Projectsamenvatting Brandon Hillert

**Probleembeschrijving**

*“De klant zit op een terras en vraagt aan de ober een biertje. Na het drinken van het biertje weet hij niet meer wat die precies gedronken heeft. Hij weet alleen dat het biertje een bepaalde percentage alcohol heeft en wat voor soort biertje het was ( BIJV een IPA) Hij gaat naar huis en vult de waardes die hij weet in het algoritme. Het algoritme zoekt uit welk biertje het meest lijkt op de eigenschappen die hij heeft ingevuld, en hieruit blijkt dat het een oud vertrouwde groene soldaat was ( Heineken).”*

Mijn vader kwam met het idee om bier te classiferen op basis van bepaalde eigenschappen. Hierbij zei hij dat hij graag een systeem wilt waar hij 2 waardes invult, en het systeem het biertje aanwijst die het meest lijkt op de waardes die beschreven zijn, een soort recommendation engine. Ik ben gaan kijken naar verschillende soorten classificatie algoritmes en ben uiteindelijk op het KNN algoritme gekomen.

**Beschrijving van de eisen:**

Bij het maken van een algoritme komen natuurlijk ook bepaalde eisen bij kijken. Ik heb mijn vader gevraagd wat voor eisen hij graag terug wilt zien. Hij vertelde mij dat het systeem gebruiksvriendelijk, makkelijk en overzichtelijk moet zijn. Verdere eisen waren dat het systeem zich beperkt moet houden aan de bekende soorten bier, en niet een zomaar een willekeurige biersoort uit India bijvoorbeeld. Ook moet het systeem snel zijn, niet dat het langer dan een minuut duurt voordat er een uitkomst is.

**Het gekozen algoritme met een referentie naar de bron**

Ik heb uiteindelijk gekozen voor het KNN algoritme. Het KNN algoritme is een classificatie algoritme, die kijkt naar waardes die het meest op lijken op de eigen waardes. Na veel scrollen en het onderzoeken van dit algoritme, leek het KNN algoritme perfect voor mijn probleem. Het is een algoritme dat ik makkelijk vind om te begrijpen. Voor het programmeren was het beeld van hoe de code structuur eruit moest komen zien al redelijk duidelijk.

Sutton, C. (2012). Nearest-neighbor methods. Wiley Interdisciplinary Reviews: Computational Statistics, 4(3), 307–309. <https://doi.org/10.1002/wics.1195>

**Onderbouwing van de gemaakte keuzes.**

Tijdens het programmeren van het KNN algoritme heb ik keuzes moeten maken. Zo begon het allereerst met de dataset. In het CSV bestand staan 9 verschillende waardes per biertype. Voor mijn gevoel waren de meeste waardes overbodig. Daarom heb ik ervoor gekozen om van de gehele database maar 3 cruciale waardes mee te nemen naar mijn algoritme: 1. Het type bier 2. Het ABV ( alcoholpercentage ) 3. De IBU ( schaal 0 / 100 hoe bitter een biertje is). Zo wordt het programmameren van het algoritme een stuk overzichtelijker, duidelijker en efficiënter, iets wat mijn opdrachtgever ook wilt.

Programma:

Zodra het programma gerund wordt heeft de gebruiker 2 opties. 1 optie om zelf waardes in te voeren en 2 het testen van het algoritme.

Zodra de gebruiker kiest om waardes in te voeren wordt er gevraagd om 2 waardes. 1. Het alcohol percentage en 2. Een waarde tussen 1 en 100 hoe bitter het biertje is. Na het invullen gaat het algoritme kijken in de database welke waardes het meest lijken. Dit doet die door de afstand van input tot punten te berekenen. Hier komen allerlei waardes uit, die gesorteerd op kortste afstanden van input tot punten. Met punten bedoelen we hier biersoorten. Zodra deze afstanden bepaald zijn, gaan we kijken naar welke range we willen kijken. Ook deze waarde, de K-value, is cruciaal voor de uitkomst van de voorspelling.

Het testen van het algoritme werkt als volgt. Het algoritme krijgt een hele lijst mee waar de ABV, IBU en biertypes in staan. De biertypes worden voor het algoritme nog verborgen. Het algoritme gaat dus kijken naar de waardes die meegegeven worden, en vergelijkt deze met de waardes in de database. Op basis van de uitkomsten die het algoritme maakt ( dus de voorspelling ) controleert de functie of de voorspelling gelijkt staan aan het onderwater meegegeven type. Klopt dit wel, dan heeft het algoritme het goed geraden, zo niet, dan is er ruimte voor verbetering in het algoritme. Afhankelijk van de K value komt er een bepaald percentage uit. In het programma ziet men dan ook dat zodra de K-value veranderd, het slagingspercentage ook veranderd.