HO
GENT

Faculteit Bedrijf en Organisatie

Webscraping van dynamisch geladen bronnen

Mathias Cloet

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Sabine De Vreese
Co-promotor:
Jonas Vanhulle

Instelling: —

Academiejaar: 2019-2020

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Webscraping van dynamisch geladen bronnen

Mathias Cloet

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Sabine De Vreese
Co-promotor:
Jonas Vanhulle

Instelling: —

Academiejaar: 2019-2020

Tweede examenperiode

Woord vooraf

In het derde jaar Toegepaste Informatica aan de Hogeschool Gent wordt een scriptie verwacht die relevant is voor deze opleiding. Aangezien Big Data steeds belangrijker wordt voor bedrijven om een competitief voordeel te verkrijgen tegenover de concurrentie of om applicaties te ontwikkelen die gebruik maken van machine learning. Daarom leek het mij dan ook interessant om een scriptie te schrijven over webscraping. Webscraping is een techniek die gebruikt wordt om data van een online bron te downloaden en formateren voor later gebruik. Ik heb in het verleden al enkele keren geprobeerd om data te scrapen van verschillende websites voor persoonlijke projecten of interesse. Deze pogingen zijn het verleden nooit echt tot een goed einde kunnen gebracht worden omwille van gebruik te maken van slechte technologieën en onwetendheid. Daarom leek het mij het ideale moment om webscraping te verwerken in mijn bachelorproef en er zo meer kennis over te verkrijgen. Meer bepaald welke technieken en methodes zijn het meest succesvol om webscraping toe te passen.

Alleen had ik deze bachelorproef zeker niet kunnen afwerken. Daarom wil ik graag enkele mensen bedanken. Eerst en vooral wil ik mijn promotor Sabine De Vreese die mij extra doorzettingsvermogen en inspiratie heeft gegeven om dit onderzoek uit te voeren bedanken. Alsook wil ik mijn copromotor Jonas Vanhulle bedanken die mij met verschillende zaken kon bijstaan en een antwoord kon bieden op nodige vragen. Ten slotte wil ik mijn ouders en vriendin Laurence Reynaert bedanken. Zij hebben mij enorm gesteund en geholpen, niet alleen tijdens deze scriptie maar in het volledige academiejaar.

Samenvatting

Op het internet zijn grote hoeveelheden data ter beschikking, deze data kan interessant zijn voor het ontwikkelen van applicaties. Maar data kan ook gebruikt worden om als bedrijf een competitief voordeel te verkrijgen tegenover de concurrentie. Grote bedrijven hebben vaak al grote hoeveelheden data verzameld over de tijd heen, deze data die zij ter beschikking hebben kan hen veel voordelen geven. Zo is het voor een bedrijf zoals Google of Facebook veel makkelijker om applicaties die gebruik maken van artificiële intelligentie te ontwikkelen dan een startup of klein bedrijf. In dit onderzoek werd er onderzocht of webscraping een goede manier is voor het verzamelen van grote hoeveelheden data van allerlei bronnen.

Als praktijkvoorbeeld hebben we in deze scriptie onderzocht wat de effecten van de corona crisis op de economische markt en de beursomgeving zijn. Op deze manier kan er worden geïllustreerd dat webscraping een goede manier is om data te verzamelen die later relevant kan zijn voor onderzoeken of voor het ontwikkelen van applicaties. Zo kan een bedrijf dat gebruik maakt van e-commerce bijvoorbeeld de webshop van een concurrent scrapen en zo de prijzen vergelijken en inspelen op eventuele verschillen.

Vooreerst werd via een literatuurstudie nagegaan wat een gepast framework zou zijn om te fungeren als de bouwsteen van een webscraper. Met behulp van deze literatuurstudie werd het duidelijk dat er twee verschillende mogelijkheden waren voor webscraping namelijk Selenium en Splash. Eenmaal de voor- en nadelen van beide frameworks overwogen waren werd er uiteindelijk gekozen voor Splash. Voor dit onderzoek werd er via een applicatie data van twee verschillende bronnen verzameld namelijk www.beursduivel.be en www.stockmonitor.be.

Uit het onderzoek bleek dat webscraping een goede manier kan zijn voor een bedrijf om

informatie van het web te verzamelen en te classificeren voor later gebruik. Een nadeel van webscraping is dat er voor iedere bron manueel code moet aangepast worden om relevante data te verzamelen.

Inhoudsopgave

Lijst van figuren

1. Inleiding

Data wordt steeds belangrijker voor bedrijven, data is de sleutel tot het verkrijgen van een competitief voordeel tegenover de concurrentie. Bedrijven gebruiken door het stijgende belang van data ook steeds meer verschillende manieren voor het verzamelen van data. Een van deze methodes om data te verzamelen is webscraping, dit is een techniek waarbij ongeorganiseerde data van het web verzameld en opgeslagen wordt in een gemakkelijk bruikbaar formaat. Meestal overloopt een webscraper iedere url van het web en met behulp van technieken zoals REGEX, CSS of XPATH wordt deze ruwe web data omgezet naar een bruikbaar data formaat. Webscraping is geliefd bij start-ups, kleine en grote bedrijven. Zeker voor start-ups is deze techniek zeer interessant want het zorgt ervoor dat ze over heel wat data kunnen beschikken zonder dat ze connecties moeten aangaan met andere organisaties.

Bedrijven kunnen webscraping gebruiken voor heel wat interessante use cases. Zo kan een e-commerce de websites van de concurrenten scrapen en prijsverschillen bekijken tussen producten om zo een voordeel te krijgen tegenover de concurrentie. Een andere use case is het ophalen van recensies en reviews van een product om inzicht te krijgen hoe een product ontvangen wordt door de markt.

Webscraping is omwille van de eerder vermelde redenen geliefd bij bedrijven, maar bedrijven hebben liever niet dat webscraping wordt uitgevoerd op hun openbare bronnen omdat dit een voordeel kan geven aan de concurrentie. Hierdoor zijn er enkele manieren die organisaties toepassen om het scrapen van hun data heel wat moeizamer te maken. Een van de problemen is afkomstig van data die dynamisch wordt ingeladen door middel van JavaScript. Zo wordt ervoor gezorgd dat veel data achter een user interaction zit, zoals scrollen naar het eind van de pagina, op een knop klikken, privacy waarschuwing, inloggen op een webpagina ...

1.1 Probleemstelling

Data staat centraal bij bedrijven omdat er met behulp van data een competitief voordeel kan bereikt worden tegenover de concurrentie. Bedrijven willen dus zelf zoveel mogelijk data verzamelen en tegelijkertijd ervoor zorgen dat de data die zij weergeven op hun openbare bronnen moeilijker is om gebruikt te worden door andere organisaties. Webscraping is één van de technieken die bedrijven kunnen toepassen om data te verzamelen van openbare bronnen.

Webscraping is een manier waarbij alle geladen data van een webpagina opgehaald en omgevormd worden naar een bruikbaar formaat. Bij traditionele statische webapplicaties waar elke pagina zijn eigen url heeft en alle data geladen wordt door te surfen naar een url doen er zich geen problemen voor. Maar tegenwoordig maken bijna alle sites gebruik van JavaScript AJAX om dynamisch data op te halen na user interaction. Voor het ontwikkelen van webapplicaties is er sowieso een trend naar JavaScript, maar bedrijven zorgen er ook voor dat het ophalen van hun openbare data veel moeizamer verloopt, want 'Big Data' is geld waard. Dit zorgt ervoor dat er tools moeten ontwikkeld worden voor webscraping om user interaction na te bootsen maar deze tools zijn vaak niet ontwikkeld met als doel om webscraping uit te voeren waardoor deze een lage betrouwbaarheid en extra overhead kunnen veroorzaken.

1.2 Onderzoeksvraag

1.2.1 Hoofdonderzoeksvragen

Dit onderzoek zal zich vooral focussen op het verzamelen van financiële data van verschillende websites met behulp van webscraping. Met de data die zal gegenereerd worden uit het project zullen we verschillende vragen beantwoorden.

- Is webscraping een goeie en betrouwbare manier om data te verzamelen om te gebruiken in onderzoeken? Heel concreet willen we deze onderzoeksvraag inbedden in een onderzoek naar bewegingen op de beurs naar aanleiding van de corona crisis.

1.2.2 Deelonderzoeksvragen

Relevante deelonderzoeksvragen:

- Wanneer is de beurs beginnen crashen?
- Welke sectoren zijn het meest beïnvloed door de corona crisis?
- Wat doen de verschillende valuta's?
- Hoe hebben de verschillende markten gereageerd?
- Wat doet de rente's en obligatie?
- Wie zijn de grootste winnaars en verliezers?

1.3 Onderzoeksdoelstelling

We willen aantonen in de praktijk dat door middel van webscraping uit de data van twee verschillende bronnen namelijk 'www.beursduivel.be' en 'www.stockmonitor.com' relevante conclusies kunnen worden getrokken uit de marktbeweging die zich heeft voorgedaan tijdens de corona crisis binnen de tijdspanne van 01-01-2020 tot en met 15-05-2020. Deze data hebben betrekking op aandelen, sectoren en rentes.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken, alsook de gebruikte frameworks voor het onderzoek.

In Hoofdstuk 4 wordt uitgelegd wat er allemaal nodig is voor er van start kan gegaan worden met het beantwoorden van de deelonderzoeksvragen.

In Hoofdstuk 5 wordt er in detail uitgelegd wat er uit de gegenereerde data kan gehaald worden om zo de deelonderzoeksvragen te beantwoorden.

In Hoofdstuk 6, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2. Stand van zaken

Dit onderzoek zal zich richten op het verzamelen van financiële data van twee verschillende webbronnen met behulp van webscraping. Met de data die verkregen zal worden uit deze twee bronnen zal er een financiële analyse uitgevoerd worden. Op deze manier kan er worden aangetoond dat webscraping een goede manier is om data te verzamelen van verschillende bronnen zonder een groot budget te moeten spenderen. Voor dit onderzoek zullen de websites 'www.beursduivel.be' en 'www.stockmonitor.com' gebruikt worden. Van de website www.beursduivel.be kunnen de beurskoersen van de meest relevante markten gevonden worden. De bron 'www.stockmonitor.com' zal worden gescraped om de verschillende aandelen die we van 'www.beursduivel.be' halen te linken aan een sector.

Voor dit project zal er geopteerd worden om de nodige software, namelijk Splash (Zie sectie 2.3) en MongoDB (Zie sectie 2.5) te installeren via Docker swarm (Zie sectie 2.6). Door Docker te gebruiken voor dit project zal er niet veel tijd verloren gaan om de nodige software te installeren. Anderzijds wordt er voor het scrapen wel enige trial en error verwacht totdat alle data correct wordt verwerkt en gepersisteerd wordt. Met behulp van Docker kan een container met enkele commando's verwijderd en terug opgebouwd worden. Een ander voordeel is dat we met behulp van Docker swarm load balancing kunnen voorzien voor webapplicaties, in dit onderzoek kan dit gebruikt worden voor Splash. Voor we van start gaan met de bespreking van het praktische aspect zal er uitleg gegeven worden over de software en tools die er zullen gebruikt worden.

2.1 Webcrawler

Een webcrawler, spider of search engine bot is een computerprogramma dat automatisch data van websites indexeert zodat zoekmachines zoals Google, Yahoo, Duckduckgo... een index van het world wide web kunnen opbouwen en meer relevante links kunnen weergeven als resultaat van de door een eindgebruiker ingegeven zoektermen.

Een webcrawler start met een voorgedefinieerde lijst van urls, deze urls worden ook wel de seeds genoemd. Iedere url in deze lijst wordt één voor één overlopen en alle hyperlinks van deze url worden opgeslagen en aan een lijst toegevoegd van te bezoeken urls. Op deze manier zou deze webcrawler bijna oneindig kunnen doorgaan. Daarom moet een webcrawler zich aan een aantal voorgedefinieerde regels houden. Deze regels duiden bijvoorbeeld aan in welke volgorde een webcrawler moet werken en hoe vaak een bron mag overlopen worden. Zo kan er bijvoorbeeld ingesteld worden dat websites die sneller updates krijgen sneller terug mogen overlopen worden dan andere websites. Een website zoals Wikipedia zal bijvoorbeeld veel meer nieuwe informatie krijgen dan bijvoorbeeld www.hogent.be.

Voor dit onderzoek zullen er vele aandelen, sectoren, valuta's... gescraped worden daarvoor zal het programma gebruik maken van een webcrawler. De webcrawler die hier gebruikt zal worden zal een lijst van vooraf gedefinieerde urls meekrijgen, deze urls zijn de markten waar we interesse in hebben. De webcrawler zal dan voor ieder gevonden beursartikel alle hyperlinks van de historiek ophalen en deze terug overlopen, eenmaal dit allemaal gebeurd is kan er naar het volgende beursartikel gekeken worden tot alles van één voorgedefinieerde url overlopen is.

2.2 Selenium

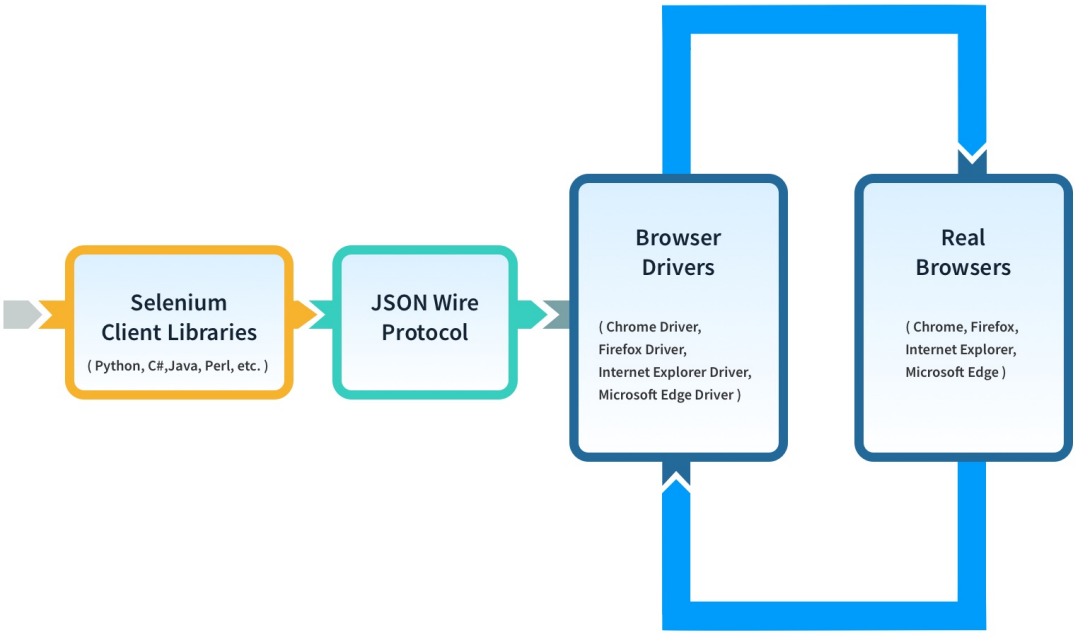
Selenium is een framework dat kan gebruikt worden om testen van webapplicaties te automatiseren. Selenium bestaat uit drie belangrijke onderdelen, Selenium IDE (Zie Sectie 2.2.2) , Selenium Grid (Zie Sectie 2.2.3) en Selenium WebDriver. Alhoewel de eigenlijke use-case van dit framework bedoeld is om testen van webapplicaties te automatiseren wordt Selenium vaak gebruikt om user interaction na te bootsen op webpagina's en dit op deze manier te gebruiken voor webscraping.

Selenium wordt vaak gebruikt bij webscraping omdat er met deze tool zeer eenvoudig user interaction nagebootst kan worden. Het komt meer en meer voor dat er enige interactie met een webpagina vereist is voordat er data, of meer data zal gegenereerd zal worden. Zo hebben Twitter, Facebook of Reddit bijvoorbeeld een infinte scroll hierbij wordt er meer data ingeladen vanaf dat een gebruiker naar beneden scrolt. Als je de data van een webpagina gewoon zou afhalen zou je deze data dus nooit kunnen verkrijgen, met Selenium is dit wel eenvoudig na te bootsen. Ook zit er soms veel data achter een login of acceptatie van cookies, met Selenium zijn deze acties ook perfect uit te voeren.

2.2.1 Selenium WebDriver

Selenium WebDriver is een framework om cross-browser tests uit te voeren, tests om webapplicaties te automatiseren en te verifiëren dat deze testscenario's correct verlopen. Met dit framework kan in een zelfgekozen programmeertaal verscheindene testscenario's uitgeschreven worden voor een webapplicatie, op deze manier kunnen alle mogelijke scenario's snel getest worden. Dit is vooral handig indien er refactoring van de code voorkomt. Voor alle courante browsers zijn er Selenium WebDrivers beschikbaar zoals bijvoorbeeld voor Chrome, Firefox, Microsoft Edge... Ook zijn er libraries beschikbaar voor de meeste populaire programmeertalen zoals Ruby, Python, Java, C#...

De Selenium Client kan dan door middel van data in een JSON formaat instructies sturen naar de gekozen Selenium WebDriver, deze webdriver voert dan op zijn beurt interacties uit met een echte browser. Eén nadeel van Selenium Webdriver is dat deze synchroon werkt, dit betekent dat er slechts één actie tegelijkertijd uitgevoerd kan worden. Ook wordt er tijdens het gebruik een visueel venster geopend op de computer, dit kan in de opties uitgeschakeld worden maar kan voor performantie problemen zorgen. Voor het testen van webapplicaties zijn dit beide geen problemen, maar bij het scrapen van grote hoeveelheden data is alle mogelijke performantie belangrijk waardoor het synchroon werken en het visueel openen van een webbrowser venster wel degelijk nadelig kan zijn voor de performantie van een webscraper. (Unadkat, 2019b)



Figuur 2.1: Grafische voorstelling van de Selenium WebDriver

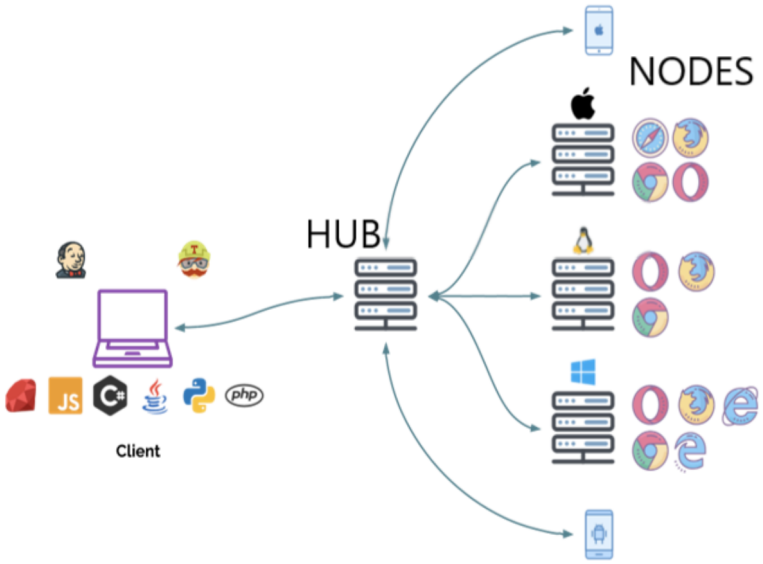
2.2.2 Selenium IDE

De Selenium IDE (Integrated Development Environment) is een tool die een developer in staat stelt om makkelijk test scenario's te creëren. Met deze tool kan je macro's opnemen. Een macro is niets anders dan een lijst van instructies die je opneemt en opslaat. Eenmaal een macro opgeslagen is kan deze steeds opnieuw uitgevoerd worden. Een macro kan bijvoorbeeld bestaan uit de volgende stappen: een webbrowser openen, naar een website navigeren en op deze website inloggen. De Selenium IDE heeft een record-save functie voor macro's waardoor er verschillende scenario's, zonder enige codeer ervaring, kunnen aangemaakt en uitgevoerd worden. (Unadkat, 2019a)

2.2.3 Selenium Grid

Selenium Grid is een tool waarbij meerdere testen parallel op verschillende browsers, machines en besturingssystemen kunnen uitgevoerd worden. Dit wordt gedaan door commando's te versturen naar remote web browser instanties en één server fungeert als hub. Deze hub verstuurt alle testcommando's in JSON formaat, naar de verschillende geregistreerde nodes. Met behulp van deze tool kan je cross-browser compatibiliteit testing zeer eenvoudig laten verlopen.

Het voordeel hiervan is dat er tests op verschillende machines en browsers kunnen worden uitgevoerd vanaf één centrale machine in plaats deze te moeten uitvoeren voor iedere machine of browser individueel. Deze manier van testen wordt vooral gebruikt om meerdere browsers te testen op een efficiënte en tijdsbesparende manier. (Tiwari, 2019)



Figuur 2.2: Grafische voorstelling van de Selenium Grid

2.3 Splash

Splash wordt beschreven als een JavaScript rendering service, het is een webbrowser die gepresenteerd wordt aan de gebruiker als een webapplicatie. Met behulp van een HTTP API kan deze webapplicatie door bijna alle programmeertalen eenvoudig aangesproken worden om websites in data om te zetten door te communiceren met een Splash service met GET of POST requests. Splash is in tegenstelling tot Selenium ontwikkeld met als doeleinde het scrapen van webpagina's. Door het feit dat Splash niet als webbrowser fungeert maar als een webservice is er veel minder overhead dan bij een volledige webbrowser.

Deze tool, is zoals eerder vermeld, ontwikkeld voor het scrapen van data van websites, daardoor zijn er ook allerlei handige features aanwezig voor webscraping. Zo kan je eenvoudig aangeven of je een screenshot wilt van de webpagina, alle netwerk activiteit die plaatsvond tijdens het laden van de website, de ruwe HTML, uitgevoerde scripts, console data... van een webpagina.

Deze service wordt aangesproken met een HTTP API, maar dit is niet voldoende om te definiëren of er eventuele user interaction uitgevoerd moet worden en wat er van een webpagina allemaal moet bewaard worden, er zouden teveel verschillende mogelijkheden zijn om dit enkel met GET of POST requests uit te voeren. Daardoor is Splash voorzien van een scripting taal, deze taal is Lua. Initieel lijkt dit misschien een vreemde keuze omdat Lua een niet zo bekende of veelgebruikte programmeer taal is. Maar zoals Korobov (2014) beschrijft werd er initieel geopteerd om eventueel Python of JavaScript te gebruiken, dit zijn programmeertalen die veel meer gebruikt worden en waar de meeste programmeurs heel wat meer kennis over hebben. Maar doordat het script uitgevoerd zal worden door Splash wordt dit 'sandboxed' uitgevoerd, dit betekent dat de computer die het script uitvoert de applicatie niet vertrouwt. Zoals beschreven door Sagalovskiy (2014) heeft Python hier geen goeie support voor. JavaScript is ook geen goede optie omdat Splash al JavaScript gebruikt voor de integratie van de webbrowser en er teveel problemen kunnen voorkomen waar JavaScript incorrect uitgevoerd zou worden.

Omwille van al deze bovenstaande redenen werd er dus uiteindelijk geopteerd om Lua te gebruiken als scripting taal. Alhoewel Lua, voor velen een onbekende taal is, moet er amper code geschreven worden om een resultaat te bekomen. Ook worden de meeste acties die kunnen uitgevoerd worden met Splash door middel van Lua scripting duidelijk beschreven in de documentatie van Splash. De scripting taal kan wat afschrikwekkend werken voor dit framework, omdat het een volledig nieuwe taal is die moet geleerd worden voor het gebruiken van Splash, maar uiteindelijk is dit een zeer eenvoudige taal en worden er geen grote hoeveelheden code geschreven in deze taal. Maar het feit dat er een scripting taal aanwezig is zorgt wel voor oneindig veel mogelijkheden met een webpagina, zo kun je zoals eerder besproken kiezen welke data er geretourneerd zal worden maar ook of er eventuele interacties met een webpagina moeten uitgevoerd worden zoals naar het einde van een webpagina scrollen, inloggen, een knop indrukken... De mogelijkheden zijn oneindig.

Ook is het mogelijk om verschillende instanties van deze applicatie te definiëren en te gebruiken wat voor het webscrapen van webpagina's voor een grote tijdsreductie kan

zorgen. (Scrapinghub, 2019)

2.4 HtmlAgilityPack

HtmlAgilityPack is een library geschreven in .NET. Het doel van dit framework is om data die in een markup taal geschreven is te formateren. In dit onderzoek zal er heel wat ruwe HTML data gescraped worden van twee verschillende webbronnen. Met deze library kunnen we zeer eenvoudig met behulp van XPATH of XSLT data selecteren uit de HTML die relevant is voor dit onderzoek, deze data opschonen en converteren naar een bruikbaar formaat zodat dit makkelijk gepersisteerd kan worden in de gekozen databank MongoDB (Zie sectie 2.5).

2.5 MongoDB

MongoDB is een databank oplossing die data op een document georiënteerde manier opslaat. Deze documenten worden opgeslagen in een formaat genaamd BSON, deze is bijna identiek aan de syntaxis van JSON-documenten. Bij een traditionele SQL databank zoals Postgres of My SQL moet er vooraf voor ieder object dat van een applicatie gepersisteerd moet worden een schema aangemaakt worden met kolommen en aangegeven worden wat de sleutel van een tabel zal worden waarna alle data als rijen toegevoegd wordt aan een tabel. Bij MongoDB is dit door de document georiënteerde manier veel flexibeler, er moet geen schema vooraf gedefinieerd worden en er kunnen eenvoudig velden toegevoegd of verwijderd worden aan een object. Hierdoor wordt er bijna geen tijd meer gespendeerd aan het voorbereiden van data voor de databank.

MongoDB is opgebouwd uit drie verschillende delen, de databank, collecties en documenten. Een databank is een soort van container voor data, op een MongoDB server kunnen er verschillende databanken bestaan. Voor dit onderzoek zullen er op de server twee verschillende databanken aangemaakt worden namelijk beursduivel en stockmonitor. Een collectie is een deel van een databank en bevat een groep aan documenten. Een collectie is te vergelijken met een SQL tabel maar een groot verschil is dat documenten in een collectie niet verplicht allemaal dezelfde velden moeten hebben. In dit onderzoek zal er per beursartikel een nieuwe collectie aangemaakt worden in de respectievelijke databank. Als laatste zijn er de documenten, deze zijn een verzameling van sleutel-waarde paren, Deze zijn dynamisch en in één document kan de sleutel naam een string bevatten en in een ander document kan dit een integer bevatten. Voor dit project zal er voor ieder beursartikel per dag een nieuw document aangemaakt worden, zo kan er een historiek opgebouwd worden.

MongoDB biedt dus heel wat voordelen om in tegenstelling tot een traditionele SQL database solution zeer snel data te persisteren. Maar voor sterk gerelateerde data is een SQL oplossing nog steeds de beste keuze. De data die uit dit onderzoek gegenereerd zal worden heeft echter geen relaties. Ook is MongoDB horizontaal gemakkelijk schaalbaar. In dit onderzoek zal er heel veel data in een korte tijd gegenereerd worden en mogelijks zou

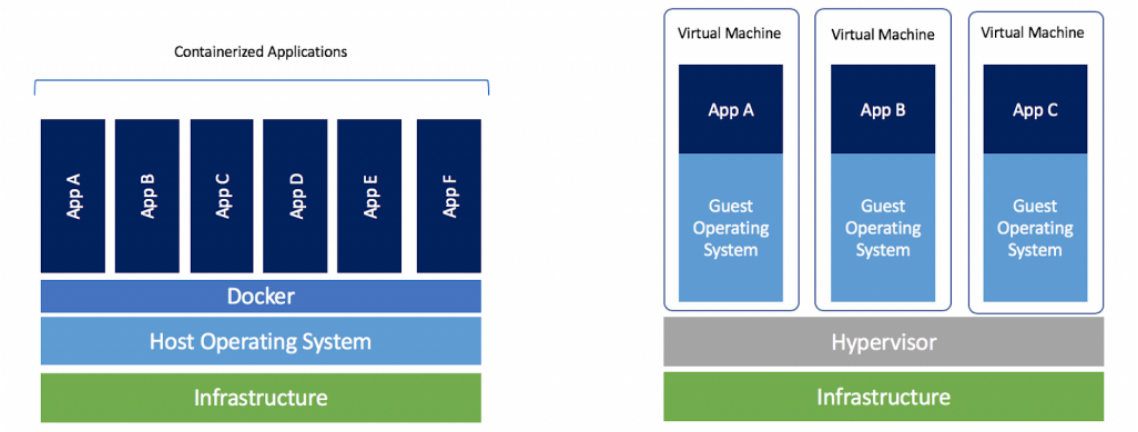
dit voor eventuele locks kunnen zorgen binnen een SQL oplossing terwijl dit bij MongoDB niet kan voorkomen. Door bovenstaande redenen leek MongoDB dus de perfecte oplossing voor deze use case. (Rouse, 2018)

2.6 Docker

Docker is een tool die kan gebruikt worden om applicaties aan te maken, uit te rollen of uit te voeren in containers. Deze containers zorgen ervoor dat alles wat de applicatie nodig heeft aanwezig is. Door deze feature van de containers kan je een nieuwe server of computer die van enkele core applicaties voorzien moet worden zeer snel installeren of updaten.

Een Docker container kan worden gezien als een soort van virtual machine, maar de docker containers zorgen voor heel wat minder overhead vergeleken met virtual machines. Een virtual machine emuleert een volledig besturingssysteem binnen een ander besturingssysteem, waardoor deze virtual machines meer plaats innemen ,meer overhead veroorzaken en veel langer duren om te installeren.

Met Docker kan je met behulp van enkele commando's applicaties zoals MongoDB of PostgreSQL installeren. Op de website van Docker heb je een sectie genaamd Docker hub waar er door de community containers gecreëerd en geüpload worden zodat deze door de rest van de community kunnen gebruikt worden. Een ander voordeel van Docker is dat je een container kunt verwijderen maar alle belangrijke data van die container kunt bewaren, waardoor je zeer makkelijk een applicatie kunt herinstalleren of configuratie en data kunt overzetten naar een nieuwe installatie.



Figuur 2.3: Grafische voorstelling van docker]

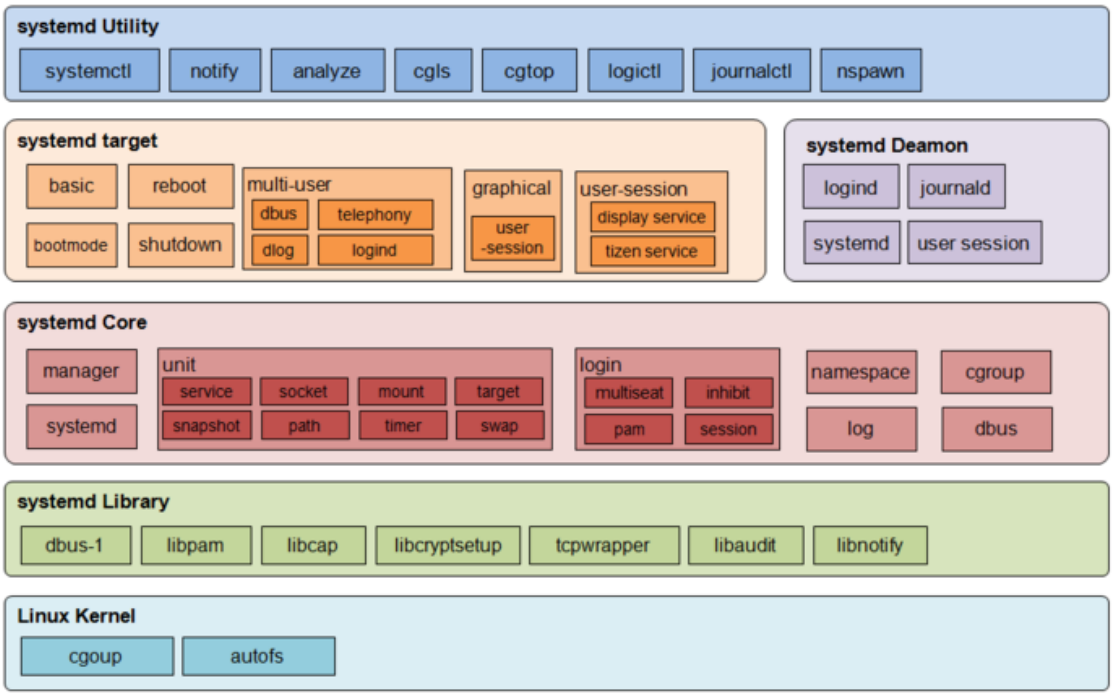
2.6.1 Docker swarm

Met Docker swarm wordt één computer gebruikt als een soort van leader en kunnen er verschillende nodes toegevoegd worden zodat taken over verschillende machines en containers makkelijker kunnen verdeeld worden. Maar Docker swarm kan ook gebruikt worden met één computer. Het voordeel van Docker swarm is dat er services kunnen aangemaakt worden, van een service kunnen er makkelijk verschillende instanties aangemaakt worden zodat er load balancing kan uitgevoerd worden. Zo kan er bijvoorbeeld van één webapplicatie meerdere instanties aangemaakt worden in een service dit worden ook wel replica's genoemd. Van de host computer is er echter maar één aanspreekpunt maar Docker zal zelf zorgen dat er altijd andere instanties van de webapplicatie zullen geserved worden aan de gebruiker. Dit kan er dus voor zorgen dat je een applicatie veel makkelijker kunt schalen.

2.7 Systemd/Timers

Systemd zijn de basis bouwstenen voor een Linux besturingssystemen. Het is een systeem en service manager en wordt gebruikt om de users en user processen te initialiseren.

Systemd/Timers zijn systemd services waarvan het bestand eindigt in .timer. Deze timers zijn een alternatief voor cron. Aan de hand van deze timers is het mogelijk om in te stellen dat een bepaalde taak op de vooraf gespecificeerde tijd moet uitgevoerd worden. Het systeem zorgt er ook voor dat er niet meerdere instanties van dezelfde taak tegelijk uitgevoerd worden. Als een vorige taak nog bezig is zal het proces niet uitgevoerd worden. (online source, 2020)



Figuur 2.4: Grafische voorstelling van Systemd/Timers

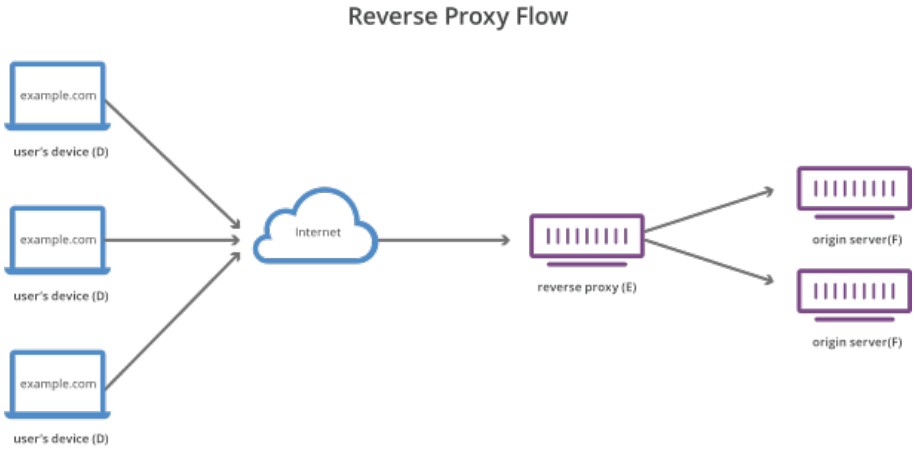
2.8 Mathplotlib

Mathplotlib is een library voor Python. Met behulp van deze library is het mogelijk om makkelijk data te plotten en grafieken of ander visuele tools te maken en zo tot conclusies te komen van de uit het onderzoek gescrapte data.

2.9 NGINX

NGINX is een web server dit kan gebruikt worden als reverse proxy, load balancer,mail proxy en HTTP cache. De software werd initieel ontwikkeld door Igor Sysoev in 2004 om een probleem genaamd 'C10K Problem' op te lossen. Zoals besproken door Kegel (1999) was er geen enkele web server die meer dan tienduizend connecties tegelijkertijd kan afhandelen. Het world wide web was echter een sterk groeiende markt en websites begonnen steeds meer en meer gebruikers aan te trekken. NGINX was de eerste web server die dit probleem oploste. Dit is ook de reden waarom NGINX zeer gericht is op performantie. Zoals door Jarrod (2016) besproken is NGINX tot op vandaag nog steeds één van de snelste web servers.

Voor dit onderzoek zal er vooral gebruik gemaakt worden van de reverse proxy en load balancing functionaliteiten waarover NGINX beschikt. Een reverse proxy is een server die interacties van bijvoorbeeld een webbrowser doorstuurt naar een webserver.



Figuur 2.5: Grafische voorstelling van een reverse proxy

3. Methodologie

3.1 Opbouw van het onderzoek

Voor dit onderzoek zal er data van twee verschillende bronnen verzameld worden. Deze bronnen zijn 'www.stockmonitor.com' en 'www.beursduivel.be'. De website van beursduivel zal gebruikt worden om verschillende relevante en interessante beursindexen te verzamelen en persisteren. Zo zal er data verzameld worden van de volgende beursindexen: de index US500 dit zijn de vijfhonderd belangrijkste aandelen van de Verenigde Staten. Alsook AEX van Nederland, BEL 20 van België, DAX30 van Duitsland, CAC40 van Frankrijk, SMI20 van Zwitserland en LDN100 van het Verenigd Koninkrijk zullen behandeld worden. Ook zullen de verschillende valuta's en rentes te vinden op deze bron verzameld worden. De tweede bron www.stockmonitor.com zal gebruikt om de verschillende sectoren die bestaan op de beurs te verzamelen. Op deze manier kunnen de aandelen die verzameld worden gelinkt worden aan een sector.

Om het proces van te faciliteren van onze webscraper zal er gebruik gemaakt worden van een framework. Vooraleer er dus kon begonnen worden met scrapen moet er eerst voor een framework gekozen dat zal fungeren als de bouwstenen van de webscraper. Er zijn twee verschillende, voor de hand liggende frameworks die er kunnen gebruikt worden om data van een website te scrapen, deze zijn Selenium en Splash.

Selenium is het bekendere framework van de twee, het originele doeleinde van dit framework is het automatiseren van testen voor webbrowsers. Met behulp van tools die Selenium ter beschikking stelt kan een ontwikkelaar eenvoudig een door hem gecreëerde webapplicatie testen. Doordat dit framework ontwikkeld is in functie van het testen van webapplicaties, heeft het heel wat tools ter beschikking om verschillende interacties met een webpagina uit te voeren. Zo kan je met Selenium bijvoorbeeld een knop op een webpa-

gina indrukken, informatie invullen, inloggen en specifieke informatie van een webpagina gaan opzoeken. Het kan dus heel wat user interactions nabootsen.

Een groot nadeel echter van Selenium is dat deze tijdens de loopduur van de applicatie visueel een webbrowser opent op de computer waar iets uitgevoerd wordt. Indien er met meerdere threads gewerkt wordt zal er per thread een webbrowser geopend worden. Dit brengt flink wat meer overhead met zich mee. Selenium biedt wel de mogelijkheid om 'headless' te werken. Dit betekent dat er visueel geen webbrowser meer geopend zal worden, dit werd ook getest voor de mogelijke webscraper maar bleek veel extra problemen teweeg te brengen in verband met het laden van webpagina's.

Het tweede framework, Splash is in tegenstelling tot Selenium ontwikkeld met als enige doeleind om websites te scrapen. Splash is in tegenstelling tot Selenium een applicatie op zich en geen library die in een ander programma kan ingebouwd worden. Doordat Splash geen library is maar een applicatie kan deze eenvoudig met behulp van Docker geïnstalleerd worden. Eenmaal de software geïnstalleerd is voorziet deze lokaal een webapplicatie, deze applicatie kan met behulp van een REST API aangesproken en bestuurd worden. Doordat deze software gebruik maakt van een webapplicatie kunnen makkelijk meerdere instanties klaargezet worden zodat er met Docker makkelijk voor load balancing kan gezorgd worden.

Ondanks dat de meeste bronnen online Selenium gebruiken voor webscraping werd er voor dit onderzoek toch geopteerd om Splash te gebruiken. Dit framework is aanzienlijk minder bekend dan Selenium maar zoals hierboven en in het vorige hoofdstuk besproken heeft Selenium buiten de mogelijkheid tot uitvoeren van user interaction geen grote meerwaarde tot webscraping terwijl Splash dezelfde functionaliteiten en meer te bieden heeft.

De data die dit onderzoek zal voortbrengen zal ook moeten gepersisteerd worden zodat dit later makkelijk kan gebruikt worden voor analyses. In de testfases van dit onderzoek werd alle data die dit onderzoek genereerde opgeslagen in bestanden van het type 'CSV'. Deze bestanden zijn niks meer dan grote tekstbestanden waar alle data gescheiden is door een scheidingsteken, meestal is dit een komma, en per lijn in dit bestand vormt dit een nieuw record. Maar als alle data in één bestand zou opgeslagen worden kan dit prestatie problemen met zich mee brengen. Naarmate de data, en dus het bestand groeit zal het ook steeds langer duren om het bestand te openen en om er data in weg te schrijven. Ook is het gebruik van CSV bestanden tijdens het onderzoek niet overzichtelijk. Omwille van van al deze redenen werd er geopteerd om voor een databank te kiezen. Omwille van de grote hoeveelheden data waartussen er bovendien weinig relaties bestaan werd er gekozen voor MongoDB als databank oplossing.

4. Voorbereiding op het onderzoek

4.1 Ontwikkeling van de webscraper

Vooreerst zal een demoapplicatie ontwikkeld worden in de programmeertaal .NET Core en zal data opgehaald worden van de eerder beschreven bronnen, namelijk de websites 'www.stockmonitor.com' en 'www.beursduivel.be'. De taal .NET Core werd gekozen omdat deze zeer eenvoudig is om een project te bootstrappen en compatibel is met bijna alle besturingssystemen wat een must is aangezien de applicatie in een Linux omgeving zal uitgevoerd worden. Ook is er voor .NET Core een MongoDB library beschikbaar, deze stelt ons in staat om eenvoudig data te persisteren.

De geschreven applicatie zal de bronnen scrapen volgens de principes van een webcrawler. De applicatie krijgt een lijst van vooraf opgestelde links mee. Deze lijst zal de applicatie één voor één overlopen en verwerken. Ook moet de applicatie van ieder aandeel dat gevonden wordt een link terugvinden naar de pagina waar de historiek wordt weergegeven. Alle webpagina's die data over de historiek van het aandeel bevatten moet de applicatie overlopen, verwerken en persisteren. Eenmaal alle historische data van een aandeel verwerkt zijn mag de applicatie op zoek gaan naar het volgende aandeel op de pagina. Eenmaal alle aandelen van een link verwerkt zijn wordt de volgende link van de lijst ingeladen en dezelfde logica terug herhaald.

Voor het verkrijgen van data moet de webscraper samenwerken met Splash, deze webapplicatie fungeert als een browser die aangesproken kan wordt via een HTTP API. De geschreven webscraper zal voor iedere link die hij terugvindt in zijn lijst een GET request versturen naar Splash. Deze zal op zijn beurt de webpagina laden en de onverwerkte HTML van deze pagina retourneren naar de webscraper. Voor ieder aandeel dat de webscraper uit deze verkregen HTML kan halen zullen er vervolgens verschillende requests gebeuren

om de historiek van dit aandeel te verkrijgen. Voor dit onderzoek zal er aan iedere GET request ook twee parameters meegegeven worden. Zo zal de 'Wait' parameter op tien seconden geplaatst worden, dit betekent dat er tot tien seconden gewacht zal worden voor eventuele updates op een webpagina. Anderzijds wordt de 'Timeout' parameter op zijn maximum waarde van negentig seconden geplaatst. Beide van deze parameters staan op vrij hoge waarden. Deze keuze is bewust gemaakt omdat er veel requests zullen verzonden worden naar één website. Indien er gekozen werd om de webscraper zo snel mogelijk te laten werken kwam vaak voor dat er geen data kon geladen worden van een link. De keuzes voor deze parameters zorgen er eveneens voor dat de applicatie minder gevoelig is voor een opeens tragere internet connectie.

De applicatie zal zoals hierboven beschreven veel ruwe HTML data binnenkrijgen. Met deze data kunnen we niet veel aanvangen, we willen enkel de data relevant voor dit onderzoek overhouden van deze HTML. Om deze data op te schonen zal er gebruik gemaakt worden van een .NET library genaamd HtmlAgilityPack, met behulp van deze library kan er zeer eenvoudig met XPATH data geselecteerd worden die interessant is voor het onderzoek, deze data naar een .NET Object omzetten en dit object vervolgens persisteren naar de gekozen databank oplossing MongoDB.

Eenmaal de applicatie zonder problemen al de data kon verwerken werd er geprobeerd om de snelheid van de webscraper te verhogen. Doordat de applicatie maar één link van de lijst tegelijkertijd verwerkt was dit een vrij traag en saai proces. De computer moest ook ver van op volle kracht werken om de set van links te verwerken. De grootste bottleneck was momenteel wachten tot er data geretourneerd werd, dit snel verwerken en deze lus herhalen. Daarom werd er ondersteuning toegevoegd aan de webscraper om asynchroon te werken, op deze manier kunnen er meerdere webpagina's tegelijkertijd verwerkt worden. Aangezien alle webpagina's die moeten verwerkt worden initieel door Splash moeten geladen worden en dan pas terug geretourneerd worden naar de webscraper zorgt dit ervoor dat deze webapplicatie onder veel meer druk komt te staan. Dit zorgt ervoor dat de applicatie niet veel sneller werkt ook al worden er in theorie meerdere requests tegelijkertijd behandelt. Ook zorgde dit ervoor dat Splash veel instabieler werd en er meer foutmeldingen werden gegeven. Maar zoals eerder beschreven werd er voor de installatie van Splash gebruik gemaakt van Docker, er werd geopteerd om een tweede container van Splash aan te maken met behulp van Docker. In de webscraper werd vervolgens geprobeerd om de requests te verdelen over deze twee containers. Deze logica is echter niet eenvoudig om te programmeren. Om dit op te lossen werd er zoals door Nelson (2016) besproken gekozen om met behulp van Docker swarm en NGINX load balancing te implementeren voor Splash. Door het gebruik van load balancing moet er in de .NET applicatie geen rekening meer gehouden worden met de verschillende containers die actief zijn van Splash. Al deze logica zal door NGINX en Docker swarm toegepast worden. Met deze technologie is er onmiddellijk veel meer flexibiliteit, indien een gebruiker meerdere webscrapers tegelijk wil gebruiken of sneller data wil scrapen kunnen er makkelijk containers toegevoegd of verwijderd worden zonder dat er aanpassingen moeten gebeuren in een webscraper.

Voor dit onderzoek werd er een 'Stack' aangemaakt in Docker met vijftien verschillende Splash containers en een MongoDB container. Met behulp van een NGINX container werd er een reverse proxy ingesteld naar de verschillende Splash containers. Deze kunnen

dan vanuit de .NET Core applicatie op hun beurt aangesproken worden via het domein 'splash.mathiascloet.com'. Splash voorziet geen authenticatie op zijn webapplicatie dus werden enkel IP-adressen van het lokaal netwerk toegelaten voor veiligheidsredenen. Hieronder staat de configuratie die gebruikt werd om deze 'Stack' voor de webscraper aan te maken. In deze configuratie staat er ingesteld dat we ten alle tijden vijftien verschillende containers actief willen, indien er een container faalt zal er onmiddellijk gekeken worden om een nieuwe container te starten. Zo mag er in theorie nooit een moment zijn waarop de Splash webapplicatie niet kan aangesproken worden.

```
version: "3.7"
services:
  splash:
    image: scrapinghub/splash
    ports:
      - 8050:8050
    deploy:
      replicas: 15
      restart_policy:
        max_attempts: 3
        condition: on-failure
    networks:
      - bachproef
  mongo:
    image: mongo:latest
    ports:
      - 27017:27017
    volumes:
      - mongo_data:/data/db
    networks:
      - bachproef
volumes:
  mongo_data:
networks:
  bachproef:
```

Figuur 4.1: Script gebruikt voor Docker swarm

4.2 Voorbereiden op het onderzoek

Het grote voordeel van de bronnen 'www.beursduivel.be' en 'www.stockmonitor.com' is dat alle pagina's die gescraped worden zeer sterk op elkaar gelijken. Dit zorgt ervoor dat als één webpagina feilloos kan gedownload, geformatteerd en gepersisteerd worden alle opvolgende pagina's ook zonder problemen kunnen gescraped worden. Zoals eerder besproken werd er gekozen om data in MongoDB te persisteren omdat dit een beter overzicht en performantie biedt in plaats van CSV bestanden te utiliseren. Om analyses uit te voeren met Python is het echter wel heel wat eenvoudiger om een CSV bestand te

gebruiken als dataset. De MongoDB Compass applicatie ondersteunt echter enkel het exporteren van één collectie per actie. Maar met behulp van een script geschreven in Bash was het mogelijk om alle collecties van 'www.beursduivel.be' pijnloos te exporteren naar CSV bestanden en al deze bestanden te combineren naar één eindbestand. Van de bron www.stockmonitor.com was er maar één collectie aanwezig in de databank dus dit kon manueel geëxporteerd worden.

Het doel van dit onderzoek is dat we met deze twee CSV bestanden al de deelonderzoeksvragen van dit onderzoek kunnen beantwoorden en op deze manier illustreren dat webscraping een goede manier is om snel en eenvoudig grote hoeveelheden data te downloaden. Hieronder de kolommen van de gebruikte CSV bestanden.

Kolommen CSV bestand beursduivel.be

id unieke identificatie mongodb object.
Currency Munt van het aandeel
Name Volledige naam van het aandeel.
StockName Verkorte naam.
Index de index waartoe het aandeel behoort.
StockNumber het unieke nummer van het aandeel.
DateTime De datum waarop de koers van het aandeel genomen werd.
Difference Het verschil in punten vergeleken met de vorige datum.
PercentageDifference Het verschil in percentage vergeleken met de vorige datum.
Volume Volume verhandelt op de gespecificeerde datum.
High Hoogste waarde van het aandeel op de gespecificeerde datum.
Low Laagste waarde van het aandeel op een dag.
Open Waarde waarop het aandeel die dag geopend is.
Close Waarde waarop het aandeel die dag geëindigd heeft.

Kolommen CSV bestand stockmonitor.com

StockName Verkorte naam van een aandeel.
SectorName Naam van de sector dit aandeel toe behoort.

5. Onderzoek

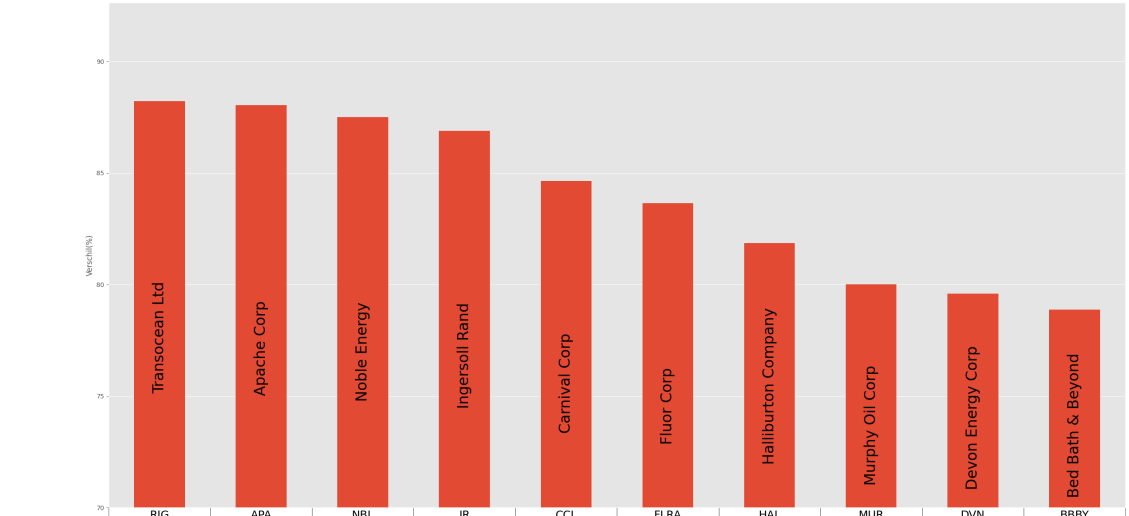
5.1 Opbouw van het onderzoek

Zoals besproken in het vorige hoofdstuk zal iedere collectie die momenteel in MongoDB gepersisteerd is geëxporteerd worden naar een CSV bestand, de bestandsnaam van dit CSV bestand zal tijdelijk de naam van de collectie in MongoDB overnemen. Deze export zal gebeuren met behulp van een script geschreven in Bash. Nadat alle collecties geëxporteerd zijn naar bestanden van het type CSV zullen vervolgens al deze bestanden samengevoegd worden tot één algemeen bestand. In dit bestand zal alle data die gescraped is van beursduivel.be te vinden zijn. Op deze manier kan de data makkelijk gemanipuleerd worden met behulp van Python en enkele Python libraries zoals Pandas en Mathplotlib.

Voor dit onderzoek hebben we de volgende indexen afgehaald van beursduivel.be, de US500 dit zijn de vijfhonderd belangrijkste aandelen van de Verenigde Staten. De AEX van Nederland, BEL20 van België, DAX30 van Duitsland, CAC40 van Frankrijk, SMI20 van Zwitserland en LDN100 van het Verenigd Koninkrijk. Deze indexen zijn gekozen omdat deze de belangrijkere aandelen bevatten. Indien we alle aandelen zouden opnemen in dit onderzoek zouden er teveel kleine bedrijven een vertekend beeld kunnen geven van onze resultaten. Daarom hebben we ons beperkt tot de belangrijkste aandelen qua beurskapitalisatie per land. Wat weerspiegeld word in de bovenvermelde indexen. Zo bevat de BEL 20 de twintig belangrijkste aandelen qua beurskapitalisatie van België. Voor de andere indexen is dit analoog. Beurskapitalisatie is de prijs van het aandeel vermenigvuldigd met het volume van het aandeel.

Met behulp van Python kunnen we met enkele lijntjes code zeer eenvoudig de meest volatiele aandelen uit onze set data halen. Voor de onderstaande grafiek werd het verschil berekend tussen de hoogste en laagste koers die een aandeel bereikt heeft tussen 1 februari

2020 en 15 mei 2020. Omdat de prijs tussen verschillende aandelen zeer sterk kan variëren werd er gekozen om het verschil uit te drukken in percentages. Anders zou deze data niet veel betekenen. Als een aandeel zoals Amazon dat normaal ongeveer rond de tweeduizend punten staat met tien punten zakt is dit veel minder serieus dan een aandeel zoals KBC of Bpost dat met tien punten zou zakken. Door het verschil procentueel uit te drukken kan er een duidelijk en eerlijk beeld worden weergegeven van de data.



	RIG	APA	NBL	IR	CCL	FLRA	HAL	MUR	DVN	BBBY
Verschil (%)	88.28 %	88.03 %	87.49 %	86.89 %	84.64 %	83.65 %	81.84 %	80.00 %	79.58 %	78.86 %
Minimum koers	0.84	4.02	3.02	19.22	7.97	3.40	4.61	5.60	5.41	3.56
Maximum koers	7.17	33.59	24.15	146.55	51.90	20.79	25.39	28.00	26.49	16.84

Figuur 5.1: De tien meest volatiele aandelen tussen 01-01-2020 en 15-05-2020

Bijvoorbeeld de koers van het energie aandeel Transocean (RIG) is gedurende deze periode het meest volatiele aandeel geweest. Deze koers is dan ook met bijna 80 procent gezakt. Een verklaring voor deze extreme beweging is dat dit aandeel actief is in de energie sector waar we van weten dat in de corona periode de olieprijzen en alle olie toeleverende bedrijven enorm in waarde zijn gezakt. Een ander voorbeeld van extreme negatieve volatiliteit is te vinden in de consumer cyclical sector, met name het aandeel Carnival Corporation (CCL). Dit bedrijf is actief in cruiseschepen. Het hoeft geen betoog dat ook deze sector enorm heeft afgezien van alle lockdown-maatregelen tijden de corona crisis.

Voor dit onderzoek hebben we ook de bron 'www.stockmonitor.com' gescraped. De bron 'www.beursduivel.be', waar al onze data vandaan komt in verband met aandelen bevatte geen informatie over tot welke sector een aandeel behoorde. Door de data van stockmonitor.com en beursduivel.be te combineren was het echter mogelijk om bijna alle aandelen aan de hand van hun tickercode toe te wijzen aan een sector. Een tickercode is de afkorting van een aandeel. Bijvoorbeeld een heel gekend aandeel Microsoft Corporation heeft als tickercode MSFT. Door deze link te maken was het eenvoudig om ook gegevens op te halen in verband met de prestaties per sector. De bron stockmonitor verdeelde alle aandelen onder in tien verschillende sectoren. Hieronder een korte beschrijving van deze tien sectoren.

Basic Materials Deze sector bevat aandelen met als hoofdbedrijfsactiviteit het ontdekken,

ontwikkelen en verwerken van grondstoffen. Enkele voorbeelden hiervan zijn bijvoorbeeld olie, goud en steen. Niet alle bedrijven die met grondstoffen werken zitten in deze sector, een bedrijf dat grondstoffen mijnt zit in deze sector maar bijvoorbeeld een juwelier niet. Bijvoorbeeld Glencore plc (GLEN.L) of Harmony Gold Mining Company Limited (HMY) behoren tot deze sector. (Kopp, 2019)

Communication Services Deze sector bevat aandelen die telefoon en internet services aanbieden. Deze sector bevat ook producenten van films en tv shows. Enkele bedrijven zijn bijvoorbeeld Netflix (NFLX), Facebook Inc (FB) en Alphabet Inc. (GOOGL). (Johnston, 2020)

Consumer Cyclical Deze sector is sterk afhankelijk van de huidige prestaties van de economie. De sector moet gezien worden als goederen en services die een luxe zijn. Voorbeelden van industrieën in deze sector zijn immobiliën, recreatie, toerisme, detailhandel en productie van wagens. Bijvoorbeeld Carnival Corporation (CCL) is een bedrijf dat in de toeristische sector actief is als uitbater van cruiseschepen. Deze sector kan nogmaals onderverdeeld worden in duurzame en niet duurzame secties. Zo worden bedrijven die goederen produceren voor consumptie, zoals kleren, eten... beschouwd als niet duurzaam. Bedrijven die goederen produceren die een langere tijd zullen bestaan worden gezien als de duurzame sectie.(Hayes, 2020)

Consumer Defensive Deze sector zijn bedrijven die eten, drank, huishoud producten en tabak produceren maar ook onderwijs behoort tot deze sector. Colruyt (COLR.BR) en AB Inbev (ABI.BR) behoren bijvoorbeeld tot deze sector.

Financial Services Dit zijn bedrijven die financiële services verlenen aan mensen of bedrijven. deze sector bevat voornamelijk banken en investeringsbedrijven. Deze sector bestaat voornamelijk uit grotere bedrijven maar bevat ook enkele kleinere bedrijven.

Energy Deze sector bevat bedrijven die energie produceren of verdelen. Deze kan zoals de consumer cyclical sector ook nog eens onderverdeeld worden in hernieuwbaar en niet-hernieuwbaar.

Healthcare Deze sector bevat bedrijven die medische diensten, medische apparatuur of geneesmiddelen produceren. Bijvoorbeeld Gilead Sciences, Inc. (GILD) is een bedrijf actief in deze sector. Welke trouwens een positieve corona impact had door de ontwikkelen van mogelijke vaccins tegen deze ziekte.

Industrials Dit zijn bedrijven die commerciële of industriële producten of services aanbieden. Deze sector is onderverdeeld in drie verschillende secties namelijk investeringsgoederen, zakelijke dienstverlening en vervoer. De sectie investeringsgoederen bestaat bijvoorbeeld uit bedrijven die goederen produceren voor constructie, machines zoals tractors, heftruck... Enkele onderdelen uit de zakelijke dienstverlening zijn bijvoorbeeld schoonmaakbedrijven, gevangenissen, veiligheidssoftware, catering, sociaal secretariaat, IT-gerelateerde adviesbureaus... Vervoer wordt gezien als de trein, vliegtuig vrachtschepen... Voorbeelden van bedrijven in deze sectoren zijn Caterpillar Inc. (CAT) en The Boeing Company (BA).(Miller, 2019)

Technology Deze sector bevat bedrijven die software, electronica, computers of diensten gerelateerd aan technologie produceren. Voorbeelden van bedrijven in deze sectoren zijn Apple Inc. (AAPL), Microsoft Corporation (MSFT) of NVIDIA Corporation (NVDA).

Utilities Dit zijn nutsbedrijven die basisbehoeften zoals water, gas, afvoer... voorzien.

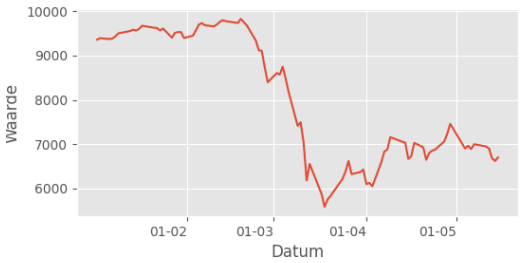
Doordat dit nutsbedrijven zijn zijn deze sterk gereguleerd en worden ze vaak gezien als een langetermijninvestering. Voorbeelden van bedrijven in deze sector zijn Bpost SA/NV (BPOST.BR) en ENGIE SA (ENGI.PA).

Doordat we de meeste aandelen verkregen uit dit project kunnen linken aan hun respectievelijke sector met behulp van de data verkregen uit de bron www.stockmonitor.com is het zeer eenvoudig om de aandelen te groeperen per sector en de som te nemen van de 'Close' van de groepering. Op deze manier kunnen we een overzicht krijgen van de prestaties per sector tussen een gekozen periode. In de grafieken hieronder is er gekozen om data te selecteren van 1 januari 2020 tot en met 15 mei 2020. Deze data zijn gekozen omdat er op 1 januari 2020 nog geen sprake was van corona en de markten nog niet beïnvloedt waren, de einddatum van 15 mei 2020 is de laatste dag dat er voor dit onderzoek data gescraped is van de webbron www.beursduivel.be.

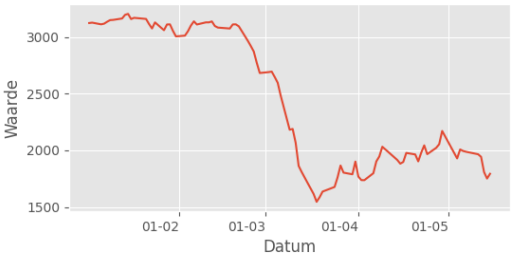
Uit onderstaande grafieken kunnen we duidelijk merken dat iedere sector geimpacteerd was in negatieve zin door het uitbreken van de corona pandemie. Iedere sector vertoont immers een scherpe daling rond het midden van maart. Echter het herstel sedert het dieptepunt vertoont aanzienlijke verschillen per sector. De zwaarst getroffen sectoren zijn de energie sector, de financiële sector, de industriële sector, de nutsbedrijven en de cyclische consumptie goederen. De sectoren het snelst terug heropleven zijn de healthcare sector, de technology en de consumer defensive sector. De absolute winnaar is duidelijk de healthcare sector. Wat enigszins logisch is door de enorme zoektocht naar een vaccin voor de uitgebroken pandemie. Alsook een immense vraag naar farmaceutische producten en nood aan medische verzorging en apparatuur. Het grootste slachtoffer van het stilleggen van het economisch leven is de energie sector. Dit kan tevens logisch verklaard worden doordat de vraag naar energie quasi was stilgevallen.

Uit onderstaande grafieken kunnen we zonder twijfel concluderen dat alle verschillende sectoren op hetzelfde moment zijn beginnen reageren op de effecten van de corona crisis. De reactie was van eind februari tot midden maart voor iedere sector in dalende lijn. Wel is er ook duidelijk te zien dat de sectoren die voornamelijk bestaan uit industrie vergeleken met sectoren die vooral uit diensten bestaan relatief gespaard zijn gebleven. Dit komt vermoedelijk doordat de industrie heeft kunnen blijven toeleveren terwijl de dienstensector nog zwaarder getroffen is geweest voornamelijk toerisme en horeca, deze zijn vertegenwoordigd door de consumer cyclical sector.

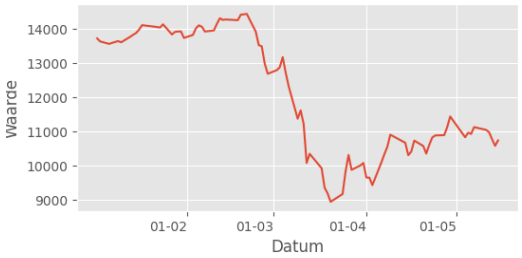
De meeste sectoren zijn zich vrij snel aan het herstellen, gezondheidszorg heeft ondertussen al een hogere beurskoers dan voor de corona.



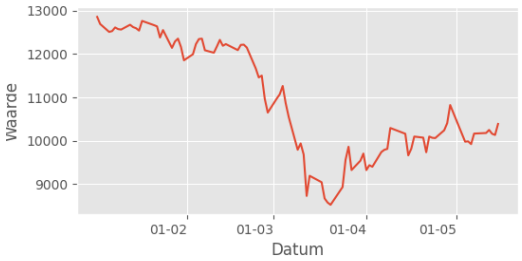
(a) Financial Services sector



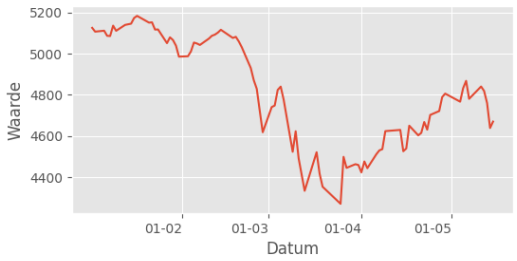
(b) Energy sector



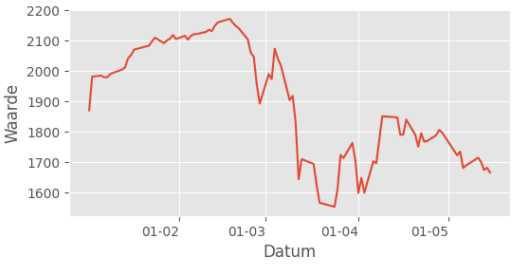
(c) Industrials sector



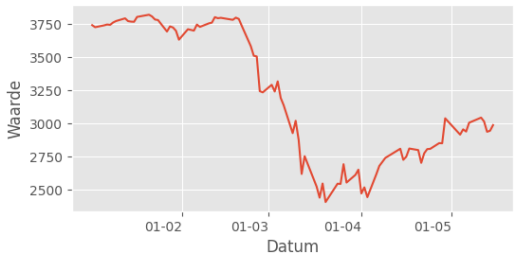
(d) Basic Materials sector



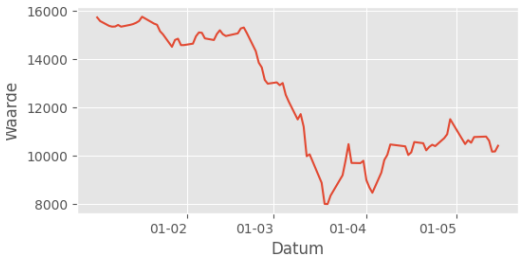
(e) Consumer Defensive sector



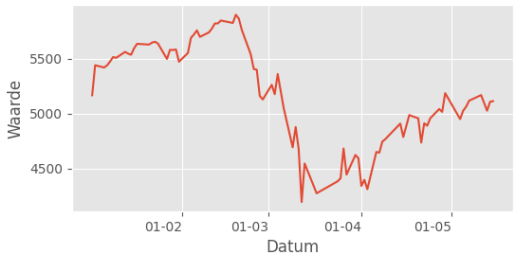
(f) Utilities sector



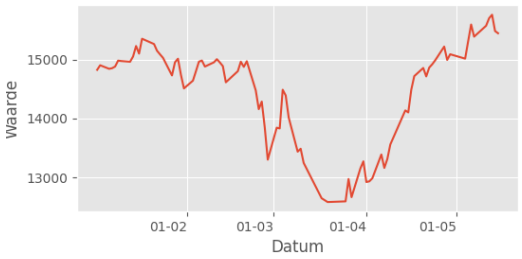
(g) Communication sector



(h) Consumer Cyclical sector



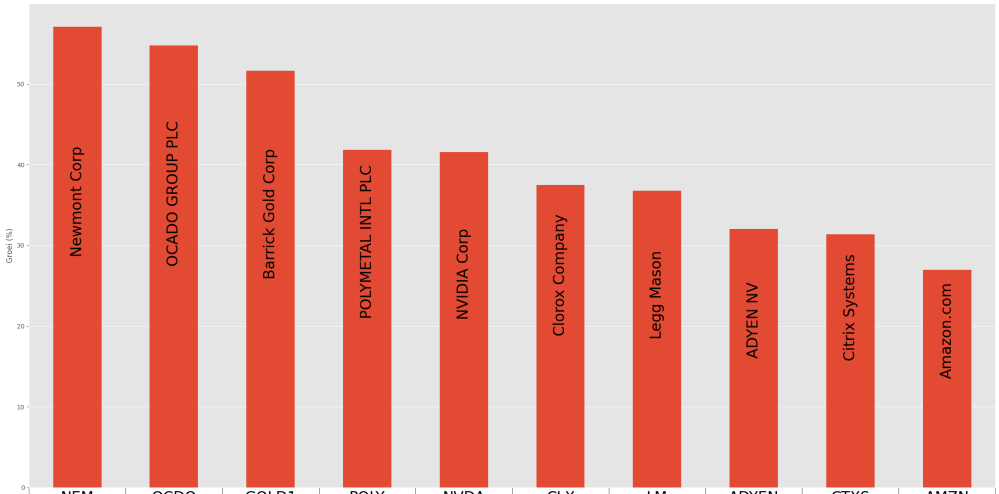
(i) Technology sector



(j) Healthcare sector

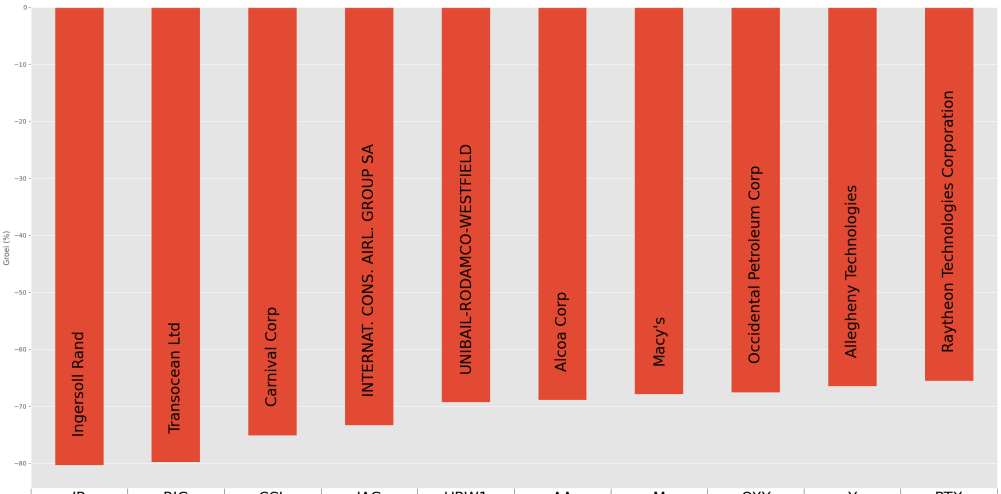
Figuur 5.2: Evolutie van de sectoren tussen 01-01-2020 en 15-05-2020.

Opnieuw met behulp van Python zijn er nu twee grafieken gemaakt. Deze grafieken geven weer welke aandelen de meeste groei hebben gehad over de periode van 1 januari 2020 tot 15 mei 2020 en welke het meeste verlies hebben geleden door de corona crisis. Een van de zwaarst getroffen aandelen is bijvoorbeeld het aluminium bedrijf Alcoa Corporation (AA). Dit bedrijf heeft immers een daling van 68,91 procent achter de rug. Logisch gezien aluminium een basis grondstof is voor heel wat producten en halffabricaten. Het goudmijn bedrijf Barrick Gold Corporation (GOLD1) behoort echter tot de winnaars niettegenstaande dat dit ook tot de basic materials sector behoort. Maar goud is nu eenmaal een toevluchtsoord in barre corona tijden. Uit beide grafieken en onderzochte data kun je zo gemakkelijk de grote winnaars als verliezers binnen het tijdsinterval selecteren.



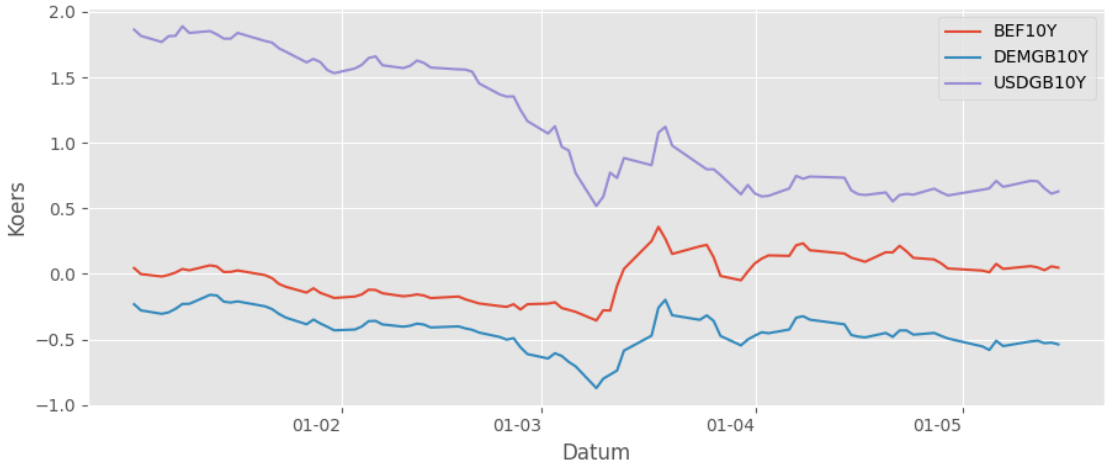
	NEM	OCDO	GOLD1	POLY	NVDA	CLX	LM	ADYEN	CTXS	AMZN
Groei (%)	57.10 %	54.78 %	51.65 %	41.82 %	41.57 %	37.49 %	36.79 %	32.04 %	31.39 %	26.96 %
koers 01/01/2020	43.22	1260.50	18.49	1192.00	239.91	152.61	36.45	744.60	111.67	1898.01
koers 15/05/2020	67.90	1951.00	28.04	1690.50	339.63	209.83	49.86	983.20	146.72	2409.78

Figuur 5.3: De aandelen die het meest gestegen zijn tussen 01-01-2020 en 15-05-2020

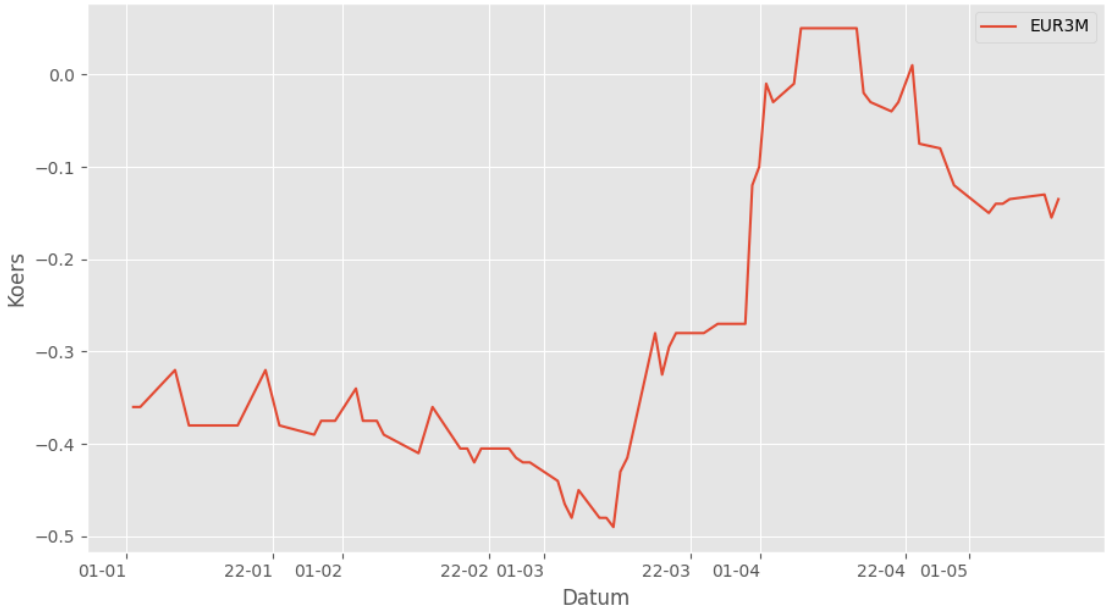


	IR	RIG	CCL	IAG	URW1	AA	M	OXY	Y	RTX
Groei (%)	-80.33 %	-79.80 %	-75.09 %	-73.35 %	-69.28 %	-68.91 %	-67.86 %	-67.54 %	-66.50 %	-65.57 %
koers 01/01/2020	134.35	6.93	51.31	635.40	141.45	21.42	16.52	42.58	20.60	153.14
koers 15/05/2020	26.42	1.40	12.78	169.35	43.46	6.66	5.31	13.82	6.90	52.73

Figuur 5.4: De aandelen die het meest gezakt zijn tussen 01-01-2020 en 15-05-2020



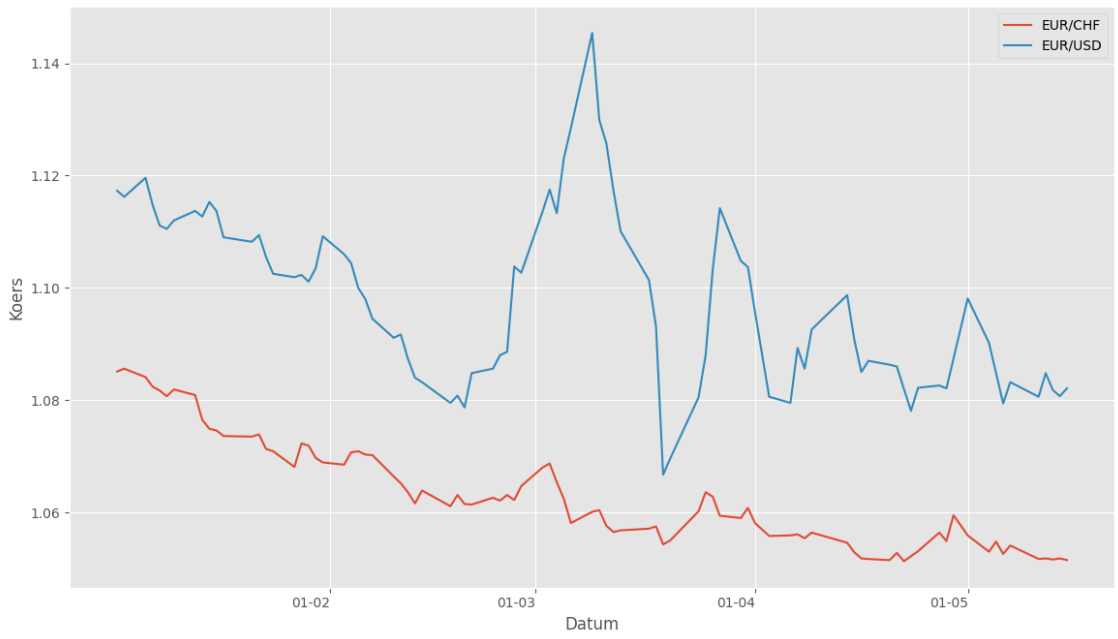
Figuur 5.5: Koers lange-termijn rente Duitsland (DEMGB10Y), België (BEF10Y) en Verenigde staten (USDGB10Y) tussen 01-01-2020 en 15-05-2020



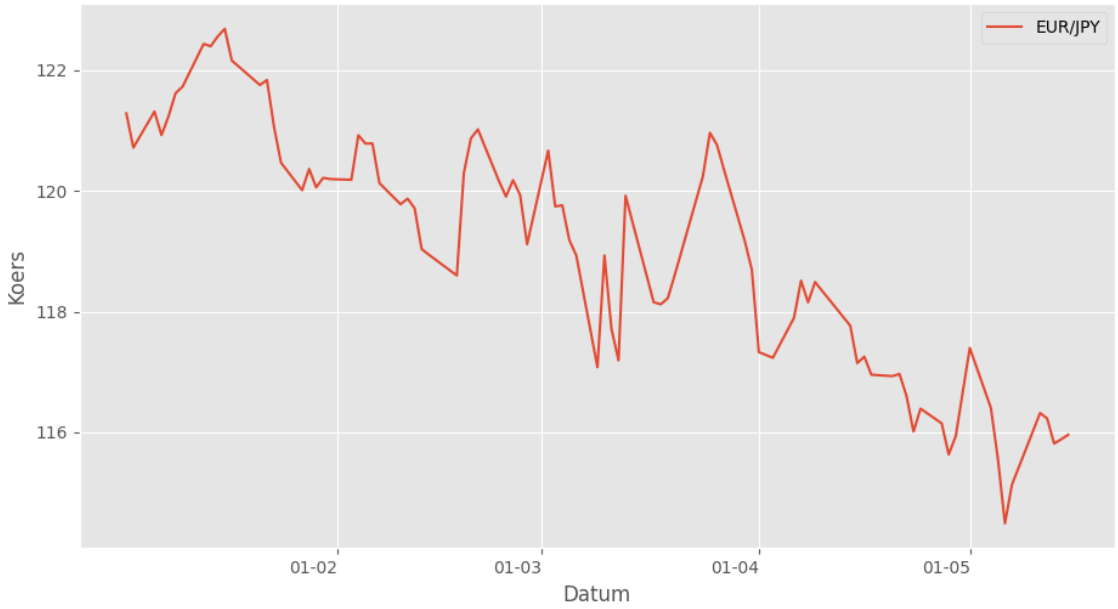
Figuur 5.6: Koers korte-termijn rente Euribor 01-01-2020 en 15-05-2020

Ook op de evolutie van de rente had de corona crisis een impact, dit wordt weerspiegeld door de twee bovenstaande grafieken. De belangrijkste referentierente in Europa is de Duitse Bund en in Amerika is dat de Treasury op tien jaar. Dit zijn lange-termijn rentes. Als referentie voor de korte-termijn rente nemen we de Euribor rente op drie maand, dit is de evolutie van tien jarige rente. Dit is altijd een goede voorspeller geweest van de beurs en de economische groei. De korte-termijn rente daarentegen is een weerspiegeling van het beleid van de centrale banken. In onzekere periodes dalen beide rentes. Bovendien is Duitsland een toevluchtsoord en wordt als een kwalitatief zeer gezond land beschouwd. Men is vandaag bereid om geld te betalen om zijn geld te parkeren in bijvoorbeeld Duitsland. Door

de data van deze verschillende looptijden in deze drie verschillende regio's te visualiseren kunnen we ook de impact meten van corona over de periode van 1 januari 2020 tot 15 mei 2020. Er is duidelijk te merken dat extreme onzekerheid en recessiegevaar zijn impact heeft op de rente. Het vooruitzicht van werkloosheid, economische achteruitgang, recessie, eventueel depressie doet de rentes sterk afnemen. Nadien herstelt de rente opnieuw geleidelijk door het vooruitzicht van een mogelijk vaccin, het ingrijpen van de diverse overheden met ondersteunende maatregelen, geleidelijk betere beurskoersen en vooruitzichten op het einde van de lockdowns. Er is een verschil na de extreme daling vanaf midden maart. De rente in Amerika blijft laag doordat heel wat marktpartijen in Amerika hun geld willen parkeren terwijl in Europa de risicopremie stijgt in zowel België als Duitsland. Omdat de vooruitzichten in de eerste maanden slechter waren voor Europa dan Amerika. Besproken door (Axa, 2020) en (Blekemolen, 2020).



Figuur 5.7: Muntkoers EUR/USD en EUR/CHF 01-01-2020 en 15-05-2020



Figuur 5.8: Muntkoers EUR/JPY tussen 01-01-2020 en 15-05-2020

Ook op de valuta heeft corona een impact zoals te zien op de twee bovenstaande grafieken. De euro is tot midden maart serieus gezakt versus de dollar. Omdat de Amerikaanse munt als belangrijkste munt ter wereld tevens beschouwd wordt als veilige haven. De reeds vermelde treasuries, namelijk obligaties in dollars zijn een zeer liquide belegging in tijden van onzekerheid. De kapitaalstromen richting Amerika doen de dollar stijgen. De snelle ingrepen naar aanleiding van de corona crisis en snelle interventie van de Amerikaanse overheid met immense steunmaatregelen zorgen voor geruststelling en opnieuw stijgende koersen in beurs en valuta. Andere toevluchtsoorden in tijden van onzekerheid en dreigende economische achteruitgang zijn de Japanse yen en Zwitserse frank. Ook daar is op de beweging van de grafieken duidelijk te zien dat de euro sterk daalde versus deze twee veilige munten. Nadien volgt weer een geleidelijke stijging van de euro als blijkt dat ook in Europa de noodzakelijke maatregelen zullen getroffen worden door de overheden.

6. Conclusie

Webscraping is handig hulpmiddel om op een snelle en efficiënte manier data te catalogeren en daaruit gerichte beslissingen te nemen. Toegepast op ons onderzoek naar het beurssentiment tijdens de corona crisis kan dit een handige tool zijn om beleggingsbeslissingen te nemen. Bijvoorbeeld in de bancaire sector zou dit kunnen gebruikt worden door analisten welke een buy, hold of sell strategie moeten bepalen. Dit kan gebruikt worden als extra ondersteunend materiaal om de juiste sectoren en aandelen te kiezen om op te nemen in beleggingsfondsen. Ook voor valuta handelaars en bedrijven welke export gericht werken kan dit een meerwaarde bieden. Alsook voor de financiële directeurs bij het nemen van investeringsbeslissingen. De toepassing die werd gebruikt was om ons onderzoek te staven was natuurlijk razend actueel.

Het internet is een bron van oneindig veel data, met webscraping is het mogelijk om deze data snel te downloaden en te formatteren zodat deze kan gebruikt worden voor andere doeleinden. Bedrijven zouden met behulp van dit onderzoek moeten inzien dat hun bedrijf voordeel kan halen uit de immense flow van data en bronnen die publiek beschikbaar zijn op het web. Deze data kan gezien worden als een grondstof voor hun bedrijf om de juiste beslissingen of conclusies te nemen.

Een nadeel van webscraping is dat de applicatie steeds een manuele interventie vereist wanneer van bron veranderd wordt. Wat niet kan geconcludeerd worden uit dit onderzoek is of de data overal even publiek beschikbaar en toegankelijk is.

In mijn voorwoord had ik vermeld dat dat ik in het verleden al enkele keren een poging had gewaagd tot webscraping maar zonder enig resultaat. Door deze scriptie ben ik al beter gewapend om met de juiste technologie en juiste selectie van bronnen en data dit toe te passen op praktijkvoorbeelden.

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Webscraping is een manier waarbij software gebruikt wordt om informatie van bv. een twitterfeed op te halen, maar doordat deze data dynamisch wordt weergegeven betekent dit dat wanneer je naar een webpagina surft er slechts een deel van deze data geladen zal worden. Enkel door interactie van de eindgebruiker kan er extra data worden opgehaald. Bedrijven maken het door methodes zoals dit steeds moeilijker om hun data te scrapen en te gebruiken in projecten want 'Big Data' is geld waard, maar deze 'Big Data' is steeds belangrijker voor de creatie van applicaties die gebruik maken van AI (Artificiële Intelligentie). De doelstelling van het onderzoek is om verschillende programmeertalen en frameworks te testen voor het ophalen van dynamisch geladen data.

A.2 Literatuurstudie

Er is een onderzoek door (Legaspi, 2016) die onderzoekt hoe je een pagina die dynamisch geladen wordt het best kunt scrapen voor data. In het onderzoek van (Legaspi, 2016) wordt er gebruik gemaakt van IntelIJ, HTMLUnit , java en JSOUP, HTMLUnit is een browser zonder GUI (Graphical User Interface) voor java programmas. In deze literatuurstudie wordt ook gebruik gemaakt van een webcrawler, dit is een bot die normaal aan web-indexering doet voor bv. Google. De basis van de meeste webscrapers zijn dan ook

webcrawlers maar in plaats van data te indexering gaat een scraper HTML data omzetten naar bruikbare data formats zoals JSON.

A.3 Stand van zaken

Voor interactie van de gebruiker na te bootsen kan in dit onderzoek gebruik gemaakt worden van Selenium deze tool was origineel gecreëerd voor het testen en debuggen van webapplicaties maar wordt momenteel veel gebruikt in functie van webscraping. Een framework die gebruikt kan worden voor een chrome browser te bekomen zonder GUI (Graphical User Interface) is Headless Chrome, zonder de GUI is er heel wat minder overhead voor de computer om te laden wat het process heel wat sneller maakt. Ook kunnen deze twee frameworks samen gebruikt worden.

A.4 Methodologie

Voor dit onderzoek zal er eerste een webscraper aangemaakt worden in twee verschillende codeertalen nl. python en java. Deze scraper wordt dan getest op een aantal websites met een tool die user interaction nabootst zoals e.g Selenium. Vervolgens zal er voor dit onderzoek een headless browser gebruikt worden om dezelfde websites te testen en performantie en reliablity te vergelijken.

A.5 Verwachte resultaten

Het verwachte resultaat van dit onderzoek is dat een scraper die gebruik maakt van een tool die user interaction nabootst, meer compatibiliteit zal hebben doordat deze op verschillende browsers kan uitgevoerd worden. Maar dat deze veel minder performant is en eventueel minder nauwkeurig. De webscraper die gebruikt maakt van een headless browser gebaseerd op chromium wordt er meer performantie en nauwkeurigheid verwacht. De setup van een scraper die gebruikt maakt van een headless browser wordt er echter wel wat meer complexiteit van verwacht, ook zal deze minder compatibiliteit hebben doordat deze enkel met chrome kan werken.

A.6 Verwachte conclusies

Hoogstwaarschijnlijk zullen de conclusies uit dit onderzoek aangeven dat een tool die user interactie moet nabootsten om een website te laden negatief is voor de performantie van de scraper maar veel makkelijker is om up and running te krijgen dan een headless browser te gebruiken, ook wordt er verwacht dat de headless browser meer resources en fysieke schrijfruimte van een computer zal opvragen.

Bibliografie

Axa. (2020). *Inzichten over het coronavirus en impact op de markten*. https://www.axabank.be/nl/blog/inzichten-coronavirus

Blekemolen, J. (2020). *Coronavirus Beurscrash | Beleggers vluchten uit aandelen – Update 12 maart 2020*. https://www.lynx.nl/kennis/artikelen/aandelen-coronavirus-goud-obligaties/

Hayes, A. (2020). *Consumer Cyclicals*. https://www.investopedia.com/terms/c/consumer_cyclicals.asp

Jarrod. (2016). *Linux Web Server Performance Benchmark - 2016 Results*. https://www.rootusers.com/linux-web-server-performance-benchmark-2016-results

Johnston, M. (2020). *Top Communication Stocks for June 2020*. https://www.investopedia.com/top-communications-stocks-4583180

Kegel, D. (1999). *The C10K problem*. https://web.archive.org/web/19990508164301/http://www.kegel.com/c10k.html

Kopp, C. M. (2019). *Basic Materials Sector*. https://www.investopedia.com/terms/b/basic_materials.asp

Korobov, M. (2014). *Lua scripting support*. https://github.com/scrapinghub/splash/issues/117

Legaspi, X. (2016). *Scraping Dynamic Websites for Economical Data: A Framework Approach* (thesis). institution.

Miller, M. (2019). *Industrials Sector: Overview and Funds*. https://www.valuepenguin.com/sectors/industrials

Nelson, R. (2016). *Docker Swarm Load Balancing with NGINX and NGINX Plus*. https://www.nginx.com/blog/docker-swarm-load-balancing-nginx-plus/

online source, C. (2020). *Systemd/Timers*. https://wiki.archlinux.org/index.php/Systemd/Timers

Rouse, M. (2018). *MongoDB*. https://searchdatamanagement.techtarget.com/definition/MongoDB

Sagalovskiy, D. (2014). *Sandboxed python*. https://wiki.python.org/moin/SandboxedPython

Scrapinghub. (2019). *Splash - A javascript rendering service*. https://splash.readthedocs.io/en/stable/

Tiwari, G. (2019). *Selenium Grid Tutorial : Learn to Set It Up*. https://www.browserstack.com/guide/selenium-grid-tutorial

Unadkat, J. (2019a). *Getting Started with Selenium IDE*. browserstack.com/guide/what-is-selenium-ide

Unadkat, J. (2019b). *Getting Started with Selenium WebDriver for Automation Testing*. https://www.browserstack.com/guide/selenium-webdriver-tutorial