

**Justification Document**

Jinne Haan en Daan Derks



28 mei 2025

fontys university of applied sciences

Inhoud

[Keuze van de dataSets 2](#_Toc199241376)

[Gemaakte keuzes 2](#_Toc199241377)

[C en Gamma resultaten 3](#_Toc199241378)

[Dataset1: 3](#_Toc199241379)

[Dataset2: 3](#_Toc199241380)

[AI gebruik 3](#_Toc199241381)

[Wie heeft wat gedaan? 4](#_Toc199241382)

# Keuze van de dataSets

# Gemaakte keuzes

Aangezien de opdracht eigenlijk 2 kleine programeer opdracht waren, hebben we de opdracht in 2 classes gemaakt. Op deze manier kunnen we 1 class runnen om dingen makkelijk te testen en wordt niet steeds alle code aangeroepen.

Ook hebben we ervoor gekozen om in een python bestand te werken en niet in een notebook bestand. De reden hiervoor is omdat wij het persoonlijk fijner vinden werken, daarnaast zie je in de praktijk bij bedrijven dat python bestanden meer voorkomen dan notebook bestanden. Om ons een zo goed mogelijke voorbereiding te geven op het bedrijfsleven maken wij de python file al ons eigen.

We hebben gekozen om 3 verschillende test-scores te berekenen:

* Accuracy\_score: gekozen omdat deze veel voorkomend is en vaak ook goed blijkt
* Balanced\_accuracy\_score: werkt bijna hetzelfde als accuracy\_score, maar houd meer rekening als klassen meer ongelijk zijn
* Macro F1\_score: geeft een gelijke score aan alle klassen.

# C en Gamma resultaten

## Dataset1:

C = 0.01 -> Accuracy=1.000 -> b\_accuracy=1.000 -> f1=1.000

C = 0.1 -> Accuracy=1.000 -> b\_accuracy=1.000 -> f1=1.000

C = 1 -> Accuracy=1.000 -> b\_accuracy=1.000 -> f1=1.000

C = 10 -> Accuracy=1.000 -> b\_accuracy=1.000 -> f1=1.000

C = 100 -> Accuracy=1.000 -> b\_accuracy=1.000 -> f1=1.000

C = 1000 -> Accuracy=1.000 -> b\_accuracy=1.000 -> f1=1.000

C = 1000000 -> Accuracy=1.000 -> b\_accuracy=1.000 -> f1=1.000

Zoals je kunt zien is alles een score van 1. Dit is super goed. Dit betekend eigenlijk gewoon dat de data perfect lineair te scheiden is en daardoor heeft de C waarde weinig tot geen invloed.

## Dataset2:

C=0.1, gamma=0.001 => accuracy=0.500 => b\_accuracy=0.500 => f1=0.333

C=0.1, gamma=0.01 => accuracy=0.538 => b\_accuracy=0.538 => f1=0.414

C=0.1, gamma=0.1 => accuracy=0.962 => b\_accuracy=0.962 => f1=0.961

C=1, gamma=0.001 => accuracy=0.615 => b\_accuracy=0.615 => f1=0.549

C=1, gamma=0.01 => accuracy=0.962 => b\_accuracy=0.962 => f1=0.961

C=1, gamma=0.1 => accuracy=0.962 => b\_accuracy=0.962 => f1=0.961

C=10, gamma=0.001 => accuracy=0.962 => b\_accuracy=0.962 => f1=0.961

C=10, gamma=0.01 => accuracy=0.962 => b\_accuracy=0.962 => f1=0.961

C=10, gamma=0.1 => accuracy=1.000 => b\_accuracy=1.000 => f1=1.000

Zoals je kunt zien wordt de data alleen perfect gesplitst als de C op 10 staat en de gamma op 0,1. Uiteraard moet er wel rekening gehouden worden dat je niet gaat overfitten.

# AI gebruik

We hebben voor verschillende onderdelen AI gebruikt.

Datasets: we hebben ai gebruikt om handige website of python packages te vinden met een grote hoeveelheid datasets die we zouden kunnen gebruiken. Het kiezen van de uiteindelijke dataset hebben we zelf gedaan door op de website rond te kijken. Bij deze datasets staat op de website ook al een hele hoop informatie over de data.

Syntax: Het programmeren doen we zelf, we bedenken wat we willen doen en hoe de flow ongeveer zal gaan. Wanneer we bezig zijn met programmeren en de syntax lukt niet helemaal, dan vragen we ai hoe dat moet en om voorbeelden. Een voorbeeld: we hebben gevraagd om voor te doen hoe je vanuit de fetch\_ucirepo datasets kunt inladen in python en transformeren in een Pandas dataframe. We wisten dat dit kon, maar niet precies hoe, want het is de eerste keer dat we met die package werken.

Test-scores: nadat we hebben gekeken naar welke test-score we gaan gebruiken hebben we nog even aan ai nagevraagd of dat die het eens is met de keuze of, of dat hij toch een andere test-score zou aanraden. Hieruit bleek dat we goed bezig waren.

Errors: soms krijgen we errors en is het moeilijk te begrijpen waar het mis gaat. Dan vragen we ai om inzicht te bieden in wat er mis gaan en een mogelijk oplossing suggestie te krijgen.

# Wie heeft wat gedaan?

We hebben samen aan alle code gewerkt. Dit hebben we gedaan door naast elkaar te zitten en te programmeren, meestal hebben we dan allebei kleine taken die de hele tijd samen vallen. We hebben naast elkaar samengewerkt via de ‘live share’ extensie van visual studio code gemaakt door Microsoft. Op deze manier kunnen we een beetje als een online word document tegelijk in een document typen.

Alle keuzes die gemaakt zijn tijdens deze opdracht zijn dan ook samen genomen en overlegd.

Het justification document is gemaakt door:

Keuze van de datasets: Daan

Gemaakte keuzes: beide

C en Gamma resultaten: Jinne

AI gebruik: Jinne

Wie heeft wat gedaan?: beide

Sources:

OpenAI. (2023). *ChatGPT* (Mar 14 version) [Large language model]. <https://chat.openai.com/chat>

Huilgol, P. (2024, January 15). Accuracy vs. F1-Score - Analytics Vidhya - Medium. *Medium*. <https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2>

*Course modules: FEB25: - Data Science 8*. (n.d.). <https://canvas.fontys.nl/courses/26895/modules>