudella ja yhdenmukaisemmin saatavissa muualta.

Koneoppimisessa hyödynnettävät aineistot

Esiprosessoinnin tuloksena `data/prosessoidut` -kansiossa on joukko tiedostoja, jotka luetaan silloin kun koneoppimismalleja rakennetaan. Alla on listattu nämä tiedostot ja niiden kuvaukset:

`data/prosessoidut/paivittaiset_ruoka-avainsanat_lv<X>.csv`

> Tämä tiedosto pitää sisällään päivittäisten ruokalistojen avainsanat koodattuna piirteiksi (ns. *bag-of-words, BOW* piirteytys). Ensimmäinen sarake on päiväys, ja loput kertovat montako kertaa otsikon mukainen ruoka-aineen avainsana (esim. *“kana”*) esiintyy päivän ruokalistan kuvauksessa. Tarkemmin tämä aineisto on kuvattu tämän raportin esiprosessointia käsittelevässä luvussa. Tiedoston nimessä tunniste <X> ilmaisee sen, montako ensimmäistä pilkulla eroteltua kuvausta on otettu huomioon. Jos tämä on `X=MAX`, koko esiprosessoitu kuvaus on otettu huomioon. Tätä voi vaihtaa `LueJaPiirteytaRuokalistat.ipynb` -prosessointiohjelman `USE_N_FIRST_DISHES` vakiota muuttamalla.

`data/prosessoidut/normalisoidut_suoritteet.csv`

Tähän tiedostoon on koottu kuukausikohtaisissa tiedostoissa kuvattu suoriteaineisto. Sitä on radikaalisti suodatettu poistamalla huonolaatuiset ja epäolennaiset suoritekirjaukset sekä suurimmat poikkeamat. Suoritemäärät on normalisoitu käyttäen kunkin lukuvuoden ruokailijamäärää jakajana (josta on vähennetty arvio eritysruokavalioruokailijoista). Lopullisessa suoriteaineistossa on siis lounassuoritteiden suhteellinen määrä, tehdyt eväät, tarjotut jäätelöt sekä kalenteritieto (viikonpäivä ja viikkonumero). Tämän aineiston laatu on tehdystä esiprosessoinnista huolimatta ehkä koneoppimista varten valmistelluista aineistoista heikoin.

`data/prosessoidut/hengitystieinfektiot.csv`

Tiedostoon on kerätty Lapuan terveyskeskuksessa tehtyjen hengitystieinfektion tai sen kaltaisten oireiden vuoksi tehtyjen käyntien viikoittaiset määrät vuoden 2015 alusta aina kevääseen 2020. Rivillä on jokaiselle päivälle sen edeltävän viikon ja sitä edeltävän viikon käynnit. Vuoden 2018 alusta käynnit ovat AvoHILMO ilmoitusten mukaiset, mutta sitä vanhemmat on ennustettu lineaariregessiomallin avulla käyttäen syötteenä kyseisen viikon ja neljän ympäröivän viikon THL raportoimia influenssaepidemiatasoluokituksia (THL 2020). Käyntimäärät siis eivät ole tarkkoja 31.12.2017 asti, vaan ennustemallin antamia arvioita.

`data/prosessoidut/saatiedot.csv`

Tiedostoon on koostettu Kauhavan lentoaseman säätiedot tammikuusta 2016 aina viime päiviin asti. Kunkin päivän klo 8:00-14:00 välillä vallinnut säätila on ilmaistu ilman keskimääräisen lämpötilan, pilvisyyden, tuulen nopeuden, näkyvyyden ja sademäärän avulla.

Edellä kuvatut aineistot yhdistettiin suoritteiden ennustamista varten niin, että jokaisella lopullisen aineiston rivillä oli kaikki tiedot: eväät, kalenteritiedot, edellisten viikkojen hengitystieinfektiokäynnit, sää ja ruoka-ainesanat. Aineistot ladattiin tiedostoista ja yhdistettiin päivänmäärän perusteella. Yhdistetyn aineiston eheys todettiin pistokoneella, jossa kolme datapistettä tarkistettiin käsin (eli päivän infektiotiedot, ruoka-avainsanatiedot, suoritteet ja sää). Ruoka-avainsana-aineistosta käytettiin suppeinta, vain pääruuan nimen koodaavaa aineistoa. Näin saatiin aineisto jossa olii 51 piirrettä (selittävää muuttujaa) ja 377 datapistettä (riviä). Syntynyt taulukko on tallennettu tiedostoon `data/prosessoidut/yhdistetty_aineisto.csv`. Hävikkiaineistot puolestaan on esiprosessoitu erikseen ja tallennettu omaan tiedostoonsa:

`data/prosessoidut/taydennetyt_havikit.csv`

Hävikkitiedot rivimuodossa ja piirteytetyillä ruokalistoilla. Puuttuvat tiedot on imputoitu lähinaapurimenetelmällä, mikä tosin tekee erityisruokavalioruokailijoiden ja uudelleenkäytettyjen hävikkien määristä jossain määrin epäluotettavia. Silti, vähäinen rivimäärä ei tarjonnut muuta vaihtoehtoa ja tämä tulee vain huomioida tuloksia tulkittaessa.

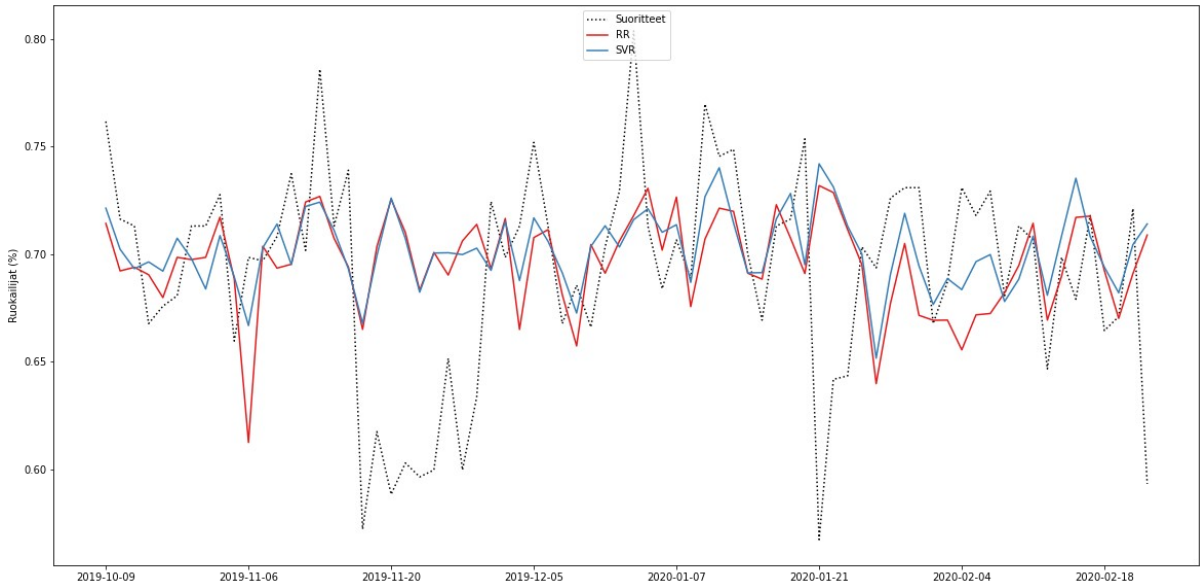
Tulokset ennustemallien rakentamisesta

Ruokailijasuoritteiden ennustemallin opetus

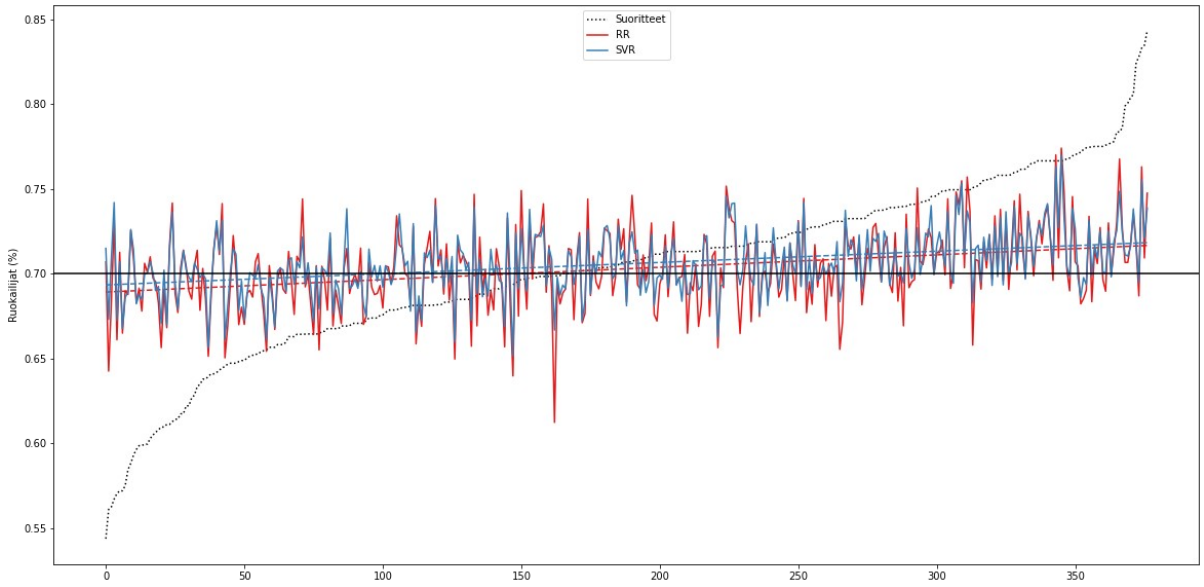
Yhdistetty aineisto ladattiin tiedostosta ja kaikki piirteet (sarakkeet) skaalattiin välille [0.0, 1.0], jonka jälkeen aineistoa yritettiin mallintaa sovittamalla lineaariregressiomalli (LR) ja tukivektorikoneregressiomalli (SVR). SVR on parametrisoitu, joten sen parametreja metaoptimoitiin mallityypin suorituskyvyn maksimoimiseksi. Hyvyysmetriikkana käytettiin ennustusvirheen neliösumman minimointia. Näin opetettu lineaariregressiomalli antoi joitain hyvin suuria tai pieniä arvoja, mikä viittasi siihen, että mallin regularisaatiota tulisi lisätä. Kokeilimme Tikhonovin regularisaatiota (tunnetaan myös nimellä eng. *Ridge Regression,* RR, Hoerl & Kennard 1970), mikä silmin nähden auttoi välttämään ylioppimista.

Valitettavasti suoritteiden määrän ennustaminen käytettävissä olevilla piirteillä osoittautui erittäin vaikeaksi. Edes metaoptimoitu SVR ei juuri pysty parempaan kuin naiivi lähestymistapa, jossa ennustetaan joka päivälle keskimääräinen suoritemäärä: naiivin lähestymistavan keskimääräinen virhe oli 25,4 suoritetta kun RR:lle se oli 24,4 ja SVR:lle parhaimmillaan 23,6. Suurimmat ennusteiden virheet olivat jopa n. 110 suoritetta. Mallintamisen vaikeus käy hyvin ilmi Kuvasta 11, missä ennusteiden (sininen ja punainen viiva) ja todellisten suoritemäärien (musta katkoviiva) välillä on hyvin suuria poikkeamia. Toisaalta, poikkeamista näkee, että huomattavan suuret ja pienet suoritemäärät ovat vaikeimpia ennustaa ja että tavanomaisina päivinä suorite-ennustetarkkuus on parempi, mitä tunnusluvut antavat ymmärtää. Näyttääkin siltä, että ne päivät tai jopa viikot, jolloin jostain syystä on tehty huomattavasti yleistä tasoa pienempi tai suurempi määrä suoritteita ovat vaikeita ennustaa luotettavasti käytettävissä olevan syötetiedon perusteella (eli päivän ruokalista, eväiden määrä, säätiedot, edellisen ja sitä edeltävän viikon hengitystieinfektiokäynnit lähimmästä terveyskeskuksesta).

Ensimmäisen yrityksen jälkeen näytti siis siltä, että mallit eivät opi riittävästi ollakseen käyttökelpoisia. Syy voi olla moninainen: dataa voi olla liian vähän tai/ja sen laatu voi olla puutteellista (ns. *garbage-in-garbage-out* -efekti). Tai, kuten edellä pohdittiin, kenties piirteet ovat vääränlaiset tehtävää ajatellen, ja eniten selitysvoimaa olisi sellaisilla tekijöillä, joita ei esiselvitysvaiheessa ollut saatavilla? Joka tapauksessa, on valitettavasti vaikea arvioida hyvään suorituskykyyn tarvittavan datan määrää, jos ennustustarkkuus ei juuri eroa satunnaisuudesta tai aina saman keskimääräisen suoritemäärän veikkaamisesta.



Kuva 11: Aineiston viimeisimmän sadan päivän suoriteennusteet.



Kuva 12: Ennustustarkkuus järjestettynä suoritemäärän mukaan.

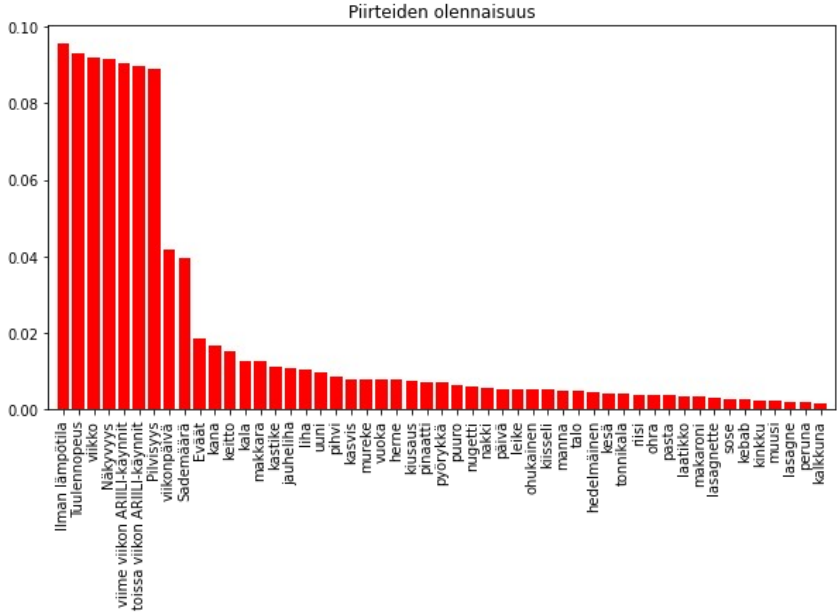
Ennusteiden satunnaisuus korostuu kun ennusteet järjestetään kasvavan todellisen suoritemäärän mukaan, kuten on tehty Kuvassa 12. Hyvin toimiva ennustusmenetelmä seurailisi mustan katkoviivan muotoa, mutta nyt molemmat ennustemallit tuottavat lähes tasaista kohinaa. Sovittamalla suorat (katkoviivat) ennustetuloksiin nähdään, että ennustetut suoritteet kasvavat hieman oikealle mennessä, eli ne oppivat joitain tekijöitä, mitkä aiheuttavat suurempia suoritemääriä. Silti, mallit eivät opi tarpeeksi ilmiöstä ja muuttujien välisistä yhteyksistä tuottaakseen luotettavia ennusteita päivittäisille suoritemäärille.

Satunnaismetsät, piirteiden tärkeyden arviointi ja dimensionpienennys

Koska yksinkertaisemmat regressiomallit eivät tuottaneet käyttökelpoisia tuloksia, yritimme vaihtoehtoista lähestymistapaa eli satunnaismetsiä. Perusteluna menetelmän käytölle oli, että niin sanottu *dimensionaalisuuden kirous,* eli liian suuri piirteiden määrä, saattaa joskus aiheuttaa sen, että hyviä malleja ei synny. Lisäksi satunnaismetsät ovat hyvin vika- ja häiriösietoinen menetelmä, minkä pitäisi auttaa mahdollisten aineiston laadullisien ongelmien kanssa.

Itse aineiston “leveyden”, eli ulottuvuuksien vähentämiseen on useita erilaisia menetelmiä. Niistä eräs tunnetuimmista on pääkomponenttianalyysi (PCA), joka muodostaa aineiston sarakkeista uudet lineaarikombinaatiot niin, että ne selittävät mahdollisimman suuren osan alkuperäisen aineiston vaihtelusta. Ensimmäiset näistä uusista sarakkeista, eli ne pääkomponentit, voidaan tulkita suuntina, joissa aineistossa on eniten vaihtelua. Tavallaan pyritään siis välttämään tilanne, jossa annettaisiin oppivalle algoritmille kaksi samaa asiaa selittävää ja siten vahvasti korreloivaa muuttujaa. Kokeilimme pääkomponenttianalyysiä ja vähensimme sillä piirteiden määrää asteittaisesti, mutta valitettavasti se ei auttanut parantamaan ennustustarkkuutta yksinkertaisempien regressiomenetelmien tapauksessa.

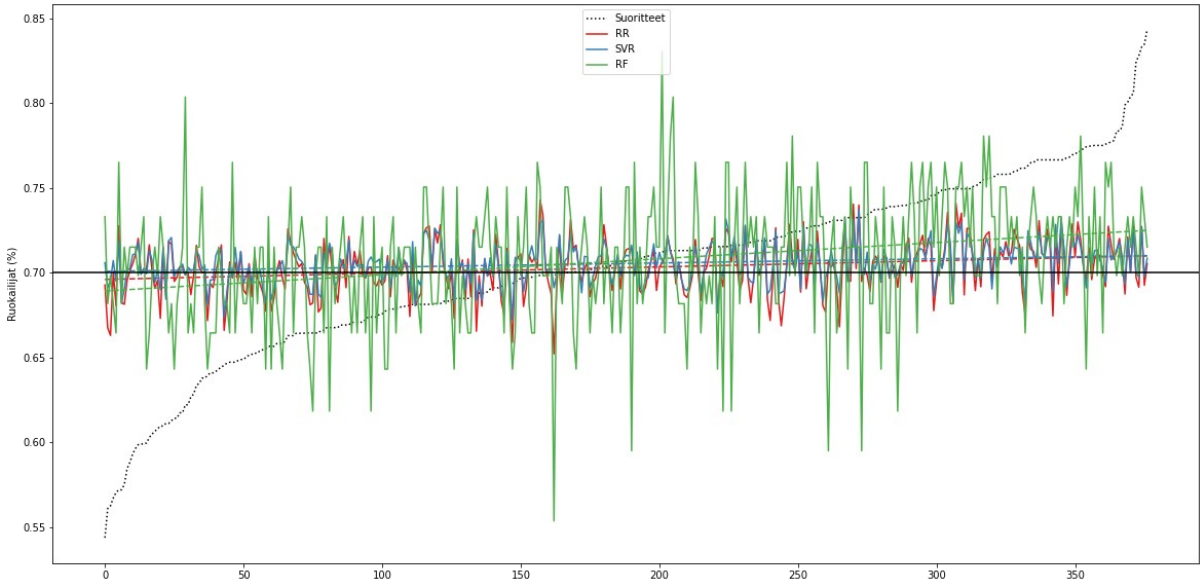
Siitä huolimatta päätimme kokeilla satunnaismetsiä piirteiden valitaan. Ne perustuvat siihen, että opetetaan suuri joukko päätöspuita, jotka sitten “äänestävät” lopputulosta. Kunkin puun opettamiseen käytetään eri piirteitä ja otantoja aineistosta, mikä saa ne katsomaan aineistoa eri näkökulmista. Satunnaismetsät ei kuitenkaan ole regressiomenetelmä, vaan luokittelumenetelmä (eng. *classifier*), joten sitä varten suoritteet pitää diskretisoida "sankoihin", joissa kussakin on suunnilleen samansuuruisia suoritemääriä. Suoriteaineistolle käytettiin 15 sankoa *k*-means ryvästykseen lajittelulla.



Kuva 13: Piirteiden olennaisuusarviointi satunnaismetsien avulla.

Satunnaismetsien etuna on, että niiden avulla voidaan tuottaa piirteille tehtävän kannalta olennainen tärkeysjärjestys. Nämä tulokset on esitetty Kuvassa 13. Selkeästi tärkeimmät piirteet ovat lämpötila, tuulennopeus, viikkonumero, influenssa- ja hengitystieinfektio-aktiivisuus, näkyvyys ja pilvisyys. Piirteiden tärkeystarkastelussa kannattaa kuitenkin huomioida, että satunnaismetsien ennustustarkkuus on myös kohtalaisen heikko. Täten, malli ei välttämättä ole oppinut oikeita asioita suoritemäärän ennustamiseksi, jolloin myös piirteiden järjestys ei välttämättä vastaa toimivamman mallin tuottamaa olennaisuustietoa. Silti, olennaisuustarkastelu antaa lisää tietoa todennäköisesti tärkeimmistä piirteistä. Esimerkiksi tärkein suoritemäärää ennustava ruoka-aine näyttäisi olevan kana, mikä ehkä on odotettu ja intuition mukainen tulos.

Tässä yhteydessä kannattaa huomioida, että satunnaismetsien olennaisuuspisteytys saattaa merkittävästi erota korrelaatiotarkastelusta. Myös korrelaatioita nimittäin voidaan käyttää piirteiden valinnassa, mutta satunnaismetsien käyttämä lähestymistapa, jossa rakennetaan satoja heikkoja luokittimia eri näyte- ja piirreyhdistelmillä tunnistaa myös erityistilanteet ja niissä ennustuspäätösten kannalta hyödylliset piirteet nousevat paremmin esiin.



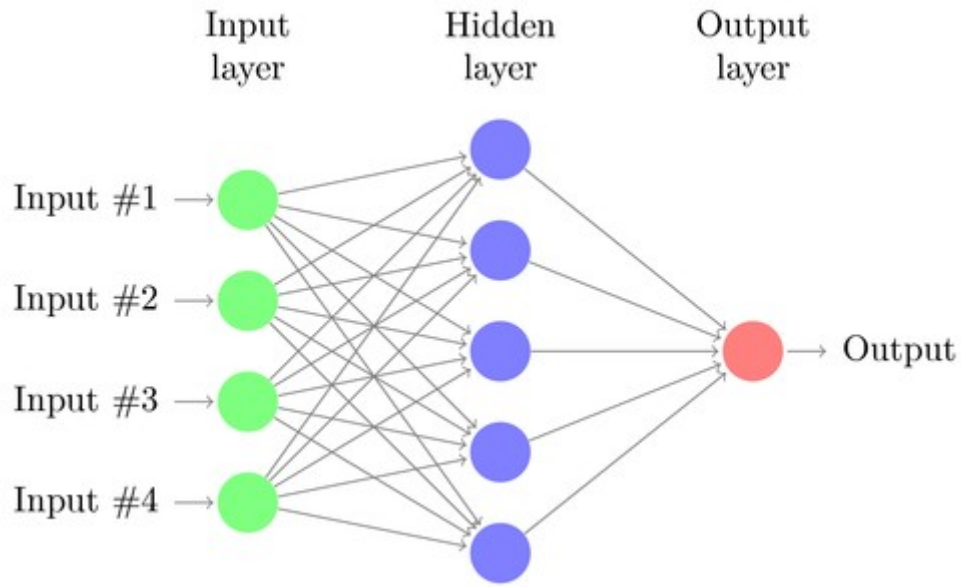
Kuva 14: Satunnaismetsien ennustustulokset verrattuna muihin malleihin kun mallin opettamiseen käytetään vain 12 tärkeintä piirrettä.

Satunnaismetsien tulos pitää palauttaa reaaliarvoksi tekemällä diskretisoinnille käänteinen muunnos. Mikäli vertaamme näin saatuja tuloksia RR- ja SVR-malleihin (esim. Kuva 14), satunnaismetsä näyttää oppivan hieman paremmin milloin suoritemäärä on suurempi, mutta diskretisoinnin vuoksi ennusteet voivat saada vain tiettyjä arvoja, jolloin keskimääräinen ennustustarkkuus ja mallin hajonta on selkeästi edellisiä menetelmiä heikompi.

Mallinnusyritys neuroverkoilla

Neuroverkot ovat viime vuosina osoittautuneet erinomaisen hyviksi erilaisissa kuvankäsittely-, kuvantunnistus- ja signaalinkäsittelytehtävissä. Niiden etuna on, että ne

osaavat oppia monimutkaisia riippuvuuksia. Ne ovat teoreettisesti universaaleja approksimaattoreita, eli kykeneviä oppimaan minkä tahansa regressiotehtävän. Neuroverkkojen käytön käytännön haasteena on, että mallien opettamiseen tarvitaan suuri määrä dataa ja laskentakapasiteettia. Lisäksi metamallin laatimiseksi pitää asettaa monta parametria, joilla määritellään neuroverkon arkkitehtuuri, opetusmenetelmä ja sen konvergenssiin vaikuttavat painotukset. Ne ovat siis ehkä yksi tehokkaimmista, mutta myös samalla vaikeimmin käytettävistä menetelmistä.



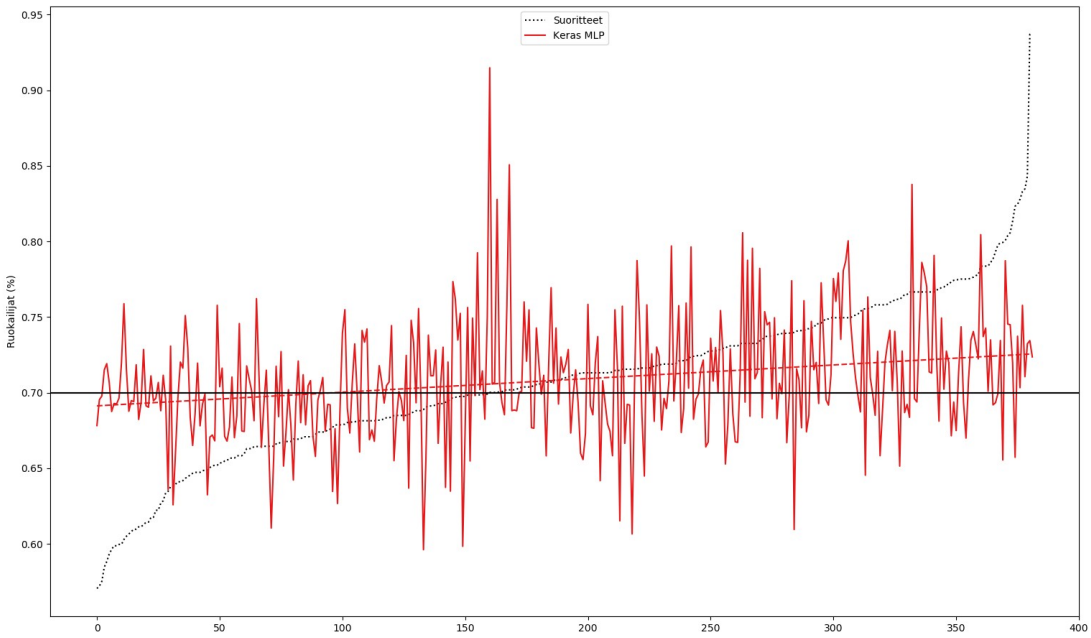
Kuva 15: Esimerkki pienestä neuroverkosta.

Silti, koska yksinkertaisemmat menetelmät eivät kyenneet löytämään eri piirteiden välisiä yhteyksiä koululounassuoritteiden määrään, kokeilimme käyttää neuroverkkoja *state-of-the-art* -koneoppimiskirjastona tunnetun Keras-rajapinnan kautta. Varsinaisessa laskennassa käytettiin Theano-kirjastoa. Vaadittavien riippuvuuksien vuoksi tämä kokeilu ei ole toteutettu Jupyter-työkirjaan, vaan kirjoitettu suoritettavaksi Python-skriptiksi. Se on luettavissa ja ajettavissa `src/keras_predict.py` -tiedostosta, mutta se vaatii ylimääräisiä toimia riippuvuuksien asentamiseksi.

Käsin tehdystä sopivan verkkoarkkitehtuurin etsinnästä ja eri opetusmenetelmien kokeilusta huolimatta monikerrokselliset neuroverkot (MLP) eivät tuottaneet juuri parempia ennusteita kuin yksinkertaisemmatkaan menetelmät. Arkkitehtuuria valittaessa tunnistettiin, että tehtävän mallintamiseen tarvitaan paljon epälineaarisia riippuvuuksia ja siksi on valittu kolmen piilokerroksen verkko, jossa ulostulokerroksen *tanh*-aktivointifunktio pitää huolen siitä, että ulostulo on välillä [0.0,1.0]. Arkkitehtuurin ja opetusmenetelmän optimointiin olisi voinut käyttää jotain hyperparametrien hakua, mutta tämän vaiheen aika- ja työmäärärajoitteet eivät sallineet tämän laajuisia kokeiluja.

Muiden menetelmien kanssa käytetty *leave-one-out* -ristiinvalidointi ei toiminut Keras-verkon kanssa, mutta käytimme sen sijaan 100-jaon ristiinvalidointia (eng. *100-fold cross-validation*), jolloin MLP:n keskimääräiseksi virheeksi jäi n. 29.0 oppilasta. Keras-MLP-mallin

tuloksia on esitelty Kuvassa 16. Kuten edellä on mainittu, tulos on samankaltainen muiden menetelmien kanssa. Lisäksi kohinaa on edelleen paljon, eikä malli näytä luotettavasti oppivan muita menetelmiä paremmin suoritemäärän säännönmukaisuuksia.



Kuva 16: Keras-neuroverkkomallin ennusteet

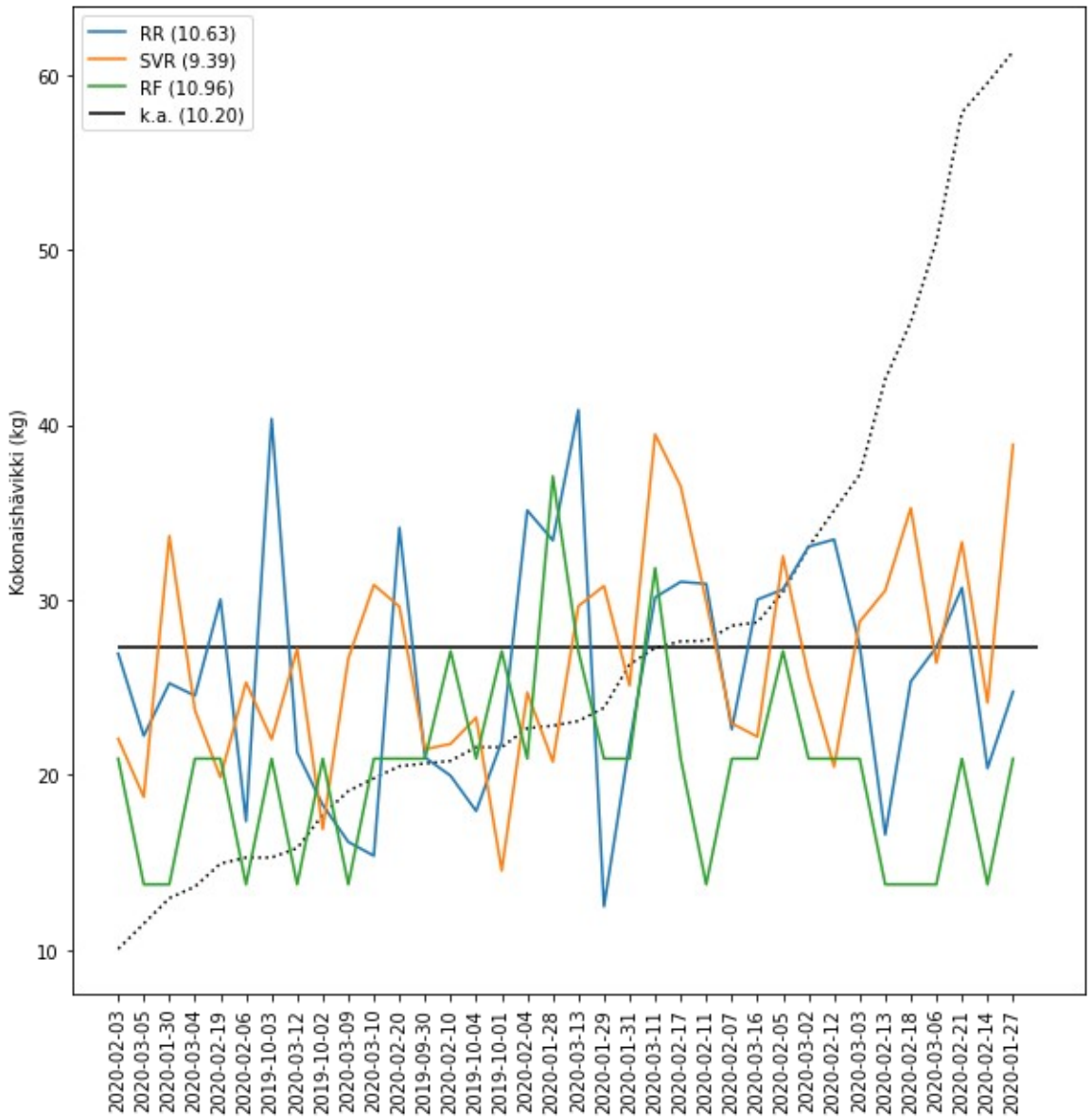
Tulokset satunnaismetsillä ja neuroverkoilla olivat tavallaan odotettavissa, sillä kokemuksemme mukaan kehittyneemmät menetelmät voivat kyllä parantaa tuloksia, mutta harvoin pystyvät korjaamaan tilannetta, jossa ilmiötä on vaikea mallintaa muilla, yksinkertaisemmilla menetelmillä. Suurempi ja korkealaatuisempi aineisto saattaisi mahdollistaa paremmat perustason tulokset ja siten myös kehittyneempien menetelmien käytön, sillä näin niiden avulla voitaisiin parantaa ennustustarkkuutta verrattuna yksinkertaisten mallien tuloksiin. Tällöin suositus olisi käyttää jotain neuro-verkkojen hyperparametrien etsimiseen tarkoitettua työkalua, sekä joko Nvidian CUDA-teknologiaa tai pilvipalveluratkaisua parametrien etsinnän nopeuttamiseen, sillä parametrien haku vaatii yleensä minuuttien tai tuntien sijaan useita päiviä prosessointiaikaa.

Ruokahävikkien ennustemallin opetus

Rivitiedoiksi muutettu hävikkiaineisto ladattiin CSV-tiedostosta, sitä täydennettiin säätiedoilla ja kalenteritiedoilla, jonka jälkeen kaikki piirteet (sarakkeet) skaalattiin välille [0.0, 1.0]. Toisin kun suoriteaineiston kanssa, hävikkiaineistoa ei normalisoitu lukuvuoden ruokailijamäärän suhteen, sillä sen tiedettiin olleen (yksityiskohtaisemman tiedon puuttuessa) vakio koko mittausajan. Mikäli myöhemmin tuodaan aineistoa muilta kouluilta, kannattaisi myös ruokahävikkiaineiston ruokailijamäärät ja hävikit muuttaa suhteellisiksi arvoiksi, jotta ne olisivat keskenään vertailukelpoisia.

Kokeilimme jälleen rakentaa ennustemallit käyttämällä Ridge-regressiota, tukivektorikone-regressiota metaoptimoinnilla ja satunnaismetsiä ja *leave-one-out*-ristiinvalidointia.

Ennustemalleja verrattiin perustasoon, joka oli hävikin keskiarvon suuruinen vakioennustus joka päivälle. Kokonaishävikkiä (linjasto+tarjotin+valmistus) ennustettaessa vain tukivektori-koneregressio oppii edes vähän jotain hyödyllistä, sillä se on ainoa, joka voittaa keskiarvoa (k.a.) ennustavan perustason (Kuva 17). Tulokset eivät siis ole kovin lupaavia, mutta jo intuitio sanoo, että toiset hävikkityypeistä ovat helpompia ennustaa kuin toiset.

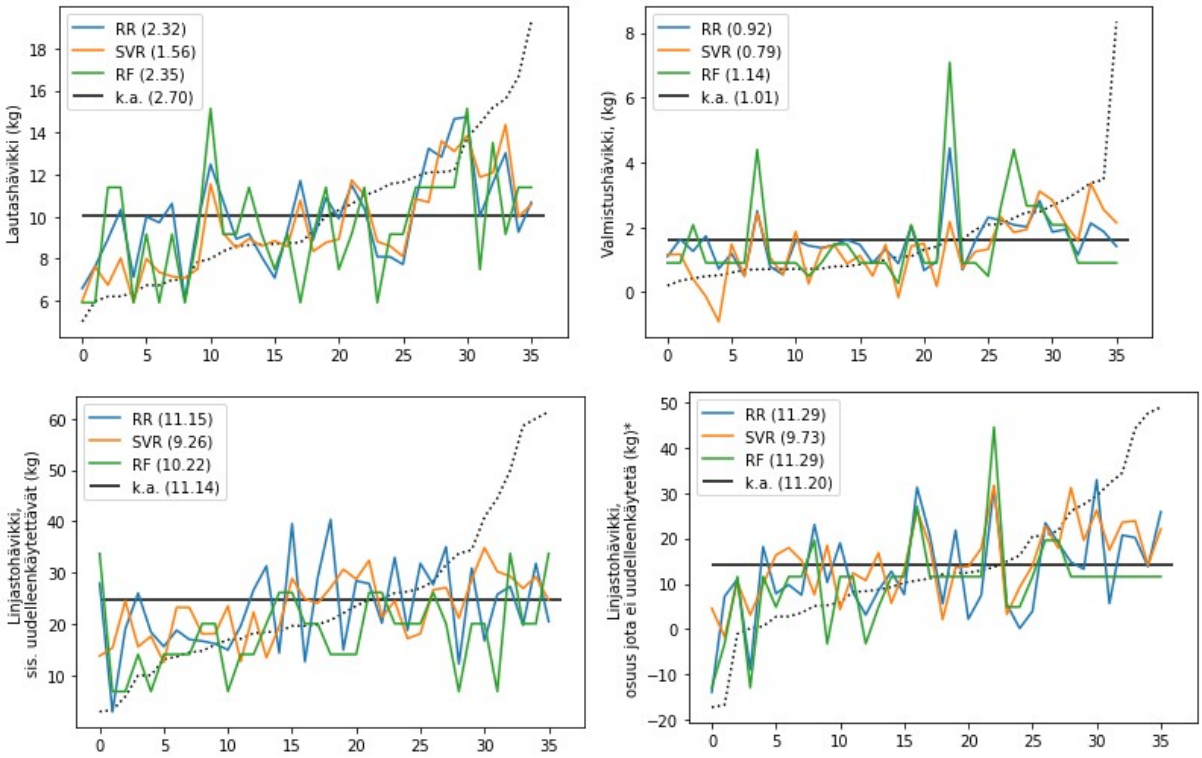


Kuva 17: Ennustettu kokonaishävikki (kg).

Kannattaa siis tutkia myös lajikohtaisia hävikkejä, eli linjastohävikkiä, valmistushävikkiä sekä lautashävikkiä ja niiden ennustamista. Erityisesti lautashävikki ja valmistushävikki ovat muuttujia, joiden ennustaminen saattaisi olla mahdollista hyödyntäen ruoka-ainepiirteitä.

Tulokset (Kuva 18) näyttävät vahvistavan nämä odotukset. Kun ruokailijamäärä tunnetaan, lautashävikkiä ja valmistushävikkiä voidaan todella ennustaa huomattavan paljon perustasoa paremmin. Erityisesti metaoptimoitu tukivektorikoneregressio näyttää pystyvän kohtalaisiin ennustuksiin jopa näin pienellä aineistolla. Sen keskimääräinen virhe lautashävikin ennustamisessa on 1,65 kg ja valmistushävikin ennustamisessa 0,68 kg. Huomattavaa on, että valmistushävikin tapauksessa yksi hävikeistä (keskiviikko 04.03.2020) on mahdollisesti virhekirjaus, sillä se on huomattavasti suurempi kuin muut kirjatut valmistehävikit. Tämä heikentää käyttämäämme hyvyysmittaria huomattavasti. Tuloksista näkee myös, että esim. päivän 10.3.2020 ennustus on aivan pielessä, mutta valitettavasti tuloksista on nopeasti vaikea nähdä miksi. Mallin tarkentaminen vaatisikin näiden erikoistapausten jahtaamista ja eliminoimista. Ajankäytöllisesti parempi lähestymistapa saattaa kuitenkin olla hankkia lisää korkealaatuista aineistoa mallin opettamiseen ja mahdollisesti lisätä mallin regulointia.

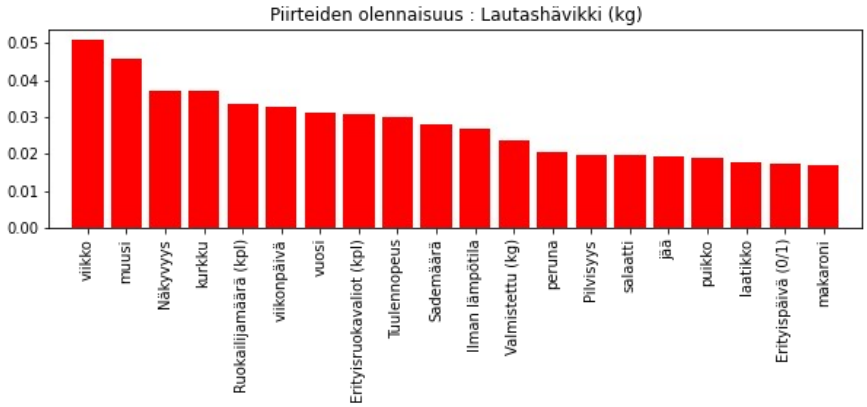
Linjastohävikin, erityisesti suuren sellaisen, ennustaminen näyttää sen sijaan olevan hyvin vaikeaa. Tämä on harmillista, sillä se on se hävikin muoto, mikä hyötyisi eniten toimivasta ennustemallista. Kenties suurempi aineisto mahdollistaisi tarkemmat ennusteet?



Kuva 18: Ennustetut osahävikit kun ruokailijamäärät tiedetään.

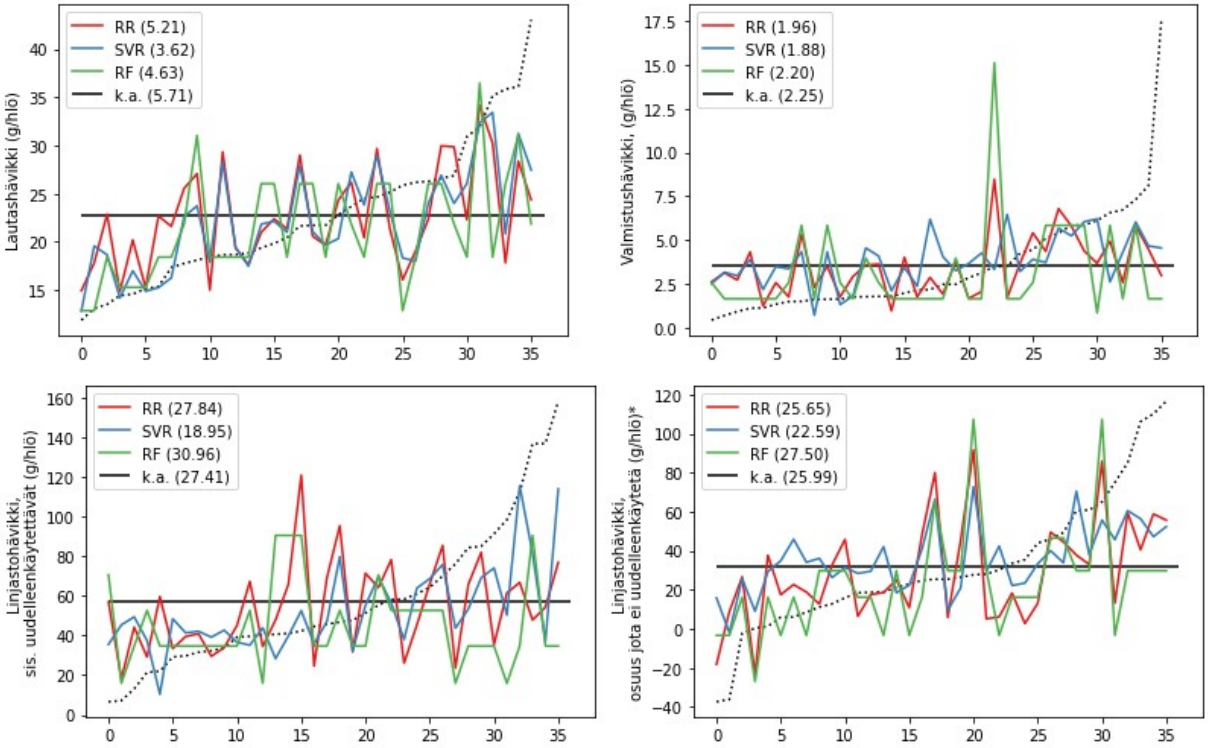
Voimme myös tutkia ennustemalleihin vaikuttavia tekijöitä aiemmin kuvatulla menetelmällä, jossa satunnaismetsän opettamalla saamme piirteet olennaisuusjärjestykseen. Olennaisuusjärjestys ei yksinomaan vaikuta muuttujan korrelaatio hävikkien kanssa, vaan myös piirteiden erottelukyky erikoistapauksissa on tärkeää. Kuvassa 19 on esimerkin vuoksi 20 satunnais-metsille olennaisinta piirrettä lautashävikin ennustustehtävässä.

Lopullinen ruokailijamäärä ei todellisuudessa olisi käytettävissä, vaan meillä olisi vain jonkinlainen arvio tai ennuste ruokailijamäärästä. Approksimaationa voimme kuitenkin laskea ruokailijakohtaiset hävikkitiedot käyttämällä todellisia ruokailijamääriä jakajina ja yrittää ennustaa niitä.

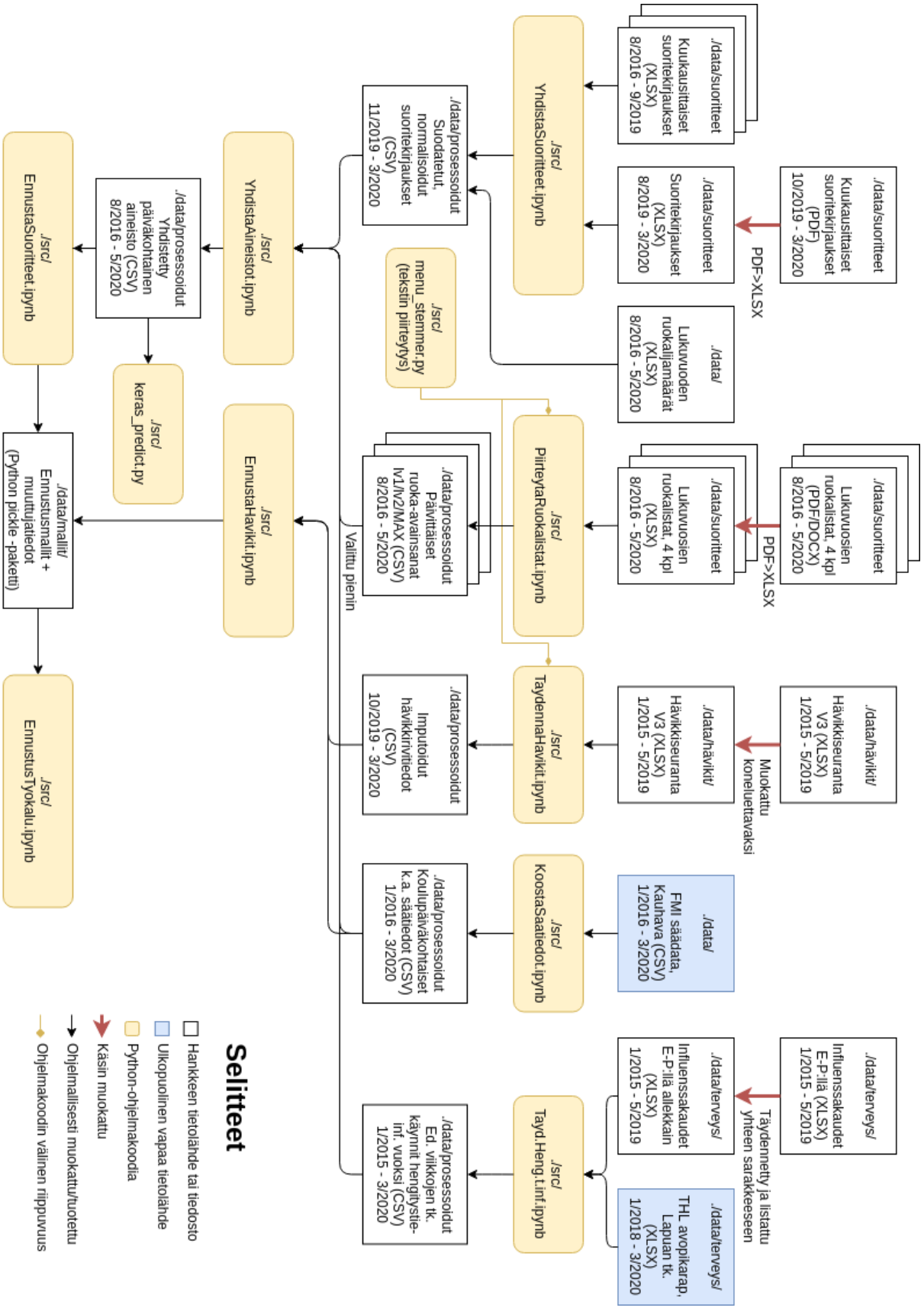


Kuva 19: Satunnaismetsien 20 tärkeintä piirrettä lautashävikkiä ennustettaessa

Samassa yhteydessä muutimme ennustettavan hävikki yksikön grammoiksi henkilöä (ruokailijaa) kohden, samoin valmistusmäärän. Tuloset ovat hyvin samansuuntaisia aiempien kanssa, missä ruokailijamäärä oli suorana syötteenä.



Kuva 20: Ennustetut hävikit ruokalijaa kohden



Kuva 21: Aineistot ja niiden käisttelyyn käytetyt työkirjat

Laskentaympäristö ja ennustemallin koekäyttö

Kaikki tämän raportin ajot tehtiin käyttäen kahdeksanytimistä AMD FX-8320 prosessoria, 8 GB muistia, Ubuntu 20.04 käyttöjärjestelmää, Python versiota 3.8.2 ja Jupyter ympäristöversiota 6.0.3. Projektia varten tehtiin virtuaaliympäristö (`venv`), jonka ansiosta tarvittavat Pyhon-paketit ja niiden versiot on dokumentoitu `requirements.txt` -tiedostoon. Vastaavan ympäristön saa asennettua edellyttäen, että koneella on Python 3.8 seuraavasti: Siirry ensin komentokehoitteessa siihen kansioon, minne laitoit esiselvitysvaiheen kansiorakenteen. Luo sinne uusi virtuaaliympäristö, ja asenna tarvittavat paketit:

```
$ cd /home/you/Projects/wasteless/

$ python3 -m venv .wastelessvenv

$ pip3 install -r requirements.txt
```

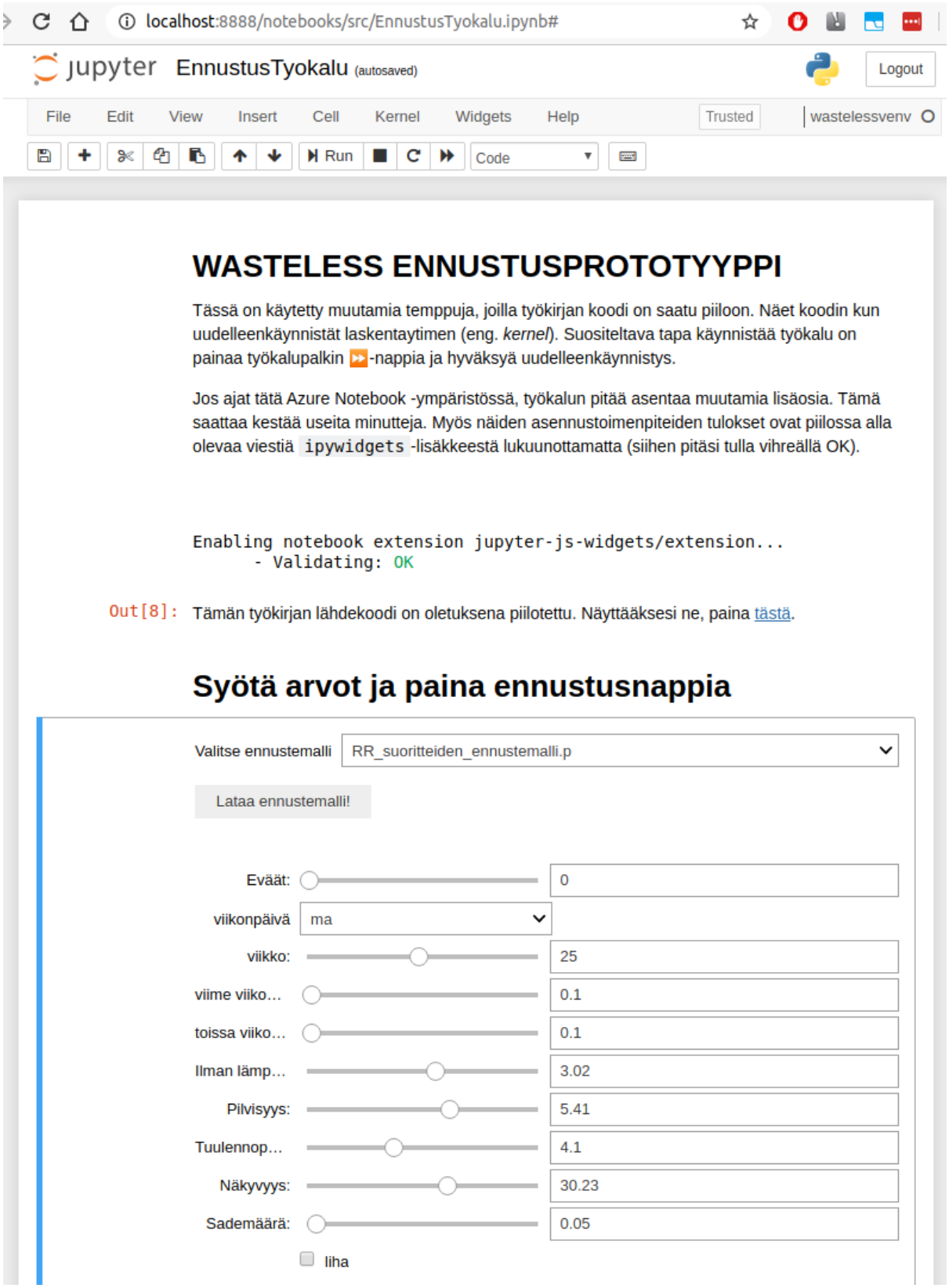
Tämän jälkeen voit käynnistää Jupyter-ympäristön selaimeesi suorittamalla käskyn

```
$ jupyter notebook
```

Työkirjat löytyvät `src`-kansiosta, mistä ne voi avata omille välilehdilleen. Ne lataavat aineistot `data`-kansiosta ja sen alikansioista ja esiprosessoinnin ja mallintamisen jälkeen tallentavat välitulokset `data/prosessoidut` -kansioon. Seuraavan vaiheen työkirjat lataavat näitä välituloksia ja jalostavat niitä edelleen. Lisäksi `src`-kansiossa on muutamia Python-skriptejä, joita käytetään työkirjoissa apuna. Työkirjojen väliset suhteet on dokumentoitu Kuvaan 21.

Yksi työkirjoista, `EnnustusTyokalu.ipynb`, on tarkoitettu ennustemallien käyttämiseen ja niiden kokeilemiseen. Työkalu mahdollistaa levylle `data/mallit`-kansioon tallennettujen mallien lataamisen ja syötearvojen antamisen työkirjan käyttöliittymän avulla. Työkalu käynnistetään painamalla laskentaytimen uudelleenkäynnistävää ja kaikki laskentasolut suorittavaa Jypyter-työkirjan työkalupalkin -painiketta (selaimessa). Pudotusvalikosta valitaan haluttu ennustemalli, jonka jälkeen [Lataa ennustemalli!] -painiketta naksauttamalla työkirja luo käyttöliittymän mallin arvojen syöttämiselle. Arvot syötettyäsi voit painaa [Laske ennuste!] -painiketta, jolloin kyseisen painikkeen alle ilmestyy ennustemallin tuottama ennuste. Suoritteiden osalta tämä on prosenttiosuus lukuvuoden ruokailijoiden kokonaismäärästä ja hävikille hävikki kiloina.

Seuraavalla sivulla on ruudunkaappaus työkalusta. Tämän raportin kirjoitushetkellä selvitämme, miten työkalun saisi toimimaan verkossa ilmaiseksi käytettävissä olevalla Microsoftin Azure Notebooks -palvelulla. Azure Notebooksin tarjoama Python versio oli eri, kuin millä mallit oli laadittu, vuoksi niitä ei suoraan voitu ladata Azure-työkirjaan. Ohjeet Azure-työkirjan käyttöön toimitetaan erikseen sitten kun mallit on saatu tuotettua vanhemmalla Python 3.6 versiolla Azure-käyttöä varten.



Kuva 22: Ruudunkaappaus ennustemallien testaustyökalusta.

Kehitysehdotukset

Tähän lukuun on kerätty erilaisia aineiston esiprosessoinnin ja ennustemallien rakennusyritysten aikana syntyneitä havaintoja, joista on johdettavissa erilaisia suosituksia tai ehdotuksia hankkeen varsinaiseen toteutusvaiheeseen. Keskeisimmät kehitysehdotukset on vielä toistettu raportin johtopäätöksissä.

Aineiston määrän ja piirteiden riittävyys sekä ennustemallit

Alkuperäisenä suunnitelmana esiselvitysvaiheeseen oli rakentaa hyvin toimiva malli, jonka jälkeen piirteitä ja rivejä vähentämällä olisi voinut ekstrapoloida kuinka suuri vaikutus uusilla piirteillä ja lisäpäivillä, eli aineiston kasvattamisella, olisi mallin suorituskykyyn. Koska koneoppimisella laadittuja malleja ei saatu toimimaan juuri vertailukohtaa (satunnaisuus tai vakioennuste) paremmin, ei tätä analyysiä valitettavasti voitu tehdä. Keskeinen suositus onkin kerätä lisää korkealaatuista aineistoa, jotta paremmin toimivia malleja voidaan laatia ja jotta lisäaineiston keräyksen vaikutuksia voidaan arvioida.

Arviomme mukaan erityisesti erilaisten teema-, poikkeus-, juhla- tms. päivien ruokalistat ja ruokailijatiedot tarvittaisiin, jotta suoriteaineiston vaihtelulle löytyisi selityksiä. Luultavasti näillä tiedoin päivittäistä ruokailijamäärää voitaisiin korjata, jolloin nyt piirteytettyjen tekijöiden vaikutukset näkyisivät selvemmin ja olisivat siten opittavissa (muistutuksena, että tällä hetkellä kaavamaisesti käytetään koko lukuvuodelle ilmoitettua ruokailijamäärää jokaisen päivän suhteellista suoritemäärää arvioitaessa).

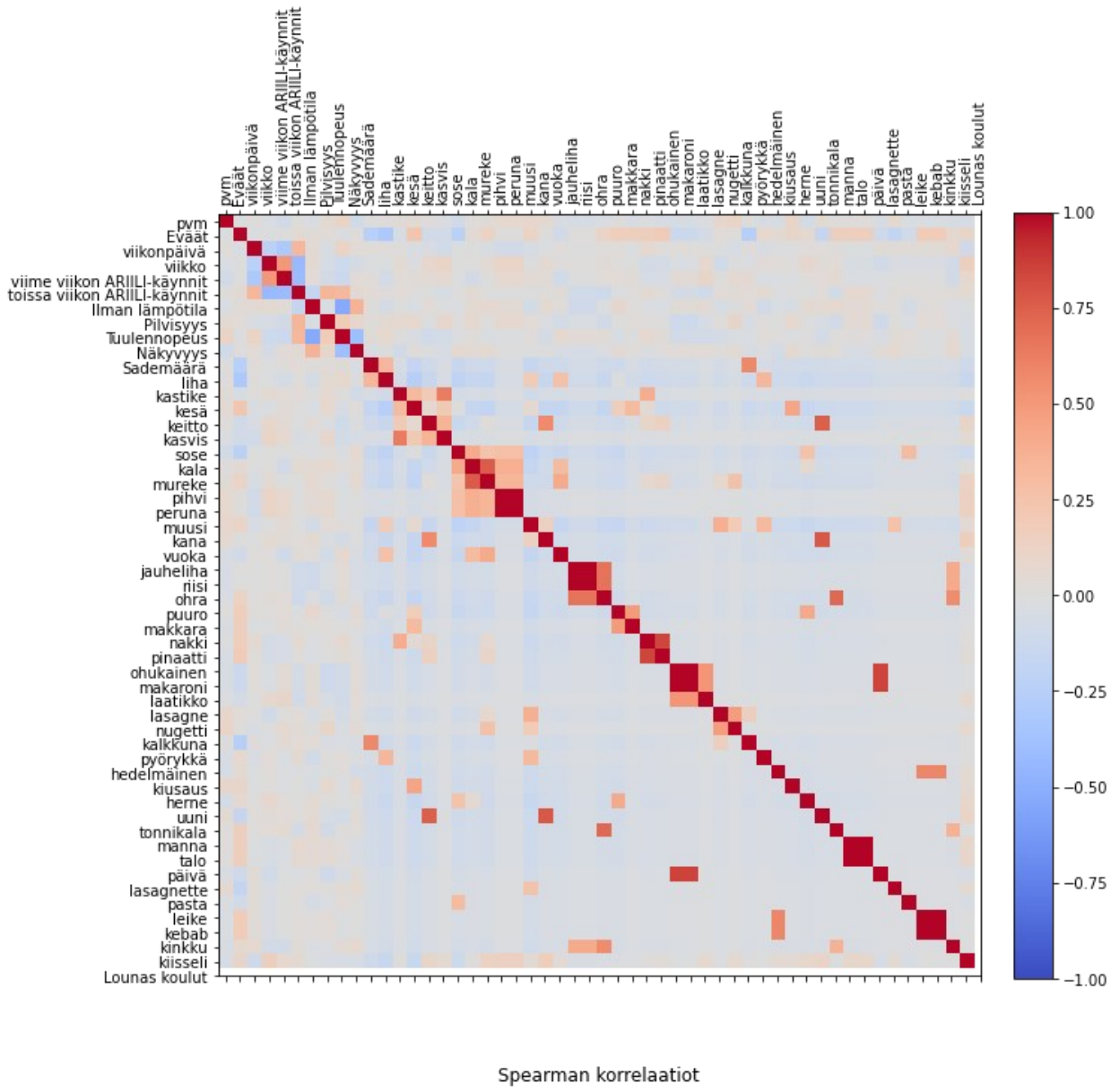
Juhlapäivien osalta alkuperäisenä ajatuksena oli karsia aineistosta juhlapyhiä edeltävät päivät. Tämän kuitenkin arvioitiin vähentävän jo valmiiksi pieneksi arvioidun aineiston rivimäärää, joten sitä ei tehty esiselvitysvaiheen aineistoa prosessoitaessa. Mikäli myöhemmin on käytettävissä lisäaineistoa, suosittelemme, että juhlapyhää edeltävät arkipäivät joko poistetaan aineistosta tai huomioidaan niin, että niiden rivitiedoissa on kyseisinä päivinä käytettyjen erityisruokalistojen todelliset tiedot.

Hävikkitiedon keräämistä kannattaa jatkaa ja keskittyä siinä erityisesti koneluettaviin rivitietoihin. Esimerkiksi säätä koskevat tiedot on luettavissa muualta, mutta koulupäivän ja erityisesti ruokailun erityisjärjestelyihin liittyvä lisätieto, tai edes arviot siitä, ovat arvokkaita arvioidessa todellista kyseisenä päivänä tulossa olleiden ruokailijoiden määrää.

Mikäli aineistoon tulee myöhemmin useita kouluja tai muita ruokaloita, tulee merkityksellisiksi myös vaihtoehtoisen ravinnon saatavuus. Tähän liittyvät piirteet voisivat olla:

- Etäisyys lähimpään kauppaan
- Etäisyys lähimpään kioskiin
- Etäisyys lähimpään ravintolaan
- Kauppoja (kpl) 0.5/1/5 km säteellä
- Kioskeja (kpl) 0.5/1/5 km säteellä
- Ravintoloita (kpl) 0.5/1/5 km säteellä

Nähdäksemme tärkeimmät lisätiedot (piirteet) koskevat kuitenkin koulun kalenteria, sen tapahtumia ja ruokailuun vaikuttavia erityispäivä/-viikkoja (esim. TET-harjoittelut jne.). Nämä tiedot on oltava saatavilla ja ne on huomioitava jo aineiston esiprosessoinnissa ja/tai piirteinä, jotta ennustemallien tarkkuutta voidaan parantaa tämän esiselvitysvaiheen tasosta.



Kuva 23: Piirteiden ja suoritteiden väliset korrelaatiot

Syytä heikkoihin ennustustuloksiin voi etsiä tulkitsemalla piirteiden ja selitysmuuttujan välisiä korrelaatioita. Kuvassa 23 on visualisoitu näitä korrelaatioita ja ne paljastavat epäilyksen, että mikään piirteistä ei kovin vahvasti korreloi suoritemäärien kanssa. Vaikuttaa siis siltä, että käytettyjen piirteiden selitysvoima on liian heikko luotettavan ennustemallin luomiseen. Luultavasti onkin jatkoa ajatellen tärkeämpää keskittyä etsimään uusia suoritemääriin

vaikuttavia muuttujia kuin yrittää saada erilaisia toinen toistaan hienostuneempia malleja toimimaan ilmiön kanssa, joka tässä raportissa käytettyjen piirteiden valossa näyttää olevan hyvin kaoottinen.

Hävikkiaineiston laatu on selkeästi suoriteaineistoa parempi, mutta sielläkin on muutamia mahdollisesti virheellisiä kirjauksia, jotka ovat huomattavasti suurempia kuin kirjausten yleinen taso. Suurin puute hävikkiaineisossa on kuitenkin puuttuvat kirjaukset. Tämä, ja hävikkiaineiston pieni koko rajoittavat sen käyttökelpoisuutta. Täydellisempi ja huomattavasti pidemmältä ajalta koottu vastaavanlainen aineisto luultavasti parantaisi ennustemallien toimintaa ja tarkkuutta.

Ehdotuksia tiedonkeruuseen

Hankkeen tavoitteita ajatellen on tärkeää kerätä lisää aineistoa. On hyvin mahdollista, että myös suoritemäärät voisi ennustaa parempilaatuisesta ja laajemmasta aineistosta. Määrän lisäksi myös aineiston laatuun tulisi kiinnittää lisää huomiota. Poikkeustietojen, kuten juhlapyhien ym. erityispäivien yhteydessä käytettyjä ruokalistoja ei ollut saatavilla, vaikka ne luultavasti merkittävästi vaikuttavatkin suoriteaineistoon. Samoin teemaviikot ja erilaisten tapahtumien ja poikkeusviikkojen aiheuttamat muutokset yleiseen ruokailijamäärään tulisi olla seurannassa.

Esiselvitysvaiheessa käsin tehty aineistojen lataus oli tarkoituksenmukaista, mutta lopullisessa tuotantopalvelussa tulisi kuitenkin hakea a) resepti-, b) sää- ja c) influenssa- ym. epidemiatiedot automaattisesti. Tätä varten tulisi selvittää onko THL:n viikoittainen hengitystieinfektiokäynti aineisto saatavissa koneluettavassa muodossa. Pikaisen tarkastelun perusteella vaikuttaa siltä, että tiedot voisi lukea suoraan rajapinnan kautta JSON-muodossa. Tätä ei kuitenkaan tehty esiselvitysvaiheessa, vaan rajapinnan käytön selvittäminen jäi toteutusvaiheen toimittajan vastuulle: https://yhteistyotilat.fi/wiki08/display/THLKA/Avoin+data+THL%3Assa

Säädatan osalta Ilmatieteenlaitos on hyvä ja luotettava lähde. Sääasemien historia-aineistot ovat helposti saatavilla. Mikäli halutaan ajantasaiset tiedot ja ennusteet, tulee tutustua Ilmatieteenlaitoksen WFS-käyttöön ja siihen liittyviin rajapintoihin. Hyvä lähtökohta vaikuttaisi olevan tämä Python-projekti: https://github.com/saaste/fmi-weather-client/

Ehdotuksia esiprosessointiin

Ruokalistojen analysoinnissa ja esiprosessoinnissa käytettiin alkeellisia luonnollisten kielten tilastollisen käsittelyn menetelmiä. Voikko on monipuolinen suomen kielen tarkistamiseen ja analysointiin tarkoitettu ohjelmisto ja ohjelmointikirjasto. Esiselvitysvaiheen analysoinnissa sitä ja sen Python `libvoikko`-modulia käytettiin palauttamaan sanat perusmuotoonsa (eng. *lemmatization*) ja suomen kielen `FinnSyll`-tavutusmodulia yhdyssanojen jakoon erillisiksi sanoiksi. Nämä tarvittiin, jotta ruokalistoille voitiin tehdä ns. *bag-of-words* -piirteytys.

Esiselvitysvaiheessa huomattiin, että `FinnSyll`-moduli ei jakanut yhdyssanoja kovin luotettavasti, vaan sen tuloksia jouduttiin korjaamaan paljolti käsin. Etenkin siinä

tapauksessa, että tekstimuotoista ruokalistadataa tulee prosessoitavaksi useasta eri lähteestä. Mikäli näin on, toteutusvaiheessa kannattaa tutkia eri suomen kielen prosessointiin tarkoitettujen ohjelmistojen soveltuvuutta. Alle on kerätty muutamia pikaisella tarkastelulla lupaavia vaihtoehtoja:

Stanfordin yliopiston Stanza Python NLP -kirjasto
https://stanfordnlp.github.io/stanza/
Kirjastolle on olemassa myös suomenkielen UD-malli.

Turun yliopiston NLP-jäsennystyökalut
https://turkunlp.org/finnish_nlp.html
Turun yliopistossa on jo vuosien ajan tehty tutkimusta suomen kielen automatisoidun analysoinnin parissa.

Aalto yliopiston FinnPos NLP -kirjasto
https://github.com/mpsilfve/FinnPos
FinnPos on lisenssiltään hyvin salliva avoimen lähdekoodin työkalu.

Testien osalta esiprosessoinnin yhteydessä kuvatut pistotestit olivat riittäviä ja tarkoituksenmukaisia esiselvitysvaiheessa tiedon ja sen käsittelyn eheyden varmistamiseksi. Tuotantojärjestelmää varten suositeltu tapa olisi pilkkoa tiedostojen prosessointi pienempiin aliohjelmiin paloihin ja testata kukin osio/vastuualue (esim. vuosien tunnistaminen, validointi ja viikkolistauksesta päivälistaksi konversio) omiksi yksikkötesteikseen.

Ohjelmiston hankintaan liittyviä keskeisiä suosituksia

Ehdottomasti suurin riski ohjelmistoprojekteille on, että ei tarpeeksi kirkkaasti tiedetä mitä halutaan. Valitettavasti tämän määrittelyä ei voi ulkoistaa, joten vision valmiista tuotteesta **on** oltava tilaajan mielessä kirkas, jotta tavoitteet ja vaatimukset osataan priorisoida. Tämän osion on tarkoitus paitsi auttaa kirkastamaan tavoitetta, myös sitä mitä ohjelmistoprojektin toimittajan valinnassa kannattaa mielestämme huomioida.

Tämän osion sisältö kannattaa huomioida tarkoin tarjouspyyntöä laadittaessa. Yleensä ottaen, mitä tarkempi ja täsmällisempi tarjouspyyntö, sitä helpompi toimittajia on arvioida ja sitä paremmin lopulta saadaan mitä halutaan (kun tiedetään mitä halutaan!). Hankinnan kohde, sen toimittamisen yksityiskohdat, sekä toimittajan valintakriteerit tulee kirjata mahdollisimman yksityiskohtaisesti. Se sijaan halutun ratkaisun teknisiä yksityiskohtia, käytettäviä teknologioita sekä kaikkia toiminnallisia ja ei-toiminnallisia vaatimuksia ei useinkaan ole tarkoituksenmukaista edes yrittää listata etukäteen kun ei ole vielä ymmärrystä siitä, miten ratkaisu kannattaisi rakentaa. Tekniset yksityiskohdat tuleekin yleensä jättää toimittajan valittaviksi, jotta kehitys voidaan tehdä käyttäen toimittajan vahvuuksia. Toki kannattaa varmistua siitä, että toimittaja noudattaa ketterän ohjelmistokehityksen periaatteita ja kriittiset toteutettavat toiminnallisuudet ja niiden ei-toiminnalliset vaatimukset alkavat tarkentua toteutustyön aikana.

Alan konsensus on, että ketterän kehityksen mallin käyttö parantaa merkittävästi ohjelmistoprojektin onnistumisen edellytyksiä. Ketterät toimintatavat mahdollistavat sen, että ohjelmistokoodi ja tuote on kokoajan tilaajan nähtävillä, tarkasteltavissa ja kokeiltavissa. Tämä ennaltaehkäisee sen molempien osapuolten kannalta kiusallisen tilanteen syntymistä, missä kehityksen kanssa ajaudutaan umpikujaan, tietoa aletaan pimittää ja vastuuta vältellä. Tähän riskiin liittyen tilaajan tuleekin toimeksiantosopimuksessa **aina** varata oikeus käyttää koodia (vaikka sitten keskeneräistäkin sellaista) ja tuotetta haluamallaan tavalla.

Erona perinteiseen ja vanhentuneeseen vesiputousmalliin ja valmiiden kokonaisratkaisujen hankintaan, ketterää kehitystä ostetaan tyypillisesti tuntityönä. Toki hankkeella voi olla budjetti, mutta tärkeämpää on luottamus tilaajan ja toimittajan välillä ja ymmärrys siitä mitä missäkin toteutushankkeen vaiheessa on olennaisinta tehdä.

Hankkeessa ratkaistavan ongelman määritteleminen

On tärkeää määritellä ne spesifiset ongelmat, mitä toteutettavalla palvelulla ollaan lopulta ratkaisemassa. Esiselvitysvaiheen käynnistystapaamisessa ongelmat kirjattiin seuraavasti:

> *Ruokaloissa syntyy paljon turhaa ruokahävikkiä.*
>
> *Näin käy siksi, että sopivan valmistettavan annosmäärän arviointi on vaikeaa. Tähän vaikuttaa suuri määrä erilaisia tekijöitä, mutta vain osa tästä tiedosta on keittiö-henkilökunnalla.*
>
> *Ongelmaa pahentaa se, että tällä hetkellä ei ole helppoa tapaa kerätä kaikkea*

osaamista ja annosten määrään vaikuttavaa tietoa yhteen. Lisäksi, asiaan vaikuttavia tekijöitä on useita ja niiden riippuvuudet ovat haastavia mallintaa. Tämä tekee menekin ennustamisesta usean päivän päähän vaikeaa.

Näitä tulee edelleen tarkentaa ja työstää palvelun toimittajan ja etenkin loppukäyttäjien kanssa. Hyvä työkalu tähän on erilaiset työpajat, joita voi rakentaa esimerkiksi käyttäjätarinoiden laatimisen ympärille (näistä lisää alempana).

Ongelmaan liittyy myös vahvasti myös hankkeen tavoite. Onko tavoitteena saada hävikki minimiin vai tasapainoilla muiden tavoitteiden välillä - kuten esimerkiksi ruokailun taloudellisuuden ja palvelutason kanssa? Optimoinnissa olennaista on se, että rajoitteet ja tavoite on hyvin valittuja ja vastaavat todellista tahtotilaa. Esimerkiksi jos tavoite on minimoida euroja, mutta halutaan myös säilyttää palvelutaso, mikä on hinta sille, että oppilas jää ilman kalapuikkoja?

Ohjelmiston tuottama lisäarvo

Ohjelmiston arvoon eri sidosryhmille voidaan pureutua ongelman määrittelyn ja sen ratkaisun kautta. Arvonmuodostusta voidaan kuitenkin kuvata myös esimeriksi hissipuheen avulla. Alle on luonnosteltu esitarkastusvaiheen raportin laatimisen aikana jalostuneen ymmärryksen mukainen hissipuhe:

> *Kouluruokaloissa ja niiden keittiöissä syntyy paljon turhaa hävikkiä, sillä ruokailijoiden määriä ja ruuan menekkiä on vaikea arvioida. Keittäjien arviointiosaaminen on hankittava kalliisti kokemuksen kautta ja sitä on vaikea monistaa. Tarjoamme ratkaisuksi työkalua, joka auttaa keittiöitä arvioimaan valmistettavien annosten määrän. Palvelu kerää, esittää ja hyödyntää saatavilla olevaa tietoa poissaoloista, retkistä, kausi-influenssatilanteesta, ja jopa säästä. Tietoa voidaan käyttää apuna paitsi ruokailijamäärää ennustettaessa, myös annosten menekkiä arvioitaessa. Järjestelmä myös kerää hävikkitietoa, jonka ansiosta ennusteet muuttuvat sitä paremmiksi, mitä enemmän palvelua käytetään. Samalla hävikkitieto auttaa kunnallisia päätöksentekijöitä arvioimaan hävikistä syntyneitä kustannuksia, reseptien käyttöä ja eri ohjaustapojen vaikuttavuutta.*

Hissipuheessa mainitut lisäarvoa tuottavat ohjelmiston ominaisuudet tulle listata ja jaotella riskin ja tarpeellisuuden mukaan. Loppukäyttäjät haluavat kysyttäessä toki kaiken ja enemmän, mutta hankkeen onnistumisen kannalta tärkeintä on selvittää mitä he *oikeasti* tarvitsevat. Ohjelmistoprojektin tavoitteena onkin hissipuheessa kuvatun arvolisän antava MVP eli minimivaivalla tehty tuote (eng. *minimum viable product*). Tähän on tarkoitus toteuttaa vain ja ainoastaan ne keskeiset ominaisuudet jotka tekevät tuotteesta/palvelusta tarpeellisen. Pienen lisäarvon ominaisuuksia siis ei toteuteta, vaan ne siirretään tuleviksi kehityskohteiksi.

Suuren arvon toiminnallisuuslista otetaan tarjouspyynnön pohjaksi ja niiden perusteella pitäisi olla mahdollista sopia toimituksen kohteen ehdottomista vaatimuksista. Muut ominaisuudet ja ei-toiminnalliset vaatimukset voidaan kirjata valinnaisiin vaatimuksiin. Tässä

tulee kuitenkin muistaa, että mitä vähemmän on vaatimuksia, sitä todennäköisemmin saa mitä haluaa ja sitä helpompi hankinnassa on onnistua.

Kuvassa 24 on hahmoteltu ohjelmiston keskeisten käyttäjätarinoiden tunnistettuja sisältöjä. Kuvan mukaisesti palvelulle on tunnistettu kahdenlaisia käyttäjiä:

- Ruokapalvelupäällikkö käyttää palvelua arvioidakseen kyseisen ja tulevien päivien ja viikkojen ruoka-aineiden menekkiä ja ruokailijoiden määrää.
- Ylläpitäjä/hallintokeskuksen virkamies puolestaan on kiinnostuneempi järjestelmään tehdyistä kirjauksista.
- Myös järjestelmä itse tekee itsenäisesti hakuja ja päivittää tietojaan.

Tämän raportin tulosten perusteella ennustusmallit eivät käytettävissä olevalla aineistolla toimi vielä riittävän hyvin, jotta niistä olisi ruokapalvelupäällikölle hyötyä. Valitettavasti epäselvää on myöskin se, missä määrin aineiston puutteet ovat määrällisiä tai laadullisia. Siksi mielestämme toiminnallisuuden painopiste tuleekin ensin siirtää päätösten kannalta relevantin tiedon esittämiseen ja lisäaineiston keräämiseen, sillä ennustustoiminnallisuuden kehittämiseksi tiedonkeruu on tärkeää.

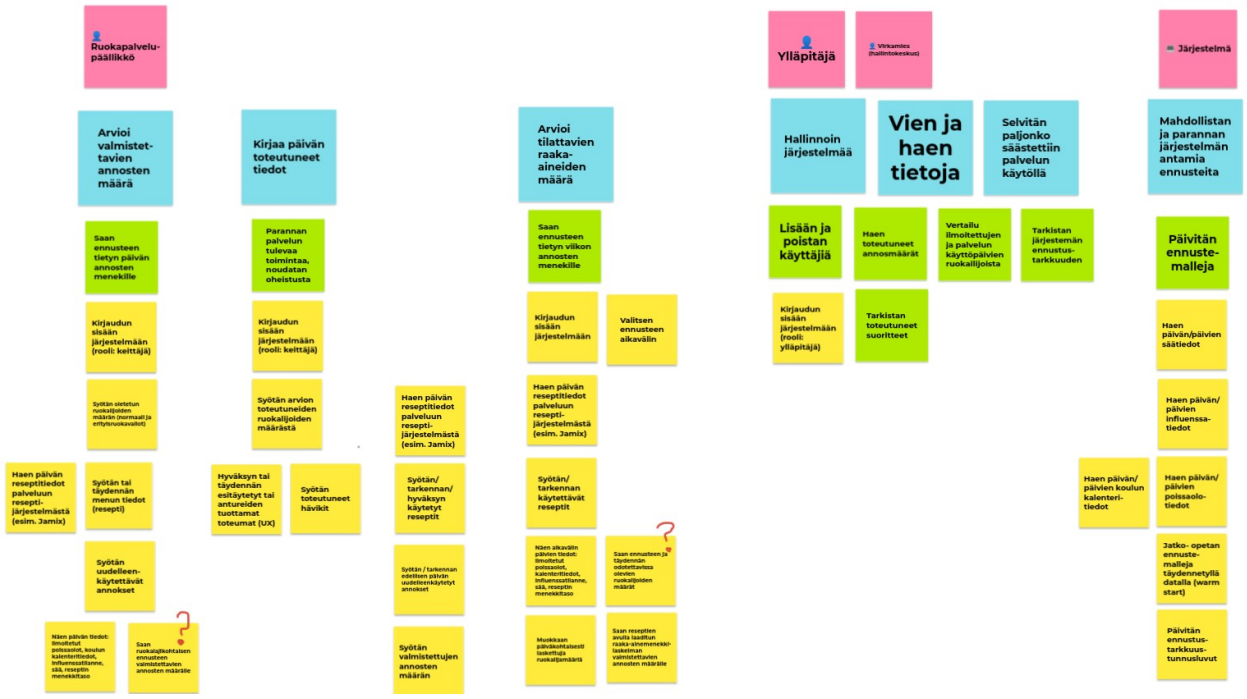
Keskeisin palvelun toiminnallisuus on ruokailijamäärään ja ruokalajien menekkiin liittyvän tiedon tarjoaminen ruokapalvelupäällikölle. Tämän toteuttaakseen järjestelmä kerää tietoa mm. säästä, alueen hengitystieinfektiotilanteesta, ilmoitetuista poissaoloista ja suunnitelluista koulun tapahtumista. Näiden lisäksi on tärkeää, että järjestelmään voidaan syöttää toteumatietoja: suoritteet, ruokailijoiden määrä, eri ruokalajien menekki ja hävikit. Näitä tietoja käytetään paitsi hallinnossa myös palvelun ennustustoiminnon kehittämiseen. Keskeistä on, että palvelu antaa kokoajan selkeää lisäarvoa käyttäjälleen. Ja siksi tiedon esittämistä priorisoidaan. Alle on listamuotoon kerätty keskeisimmät toiminnot:

- Mahdollisuus nähdä tulevan viikon ruokailijamäärään vaikuttavat tiedot, joiden avulla ruokapalvelupäällikkö tai vastaava palvelun käyttäjä voi tilata raaka-aineita ja arvioida valmistettavien annosten määrää. Tiedot tulisi olla ja esittää ainakin:
    - Ruokailijamäärään vaikuttavista tapahtumista koulun kalenterissa, lukujärjestyksissä ja etukäteen tiedossa olevista oppilaspoissaoloista.
    - Sääennusteista ja edellisten viikkojen hengitystieinfektiotilanteesta alueella.
    - Samankaltaisimpien aiempien päivien hävikki- ja suoritemäärätiedoista. Tässä voisi käyttää koneoppimismenetelmiä ja tarkemmin ns. suosittelujärjestelmiä, jotka pystyvät näyttämään tietokannasta samankaltaisimmat kirjaukset.
- Helppo tapa kirjata toteutuneet suoritemäärät, hävikit ja erityisesti arviot niihin johtaneista seikoista. Tässä voisi käyttää hyväksi ennustemallia, joka esitäyttää tiedot tilauksen, valmistetun annosmäärän, reseptiikka- ym. saatavilla olevan tiedon mukaisesti. Käyttäjän vastuulle jäisi täydentää ja korjata tiedot. Malli voisi jopa värein ilmaista milloin ennuste on luotettava ja milloin ei. Ennusteet tarkentuvat palvelua

käytettäessä sitten kun taustajärjestelmä on ajastettuina tehtävinä päivittänyt käytettäviä malleja.

Tärkeitä toimintoja palvelun lisäarvon kannalta ovat myös ne, jotka liittyvät kunnan hallinnon seuranta- ja raportointikäyttöön. Tämän arvioimiseksi suosittelemme, että myös kyseiset käyttäjät otetaan mukaan toteutusvaiheen toiminnallisuuden suunnittelutyöhön, sillä siellä tarvittavien toimintojen listaaminen käytettävissä olevin tiedoin ei ollut mahdollista.

Käyttäjienhallinta ja vastaavat toiminnot eivät nähdäksemme ole täysin välttämättömiä, mutta voivat toteutustavasta riippuen olla tarpeen esimerkiksi kirjausten jäljitettävyyden varmistamiseksi tai ilkivallan torjumiseksi. Muutenkin erilaiset käyttäjätarinakartat ja niiden laatimiseen tähtäävät työpajat ovat suositeltavia keskeisen toiminnallisuuden tunnistamiseksi. Näihin tulisi saada ideoimaan kaikkien ohjelmiston kehittämiseen ja sen käyttämiseen liittyvien sidosryhmien edustajat.



Kuva 24: Keskeisten käyttäjätarinoiden luonnostelua

Huomaa, että toimittajan tulisi aina aloittaa ison arvon / ison riskin ominaisuuksista. Mikäli aloitetaan ison arvon, mutta pienen riskin ominaisuuksista, riskit kasvavat, sillä vaikeiden tehtävien vaikeus saattaa yllättää vasta myöhemmin. Olennaista tässä on tunnistaa varhain, että mitkä ison arvon / ison riskin ominaisuudet ovat välttämättömiä ohjelmiston tuottaman olennaisen lisäarvon kannalta ja keskittyä niihin. Järkevintä on paljastaa näiden riskien vaikutukset niin pian kuin mahdollista.

Ohjelmiston laatu

Ohjelmiston laatu on tärkeä aihe. Se ei vain tarkoita suhteellista virheettömyyttä, vaan myös sitä, että ohjelmisto on oikea ja sopiva ratkaisu itse ongelmaan. Jotta valmistettavan

ohjelmiston laatu voidaan taata, on kehitys tehtävä järjestelmällisesti ja hyvää ohjelmistokehitystapaa noudattaen. Alla on listattu joitain työkaluja ja suosituksia, joiden avulla ohjelmiston laatu-ulottuvuuteen voidaan kiinnittää huomiota.

- Käyttäjätarinoiden (eng. *user story*) ja käyttötapausten (eng. *use case*) kartoitus tilaajan ja loppukäyttäjien kanssa, niiden huolellinen kirjaus, dokumentointi ja jalostaminen toiminnallisuuksien hyväksyntätesteiksi.

- Automatisoidut testit (yksikkö- ja integraatiotestaus) ja ajantasaiset ohjeet siihen miten ne ajetaan.

- Tarkoituksenmukaisesti dokumentoidut aliohjelmat ja luokat ja

    - hyvä ja johdonmukainen nimeämiskäytäntö, sekä

    - aliohjelmien dokumentointi silloin kun nimet eivät kerro riittävästi käyttötarkoituksesta ja reunaehdoista.

- Käyttöönoton dokumentaatio sis. riippuvuudet ja ympäristön asennustoimenpiteet ja käskyt. Mikäli kyseessä on verkkopalvelu, ympäristön pystytysohje/-skripti, jolla palvelun voi ottaa käyttöön esim. hankkeen aikana valitussa pilvipalvelussa.

- Käytön dokumentaatio (erit. käytön ensimmäiset askeleet esimerkein).

- Rajapintojen dokumentaatio tarkoituksenmukaisella tarkkuudella.

- Tulevaisuuden kehitystarpeiden keräys, listaus ja kuvaus
  (mitä pitäisi tehdä, miksi ja minne).

- Versionhallinta ja hyvät, lyhyet ja ytimekkäät kuvaukset koodiin tehdyistä muutoksista.

Ohjelmiston laadun kannalta on myös tärkeää tunnistaa ohjelmistoprojektin rajoitteet, jotka voivat koskea aikaa, rahaa, tai sisältöjä (yleensä kaikkia näitä). On päätettävä mitkä näistä voivat joustaa ja kuinka paljon. Rajoitteisiin törmäämiseen tulee varautua ja on tiedettävä mistä silloin tingitään. Kun aika joka tapauksessa toteutusvaiheen loppupuolella uhkaa loppua, tulee tilaajan varautua jo heti alusta alkaen, että matalan prioriteetin toiminnallisuudet jäävät toteuttamatta. Tämän vuoksi kaikkien projektin osapuolten tulisi osallistua keskusteluun toiminnallisuuksien lisäarvoista, tarpeellisuudesta ja toteutettavuudesta, jotta yhdessä voidaan tehdä mielekkäitä valintoja toteutushankkeen rajoitteet huomioiden.

Ohjelmistoprojektia hankittaessa tulee kiinnittää huomiota myös käytettävyyteen. Varmistu, että toimittajalla on osaamista käyttäjäkokemuksen ymmärtämiseen ja suunnitteluun. Tämä tulee myös mainita hankintapyynnössä esim. niin, että osana toimitusta saadaan myös graafista osaamista ja käyttäjäkokemuksen suunnittelutyötä.

Ketterä kehitys

Nykytietämyksen mukaan uusia tuotteita ei kannata tehdä muulla tavalla kuin ketterästi. Samaa suosittelemme myös tässä hankkeessa. Ketterissä malleissa keskitytään ihmisiin, lisäarvoon ja tuotteeseen, ei prosessiin. Niissä tunnustetaan, että kaikkea ei tiedetä etukäteen, vaan suunnitelmaa tarkennetaan yhdessä toimittajan kanssa projektin edetessä. Tyypillinen jakso on 2 viikkoa ja jokaisen jakson jälkeen esitellään mahdollisimman valmiita ja kokeiltavissa olevia tuloksia. Tilaajan kannalta haasteena on tietty toteutusvaiheen ennakoimattomuus, mutta vastapainona tilanteen seurattavuus ja ohjattavuus paranee.

Ketterissä toimintatavoissa tilaaja ja toimittaja tekevät tuotteen yhdessä. Siksi hyvä, läheinen ja luottamuksellinen suhde on ensiarvoisen tärkeää. Toteutustekniset asiat ovat toteutustiimin huoli ja murhe, mutta tilaajalla on paras tietämys tehtävien ja toiminnallisuuksien tärkeydestä ja toteutusjärjestyksestä.

Ketterille ohjelmistoprosesseille tyypillistä on, että kirjoitetaan käyttäjätarinat yhdessä käyttäjien kanssa, priorisoidaan ne ja valitaan niistä osa toteutettavaksi seuraavassa jaksossa (sprint). Tarinoiden pitää olla siksi riittävän pieniä, selkeitä ja toisistaan riippumattomia. Sprintin jälkeen todetaan mitä opittiin, täydennetään tarinoita ja säädetään niiden prioriteetteja, sekä valitaan seuraavat toteutettavat käyttäjätarinat. Prototyyppi on kokoajan toimivassa tilassa, jolloin se on tilaajan ja ihannetapauksessa myös loppukäyttäjien testattavissa ja arvioitavissa. Näin toimittaessa projektisuunnitelmaa tarkastellaan, muokataan ja arvioidaan jatkuvasti sitä mukaa kun oppia ja ymmärrystä ongelmasta, lisäarvosta, keskeisistä toiminnallisuuksista ja niiden toteuttamiseen liittyvistä riskeistä kertyy.

Ketterä kehitys ei tarkoita suuripiirteisyyttä. Tilaajan kannattaa varmistua, että tuotettu koodi on testattua (mielellään automaattisin testein), hyvälaatuista, luettavaa ja ylläpidettävää. Hyvät ohjelmistokehitystavat, kuten harkittu ja tarkoituksenmukainen arkkitehtuuri ja teknologiat, versionhallinta, riippuvuuksien kirjaus sekä käyttöönoton automatisointi ja dokumentointi pitää olla kunnossa ja näiden tulee kehittyä positiiviseen suuntaan ohjelmistoprojektin aikana.

Yksi projektin tilaajan kannalta olennaisin rooli ketterässä kehityksessä on tuoteomistaja (eng. *product owner, PO*). Tuoteomistaja päättää siitä mitä tehtäviä ja toiminnallisuuksia ohjelmistokehittäjätiimi seuraavaksi tekee, mitkä ovat heille syötteenä annetut tehtävät (eli yleensä käyttäjätarinat), ja että ne ovat oikein valittuja, hyvälaatuisia ja oikein priorisoituja. Yksi tärkeimmistä tuoteomistajan tehtävistä on kuitenkin päättää mitä **ei** tehdä, sillä ideoita on lähes aina enemmän kuin on aikaa. Jotkut tarinat ovat arvokkaita, toiset ovat vähemmän tärkeitä, vaikeita myydä tai liian työläitä toteuttaa. Keskusteluissa toimittajan kanssa tuoteomistaja saa lisää tietoa em. seikoista ja tähän tulee varata riittävästi aikaa kalenterista.

Sisällöt

Toimitettavat sisällöt ja niiden käyttöoikeudet on syytä kirjata ja hankintapyyntöön. Ymmärtääksemme tässä hankkeessa syntyvä ohjelmistokoodi tulee kokonaisuudessaan luovuttaa tilaajan käyttöön avoimella lisenssillä kuten esim. MIT-lisenssillä (Open Souce Initiative 2020). Luontevin tapa on käyttää versionhallintajärjestelmää, johon tilaajalla on

jatkuva pääsy, käyttö- ja kopiointioikeus (eli mahdollisuus haarauttaa kehitys, eng. *fork*). Kyseessä olevan hankkeen ominaispiirteet huomioiden on toivottavaa, että toimittaja käyttää julkista versionhallintajärjestelmää jo kehitystyön alkaessa, jolloin kuka tahansa voi seurata työn edistymistä ja tarvittaessa osallistua siihen. Tämä pienentää merkittävästi ohjelmistoprojektin riskejä, sillä toimittajat, etenkin hyvien ohjelmistokehityskäytänteiden mukaan toimiessaan, ovat jossain määrin vaihdettavissa. Tilaajan tulee viimeiseen asti välttää tilannetta, jossa syntyy ns. toimittajalukko (eng. *vendor lock-in*), eli tilanne jossa toimittajan vaihtaminen ei ole mahdollista sillä sisällöt ovat ainoastaan toimittajan hallussa ja hallinnassa. Näin tulisi tehdyksi myös toimittajasta irtautumiseen liittyvien riskien arviointi ja hallinta.

Sisältöjen osalta kannattaa määritellä, että kehitys tehdään avoimien ja vapaasti saatavilla olevien rajapintojen ja palveluiden päälle. Tämä sisältää myös avoimen datan käytön. Perustelut ovat samat kuten edellä: tämä lisää liikkumavaraa ja pienentää riskejä.

Avoimena lähdekoodina toimitetun ohjelmiston jatkokehitys- ja ylläpitomahdollisuudet ovat merkittävästi laajemmat kuin suljettujen ohjelmistojen. Myös teknologiavalinnat heijastuvat tähän ohjelmistojen ulottuvuuteen. On tärkeää, että valittu teknologia on vakiintunut ja yleisesti käytössä. Samoin on tärkeää, että vapaasti saatavilla olevia valmiskomponentteja käytetään aina kun se niiden laadun ja lisensoinnin puolesta on mahdollista. Kiusaus keksiä ja kirjoittaa pyörää uudelleen on ohjelmistoalalla vahvasti läsnä, ja tähän kannattaa kiinnittää huomiota jo hankintavaiheessa ja toimittajia arvioitaessa.

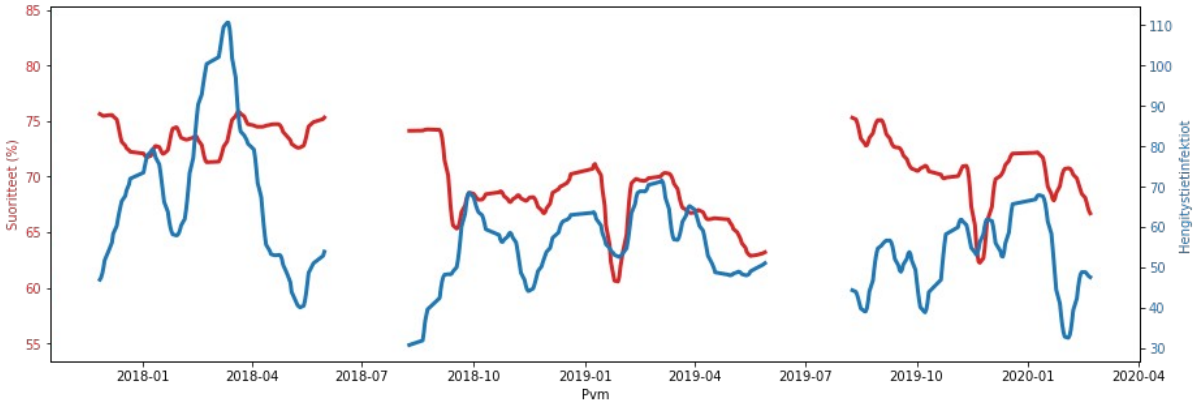
Lopuksi, sisältöjä ajatellen tulee tunnistaa, että kaikkea koodia täytyy jatkokehittää ja ylläpitää mikäli se otetaan käyttöön. Modulaarisuus, kestävä arkkitehtuuri ja kattavat automaattiset testit ovat tässä erittäin tärkeässä asemassa. Niiden ansiosta jatkokehitys ja ylläpito ovat mahdollisia myös kolmannelle osapuolelle.

Toimittajan valintatilanne

Toimittajan valintaan tulisi suhtautua kuten työhaastatteluun. Tärkeää on, että tilaaja vakuuttuu toimittajan osaamisesta ja siitä, että hän ymmärtää tilaajan tarpeita. Pelkän halvimman hinnan mukaan ei voi suositella tekemään toimittajan valintaa, vaan kannattaa arvioida ainakin erityisosaamisen sopivuutta (kuten käytettävyysosaamista tai data-analytiikkaosaamista), toimittajan käyttämiä ja mainitsemia ohjelmistokehityksen hyviä käytänteitä (versionhallinta, automaattinen testaus, käyttöönoton automatisointi). Mikäli mahdollista, kannattaa käyttää ulkopuolista ja riippumatonta asiantuntijaa edellä mainittujen seikkojen arviointiin. Viime kädessä luottamus toimittajan edustajiin ja heidän ymmärrykseensä ja osaamiseensa on kuitenkin kaikkein tärkein valintaan vaikuttava tekijä. Siksi on todella tärkeää varmistua siitä, että juuri ne toimittajan edustajat, joiden kanssa olet asioinut ovat myös osallisia itse projektissa.

Yhteenveto ja johtopäätökset

Eräs esiselvitysvaiheen keskeisistä tavoitteista oli selvittää, voiko ruokailijamäärää (suoritemäärää) ennustaa kalenterin, ruokalistan, sään ja hengitystieinfektiotilanteen perusteella. Koneoppimista hyödyntävän lähestymistavan todentamiseksi esiselvitys-vaiheessa yritettiin mallintaa esiprosessoitua ja siivottua aineistoa lukuisilla erilaisilla koneoppivilla algoritmeilla. Valitettavasti testatut koneoppimiseen perustuvat ennustemallit eivät näytä oppivan mallintamaan ilmiötä riittävällä tarkkuudella, jotta mallit olisivat sellaisenaan käyttökelpoisia. Siksi onkin tärkeää tässä yhteenvedossa analysoida ja arvioida mahdollisia syitä tulosten taustalla.



Kuva 25: Vertailussa hengitystieinfektiokäynnit ja lounassuoritemäärät (josta 5 % piememmistä ja 2,5 % suurimmista suoritekirjauksta poistettu ja jossa päivittäistä vaihtelua on keskiarvoistettu).

Mallintamisen haasteita havainnollistaa yllä oleva esimerkki hengitystieinfektiokäyntien ja lounassuoritekirjausten yhteydestä. Tässä raportissa on todettu, että niiden korrelaatio on heikosti positiivinen, mikä on intuitionvastaista – voisi olettaa että mikäli influenssatilanne on Lapualla paha, myös yläkoululla olisi vähemmän ruokailijoita. Keskiarvostamalla suorite- ja hengitystieinfektiokäyntikäyriä voimme yrittää kuitenkin selvittää syitä tähän (Kuva 25). Huomaamme, että muut lounassuoritemäärään vaikuttavat ilmiöt, kuten suoritemäärien tasainen jatkuva pieneneminen lukuvuoden aikana lukuvuosina 2018-2019 ja 2019-2020, influenssa-aktiivisuuden ja suoritetason lukuvuosittainen vaihtelu ja poikkeukselliset viikot (tammi-helmikuussa ja marraskuun loppupuolella 2019) vaikuttavat suoritemääriin suuremmin kuin nyt mallinnetut tiedot. Sen vuoksi, etenkin aineiston ollessa pieni, tulisi tietää ja korjata ennustetut suoritemäärät poikkeustilanteiden aiheuttamilla muutoksilla ruokailijamäärään. Poikkeustilanteiden vaikutukset toimisivat mallin syötteinä tai vaihtoehtoisesti ne kaavamaisesti vähennettäisiin päivän ruokailijamäärästä (joka sitten olisi syöte mallille). Ilman tietoa poissaoloista, erityispäivistä, koeviikoista, TET-jaksoista jne. on todella vaikea arvioida tai mallintaa ruokalistan, sään tai hengitystieinfektioiden vaikutusta suoritemääriin. Olisi myös tärkeää tietää teema-, poikkeus-, juhla- tms. päivien ruokalistat, jotta suoritemäärän tarkempi ennustaminen olisi mahdollista.

Yhteenvetona suoriteaineistosta voidaan sanoa, että yksi sen suurimmista haasteista on, että äärikirjausten (eng. *outliers*) syytä ei ole merkitty itse aineistoon eikä sitä voida jälkikäteen päätellä annetusta tai vapaasti saatavilla olevista tiedoista. Tämä teki paitsi aineiston esiprosessoinnista, myös aineiston vaihtelun syy-yhteyksien päättelystä hyvin vaikeaa. Johtopäätöksemme suoriteaineistosta onkin, että käyttökelpoisen suoriteaineiston suhteellisen pienestä määrästä, sen poikkeamista, ja niitä selittävien muuttujien puutteesta johtuen koneoppivat algoritmit eivät löydä hyvää piirteiden (selittävät muuttujat) ja selitettävän muuttujan (lounassuoritteet) välistä yhteyttä kuvaavaa mallia. Selittäviä muuttujia tulee siis hankkia lisää.

Hankkeessa kerätty hävikkiaineisto puolestaan on äärimmäisen rikas ja monipuolinen. Siinäkin tosin on pieniä laadullisia puutteita, mutta on todennäköistä, että mikäli kyseisiä tietoja olisi saatavilla n. kymmenkertainen määrä, sen avulla voitaisiin laatia jo kohtalaisen hyviä ennustemalleja. Silti, myös sen osalta kirjauskäytänteitä ja ohjeistusta tulee kehittää. Sään kirjaamisesta voi luopua, mutta ruokailijamäärät tulee kirjata yksikäsitteisesti, sekä valmistetut, hyödynnetyt ja linjastohävikkiin menneet erityisruokavalioruuat ilmoittaa. Tärkeää on myös täyttää kaikki tiedot jokainen päivä, sillä puuttuvat tiedot heikentävät aineiston käyttökelpoisuutta, ja valitusta putsausmenetelmästä riippuen voivat pienentää merkittävästi myös sen laajuutta.

Tällä hetkellä hävikkien ennustusarkuutta heikentää eniten käytettävissä olevan aineiston vähäinen määrä ja mahdolliset virhekirjaukset. Lisäksi se, että osa puuttuvista arvoista jouduttiin imputoimaan lähinaapurimenetelmällä hävikkiaineistoa esiprosessoitaessa, saattaa tehdä ennusteista osin epäluotettavia. Hävikkiaineistoon kirjatun ruokailijamäärän käyttäminen mallin syötteenä ei ollut alkuperäisen suunnitelman mukainen, mutta kuten aiemmin todettiin, suoritemääräennustusta ei saatu toimimaan luotettavasti, eikä ruokailijamäärää siten voitu testeissä korvata ennusteella. Tuloksia tulkitessa onkin tärkeää huomioida, että todellista ruokailijamäärää ei välttämättä oikeasti tunneta, mikä vaikuttaa todelliseen ennustustarkkuuteen.

Ruokailijakohtaista hävikkiä ennustavat mallit osaltaan pyrkivät paikkaamaan tätä puutetta. Toinen hyödyllinen, vaikkakin toistaiseksi hävikkiaineistosta puuttuva tieto, olisivat päivittäiset poissaolot. Mikäli niiden määrä tiedettäisiin, poissaolojen ennustaminen kalenteri-, sää- ja hengitystieinfektiotiedoilla voisi olla syöte varsinaiselle ruokailijamäärän ennustamiselle. Näin yhden ennustemallin ei tarvitsisi löytää niin paljon piirteiden välisiä riippuvuuksia. Tämä mahdollistaisi myös ennustamistehtävän pilkkomiseen useisiin alitehtäviin, mikä tekisi malleista helpompia rakentaa pienemmällä aineistomäärällä. Samalla lähestymistapa tekisi mallien suorituskyvyn mittaamisesta helpompaa. Ennustemallien osalta keskeinen suositus onkin kerätä lisää korkealaatuista aineistoa, sisältäen myös poissaolotiedot, jotta paremmin toimivia malleja voidaan laatia ja jotta lisäaineiston keräyksen vaikutuksia voidaan arvioida.

Aineistosta ja opetettujen mallien heikosta ennustustarkkuudesta johtuen tätä hanketta ei mielestämme voi rakentaa toimivien ennustemallien varaan. On lopulta pelkkää toiveikasta arvelua, että suurempi aineisto johtaisi toimivaan ja käyttökelpoiseen malliin. Täten tiedonkeruun osalta tulee ensisijaisesti varmistua siitä, että kerättävä aineisto palvelee myös muita hankkeen tavoitteita, kuten tarkempaa ja helpompaa hävikin kustannusten seurantaa

ja reseptiikan kehittämistä. Edelliseen liittyen: paras arvaus keskeisimmälle syylle siihen, miksi ennusteita on tällä hetkellä vaikea tehdä keittiöillä, on, että tieto siitä on hajallaan eri järjestelmissä ja palveluissa. Esiselvitysvaiheen tarkoitus olikin myös tutkia eri aineistolähteiden käyttökelpoisuutta. Tuloksena voidaan todeta, että THL:n ja Ilmatieteenlaitoksen aineistot ovat molemmat hyödynnettävissä erilaisissa ruokahävikkiä koskevissa ennustetehtävissä. Edellä mainituista seikoista johtuen toiminnallisuuden painopiste tuleekin mielestämme ensin siirtää päätösten kannalta relevantin tiedon esittämiseen ja lisäaineiston keräämiseen, sillä ennustustoiminnallisuuden kehittämiseksi tiedonkeruu on edelleen tärkeää. Toteutusvaiheessa tehtävään työkaluun voidaan myöhemmin lisätä ennustavia ominaisuuksia sitten kun korkealaatuista aineistoa on enemmän saatavilla ja on saatu todettua, että sen perusteella voidaan opettaa käyttökelpoisia malleja.

Keskeisin suositus koskien toteutusvaiheen hankintapyynnön valmistelua on selvittää mahdollisimman tarkasti mitä ollaan hankkimassa, ja vasta sitten hankkia ohjelmiston kehitys käyttäen ketterän kehityksen mallia. Raportin sisältämä ongelman kuvaus ja hissipuhe toivottavasti auttavat tässä. Mitä tarkempi ja täsmällisempi tarjouspyyntö pystytään laatimaan, sitä helpompi toimittajia on arvioida. Varsinaisessa kilpailutuksessa tulee varmistua toimittajan riittävästä osaamisesta ja solveltuvuudesta hankkeen toteuttajaksi. Toimittajan ammattitaitoisuus, ymmärrys toteutettavasta kokonaisuudesta ja etenkin luottamus toimittajan ja tilaajan välillä ovat tärkeimpiä mittareita soveltuvuutta arvioitaessa. Avoimen lähdekoodin lähestymistavan, käyttäjäkokemuksen ja graafisen suunnittelun tärkeyttä ei pidä myöskään väheksyä toteutusvaiheen riskejä pienentävinä tekijöinä.

Lopuksi, osana hankintapyynnön valmistelua tulisi käydä keskustelua loppukäyttäjien kanssa palvelun keskeisten lisäarvoa tuottavien toiminnallisuuksien tunnistamiseksi. Näihin tapaamisiin tulisi saada ideoimaan kaikkien ohjelmiston kehittämiseen ja sen käyttämiseen liittyvien sidosryhmien edustajat. Tämä lisäisi nähdäksemme merkittävästi ymmärrystä suuren lisäarvon ominaisuuksista, jotka kannattaa ottaa tarjouspyynnön välttämättömiksi toimitettaviksi ohjelmiston toiminnallisuuksiksi.

Lähdeluettelo

Hoerl, A. E.; R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics. 12 (1): 55–67.

Ikonen, Niina et al. (2015-2019) Influenssakausi Suomessa, Seurantaraportit. THL, 2015-2019.

Open Source Initiative (2020). The MIT License. Verkkoaineisto: *https://opensource.org/licenses/MIT Haettu 7.4.2020*

PSHP (2019). Pirkanmaan influenssaepidemia on käynnistynyt. Verkkoaineisto: https://www.tays.fi/fi-FI/Pirkanmaan_influenssaepidemia_on_kaynnis(83125) *Haettu 31.3.2020.*

THL (2020). Influenssa-aktiivisuus : Influenssakäynnit terveyskeskuksissa sairaanhoitopiireittäin ja viikoittain, kaudet 2016−2020 (ILI). Verkkoaineisto: *https://www.thl.fi/ttr/gen/kuvaaja/shpkartta.pdf Haettu 31.3.2020.*

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.