Workshop AI in smart industry - SMS dataset

Onno Huijgen

28 maart 2025

Introductie

We gaan nu aan de slag met wat complexere data. We hebben data verzameld van 1000 SMS-berichten. Een gedeelte van die berichten is ongewenste spam, dus het is aan ons om een spamfilter te ontwikkelen.

1 Data inladen

1.1 Open een nieuw canvas in de Orange tool.

1.2 Dit keer is de dataset niet al aanwezig in de tool, en moeten we hem zelf inladen. Maak een module **file** aan op je canvas en laad de SMS-data met labels in. Zorg dat de feature SPAM als *categorical target* variabele staat gemarkeerd.

1.3 Laad de data in een **Data Table** en kijk hoe de data eruitziet.

1.4 We moeten de data eerst in een formaat zetten waar de Orange tool verder mee kan werken. Dat doen we door een corpus te maken. Voeg een **Corpus** module toe en neem als input de data uit het bestand.

1.5 Je kunt een overzicht maken van de woorden die vaak in de teksten voorkomen door een wordcloud te maken. Doe dit met de **wordcloud** module

2 Preprocessing

Vaak is het niet een goed idee om de data direct zoals hij is in een machine learning model te gooien. Het is voor een model belangrijk dat de data in het juiste formaat staat, dat er geen missende waardes zijn, en dat er de juiste features zijn waar het model ook van kan leren. In dit geval is er geen missende data, maar zijn er ook geen duidelijke (numerieke) features waar een model iets mee kan. Daarnaast is de tekst in de SMSjes nog vrij grof. Identieke woorden kunnne er bijvoorbeeld nog met of zonder hoofdletter instaan, en voor we beginnen met trainen is het handig om dit uniform te maken. Ook staan er nog allerlei leestekens in de tekst. Om dat op te lossen, gaan we de data eerst preprocessen.

2.1 Voeg de module **Preprocess text** toe en verbind je corpus.

2.2 In de preprocessor kun je verschillende preprocessing stappen tegelijk uit laten voeren. Begin ermee te zorgen dat de preprocessor alle woorden in lowercase zet.

2.3 Zorg dat je preprocessor alle smsjes opbreekt in losse woorden. Dit heet tokenization.

2.4 Filter weinig zeggende tussenwoorden uit de dataset. Denk aan woorden als 'de', 'een' of 'en'. Deze worden ook wel *Stopwords* genoemd.

2.5 Maak een wordcloud van de tekst na het preprocessen. Ben je tevreden met de woorden die je ziet? Als je nog stopwords toe wilt voegen, kan dat handmatig door een bestandje te maken (stopwords.txt) en in te laden in de preprocessor. Filter stopwords uit de tekst tot je tevreden bent met de wordcloud.

2.6 We gaan de data nu omzetten naar een numerieke vorm. Daarvoor gaan we bij elk bericht tellen hoevaak woorden voorkomen. Deze vorm van data heet **Bag of Words**. Maak van de data een **Bag of Words** door de gelijknamige module toe te voegen.

2.7 Voor we naar het modelleren overgaan is het handig om nog een transform **Select Columns** toe te voegen. Daarin kun je aangeven welk van de features de target is en wat je allemaal als input gebruikt. Zoals je ziet is er een nieuwe feature aangemaakt voor elk woord dat voorkomt in de dataset. De waarde van die feature is dan hoe vaak het woord voorkomt in de tekst.

Modellen trainen

Nu we de data opgeschoond hebben en in een nuttige vorm gegoten hebben, kunnen we modellen trainen.

2.1 Train drie verschillende machine learning modellen. Begin met een **Tree**, een **Neural Network** en **Logistic Regression**.

2.2 Het is mogelijk de parameters van het model zelf aan te passen. Daarvoor dubbelklik je op een model. Zo kun je voorkomen dat een model *over-,* of *underfit*.

2.3 Evalueer de modellen met een **Test and Score** module. Net als bij de iris dataset worden er verschillende metrieken weergegeven waar je naar kunt kijken.

2.4 Welk metriek is voor dit probleem het belangrijkst? Waar zou je naar kijken?

2.5 Experimenteer met verschillende modellen en kijk welke het beste presteert.

De drie modellen functioneren anders, en modelleren op een unieke manier de data. Dat betekent dat ze allemaal hun sterke en zwakke punten zullen hebben.

2.6 Voeg een **Predictions** module toe en verbind deze met de **Test and Score** module. In de **Predictions** module kun je in meer detail zien wat voor voorspellingen je modellen maken en kun je ze met elkaar vergelijken.

2.7 Stuur de resultaten van de **Test and Score** module naar een **Confusion Matrix** module en bekijk hoe de confusion matrix eruit ziet. Ben je tevreden met de resultaten? Welk model werkt het beste?

Het is ook mogelijk om de modellen die je getraind hebt te combineren. Je kunt ze bijvoorbeeld allemaal bij elk berichtje laten stemmen of ze denken dat het spam is of niet, en uiteindelijk voor de meerderheid kiezen. We maken dan een *ensemble* aan modellen en gebruiken het ensemble gezamenlijk om voorspellingen te doen. In een aantal modellen zitten dit soort metodes al impliciet verwerkt: *Random forest* is eigenlijk gewoon een ensemble met een heel aantal decision trees die samen stemmen over de oplossing, en ook bij *Gradient boosting* wordt gebruik gemaakt van meerdere decision trees.

2.8 Voeg de module **Stacking** toe. Deze module kan een ensemble maken van meerdere modellen samen.

2.9 Verbind je modellen. Let erop dat je ze verbind als *Learner*.

2.10 Hoe goed doet het ensemble het in vergelijking met de losse modellen? Maakt het veel verschil?

2.11 Het is ook nog mogelijk om weer een apart Machine Learning model te trainen op de output van andere modellen in plaats van gewoon de stem van de meerderheid te kiezen. Daarvoor voeg je een nieuw model toe en verbind je die met de **stacking** module in de rol van *aggregate*.

3 Nieuwe data voorspellen

Nu we een getest model hebben waar we tevreden mee zijn, kunnen we kijken hoe het model het doet op nieuwe, ongeziene data. Er is nog een aparte dataset met 100 nieuwe SMSjes die nog niet gelabeled zijn.

3.1 Voeg de nieuwe dataset toe aan je canvas en preprocess deze data op precies dezelfde manier.

3.2 Voeg een nieuwe **Predictions** module toe en sleep de nieuwe data naar de input.

3.3 Sleep ook een of meerdere van je getrainde modellen naar de input.

3.4 Nu kun je zien hoe je model nieuwe SMSjes markeert als spam of als geen spam. Ben je tevreden met de resultaten?

In deze oefening heb je data geïmporteerd, schoongemaakt en features ge-engineered. Je hebt meerdere machine learning modellen getraind en de resultaten geëvalueeerd. Daarnaast heb je modellen samengevoegd in een *ensemble*. Vevolgens heb je je modellen toegepast op nieuwe, ongeziene data. Dit zijn de stappen die doorgaans doorlopen worden in de machine learning pipeline, en je kunt met dezelfde methode heel veel verschillende datasets aanvliegen.