Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Lari Niittylä

# Data-analytiikkatoiminnallisuus startup-yritykselle

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Insinöörityö

29.05.2019

| | |
|---|---|
| Tekijä<br>Otsikko<br><br>Sivumäärä<br>Aika | Lari Niittylä<br>Data-analytiikkatoiminnallisuus startup-yritykselle<br><br>46 sivua + 1 liitettä<br>29.5.2019 |
| Tutkinto | insinööri (AMK) |
| Tutkinto-ohjelma | Tieto- ja viestintätekniikka |
| Ammatillinen pääaine | Ohjelmistotuotanto |
| Ohjaajat | Toimitusjohtaja Teemu Makkonen<br>Lehtori Vesa Ollikainen |


Tämän insinöörityön tarkoituksena on tarkastella, kuinka data-analytiikkaa ja growth-hacking -menetelmiä voidaan hyödyntää startup-yrityksen kasvussa. Työ on tehty Necunos Oy -nimiselle startup-yritykselle. Tarkoituksena on kerätä dataa yrityksen verkkosivuvierailijoista ja sähköpostilistalle kirjautuneista henkilöistä ja tutkia, kuinka data-analytiikkaa voidaan hyödyntää yrityksen kasvun tarkastelemiseksi ja kehittämiseksi.

Tavoitteena on kehittää toiminto, joka kerää tiedot yrityksen verkkosivuvierailijoista sekä sähköpostilistalle kirjautuneista henkilöistä. Toiminnon tarkoituksena on myös käsitellä nämä tiedot alustavasti, jotta niiden jatkokäsittely ja tulosten analysoiminen BI-järjestelmällä helpottuu. Toiminto ja datan käsittely sekä analysointi kuvataan sillä tasolla, että soveltaminen on mahdollista muissa vastaavanlaisissa tapauksissa.

Kasvua tarkastellaan verkkosivuvierailijoiden lukumäärällä ja aktiivisten käyttäjien osuudella kaikista sähköpostilistalle kirjautuneista henkilöistä. Samalla mitataan, kuinka paljon sosiaalisen median julkaisut sekä yrityksen nettisivuilla julkaisemat blogit ja uutiset vaikuttavat kasvuun.



| | |
|---|---|
| Avainsanat | data-analytiikka, python, pandas |

| | |
|---|---|
| Author | Lari Niittylä |
| Title | Data analytics functionality for a startup company |
| Number of Pages | 46 pages + 1 appendices |
| Date | 29 May 2019 |
| Degree | Bachelor of Engineering |
| Degree Programme | Information and Communication Technology |
| Professional Major | Software Engineering |
| Instructors | Teemu Makkonen, Vice President<br>Vesa Ollikainen, Senior Lecturer |

The focus of this thesis is to scrutinize the utilization of data-analytics and growth-hacking methods in growth of a startup company. The thesis project has been carried out for a startup company called Necunos Oy. The purpose is to gather information about the company's website visitors and subscribers and examine how data analytics can be used to scrutinize and develop the company's growth.

The goal is to create a feature which collects data about the company's website visitors and subscribers as well as cleans the data to make further processing and analysis within a BI system easier. This procedure is meant to be written in such detail that it's easy to apply to different cases.

Growth is measured with number of website visitors, subscribers and number of active subscribers amongst them. The effect of social media and other publications on company's webpage are measured as well.

| | |
|---|---|
| Keywords | data analytics, python, pandas |

## Sisällys

Liite 1. Datan hakemiseen tehdyn toiminnon ohjelmakoodi kokonaisuudessaan

## Lyhenteet

| | |
|---|---|
| BI | Business intelligence. Liiketoimintatiedon hallinta, yrityksen liike-elämän tietojen hallinnan ja analysoinnin suorittamista. |
| SEO | Search Engine Optimization. Hakukoneoptimointi, toimenpide, jolla pyritään parantamaan verkkosivun tai -sivuston tuloksia hakukoneiden listaamissa hakutuloksissa. |
| KPI | Key performance indicator. Suorituskykymittari, jolla mitataan yrityksen toiminnalle tärkeitä lukuja tai arvoja. |
| CTR | Click through rate. Mittari, joka kertoo, kuinka moni sähköpostin saaneista klikkasi sen sisältämää linkkiä. |
| EOR | Email open rate. Mittari, joka kertoo, kuinka moni sähköpostin saaneista avasi sähköpostin. |
| CTOR | Click to open rate. Mittari, joka kertoo, kuinka moni sähköpostin avanneista klikkasi sen sisältämää linkkiä. Käytetään sähköpostien sisällön laadun mittaamisessa. |
| VC | Viral Coefficient. Kerroin, joka kertoo, kuinka monta uutta käyttäjää nykyinen tuotteen käyttäjä saa houkuteltua tuotteen käyttäjäksi. |
| A/B-T | A/B-Testaus. Testimalli, jolla testataan tuotetta, antamalla kaksi erilaista versiota tuotteesta, kahdelle eri käyttäjäryhmälle. |
| CAC | Customer acquisition cost. Summa, joka kertoo, paljonko rahaa kuluu yhden uuden asiakkaan hankkimiseksi. |
| LTV | Lifetime value of the customer. Oletettu summa, joka kertoo, kuinka paljon rahaa yksi asiakas tuottaa yritykselle arviolta sinä aikana, kun käyttää yrityksen tuotetta. |

| | |
|---|---|
| OOP | Object-oriented programming. Ohjelmointiparadigma, jossa ohjelmointiongelmien ratkaisut jäsennetään olioiden yhteistoimintana. |
| BSD | Berkley software distribution lisenssi. Vapaa ohjelmistolisenssi, joka on yksi käytetyimmistä avoimen lähdekoodin lisensseistä. Lisenssin yksinkertaiset ehdot takaavat sen, että BSD-lisenssin omaavalle koodille, kuka tahansa saa käytännössä tehdä mitä vain. |
| PSF | Python software foundation lisenssi. Vapaa ohjelmistolisenssi, jota käytetään Python-projektin ohjelmistoissa ja niiden levityksessä. Lisenssi sallii muokattujen versioiden levittämisen ilman lähdekoodia. |
| 2D | 2-dimensional. Kaksiulotteisen digitaalisen kuvan tai tekstin tuottamiseen tarkoitettu tekniikka. Ulottuvuudet ovat leveys ja pituus. |
| CSV | Comma-separated values. Tiedostomuoto, jolla tallennetaan yksinkertaista taulukkomuotoista tietoa tekstitiedostoon. |
| HTML | Hypertext Markup Language. Avoimesti standardoitu kuvauskieli, jolla kuvataan hypertekstiä. Tunnetaan parhaiten internetsivujen kirjoituskielenä. |
| XML | Extensible Markup Language. Metakieli, jolla kuvataan tiedon rakenne ilman ennalta määrättyjä koodeja. |
| URL | Uniform Resource Locator. Osoite, joka viittaa verkkosivun sijaintiin internetissä. Arkikielessä puhutaan internetosoitteesta tai nettisivusta. |
| HTTP | Hypertext Transfer Protocol. Käytäntö, jota WWW-palvelimet ja selaimet käyttävät tiedonsiirtoon. |

# 1 Johdanto

Tämän insinöörityön tarkoituksena on käsitellä data-analytiikan ja growth-hacking-menetelmän hyödyntämistä startup-yrityksen kasvussa. Työ on tehty Necunos Oy:lle ja siinä on hyödynnetty yrityksen keräämää dataa nettisivuvierailijoista sekä yrityksen sähköpostilistalle kirjautuneista henkilöistä 09/2018 – 02/2019 väliseltä ajanjaksolta. Necunos Oy on nuori yritys, jolla ei ole vielä toistaiseksi liikevaihtoa, jonka vuoksi työssä ei voida tarkastella yrityksen kasvua perinteisemmillä mittareilla. Tästä syystä työssä tarkastellaan ja mitataan kasvua yrityksen nettisivuilla kävijöillä, sähköpostituslistalle kirjautuneilla sekä heidän aktiivisuudellaan. Työssä pyritään huomioimaan, miten sosiaalisen median postaukset ja uutiskirjeet ovat tähän vaikuttaneet sekä pohtimaan mahdollisia kehitystarpeita tai -ehdotuksia.

Työn tavoitteena on toteuttaa toiminto, joka hakee verkkosivuvierailijoiden tiedot ja muokkaa ne BI-järjestelmän jatkokäsittelyä varten sekä tuottaa näistä lopulta yhteenvedon. Tämä prosessi ja siinä käytetyt tekniikat pyritään kuvaamaan sillä tarkkuudella, että työn lukenut pystyy soveltamalla hyödyntämään niitä omassa kontekstissaan. Toiminto toteutetaan Python-ohjelmointikoodilla hyödyntämällä pääosin Pythonin Pandas-kirjastoa, joka on erinomainen työkalu datan analysoimiseen sekä muokkaamiseen. Verkkosivuvierailijoiden tiedot on tarkoitus hakea Necunos Oy:n käyttämän verkkosivuanalytiikka palvelun kautta html-taulukkoformaatissa. Toiminto tulee toteuttaa siten, että tietojen haku, siistiminen ja tallennus on automatisoitu. Tämän jälkeen kehitetään toiminnallisuudet datan käsittelemiseen ja analysoimiseen, jotta sitä voidaan hyödyntää BI-järjestelmässä ja tuottaa yhteenvedot sen avustuksella. Toiminnon toteuttamisessa tulee ottaa huomioon, minkälaisia tiedostoformaatteja BI-järjestelmät ja tietokoneen oma käyttöjärjestelmä tukevat, jotta järjestelmät pystyvät ongelmitta käsittelemään toiminnon luomia tiedostoja. Tiedostojen salaustyyppien yhteensopivuus on toteutettava niin, että ne ovat luettavissa erilaisilla BI-järjestelmillä ja Pythonissa.

Työn teoreettisessa osassa käsitellään startup-yrityksen kasvussa hyödynnettäviä menetelmiä tämän työn konteksti huomioon ottaen. Tarkoituksena on selventää, miten sosiaalisen median ja datan hyödyntäminen auttaa kasvussa, mitä tarkoittaa growth

hacking ja kuinka näitä menetelmiä pystyttäisiin käyttämään työssä. Sivutaan myös hieman vaihtoehtoisia lähestymistapoja sekä pyritään pohtimaan lyhyesti, miten kyseisiä toimintoja ja tekniikoita voitaisiin soveltaen hyödyntää toisenlaisissa töissä.

Työn teknisessä osiossa käydään läpi, miten edellisessä kappaleessa esiteltyjä tekniikoita on hyödynnetty ja mitä ohjelmistoja työssä on käytetty. Tarkoituksena on esittää kattavasti, miten Pythonia ja sen tarjoamia kirjastoja on käytetty datan keräämisessä, käsittelyssä ja analysoinnissa sekä käydä läpi muutamia vaihtoehtoisia esimerkkejä, miten niitä voitaisiin vaihtoehtoisesti hyödyntää erilaisessa tilanteessa. Lopussa selitetään, miten datan visualisointi on toteutettu ja käydään läpi muutamia erilaisia vaihtoehtoja datan visualisoimiseen ja -esittämiseen.

Lopullisten ratkaisujen ja tulosten arvioinnissa pohditaan, kuinka hyvin työssä käytetyt teknologiat ovat soveltuneet tavoitteiden saavuttamiseksi ja miten tavoitteet ovat ylipäätään täyttyneet. Pohditaan rehellisesti työn tuottamaa hyötyä yritykselle ja pyritään esittämään vaihtoehtoisia toimintatapoja saman ratkaisun ja tavoitteen saavuttamiseksi sekä löytämään mahdollisia parannuksia tai toimintamalleja tulevaisuuteen peilaten.

Viimeisessä luvussa käydään läpi yhteenveto työstä kokonaisuudessaan. Käydään lyhyesti läpi, mitä tehtiin ja arvioidaan omaa prosessia työn toteuttamisessa. Lopuksi pyritään hieman kriittiseen ja rehelliseen pohdintaan työn etenemisessä kohdatuista haasteista ja niiden ratkaisemisesta.

## 2 Teoria työn taustalla

Tässä luvussa kerrotaan lyhyesti Necunos Oy:stä ja valaistaan hieman teoriaa työn taustalla. Startup-yrityksen kasvuun käytettyjä menetelmiä on monenlaisia, joista muutamat suosituimmat ja käytetyimmät menetelmät ovat valikoituneet hyödynnettäväksi tässä työssä. Kokonaisuudessaan näitä teorioita ei tulla käymään läpi vaan teoriaa pyritään avaamaan työssä hyödynnettyjen menetelmien osalta, jotka soveltuvat Necunos Oy:n tämänhetkiseen profiiliin nuorena startup-yrityksenä.

## 2.1 Necunos Oy ja yrityksen kasvun lisäämisessä hyödynnettävät menetelmät

Necunos Oy on muutaman vuoden ikäinen startup-yritys, jonka pääasiallinen toimiala on tietoturva. Yritys on työskennellyt muutamissa erilaisissa ohjelmistotuotannollisissa projekteissa, tällä hetkellä se on yhteistyökumppanin kanssa luomassa uutta, tietoturvallista puhelinta, joka perustuu avoimeen lähdekoodiin niin ohjelmisto- kuin laitetasollakin. Vaikka yrityksen tulevaisuuden näkymät vaikuttavat lupaavilta, on se vasta alkutaipaleella, eikä tuote ole vielä täysin valmis tätä työtä tehdessä.

Startup-yrityksellä tarkoitetaan nuorta, yleensä korkeintaan muutaman vuoden ikäistä yritystä, joka ei tuota voittoa kehittäessään vasta ensimmäistä tuotettaan. Tämän vuoksi startup-yritykselle ominaista ja tärkeää on mahdollisimman nopean kasvun tavoittelu. [1.] Necunos Oy:n kasvun vauhdittamiseksi on tärkeää tarkastella niitä keinoja, joilla ihmiset saadaan tietoisemmiksi yrityksestä ja sen palveluista sekä kehitteillä olevista tuotteista.

### 2.1.1 Sosiaalinen media

Sosiaalisen median käyttö lisää yrityksen näkyvyyttä ja postaukset sosiaalisen median eri alustoilla kasvattavat ihmisten tietoisuutta yrityksestä, mikä kasvattaa verkkosivuvierailijoiden määrää sekä sivujen SEO:ta [2].

Necunos Oy:llä on Twitter-, Facebook- ja LinkedIn-tilit. Tämän työn osalta on tarkasteltu Twitterissä ja Facebookissa julkaistujen postauksien vaikutuksia yrityksen verkkosivuvierailijoiden sekä sähköpostituslistalle kirjautuneisiin, yrityksestä ja sen tuotteista kiinnostuneiden käyttäjien lukumäärään. Yrityksen Twitter-tili on luotu 2017 ja Facebook-tili on luotu joulukuussa 2018. Tästä syystä Twitter-tilillä on huomattavasti enemmän seuraajia kuin Facebook-tilillä. Teoreettisesti voidaan olettaa, että postausten määrät ja verkkosivuvierailijoiden lukumäärä korreloivat jollain tasolla keskenään.

### 2.1.2 Datan hyödyntäminen yrityksen kasvussa

Valtioneuvoston selvitys- ja tutkimustoiminnan raportista selviää, että vuosina 2012 – 2014 liikevaihto kasvoi 17 prosenttia niillä informaatio- ja viestintätoimialan yrityksillä, jotka hyödynsivät dataa uusia palveluita tai tuotteita kehittäessään verrattuna saman alan yrityksiin, jotka eivät dataa hyödyntäneet [3]. Yrityksen nykytila sekä sen tarjoamat

palvelut ja tuotteet vaikuttavat luonnollisesti siihen, minkälaista dataa on käytössä ja kuinka sitä voidaan hyödyntää.

Croll ja Yoskovitz (2013) käsittelevät teoksessaan *Lean Analytics: Use Data to Build a Better Startup Faster*, kattavasti ja monipuolisesti datan hyödyntämistä Startup-yrityksen kasvussa [4]. Crollin ja Yoskovitzin (2013, 9-10) mukaan, on tärkeää löytää yrityksen kasvuun vaikuttava mittari, jolla kasvua seurataan. Hyvä mittari on suhdeluku, joka on verrattavissa ja helposti ymmärrettävä. Necunos Oy:n nykytilanteen huomioiden kasvun mittaamiseen ei voida käyttää monipuolista valikoimaa erilaisia mittareita, vaan joudutaan tyytymään verkkosivuvierailijoiden ja sähköpostilistalle kirjautuneiden lukumäärään sekä viimeksi mainitun ryhmän aktiivisuuteen sähköpostien avaamisen ja sen sisältämien linkkien klikkaamisen suhteen.

Kasvun mittaamista varten kerättävää dataa hankittaessa on hyvä ottaa huomioon, minkälaisia arvoja datasta voidaan ammentaa ja minkälaista hyötyä niillä pyritään saavuttamaan. Tässä työssä käytetyn datan kannalta on hyvä erottaa toisistaan varsinkin seuraavat asiat.

- turhat ja vaikuttavat arvot
- laadulliset ja määrälliset arvot
- johdattelevat ja vanhat arvot
- korreloivat ja syyperäiset arvot.

Dataa kerätessä kannattaa pitää mielessä sen lopullinen käyttötarkoitus. Nykypäivänä dataa on niin paljon saatavilla, että siitä tulee helposti analysoitua vääriä asioita. Täytyy pystyä hahmottamaan, mitä datan avulla halutaan selvittää ja keskittyä siihen liittyvien arvojen tutkimiseen. Vaikuttava data auttaa hahmottamaan suuntaa, johon yritys on menossa ja sillä on enemmän merkitystä kuin turhalla datalla, joka saattaa korkeintaan näyttää hyvältä. Tässä työssä vaikuttavaa dataa on niiden sähköpostilistalle kirjautuneiden käyttäjien, jotka lukevat suurimman osan sähköposteista, osuus heistä, jotka eivät niitä lue. Tämä siitä syystä, että yritykselle eniten hyötyä tuottavat sitoutuneet asiakkaat. Työn luonteesta riippuen, esiintyy siinä myös turhia arvoja, kuten esimerkiksi verkkosivuvierailijoiden kansallisuudet, joista ei varsinaisesti ole mitään hyötyä yrityksen kasvuun liittyen.

Työssä esiintyy ainoastaan määrällisiä arvoja, sillä kerätty data koostuu pelkästään tilastoista, joista saadaan vain numeraalisia arvoja. Työssä ei ole hyödynnetty laadullisia arvoja, joita saadaan käyttäjäkyselyistä ja -haastatteluista. Tämä johtuu yksinkertaisesti siitä, että Necunos Oy:llä ei ole vielä varsinaisesti tuotetta käyttäviä asiakkaita, joiden mielipiteitä tiedustelemalla voitaisiin hankkia laadullisia arvoja yrityksen hyödynnettäväksi.

Johdatteleva data tarjoaa ennustetta tulevaisuudelle, jonka vuoksi sillä on tärkeä osa yrityksen kehityksessä. Johdattelevan datan etu on siinä, että sen tuomiin muutoksiin pystytään vielä reagoimaan. Yrityksen alkutaipaleella sen arvoa ei välttämättä ole vielä tiedostettu tai edes löydetty. Tässä työssä ei välttämättä vielä tulla tekemään löytöjä Necunos Oy:lle johdattelevan datan suhteen, mutta työn edetessä on hyvä pitää silmät auki sen varalta. Vanha data sen sijaan selittää mennyttä ja sitä voidaan käyttää lähinnä tilastoimiseen ja yrityksen historian tarkastelemiseen. Tässä työssä esitelty data tulee pääosin olemaan vanhan datan analysoimista, jonka avulla voidaan mahdollisesti löytää arvoja, joiden avulla pystytään tekemään johtopäätöksiä tulevaisuuden suhteen.

Kahden eri arvon muuttaessa toisiaan on kyseessä korreloiva data, kun taas jonkin arvon muuttuminen aiheuttaa toisen arvon muuttumisen. Tällöin puhutaan syyperäisestä datasta. Korreloivaa dataa on kaikkialla, eikä siitä välttämättä ole aina kovinkaan paljon hyötyä, sillä useat asiat korreloivat keskenään ilman, että ne vaikuttaisivat suoraan toisiinsa. Hyvänä esimerkkinä voidaan ottaa jäätelön myynti ja hukkumiset. Vaikka nämä kaksi asiaa korreloivat keskenään, ei se tarkoita, että pelkästään jäätelön myynnin kasvu lisää hukkumisten määrää. Hukkumisten määrän kasvuun liittyy paljon muitakin syitä esimerkin tapauksessa, kuten kesälomat, alkoholin kulutus ja muut vastaavat seikat (Croll & Yoskovitz, 2013, s. 20). Syyperäinen data sen sijaan on hyvinkin arvokasta yritykselle, sillä sen avulla voidaan vastata kysymykseen, miksi jotain tapahtuu. Tässä työssä korreloivaa dataa esiintyy luultavasti esimerkiksi sosiaalisen median ja muiden julkaisujen ja verkkosivuvierailijoiden määrän suhteen. Syyperäinen data auttaa esimerkiksi selventämään, mikä näissä julkaisuissa on sellaista, joka lisää tai vähentää verkkosivuvierailijoiden määrää.

Tärkeintä on löytää yksi tai useampi KPI (Suorituskykymittari, jolla mitataan yritykselle tärkeiden arvojen muutoksia), minkä avulla pystytään pitämään analyysin tavoite koko

ajan kirkkaana ja voidaan seurata yrityksen edistymistä halutulla osa-alueella. Necunos Oy:n KPI on tällä hetkellä aktiivisemmat ja sitoutuneemmat sähköpostilistalle kirjautuneet käyttäjät, sillä heidän lukumääränsä antaa parhaan arvion yksityisten ihmisten kiinnostuksesta kehitteillä olevaa tuotetta kohtaan.

Sähköpostilistalle kirjautuneiden käyttäjien aktiivisuutta ja samalla sähköpostien sisällön kiinnostavuutta voidaan mitata kolmella erilaisella mittarilla.

- CTR (Click through rate)
- EOR (Email open rate)
- CTOR (Click to open rate).

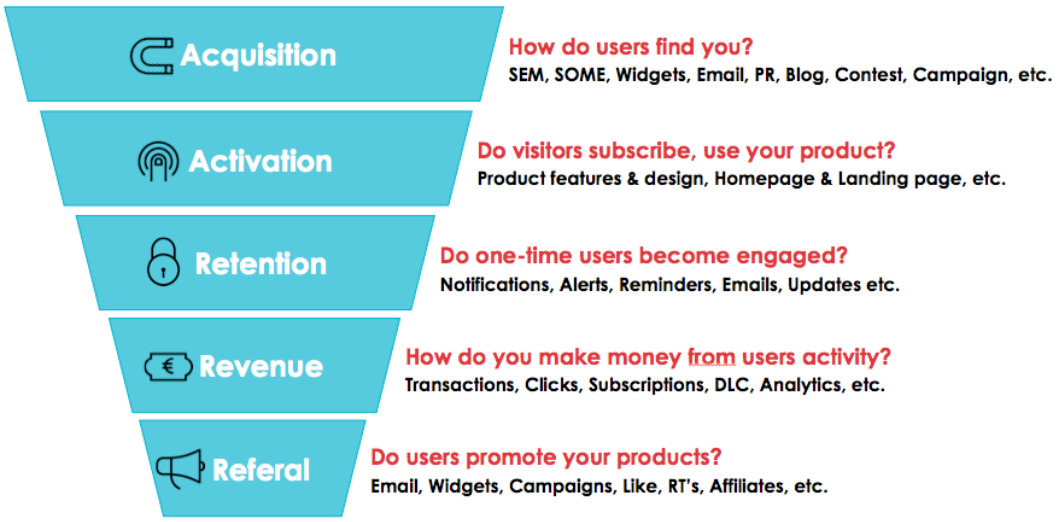
CTR-arvoon [5] vaikuttaa luonnollisesti sähköpostin sisältämien linkkien määrä, se kertoo, kuinka houkuttelevaa sähköpostin sisältö lukijalle on. Hyvät CTR-arvot ovat yleensä noin 15 % pinnassa [6]. EOR-arvoon [7] vaikuttavat esimerkiksi kohderyhmä ja sähköpostin otsikointi. Se kertoo, kuinka moni käyttäjistä on avannut sähköpostin ja hyvä EOR-arvo on yleensä 20 %:n ja 40 %:n välillä [8]. CTOR-arvo [9] kertoo, kuinka kiinnostavana käyttäjät pitävät sähköpostien sisältöä ja se on hyvä verrokki arvo, kun verrataan sen keskiarvoa ja yksittäisin sähköposti kampanjan saavuttamaa arvoa. Hyvä CTOR-arvo on yleensä 20 %:n ja 30 %:n välillä [10].

### 2.1.3 Growth Hacking

Sean Ellis on ensimmäisenä keksinyt termin ’growth hacker’ vuonna 2010.

> “A growth hacker is a person whose true north is growth."

Termin mukaan growth hacking eroaa perinteisestä markkinoinnista siten, että siinä keskitytään vain ja ainoastaan kasvuun, mikä on ainoa tavoite aloittelevalle startup-yritykselle. Neil Patel ja Bronson Taylor määrittelevät kattavasti growth hacking -menetelmän julkaisussaan ”*The Definitive Guide to Growth Hacking*”, jossa käydään läpi erilaisia tekniikoita yrityksen nettisivujen kävijämäärän kasvattamiseksi, asiakkaiden hankkimiseksi ja sitoutuneiden asiakkaiden määrän kasvattamiseksi. [11.]




Kuva 1. Havainnekuva growth hacking suppilosta [12].

Patelin ja Taylorin mukaan (ibid.) yrityksen kasvun vauhdittamiseksi on ensin luotava suunnitelma, joka ei ole turhan laaja ja tavoite on hyvä rajata mahdollisimman tarkasti. Kun tämä suunnitelma ja tavoite on selvillä, hyödynnetään analytiikkaa tavoitteen edistymisen tarkastelussa.

Necunos Oy:n kasvun vauhdittaminen on vasta alkutaipaleella, joten aivan kaikkiin suppilon osiin ei pystytä työssä kovinkaan kattavasti pureutumaan. Ensimmäisenä on luonnollisesti lisättävä yrityksen näkyvyyttä mm. sosiaalisen median kautta, jotta yrityksen verkkosivuvierailijoiden lukumäärää saadaan kasvatettua. Tätä kautta pyritään nostamaan myös yrityksen sähköpostilistalle kirjautuneiden lukumäärää, jotta mahdollisten tuotteesta ja yrityksestä kiinnostuneiden henkilöiden lukumäärää saadaan kasvatettua. Heistä aktiivisten käyttäjien osuutta tarkastelemalla pyritään löytämään keinoja kävijöiden aktiivisuuden lisäämiseksi.

Growth Hacking -menetelmiin käytettyjä tekniikoita on monenlaisia, eikä niitä kaikkia ole luonnollisesti työssä hyödynnetty. Pääosin nämä tekniikat jakautuvat ”push”-, ”pull”- ja ”product”-tekniikoihin. ”Push”-tekniikalla pyritään ns. pakottamaan käyttäjät sivustolle esimerkiksi ostamalla mainostilaa muualta, tekemällä promootiovaihtoja toisen yrityksen kanssa tai hyödyntämällä yhteistyökumppaneita muulla tavalla. ”Product”-tekniikka vaatisi jo valmiin tuotteen, jolla käyttäjiä voitaisiin kannustaa tuotteen levittämiseen ja kehumiseen erilaisin kannustimin.

Necunos Oy:n nettisivujen liikennettä on kasvatettu täysin ’pull’-tekniikalla, joka tarkoittaa kävijöiden ohjaamista sivustolle. Siihen käytettyjä keinoja ovat olleet.

- sosiaalisen median julkaisut
- blogit ja uutisointi.

Pelkkä verkkosivuvierailijoiden tietojen kerääminen ei luonnollisesti riitä, vaan heitä pitää aktivoida käyttämään tai hankkimaan tuotetta, eli tässä tapauksessa kirjautumaan sähköpostilistalle. Seuraavia keinoja on hyödynnetty kävijöiden sähköpostilistalle kirjautumiseen.

- Houkutellaan kävijät kirjautumaan sähköpostilistalle tai muutoin aktivoitumaan heti ensimmäisellä sivulla, joka avautuu heille nettisivuilla vierailessaan.
- Kerrotaan käyttäjille, miksi yrityksen tuote on uniikki ja tarpeellinen.

Vierailijoiden muuttuessa aktiivisiksi käyttäjiksi on tärkeää pitää heistä kiinni. Sähköpostitse säännöllisesti lähetettävät uutiskirjeet ja vastaavanlaiset kampanjat ovat mainio tapa, joita myös Necunos Oy hyödyntää. Kommuunin rakentaminen edistää myös asiakkaiden sitoutumista. Necunos Oy:n tavoitteena on saada avoimenlähdekoodin kommuunia tietoisemmaksi tuotteistaan ja kasvattaa sitä niin, että kiinnostusta kehittää sisältöä yrityksen tulevalle mobiililaitteelle löytyy kattavasti kommuunin sisältä. Ensi askeleet tähän on jo otettu yhteistyösopimuksilla useiden avoimen lähdekoodin ja mobiililaitteiden parissa toimivien yritysten kanssa.

Tulevaisuudessa yrityksen saavutettua vaiheen, jossa tuotteen käyttäjiä on useita ja se tuottaa rahaa yritykselle, voidaan alkaa hyödyntää erilaisia tekniikoita tuotteen kehittämiselle.

- VC
- A/B-T
- Kohortti
- Segmentointi
- CAC
- LTV.

Analytiikan avulla voidaan alkaa selvittämään erilaisten käyttäjäryhmien mieltymyksiä ja käyttötapoja, optimoida yrityksen rahankäyttöä sekä luoda omia, yrityksen tarpeisiin kustomoituja malleja, joilla tuotteen kehitystä voidaan ohjata oikeaan suuntaan.

# 3 Työn toteutus

Necunos Oy on pieni startup-yritys, joka sijaitsee Helsingissä. Yrityksen päätoimiala on tietoturvallisuus, ja se työstää useampaa erilaista projektia samanaikaisesti. Tämä työ on toteutettu yrityksen tuotteiden ja sen nettisivujen näkyvyyden lisäämiseksi sekä potentiaalisten asiakkaiden määrän kasvattamiseksi. Necunos Oy kehittää tietoturvallista, täysin avoimeen rautaan ja lähdekoodiin perustuvaa puhelinta, jota on tarkoitus markkinoida aluksi yritysten sisäiseen käyttöön ja samalla saada varhaiset käyttäjäryhmät tietoisiksi ja kiinnostuneiksi tuotteesta.

## 3.1 Työssä käytetyt teknologiat

Työn toteutuksessa käytetyt teknologiat ja ohjelmistot, kuten Matomo ja MailerLite, valikoituivat siitä syystä, että ne olivat jo valmiiksi Necunos Oy:n käytössä. Ohjelmointikielet ja niiden alusta valittiin työhön sekä ominaisuuksiensa perusteella ja niistä oli jo ennestään hieman kokemusta, jonka vuoksi ne koettiin parhaaksi tavaksi työn toteutuksessa.

### 3.1.1 Matomo

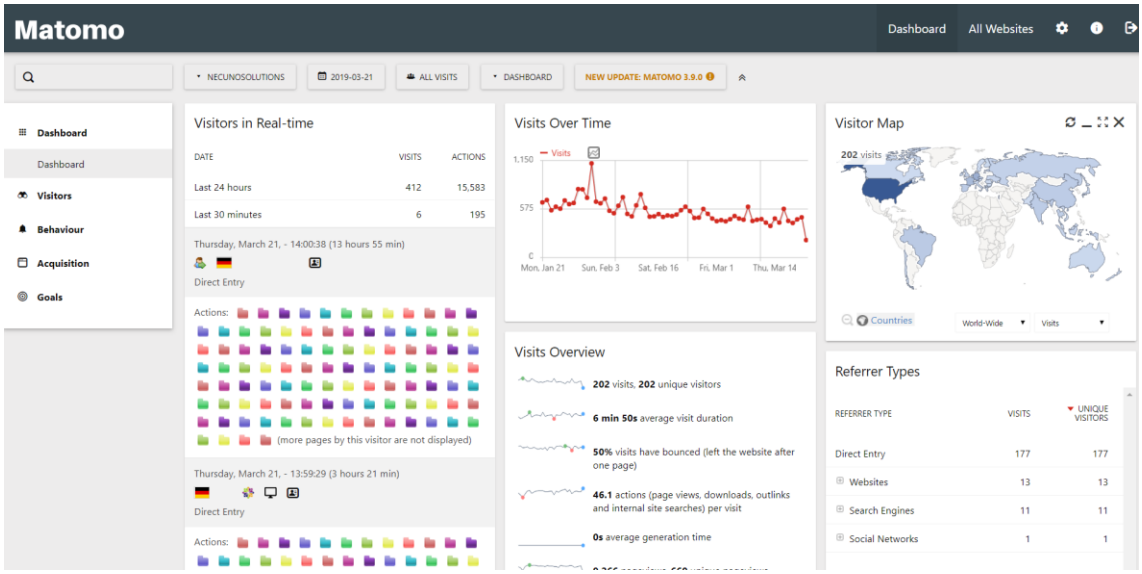
Matomo, alun perin Piwik, on avoimeen lähdekoodiin perustuva web-analytiikkaohjelmisto, joka on saanut alkunsa Matthieu Aubry -nimisen insinööriopiskelijan projektista vuonna 2007. Alkuperäisenä tavoitteena oli kehittää vaihtoehtoinen analytiikkaohjelmisto, joka suojaa käyttäjien yksityisyyttä ja antaa heille täyden hallinnan datan käyttöön. Nykyään Matomo on johtava avoimeen lähdekoodiin perustuva web-analytiikkaohjelmisto, jota käyttää yli 1,4 miljoonaa nettisivua yli 190 eri maassa. [13.]

Matomo tarjoaa useita erilaisia valmispaketteja, joiden hinta riippuu täysin paketin sisällöstä ja kattavuudesta. Koska kyseessä on kuitenkin avoimeen lähdekoodiin perustuva

ohjelmisto, voi sitä hyödyntää maksutta. Tässä työssä on käytetty Matomon versiota, joka on asennettu yrityksen omille palvelimille. Asennetun version toiminta perustuu apachen lokitietoihin, JavaScript-seurannan sijasta, jonka vuoksi yrityksen nettisivuvierailijoista saatavilla oleva informaatio eikä data ole aivan niin monipuolista. Matomon avulla hankittu informaatio sisältää kattavat tiedot yrityksen nettisivuvierailijoista, kuten

- IP-osoite (ainoastaan verkon peite)
- vierailun ajankohta
- käyttäjän tekemät toiminnot ja niiden lukumäärä
- vierailun kesto
- yrityksen sivustolle viitannut lähdesivusto
- käyttäjän sijainti
- käyttäjän käyttämä laite, resoluutio ja käyttöjärjestelmä.

Matomo on käynnissä yrityksen omilla palvelimilla, joihin ei pääse ulkopuolelta käsiksi. Matomon log analytics -toiminto hakee kerran vuorokaudessa yrityksen verkkosivujen apache-lokitiedot ja luo niiden perusteella tiedot päivittäisistä verkkosivuvierailuista.




Kuva 2. Kuvakaappaus Matomo web -analytiikkaohjelmiston etusivulta.

### 3.1.2 Python

Python on OOP-ohjelmointikieli, jonka on alun perin kehittänyt Guido van Rossum 1980-luvun lopulla [14]. Pythonin monipuolisuus, tulkattavuus ja selkeä syntaksi tekevät siitä erinomaisen ohjelmointikielen useisiin eri tarkoituksiin ja monille eri alustoille [15]. Ominaisuuksiensa ansiosta Pythonilla kirjoitetut ohjelmat voidaan ajaa välittömästi, eikä niitä tarvitse kääntää ensin. Laajan ja monipuolisen, avoimeen lähdekoodiin perustuvan, kirjaston avulla voidaan Pythoniin lisätä paljon erilaisia toiminnallisuuksia, joita voidaan hyödyntää nettisivujen luonnissa, tieteellisissä ja matemaattisissa tehtävissä sekä useissa erilaisissa sovelluksissa. [16.]

### 3.1.3 Pandas

Pandas on BSD-lisensoitu, avoimeen lähdekoodiin perustuva kirjasto, jonka avulla pythonia pystyy helposti ja monipuolisesti hyödyntämään datan käsittelyssä ja analysoinnissa. [17.]

- nopea ja tehokas DataFrame olio datan käsittelyyn
- kattavat työkalut datan lukemiseen ja kirjoittamiseen useilla eri tiedostoformaateilla
- tehokkaat ja valmiit funktiot monipuoliseen datan käsittelyyn ja analysointiin.

### 3.1.4 Matplotlib

Matplolib [18] on avoimeen lähdekoodiin perustuva, PSF-lisensoitu kirjasto, joka tarjoaa laajan valikoiman 2D-graafikalla toteutettuja kuvaajia datan, tilastojen sekä lukujen havainnollistamiseen. Erityisen hyödyllisiä kuvaajia datan käsittelyssä ovat histogrammit ja boxplotit. [18.]

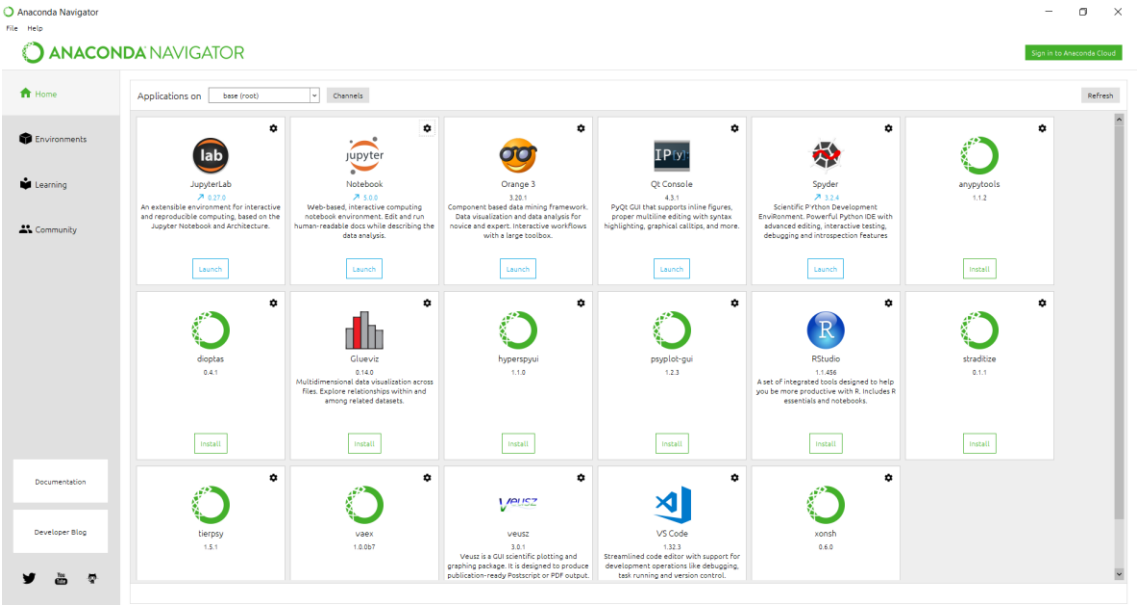
### 3.1.5 BeautifulSoup

BeautifulSoup on Python-kirjasto, jonka avulla voidaan hankkia dataa HTML- ja XML-tiedostoformaateista. Kirjasto tarjoaa monipuoliset työkalut tarvittavan tiedon haalintaan verkkosivuilta. [19.] Yleisesti ottaen tiedon haalimisessa verkkosivuilta on syytä ottaa

huomioon muutama asia. Useasti sivujen tieto on peräisin joltakin toiselta sivustolta, eikä välttämättä pidä paikkaansa. Joillakin sivustoilla tiedon haaliminen saattaa olla rajoitettua tai estetty kokonaan, jolloin kaiken halutun tiedon hankkiminen on hyvin haasteellista tai jopa mahdotonta ja joissain tapauksissa se saattaa olla jopa laitonta. [20.]

### 3.1.6 Anaconda ja Jupyter Notebook

Anaconda on avoimeen lähdekoodiin perustuva, erityisesti Python- ja R-ohjelmointikieliin soveltuvien datatiede- ja koneoppimissovelluksia sisältävä sovellusalusta. Anacondan avulla työympäristön kirjastojen ja niiden välisten riippuvuuksien hallinta on erittäin helppoa ja yksinkertaista. [21.] Anaconda-sovellusalusta sisältää useita eri ohjelmistoja, joista tässä työssä on käytetty Jupyter Notebook -nimistä sovellusta.




Kuva 3. Anaconda Navigator -sovellusalustan käyttöjärjestelmä.

Jupyter Notebook on avoimeen lähdekoodiin perustuva sovellus, jota käytetään yleensä datan käsittelyyn, muokkaamiseen ja visualisointiin, tilastojen mallintamiseen sekä koneoppimiseen [22]. Jupyter Notebook koostuu selainpohjaisesta verkkosovelluksesta sekä Notebook-dokumenteista, jotka mahdollistavat dokumenttien interaktiivisen toteutuksen useisiin erilaisiin tarkoituksiin. Dokumentit sisältävät kaikki syötteet ja niiden tulokset, jotka voi toteuttaa useilla eri formaateilla. Sovellus pyörii oletuksena käyttäjän

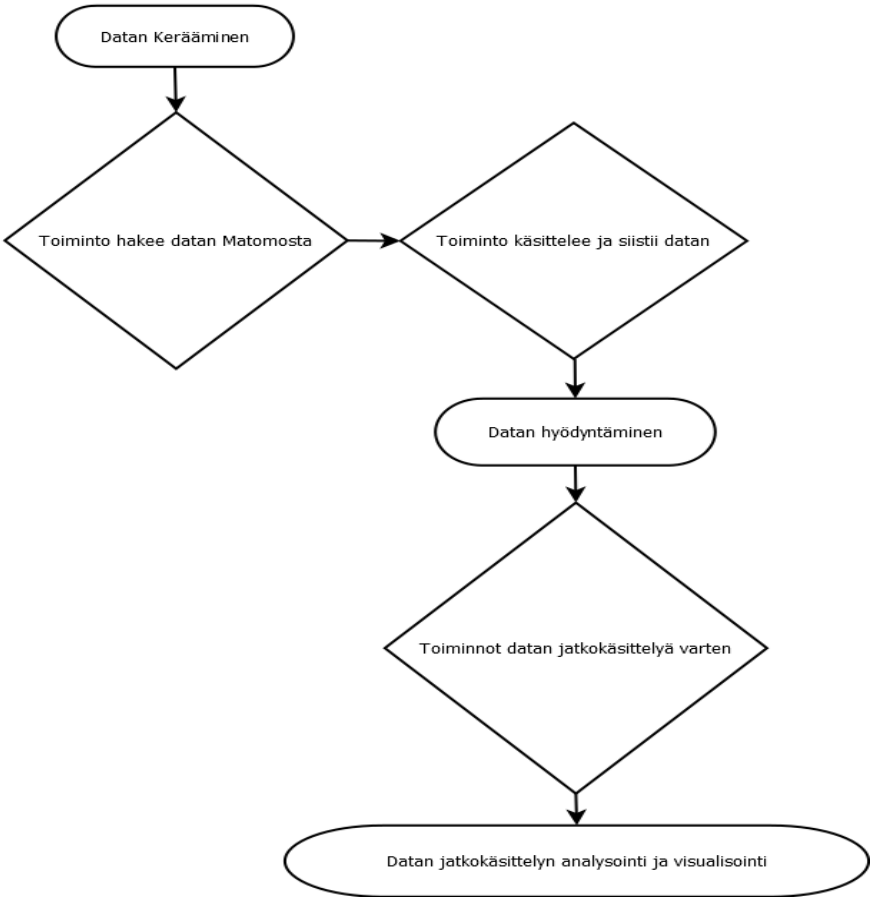
koneen omalla palvelimella, joka takaa tietojen turvallisen käytön. Oletusohjelmointikielenä on Python, mutta kernelin asetuksia muokkaamalla sovellusta voi käyttää myös muilla ohjelmointikielillä. [23.]

### 3.1.7 MailerLite

MailerLite on vuonna 2005 perustettu yritys, joka aloitti verkkosivujen suunnittelulla, mutta siirtyi 2010 sähköpostimarkkinoinnin pariin. Yritys tarjoaa erilaisia paketteja toisille yrityksille sähköpostikampanjoiden tietojen keräämiseksi ja analysoimiseksi. 609 582 yritystä ympäri maailmaa käyttää MailerLite-ohjelmistoa, ja se on erinomainen työkalu yrityksen sähköpostilistalle kirjautuneiden käyttäjien aktiivisuuden tarkastelemiseksi. [24.]

## 3.2 Työvaiheet

Tässä luvussa on kuvattu työn eteneminen vaiheittain. Tavoitteen ollessa tiedossa pystyttiin miettimään, minkälaista dataa halutaan hankkia ja miten sitä tulisi hyödyntää. Samalla pyrittiin noudattamaan luvussa 2 esitettyjä teorioita ja kuvaamaan prosessi sillä tarkkuudella, että se on toistettavissa, myös toisenlaisessa työssä. Kuva 4 esittää työvaiheiden etenemistä, joista ensimmäinen vaihe on datan kerääminen. Datan keräämistä varten luodaan toiminto, joka myös käsittelee ja siistii datan ennen sen tallentamista. Seuraavassa vaiheessa tallennetulle datalle luodaan toiminnot jatkokäsittelyä varten, joiden avulla siitä voidaan selvittää verkkosivuvierailijoiden ja yrityksen sähköpostilistalle kirjautuneiden lukumäärät, lähdesivustot, sähköpostikampanjoinnin tehokkuus sekä aktiivisten ja sitoutuneiden käyttäjien osuus kaikista sähköpostilistalle kirjautuneista.




Kuva 4. Vuokaavio työvaiheista.

### 3.2.1 Datan kerääminen

Työskentelyn ensimmäinen vaihe oli datan hankkiminen. Matomo-ohjelmiston keräämät verkkosivuvierailijoiden tiedot voi ladata useissa eri tiedostoformaateissa omalle koneelle. Alun perin tiedot ladattiin koneelle csv-formaatissa, mutta pian huomattiin, että niiden päivien osalta, joiden kävijämäärät olivat useita satoja, ei kaikkia taulukon tietoja pystynyt lataamaan kerralla, vaan ne olisi pitänyt pilkkoa lataamalla osa taulukosta aina kerrallaan. Tiedot päätettiin ladata html-formaatissa niin, että html-taulukko tallennettiin suoraan koneelle.

Yrityksen sähköpostilistalle kirjautuneiden käyttäjien tiedot on ladattu Necunos Oy:n käyttämältä MailerLite-ohjelmistosta ja ne tallennettiin koneelle csv-tiedostoformaatissa.

Yrityksen verkkosivuvierailijoiden ja sähköpostilistalle kirjautuneiden henkilöiden suhdetta tarkastellessa ilmeni uusi ongelma. Muutamalta päivältä postituslistalle kirjautuneiden kävijöiden osuus kohosi jopa 75 prosenttiin, mikä ei ole millään tavalla realistista. Pian selvisi, että kyseisinä päivinä kävijämäärät olivat aivan liian alhaiset, esimerkiksi 4 verkkosivuvierailua ja 3 sähköpostilistalle kirjautunutta. Lopulta huomattiin, että jostain syystä pandas-kirjaston toiminto, jolla html-tiedostot luetaan dataframe-objekteiksi, ei toiminut kunnolla. Toiminto ei jostain syystä osannut lukea html-tiedostosta kaikkia taulukon tietoja, koska se ei löytänyt kaikkia table tagejä, ja tämän vuoksi joidenkin päivien tietojen osalta toiminto luki vain muutaman rivin taulukosta. Ongelma ratkaistiin lopulta hyödyntämällä Pythonin BeautifulSoup-kirjastoa, joka on erinomainen työkalu webscraping-tekniikan hyödyntämisessä. Kirjastoa hyödyntämällä tiedot haettiin suoraan verkosta niin, että jokaisen päivän kävijätietojen osalta haettiin vastaava url-osoite yrityksen palvelimella pyörivän Matomo-analytiikkaohjelmiston kautta.

Tiedot on kerätty 09/2018 – 02/2019 väliseltä ajalta. Työn luonteen vuoksi jonkinlainen aikaikkuna oli asetettava ja koettiin, että kuuden kuukauden mittainen ajanjakso on sopivan mittainen tarkastelemaan verkkosivuvierailuiden, yrityksen sähköpostilistalle kirjautuneiden henkilöiden määrää ja aktiivisuutta sekä sosiaalisen median että muiden kampanjoiden vaikutusta yrityksen kasvuun.

Päivittäiset verkkosivuvierailijoiden tiedot luettiin pandas-kirjaston dataframe-olioiksi ja ne uudelleen käsiteltiin sekä muokattiin työn tavoitteisiin nähden sopivammiksi, sillä alkuperäiset tiedot sisälsivät 212 saraketta, joista suurin osa oli turhia työn tavoitteiden kannalta. Muokatut tiedostot tallennettiin lopuksi koneelle csv-tiedostoformaatissa myöhempää käyttöä varten. Tallennuksessa oli huomioitava Windows-käyttöjärjestelmän oletussalauksen ja Pythonin tukemien salausten epäsopivuus, jonka vuoksi muokattu tiedosto tallennettiin antamalla sille parametrina utf-8-salaus. Python tukee useita salauksia, mutta Windows-käyttöjärjestelmän oletussalauksien kanssa saattaa tulla yhteneväisyysongelmia. Toinen yleisesti käytetty salaus on ascii-salaus, joka tukee englanninkielisiä versioita.

```
import pandas as pd
df = pd.read_csv(“C:/haluttu tiedosto”)
df.to_csv(”C:/tiedoston nimi”, encoding=’utf-8’)
```

Esimerkkikoodi 1. Csv-tiedoston lukeminen pandas-dataframe-olioksi ja tallentaminen csv-tiedostoksi Pythonilla.

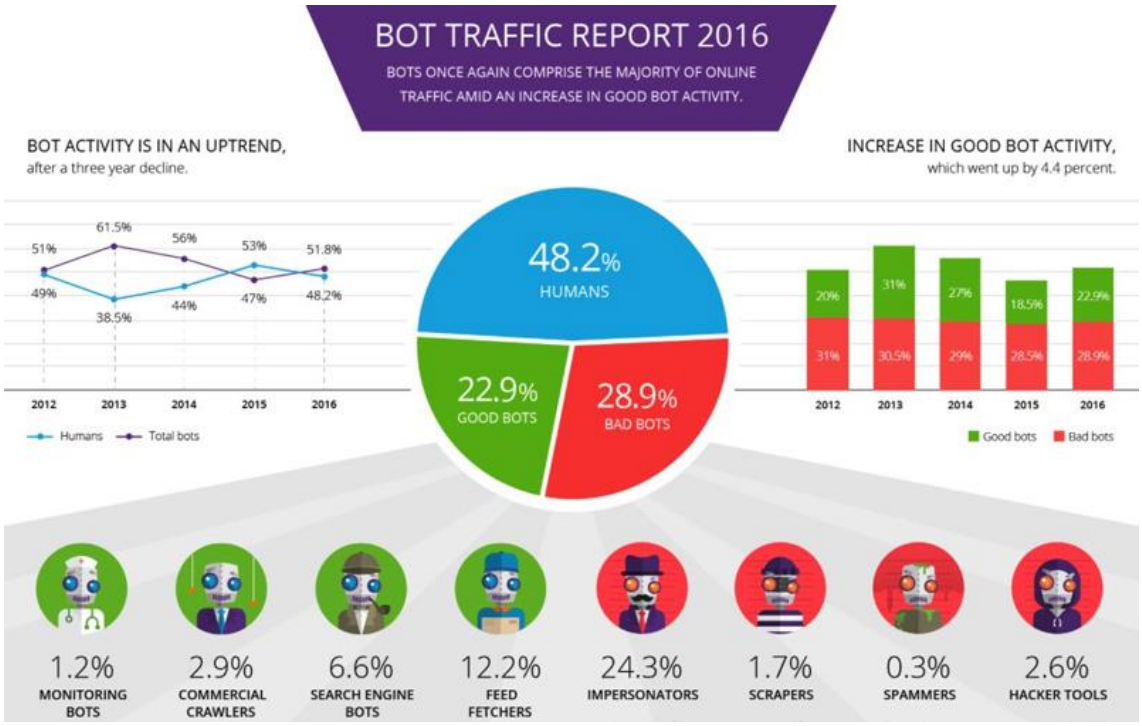
Pandas-kirjaston avulla tiedostot voidaan tallentaa ja avata useissa eri tiedostoformaateissa. Kirjasto tukee monenlaisia taulukkoformaatteja, jotka voidaan lukea dataframe- tai series-objekteiksi.

Verkkosivuvierailijoiden tietojen ollessa kunnossa siirryttiin yrityksen sähköpostituslistalle kirjautuneiden käyttäjien tietojen hankintaan. Nämä tiedot on ladattu Necunos Oy:n käyttämältä MailerLite-ohjelmistosta ja ne tallennettiin csv-tiedostoformaatissa, jonka jälkeen tiedostojen muokkaus toteutettiin pandas-kirjastoa hyödyntäen ja muokatut tiedostot tallennettiin csv-tiedostoformaatissa suoraan koneelle. Tiedot päätettiin tallentaa csv-tiedostoformaatissa, jotta niiden jatkokäsittely Microsoft Excel-ohjelmistolla olisi helpompaa.

### 3.2.2 Datan käsittely

Tietojen käsittely ja muokkaus toteutettiin Anacondan sisältämällä Jupyter Notebook -ohjelmistolla, Python-ohjelmointikielellä sekä Pythonin pandas-, BeautifulSoup- ja matplotlib-kirjastoja hyödyntämällä.

Hankittu data käsiteltiin ja siistittiin tulevaa jatkokäyttöä varten. Verkkosivuvierailijoiden tiedoista haluttiin poistaa kaikki tarpeeton tieto muistin säästämiseksi sekä datan jatkokäsittelyn selkeyttämiseksi. Lisäksi käyttäjien joukosta haluttiin karsia todennäköiset botit, eli tietokoneohjelmat, jotka on valjastettu suorittamaan yksinkertaisia tehtäviä omatoimisesti, ilman ihmisen myötävaikutusta [25]. Botteja on monenlaisia, ja ne kattavat noin puolet kaikista verkon käyttäjistä [26].




Kuva 5. Bottien osuus kaikesta verkkoliikenteestä. Haettu osoitteesta: https://ppcprotect.com/how-many-of-the-internets-users-are-robots/

Vaikka useimmat boteista on helppo tunnistaa, on edistyneempien bottien tunnistaminen erittäin hankalaa, sillä niiden käyttäytyminen muistuttaa hyvin paljon ihmisen käyttäytymistä. Botit ovat yleensä nopeita toiminnoissaan ja selaavat kaikki verkkotunnuksen alla olevat internetsivut samanaikaisesti, jonka vuoksi käyttäjä, joka suorittaa useita toimintoja hyvin lyhyessä ajassa tai lataa useita sivuja kerralla, on mitä suurimmalla todennäköisyydellä botti. Robots.txt-tiedoston avulla pystytään hallitsemaan verkkosivun bottiliikennettä. Botit lukevat tiedoston, joka kertoo niille, millä sivuilla ne saavat tai eivät saa vierailla. Tämän vuoksi kaikki verkkosivuvierailijat, jotka lataavat sivuston robots.txt-tiedoston ovat mitä suuremmalla todennäköisyydellä botteja. Osa boteista pystyy kuitenkin ohittamaan robots.txt-tiedoston, joten kaikkia botteja ei pysty aivan täysin senkään avulla karsimaan. [27.]

Matomo-analytiikkaohjelmisto osaa tunnistaa vain murto-osan boteista, jonka vuoksi todennäköisten bottien karsiminen verkkosivuvierailijoista on toteutettu poistamalla kaikkien käyttäjien tiedot, seuraavin perustein:

- Käyttäjät, joiden vierailun kesto on yli tunnin mittainen.

- Käyttäjät, jotka ovat vierailunsa aikana suorittaneet yli 30 toimintoa.
- Käyttäjät, jotka ovat ladanneet robots.txt-tiedoston.

Tämä ei tietenkään takaa, että kaikki botit saataisiin karsittua käyttäjätiedoista, mutta se vähentää riittävästi bottien vaikutusta lopullisiin tuloksiin.

Ajanjakso, jolta data on kerätty, käsittää yli 150 päivää, jonka vuoksi datan kerääminen ja käsittely haluttiin automatisoida. Tätä varten luotiin Python-ohjelmakoodi, joka ajettaessa hakee html-taulukkoina verkkosivuvierailijoiden päivittäiset tiedot sille annetuilta päivämääriltä ja tallentaa käsitellyt tiedot csv-tiedostoformaatissa koneelle.

Pandas-kirjaston read_html-toiminto toimii hieman eri tavalla kuin read_csv-toiminto. Jälkimmäinen lukee csv-tiedostot suoraan dataframe-objekteiksi, kun taas ensin mainittu lukee html-taulukot dataframe-objekteja sisältävään listaan.

Ohjelmakoodin alussa ladataan tarvittavat kirjastot ja määritellään tarvittavat muuttujat.

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
from datetime import timedelta, date
import pandas as pd

#Assign 'robots.txt' to robots for later use to check if visitor tried to
download robots.txt file from webpage

robots = ['robots.txt']
```

Esimerkkikoodi 2. Python-ohjelmakoodin osio, jossa ladataan tarvittavat kirjastot ja määritellään tarvittavat muuttujat.

Pythonin datetime-moduulia hyödyntämällä voidaan helpottaa päivämäärien ja aikojen käsittelyä ohjelmakoodissa. Kyseisen moduulin avulla toteutettiin daterange-funktio, joka käy sille annetun ajanjakson läpi päivä kerrallaan. Funktiolle annetaan parametreina aloitus- ja lopetuspäivä ’yyyy, m, d’-muodossa, jotka se käy läpi for-silmukassa. Aloitus- ja lopetuspäivän välinen ajanjakso muutetaan kyseisen ajanjakson välisten päivien lukumääräksi ja funktio palauttaa yield-toiminnolla päivän kerrallaan. Yield-toiminnon ja yleisemmin käytetyn return-toiminnon ero on siinä, että yield palauttaa generaattorin, joka tässä tapauksessa, aina daterange-funktiota kutsuttaessa, palauttaa päivämäärän kerrallaan aina aloituspäivästä lähtien niin kauan, kunnes lukumäärä ’n’ on kasvanut suuremmaksi kuin ajanjakson välisten päivien lukumäärä.

```
# Function to loop through given dates

def daterange(start_date, end_date):
     for n in range(int ((end_date - start_date).days)):
          yield start_date + timedelta(n)

# First date of daterange function

start_date = date(2018, 9, 23)

# End date (not included in the loop, last date will be previous date)

end_date = date(2019, 2, 27)
```

Esimerkkikoodi 3. Python-ohjelmakoodin daterange-funktio ja sille annettavat parametrit.

Daterange-funktiosta saatua päivämäärää hyödynnetään toisessa for-silmukassa, jossa se talletetaan single_date-muuttujaan. For-silmukka ajetaan niin kauan, kunnes kaikki daterange-funktiolle annetut päivämäärät on käyty läpi. Kolme tyhjää listaa alustetaan aina for-silmukan jälkeen, jotta niiden sisältö nollautuu joka kerralla, kun tämä osa ohjelmakoodista ajetaan.

```
# Iterate through given dates

for single_date in daterange(start_date, end_date):

     # Empty list to store urls on each iteration

     html_list = []


     # Empty list to store dataframes on each iteration

     visitors = []

     # Empty list to store multiple dataframes from single day on each itera-
     tion, if any

     concatenation = []
```

Esimerkkikoodi 4. Python-ohjelmakoodin osa, jossa for-silmukka iteroi kaikki daterange-funktiolle annetut päivämäärät sekä kolme lista-kokoelmatyyppiä.

Huomattiin, että niiden päivien osalta, jolloin verkkosivuvierailijoiden määrä nousi useiksi tuhansiksi, ei Matomo suostunut palauttamaan kokonaista html-taulukkoa vaan http response palautti virhearvon 500 (’Internal Server Error’), mikä yleensä viittaa siihen, että palvelimen toiminnassa on jonkinlainen tarkemmin määrittelemätön virhe. Tämä voi johtua monesta syystä, mutta tässä tapauksessa todennäköisin syy, myös yksi yleisimmistä syistä, on ’external resource timeout’ eli määriteltyä toimintoa ei pystytty toteuttamaan

siinä ajassa, mikä on palvelimelle määriteltynä. Ohjelmakoodiin lisättiin try-catch-lohko, joka tarkistaa, meneekö http-pyyntö annettuun url-osoitteeseen läpi. Siinä tapauksessa, että pyyntö ei mene läpi, muokataan url-osoitetta asettamalla sille ”filter_offset”- ja ”filter_limit”-parametrit. Kyseisten parametrien avulla voidaan url-osoitetta muokata niin, että se ei näytä kaikkia tietoja kerralla vaan hakee ainoastaan tuhat riviä taulukosta kerrallaan, kunnes kaikki taulukon tiedot on käyty läpi. Kaikki url-osoitteet tallennetaan html_list-nimiseen listaan string-merkkijonona. Python-kirjaston http.client-moduulin read toimintoa, joka palauttaa halutun http-vastauksen rakenteen sekä Pythonin len-funktiota, joka palauttaa sille annetun olion sisältämien merkkien lukumäärän, hyödyntämällä testattiin, kuinka monta merkkiä on tyhjässä taulukossa. Tästä saatiin tulokseksi useammalla testikerralla 106, joten url-osoitteen ”filter_offset”- ja ”filter_limit”-parametreja muokataan while-silmukassa niin kauan, kunnes viimeisin http-vastaus sisältää maksimissaan 106 merkkiä.

```
# Try-Catch to check if HTTP Response is 200 (Should be 200 or 500 in
this case)

try:
      # Url address to access data in Matomo software running on company’s
      server

      url = "http:// 192.168.2.10: 8888/index.php?date="+sin-
      gle_date.strftime("%Y-%m-%d")+"&expanded=1&filter_limit=-1&for-
      mat=HTML&idSite=2&method=Live.getLastVisitsDetails&module=API&pe-
      riod=day&token_auth=put token here"

      # Send HTTP Request to given url

      urlopen(url)

      # Store url as a string to html_list

      html_list.append(url)

      # If HTTP Response is not 200 (Unable to load all the visitor infor-
      mation at once, if too many visitors)
      # Load a maximum of 1000 rows (visitors) at a time

except:

      # Add filter_offset and filter_limit parameters to url to modify
      amount of rows
      # filter_offset = 0, starts from first row on a given date

      filter_offset = 0

      # filter_limit = 1000, number of rows

      filter_limit = 1000

      # Url address with modified parameters
```

```
url = "http://192.168.2.10:8888/index.php?date="+sin-
gle_date.strftime("%Y-%m-%d")+"&expanded=1&filter_limit="+str(fil-
ter_limit)+"&filter_offset="+str(filter_offset)+"&for-
mat=HTML&idSite=2&method=Live.getLastVisitsDetails&module=API&pe-
riod=day&token_auth=put token here"

# Store url to html_list

html_list.append(url)

# Continue adding urls to html_list until html table element is
empty

while True:
     # Increase filter_offset by 1000 to get the next 1000 rows

     filter_offset += filter_limit

     # Url address with modified parameters

     url = "http://192.168.2.10:8888/index.php?date="+sin-
     gle_date.strftime("%Y-%m-%d")+"&expanded=1&fil-
     ter_limit="+str(filter_limit)+"&filter_offset="+str(fil-
     ter_offset)+"&format=HTML&idSite=2&method=Live.getLastVis-
     itsDetails&module=API&period=day&token_auth=token here"

     # Store HTTP Response to html_response variable

     html_response = urlopen(url)

     # Check if HTTP Response includes only empty table and break
     while loop if true.

     if len(html_response.read()) <= 106:
          break

     # Store all the gathered urls in html_list

     html_list.append(url)
```

Esimerkkikoodi 5. Python-ohjelmakoodi, joka tarkistaa http-vastauksen try-catch-lohkossa ja tallentaa url-osoitteet html_list-listaan. Muokataan tarvittaessa url-osoitteen parametreja, kunnes kaikki tiedot kyseiseltä päivämäärältä on saatu.

Alun perin oli tarkoituksena tallentaa ainoastaan http-vastaukset listaan ja lukea Pythonin BeautifulSoup-kirjaston avulla taulukot kaikista http-vastauksista, mutta jostain syystä BeautifulSoup ei löytänyt taulukkoelementtejä suoraan vastauksista, jos niitä oli useampia. Ongelma ratkaistiin niin, että http-vastausten sijaan tallennettiin kaikki url-osoitteet listaan. Tämä lista iteroitiin for-silmukassa lähettämällä http-pyyntö kaikkiin listan sisältämiin url-osoitteisiin ja BeautifulSoup-kirjaston avulla eroteltiin kaikki html-taulukot, jotka url-osoitteet sisälsivät.

```
# Iterate through urls to scrape visitor information from each

for h in range(0,len(html_list)):
```

```
        #Send request to each url in html_list

        content = urlopen(html_list[h])

        # Read html document to BeautifulSoup object

        bs_obj = BeautifulSoup(content, 'lxml')

        # Find "table" tags from the object and store them to tables varia-
        ble

        tables = bs_obj.findAll('table')
```

Esimerkkikoodi 6. Python-ohjelmakoodin osio, jossa iteroidaan html_list-listan sisältämät url-osoitteet läpi ja talletetaan niistä löydetyt taulukot tables-muuttujaan.

Html-taulukkojen ollessa hankittuna iteroitiin for-silmukassa jokainen näistä taulukoista ja ne luettiin pandas-kirjaston pd.read_html-toiminnolla pandas-dataframe-objekteiksi, jotka talletettiin visitors-nimiseen listaan. Tämän jälkeen seuraava for-silmukka iteroi visitors-listan läpi, joka sisältää itsessään listan dataframe-objekteja. Jokaisesta listan alkiosta luodaan uusi dataframe-olio, jotta se voidaan käsitellä ja tallentaa csv-tiedostoformaatissa koneen kovalevylle. Dataframe-olioiden käsittelyssä poistetaan kaikki verkkosivuvierailijoiden tiedot, jotka ovat hyvin suurella todennäköisyydellä botteja, säilytetään myöhempää käyttöä varten vain tarpeelliset sarakkeet ja täydennetään puuttuvat arvot ”Unknown”-sanalla. Siinä tapauksessa, että samalta päivältä on useampi dataframe-olio, tallennetaan ne ”concatenation”-listaan ja yhdistetään yhdeksi dataframe-olioksi. Lopuksi muokatut dataframe-oliot tallennetaan koneen kovalevylle csv-tiedostoformaatissa nimeämällä tiedostot siten, että sanan ”Visitor” jälkeen tulee aina se päivämäärä, jolta tiedot ovat. Salausparametriksi annetaan utf-8, jotta Python pystyy lukemaan tiedostot ongelmitta jatkokäsittelyä silmällä pitäen.

```
        # Read each html table with pandas to create list of dataframes

        for table in tables:

             dataframe = pd.read_html(str(table))

             visitors.append(dataframe)

#Iterate through visitors list

for i in range(0,len(visitors)):

        #Create a pandas DataFrame from each list item

        visitorsdf = pd.DataFrame(visitors[i][0])
```

```
#Assign all rows containing robots.txt in actionDetails to bot vari-
able

bot = visitorsdf.actionDetails[visitorsdf.actionDetails.str.con-
tains('|'.join(robots), na=False)]

#Remove visitors who downloaded robots.txt file

Visit = visitorsdf[~visitorsdf.actionDetails.isin(bot)]

# Keep only useful columns

Visit = visitorsdf.filter(['idVisit', 'visitIp', 'serverDatePretty',
'siteName', 'visitorType', 'visitCount', 'daysSinceFirstVisit',
'visitDuration', 'actions', 'referrerType', 'referrerName', 'device-
Type', 'deviceBrand', 'operatingSystemName', 'browserName', 'coun-
try', 'countryCode', 'location'])

#Remove visitors identified as Bots

Visit = Visit.loc[Visit['operatingSystemName'] != 'Bot']

#Remove visitors with visit duration more than 1h as bots

Visit = Visit[(Visit['visitDuration'] < 3600)]

#Remove users with more than 30 actions as bots

Visit = Visit[(Visit['actions'] < 30)]

#Fill missing values of referrerName with Unknown

Visit['referrerName'].fillna('Unknown', inplace=True)

if len(visitors) > 1:
     concatenation.append(Visit)

     for i in range(0,len(concatenation)):

          Visit = Visit.append(concatenation[i])


#Save Datframe as csv file to given directory

Visit.to_csv("C:/Users/Lari/Documents/Visit/Visitors-" + sin-
gle_date.strftime("%Y-%m-%d") + '.csv', encoding='utf-8')
```

Esimerkkikoodi 7. Python-ohjelmakoodi, joka luo dataframe-olion päivittäisten verkkosivuvierailijoiden tiedosta, käsittelee tiedot sekä tallentaa ne csv-tiedostoformaatissa koneen kovalevylle.

### 3.2.3 Hankitun datan hyödyntäminen

Hankitusta ja esikäsitellystä datasta luotiin uudet dataframe-oliot, joista selvitettiin verkkosivuvierailijoiden ja yrityksen sähköpostilistalle kirjautuneiden lukumäärät kuukausittain, yleisimmät lähdesivustot sekä sähköpostilistalle kirjautuneiden käyttäjien aktiivi-

suus. Yrityksen sähköpostilistalle kirjautuneet jaettiin kuuteen eri ryhmään sillä perusteella, missä kuussa he ovat postituslistalle liittyneet. Postituslistalle kirjautuneiden tiedot ovat 14.09.2018 – 26.02.2019 väliseltä ajalta.

Päivittäisten verkkosivuvierailijoiden tiedot ladattiin Jupyter Notebook -sovellukseen ja ne yhdistettiin yhdeksi isoksi dataframe-olioksi. Sähköpostilistalle kirjautuneiden henkilöiden tiedot on hankittu suoraan yrityksen käyttämästä MailerLite-palvelusta 26.02.2019 ja ne luettiin suoraan dataframe-olioiksi csv-tiedostosta.

```
import pandas as pd
import glob

#Assign dir of files to visitlist
visitlist = glob.glob("C:/Users/Lari/Documents/Visit/Visitors*.csv")

#Create empty list to fill with dataframes
dflist = []

#Iterate through files in given dir
for file in visitlist:

    #Read files to pandas Dataframes
    df = pd.read_csv(file, parse_dates=True, index_col='serverDatePretty')

    #Fill dflist with Dataframes
    dflist.append(df)

#Concatenate all Dataframes to single Dataframe with visiting date as an index
All_Visitors = pd.concat(dflist, ignore_index=False)

# Read given file to dataframe
Active_Subscribers = pd.read_csv('C:/Users/Lari/Documents/Necunos Visit
Data/Signed Visitors/Data (1)/Active_Subscribers_26_02_2019.csv')
```

Esimerkkikoodi 8. Python-ohjelmakoodi, joka hakee kaikki Visitor-alkuiset csv-tiedostot annetusta kansiosta ja luo niistä yhden ison dataframe-olion, jonka indeksi on vierailun päivämäärä. Lukee myös sähköpostilistalle kirjautuneiden tiedot dataframe-olioksi.

Dataframeille annettiin nimet All_Visitors ja Active_Subscribers, joista molemmista luotiin uusi dataframe-olio muokkaamalla niitä siten, että jokaiselta päivältä otettiin verkkosivuvierailijoiden lukumäärä sekä sähköpostilistalle kirjautuneiden lukumäärä. Active_Subscribers-dataframen lisättiin datetime-formaatissa oleva Date-sarake, josta tehtiin kyseisen dataframen indeksi. Muokatusta dataframesta luotiin uusi dataframe, Subscriber_Count, joka sisältää ainoastaan postituslistalle kirjautuneiden lukumäärän jokaiselta päivältä. Tämän jälkeen luotiin Visitor_Count dataframe, joka sisältää päivittäisten verkkosivuvierailijoiden lukumäärät. Lopullinen Visitor_and_Subscribers-dataframe

muodostettiin yhdistämällä Subscriber_Count- ja Visitor_Count-dataframet. Pandas-kirjasto lisää automaattisesti NaN-merkinnän puuttuvien tietojen kohdalle, joten nämä muutettiin nolliksi fillna-toiminnon avulla. Vierailijoiden- ja postituslistalle kirjautuneiden lukumäärät sisältävät sarakkeet nimettiin kuvaavammiksi ja viimeiseen sarakkeeseen laskettiin päiväkohtaisesti postituslistalle kirjautuneiden suhde kaikkiin päivittäisiin verkkosivuvierailijoihin. Tämän jälkeen muutettiin puuttuvat arvot nolliksi ja valmis dataframe tallennettiin csv-tiedostoformaatissa koneelle.

```
# Create column: 'Date' by changing 'Subscribed' column to datetime format.
Active_Subscribers['Date'] = pd.to_datetime(Active_Subscribers['Sub-
scribed']).dt.date

# Set Date as index.
Active_Subscribers.set_index('Date', inplace=True)

# Get number of daily subscribers
Subscriber_Count = Active_Subscribers.groupby(Active_Subscribers.in-
dex).agg({'Email' : 'count'})

# Get number of daily visitors
Visitor_Count = All_Visitors.groupby(level=0).agg({'Unnamed: 0' : 'count'})

# Create new dataframe from concatenation of two dataframes
Visitor_and_Subscribers = pd.concat([Visitor_Count,Subscriber_Count], ig-
nore_index=True, axis=1)

# Replace missing values with zeros
Visitor_and_Subscribers.fillna(0)

# Rename first column
Visitor_and_Subscribers = Visitor_and_Subscribers.rename(columns={0: 'Number
Of Visitors'})

# Rename second column
Visitor_and_Subscribers = Visitor_and_Subscribers.rename(columns={1: 'Number
Of Subscribers'})

# Create third column by computing the ratio of first and second column.
Visitor_and_Subscribers['Visitor Subscriber Ratio'] = Visitor_and_Subscrib-
ers['Number Of Subscribers'] / Visitor_and_Subscribers['Number Of Visitors']

# Replace missing values with zeros.
Visitor_and_Subscribers = Visitor_and_Subscribers.fillna(0)

# Save dataframe in csv-format.
Visitor_and_Subscribers.to_csv("C:/Users/Lari/Documents/Necunos Visit
Data/Visitor Subscriber Ratio.csv", encoding='utf-8')
```

Esimerkkikoodi 9. Komennot, joilla kahdesta erillisestä csv-tiedostosta luoduista dataframe-objekteista haetaan verkkosivuvierailijat ja sähköpostilistalle kirjautuneet, jokaiselta päivältä. Luodaan uusi dataframe-olio, johon tiedot yhdistetään ja lasketaan vierailijoiden sekä sähköpostilistalle kirjautuneiden suhde.

Lopullinen dataframe sisältää 152 riviä, joten tilan säästämisesi kuvassa 6 on esitetty sama dataframe pandas-kirjaston resample-toimintoa käyttämällä, joka näyttää tiedot kahdeksan päivän periodeissa.

```
In [96]: 1 VSR = Visitor_and_Subscribers.resample('8D').sum()
```

Out[96]:

| | Number Of Visitors | Number Of Subscribers | Visitor Subscriber Ratio |
|---|---|---|---|
| **2018-09-23** | 3289 | 36.0 | 0.095248 |
| **2018-10-01** | 2241 | 24.0 | 0.096113 |
| **2018-10-09** | 986 | 14.0 | 0.123969 |
| **2018-10-17** | 1015 | 13.0 | 0.090836 |
| **2018-10-25** | 652 | 6.0 | 0.076228 |
| **2018-11-02** | 947 | 10.0 | 0.102630 |
| **2018-11-10** | 2132 | 14.0 | 0.021914 |
| **2018-11-18** | 837 | 5.0 | 0.047090 |
| **2018-11-26** | 19826 | 750.0 | 0.213457 |
| **2018-12-04** | 10937 | 404.0 | 0.294923 |
| **2018-12-12** | 4463 | 136.0 | 0.244493 |
| **2018-12-20** | 3602 | 29.0 | 0.062314 |
| **2018-12-28** | 20863 | 32.0 | 0.036938 |
| **2019-01-05** | 8366 | 15.0 | 0.020142 |
| **2019-01-13** | 4516 | 6.0 | 0.011542 |
| **2019-01-21** | 3954 | 9.0 | 0.018179 |
| **2019-01-29** | 4844 | 15.0 | 0.028068 |
| **2019-02-06** | 3655 | 25.0 | 0.058108 |
| **2019-02-14** | 2983 | 19.0 | 0.050810 |

Kuva 6. Pandas-kirjaston resample-toiminnon avulla muokattu tuloste luodusta dataframe-oliosta.

Necunos Oy on luonut Facebook-tilin 10.12.2018 ja Twitter-tilin jo vuonna 2017. Tästä syystä haluttiin selvittää, mistä sivulähteistä verkkosivuvierailijat tulevat ja kuinka suuri on sosiaalisen median sekä hakukoneiden osuus. Verkkosivuvierailijoiden tiedoista löytyvät sarakkeet ’referrerType’ ja ’referrerName’, jotka kertovat lähdesivuston tyypin ja nimen. Näistä tiedoista luotiin useampikin taulukko, jotka sisälsivät kunkin lähdesivustotyypin ja lähdesivuston suuntaamien verkkosivuvierailijoiden lukumäärät, eri lähdesivustotyyppien osuuden kaikista verkkovierailijoista ja yleisimpien sosiaalisen median tai hakukoneiden osuuden omassa lähdesivustotyypissään. Lähdesivustotyyppejä on neljä erilaista: direct, search, website ja social. Näistä laskettiin kunkin lähdesivustotyypin osuus päivittäisistä vierailijoista Pythonilla pandas-kirjastoa hyödyntäen.

Alussa poistettiin kaikki välilyönnit sarakkeista ja luotiin 4 pandas-kirjaston series-oliota jokaista lähdesivustotyyppiä kohden siten, että haettiin referrerType-sarakkeesta kaikki

rivit, jotka sisältävät saman lähdesivustotyypin arvon. Indeksinä taulukossa on serverDate, joka on datetime-formaatissa. Sen avulla hyödynnettiin pandasin resample-metodia parametrilla: D, eli päivä ja ketjutettiin loppuun sum-komento, jolloin lopputuloksena saatiin jokaisen päivän osalta verkkosivuvierailijoiden lukumäärä. Tämän jälkeen series-oliot muutettiin dataframe-objekteiksi ja jokaisen taulukon sarakkeen nimi vaihdettiin kuvaavaksi ennen taulukoiden yhdistämistä. Lopuksi taulukot yhdistettiin ja luotiin vielä sarake, jossa kaikki päivittäiset vierailijat, jonka avulla laskettiin kunkin lähdesivustotyypin osuus kaikista päivittäisistä vierailuista omiin sarakkeihinsa. Lopputulos tallennettiin tietokoneen kovalevylle csv-tiedostoformaatissa.

```
# Remove whitespace between columns
All_Visitors.columns = All_Visitors.columns.str.strip()

# Create 4 different series objects: social, search, website, direct
social = All_Visitors['referrerType'].str.contains('social')
search = All_Visitors['referrerType'].str.contains('search')
website = All_Visitors['referrerType'].str.contains('website')
direct = All_Visitors['referrerType'].str.contains('direct')

# Modify series to show number of daily visitors
social = social.resample('D').sum()
search = search.resample('D').sum()
website = website.resample('D').sum()
direct = direct.resample('D').sum()

# Create dataframe objects
social = social.to_frame()
search = search.to_frame()
website = website.to_frame()
direct = direct.to_frame()

# Rename columns
social = social.rename(columns={'referrerType' : 'Visitors via Social Media'})
search = search.rename(columns={'referrerType' : 'Visitors via SearchEn-
gines'})
website = website.rename(columns={'referrerType' : 'Visitors via Other Web-
sites'})
direct = direct.rename(columns={'referrerType' : 'Direct Visitors'})

# Merge all dataframes to one
dailyreftypes = social.join([search, website, direct], how='outer')

# Create new column which shows total number of daily visitors
dailyreftypes['Number of Visitors'] = dailyreftypes.sum(axis=1)

# Create columns for percentage of visitors by each referrer type
dailyreftypes['Percentage of Visitors via Social Media'] = dailyreftypes['Vis-
itors via Social Media'] / dailyreftypes['Number of Visitors'] * 100
dailyreftypes['Percentage of Visitors via SearchEngines'] = dai-
lyreftypes['Visitors via SearchEngines'] / dailyreftypes['Number of Visitors']
* 100
dailyreftypes['Percentage of Visitors via Other Websites'] = dai-
lyreftypes['Visitors via Other Websites'] / dailyreftypes['Number of Visi-
tors'] * 100
```

```
dailyreftypes['Percentage of Direct Visitors'] = dailyreftypes['Direct Visi-
tors'] / dailyreftypes['Number of Visitors'] * 100

# Save file in csv format
dailyreftypes.to_csv('C:/Users/Lari/Documents/Necunos Visit Data/Signed Visi-
tors/DailyReferrerTypesRatio.csv', encoding='utf-8')
```

Esimerkkikoodi 10. Python-ohjelmakoodin komennot erilaisten lähdesivustotyyppien osuuksien laskemiseen verkkosivuvierailijoihin nähden.

```
df = pd.read_csv('C:/Users/Lari/Documents/Necunos Visit Data/Signed Visitors/DailyReferrerTypesRatio.csv')
df.head(10)
```

ut[7]:

| | serverDatePretty | Visitors via Social Media | Visitors via SearchEngines | Visitors via Other Websites | Direct Visitors | Number of Visitors | Percentage of Visitors via Social Media | Percentage of Visitors via SearchEngines | Percentage of Visitors via Other Websites | Percentage of Direct Visitors |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2018-09-23 | 0.0 | 0.0 | 0.0 | 241.0 | 241.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 1 | 2018-09-24 | 0.0 | 0.0 | 0.0 | 393.0 | 393.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 2 | 2018-09-25 | 0.0 | 0.0 | 0.0 | 560.0 | 560.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 3 | 2018-09-26 | 0.0 | 0.0 | 0.0 | 286.0 | 286.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 4 | 2018-09-27 | 0.0 | 0.0 | 0.0 | 795.0 | 795.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 5 | 2018-09-28 | 0.0 | 0.0 | 0.0 | 381.0 | 381.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 6 | 2018-09-29 | 0.0 | 0.0 | 0.0 | 570.0 | 570.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 7 | 2018-09-30 | 0.0 | 0.0 | 0.0 | 63.0 | 63.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 8 | 2018-10-01 | 0.0 | 0.0 | 0.0 | 312.0 | 312.0 | 0.0 | 0.0 | 0.0 | 100.0 |
| 9 | 2018-10-02 | 0.0 | 0.0 | 0.0 | 169.0 | 169.0 | 0.0 | 0.0 | 0.0 | 100.0 |

Kuva 7. Kuvakaappaus Jupyter Notebookista, jossa edellä luodun dataframe-olion ensimmäiset 10 riviä tulosteena.

Samalla tavalla eroteltiin ’referrerName’-sarakkeesta haluttujen sosiaalisen median ja hakukoneiden osuus verkkosivuvierailijoista. Näistä luotiin oma taulukkonsa, johon laskettiin sosiaalisen median osalta Twitterin, Facebookin sekä Redditin kautta tulleet verkkosivuvierailijat ja hakukoneiden osalta Googlen sekä muiden hakukoneiden osuus verkkosivuvierailijoista. Samalla lisättiin muista verkkosivuista sekä ilman lähdesivustoa käyneiden verkkosivuvierailijoiden määrät. Lopuksi kaikki nämä dataframe-oliot yhdistettiin yhdeksi isoksi dataframe-olioksi ja tallennettiin csv-tiedostoformaatissa koneelle.

```
# Create dataframe with one column as visitors via google

google = All_Visitors['referrerName'].str.contains('Google')
google_ref = google.resample('D').sum()
google_ref = google_ref.to_frame()
google_ref = google_ref.rename(columns={'referrerName' : 'Visitors via Goog-
le'})

# Create similar dataframe with one column as visitors via all other search
engines

search = (All_Visitors['referrerType'].str.contains('search')) & (~All_Visi-
tors['referrerName'].str.contains('Google'))
search_ref = search.resample('D').sum()
search_ref = search_ref.to_frame()
```

```
search_ref = search_ref.rename(columns={0 : 'Visitors via Other Search En-
gines'})

# Merge all dataframes together

daily_refs = daily_social_referrers.join([dir_ref, web_ref, search_ref,
google_ref], how='outer')

# Save file in csv-format

daily_refs.to_csv('C:/Users/Lari/Documents/Necunos Visit Data/Signed Visi-
tors/DailyRefs.csv', encoding='utf-8')
```

Esimerkkikoodi 11. Python-ohjelmakoodin komennot, joiden avulla eri lähdesivustojen kautta tulleiden verkkosivuvierailijoiden lukumäärät eritellään taulukkoon. Samalla tavalla toteutettu muut lopulliseen dataframe-olioon liitetyt taulukot, joissa on muutettu vain ’referrerName’ sarakkeesta haettavaa arvoa.

```
In [52]: 1 weekly_refs = daily_refs.resample('7D').sum()
         2 weekly_refs.head(10)
```

Out[52]:

| serverDatePretty | Visitors via Twitter | Visitors via Facebook | Visitors via Reddit | Visitors via Other Social Media | Direct Visitors | Visitors via Other Websites | Visitors via Other Search Engines | Visitors via Google |
|---|---|---|---|---|---|---|---|---|
| 2018-09-23 | 0.0 | 0.0 | 0.0 | 0.0 | 3226.0 | 0.0 | 0.0 | 0.0 |
| 2018-09-30 | 0.0 | 0.0 | 0.0 | 0.0 | 1997.0 | 0.0 | 0.0 | 0.0 |
| 2018-10-07 | 21.0 | 0.0 | 27.0 | 0.0 | 851.0 | 19.0 | 5.0 | 9.0 |
| 2018-10-14 | 10.0 | 2.0 | 113.0 | 0.0 | 782.0 | 40.0 | 14.0 | 24.0 |
| 2018-10-21 | 21.0 | 0.0 | 57.0 | 3.0 | 526.0 | 19.0 | 12.0 | 18.0 |
| 2018-10-28 | 19.0 | 1.0 | 13.0 | 2.0 | 654.0 | 13.0 | 11.0 | 14.0 |
| 2018-11-04 | 35.0 | 3.0 | 13.0 | 7.0 | 602.0 | 10.0 | 8.0 | 24.0 |
| 2018-11-11 | 84.0 | 5.0 | 282.0 | 6.0 | 1595.0 | 31.0 | 10.0 | 28.0 |
| 2018-11-18 | 26.0 | 4.0 | 49.0 | 6.0 | 615.0 | 15.0 | 8.0 | 20.0 |
| 2018-11-25 | 150.0 | 60.0 | 288.0 | 143.0 | 7240.0 | 5928.0 | 154.0 | 1011.0 |

Kuva 8. Kuvakaappaus Jupyter Notebookista. Tulosteena 10 ensimmäistä riviä edellä luodusta taulukosta lasketun viikoittaisten kävijämäärien osuus uutena taulukkona.

Necunos Oy:n sähköpostilistalle kirjautuneiden henkilöiden tiedot on hankittu suoraan yrityksen käyttämän MailerLite-palvelun kautta. Ne on haettu 26.02.2019 ja tallennettu koneelle csv-tiedostoformaatissa. Tiedosto luettiin pandas-kirjaston avulla dataframe-olioksi ja siihen lisättiin ’Month’-niminen sarake, joka sai arvonsa taulukon indeksinä olleesta kirjautumispäivämäärän kuukaudesta. Eli jos käyttäjä on kirjautunut syyskuussa, niin arvo on 9. Tämän sarakkeen perusteella postituslistalle kirjautuneet henkilöt jaettiin kuuteen ryhmään sen perusteella, missä kuussa he ovat listalle kirjautuneet.

```
import pandas as pd

# Read file in to pandas dataframe
Active_Subscribers = pd.read_csv('C:/Users/Lari/Documents/Necunos Visit
Data/Signed Visitors/Data (1)/Active_Subscribers_26_02_2019.csv')

# Change index
Active_Subscribers = Active_Subscribers.set_index('Subscribed')
```

```
# Convert index to normal datetime format (YYYY-MM-DD)
Active_Subscribers.index = pd.DatetimeIndex(Active_Subscribers.index).normal-
ize()

# Replace missing values as Unknown
Active_Subscribers = Active_Subscribers.fillna('Unknown')

# Create new column: Month
Active_Subscribers['Month'] = Active_Subscribers.index.month

# Create new dataframes based on the month when visitor subscribed
Subscribers_Sep = Active_Subscribers.loc[Active_Subscribers['Month'] == 9]

Subscribers_Oct = Active_Subscribers.loc[Active_Subscribers['Month'] == 10]

Subscribers_Nov = Active_Subscribers.loc[Active_Subscribers['Month'] == 11]

Subscribers_Dec = Active_Subscribers.loc[Active_Subscribers['Month'] == 12]

Subscribers_Jan = Active_Subscribers.loc[Active_Subscribers['Month'] == 1]

Subscribers_Feb = Active_Subscribers.loc[Active_Subscribers['Month'] == 2]
```

Esimerkkikoodi 12. Python-ohjelmakoodin komennot, joilla luetaan csv-taulukko dataframe-olioksi, muokataan sitä edellä esitetyllä tavalla ja luodaan uudet dataframe-oliot, jotka sisältävät samat tiedot aina tietyn kuukauden ajalta.

Seuraavaksi luotiin muutama funktio, joiden avulla lisätiin taulukoihin neljä uutta saraketta. Ensiksi laskettiin käyttäjäkohtaisesti avattujen sähköpostien ja lähetettyjen sähköpostien suhde sekä avattujen sähköpostien, joissa käyttäjä on myös klikannut linkkiä, suhde lähetettyihin sähköposteihin. Tästä syystä käyttäjät on jaettu ryhmiin kirjautumiskuukauden perusteella, sillä luonnollisesti syyskuussa kirjautuneet ovat saaneet enemmän postia kuin helmikuussa kirjautuneet. Tämän jälkeen luotiin vielä ”Interest_rate”- ja ”Engagement_rate” -sarakkeet, joiden arvo määriteltiin itse asteikolla 1-4 sen mukaan, mikä on käyttäjän avattujen ja lähetettyjen sähköpostien suhde (Interest) sekä klikattujen ja lähetettyjen sähköpostien suhde (Engagement). Ajoittain pandas-kirjasto ei hyväksy ketjutettuja komentoja, ja se varoittaa, että dataframe-olio on kopio slice-oliosta, eikä hyväksy syntaksia. Varoitus on useasti hieman turha, sillä monissa tapauksissa oikeaa syntaksia käyttämällä, ei välttämättä päästä haluttuun lopputulokseen, jolloin varoituksen voi ottaa pois päältä. [28.]

```
# Function for creating two new columns to given dataframe.
def ratio_columns(dataframe):
      dataframe['Opened_to_sent_ratio'] = dataframe['Opened'] / data-
      frame['Emails sent'] * 100

      dataframe['Clicked_to_sent_ratio'] = dataframe['Clicked'] / data-
      frame['Emails sent'] * 100
```

```
      dataframe[‘CTOR’] = dataframe['Clicked'] / dataframe['Opened'] * 100

# Set off SettingWithCopyWarning.
pd.options.mode.chained_assignment = None  # default='warn'

# Apply function to all given dataframes.
ratio_columns(Subscribers_Sep)
ratio_columns(Subscribers_Oct)
ratio_columns(Subscribers_Nov)
ratio_columns(Subscribers_Dec)
ratio_columns(Subscribers_Jan)
ratio_columns(Subscribers_Feb)

# Import nympy library.
import numpy as np

# Function for calculating interest value for all subscribers.
def interest(opened_to_sent_ratio_value):
      if opened_to_sent_ratio_value >= 0.75:
         return 1

      elif (opened_to_sent_ratio_value < 0.75) & (opened_to_sent_ratio_value >=
      0.5):
         return 2

      elif (opened_to_sent_ratio_value < 0.5) & (opened_to_sent_ratio_value >=
      0.25):
         return 3

      elif (opened_to_sent_ratio_value < 0.25) & (opened_to_sent_ratio_value >=
      0):
         return 4

      else :
         return np.nan

# Function for calculating engagement value for all subscribers.
def engagement(clicked_to_sent_ratio_value):
      if clicked_to_sent_ratio_value >= 0.75:
         return 1

      elif (clicked_to_sent_ratio_value < 0.75) & (clicked_to_sent_ratio_value
      >= 0.5):
         return 2

      elif (clicked_to_sent_ratio_value < 0.5) & (clicked_to_sent_ratio_value
      >= 0.25):
         return 3

      elif (clicked_to_sent_ratio_value < 0.25) & (clicked_to_sent_ratio_value
      >= 0):
         return 4

      else :
         return np.nan

# Apply interest function to all given dataframes.
Subscribers_Sep['Interest_Class'] = Subscribers_Sep.Opened_to_sent_ratio.ap-
ply(interest)
Subscribers_Oct['Interest_Class'] = Subscribers_Oct.Opened_to_sent_ratio.ap-
ply(interest)
Subscribers_Nov['Interest_Class'] = Subscribers_Nov.Opened_to_sent_ratio.ap-
ply(interest)
```

```
Subscribers_Dec['Interest_Class'] = Subscribers_Dec.Opened_to_sent_ratio.ap-
ply(interest)
Subscribers_Jan['Interest_Class'] = Subscribers_Jan.Opened_to_sent_ratio.ap-
ply(interest)
Subscribers_Feb['Interest_Class'] = Subscribers_Feb.Opened_to_sent_ratio.ap-
ply(interest)

# Apply engagement function to all given dataframes.
Subscribers_Sep['Engagement_Class'] = Subscribers_Sep.Clicked_to_sent_ra-
tio.apply(engagement)
Subscribers_Oct['Engagement_Class'] = Subscribers_Oct.Clicked_to_sent_ra-
tio.apply(engagement)
Subscribers_Nov['Engagement_Class'] = Subscribers_Nov.Clicked_to_sent_ra-
tio.apply(engagement)
Subscribers_Dec['Engagement_Class'] = Subscribers_Dec.Clicked_to_sent_ra-
tio.apply(engagement)
Subscribers_Jan['Engagement_Class'] = Subscribers_Jan.Clicked_to_sent_ra-
tio.apply(engagement)
Subscribers_Feb['Engagement_Class'] = Subscribers_Feb.Clicked_to_sent_ra-
tio.apply(engagement)

# Save dataframes in csv-format.
Subscribers_Sep.to_csv('C:/Users/Lari/Documents/Necunos Visit Data/Subscrib-
ers_Sep.csv', encoding='utf-8')
Subscribers_Oct.to_csv('C:/Users/Lari/Documents/Necunos Visit Data/Subscrib-
ers_Oct.csv', encoding='utf-8')
Subscribers_Nov.to_csv('C:/Users/Lari/Documents/Necunos Visit Data/Subscrib-
ers_Nov.csv', encoding='utf-8')
Subscribers_Dec.to_csv('C:/Users/Lari/Documents/Necunos Visit Data/Subscrib-
ers_Dec.csv', encoding='utf-8')
Subscribers_Jan.to_csv('C:/Users/Lari/Documents/Necunos Visit Data/Subscrib-
ers_Jan.csv', encoding='utf-8')
Subscribers_Feb.to_csv('C:/Users/Lari/Documents/Necunos Visit Data/Subscrib-
ers_Feb.csv', encoding='utf-8')
```

Esimerkkikoodi 13. Python-ohjelmakoodilla luodut funktiot ja niiden käyttäminen dataframe-olioiden muokkaamisessa sekä tallennus csv-tiedostoformaatissa.

```
In [10]: 1 Subscribers_Sep.head(10)
```

Out[10]:

| | Signup Timestamp | Confirmation IP | Confirmation Timestamp | Month | Opened_to_sent_ratio | Clicked_to_sent_ratio | Interest_Class | Engagement_Class |
|---|---|---|---|---|---|---|---|---|
| b27:b97a | 2018-09-30 17:21 | 2600:1700:8f01:1d4f:f56f:e18d:db27:b97a | 2018-09-30 17:21 | 9 | 0.000000 | 0.000000 | 4 | 4 |
| .6.226.59 | 2018-09-29 15:28 | 46.6.226.59 | 2018-09-29 15:28 | 9 | 0.636364 | 0.636364 | 2 | 2 |
| 40.208.73 | 2018-09-29 13:26 | 68.40.208.73 | 2018-09-29 13:26 | 9 | 0.454545 | 0.090909 | 3 | 4 |
| 247.83.12 | 2018-09-29 4:09 | 146.247.83.12 | 2018-09-29 4:09 | 9 | 1.000000 | 0.454545 | 1 | 3 |
| 229.7.167 | 2018-09-28 19:33 | 80.229.7.167 | 2018-09-28 19:33 | 9 | 0.818182 | 0.363636 | 1 | 3 |
| 1.114.150 | 2018-09-28 18:16 | 82.221.114.150 | 2018-09-28 18:16 | 9 | 0.272727 | 0.000000 | 3 | 4 |
| 78.233.81 | 2018-09-28 11:02 | 101.78.233.81 | 2018-09-28 11:02 | 9 | 0.583333 | 0.083333 | 2 | 4 |
| 43.151.60 | 2018-09-28 6:58 | 77.243.151.60 | 2018-09-28 6:58 | 9 | 0.000000 | 0.000000 | 4 | 4 |
| 2ea:31b5 | 2018-09-27 18:00 | 2600:8805:9000:43:dc62:f178:c2ea:31b5 | 2018-09-27 18:00 | 9 | 0.500000 | 0.500000 | 2 | 2 |
| .208.224 | 2018-09-27 15:56 | 46.165.208.224 | 2018-09-27 15:56 | 9 | 0.666667 | 0.083333 | 2 | 4 |

Kuva 9. Kuvakaappaus Jupyter Notebookista, jossa näkyy 10 ensimmäistä riviä ja uudet sarakkeet yhdestä edellä luodusta dataframe-oliosta.

### 3.2.4 Datan jatkokäsittely ja tulokset

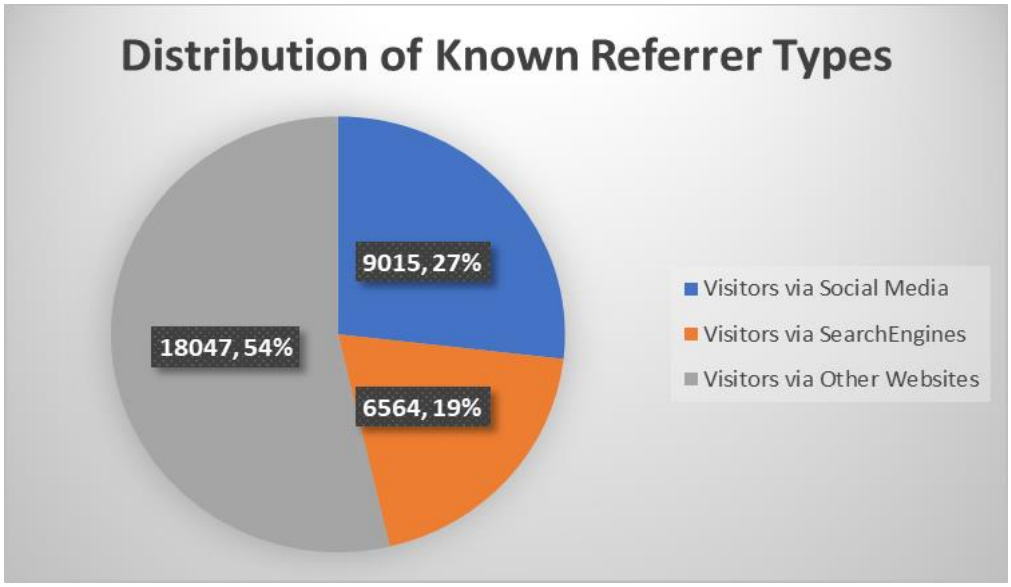
Datan ollessa kerättynä ja muokattuna analyysia varten ryhdyttiin sitä peilaamaan luvussa 2 esitettyjen teorioiden ja kuvassa 1 esitetyn suppilon kanssa. Analyysissa tarkastellaan nettisivuvierailijoiden, sähköpostilistalle kirjautuneiden henkilöiden ja näiden suhteen muutosta mitattuna ajanjaksona sekä siihen mahdollisesti vaikuttaneita tekijöitä, erityisesti sosiaalista mediaa ja muita kampanjoita. Tällä keinolla pyritään tarkastelemaan, minkälaiset asiat ovat vaikuttaneet käyttäjien haalimiseen ja kuinka hyvin verkkosivuvierailijat ollaan saatu kiinnostuneiksi yrityksestä sillä tasolla, että he ovat kirjautuneet sähköpostilistalle. Analyysin tärkein osa on sitoutuneet käyttäjät, eli tässä tapauksessa niiden sähköpostilistalle kirjautuneiden käyttäjien osuus, jotka ovat aktiivisesti avanneet ja lukeneet heille lähetetyt sähköpostit. Tarkastelussa otetaan huomioon myös sähköpostilistalle kirjautuneiden kiinnostus yritykseen kokonaisuudessaan, jota arvioidaan CTR-, EOR- ja CTOR-arvoilla.

Verkkosivuvierailijoiden lähdesivustojen tyyppejä tarkastellessa huomattiin, että aivan liian suuri osa oli ”direct”-tyyppisiä, eli verkkosivuvierailija on joko kirjoittanut osoitteen suoraan selaimen osoitekenttään tai käyttänyt siitä luotua kirjanmerkkiä. Tämä johtuu siitä, että kaikki liikenne, mitä ei tunnisteta joksikin muuksi, luokitellaan ”direct”-tyyppiseksi. Syitä tähän on monia ja todennäköisesti reilusti yli puolet näistä vierailuista on

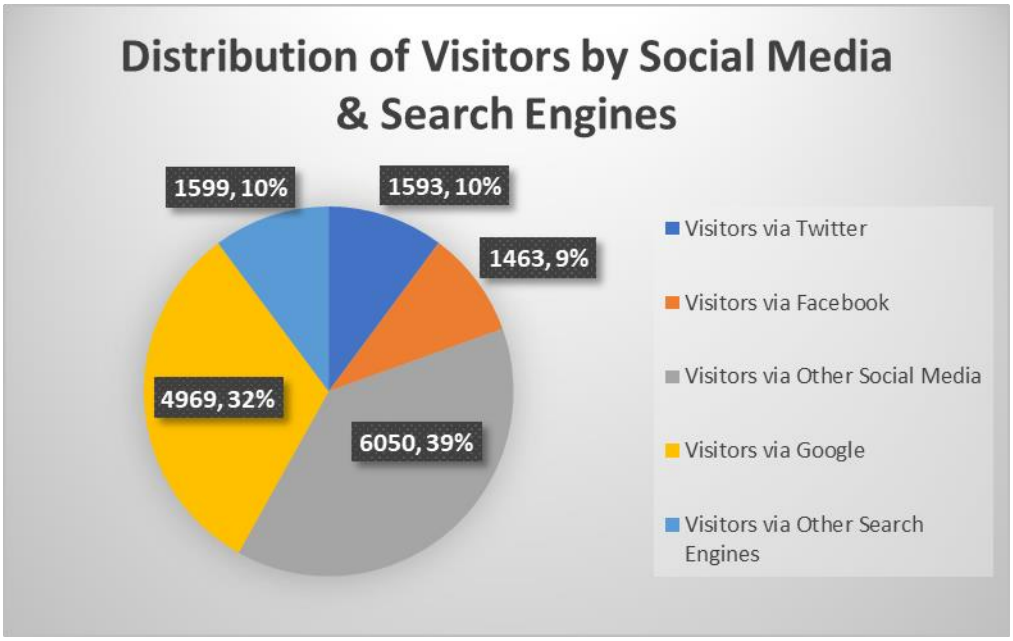
oikeasti jonkin toisen lähdesivustotyypin kautta tulleita. Vaikuttavia tekijöitä ovat muun muassa. [29.]

- selainlaajennus, joka piilottaa käyttäjän verkkoliikenteen tiedot
- ”pimeät linkit”, jotka on jaettu sähköpostin tai chat-keskustelun kautta
- muut tekijät, joiden takia palvelu ei tunnista oikeaa lähdesivustotyyppiä.

Tämän vuoksi sosiaalisen median ja uutisoinnin vaikutusta verkkosivuvierailijoihin ei pystytä kovinkaan tarkasti analysoimaan, sillä mitä todennäköisimmin useat näistä lähdesivuista tulleet verkkosivuvierailijat on luokiteltu virheellisesti ”direct”-tyyppisiksi. Tästä huolimatta päätettiin tarkastella tunnistettujen lähdesivustotyyppien osuutta verkkosivuvierailuissa sekä purkaa hieman näitä alaluokkiin ja tarkastella yksittäisten sosiaalisten medioiden ja hakukoneiden osuutta omissa luokissaan, jotka on esitetty alla olevissa ympyräkaavioissa.




Kuva 10. Microsoft Excelillä luotu ympyräkaavio, jossa näkyy valittujen lähdesivustotyyppien osuus kaikista verkkosivuvierailuista.




Kuva 11. Microsoft Excelillä luotu ympyräkaavio, jossa näkyy valittujen lähdesivustojen osuus kaikista verkkosivuvierailuista.

Yrityksen verkkosivuvierailut, sähköpostilistalle kirjautuneet ja näiden suhde otettiin kuukausittaiseen tarkasteluun, jossa katsottiin jokaisen kuukauden osalta päivittäiset mediaani arvot kyseisten tietojen osalta. Mediaaniarvo on valittu keskiarvon (mean) sijaan siitä syystä, että data ei ole normaalisti jakautunutta, jolloin mediaani on parempi vaihtoehto keskiarvon määrittämiseen. Kun nämä tiedot sisältävän taulukon indeksi oli muokattu ’datetime’-muotoon, pystyttiin helposti pandas-kirjaston loc-komennolla luomaan uusi taulukko jokaiselle kuukaudelle.

```
Vis_Sep = Vis.loc['2018-Sep']
```

Esimerkkikoodi 14. Pandas-kirjaston loc-komennon hyödyntäminen.

Tämän jälkeen haettiin jokaisen kuukauden osalta tiedot pandas-kirjaston ’describe’-komennolla.

```
1 Vis_Sep.describe()
```

| | Number Of Visitors | Number Of Subscribers | Visitor Subscriber Ratio |
|---|---|---|---|
| count | 8.000000 | 8.000000 | 8.000000 |
| mean | 411.125000 | 4.500000 | 0.011925 |
| std | 227.342057 | 3.070598 | 0.006705 |
| min | 63.000000 | 1.000000 | 0.004200 |
| 25% | 274.750000 | 2.500000 | 0.006650 |
| 50% | 387.000000 | 4.000000 | 0.011550 |
| 75% | 562.500000 | 6.250000 | 0.015450 |
| max | 795.000000 | 10.000000 | 0.024500 |

Kuva 12. Pandas-kirjaston describe-komento annettuna dataframe-oliolle. Kuvakaappaus Jupyter Notebookista.

Edellä kuvatut toiminnot suoritettiin kaikille taulukoille, joista näkee suoraan mediaanin (50 %).

Taulukko 1. Necunos Oy:n verkkosivuvierailut, sähköpostilistalle kirjautuneet ja näiden suhteen päivittäinen mediaani jokaisen kuukauden osalta sekä muutos prosenteissa edelliseen kuukauteen verrattuna.

| Kuukausi | Päivittäiset vierailijat (mediaani) | Muutos edelliseen kuukauteen (%) | Päivittäiset sähköpostilistalle kirjautuneet (mediaani) | Muutos edelliseen kuukauteen (%) | Päiväkohtainen vierailijoiden ja postituslistalle kirjautuneiden suhde (mediaani) | Muutos edelliseen kuukauteen (%) |
|---|---|---|---|---|---|---|
| Syyskuu 2018 | 387 | 0 | 4 | 0 | 1,16% | 0 |
| Lokakuu 2018 | 109 | -71,83% | 1 | -75% | 0,96% | -16,88% |
| Marraskuu 2018 | 83,5 | -23,40% | 1 | +/-0% | 1,47% | -46,35% |
| Joulukuu 2018 | 542 | +549,10% | 19 | +1800% | 2,13% | +313,59% |
| Tammikuu 2019 | 563 | +3,88% | 1 | -94,74% | 0,19% | -91,08% |
| Helmikuu 2019 | 423 | -24,87% | 2 | +100% | 0,57% | +200% |

| | | | | | | |
|---|---|---|---|---|---|---|
| Koko ajalta yh-teensä | 404 | - | 2 | - | 0,59% | - |

Kuten taulukosta 1 näkyy, keskimääräinen päivittäisten verkkosivuvierailijoiden lukumäärä on ailahdellut jonkin verran 09/2018 – 02/2019 välisenä ajanjaksona ja kasvua on saatu hieman vuodenvaihteessa, mutta mitään merkittäviä muutoksia ei tuona ajanjaksona ole syntynyt. Päivittäisten sähköpostilistalle kirjautuneiden kesimääräinen lukumäärä sekä postituslistalle kirjautuneiden ja verkkosivuvierailijoiden suhde on pysynyt suhteellisen samana, pois lukien joulukuu, jolloin postituslistalle on kirjautunut keskimäärin 19 uutta käyttäjää päivässä.

Sähköpostilistalle kirjautuneiden kävijöiden lukumäärää ei ole saatu nostettua lähes lainkaan, vaikka verkkosivuvierailijoiden lukumäärä onkin kasvanut. Ainoa positiivisesti eroava poikkeus on joulukuu, joten tarkempi perehtyminen siihen vaikuttaneisiin syihin on tarpeen. Joulukuussa on luotu yrityksen Facebook-tili ja uusittu nettisivut, mutta nämä yksistään ovat tuskin syy kyseiseen piikkiin, sillä tammi- ja helmikuussa lukemat ovat tippuneet taas samalle tasolle kuin aiemmin syksyllä. Todennäköisin syy on joulukuussa julkaistujen uutisten tuoma näkyvyys ja kiinnostus yritystä kohtaan. Joulukuun alkupuoliskolla julkaistiin uutiset Necunos Oy:n ja usean eri avoimeen lähdekoodiin perustuvan mobiilikäyttöjärjestelmän yhteistyöstä, joka huomioitiin muutamissa alan verkkojulkaisuissa, kuten esimerkiksi ”LinuxNews”- ja ”Fossbytes”-sivustoilla. [30; 31.]

Verkkosivuvierailijoiden määrän kasvuun on hieman vaikuttanut sosiaalisen median tuoma lisänäkyvyys ja yrityksen noteeraaminen muilla alan julkaisuihin perehtyneillä nettisivuilla sekä foorumeilla. Taulukossa 2 on esiteltynä promootion vaikutus verkkosivuvierailujen- ja sähköpostilistalle kirjautuneiden lukumäärään.

Taulukko 2. Promootion vaikutus verkkosivuvierailijoiden ja sähköpostilistalle kirjautuneiden lukumäärään sekä näiden suhteeseen

| **Promootio** | **Verkkosivuvierailijoiden lkm (ka.)** | **Sähköpostilistalle kirjautuneiden lkm (ka.)** | **Sähköpostilistalle kirjautuneiden suhde verkkosivuvierailijoihin (ka.)** |
|---|---|---|---|
| Kyllä | 602 | 3 | 0,76% |
| Ei | 153,5 | 1 | 0,62% |

Niinä päivinä, jolloin yritys on julkaissut sosiaalisessa mediassa, omilla nettisivuillaan tai kampanjoinut sähköpostitse on verkkosivuvierailuiden määrä ollut koko ajanjaksolta tarkasteltuna keskimäärin huomattavasti suurempi verrattuna päiviin, jolloin minkäänlaista promootiota ei ole ollut. Julkaisujen tuoma näkyvyys on lisännyt sähköpostilistalle kirjautuneiden osuutta vain yhdellä prosentin kymmenyksellä, mutta luonnollisesti isompi kävijämäärä sivustolla tarkoittaa myös useampaa sähköpostilistalle kirjautunutta kävijää.

Sähköpostilistalle kirjautuneiden määrä on kasvanut koko ajan ja vain harva on poistanut itsensä listalta. Itse listalle kirjautuneiden lukumäärä ei sinänsä ole kovinkaan merkityksellinen vaan merkittävää on se, kuinka moni on avannut heille lähetetyt sähköpostit ja ennen kaikkea klikannut sähköpostin sisältämää linkkiä. Nämä käyttäjät on jaettu ryhmiin kirjautumiskuukauden mukaan.

Alla olevassa taulukossa on kyseisen kuukauden aikana kirjautuneiden keskimääräiset CTR-, EOR- ja CTOR arvot. Helmikuussa kirjautuneita käyttäjiä ei ole tässä listassa huomioitu siitä syystä, että useat heistä ovat ehtineet vastaanottamaan vain yhden sähköpostin, joka vääristäisi tilastoja turhaan.

Taulukko 3. Syyskuussa 2018 – Tammikuussa 2019 sähköpostilistalle kirjautuneiden käyttäjien CTR-, EOR- ja CTOR-arvot.

| **Kuukausi** | **CTR (keskiarvo)** | **EOR (keskiarvo)** | **CTOR (keskiarvo)** |
|---|---|---|---|
| **Viitearvo** | **n. 14%** | **20% - 40%** | **20% - 30%** |
| Syyskuu 2018 | 19,50% | 41,70% | 48,87% |
| Lokakuu 2018 | 16,78% | 31,99% | 54,11% |
| Marraskuu 2018 | 15,71% | 35,19% | 48,89% |
| Joulukuu 2018 | 20,87% | 42,79% | 49,96% |
| Tammikuu 2019 | 11,40% | 31,58% | 36,84% |
| Koko Ajalta | 18,56% | 39,41% | 49,41% |

Necunos Oy:n sähköpostikampanjointi on tarkasteltuna ajanjaksona toiminut erinomaisen hyvin. Ainoastaan tammikuun CTR-arvo on hieman heikko, mutta muutoin kaikki arvot ovat viiterajoissa tai jopa niiden yläpuolella. CTOR-arvo, joka kuvaa sähköpostien sisällön laatua parhaiten on pysynyt jokaisen kuukautena todella hyvänä, joten tällä osa-alueella yrityksen toiminta on erinomaista.

Aktiivisten käyttäjien osuus kaikista sähköpostilistalle kirjautuneista käyttäjistä kuukausittaisella tasolla määriteltiin siten, että ne käyttäjät, jotka ovat avanneet yli puolet heille

lähetetyistä sähköposteista, määriteltiin aktiivisiksi. Sitoutuneiksi käyttäjiksi määriteltiin ne käyttäjät, jotka ovat avanneet ja klikanneet sähköpostin sisältämää linkkiä yli puolista heille lähetetyistä sähköposteista. Nämä arvot määriteltiin asteikolla 1-4 aiemmin tässä luvussa esitetyllä funktiolla niin, että käyttäjät, joiden 'Interest_Rate'-arvo on 1-2 lasketaan aktiivisiksi, ja käyttäjät, joiden 'Engagement_Rate'-arvo on 1-2, lasketaan sitoutuneiksi.

```
In [43]:  1 Subscribers_Sep['Interest_Class'].value_counts(normalize=True) * 100

Out[43]: 4    37.500000
         1    24.431818
         3    21.022727
         2    17.045455
         Name: Interest_Class, dtype: float64
```

Kuva 13. Komento, jolla saadaan halutun sarakkeen sisältämien arvojen prosenttijakauma tulostettua. Kuvakaappaus Jupyter Notebookista.

Taulukko 4. Aktiivisten ja sitoutuneiden käyttäjien osuus kaikista sähköpostilistalle kirjautuneista kuukausittain.

| Kirjautumis-Kuukausi | Aktiivisten Käyttäjien Osuus | Sitoutuneiden Käyttäjien Osuus | Uusien kirjautuneiden lukumäärä |
|---|---|---|---|
| Syyskuu 2018 | 41,48% | 11,93% | 176 |
| Lokakuu 2018 | 28,07% | 10,53% | 57 |
| Marraskuu 2018 | 33,21% | 5,54% | 542 |
| Joulukuu 2018 | 47,12% | 20,12% | 815 |
| Tammikuu 2019 | 33,33% | 14,04% | 57 |
| Helmikuu 2019 | 33,33% | 6,67% | 56 |

Aktiivisten käyttäjien osuus on ollut syyskuussa ja joulukuussa hieman suurempi, kuin muina kuukausina, mutta suurin piirtein noin kolmasosa käyttäjistä on aktiivisia. Sitoutuneiden käyttäjien osuus on joulukuussa (20 %) huomattavasti suurempi kuin muina kuukausina, jolloin sitoutuneita käyttäjiä on suurin piirtein yksi kymmenestä. Käyttäjien lukumäärä ei myöskään näytä vaikuttavan näihin prosentteihin, ja se kuvaa ainoastaan, kuinka monta käyttäjää kyseisenä kuukautena on kirjautunut yrityksen sähköpostilistalle. Näille arvoille ei ole olemassa mitään kiveen hakattua viite arvoa, vaan se on yrityksen itsensä määrittelemä. Tästä syystä niiden tuloksia tulee verrata ainoastaan yrityksen asettamiin tavoitteisiin, jolloin voidaan ryhtyä arvioimaan, miksi tavoite on tai ei ole saavutettu. Necunos Oy:llä ei ole tällaista arviota vielä olemassa, joten siinä mielessä näitä

arvoja ei ole mahdollista tarkastella kovin kriittisesti. Positiivista kuitenkin on se, että uusia käyttäjiä kirjautuu sähköpostilistalle jatkuvasti ja heistä noin kolmasosa lukee aktiivisesti sähköpostit ja keskimäärin joka kymmenes klikkaa sähköpostissa olevaa linkkiä.

# 4 Ratkaisun arviointi

Tässä luvussa tarkastellaan, kuinka hyvin työssä käytetyt teknologiat ja menetelmät ovat toimineet tavoitteeseen nähden ja pohditaan vaihtoehtoisia lähestymistapoja sekä mahdollisia muutos-/parannusehdotuksia.

## 4.1 Työn tulokset

Growth hacking -menetelmät vaikuttavat toimivilta, vaikka niihin ei tässä työssä aivan niin syvällisesti pystytty perehtymään, kuin alun perin oli tarkoitus. Potentiaalisten asiakkaiden hankkiminen, eli tässä tapauksessa sähköpostilistalle kirjautuneiden kävijöiden määrää saatiin kasvatettua ja verkkosivuvierailujen määrää lisättyä sosiaalisen median ja muiden julkaisujen avulla. Sähköpostilistalle kirjautuneiden käyttäjien kiinnostus on saatu pysymään yrityksessä ja sen tuotteissa laadukkaan sähköpostikampanjoinnin avulla. Sähköpostilistalle kirjautuneiden osuuteen kaikista verkkosivuvierailijoista ei työssä päästy valitettavasti vaikuttamaan. Tähän löytyisi kyllä erilaisia keinoja kuten yksinkertaiset testit. Voitaisiin esimerkiksi vaihtaa sähköpostilistalle kirjautumiseen tarkoitetun kentän sijaintia yrityksen verkkosivuilla, mutta kyseisen toiminnon vahvistus painikkeen kokoa tai siinä lukevan tekstin fonttia/kokoa. Jokaista testiä voitaisiin suorittaa muutaman viikon ajan ja katsoa, kuinka se vaikuttaa sähköpostilistalle kirjautuneiden määrään.

Python-ohjelmointikieli ja sen tarjoamat kirjastot ovat toimineet hyvin työn tarkoituksiin. Erityisesti pandas-kirjasto tarjoaa monipuoliset ja helposti käytettävät työkalut datan käsittelyyn ja analysointiin. Toinen mahdollisuus olisi ollut käyttää R-ohjelmointikieltä [32], joka on laajasti käytetty ohjelmointikieli data-analytiikassa. Jupyter Notebookin selkeät syöte- ja tulosterivit toimivat hyvin datan käsittelyssä, sillä se tulosti luodut taulukot ja niiden rakenteen aina näkyviin työtä tehdessä, mikä edisti tulosten tarkastelua. Datan

visualisointiin ei työssä käytetty kovinkaan paljon aikaa ja kaikki visualisoinnit on toteutettu joko käsin luoduilla taulukoilla tai MS Excelin tarjoamilla valmiilla kuvaajilla. Joka tapauksessa nämä kuvaajat ajavat asiansa tulosten tarkastelussa.

Työssä käytetyt teknologiset ratkaisut saavuttivat tavoitteen toiminnon luomisesta, jolla Necunos Oy pystyy hakemaan halutulta ajanjaksolta verkkosivuvierailijoiden tiedot, käsittelemään ja tallentamaan ne ennen jatkokäsittelyä. Koodiesimerkkejä esiintyy työssä paljon siitä syystä, että niitä soveltamalla työtä voitaisiin hyödyntää myös jossain toisessa samanlaisessa projektissa. Ne pyrittiin esittämään mahdollisimman selkeästi ja avaamaan näiden toimintojen käyttötarkoituksia samalla. Sitä, kuinka hyvin tässä tavoitteessa onnistuttiin, on hieman haasteellista arvioida itse, mutta datan käsittely- ja analysointivaihe on kuvattu suhteellisen kattavasti näyttämättä käyttäjien henkilötietoja, joten tämäkin tavoite on jossain määrin saavutettu.

## 4.2 Muut analyysimahdollisuudet

Data-analytiikassa on lukemattomia erilaisia mahdollisuuksia edetä ja riippuu täysin siitä, mitä datasta halutaan selvittää. Tässäkin työssä olisi moni asia pystytty tekemään eri tavalla. Toiminnosta, joka hakee verkkosivuvierailijoiden tiedot yrityksen palvelimelle asennetusta Matomo-analytiikkaohjelmistosta, olisi voinut tehdä asynkronisen, jolloin se olisi huomattavasti tehokkaampi ja nopeampi ajaa. CSV-tiedostoihin voitaisiin jättää enemmän sarakkeita, jolloin pystyttäisiin selvittämään jatkossa myös muita asioita kuin niitä, mitä tässä työssä on pidetty tärkeänä. Esimerkiksi voitaisiin selvittää:

- Mistä maista verkkosivuvierailijat ja sähköpostilistalle kirjautuneet ovat kotoisin?
- Millä laitteilla yrityksen verkkosivuilla yleensä vieraillaan ja mitkä ovat näiden laitteiden näyttöjen koot?
- Mitä selaimia verkkosivuvierailijat yleensä käyttävät?

Yllä mainittuja asioita voitaisiin tarkastella yrityksen verkkosivujen optimoinnissa, jotta ne olisivat mahdollisimman käyttäjäystävälliset. Necunos Oy haluaa turvata käyttäjien datan yksityisyyden, jolloin tässä työssä monet analyysi mahdollisuudet eivät ole toteutettavissa. Toisenlaisessa tilanteessa olisi mahdollista lisätä Matomo-analytiikkaohjelmistoon

JavaScript Cookiet, joilla voitaisiin jäljittää verkkosivuvierailijoiden hiiren liikkeitä ja vietettyä aikaa sivuston eri sivuilla, mikä auttaisi tehostamaan verkkosivujen optimointia.

Analyysin kannalta olisi tehokasta hyödyntää koneoppimista. Esimerkiksi Scikit-Learn [33] on Python-kirjasto, joka tarjoaa hyvät ja monipuoliset työkalut datan louhintaan ja analysoimiseen sekä sisältää paljon erilaisia algoritmeja hyödynnettäväksi koneoppimisessa. Kerätystä datasta voitaisiin esimerkiksi klusteroida verkkosivuvierailijoista ne, jotka ovat sähköpostilistalle kirjautuneet ja katsoa, löytyykö heidän joukossaan mitään yhteneväisyyksiä. Mahdollista olisi myös tarkastella sähköpostikampanjoita tekstinlouhinnan avulla ja selvittää, millainen sisältö on ollut suosittua ja millainen ei. Koneoppimisen avulla pystyttäisiin varmasti monipuolistamaan ja tehostamaan erilaisia analyysimahdollisuuksia.

# 5 Yhteenveto

Työssä toteutettiin toiminnot Necunos Oy:lle, joiden avulla verkkosivuvierailijoiden tiedot saadaan haettua haluttujen päivämäärien väliseltä ajanjaksolta, muokattua ja tallennettua csv-tiedostoformaatissa. Luotiin toiminnot, joiden avulla toteutettiin haetun datan jatkokäsittely niin, että sitä voidaan hyödyntää BI-järjestelmien avulla. Samassa arvioitiin valittujen tietojen analysoinnin tuottamia tuloksia. Toimintojen toteutus pyrittiin kuvaamaan sillä tarkkuudella, että niitä pystyttäisiin soveltamalla hyödyntämään jossain vastaavassa, toisenlaisessa työssä.

Työprosessi kesti ajallisesti suhteellisen kauan, sillä datan keräämisessä hyödynnetty Matomo-analytiikkaohjelmisto on asennettu yrityksen palvelimelle vasta syyskuussa 2018. Jotta työssä olisi ollut käytettävissä tarpeeksi dataa, päätettiin sitä kerätä 09/2018 – 02/2019 väliseltä ajalta, minkä vuoksi työn laatiminen ei onnistunut lyhyellä aikavälillä. Tämä toi hieman haasteita työn tekemiselle, sillä itse toimintoja luotiin pitkin talvea ja raportin kirjoittaminen tapahtui keväällä, jonka vuoksi raporttia kirjoittaessa on välillä ollut niin sanotusti punainen lanka hieman hukassa. Suurena apuna ja ohjekirjana startup-yritykseen ja sen kasvuun liittyvässä datan analysoimisessa on ollut Crollin ja Yoskovitzin (2013) teos*, Lean Analytics: Use Data to Build a Better Startup Faster*. Kirja sisältää paljon hyviä vinkkejä siitä, mitä datasta kannattaa erilaisissa startup-yrityksissä selvittää

ja mitä analyysia tehdessä kannattaa pitää silmällä. Ongelmana oli datan valtava määrä ja sen tuomat tuhannet erilaiset lähestymistavat sekä useat eri analyysimahdollisuudet. Ajoittain tämä johti pienimuotoiseen analyysiparalyysiin, jolloin työn eteneminen ei hirveästi edennyt, kun aika kului pohtiessa, mitä tehdä seuraavaksi. Aiheen rajaaminen oli myös ajoittain hieman hankalaa, mutta työn edetessä rajat selkeytyivät koko ajan ja lopulta saatiin työn tavoitteet ja päämäärä huomattavasti selkeämmäksi. Toimintojen luominen edellytti aiheeseen perehtymistä ja itsenäistä opiskelua. Niiden luominen tapahtui suhteellisen pitkällä aikavälillä, jonka aikana myös tietotaitoa aiheesta karttui koko ajan. Tästä syystä toimintoja joutui hieman hiomaan jälkikäteen, ja niiden tarkastelu vei jonkin verran aikaa. Ajan puitteissa monia työssä esitettyjä toiminnallisuuksia olisi voinut luoda tehokkaammiksi, jotta ne veisivät vähemmän koneen muistia ja niiden ajaminen olisi nopeampaa. Datan määrä ei kuitenkaan ollut niin valtava, että tästä olisi ollut mainittavaa hyötyä, mutta ottaen huomioon työn tavoitteen esitellä lukijalle toimintojen soveltamisen mahdollisuus, olisi toimintojen suorituskykyä voinut mahdollisesti tarkastella kriittisemmin. Ennen kaikkea työtä tehdessä oppi paljon uutta data-analytiikasta ja siihen käytetyistä teknologioista sekä menetelmistä ja niiden hyödyntämisestä startup-yrityksessä.

## Lähteet


1 Kielikello. Verkkojulkaisu. <https://www.kielikello.fi/-/kasvuyritys-ja-startup-yritys>. Julkaistu: 2/2013. Luettu: 07.03.2019.

2 Sosiaalinen media hyötykäyttöön. Verkkojulkaisu. <http://www.sosiaalinenmedia-opetuksessa.com/mita-hyotya-sosiaalisesta-mediasta-on/>. Julkaistu: 31.10.2016. Luettu: 12.03.2019.

3 Valtioneuvoston selvitys- ja tutkimustoiminta. Raportti: Datan hyödyntäminen lisää innovaatioita ja kasvua. Verkkojulkaisu. <https://tietokayttoon.fi/artikkeli/-/asset_publisher/10616/raportti-datan-hyodyntaminen-lisaa-innovaatioita-ja-kasvua>. Julkaistu: 16.03.2017. Luettu: 12.03.2019.

4 Croll, Alistair & Yoskovitz, Benjamin (2013) Lean Analytics. Use Data to Build a Better Startup Faster. USA: O'Reilly.

5 Campaign Monitor. CTOR – the Email Marketing Metric You May Not Know. Verkkojulkaisu. (https://www.campaignmonitor.com/resources/glossary/click-through-rate-ctr/). Julkaistu: 09/2019. Luettu: 09.05.2019.

6 Campaign Monitor. CTOR – the Email Marketing Metric You May Not Know. Verkkojulkaisu. (https://www.campaignmonitor.com/blog/email-marketing/2019/01/ctor-email-marketing-metrics-you-may-not-know/). Julkaistu: 09/2019. Luettu: 09.05.2019.

7 Campaign Monitor. Glossary – Email Marketing Terms. Verkkosivusto. (https://www.campaignmonitor.com/resources/glossary/email-open-rate/). Luettu: 09.05.2019.

8 Campaign Monitor. CTOR – the Email Marketing Metric You May Not Know. Verkkojulkaisu. (https://www.campaignmonitor.com/blog/email-marketing/2019/01/ctor-email-marketing-metrics-you-may-not-know/). Julkaistu: 09/2019. Luettu: 09.05.2019.

9 Campaign Monitor. Glossary – Email Marketing Terms. Verkkosivusto. (https://www.campaignmonitor.com/resources/glossary/click-to-open-rate-ctor/). Luettu: 09.05.2019.

10 Campaign Monitor. CTOR – the Email Marketing Metric You May Not Know. Verkkojulkaisu. (https://www.campaignmonitor.com/blog/email-marketing/2019/01/ctor-email-marketing-metrics-you-may-not-know/). Julkaistu: 09/2019. Luettu: 09.05.2019.

11 Patel, Neil & Taylor, Bronson, The Definitive Guide to Growth Hacking, (https://www.quicksprout.com/the-definitive-guide-to-growth-hacking-chapter-1/) Luettu: 15.03.2019.

12 Maxime Pudzeis, Top 10 best growth hacking tools to boost your digital marketing efficiency, (https://www.thehouseofmarketing.be/blog/top-10-best-growth-hacking-tools-to-boost-your-digital-marketing-efficiency), Julkaistu: 06.12.2017. Luettu: 15.03.2019.

13 Matomo's History. Verkkojulkaisu. (https://matomo.org/history/), Luettu: 21.03.2019.

14 Venners Bill, The Making of Python. A Conversation with Guido van Rossum, Part I. Verkkojulkaisu. (https://www.artima.com/intv/pythonP.html), Julkaistu: 13.01.2003. Luettu: 21.03.2019.

15 The Python Tutorial. Verkkojulkaisu. (https://docs.python.org/3/tutorial/index.html#tutorial-index), Luettu: 21.03.2019.

16 Applications for Python. Verkkojulkaisu. (https://www.python.org/about/apps/). Luettu 22.03.2019.

17 Pandas. Verkkosivu. (https://pandas.pydata.org/), Luettu: 22.03.2019.

18 Matplotlib. Verkkosivu. (https://matplotlib.org/), Luettu: 22.03.2019.

19 Beautiful Soup Documentation. Verkkosivu. (https://www.crummy.com/software/BeautifulSoup/bs4/doc/), Luettu: 22.03.2019.

20 Jacob Coshy, Web Scraping: Challenges and Roadblocks. Verkkojulkaisu. (https://www.promptcloud.com/blog/challenges-roadblocks-in-web-scraping/). Julkaistu: 18.08.2017. Luettu: 29.05.2019.

21 Anaconda Distribution. Verkkosivu. (https://www.anaconda.com/distribution/), Luettu: 25.03.2019.

22 Jupyter. Verkkosivu. (https://jupyter.org/), Luettu: 25.03.2019.

23 The Jupyter Notebook Docs. Verkkosivu. (https://jupyter-notebook.readthedocs.io/en/latest/notebook.html), Luettu: 25.03.2019.

24 MailerLite. Verkkosivu. (https://www.mailerlite.com/about). Luettu: 09.05.2019.

25 Wikipedia, Botti. Verkkosivu. (https://fi.wikipedia.org/wiki/Botti), Luettu: 27.03.2019.

26 Neil, Andrew, What Percentage of Online Users Are Robots, Verkkojulkaisu, (https://ppcprotect.com/how-many-of-the-internets-users-are-robots/), Julkaistu: 01.12.2017. Luettu: 27.03.2019.

27 Neil, Andrew, How to Detect Bot Traffic on Your Website, Verkkojulkaisu, (https://ppcprotect.com/how-to-detect-bot-traffic/), Julkaistu: 08.12.2017. Luettu: 27.03.2019.

28 Stackoverflow, How to deal with SettingWithCopyWarning in Pandas?, Verkkosivusto, (https://stackoverflow.com/questions/20625582/how-to-deal-with-settingwithcopywarning-in-pandas?noredirect=1&lq=1), Julkaistu: 01/2014, Luettu: 14.04.2019.

29 Firstscribe, What is Direct Traffic? Internet Marketing Mysteries, Verkkosivusto, (https://www.firstscribe.com/exactly-direct-traffic-intenet-marketing-mysteries/), Julkaistu: 16.01.2015, Luettu: 16.04.2019.

30 LinuxNews, PostmarketOS partnert mit Necunos, Verkkojulkaisu, (https://linux-news.de/2018/12/postmarketos-partnert-mit-necunos/), Julkaistu: 03.12.2018, Luettu: 14.05.2019.

31 FossBytes, Necuno To Launch Linux-Based Smartphone With KDE Plasma Mobile, Verkkojulkaisu, (https://fossbytes.com/necuno-to-launch-linux-based-smartphone-with-kde-plasma-mobile/), Julkaistu: 02.12.2018, Luettu: 14.05.2019.

32 The R Project for Statistical Computing, Verkkosivusto, (https://www.r-project.org/), Luettu: 14.05.2019.

33 Scikit-Learn, Verkkosivusto, (https://scikit-learn.org/stable/), Luettu: 15.05.2019.

## Datan hakemiseen tehdyn toiminnon ohjelmakoodi kokonaisuudessaan

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
from datetime import timedelta, date
import pandas as pd

#Assign 'robots.txt' to robots for later use to check if visitor tried to
download robots.txt file from webpage

robots = ['robots.txt']


# Function to loop through given dates

def daterange(start_date, end_date):
     for n in range(int ((end_date - start_date).days)):
          yield start_date + timedelta(n)

# First date of daterange function

start_date = date(2018, 9, 23)

# End date (not included in the loop, last date will be previous date)

end_date = date(2019, 2, 22)

# Iterate through given dates

for single_date in daterange(start_date, end_date):

     # Empty list to store urls on each iteration

     html_list = []



     # Empty list to store dataframes on each iteration

     visitors = []

     # Empty list to store multiple dataframes from single day on each itera-
     tion, if any

     concatenation = []

     # Try-Catch to check if HTTP Response is 200 (Should be 200 or 500 in
     this case)

     try:
          # Url address to access data in Matomo software running on company's
          server

          url = "http:// 192.168.2.10: 8888/index.php?date="+sin-
          gle_date.strftime("%Y-%m-%d")+"&expanded=1&filter_limit=-1&for-
          mat=HTML&idSite=2&method=Live.getLastVisitsDetails&module=API&pe-
          riod=day&token_auth=token here"

          # Send HTTP Request to given url
```

```
        urlopen(url)

        # Store url as a string to html_list

        html_list.append(url)

        # If HTTP Response is not 200 (Unable to load all the visitor infor-
        mation at once, if too many visitors)
        # Load a maximum of 1000 rows (visitors) at a time

    except:

        # Add filter_offset and filter_limit parameters to url to modify
        amount of rows
        # filter_offset = 0, starts from first row on a given date

        filter_offset = 0

        # filter_limit = 1000, number of rows

        filter_limit = 1000

        # Url address with modified parameters

        url = "http://192.168.2.10:8888/index.php?date="+sin-
        gle_date.strftime("%Y-%m-%d")+"&expanded=1&filter_limit="+str(fil-
        ter_limit)+"&filter_offset="+str(filter_offset)+"&for-
        mat=HTML&idSite=2&method=Live.getLastVisitsDetails&module=API&pe-
        riod=day&token_auth=token here"

        # Store url to html_list

        html_list.append(url)

        # Continue adding urls to html_list until html table element is
        empty

        while True:
            # Increase filter_offset by 1000 to get the next 1000 rows

            filter_offset += filter_limit

            # Url address with modified parameters

            url = "http://192.168.2.10:8888/index.php?date="+sin-
            gle_date.strftime("%Y-%m-%d")+"&expanded=1&fil-
            ter_limit="+str(filter_limit)+"&filter_offset="+str(fil-
            ter_offset)+"&format=HTML&idSite=2&method=Live.getLastVis-
            itsDetails&module=API&period=day&token_auth=token here"

            # Store HTTP Response to html_response variable

            html_response = urlopen(url)

            # Check if HTTP Response includes only empty table and break
            while loop if true.

            if len(html_response.read()) <= 106:
                break

            # Store all the gathered urls in html_list
```

```
            html_list.append(url)

# Iterate through urls to scrape visitor information from each

for h in range(0,len(html_list)):

      #Send request to each url in html_list

      content = urlopen(html_list[h])

      # Read html document to BeautifulSoup object

      bs_obj = BeautifulSoup(content, 'lxml')

      # Find "table" tags from the object and store them to tables vari-
      ble

      tables = bs_obj.findAll('table')

      # Read each html table with pandas to create list of dataframes

      for table in tables:

            dataframe = pd.read_html(str(table))

            visitors.append(dataframe)

#Iterate through visitors list

for i in range(0,len(visitors)):


      #Create a pandas DataFrame from each list item

      visitorsdf = pd.DataFrame(visitors[i][0])

      #Assign all rows containing robots.txt in actionDetails to bot vari-
      able

      bot = visitorsdf.actionDetails[visitorsdf.actionDetails.str.con-
      tains('|'.join(robots), na=False)]

      #Remove visitors who downloaded robots.txt file

      Visit = visitorsdf[~visitorsdf.actionDetails.isin(bot)]

      # Keep only useful columns

      Visit = visitorsdf.filter(['idVisit', 'visitIp', 'serverDatePretty',
      'siteName', 'visitorType', 'visitCount', 'daysSinceFirstVisit',
      'visitDuration', 'actions', 'referrerType', 'referrerName', 'device-
      Type', 'deviceBrand', 'operatingSystemName', 'browserName', 'coun-
      try', 'countryCode', 'location'])

      #Remove visitors identified as Bots

      Visit = Visit.loc[Visit['operatingSystemName'] != 'Bot']

      #Remove visitors with visit duration more than 1h as bots

      Visit = Visit[(Visit['visitDuration'] < 3600)]
```

```
        #Remove users with more than 30 actions as bots

        Visit = Visit[(Visit['actions'] < 30)]

        #Fill missing values of referrerName with Unknown

        Visit['referrerName'].fillna('Unknown', inplace=True)

        if len(visitors) > 1:
             concatenation.append(Visit)

             for i in range(0,len(concatenation)):

                  Visit = Visit.append(concatenation[i])


#Save Datframe as csv file to given directory

Visit.to_csv("C:/Users/Lari/Documents/Visit/Visitors-" + sin-
gle_date.strftime("%Y-%m-%d") + '.csv', encoding='utf-8')
```