

## Tijdreeks voorspellende modellen:

### 1. Autoregression (AR)

Inhoud:

Modelleert de volgende stap in een volgorde van een lineaire functie van vorige tijdstappen.

Voor- en nadelen:

Redelijk simpele code om toe te passen.

Gebruikt voorgaande data om toe te passen op de toekomst, kan fout voorspellen door grote economische veranderingen.

Toepasbaarheid:

Nee, Redelijk simpele code om toe te passen, maar speelt niet in op trends en seizoenen

```
# AR example
from statsmodels.tsa.ar_model import AutoReg
from random import random
# contrived dataset
data = [x + random() for x in range(1, 100)]
# fit model
model = AutoReg(data, lags=1)
model_fit = model.fit()
# make prediction
yhat = model_fit.predict(len(data), len(data))
print(yhat)
```

### 2. Moving Average (MA)

Inhoud: De Moving Average methode modelleert de volgende stap in de reeks als een lineaire functie van een gemiddeld proces in eerdere tijdstappen en eerdere foutmarges. De lineaire functie hangt dus af van voorgaande waarden uit het verleden.

Voor- en/of Nadelen: De methode is geschikt voor tijdreeksen zonder trends en seizoenscomponenten. Tegengesteld met het AR model, het MA model eindigt altijd stationair (dat wil zeggen dat er geen grote spelingen in het model zullen voorkomen )

Toepasbaar?:Nee, want de data die onze groep heeft bevat wel trends en seizoenscomponenten.

```
1 # MA example
2 from statsmodels.tsa.arima_model import ARMA
3 from random import random
4 # contrived dataset
5 data = [x + random() for x in range(1, 100)]
6 # fit model
7 model = ARMA(data, order=(0, 1))
8 model_fit = model.fit(dispatch=False)
9 # make prediction
10 yhat = model_fit.predict(len(data), len(data))
11 print(yhat)
```

### 3. Autoregressive Moving Average (ARMA)

Inhoud:ARMA is een mix van AR& MA

Voor- en/of Nadelen:het is niet stationair wat neerkomt dat het niet geschikt is voor modellen met trends en seizoenscomponenten

Toepasbaar?:nee,want de data die onze groep heeft bevat wel trends en seizoenscomponenten

```
1 # ARMA example
2 from statsmodels.tsa.arima_model import ARMA
3 from random import random
4 # contrived dataset
5 data = [random() for x in range(1, 100)]
6 # fit model
7 model = ARMA(data, order=(2, 1))
8 model_fit = model.fit(dispatch=False)
9 # make prediction
10 yhat = model_fit.predict(len(data), len(data))
11 print(yhat)
```

### 4. Autoregressive Integrated Moving Average (ARIMA)

Inhoud:ARIMA is een mix van AR & MA, maar r komt integratie(I) bij om het model stationair te maken

Voor- en /of Nadelen: De methode is geschikt voor tijdreeksen met trends en zonder seizoens componenten

Toepasbaar?:ja, er komen veel trends naar voren binnen ons model

```
1 # ARIMA example
2 from statsmodels.tsa.arima_model import ARIMA
3 from random import random
4 # contrived dataset
5 data = [x + random() for x in range(1, 100)]
6 # fit model
7 model = ARIMA(data, order=(1, 1, 1))
8 model_fit = model.fit(dispatch=False)
9 # make prediction
10 yhat = model_fit.predict(len(data), len(data), typ='levels')
11 print(yhat)
```

## 5. Seasonal Autoregressive Integrated Moving-Average (SARIMA)

Inhoud: SARIMA is een mix van AR & MA, maar r komt integratie(I) bij om het model stationair te maken. En een S bij om seizoenscomponenten geschikt te maken voor het model.

Voor- en/of Nadelen: De methode is geschikt voor tijdreeksen met trends en seizoens componenten

Toepasbaar?: ja, er komen veel trends en seizoens gebonden data naar voren binnen ons model

```
1 # SARIMA example
2 from statsmodels.tsa.statespace.sarimax import SARIMAX
3 from random import random
4 # contrived dataset
5 data = [x + random() for x in range(1, 100)]
6 # fit model
7 model = SARIMAX(data, order=(1, 1, 1), seasonal_order=(1, 1, 1, 1))
8 model_fit = model.fit(dispatch=False)
9 # make prediction
10 yhat = model_fit.predict(len(data), len(data))
11 print(yhat)
```

## 6. Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX)

Inhoud: SARIMAX is een mix van AR & MA, maar r komt integratie(I) bij om het model stationair te maken. En een S bij om seizoenscomponenten geschikt te maken voor het model. De X die er bij komt kijken staat voor een melding van buiten de data

Voor- en/of Nadelen: De methode is geschikt voor tijdreeksen met trends en seizoens componenten en exogene variabelen. De exogene variabelen kunnen ook toegepast worden als arx,max,armax en arimax

Toepasbaar?:nee, niet veel te maken met exogene variabelen.

```
1 # SARIMAX example
2 from statsmodels.tsa.statespace.sarimax import SARIMAX
3 from random import random
4 # contrived dataset
5 data1 = [x + random() for x in range(1, 100)]
6 data2 = [x + random() for x in range(101, 200)]
7 # fit model
8 model = SARIMAX(data1, exog=data2, order=(1, 1, 1), seasonal_order=(0, 0, 0, 0))
9 model_fit = model.fit(dispatch=False)
10 # make prediction
11 exog2 = [200 + random()]
12 yhat = model_fit.predict(len(data1), len(data1), exog=[exog2])
13 print(yhat)
```

## 7. Vector Autoregression (VAR)

Inhoud: Gebruikt AR model als parameter voor de VAR functie, werkt bijna hetzelfde als AR

Voor- en/of Nadelen: Niet specifiek genoeg

Toepasbaar?: Niet voor onze situatie

```
1 # VAR example
2 from statsmodels.tsa.vector_ar.var_model import VAR
3 from random import random
4 # contrived dataset with dependency
5 data = list()
6 for i in range(100):
7     v1 = i + random()
8     v2 = v1 + random()
9     row = [v1, v2]
10    data.append(row)
11 # fit model
12 model = VAR(data)
13 model_fit = model.fit()
14 # make prediction
15 yhat = model_fit.forecast(model_fit.y, steps=1)
16 print(yhat)
```

## 8. Vector Autoregression Moving-Average (VARMA)

Inhoud: gebruikt AR en MA als parameters

Voor- en/of Nadelen: iets uitgebreider maar niet uitgebreid genoeg t.o.v. VAR

Toepasbaar?: Redelijk, waarschijnlijk niet voor onze situatie

```
1 # VARMA example
2 from statsmodels.tsa.statespace.varmax import VARMAX
3 from random import random
4 # contrived dataset with dependency
5 data = list()
6 for i in range(100):
7     v1 = random()
8     v2 = v1 + random()
9     row = [v1, v2]
10    data.append(row)
11 # fit model
12 model = VARMAX(data, order=(1, 1))
13 model_fit = model.fit(dispatch=False)
14 # make prediction
15 yhat = model_fit.forecast()
16 print(yhat)
```

## 9. Vector Autoregression Moving-Average with Exogenous Regressors (VARMAX)

Inhoud: modelled niet op de zelfde manier als primary modellen zoals AR en MA

Voor- en/of Nadelen: Houdt meer rekening met andere factoren, maar alleen voor data zonder seizoenen

Toepasbaar?: nee, naar verwachting nie te gebruiken voor onze dataset

```
1 # VARMAX example
2 from statsmodels.tsa.statespace.varmax import VARMAX
3 from random import random
4 # contrived dataset with dependency
5 data = list()
6 for i in range(100):
7     v1 = random()
8     v2 = v1 + random()
9     row = [v1, v2]
10    data.append(row)
11 data_exog = [x + random() for x in range(100)]
12 # fit model
13 model = VARMAX(data, exog=data_exog, order=(1, 1))
14 model_fit = model.fit(dispatch=False)
15 # make prediction
16 data_exog2 = [[100]]
17 yhat = model_fit.forecast(exog=data_exog2)
18 print(yhat)
```

#### 10. Simple Exponential Smoothing (SES)

Inhoud: Simpel model die volgende step modelleerd

Voor- en/of Nadelen: Niet te gebruiken met trends en seasonal componenten

Toepasbaar?: Nee, niet te gebruiken doordat er geen rekening wordt gehouden met trends

```
1 # SES example
2 from statsmodels.tsa.holtwinters import SimpleExpSmoothing
3 from random import random
4 # contrived dataset
5 data = [x + random() for x in range(1, 100)]
6 # fit model
7 model = SimpleExpSmoothing(data)
8 model_fit = model.fit()
9 # make prediction
10 yhat = model_fit.predict(len(data), len(data))
11 print(yhat)
```

#### 11. Holt Winter's Exponential Smoothing (HWES)

Inhoud:dit is een exponentieel gewogen lineaire functie van waarnemingen in eerdere tijdsreeksen, rekening houdend met trends en seizoensinvloeden.

Terwijl in MA de waarnemingen uit het verleden gelijk

worden gewogen, worden exponentiële functies gebruikt om exponentieel afnemende gewichten in de tijd toe te wijzen.

Holt winter exponential smoothing past driemaal exponentiële afvlakking toe, die gewoonlijk wordt gebruikt wanneer er drie hoogfrequente signalen moeten worden verwijderd uit een tijdreeks die wordt bestudeerd.

Voor- en/of Nadelen: De methode is geschikt voor tijdreeksen met trends en seizoenscomponenten.

Het is een gemakkelijk te leren en toe te passen procedure om een besluit te nemen op basis van eerdere aannames van de data. Maar vraagt veel features en tijd

Toepasbaar?: Nee, want onze dataset bevat niet teveel features

Dit zijn 11 modellen die te gebruiken zijn voor tijdreeksen.

Uit deze modellen uitzoeken welk model te gebruiken zijn voor onze situatie.

Kijken naar:

- Inhoud
- Voor en na delen
- Toepasbaarheid

Bruikbare modellen voor ons project:

1. AutoRegression(AR)
2. Seasonal Autoregressive Integrated Moving-Average / Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors(SARIMA/SARIMAX)
1. Vector Autoregression/ Vector Autoregression Moving-Average with Exogenous Regressors(VAR/VARMAX)
3. Holt Winter's Exponential Smoothing (HWES)

Sarima:

Inhoud:

Seasonal AutoRegressive Integrated Moving Average is een uitbreiding van ARIMA dat time series data ondersteunt met seizoengebonden componenten.

Het voegt 3 nieuwe hyperparameters toe om de AutoRegression, differentiatie en het moving average te specificeren voor de seizoensgebonden componenten van de timeseries. Ook is er een parameter te vinden voor de periode van een seizoen.

Het configureren van SARIMA benodigt het selecteren van 3 hyperparameters voor de trend en 3 voor de seizoenselementen van de timeserie.

Trend:

p -> Trend autoregression order

d -> Trend difference order / is the number of differencing required to make the time series stationary

q -> Trend moving average order

Seizoensgebonden elementen:

P -> Seasonal autoregressive order

D -> Seasonal difference order

Q -> Seasonal moving average order

m -> The number of time steps for a single seasonal period

Code:

`SARIMA(p,d,q)(P,D,Q)m`

Te gebruiken in Python:

