Onderzoek naar Principal component analysis (PCA)

In de paper over de dataset van ons project [1] wordt gewerkt met een PCA om anomalieën te herkennen in de dataset. Om beter te begrijpen waarom er voor een PCA is gekozen wordt onderzoek gedaan naar twee vragen:

- Hoe werkt een PCA;

- Hoe kan PCA toegepast worden in ons project.

Hoe werkt een PCA

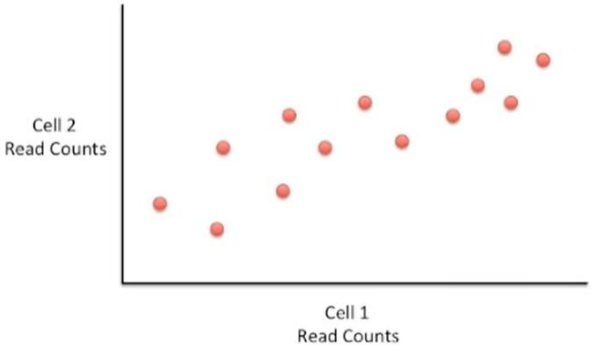
"Principal component analysis (PCA) is a technique for reducing the dimensionality of large datasets, increasing interpretability but at the same time minimizing information loss." [2] In het kort wordt hier beschreven dat er bij PCA, principal components worden berekend om de data op een andere manier te kunnen beschrijven. PCA is een vorm van dimensionality reduction (een transformatie van data waarbij het aantal dimensies wordt gereduceerd). Dit betekent dat de uitkomst van een PCA niet de data volledig beschrijft maar de belangrijkste onderdelen samenvat in een nieuw perspectief.

Om de werking van een PCA te beschrijven wordt gebruik gemaakt van beeldmateriaal van de Universiteit van North Carolina. [3] In het voorbeeld van de video wordt gewerkt met twee cellen die van elkaar verschillen in de genen. De dataset ziet eruit als weergegeven in Tabel 1.

Tabel 1: Dataset voorbeeld uit de video van StatQuest

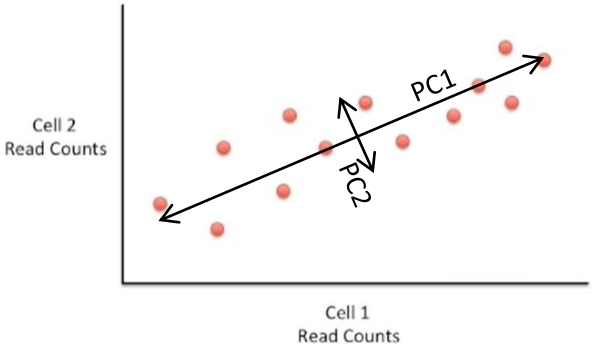
Gene	Cell1 reads	Cell2 reads
a	10	8
b	0	2
c	14	10
d	33	45
e	50	42
f	80	72
g	95	90
h	44	50
i	60	50
... (etc)	... (etc)	... (etc)

Alle genen kunnen in een scatterplot worden uitgezet tegen de verschillende cellen. Omdat er in dit voorbeeld twee cellen zijn, kan dit plot in twee dimensies getekend worden. De uitwerking hiervan is te zien in figuur 1. Dezelfde berekening kan worden uitgevoerd in meerdere cellen, in dit geval zal elke cel voor een nieuwe dimensie zorgen.



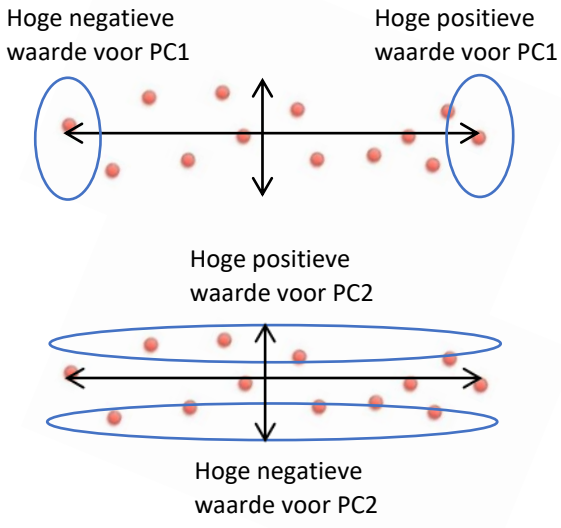
Figuur 1: Genen uitgezet tegen cel1 en cel2

In het scatterplot kan nu een lijn getrokken worden om de grootste afwijking in de data weer te geven. Deze lijn wordt Principal component één (PC1) genoemd. Dit wordt ook gedaan voor de één na grootste afwijking, deze wordt PC2 genoemd.



Figuur 2: Principal components

Voor de uitlegbaarheid van het model wordt de pijl nu getekend, maar tijdens een PCA wordt deze natuurlijk berekent. Dit wordt gedaan door aan elke gen een nummer te geven dat aangeeft hoeveel invloed deze heeft op de richting van PC1 en PC2. De genen die het verst afwijken krijgen het hoogste getal. De genen rechts en boven krijgen een positief nummer en de genen links en onder een negatief nummer. Dit is weergegeven in figuur 3 en in tabel 2 staan een aantal van de waarderingen.



Figuur 3: Waardering van genen per PC in een grafiek

PC1

Gene	Influence on PC1	In numbers
a	high	10
b	low	0.5
c	low	0.2
d	low	-0.2
e	high	13
f	high	-14
...	...	

PC2

Gene	Influence on PC2	In numbers
a	medium	3
b	high	10
c	high	8
d	high	-12
e	low	0.2
f	low	-0.1
...	...	

Tabel 2: Waardering van de genen per PC in een tabel

Als laatste stap moet voor de cellen de afhankelijkheid voor PC1 en PC2 berekend worden. Dit wordt gedaan door de getallen uit tabel 1 en de getallen uit tabel 2 met elkaar te vermenigvuldigen en bij elkaar op te tellen. Hieronder staat de berekening voor PC1 voor cel 1.

Gene	Cell1 reads
a	10
b	0
c	14
d	33
e	50
f	80
g	95
h	44
i	60
... (etc)	... (etc)

PC1

Gene	Influence on PC1	In numbers
a	high	10
b	low	0.5
c	low	0.2
d	low	-0.2
e	high	13
f	high	-14
...	...	

$$PC1_{cel1} = \sum(waarde\ Cel1\ voor\ gen\ x * waarde\ PC1\ voor\ gen\ x)$$

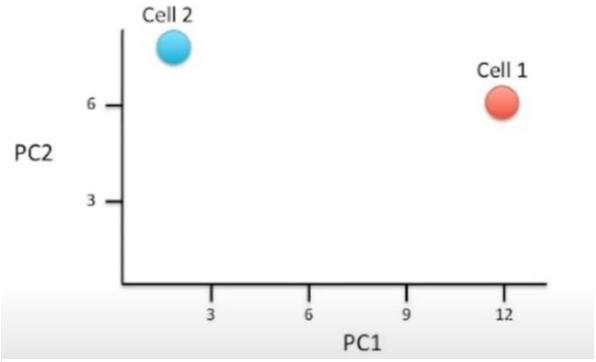
$$PC1_{cel1} = (10 * 10) + (0 * 0{,}5) + (14 * 0{,}2) + etc.$$

$$PC1_{cel1} = 12$$

Door deze berekening ook uit te voeren voor PC2 en voor de andere cel, worden de waarde uit tabel 3 verkregen. Met deze waarden kan vervolgens figuur 4 gemaakt worden.

Tabel 3: PC waarden per cel

	PC1	PC2
Cel 1	12	6
Cel 2	2	8

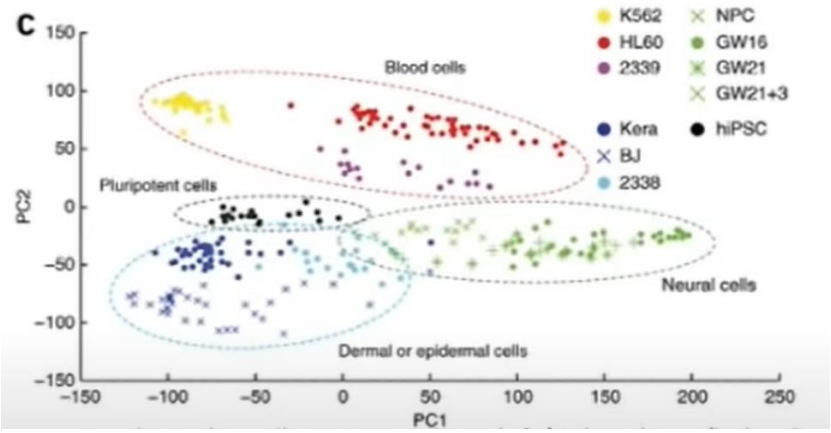


Figuur 4: Cellen tegen PC waarden

Deze grafiek en de waarden die erbij horen zijn het eindproduct van de PCA. De hoeveelheid data per cel is gereduceerd van 13 naar 2 waarden. Hiernaast kan, als dezelfde analyse wordt uitgevoerd met meerdere cellen, nu een clustering model worden toegepast om groepen van cellen te onderscheiden. Naast het zien van een verschil tussen deze cellen, kan ook worden teruggerekend welke genen er het meeste voor hebben gezorgd dat deze cellen van elkaar verschillen.

Belangrijk om te weten bij het bekijken van de grafiek is dat een verschil over PC1 meer waarde heeft dan een verschil over PC2. Dit omdat we in een van de eerste stappen, de grootste afwijking in de dataset hebben toegekend aan PC1 en de één na grootste afwijking hebben toegekend aan PC2.

PCA kan ook worden uitgevoerd met grotere datasets. In dit geval zou er ook een PC3, PC4, etc. kunnen worden berekend. In figuur 5 is de uitkomst van PCA te zien met een groter aantal cellen. Elke stip in deze grafiek is een cel en kan, zoals gedaan in het figuur, geclassificeerd worden met een simpel algoritme.



Figuur 5: Voorbeeld uitkomst PCA met grotere dataset.

Hoe kan PCA toegepast worden in ons project?

Voor dat besproken wordt hoe PCA toegepast kan worden in ons project is het eerst belangrijk waarom PCA een goede oplossing zou kunnen zijn voor het probleem. Om deze vraag te beantwoorden zijn de paper van de dataset [1] en een paper die verschillende modellen voor anomalie detectie vergelijken [4] gebruikt. Uit de eerste paper blijkt dat PCA een goede oplossing is voor anomalie detectie en uit de tweede paper blijkt dat KNN een goed model is. Dit geeft genoeg reden om met deze modellen verder te gaan tot een oplossing voor dit probleem.

In het project hebben we te maken met een dataset van 2239 kolommen en 100 batches van ieder 1100 timestamps. De opzet van deze dataset is in tabel 4.

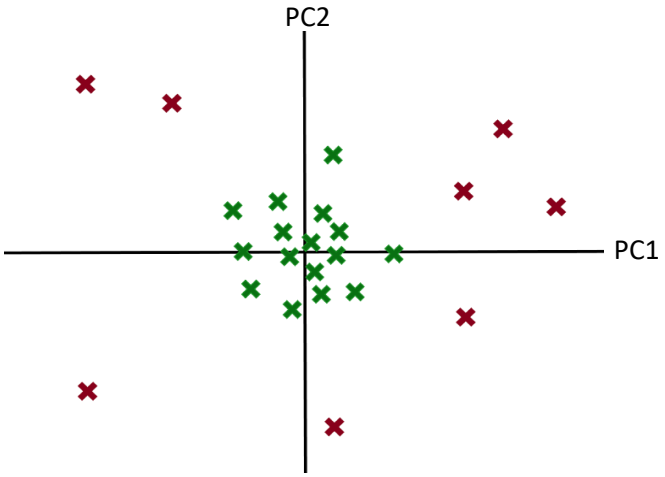
Tabel 4: Opzet van dataset bij het project

		Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
Batch 1	Timestamp 1	----------	----------	----------	----------	----------
	…	----------	----------	----------	----------	----------
	Timestamp n	----------	----------	----------	----------	----------
…	Timestamp 1	----------	----------	----------	----------	----------
	…	----------	----------	----------	----------	----------
	Timestamp n	----------	----------	----------	----------	----------
Batch n	Timestamp 1	----------	----------	----------	----------	----------
	…	----------	----------	----------	----------	----------
	Timestamp n	----------	----------	----------	----------	----------

Omdat de data van het project aanzienlijk complexer is dan die van het voorbeeld, is ervoor gekozen om te beginnen met het benoemen van een gewenst resultaat en vandaar uit terug te redeneren naar de input data. Het doel van de PCA is het onderscheiden van batches die fout gaan van de batches die goed gaan. Hiernaast zou het mooi zijn als bij de foutieve batches kan worden bepaald welke feature hier de oorzaak van was.

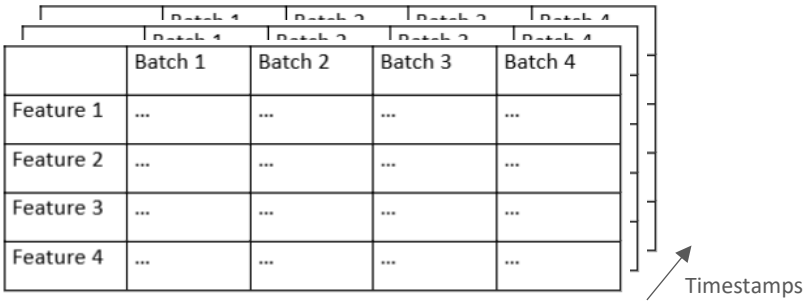
Uitgangspunt

Om de foutieve batches van de goede batches te kunnen onderscheiden zou figuur 6 het ideale uitgangspunt zijn. Door het toepassen van een cluster algoritme kunnen hiermee de goede batches geïsoleerd worden maar kan er ook onderscheid gemaakt worden tussen verschillende foute batches.



Figuur 6: Gewenst eindresultaat PCA

Om deze grafiek te kunnen krijgen uit een PCA is een tabel nodig van batches tegen de features. Deze tabel ziet eruit zoals in figuur 7. Omdat er wel rekening moet worden gehouden met de tijdserie is dit dus een driedimensionale array.



Figuur 7: Input dataset voor gewenste uitput PCA

Als input van een PCA kan geen driedimensionale array gebruikt worden. Om deze derde dimensie weg te werken wordt een gemiddelde genomen van de timestamps. Dit kan op twee manieren gedaan worden:

1. Door een batch in kleinere stukken te hakken. Op deze manier krijg je een gemiddelde van een gedeelte van de timestamps. Hierdoor wordt de focus gelegd op enkele timestamps en springen afwijkingen er het meeste uit;
2. Door een gemiddelde te nemen van alle timestamps tot op een bepaald moment. Hierdoor wordt de hele tijdseries meegenomen maar kunnen kleine afwijkingen wegvallen.

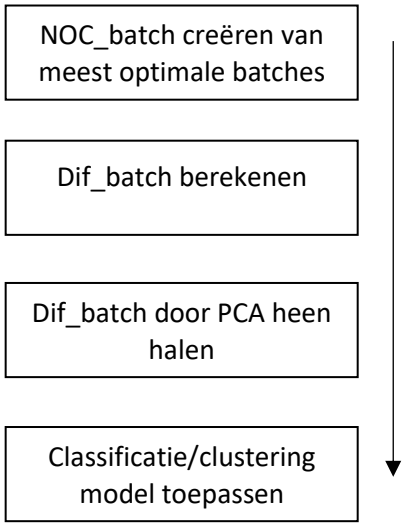
NOC-batch

Om regelmaat in de uitkomsten van de PCA te krijgen kan worden gewerkt met een normal operating conditions (NOC) batch. Deze NOC-batch kan als referentie gebruikt worden van alle nieuwe batches waar een model een uitspraak over moet gaan doen. Hierdoor zullen net als in figuur 6 de goede batches rond het nulpunt liggen omdat zij weinig afwijken van de NOC-batch en zullen de batches de fout gaan verder van het nulpunt afliggen. Als input van de PCA wordt dus het verschil gepakt tussen de NOC-batch en de nieuwe batch. Deze batch noemen we ‘dif_batch’.

$$dif\_batch = \text{new\_batch} - \text{NOC\_batch}$$

De NOC-batch kan gecreëerd worden door het samenvoegen van verschillende batches waarvan het proces zo ideaal mogelijk is verlopen. In de paper hebben ze hier 17 batches voor gebruikt. Helaas staat er nergens beschreven welke 17 dit zijn. Hierom zal zelf onderzocht moeten worden welke batches het beste gebruikt kunnen worden voor het maken van een NOC.

In figuur 8 staat een overzicht van de stappen die uitgevoerd dienen te worden, om PCA op een nuttige manier toe te kunnen passen in het project.



Figuur 8: Stappenplan voor gebruik van PCA in het project

Bibliografie

[1] University College London, „Modern day monitoring and control challenges outlined on an industrial-scale benchmark fermentation process," Elsevier, London, 2019.

[2] University of Exeter, „Principal component analysis: a review and recent developments," 13 4 2016. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202. [Geopend 10 9 2022].

[3] University of North Carolina, „Youtube," 2015. [Online]. Available: https://www.youtube.com/watch?v=_UVHneBUBW0&t=545s. [Geopend 18 9 2022].

[4] A. Kharitonov, „Comparative analysis of machine learning models for anomaly detection in manufacturing," Elsevier, Magdeburg, 2022.