Machine learning

0365-WFHBOICT.MCL.21

Auteur:

Naam	*Student nummer*	*Email*
Berat Guzel	S1127994	S1127994@student.windesheim.nl
Steven de Valk	S1129787	S1129787@student.windesheim.nl

1. Inhoud

2. Onafhankelijke & afhankelijk variabelen

Onafhankelijk

Onafhankelijke variabelen zijn variabelen die niet voorspelbaar zijn en gebruikt kunnen worden als de inputs die nodig zijn om de afhankelijke variabelen te berekenen/voorspellen, deze variabelen worden dus als 'oorzaak' beschouwt. De onafhankelijke variabelen zijn als volgt beschouwt:

- Aged 65 older
- Aged 70 older
- Cardiovascular death rate
- Continent
- Date
- Diabetes prevalence
- Extreme poverty
- Female smokers
- Gdp per capita
- Handwashing facilities
- Hospital beds per thousand
- Iso code
- Life expectancy and human development index
- Location
- Male smokers
- Median age
- New tests
- Population, population density
- Stringency index
- Tests per case
- Tests units

Dit zijn dus allemaal variabelen die in deze dataset niet voorspelbaar zijn met andere variabelen, en gebruikt kunnen worden om andere variabelen te voorspellen.

Afhankelijk

Afhankelijke variabelen zijn variabelen die voorspeld kunnen worden doormiddel van bijvoorbeeld een berekening met onafhankelijke variabelen. Deze variabelen worden dus als 'gevolg' beschouwt, De afhankelijke variabelen zijn als volgt beschouwt:

- New cases
- New deaths
- New cases per million
- New cases smoothed
- New cases smoothed per million
- New deaths per million
- New deaths smoothed

- New deaths smoothed per million
- New tests per thousand
- New tests smoothed
- New tests smoothed per thousand
- Positive rate
- Total cases
- Total cases per million
- Total deaths
- Total deaths per million
- Total tests
- Total tests per thousand

Dit zijn dus allemaal variabelen die bijvoorbeeld doormiddel van onafhankelijke variabelen in een berekening te gebruiken, het zijn variabelen die een gevolg weergeven.

3. The dataset

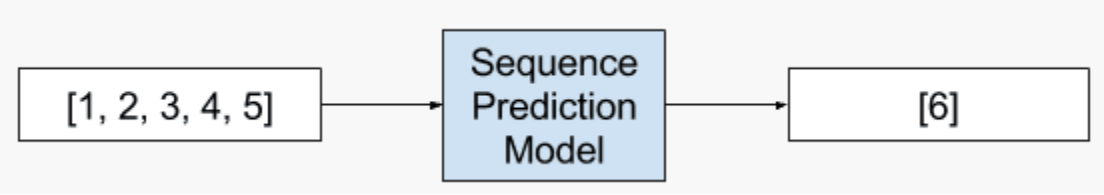
Tijdens het inzien van de dataset was het al meteen duidelijk dat er een heleboel datapunten incompleet waren. Wij hadden ervoor gekozen om een ML model te ontwikkelen waarmee er een voorspelling op de 'total_deaths' variabel gemaakt kon worden, voordat we konden beginnen met het bouwen van het model moesten wij kijken of andere variabelen effect hadden op de 'total_deaths' variabel. Om dit te doen wilden wij gebruik maken van lineaire regressie, hierbij hebben wij de volgende stappen genomen om de dataset compleet te krijgen om regressie uit te kunnen voeren.

- We hebben eerst onze scope afgebakend waarbij we alleen keken naar de datapunten die in Europa gevestigd waren.
- Variabelen die samengevoegd konden worden hebben we samengevoegd, dit waren waardes zoals 'female_smokers' en 'male_smokers'.
- Variabelen waarbij wij het eens over waren dat het geen effect zou hebben op de 'total_deaths' hebben wij handmatig verwijderd. Dit waren variabelen zoals 'new_cases_smoothed' en 'new_tests'.
- Sommige variabelen die een aantal datapunten misten konden handmatig ingevoerd worden met betrekking tot de datapunten die er omheen stonden. Dit waren variabelen zoals 'total_cases' en 'new_cases'.
- Ten slotte hebben we landen waarvan de datapunten gedeeltelijk leeg waren en het een dynamische waarde was gebruik gemaakt van interpolatie. Interpolatie is het voorspellen van de waardes met voorafgaande en/of aansluitende waardes. Als het een statische waarde was hebben wij dezelfde waarde met betrekking tot de groep ingevuld.

De regressie resultaten kunnen in de 'Regression_Plots' map in de repository terug gevonden worden.

4. Keuze model

Na regressie zijn wij tot de conclusie gekomen dat er geen variabelen waren die specifiek effect uitoefende op de 'total_deaths' variabel, hierbij hebben wij rondgezocht naar een passend model om alsnog de 'total_deaths' variabel te kunnen voorspellen. We zochten dus naar een model waarmee het mogelijk zou zijn doormiddel van een serie aan nummers het volgende nummer in de reeks te voorspellen (Afbeelding 4.1).



Figuur 4.1: Voorbeeld van het voorspellen van het volgende nummer in een reeks.

Toen wij online keken naar verschillende voorbeelden in de praktijk hebben wij opgemerkt dat iedereen hiervoor een neuraal netwerk voor hadden gebouwd, en konden we geen specifiek voorbeeld vinden waarbij er alleen gebruik werd gemaakt van een simpel ML model. Dit komt vanwege het feit dat dit veel betrouwbaarder is om met een neuraal netwerk uit te voeren.

Na het zoeken naar een passende modellen zijn wij op de volgende twee keuzes een Recurrent Neural Network model (RNN-model), of een Long Short Term Memory model (LSTM-model). Beide modellen doen in principe hetzelfde, namelijk het voorspellen van de volgende waarde doormiddel te kijken naar de vorige waardes. Hierbij is het LSTM-model een aftakking van het RNN-model, het belangrijkste verschil is dat tussen de twee modellen is dat LSTM zowel gebruik maakt van short- en longterm memory terwijl RNN alleen gebruik maakt van shortterm memory [1]. Dit was voor ons de besluitende factor sinds wij van mening waren dat dit veel effectiever zou zijn op de dataset die wij ter beschikking hadden, het model zou theoretisch gezien meer naar het verleden kunnen kijken en hier rekening mee kunnen houden.

5. Training en validatie data

Onze dataset is op chronologische volgorde met betrekking tot de datum geordend, bij het splitsen van deze dataset in training en test data hebben wij ervoor gekozen om de komende vuistregels te gebruiken [2]:

- We beginnen eerst met een 80/20 splits om te kijken hoe effectief dit is, dit houdt met onze chronologische dataset in dat de eerste 80 procent wordt gebruikt om het model te trainen, en dat de laatste 20 procent gebruikt wordt als test data. Dit houdt in principe in dat als we een dataset hebben van 10 maanden dat de eerste 8 maanden uit training data en de laatste 2 maanden uit test data bestaan.
- De resultaten/effectiviteit hiervan worden ergens gemarkeerd
- Vervolgens zullen we het model opnieuw proberen te train met verschillende splitsingen zoals 70/30 en/of 90/10.
- De resultaten/effectiviteit per split wordt ergens gemarkeerd en vergeleken met de 80/20 split, hierbij worden opgemerkte verschillen in acht genomen.
- Aan het einde zal de split met de beste resultaten/effectiviteit gekozen worden als de splitsing die voor het model gebruikt zal worden.

Wij hebben deze vuistregels toegepast bij ons model en zijn tot de 80/20 Splitsing gekomen als de meest effectieve splitsing.

6. Supervised learning model prestatie

De stappen en keuzes die hierboven staan beschreven zijn gebruikt en toegepast om het model te ontwikkelen, er is dus een neuraal netwerk met een LTSM-model ontwikkeld waarbij we de totale doden per dag voorspeld worden door te kijken naar wat de doden voorheen waren. De code die wij hiervoor hebben geschreven is terug te vinden in onze repository in het 'Total_Deaths_LSTM.ipynb' bestand, Hierbij wordt er in de code per functie en/of lijn met een kleine comment toegelicht wat het is en/of waar het gebruikt voor wordt. Hierbij kan er in de repository de datasets, code voor het uitvoeren van de regressie, de 'Predict_Total_Deaths_With_Saved_Model.ipynb' bestand, en een map met alle resultaten van de regressies, terug gevonden worden.

Het neurale netwerk heeft met een training sessie van 30 epochs de volgende resultaten bereikt:

```
12/12 [==============================] - 0s 3ms/step - loss: 282820608.0000 - accuracy: 0.1839
Epoch 29/30
12/12 [==============================] - 0s 3ms/step - loss: 406638208.0000 - accuracy: 0.1839
Epoch 30/30
12/12 [==============================] - 0s 3ms/step - loss: 383565376.0000 - accuracy: 0.1839
```

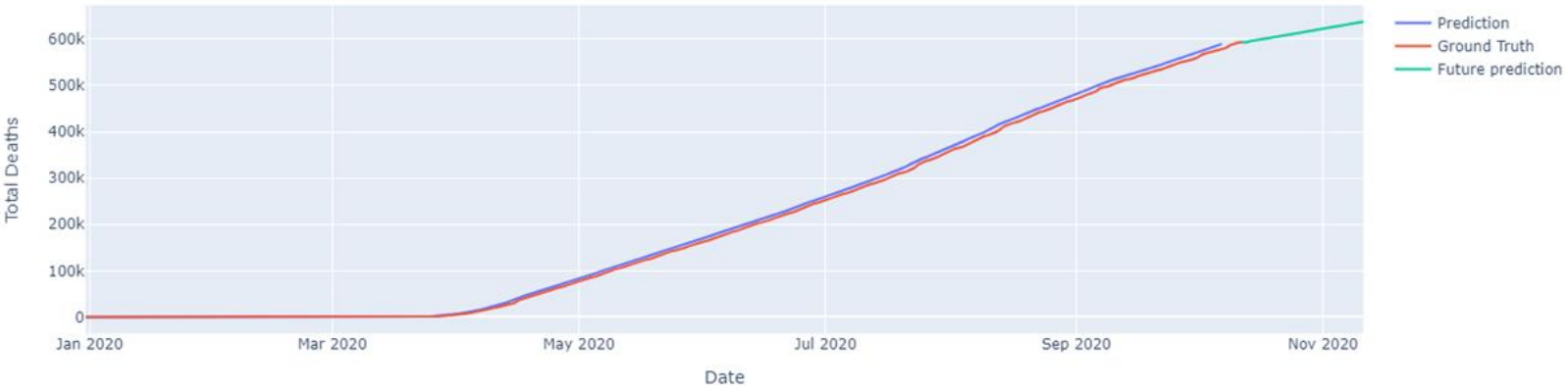
Figuur 6.1: Resultaten model.

De accuraatheid en de loss geven het model niet het beste beeld, je zou namelijk een model die niet eens 20 procent accuraatheid bereikt en aangeeft dat het veel verkeerd voorspeld niet vertrouwen. Echter is dat in deze situatie niet het geval, als je de voorspelde resultaten en de daadwerkelijke resultaten in een grafiek zou plotten zou dit er als volgt uit zien:

Figuur 6.2: De voorspelde waardes vergeleken met de echte waardes.

Uit de afbeeldingen is het te zien dat de voorspellingen de totale doden zodanig goed hebben voorspeld dat het bijna lijkt dat de twee lijnen met elkaar overlappen. Het feit dat het model een lage accuraatheid heeft komt vanwege het feit dat het model af en toe een aantal doden ernaast voorspeld, wat dus niet als een juiste waarde wordt geteld in het model. Wij zochten echter niet naar precieze waardes en wouden voorspellingen maken dat een realistisch beeld zou kunnen geven over wat de komende totale doden zouden kunnen zijn, en wij zijn van mening dat we dat bereikt hebben met dit model.

Ook hebben we de na het trainen van het model deze opgeslagen en op een andere dataset getest, dit hebben we in de 'Predict_Total_Deaths_With_Saved_Model.ipynb' bestand uitgevoerd. In dit geval hebben we een dataset voorbereid die zich focuste op noord en zuid Amerika, ook hebben we in dit model een toekomstige voorspelling van 30 dagen toegevoegd. De resultaten hebben we in een grafiek afgebeeld, en deze zagen er als volgt uit



Figuur 6.3: Voorspellingen met het opgeslagen model op de Amerika dataset.

Zoals te zien heeft dit ook net als in de test data bijna precies dezelfde waardes voorspeld als de daadwerkelijke waardes, hiermee waren wij zelfverzekerd over het feit dat het neurale netwerk op een accurate manier de totale hoeveelheid aan doden kan voorspellen aan de hand van voorgaande waardes.

7. Inzichten

Zoals te zien in de grafieken in afbeelding 6.2 en 6.3 maakt het model accurate voorspellingen over de totale doden, alhoewel er alsnog een verschil zit in de daadwerkelijke waardes is dit alsnog te verwaarlozen sinds de voorspelde waardes niet veel verschillen met de daadwerkelijke waardes. Hieruit zijn wij van mening dat het model op een correcte manier voorspellingen kan maken.

8. Bronnen

[1] A. Tripathi, „What is the main difference between RNN and LSTM | NLP | RNN vs LSTM,” Data Science Duniya, 05 10 2021. [Online]. Available: https://ashutoshtripathi.com/2021/07/02/what-is-the-main-difference-between-rnn-and-lstm-nlp-rnn-vs-lstm/. [Geopend 28 01 2022].

[2] B. Allison, „Is there a rule-of-thumb for how to divide a dataset into training and validation sets?,” Stackoverflow, 28 11 2012. [Online]. Available: https://stackoverflow.com/questions/13610074/is-there-a-rule-of-thumb-for-how-to-divide-a-dataset-into-training-and-validatio. [Geopend 25 01 2022].