DATAKWALITEIT: DEEL 2

INLEIDING

In dit vierde deel van het project willen we het ETL-proces opnieuw uitbreiden door verdere controles op datakwaliteit in te voeren. In deel 3 hebben we dit al gedaan door controles op consistentie in te voeren. In het huidige deel zullen we ons richten op het vinden van dubbele data.

KWALITEITSPROBLEMEN

Het probleem dat we in dit deel van het project aanpakken, is dat van dubbele data. Meer bepaald bevat de brondata een aantal personen die meer dan één keer voorkomen. We willen in de eerste plaats een model ontwikkelen om deze personen te ontdekken.

Om dit te doen gebruiken we grotendeels de strategie van Fellegi en Sunter die in de les werd uitgelegd. Dat betekent dat we eerst een score vector berekenen voor elk koppel van tuples. Om een dergelijke score vector te berekenen, kan je twee tuples vergelijken op basis van de attributen. Aangezien de gegevens van personen grotendeels tekstuele data zijn, gebruik je hiervoor best de extensie "fuzzystrmatch". Hier vind je een implementatie van zowel de Levenshtein afstand als van de Soundex index. Je kan ook even kijken naar de voorbeeld code die tijdens de les werd getoond. Deze code is beschikbaar op Ufora onder "Inhoud".

Het voornaamste verschil met het Fellegi Sunter model, is dat we hier een veel eenvoudiger model hanteren om een score vector te mapping op een finale beslissing. Dit doen we voornamelijk omdat we niet beschikken over trainingsdata. We zullen de score vector daarom verwerken door middel van de Ordered Weighted Average (OWA) operator.

ORDERED WEIGHTED AVERAGE (OWA)

Een OWA-operator is een aggregatie functie die een vector $\mathbf{v}$ van scores tussen 0 en 1 afbeeldt op een geaggregeerde score, eveneens een getal tussen 0 en 1. De procedure is hierbij dat de elementen van $\mathbf{v}$ eerst worden *gesorteerd* van *groot naar klein*. De gesorteerde vector noemen we $\mathbf{v}^*$. Vervolgens beschouwt men een gewichtsvector $\mathbf{w}$ die:

1. eenzelfde dimensie heeft als $\mathbf{v}$,
2. elementen heeft in het eenheidsinterval $[0,1]$
3. en waarvoor de som van alle elementen gelijk is aan 1.

De OWA-operator beeldt dan de vector $\mathbf{v}$ af op het getal

$$\mathrm{OWA}(\mathbf{v}) = \sum_{i=1}^{|\mathbf{v}|} w_i \cdot v_i^*$$

Eenvoudig gesteld berekent de OWA-operator een *gewogen* gemiddelde van de *gesorteerde* vector $\mathbf{v}^*$.

Aangezien je reeds weet hoe je een score vector moet berekenen, blijft nog de vraag hoe je de gewichten berekent. Hiervoor kan je het principe van een kwantor gebruiken. Dit is een niet-dalende functie $Q: [0, n] \rightarrow [0,1]$ die getallen tussen 0 en n (dit is het aantal dimensies van je score vector) afbeeldt op een getal tussen 0 en 1. Bij conventie kiezen we steeds $Q(0) = 0$. Een kwantor drukt een bepaalde *hoeveelheid* uit en $Q(i)$ drukt uit in welke mate het getal i compatibel is met die hoeveelheid.

In dit project zullen we werken met de volgende eenvoudige kwantor:

$$Q(x) = \left(\frac{x}{n}\right)^n$$

Je kan deze functie gebruiken om de gewichten te berekenen als volgt:

$$w_i = Q(i) - Q(i-1)$$

waarbij i dus een getal is tussen 1 en n. Als we gewichten op deze manier kiezen en we hebben een score vector $\mathbf{v}$ die resulteert uit de vergelijking van twee tuples, dan drukt de OWA-operator de mate dat twee tuples dubbels zijn van elkaar uit als de mate dat Q attributen overeenkomen, waarbij Q een hoeveelheid is die wordt uitgedrukt door de kwantor. Je zou kunnen zeggen dat je op deze manier twee tuples als dubbel beschouwt als “de meeste” attributen overeenkomen.

TECHNISCHE IMPLEMENTATIE

De controle op dubbele data moet opnieuw worden geïmplementeerd als onderdeel van het ETL-proces. Je kan dit in een aparte stap implementeren. Het beste is om een aparte functie te maken die twee tuples neemt, een score vector berekent en vervolgens een OWA-score berekent voor deze vector. Opgepast: de scores in de score vector moeten getallen zijn tussen 0 en 1. Eens je dit hebt gedaan, is het kwestie van een drempelwaarde te beschouwen. Ligt dit OWA-score boven de drempelwaarde, dan zijn de tuples dubbel, anders niet.

Beperk je tot het detecteren van dubbele data voor bezoekers. Kies zelf een goede drempelwaarde voor de score die je voor twee tuples uitkomt. Als je de gevonden tuples zou moeten samenvoegen, wat zou een goede strategie zijn om dit te doen? Kan je altijd een beslissing nemen? Indien niet, wat doe je dan met dubbele data? Hoe uiteindelijke samenvoegen en/of verwerken van dubbele data hoef je in deze stap nog niet te doen, dat komt in het laatste deel.

VERSLAG

In het uiteindelijke verslag van je project rapporteer je voor dit deel hoe je te werk bent gegaan en wat je bevindingen zijn. Hoe heb je attributen vergeleken met elkaar? Hoe heb je de scores getransformeerd naar het eenheidsinterval? Welke drempelwaarde heb je gekozen om de output van de OWA-operator te mappen op een beslissing? Geeft dit goede resultaten? Denk je dat je veel dubbele data mist? Zitten er veel valse positieven in je resultaat? Bespreek ook de voor- en nadelen van te werken met de OWA-operator.