

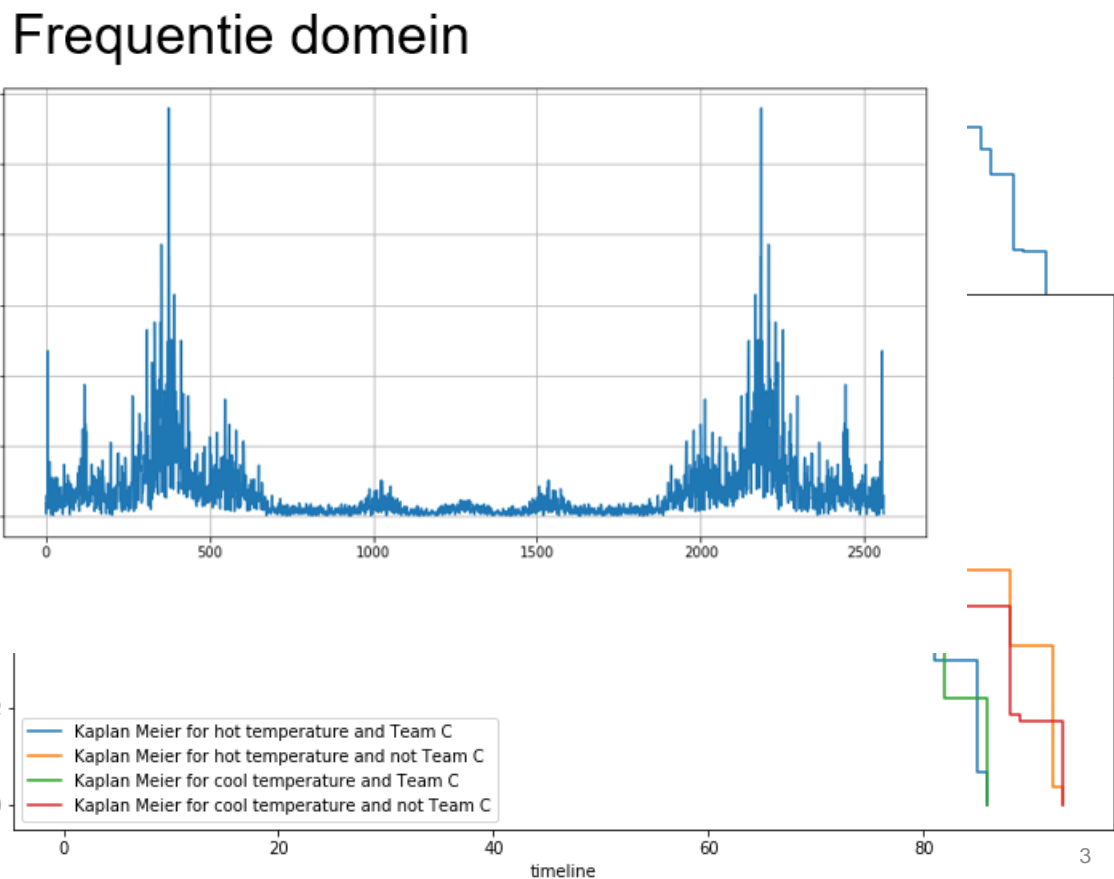
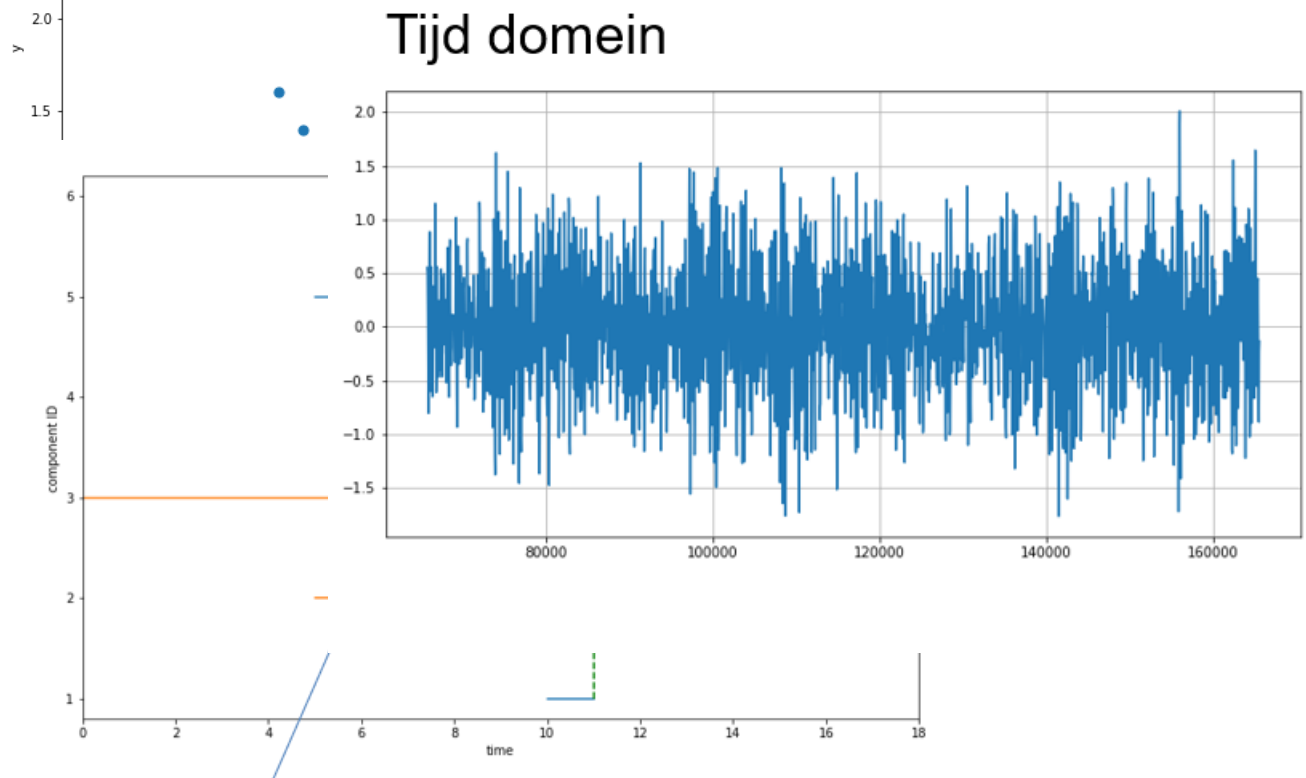
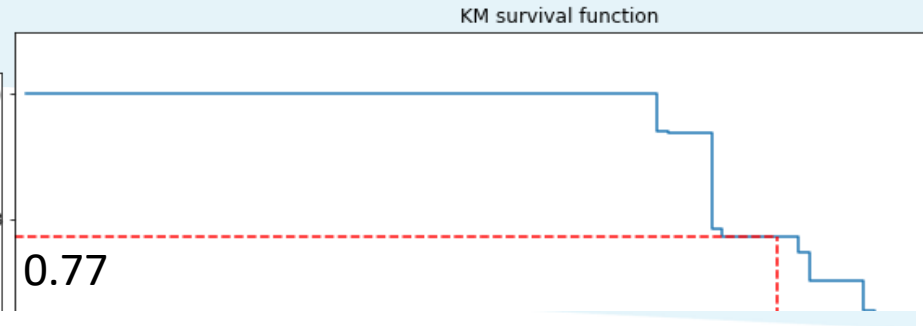
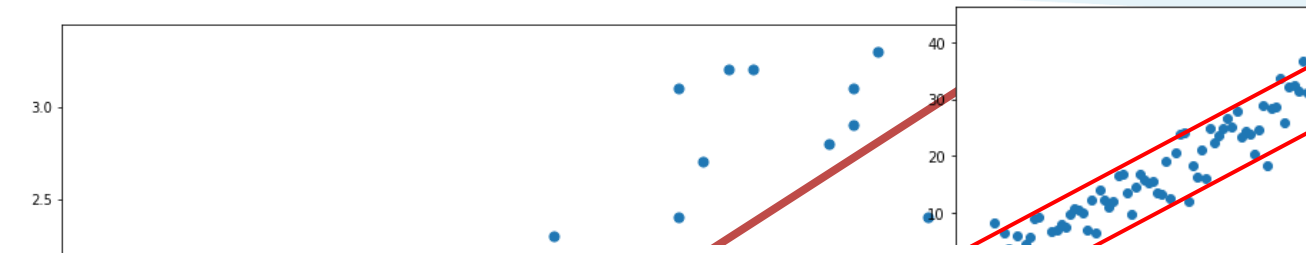
Simcha van Helvoort
Dag 3



AI voor Engineers

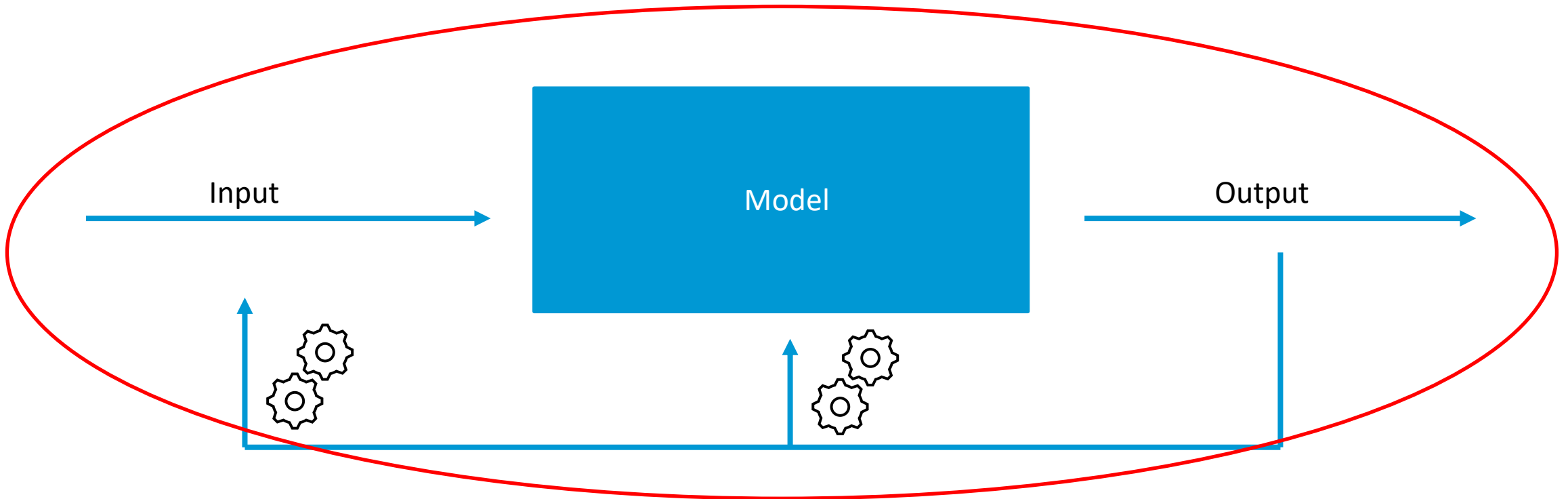
- De cursist snapt het verschil tussen ruwe data en input data
- De cursist kan van ruwe data, input data maken in Python
- De cursist kan omgaan met feature scaling, uitschieters en andere data cleaning technieken
- De cursist snapt wat overfitting is en hoe dit tegengegaan moet worden
- De cursist kan met ruwe data een geschikt model maken om specifieke analyses te doen
- De cursist weet hoe een model geëvalueerd moet worden en kan op basis daarvan de volgende stappen voor het model bepalen

Samenvatting vorige week



Censored data

HUISWERK OPDRACHT



EVALUATIE

Performance metrics (classification)

		Predicted	
		A	B
Ground truth	A	True A	False B
	B	False A	True B

$$\text{Accuracy} = \frac{\text{True A} + \text{True B}}{\text{True A} + \text{False A} + \text{True B} + \text{False B}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$$

		Positive	Negative
Ground truth	Positive	TP	FN
	Negative	FP	TN

Performance metrics (classification)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

➔ `sklearn.metrics.precision_score`
Hoeveel van de positieve voorspellingen waren goed?

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

➔ `sklearn.metrics.recall_score`
Hoeveel van de positieve datapunten waren goed?

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

➔ `sklearn.metrics.f1_score`
Mix van Precision en Recall



Ground truth
Negative Positive

Predicted	
Positive	Negative
40	5
10	45

Voorbeeld: voorspellen of een machine onderhoud nodig heeft.

40 machines correct voorspeld dat ze onderhoud nodig hebben

45 machines correct voorspeld dat ze geen onderhoud nodig hebben

10 machines fout voorspeld dat ze onderhoud nodig hebben, terwijl ze dat niet nodig hadden

5 machines fout voorspeld dat ze geen onderhoud nodig hebben, terwijl ze dat wel nodig hadden

Voorbeeld: voorspellen of een machine onderhoud nodig heeft.

40 machines correct voorspeld dat ze onderhoud nodig hebben

45 machines correct voorspeld dat ze geen onderhoud nodig hebben

10 machines fout voorspeld dat ze onderhoud nodig hebben, terwijl ze dat niet nodig hadden

5 machines fout voorspeld dat ze geen onderhoud nodig hebben, terwijl ze dat wel nodig hadden

Ground truth
Negative Positive

Predicted	
Positive	Negative
40	5
10	45

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \rightarrow \frac{40}{40 + 10} = 0.8$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \rightarrow \frac{40}{40 + 5} = 0.89$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \rightarrow \frac{2 \times 0.8 \times 0.89}{0.8 + 0.89} = 0.84$$

Voorbeeld: voorspellen of een machine onderhoud nodig heeft.

80% van de voorspelde machines dat onderhoud nodig had, had daadwerkelijk onderhoud nodig

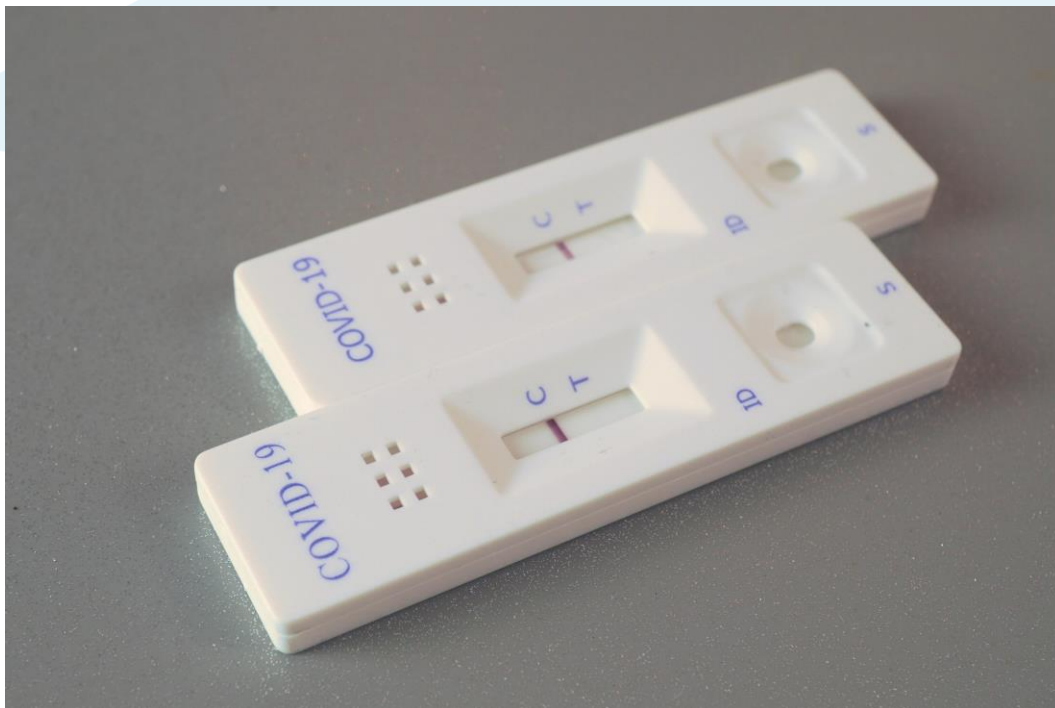
89% van de machines dat onderhoud nodig had, zijn correct voorspeld

		Predicted	
		Positive	Negative
Ground truth	Positive	40	5
	Negative	10	45

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \rightarrow \frac{40}{40 + 10} = 0.8$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \rightarrow \frac{40}{40 + 5} = 0.89$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \rightarrow \frac{2 \times 0.8 \times 0.89}{0.8 + 0.89} = 0.84$$



Ground truth
Negative Positive

Predicted	
Positive	Negative
7	10
14	840

Voorbeeld: corona antigeentesten onder 871 studenten zonder symptomen¹

7 positieve testen die correct waren²

14 positieve testen die incorrect waren

10 negatieve testen die incorrect waren

840 negatieve testen die correct waren

¹ https://www.cdc.gov/mmwr/volumes/69/wr/mm695152a3.htm#T2_down

² Correct betekent dat ze dezelfde uitslag namen als een PCR test. We nemen aan de de PCR de waarheid is

Voorbeeld: corona antigeentesten
onder 871 studenten zonder
symptomen

7 positieve testen die correct waren

14 positieve testen die incorrect waren

10 negatieve testen die incorrect waren

840 negatieve testen die correct waren

		Predicted	
		Positive	Negative
Ground truth	Positive	7	10
	Negative	14	840

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \rightarrow \frac{7}{7 + 14} = 0.33$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \rightarrow \frac{7}{7 + 10} = 0.41$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \rightarrow \frac{2 \times 0.33 \times 0.41}{0.33 + 0.41} = 0.37$$

CONTEXT IS ALLES

Hoe werden de testen voor studenten gebruikt?

Neem een test voordat je naar school gaat. Is het negatief? Dan ben je negatief.

Is het positief? Doe een PCR test

Voorbeeld: corona antigeentesten onder 871 studenten zonder symptomen

33% van de positieve testen was correct

41% van de positieve mensen kreeg positief resultaat

98% van de negatieve mensen kreeg negatief resultaat

		Predicted	
		Positive	Negative
Ground truth	Positive	7	10
	Negative	14	840

Specificity = $\frac{TN}{TN + FP} \rightarrow \frac{840}{840 + 14} = 0.98$

$$\text{Precision} = \frac{TP}{TP + FP} \rightarrow \frac{7}{7 + 14} = 0.33$$

$$\text{Recall} = \frac{TP}{TP + FN} \rightarrow \frac{7}{7 + 10} = 0.41$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \rightarrow \frac{2 \times 0.33 \times 0.41}{0.33 + 0.41} = 0.37$$

Deze metrics zijn allemaal voorbeelden dus:

**BEDENK WAT JE WIL EVALUEREN,
ZOEK/BEDENK DAN PAS EEN METRIC**

- Bereken de Precision, Recall en F1 score van alle classificatie modellen van vorige week
- Maak ook een confusion matrix van elk classificatie model
- Welk model is beter en waarom? Hoe is dit anders dan naar de accuracy kijken?

REGRESSIE: EEN SIMPELER MODEL IS BETER

Performance metrics (regression)

Meer variabelen =
groter R^2

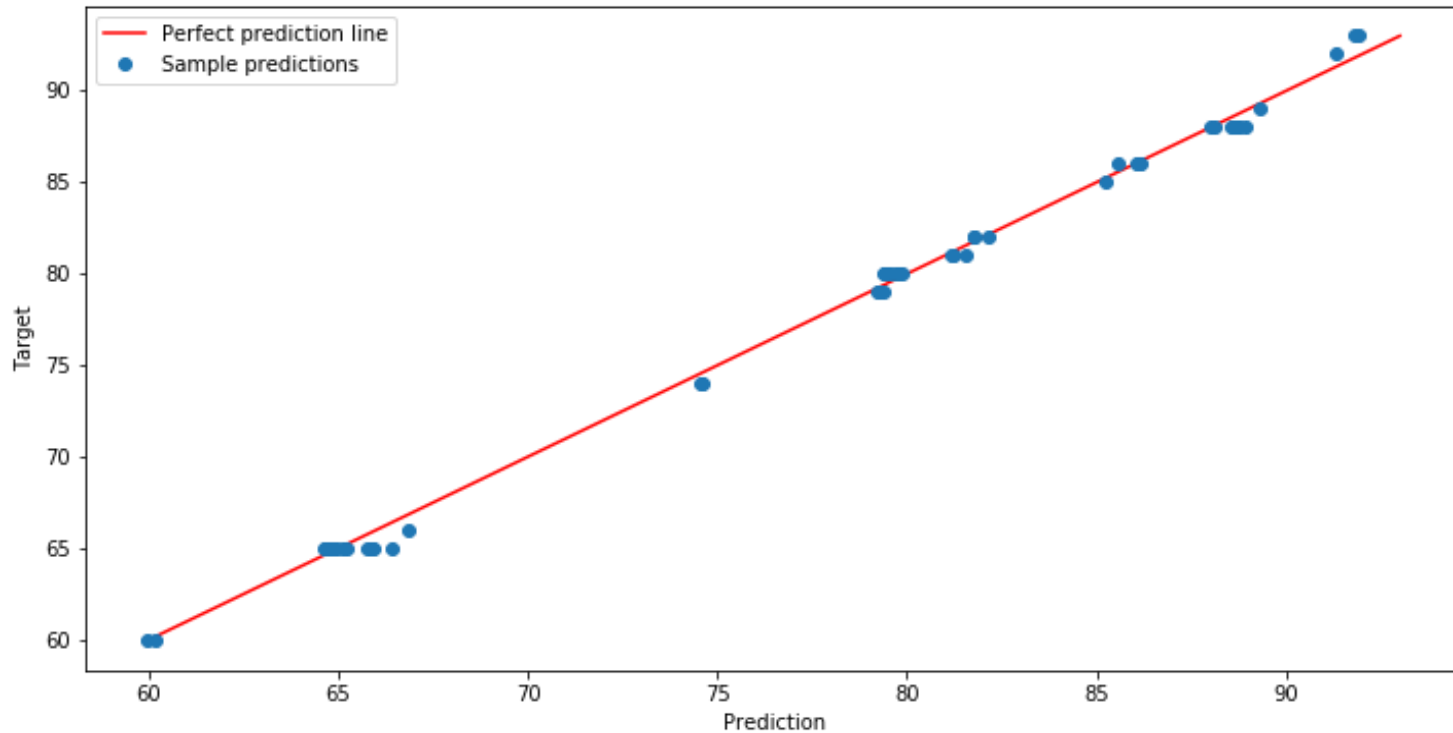
$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \rightarrow \text{sklearn.metrics.r2_score}$$

$$R^2_{\text{adj}} = 1 - (1 - R^2) \frac{n - 1}{n - p - 1} \rightarrow \text{Geen sklearn functie}$$

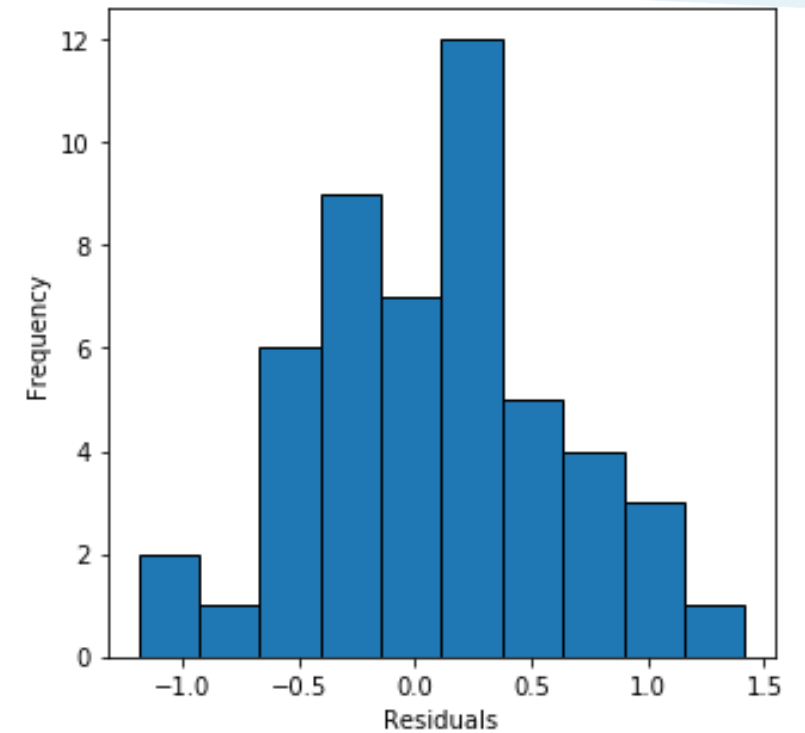
n = aantal observaties

p = aantal verklarende variabele

Visualisaties



```
plt.plot([y.min(), y.max()], [y.min(), y.max()], c = "r")  
plt.plot(ypred, y, "o")
```



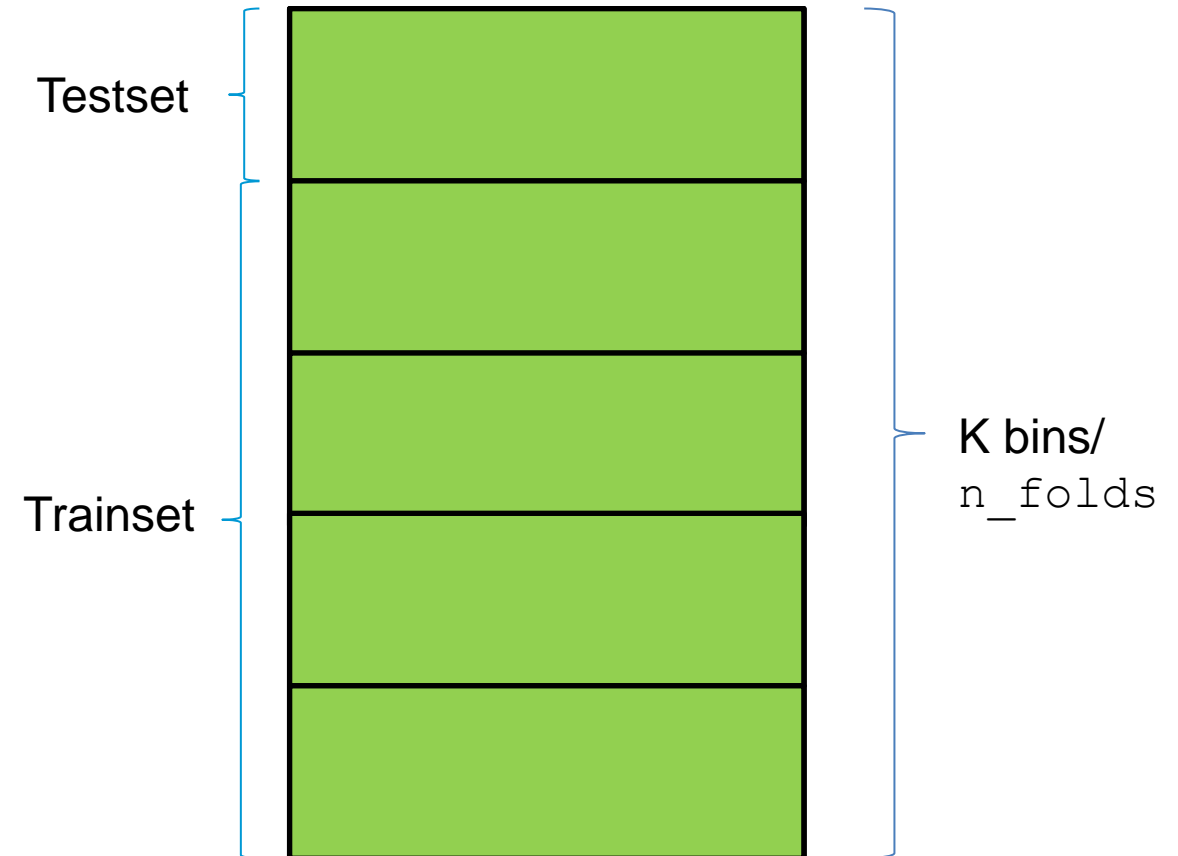
```
plt.hist(ypred-y, edgecolor = "black")
```

- Bereken de R^2 en R^2_{adj} van je regressiemodellen
- Visualiseer de uitkomsten van je regressiemodel

Verandert dit welk regressie model beter is? (Linear, ridge, lasso...)

GRID SEARCH CROSS-VALIDATION

- K-Fold Cross Validation
- Nut van Gridsearch CV
 - Het zoeken naar de beste hyperparameters
 - Valideren of die parameters ook generiek genoeg zijn
- Hoe werkt het

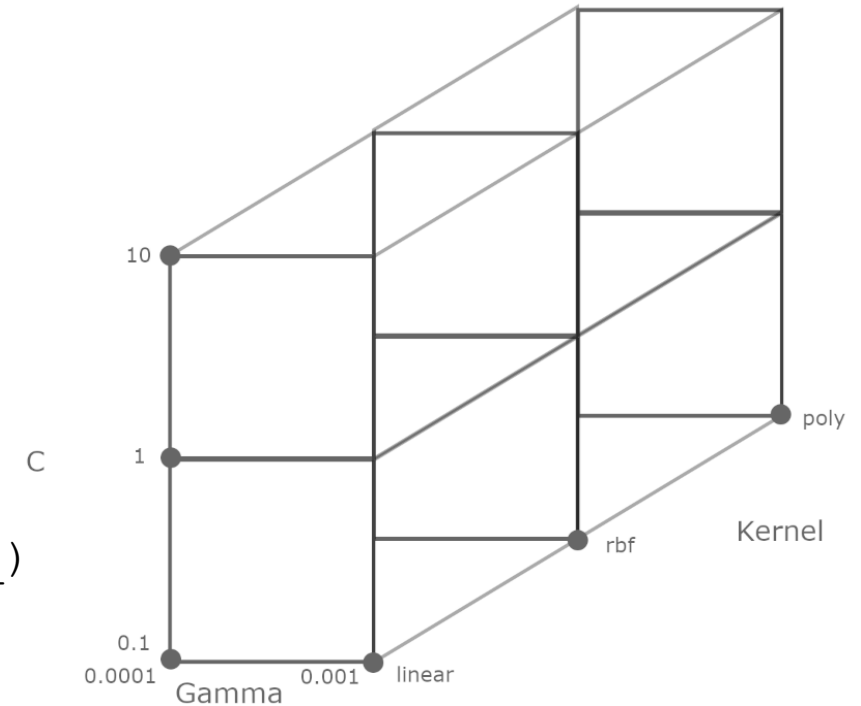


Gridsearch CV

```
from sklearn.svm import SVC
svmc = SVC()
parameter_space = {
    'gamma': [0.0001, 0.001],
    'C': [0.1, 1, 10],
    'kernel': ["rbf", "linear", "poly"]
}
gridcv = GridSearchCV(svmc, parameter_space, cv=3)
gridcv.fit(X, Y)

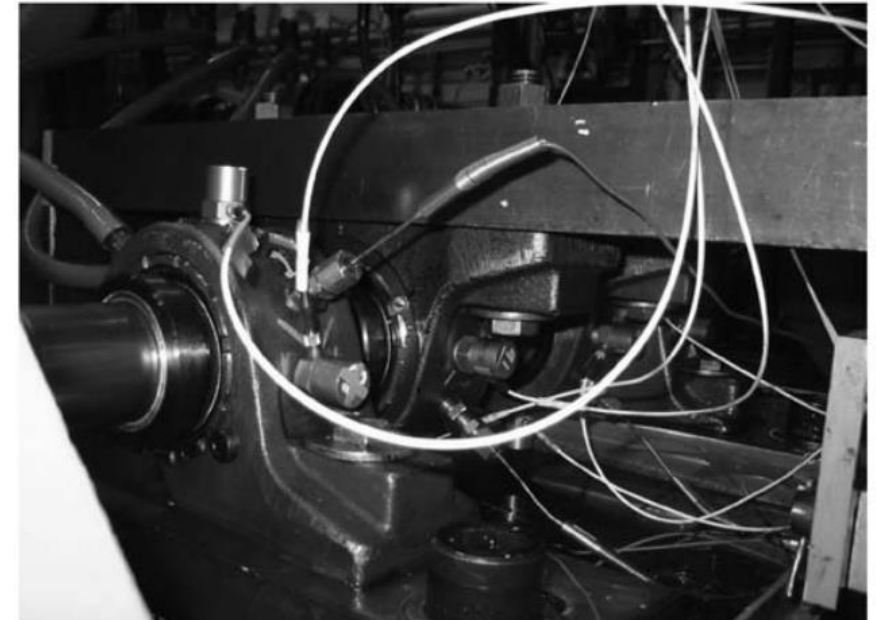
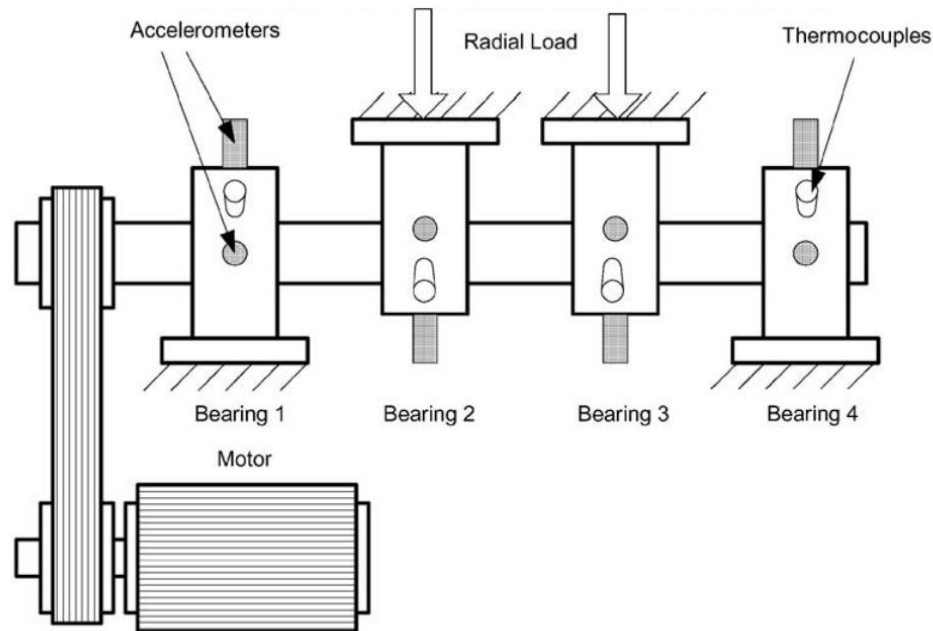
# Best parameter set
print('Best parameters found:\n', gridcv.best_params_)
print('With score:\n', gridcv.best_score_)

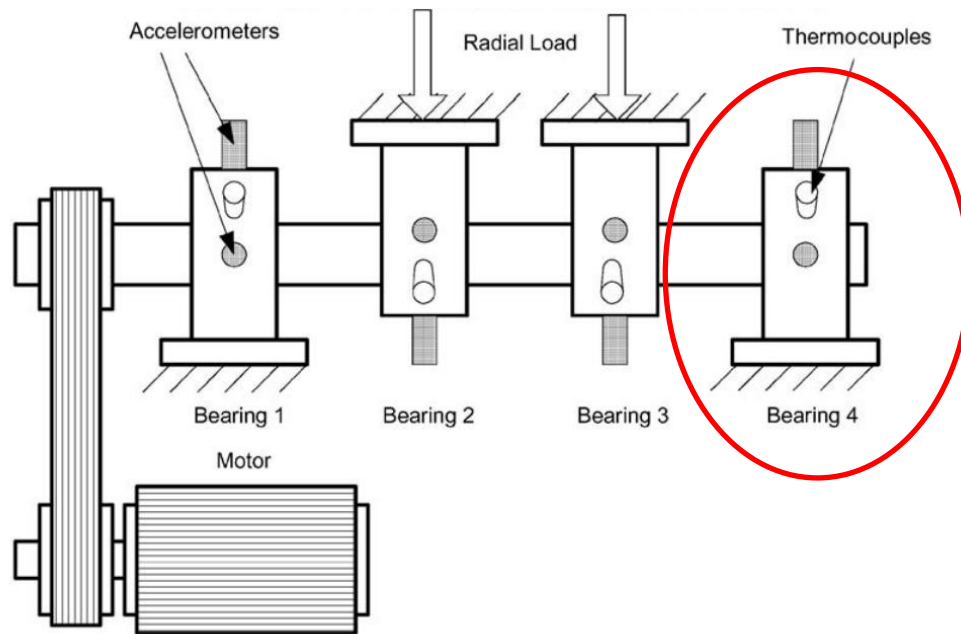
Best parameters found:
{'C': 0.1, 'gamma': 0.0001, 'kernel': 'linear'}
With score:
1.0
```
























CASUS – LASTIGE LOGGE LAGERS LATEN LOS

- IMS dataset
 - Format
 - Betekenis klassen
- Challenge





0 = Vroeg
1 = Normaal
2 = Verdacht
3 = Rol element *failure*
4 = Failure

Name	Status	Date modified	Type	Size
 .ipynb_checkpoints	✓	28-6-2021 15:57	File folder	
 0.txt	✓	6-10-2020 15:07	TXT File	157 KB
 1.txt	✓	6-10-2020 15:07	TXT File	157 KB
 2.txt	✓	6-10-2020 15:07	TXT File	157 KB
 3.txt	✓	6-10-2020 15:07	TXT File	157 KB
 4.txt	✓	6-10-2020 15:07	TXT File	157 KB
 5.txt	✓	6-10-2020 15:07	TXT File	150 KB
 6.txt	✓	6-10-2020 15:07	TXT File	157 KB
 7.txt	✓	6-10-2020 15:07	TXT File	155 KB
 8.txt	✓	6-10-2020 15:07	TXT File	156 KB
 9.txt	✓	6-10-2020 15:07	TXT File	157 KB
...				
 1715.txt	✓	6-10-2020 15:07	TXT File	154 KB
 1716.txt	✓	6-10-2020 15:07	TXT File	154 KB
 1717.txt	✓	6-10-2020 15:07	TXT File	153 KB
 1718.txt	✓	6-10-2020 15:07	TXT File	153 KB
 1719.txt	✓	6-10-2020 15:07	TXT File	153 KB
 1720.txt	✓	6-10-2020 15:07	TXT File	153 KB
 1721.txt	✓	6-10-2020 15:07	TXT File	153 KB
 1722.txt	✓	6-10-2020 15:07	TXT File	153 KB
 1723.txt	✓	6-10-2020 15:07	TXT File	153 KB
 bearing_conditions.csv	✓	6-10-2020 15:07	CSV File	6 KB

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from dataset.bearing_dataset import bearing_dataset
import scipy as sp

import tqdm # voor loading bars
```

```
In [2]: ds = bearing_dataset('dataset/data', 'bearing_conditions.csv')
print(ds)
```

Class bearing_dataset with files from 'dataset/data' and size (1724,). It holds the following files: ['0.txt', '1.txt', '10.txt'] ... ['997.txt', '998.txt', '999.txt'].
Get full list of files with ds.files

```
In [3]: ds.labels
```

Out[3]:

	b4
0	0
1	0
2	0
3	0
4	0
...	...
1719	4



Lees de README

Stappenplan

1. Data correct inladen
2. Data visualiseren (gevoel krijgen bij de data)
3. Features maken
4. Cleaning/scaling/etc.
5. Modelleren en evalueren

Challenge

Maak een model dat zo goed mogelijk kan voorspellen hoe gezond de rollagers zijn.

Degene met de hoogste score wint. Welke score? Blijft geheim!

Mail naar simchavanhelvoort@tauomega.nl:

- Notebook in het format op dia 32
- Pickle bestand van het model

Maak een presentatie met de stappen die je hebt doorlopen en wat je uiteindelijk hebt gemaakt.

Als er tijd over is

Packages inladen

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from bearing_dataset import bearing_dataset
# voeg hier extra packages toe
```

Dataset inladen

```
In [ ]: df = #...
```

Preprocessing

```
In [ ]: # Doe hier al je preprocessing stappen uitvoeren
```

```
In [ ]: # Wat wordt je voorspellende variabele en wat worden de afhankelijke variabele
X = #...
Y = #...
```

Model inladen

```
In [ ]: # laad hier je pickle model in
import pickle
filename = "file.sav"
with open(filename, "rb") as f:
    model = pickle.load(f)
```

Voorspelling maken

```
In [ ]: ypred = model.predict(X)
```


Model opslaan en inladen

```
import pickle
s = pickle.dumps(clf)
with open("decision_tree.sav", "wb") as f:
    f.write(s)
```

```
import pickle
with open("decision_tree.sav", "rb") as f:
    clf = pickle.load(f)
```

EVALUATIE VIA EVALYTICS

<https://app.evalytics.nl/#/login>

cen-307

Nog een paar vraagjes

- Wat vonden jullie van de verhouding tussen theorie en praktijk?
- Vonden jullie de cursus te lang/te kort?
- Wat vonden jullie van het tempo? Ging ik vaak te snel? Of te langzaam?
- Hoeveel tijd waren jullie aan huiswerk kwijt? Was dat (te) veel?

**BEWIJS VAN DEELNAMEN
NIET VERGETEN!**

- Covid-antigeentest tabel:
https://www.cdc.gov/mmwr/volumes/69/wr/mm695152a3.htm#T2_down
- Bekijk de uitwerkingen van mensen op Kaggle
- Kaggle dataset hartfalen:
<https://www.kaggle.com/andrewmvd/heart-failure-clinical-data>
- Kaggle dataset autosales:
<https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes>
- Iris dataset:
https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html
- Kaggle dataset paddenstoelen (giftig/niet giftig):
<https://www.kaggle.com/uciml/mushroom-classification>
- Meer kaggle datasets om te oefenen:
<https://www.kaggle.com/datasets?datasetsOnly=true>