DataFoundations

Portfolio

Kobe Vervoort   
3ITSOF1

Inhoudsopgave

[Labo 1 – Data-verwerking en -visualisatie met Python 3](#_Toc182746828)

[Labo 2 – Power BI en data visualisatie 5](#_Toc182746829)

[Labo 3 – Data Analytics in Google Cloud 7](#_Toc182746830)

[Labo 4 – Data Management 10](#_Toc182746831)

[Bronnen 11](#_Toc182746832)

# **Labo 1 – Data-verwerking en -visualisatie met Python**

Deze week hebben we het in de les gehad over het gebruik van data in ons dagelijkse leven. We zagen de verschillen tussen Data Science & ML, Data Engineering en DevOps en hoe dit allemaal deel uitmaakt van MLOps. Ik leerde hier veel over bij en stond versteld van de hoeveelheid Deep Learning die eigenlijk aanwezig is in ons leven (of de hoeveelheid van de data die we uit dagelijkse dingen kunnen halen, die gebruikt kan worden voor ML en DL).

Verder zagen we enkele types van data: gestructureerd, semi-gestructureerd en ongestructureerd. Ook leerde ik dat er een massa aan tools is voor het visualiseren van data, waaronder de “pandas”-library in Python. Ook zagen we de Data Science Lifecycle en hoe deze in zijn werk gaat.

De Data Science Lifecycle is een handig model waardoor je aan de hand van zes stappen gemakkelijk conclusies kunt trekken uit een bepaalde dataset (en zijn visualisatie). Dit heeft mij erg geholpen bij het oplossen van labo 1 en vooral bij het beschrijven en het nemen van conclusies bij vragen drie en vier. Door middel van de Data Science Lifecycle is het eenvoudiger om aan Data Storytelling te doen. Dit heeft me bewuster gemaakt van het gebruik van data in de hedendaagse wereld. Data wordt namelijk overal gebruikt: als informatiebron, als besluitvorming, maar ook als AI.

Verder vond ik het leuk en interessant om in de labo’s (labo 1 en labo 1 extra) zelf aan de slag te kunnen gaan met enkele datasets door middel van Python. Ik heb dan ook veel geleerd uit de labo-oefeningen, namelijk hoe ik op een nuttige en overzichtelijke manier data kan visualiseren en hier conclusies uit kan trekken.

Dit vond ik in het begin echter eerder moeilijk, aangezien ik niet goed wist wat er juist van mij verwacht werd dat ik neerschreef. Gelukkig kon ik hier hulp en feedback voor vragen aan Dhr. Haddouchi. Hij legde de theorie, maar ook de opdrachten erg grondig en op een goed tempo uit, zodat we de lesinhoud goed konden begrijpen. Uiteindelijk slaagde ik er wel in om de labo’s te voltooien.

Zoals je hieronder kunt zien, heb ik een zo goed mogelijke conclusie proberen trekken uit de verkregen data-visualisatie van vragen drie en vier:

*Stap 3: Interpretatie*

*Zoals we kunnen zien op de boxplot en het histogram is de data van variabele 'Vic' niet normaal verdeeld. We zien wel degelijk uitschieters. We zien in de boxplot dat de mediaan eerder rond de 2500 ligt. De spreiding van de data is groot, dit zien we aan de lengte van de boxplot. De standaardafwijking is 1177.*

*Stap 4: Rapportage*

*Bij het vergelijken van de variabelen 'Vic' en 'Qld' zien we dat het gemiddelde, de mediaan en de modus van 'Qld' hoger liggen dan die van 'Vic'. De interkwartielafstand (IQR) van 'Qld' is kleiner (1137 tegenover 1984), wat wijst op een meer geconcentreerde verdeling van de data rond de mediaan. Het histogram van 'Qld' toont een scheve verdeling met de meeste waarden onder 1000, wat op links-scheefheid wijst. 'Vic' heeft een bredere spreiding, wat duidt op meer variatie binnen de dataset. Deze verschillen kunnen wijzen op regionale of demografische verschillen in de dataset.*

# **Labo 2 – Power BI en data visualisatie**

Deze week zijn we tijdens de lessen aan de slag gegaan met PowerBI. PowerBI is een tool van Microsoft om data te verwerken/transformeren en handig te visualiseren aan de hand van een aanpasbaar dashboard. Ook leerden we bij over het nut van data visualisatie en de mogelijke manieren waarop data gevisualiseerd kan worden, ieder met een andere focus, dus ieder geschikt voor een ander doel van het weergeven van deze data.

Tijdens de theorieles leerden we dat data visualisatie erg belangrijk is. Data visualisatie zorgt er namelijk voor dat we data op een duidelijke en betere manier kunnen bestuderen en begrijpen, door dit in een visuele context te plaatsen. Door bepaalde statistieken in een andere weergave te zien, kunnen we er ook andere dingen van afleiden. Dit geeft ons andere inzichten!

Echter blijft het wel belangrijk om de keuze van de manier waarop je de data gaat visualiseren steeds af te stemmen op (1) het publiek, de personen voor wie je deze visualisatie realiseert, (2) de boodschap, datgene dat je graag wil overbrengen, maar ook (3) de impact die je wil bereiken aan de hand van je visualisatie. Voor elke “case”/context is er een andere, beter passende manier van visualiseren. Aan de hand van elk verschillende weergave of ander diagram, zul je je boodschap dus anders overbrengen.

In de Business Intelligence is data visualisatie handig voor het analyseren van data en het bekomen van nuttige inzichten; op deze manier kan men die business ondersteunen. Daarom zijn er ook verschillende types van rapporten: statisch, interactief en dashboards. Ook deze rapporten zijn verschillende soorten van visualisaties die in verschillende contexten gebruikt worden en die in bepaalde gevallen nuttiger en duidelijker zullen zijn dan de andere types.

Verder maakten we kennis met enkele data visualisatie tools, zoals Microsoft Power BI (dit gebruikten we in de labo’s), Tableau en QlikView. Ook hebben we enkele mogelijke visualisaties en diagrammen overlopen, zoals een tabel/matrix, staafdiagram, lijn- en oppervlaktediagram, spreidingsdiagram (scatter plot), kaarten, taart- en donutdiagram, tree map, combo diagram …

Ook leerden we over de “Grammar of Graphics”. Dit wil zeggen dat we bij het maken van een visualisatie steeds rekening moeten houden met zes componenten! Deze componenten helpen bij het begrijpen en het verduidelijken van de data die we willen weergeven. Aan de hand van deze componenten kunnen we betere inzichten geven en kunnen we bepaalde gegevens benadrukken!

1. Aan de hand van welke **data** we willen tonen, zullen we een keuze maken voor het diagram/de visualisatie waarop deze data het beste tot uiting zal komen.
2. Met behulp van de getoonde, **esthetische** componenten (assen, relatieve posities, kleur, grootte …), kunnen we bepaalde inzichten bekomen.
3. Door middel van de manier van weergave van elk onderdeel van de data (**geometrische** vormen), kunnen we gemakkelijk bepaalde verschillen zien.
4. De gebruikte **facetten** helpen bij het opmerken van verschillen tussen relatieve diagrammen.
5. Door het verschil tussen de **statistische** gegevens die getoond worden, zullen we andere, duidelijkere manieren van visualisatie gebruiken.
6. Ook de **coördinaten** en assen hebben invloed op het gegeven inzicht van een bepaald(e) visualisatie/diagram.

Ten slotte kregen we nog enkele tips voor het maken van een data visualisatie:

* Houd het functioneel.
* Vermijd taartdiagrammen en 3D-grafieken.
* Probeer niet te veel data tegelijk weer te geven (data overload).
* Toon steeds de volledige as.
* …

Daarna gingen we zelf aan de slag met Power BI tijdens de labo’s. Tijdens labo 1 was het de bedoeling om wat wegwijs te geraken in deze nieuw applicatie. Ik volgde de gegeven tutorials van Microsoft zelf, maar raakte niet zo ver, aangezien er stappen ontbraken… Toch las ik alle tutorials door en begreep ik wel al een beetje van Power BI, omdat ik in een online virtuele omgeving dit zelf al kon verkennen.

Tijdens labo 2 was het de bedoeling om het stappenplan van de docent (Dhr. Haddouchi) te volgen om zo tot een mooi Netflix-dashboard te komen. Dit is me wel goed gelukt, aangezien het gegeven stappenplan wel klopte en zeer duidelijk uitgelegd was! Ik kwam tot een mooi eindresultaat.

In de afgelopen lessen heb ik dus veel bijgeleerd over het gebruik van Power BI en het visualiseren van data. Ik onthoud dat ik altijd rekening moet houden met wat voor data ik wil visualiseren en de manier waarop ik dit best doe. Het blijft natuurlijk belangrijk om dit op zo’n manier te doen, dat men er de juiste inzichten uit kan halen.

# **Labo 3 – Data Analytics in Google Cloud**

Tijdens de labo’s van afgelopen weken zijn we bezig geweest met het ontdekken van data analytics in Google Cloud. Google heeft namelijk ook een platform waarin men met data en analyse aan de slag kan gaan: Google Cloud.

In Google Cloud vinden we enkele modules, zoals BigQuery. Deze modules hebben telkens een eigen doel met een eigen functie. Zo is BigQuery bv. een enterprise data warehouse en dient het vooral voor het verwerken, transformeren en visualiseren van grote hoeveelheden data. Verder vind je ook een aantal interessante cursussen/oefeningen op Google Cloud.

Tijdens de les zijn we dan ook aan de slag gegaan met deze cursussen. We maakten de introductie- en de “Weather Data in BigQuery”-cursus. Deze waren niet zo moeilijk, aangezien de opgave goed uitgelegd werd. Het was interessant om hiermee aan de slag te gaan via Google Cloud. Hoewel ik eerder al wat over Google Cloud had gehoord, had ik er zelf nog niet mee gewerkt tot aan deze labo’s.

Ook tijdens de theorieles kregen we een duidelijke uitleg over data analytics. Zo leerden we bv. over de componenten van data analytics (hoe data verwerkt worden tot bruikbare gegevens en dan kan worden geanalyseerd).

* Zo gaat men de data van verschillende bronnen (logs, IoT, ...) en verschillende types (semi- of (on)gestructureerd) verzamelen en injecteren (**Ingest Collect**).
* Daarna wordt deze verwerkt met behulp van een goede data infrastructuur, die veilige schaalbare en duurzame opslag toelaat (zoals databases, een data warehouse of een data lake) (**Store**).
* Vervolgens moet de data worden verwerkt en getransformeerd, vooraleer nuttige operaties, zoals sorteren, aggregeren en joinen, uitgevoerd kunnen worden (**Process Analyze**).
* Tot slot gaan we de data consumeren. Er zijn twee manieren om dit te verkrijgen: met een query (snelle analyse) of via BI tools (visualisatie van rapporten en dashboards).

Natuurlijk zijn er ook enkele uitdagingen bij het analyseren van data, dit noemen we de vijf V’s.

* Allereerst is het belangrijk dat we bij het analyseren van data rekening houden met de hoeveelheid data die wordt verwerkt (**volume**). De totale grootte van de inkomende data mag niet te groot (te veel onbruikbare data), maar ook niet te klein (te weinig data) zijn.
* Ook is de snelheid waarmee de data binnenkomen (**velocity**) van belang, aangezien er steeds vaker naar een quasi real-time dataverwerking gestreefd wordt.
* Data kunnen echter ook uit verschillende bronnen komen. Het is belangrijk dat men niet alleen data komende uit één enkele bron gaat verwerken (**variety**), aangezien dit eventueel een vertekend beeld kan geven.
* De accuraatheid en betrouwbaarheid van de data (**veracity**) spelen ook een grote rol in het analyseren en verwerken van data; dit noemen we data integriteit.
* Ten slotte is het ook van belang dat we rekening houden met de mate waarin een data product erin slaagt om nuttige informatie te genereren (**value**).

Verder zijn er ook verschillen tussen een **data warehouse** en een **data lake**:

* Een data warehouse focust nl. op relationele data, terwijl een data lake ook (niet-)relationele en ongestructureerde data opslaat.
* Bij data warehousing wordt een schema vooraf opgesteld, terwijl in een data lake het schema pas wordt gedefinieerd bij de data-analyse.
* Een data warehouse is duurder en levert snelle toegang tot gestructureerde data voor BI en rapportage, maar een data lake biedt goedkopere opslag en is gericht op data scientists voor ML en voorspellende analyses.

Ook hadden we het tijdens de theorieles over de methoden voor data verwerking, namelijk batch en streaming.

* Bij **batchverwerking** wordt data in grote hoeveelheden verzameld en verwerkt wanneer aan bepaalde voorwaarden wordt voldaan, zoals op een specifiek tijdstip. De verwerking gebeurt automatisch en resulteert in data die naar een opslaglocatie worden gestuurd voor verdere analyse of rapportage.
* Bij **streaming** worden de data echter in toenemende mate in real-time verzameld en verwerkt, waardoor inzichten sneller beschikbaar komen. Bij streaming worden de data in continue stroom verwerkt zodra deze binnenkomt. Het systeem dat de data verzamelt, noemen we dan de producer en het systeem dat deze verwerking uitvoert noemen we de consumer.

Beide methoden hebben hun specifieke toepassingen en voordelen, afhankelijk van de snelheid en het volume waarmee de data binnenkomen.

We hebben ook de verschillende types databronnen en opslagmethoden besproken, wat essentieel is voor het begrijpen van data management en analytics. De databronnen kunnen worden ingedeeld in drie hoofdcategorieën:

* **Gestructureerde data**: Deze data zijn georganiseerd in tabellen en kolommen, vaak opgeslagen met een Database Management System (DBMS). Ze zijn geordend volgens een relationeel datamodel, waarbij relaties tussen data-elementen worden weergegeven. Voorbeelden zijn MySQL, PostgreSQL, MS SQL Server en Oracle. Een nadeel van gestructureerde data is de beperkte flexibiliteit.
* **Semi-gestructureerde data**: Deze data zijn opgeslagen in elementen met bijbehorende attributen, zoals key-value paren. Ze bieden meer flexibiliteit en schaalbaarheid dan gestructureerde data. Voorbeelden zijn CSV, JSON, XML en Amazon Neptune.
* **Ongestructureerde data**: Deze data zijn opgeslagen als bestanden zonder een vooraf gedefinieerde structuur, zoals e-mails, foto's, video's en clickstream-data. Deze vorm van data vereist speciale technieken voor analyse en verwerking.

Ook heeft de heer Haddouchi het even gehad over **flat files**. Dit zijn nl. bestanden die gestructureerde data bevatten, maar vaak problemen hebben zoals ontbrekende, dubbele of onduidelijke waarden.Daarnaast zagen we ook dat **relationele databases** de uitdagingen van flat files oplossen door data te groeperen in tabellen op basis van objecten. De kolommen vertegenwoordigen de eigenschappen, terwijl de rijen de elementen van de groep weergeven. De relaties worden dan aangegeven met (foreign) keys.

We hebben ook de twee algemene manieren besproken om data in relationele databases te organiseren:

* **Online Transaction Processing (OLTP)** is gericht op snelheid bij het inladen van data, zoals bij boekhoudsystemen en online verkooptransacties.
* **Online Analytical Processing (OLAP)** is echter gericht op het snel uitlezen en gebruiken van data, vaak voor rapportages en managementondersteuning. OLAP-databases worden vaak gevoed door OLTP-databases voor analytische doeleinden.

Tot slot leerden we ook nog over het verschil tussen **niet-relationele databases** en **key-value** **databases**. Niet-relationele databases bieden documentopslag voor semi-gestructureerde en ongestructureerde data in formaten zoals JSON en XML, terwijl **k**ey-value databases net ongestructureerde data opslaan in de vorm van key-value paren.

Door deze verschillende soorten databronnen en opslagmethoden te begrijpen, krijgen we een beter inzicht in hoe we data kunnen verzamelen, opslaan en analyseren voor effectieve besluitvorming en inzichten. Deze lesinhoud is erg handig bij het kiezen van een geschikte database voor een bepaald project als toekomstige software-ontwikkelaar.

# **Labo 4 – Data Management**

Tijdens de theorieles en de labo’s van afgelopen weken zijn we bezig geweest met data management. Data management gaat over het behandelen, filteren en transformeren van data, zodat de datakwaliteit en privacy geoptimaliseerd kunnen worden.

We doen aan data mangement om bruikbare data te bekomen, zodat we hier besluiten uit kunnen trekken (data = macht). Om data bruikbaar te maken, volgen we enkele stappen:

1. De benodigde data verzamelen (zoek)
2. Het doel bepalen en de inhoud en betekenis van deze data begrijpen (begrijp)
3. De kwaliteit en nauwkeurigheid van deze data verzekeren (vertrouw)
4. De data delen en effectief publiceren (werk samen)
5. Snel toegang verkrijgen tot data en deze volgens de wet gebruiken (toegang)

Dit kunnen we verkrijgen door het verwerken van data m.b.v. het Data Mangement Body of Knowledge (DMBoK) Wheel, dat bestaat uit verschillende focusgebieden voor datamanagement, waarvan data governance centraal staat. Belangrijke componenten van het DMBoK Wheel zijn:

* Data-architectuur
* Datamodellering en -ontwerp
* Dataopslag en -bewerking
* Data-integratie en interoperabiliteit
* Referentie- en masterdata
* Datakwaliteit

Ook het verwerken van metadata is van belang. Dit is de contextuele laag die het de gebruiker eenvoudiger maakt om de data beter te begrijpen. Deze kan bestaan uit:

* Bedrijfsmetadata
* Technische metadata
* Structurele metadata
* Operationele metadata
* Governance-metadata

Data governance gaat over het kader van processen, regels en verantwoordelijkheden. Deze zorgen ervoor dat data binnen een organisatie accuraat, veilig, toegankelijk en consistent wordt beheerd en gebruikt. Data governance heeft volgende voordelen:

* Er wordt een gemeenschappelijke taal gecreëerd
* Het brengt mensen samen om samen te werken
* Beveiligingsrisico’s worden verminderd
* Data wordt betekenisvol

Om data governance toe te passen moet er een goede datamanagement strategy toegepast worden. Het kan hier gaan om defensief (datakwaliteit, beveiliging, privacy en naleving van regelgeving) of offensief (monetisatie, kostenreductie en operationele efficiëntie) gebruik van data.

Ook Master Data Management (MDM) is erg belangrijk, aangezien deze kerngegevens door meerdere afdelingen binnen een organisatie gebruikt worden. Deze zijn essentieel voor bedrijfsvoering en moeten daarom consistent en nauwkeurig zijn (single source of truth).

Tijdens de labo’s gingen we aan de slag met Looker Studio voor datavisualsiatie en Python voor data lineage. We maakten het Google CloudSkills labo *Visualizing Billing Data with Looker Studio*. Aangezien in de vorige labo’s Looker Studio ook al even aangehaald werd, maar omdat we hier nog niet mee gewerkt hadden, vond ik het dus erg interessant om hiermee aan de slag te kunnen gaan. Ook het labo rond data lineage vond ik interessant, want hierdoor werd het nut en doel van data lineage echt duidelijk.

De transformaties die de data heeft ondergaan, worden weergegeven in de data lineage grafiek. Zoals we kunnen zien op de grafiek werden volgende transformaties uitgevoerd op de data:

1. **Verwijder duplicaten:** Deze transformatie verwijderde rijen die identieke waarden bevatten voor alle kolommen. Het doel was om de dataset schoon te maken en de overbodige records eruit te halen.
2. **Toevoegen van de “total\_sales” kolom:** Deze kolom werd toegevoegd door de prijs per eenheid (UnitPrice) te vermenigvuldigen met het aantal verkochte eenheden (Quantity). Dit helpt bij het berekenen van de totale verkoopwaarde voor elke transactie.
3. **Filteren op Quantity > 10:** Deze transformatie filterde alle records uit waarin de hoeveelheid (Quantity) kleiner was dan 10. Dit werd gedaan om te focussen op de grotere sales, die relevanter zijn voor sales analyses.

Deze transformaties hadden telkens een andere **impact op de dataset**. Door duplicaten te verwijderen werd de dataset schoner, wat de betrouwbaarheid verhoogde. De toevoeging van de “total\_sales” kolom maakte het mogelijk om de omzet per transactie te berekenen. Het filteren van de dataset op grotere hoeveelheden zorgt voor een beter overzicht van de belangrijkste sales.

**Data lineage is belangrijk**, aangezien het helpt bij het traceren van elke stap in de transformatie van gegevens, waardoor de transparantie, controleerbaarheid en kwaliteit van de data verzekerd kan worden. Dit is cruciaal voor het maken van betrouwbare analyses en handig bij het trekken van conclusies (uit de data) en het maken van besluiten.

# **Bronnen**

*Dashboard*. (2024, oktober 27). Retrieved from Google Cloud Skills Boost: https://www.cloudskillsboost.google/

*Google Cloud Platform lets you build, deploy, and scale applications, websites, and services on the same infrastructure as Google.* (2024, oktober 27). Retrieved from Google Cloud Platform: https://console.cloud.google.com/

s.n. (2024, oktober 10). *Oefening: gegevens laden in Power BI Desktop*. Retrieved from Microsoft Learn: https://learn.microsoft.com/nl-nl/training/modules/clean-data-power-bi/8-lab