

- Log Reg.
- SVM
- Naive Bayes (NLP)
- D.T. ENSEMBLES



Machine learning

Logistic Regression

CLASSIFICATION

Wouter Gevaert & Marie Dewitte

Inhoud

Introductie

Logistic regression

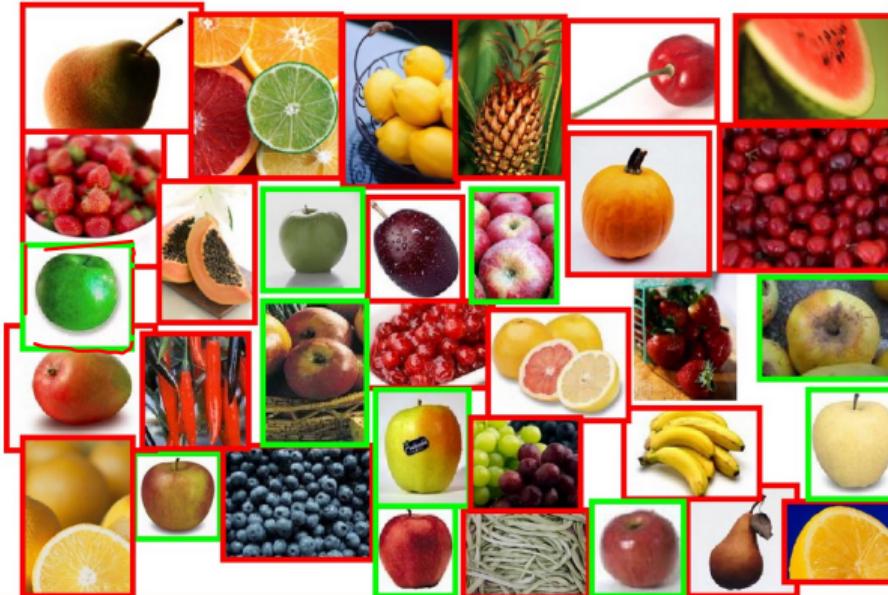
Evaluatie van een classifier

Introductie

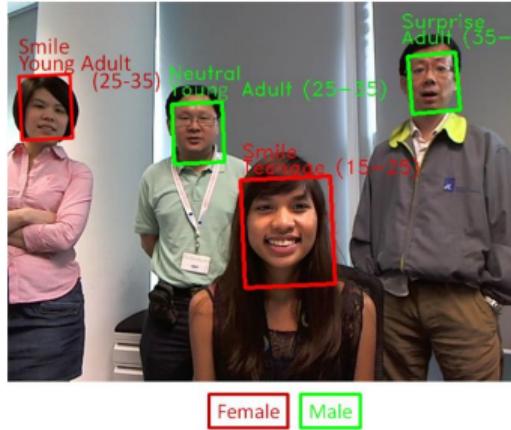
Wat is classificatie?



Classificatie is een supervised learning techniek waarbij een getraind model niet geziene inputs toewijst aan één of meerdere gelabelde categorieën (classes)



- Gezichtsherkenning
- Nummerplaatherkenning
- Spam detectie
- Medische diagnoses
- Voorspelling of een klant op een advertentie zal klikken
- Kwaliteitscontrole
- ...

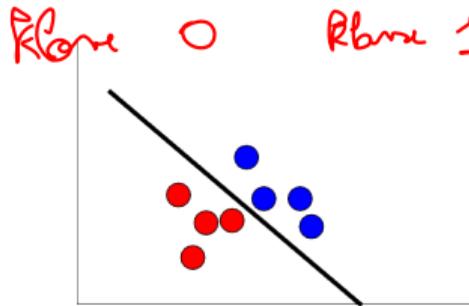


Types van classifiers

Binairy (binomial) classifier

Verdeel de samples in **twee verschillende klassen**

Voorbeeld: bepaal of een kanker goedaardig of kwaadaardig is

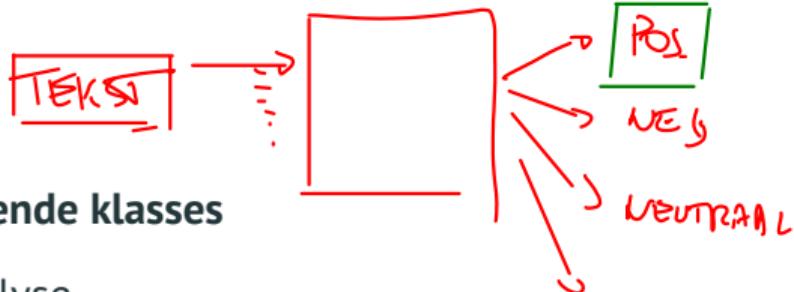


Types van classifiers

Multiclass classifier

Verdeel de samples in **drie of meerdere verschillende klassen**

Voorbeelden: gezichtsherkenning, sentiment analyse



SENTIMENT ANALYSIS

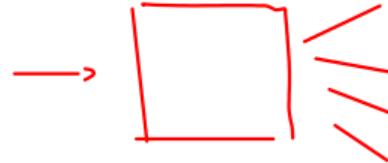


Discovering people opinions, emotions and feelings about
a product or service

Types van classifiers

Multilabel classifier

MULTINOMIALE



Er kunnen meerdere labels aan een sample toegewezen worden. Een sample kan tot meerdere klassen behoren

Voorbeelden: image content analysis, een film kan tot meerdere genres behoren.



Huis	Boom	Strand	Wolken	Bergen	Dieren
Ja	Ja	Nee	Ja	Nee	Nee

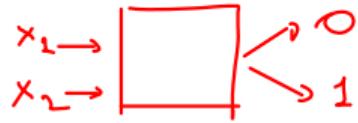
Voorbeeld

TRAIN

	x_1 vorm	x_2 kleur	y
	rondheid	groenheid	appel
0	9	8	1
1	10	7	1
2	2	3	0
3	1	2	0
4	5	8	1
5	7	7	1
6	6	3	0
7	3	3	0
8	9	5	1
9	9	3	1
10	4	5	0
11	6	1	0
12	5	7	1
13	8	8	1
14	2	7	0

1 = appel

0 = geen appel



Voorspel of een stuk fruit een appel is op basis van vorm en kleur

features: rondheid en groenheid

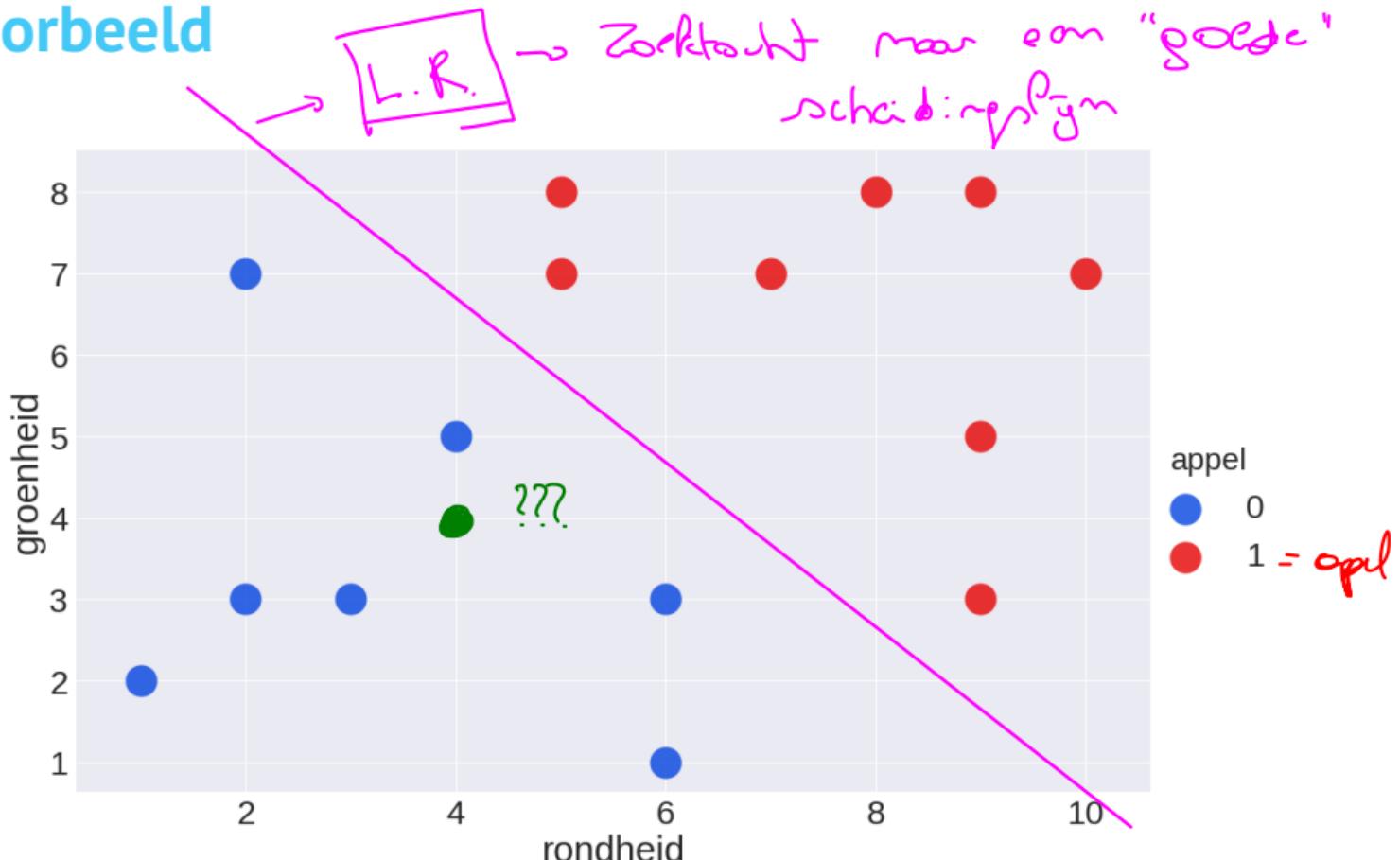
target: appel: ja/nee

trainingset met 15 training examples

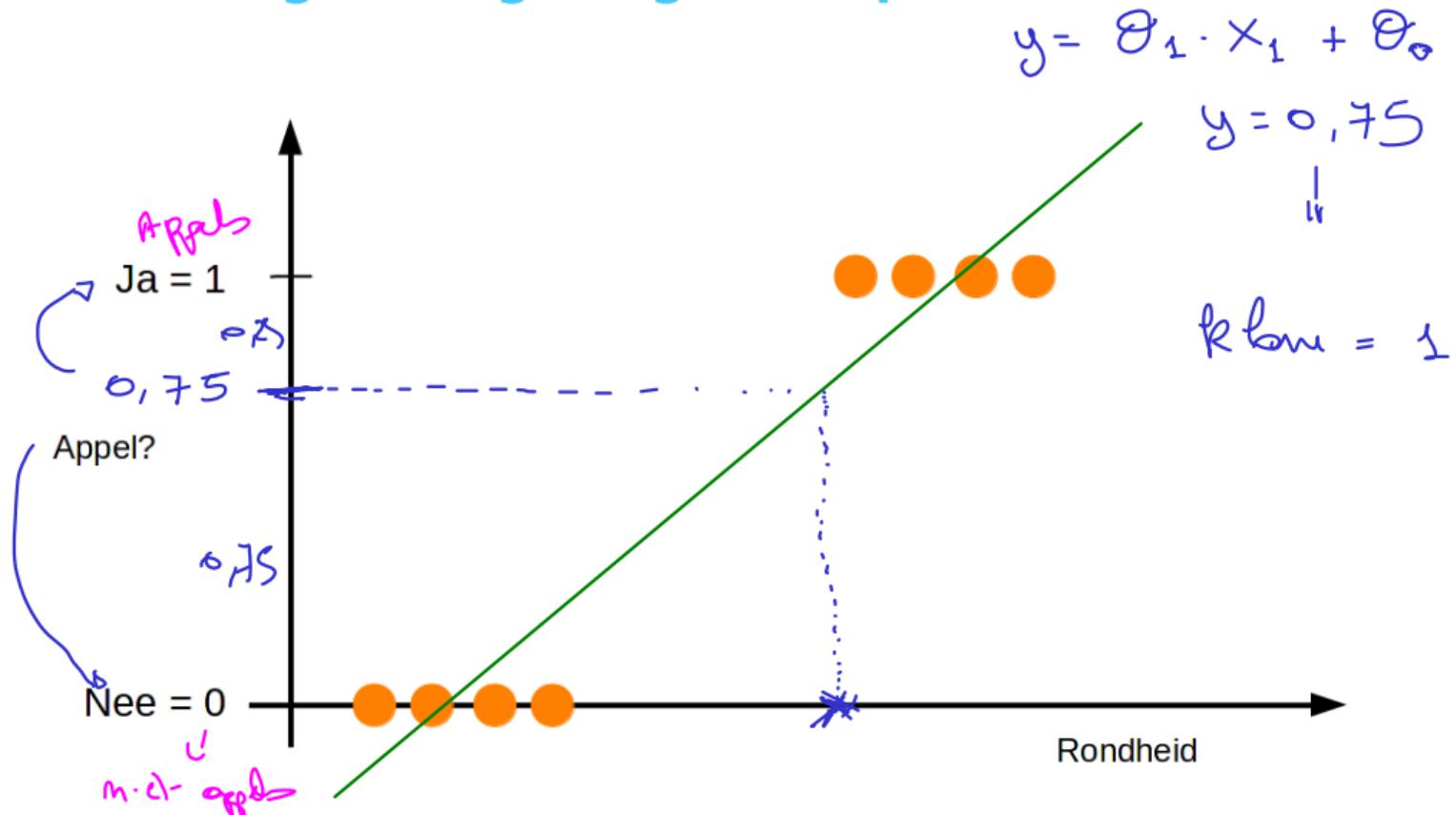
Classificatie

Bij classificatie is de output/target een (discrete) variabele / klasse

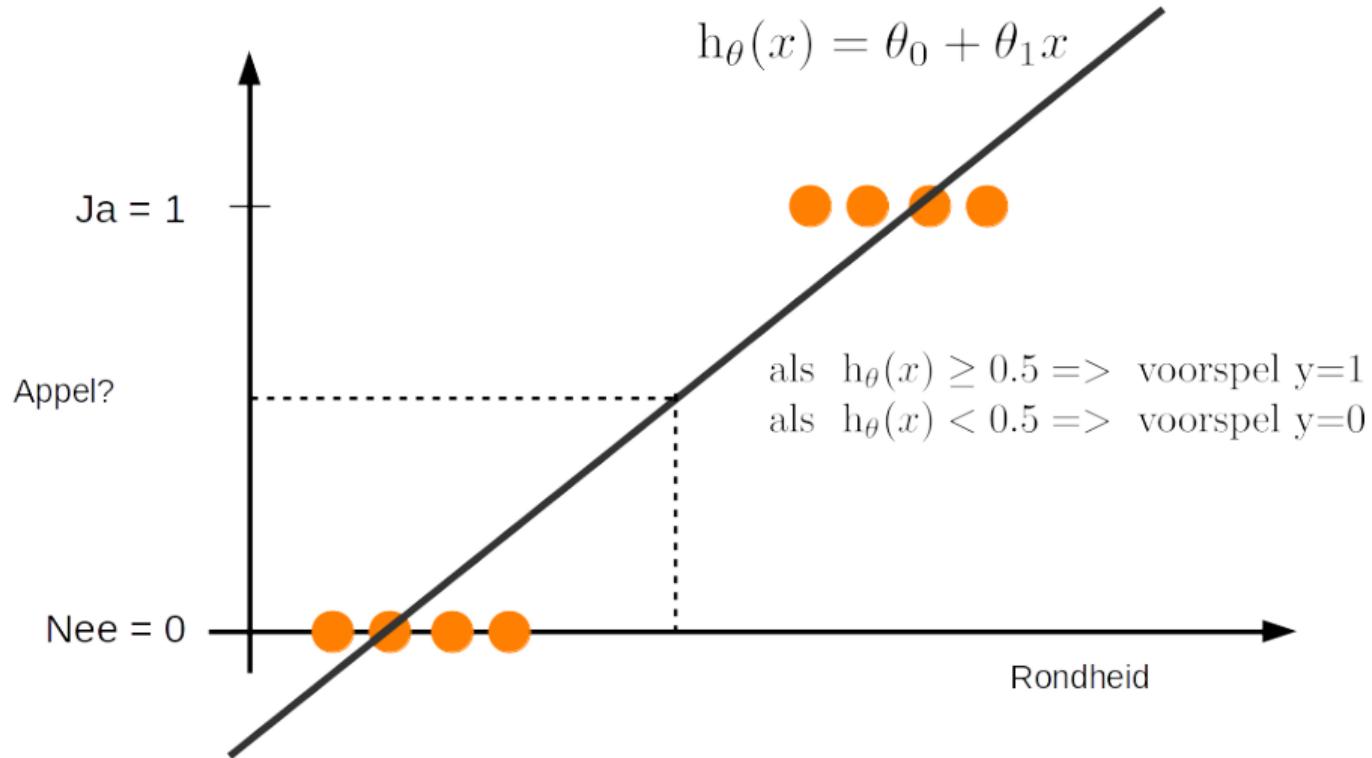
Voorbeeld



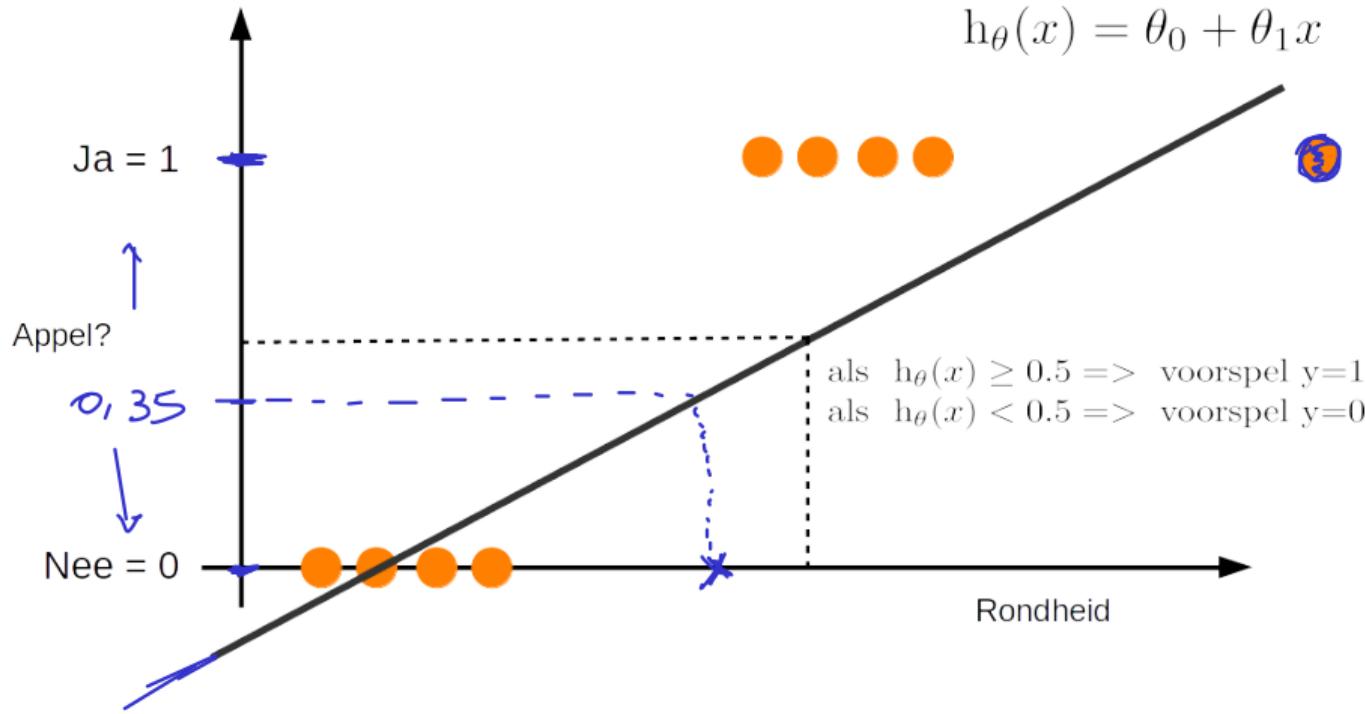
Waarom regressie geen goede optie is



Waarom regressie geen goede optie is



Waarom regressie geen goede optie is

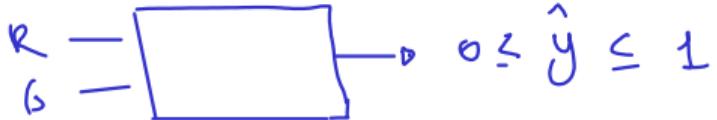


Sigmoid



Logistic regression

Logistic regression



Het model

We willen dat het model $h_\theta(x)$ voldoet aan

$$0 \leq h_\theta(x) \leq 1$$

1

0

- $h_\theta(x)$ = de geschatte kans dat $y=1$ bij input x
- Voorbeeld: $h_\theta(x) = 0.80$
Het model is voor 80% zeker dat het om een appel gaat

$$\hat{y} = 0,80 \rightarrow \begin{matrix} \text{klaar} \\ 1 \end{matrix}$$

$$\hat{y} = 0,15 \rightarrow \begin{matrix} \text{klaar} \\ 0 \end{matrix}$$

Logistic regression

Het model

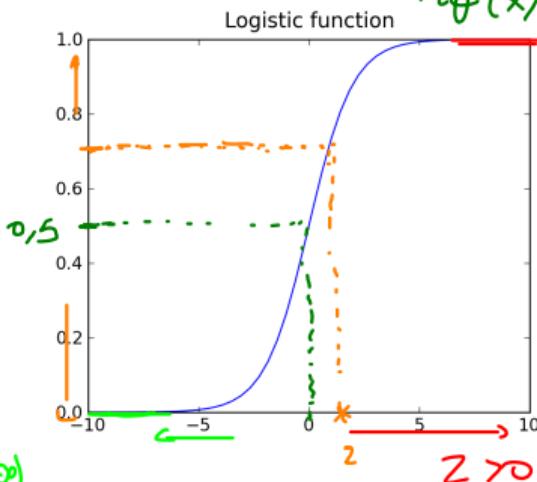
→ SIGMOID

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$y = \left(\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5 \right)$$

$$h_{\theta}(x) = 0,7$$

$$h_{\theta}(x) = 0,5$$



$$\frac{1}{e} \cdot 0$$

$$z = 0 \rightarrow h_{\theta}(x) = 0,5 \quad \frac{1}{1+e^{-z}}$$

$$z = +\infty \rightarrow 1 \quad \frac{1}{1+e^{-\infty}} = 0 \quad \frac{1}{1+e^{(-\infty)}} = 1$$

$$y = 1 \text{ als } h_{\theta}(x) \geq 0,5 \Rightarrow \theta^T x \geq 0$$

$$y = 0 \text{ als } h_{\theta}(x) < 0,5 \Rightarrow \theta^T x < 0$$

e = getal van evolutie
2,7

Logistic regression

$$R = 8 \quad 6 = 6 \quad ?$$

Het model - interpretatie via voorbeeld appels

Het model is van de vorm: $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

met x_1 = rondheid en x_2 = groenheid

Veronderstel na training: $\underline{\theta_0} = -40$, $\underline{\theta_1} = 4$, $\underline{\theta_2} = 4$

$$4 \cdot 8 + 4 \cdot 6 - 40 \\ = + \boxed{16}$$

$$\frac{1}{1 + e^{-16}} = 0,9999 \dots$$

Voorspel $y = 1$ als $-40 + 4x_1 + 4x_2 \geq 0$

Voorspel $y = 0$ als $-40 + 4x_1 + 4x_2 < 0$

Gegeven een rondheid van 8 en een groenheid van 6:

$$-40 + 4 \times 8 + 4 \times 6 = 16 \Rightarrow \text{Appel}$$

$$h_\theta(x) = \frac{1}{1 + e^{-16}} = 0.999999887$$

→ 99,9999887

Logistic regression

Het model - interpretatie via voorbeeld appels

Gegeven een rondheid van 5 en een groenheid van 4.5:

$$-40 + 4 \times 5 + 4 \times 4.5 = -2 \Rightarrow \text{Geen appel}$$

$$h_{\theta}(x) = \frac{1}{1 + e^{+2}} = 0.12$$

$\frac{1}{1 + e^{+2}}$
Rond e 0

- Het model is maar voor 12% zeker dat het om een appel gaat
- Met 88% zekerheid gaat het volgens het model niet om een appel

Logistic regression

Het model - grafische interpretatie via voorbeeld appels

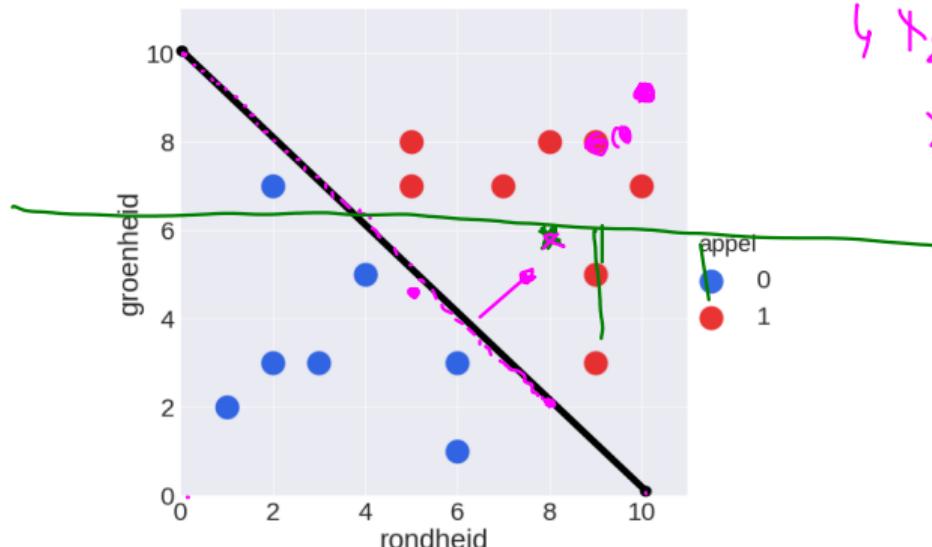
Op de scheidingslijn is: $\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$

In het voorbeeld: $-40 + 4x_1 + 4x_2 = 0$

$$-40 + 0 + 4x_2 = 0$$

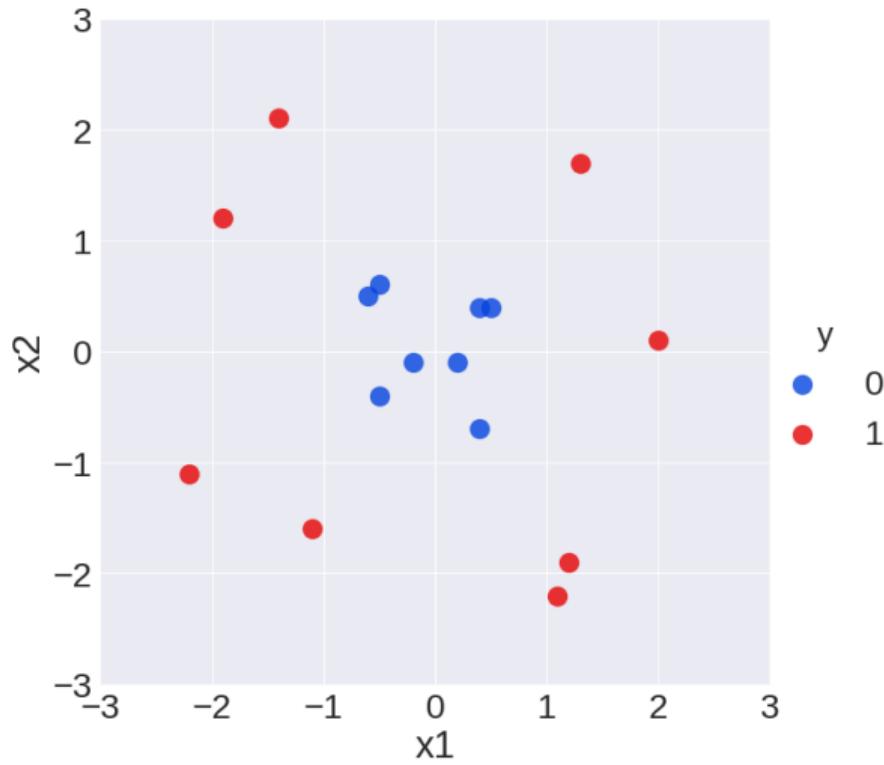
$$4x_2 = 40$$

$$x_2 = \frac{40}{4} = 10$$



Logistic regression

Het model - grafische interpretatie - niet lineair



Logistic regression

$$\theta_0 + \theta_1 x_1^2 + \theta_2 x_2^2 = r^2$$

θ_0
 θ_1
 θ_2

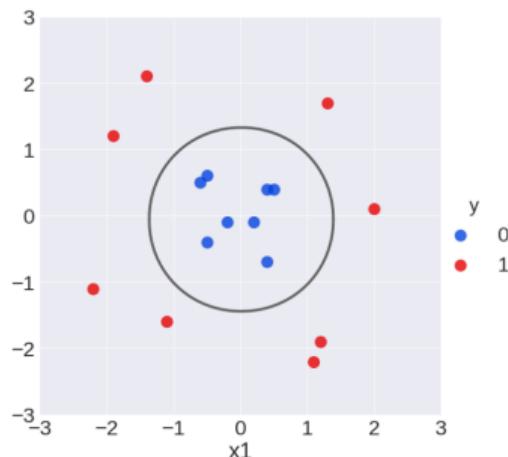
Het model - grafische interpretatie - niet lineair

Extra features: $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$

Veronderstel: $\theta_0 = -2, \theta_1 = 0, \theta_2 = 0, \theta_3 = 1, \theta_4 = 1$

Voorspel $y = 1$ als $-2 + x_1^2 + x_2^2 \geq 0$

$x_1^2 + x_2^2 \geq 2$ (vergelijking van een cirkel met straal $\sqrt{2}$)



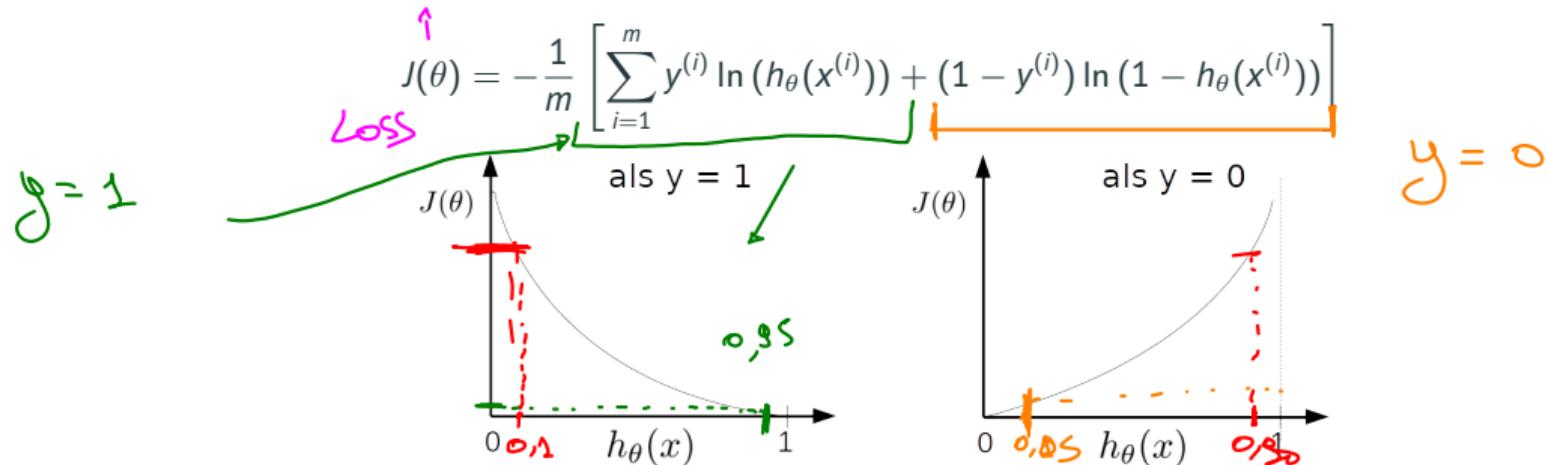
Logistic regression

Het model

De kostenfunctie wordt:

Effde

$$J(\theta) = \begin{cases} -\ln(h_\theta(x)) & \text{als } y = 1 \\ -\ln(1 - h_\theta(x)) & \text{als } y = 0 \end{cases}$$



Zoek de waarden voor θ die de kostenfunctie $J(\theta)$ minimaliseert via Gradiënt Descent (GDS).

Logistic regression met Sklearn

Preprocessing van de data

Analoog aan preprocessing bij lineaire regressie:

- Data inlezen
- Check op inconsistenties
- Check uitschieters
- Plot de data
- Splits op in features en targets
- Verdeel in een training en test set

Logistic regression met Sklearn

Preprocessing van de data

```
# Importeer de dataset
dataset = pd.read_csv('appels.csv')
features = list(dataset.columns[:dataset.columns.size -1])
X = dataset[features].values
y= dataset['appel'].values
sns.set(font_scale = 2) # lettergrootte van de axis labels
colors = ["blue", "red", "greyish", "faded_green",
          "dusty_purple"]
sns.lmplot(x='rondheid',y='groenheid',data=dataset,fit_reg=False,
            hue='appel',palette =sns.xkcd_palette(colors),
            scatter_kws={'s':500}, size=7, aspect=1.5)
```

Logistic regression met Sklearn

Trainen van het logistic regression model

```
# Train een logistic regression classifier

logreg = linear_model.LogisticRegression(C=1e5)
# C= Inverse of regularization strength;
# must be a positive float. Like in support vector machines,
# smaller values specify stronger regularization.
logreg.fit(X, y)

print('coëfficiënten:', logreg.coef_)
print('intercept:', logreg.intercept_)
```

coëfficiënten: $\theta_1 = 4.28747762$ en $\theta_2 = 4.06244327$

intercept: $\theta_0 = -43.94116677$

Logistic regression met Sklearn

Classificeren van een nieuwe sample

```
#voorspel de klasse van een rondheid=8 en een groenheid van 6  
  
print(logreg.predict(np.array([8,6]).reshape(1,-1)))  
  
print('kans_op_een_appel/geen_appel',  
logreg.predict_proba(np.array([8,6]).reshape(1,-1)))  
  
#voorspel de klasse van een rondheid=4 en een groenheid van 4  
  
print(logreg.predict(np.array([4,4]).reshape(1,-1)))  
print('kans_op_een_appel/geen_appel',  
logreg.predict_proba(np.array([4,4]).reshape(1,-1)))
```

[1] kans op een appel/geen appel 3.99395302e-07 | 9.99999601e-01

[0] kans op een appel/geen appel 9.99973583e-01 | 2.64168196e-05

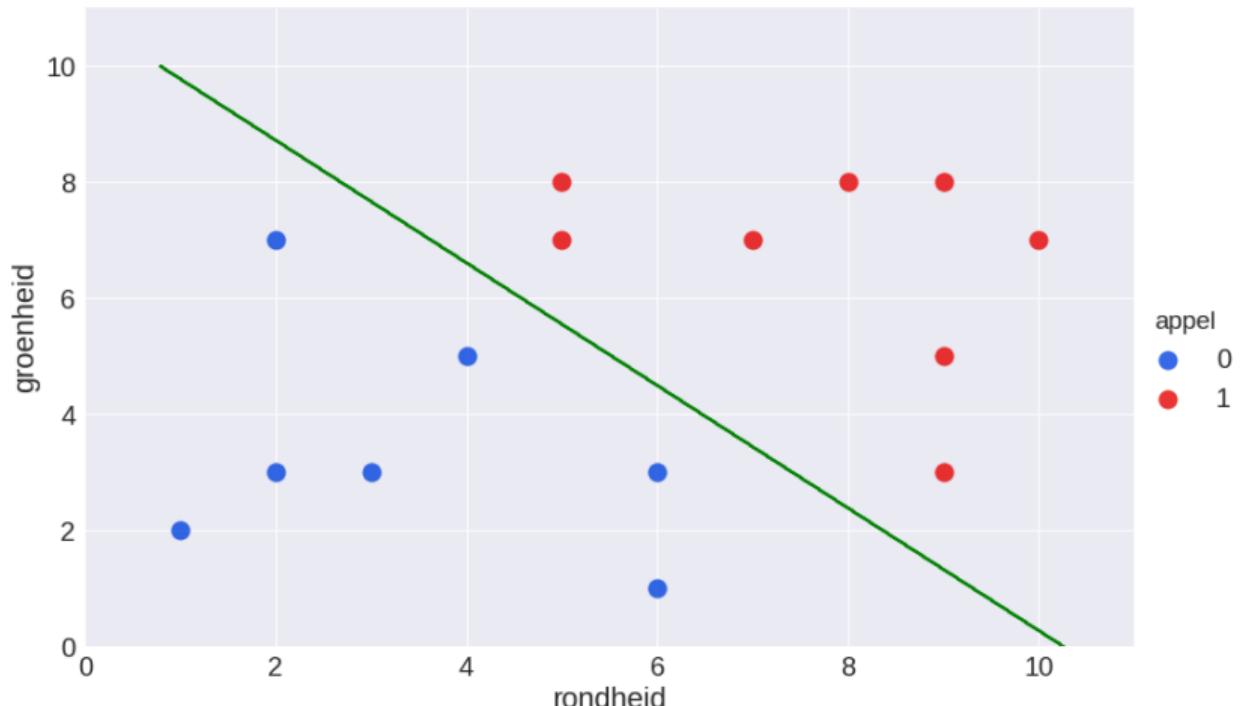
Logistic regression met Sklearn

Visualiseer de decision boundary

```
h = 0.01
rond_min = X[:,0].min()-2
rond_max = X[:,0].max()+2
groen_min = X[:,1].min()-2
groen_max = X[:,1].max()+2
xx, yy = np.meshgrid(np.arange(rond_min, rond_max, h),
np.arange(groen_min, groen_max, h))
Z = logreg.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
sns.set(font_scale = 2)
colors = ["blue", "red"]
sns.lmplot(x='rondheid',y='groenheid',data=dataset,
fit_reg=False,hue='appel',palette =sns.xkcd_palette(colors),
scatter_kws={'s':200}, size=8, aspect=1.5)
sns.plt.ylim(0, 11)
sns.plt.xlim(0, 11)
plt.contour(xx, yy, Z, colors='green')
```

Logistic regression met Sklearn

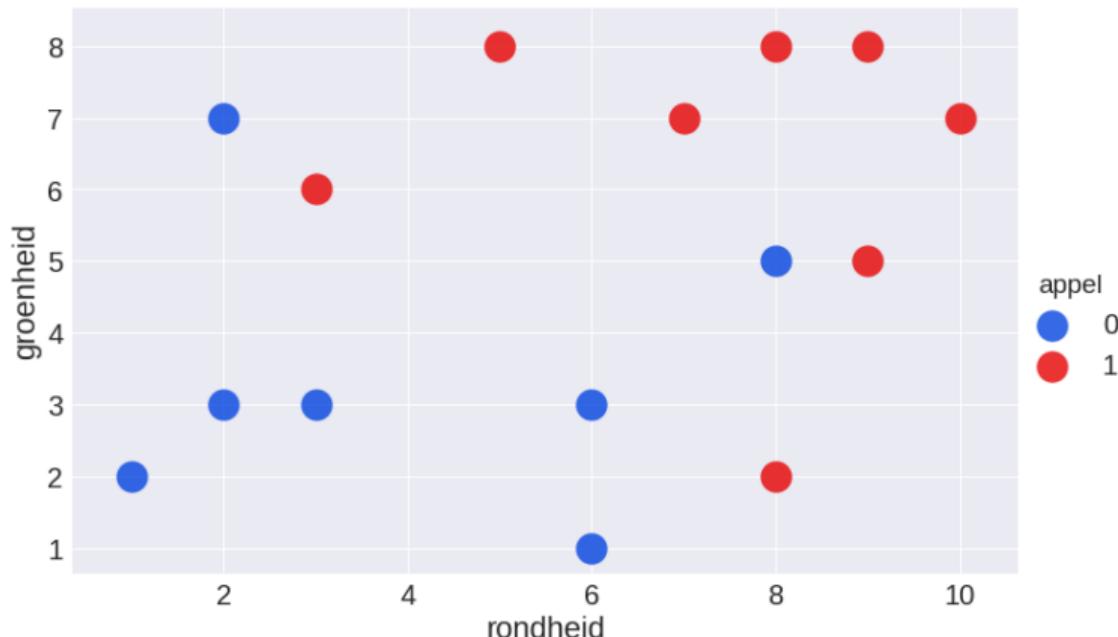
Visualiseer de decision boundary



Logistic regression met Sklearn

Feature engineering

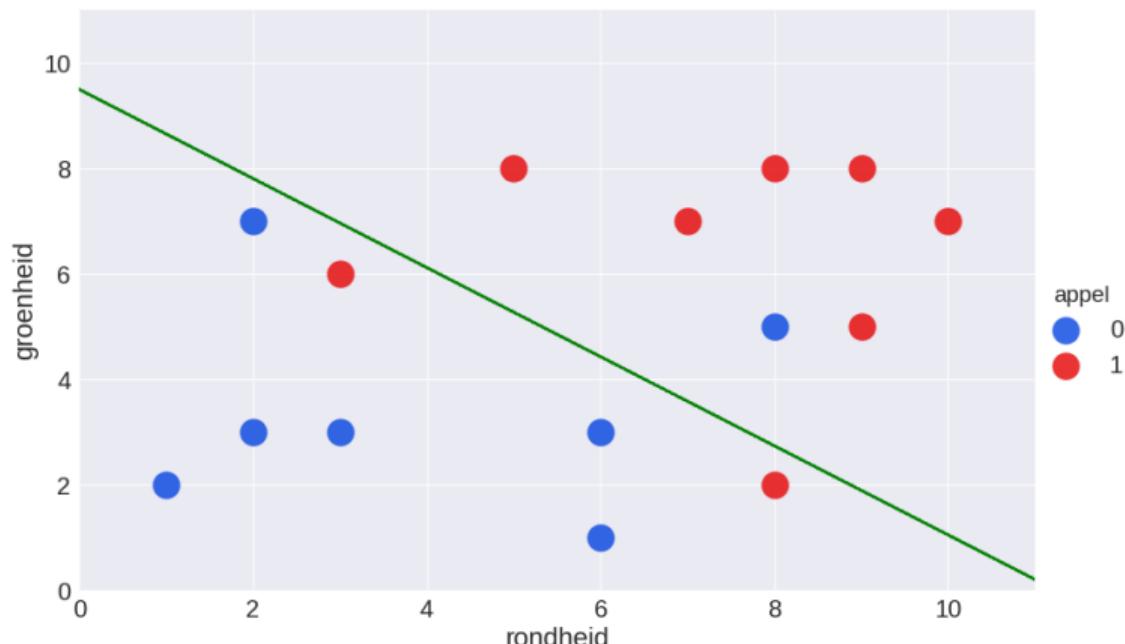
Niet lineair scheidbare dataset



Logistic regression met Sklearn

Feature engineering

Niet lineair scheidbare dataset



Logistic regression met Sklearn

Feature engineering

Toevoegen van hogere orde features:

```
# Aanmaken van de hogere orde features
graad = 3

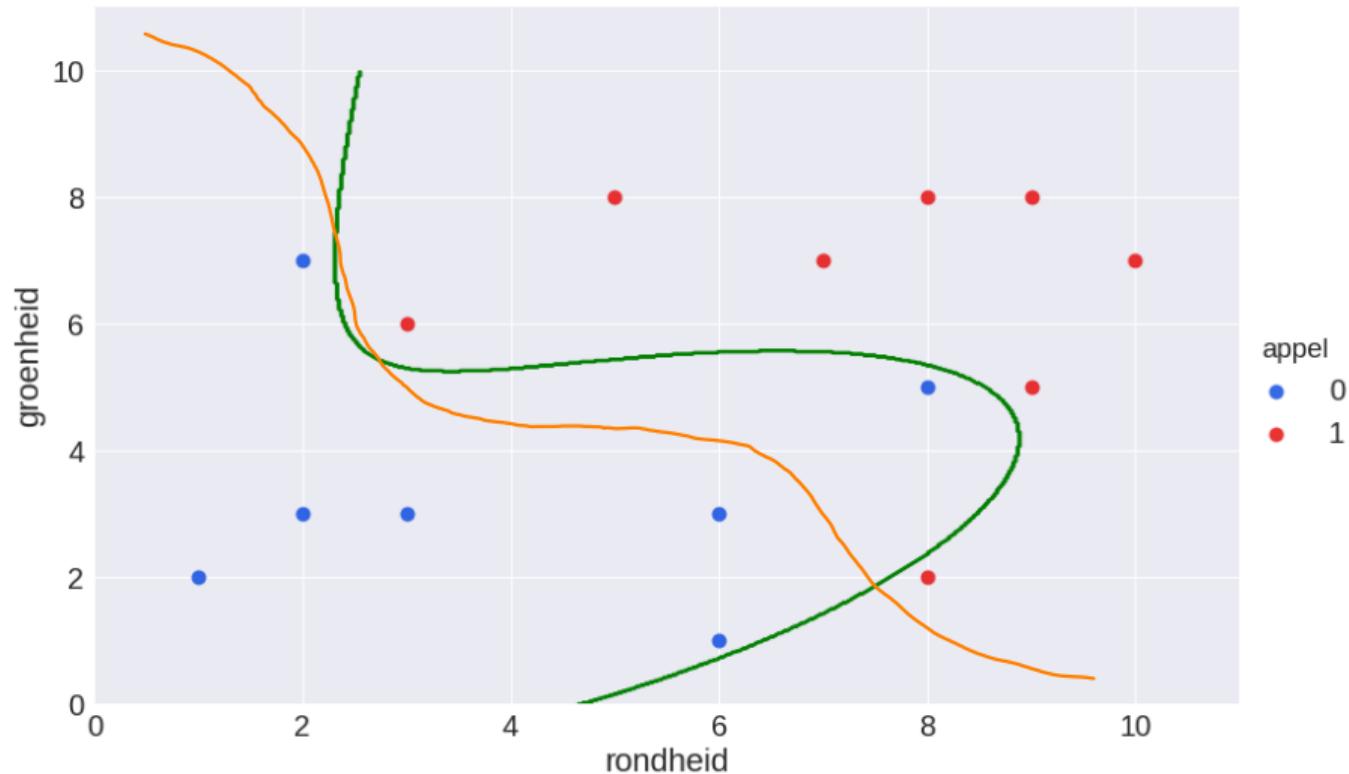
poly = PolynomialFeatures(graad)
Xp = poly.fit_transform(X)

# Train model op hogere orde features en visualiseer de decision boundary

logreg_poly = linear_model.LogisticRegression(C=1)
logreg_poly.fit(Xp, y)
```

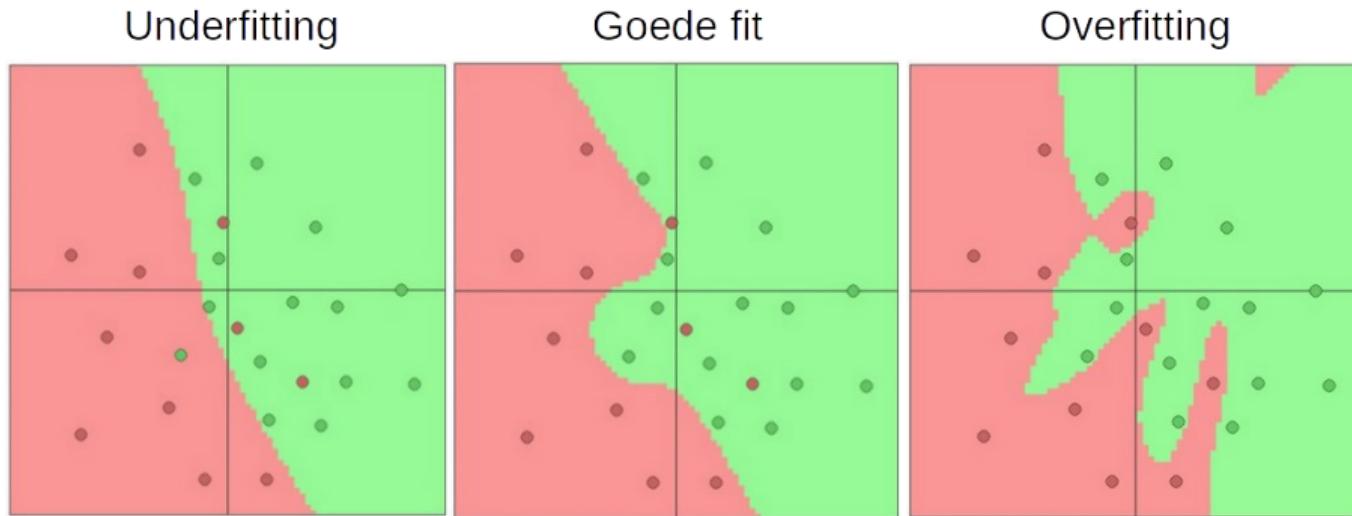
Logistic regression met Sklearn

Feature engineering - voorbeeld appels met derde-orde features



Logistic regression met SKlearn

Underfitting en overfitting



Regelen tussen underfitting en overfitting via regularisatie

Logistic regression met SKlearn

Regression = $\sum \alpha_j x_j$

Underfitting en overfitting

Via regularisatie een goed evenwicht zoeken tussen underfitting en overfitting

In Scikit Learn linear_model.LogisticRegression

- C = inverse regularisatie sterkte
 - kleine waarden voor C zorgen voor een sterke regularisatie (underfitting)
 - grote waarden voor C zorgen voor een zwakke regularisatie (overfitting)

Confid.: $= C$

```
logreg = linear_model.LogisticRegression(C=100)  
logreg.fit(Xf, y)
```

~~α_j~~ = $\sum \alpha_j x_j$

Multi-class classification

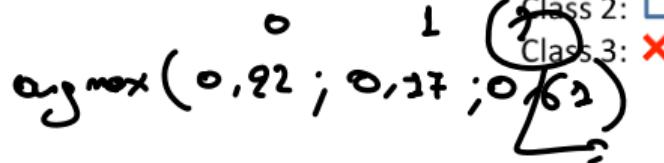
One-vs-All

One-vs-Rest

$$P(x = \text{Peren}) = 0,22$$

$$P(x = \text{banaan}) = 0,37$$

$$P(x = \text{appel}) = 0,61$$

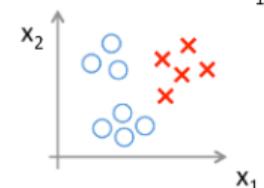
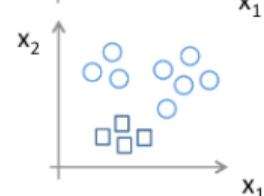
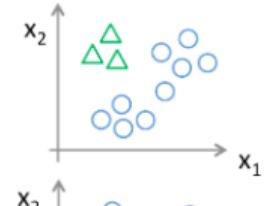


$$P(x = 1)$$



72% → 1

28% → 0



Je hebt 1 classifier per klasse. Totaal aantal classifiers: N

Gevoeliger voor niet-gebalanceerde data

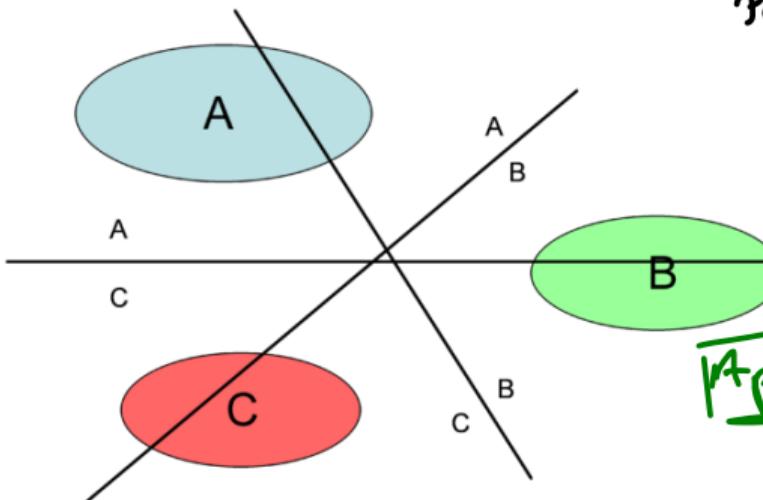
Multi-class classification

One-vs-One

→ SVM (Kernels)

$\times \rightarrow ?$

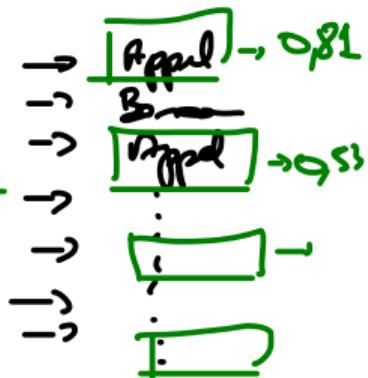
$N = 5$



Je hebt $\frac{N(N - 1)}{2}$ classifiers (met N aantal klasses) \Rightarrow Rekenintensief

Minder gevoelig voor niet-gebalanceerde data

Appels \leftrightarrow Peren
Appels \leftrightarrow Bananen ..
Peren \leftrightarrow Bananen ..

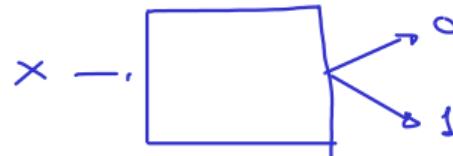
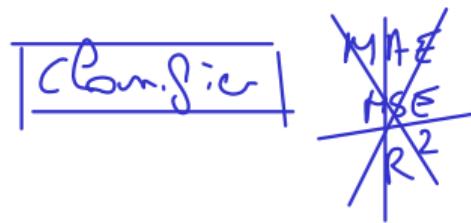


$$\frac{5 \cdot (5 - 1)}{2} = \frac{20}{2} = 10$$

Regressie

- MAE
- MSE
- R^2

$$\begin{array}{l} \hat{y} \\ 18 \\ y_{true} \\ 20 \\ \vdots \end{array} \rightarrow \text{Lo}$$



\hat{y}	y_{true}
0	1
1	0

Evaluatie van een classifier

—
0, 62

Evaluatie van een classifier

→ Accuracy

Positive → klasse 1

Negative → klasse 0

WERKELIJKE
WAARDE

		p' (Predicted)	n' (Predicted)
		P (Actual)	n (Actual)
		True Positive <i>3</i>	False Negative <i>1</i>
P (Actual)	False Positive <i>3</i>	True Negative <i>3</i>	

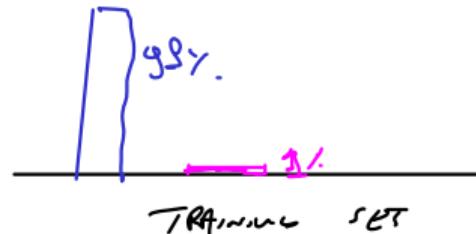
$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

voorspellings \hat{y}	Actual y_{act}
1	1
1	1
0	1
0	0
0	1
1	1
1	1
1	0
0	1
0	0
0	0

$$\frac{6}{10} = 0,6 \quad 60\%$$

99 %.

1 %.



Malignant \rightarrow 1 Pmt.

Benign \rightarrow 0 Neg.

x_{text}



		1	0
1	1	0	10
	0	990	10
0			

$$\text{Acc} = \frac{990}{1000} = 99\%$$

$$\text{Recall} = \frac{0}{0+10} = \frac{0}{10} = 0 = 0\%.$$

$$\frac{990}{990+10} = \frac{990}{1000} = 1 \Rightarrow 100\%.$$

Evaluatie van een classifier

Recall = Sensitivity = Hit rate = True Positive Rate (TPR)

HL
St. 1st, St. 2nd

	p' (Predicted)	n' (Predicted)
p (Actual)	<u>True Positive</u>	False Negative
n (Actual)	False Positive	True Negative

✓. onvolledig

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

$$\frac{TN}{TN + FP}$$

Evaluatie van een classifier

Precision = Positive Predictive Value (PPV)

	p' (Predicted)	n' (Predicted)
P (Actual)	True Positive False Negative	
n (Actual)	False Positive	True Negative

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

Evaluatie van een classifier

→ F1 score

	p' (Predicted)	n' (Predicted)
P (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

$$F1 \text{ score} = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})}$$

Is een harmonisch gewogen gemiddelde van de recall en de precision

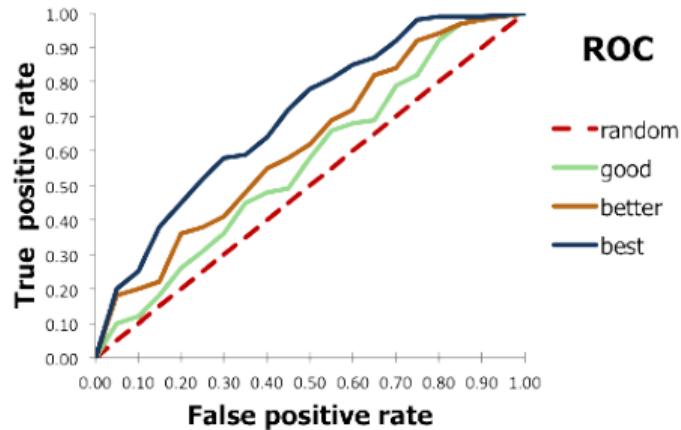
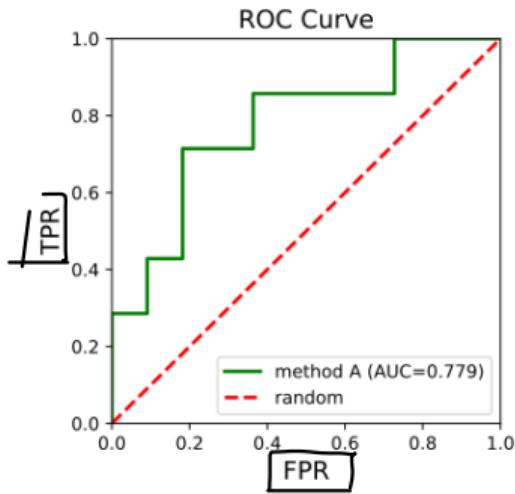
Evaluatie van een classifier

Binaire classificatie
2 klassen

Receiver Operating Characteristic (ROC)



- Gebruikt bij binaire classifiers
- Gebruikt om betere modellen te selecteren en minder goede te verwerpen



Evaluatie van een classifier

Receiver Operating Characteristic (ROC)

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

$$\text{True Positive Rate (TPR)} = \text{Sensitivity} = \boxed{\text{Recall}} = \frac{TP}{TP + FN}$$

$$\text{False Positive Rate (FPR)} = 1 - \text{specificity} = \frac{FP}{FP + TN}$$

$$\text{Specificity} = \frac{TN}{FP + TN}$$

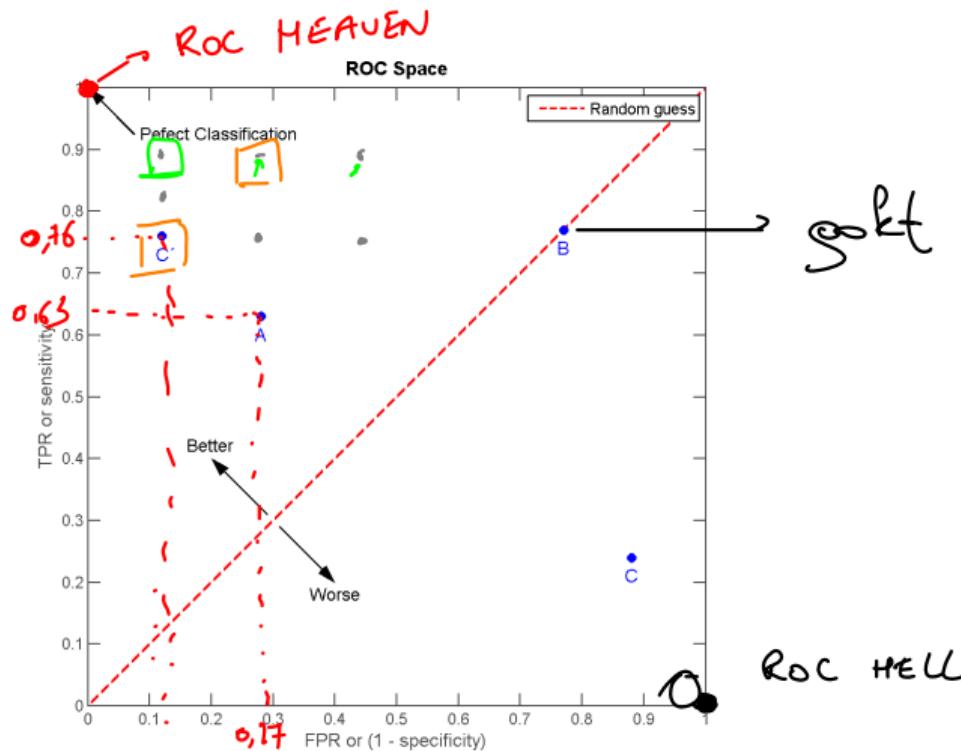
$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall

VALS ALARM

Evaluatie van een classifier

Receiver Operating Characteristic (ROC)



Evaluatie van een classifier

THRESHOLD MOVING

ROC curve en AUC (Area Under ROC Curve)

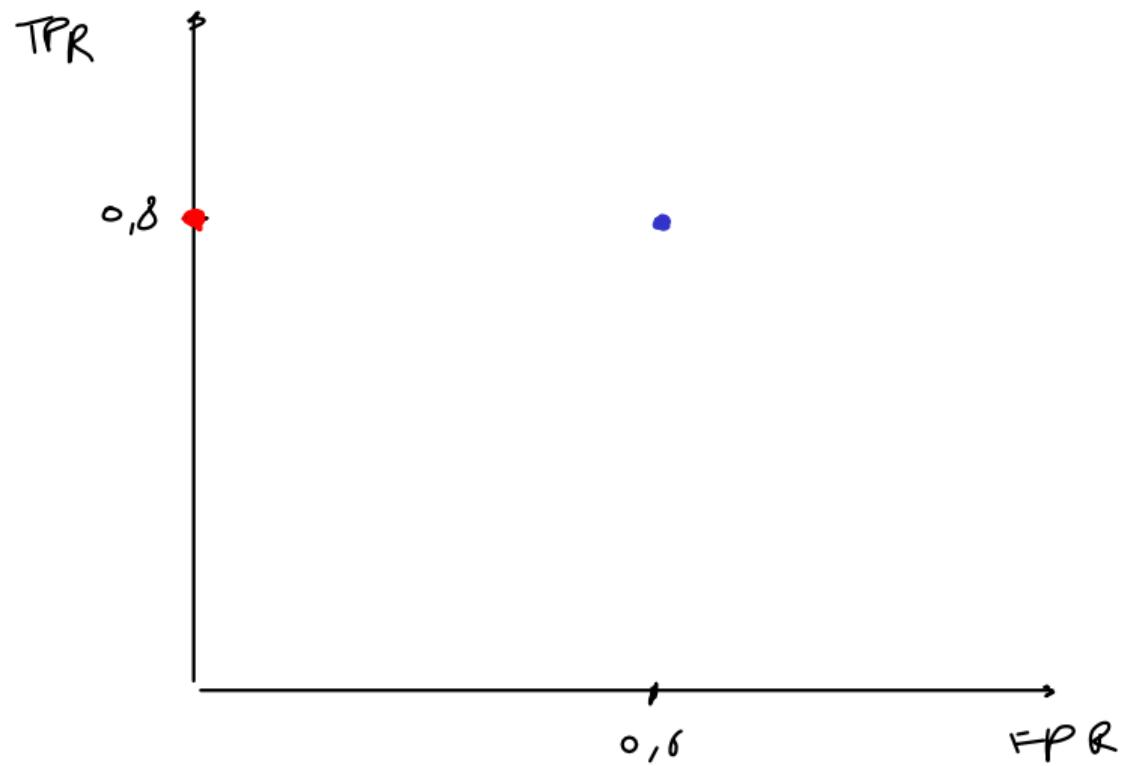
ROC bij verschillende threshold settings voor $p(y = 1|x)$ - threshold = 0.5 = 50%.

Example nr.	y_true	P(y=1 x)	→ ↗
1	1	0.93	→ 1
2	0	0.55	→ 1
3	0	0.30	→ 0
4	0	0.53	→ 1
5	1	0.81	→ 1
6	1	0.69	→ 1
7	0	0.42	→ 0
8	1	0.28	→ 0
9	1	0.96	→ 1
10	0	0.51	→ 1

	CONFUSION MATRIX	
	Predicted (y=1)	Predicted (y=0)
Actual (y=1)	4	1
Actual (y=0)	3	2

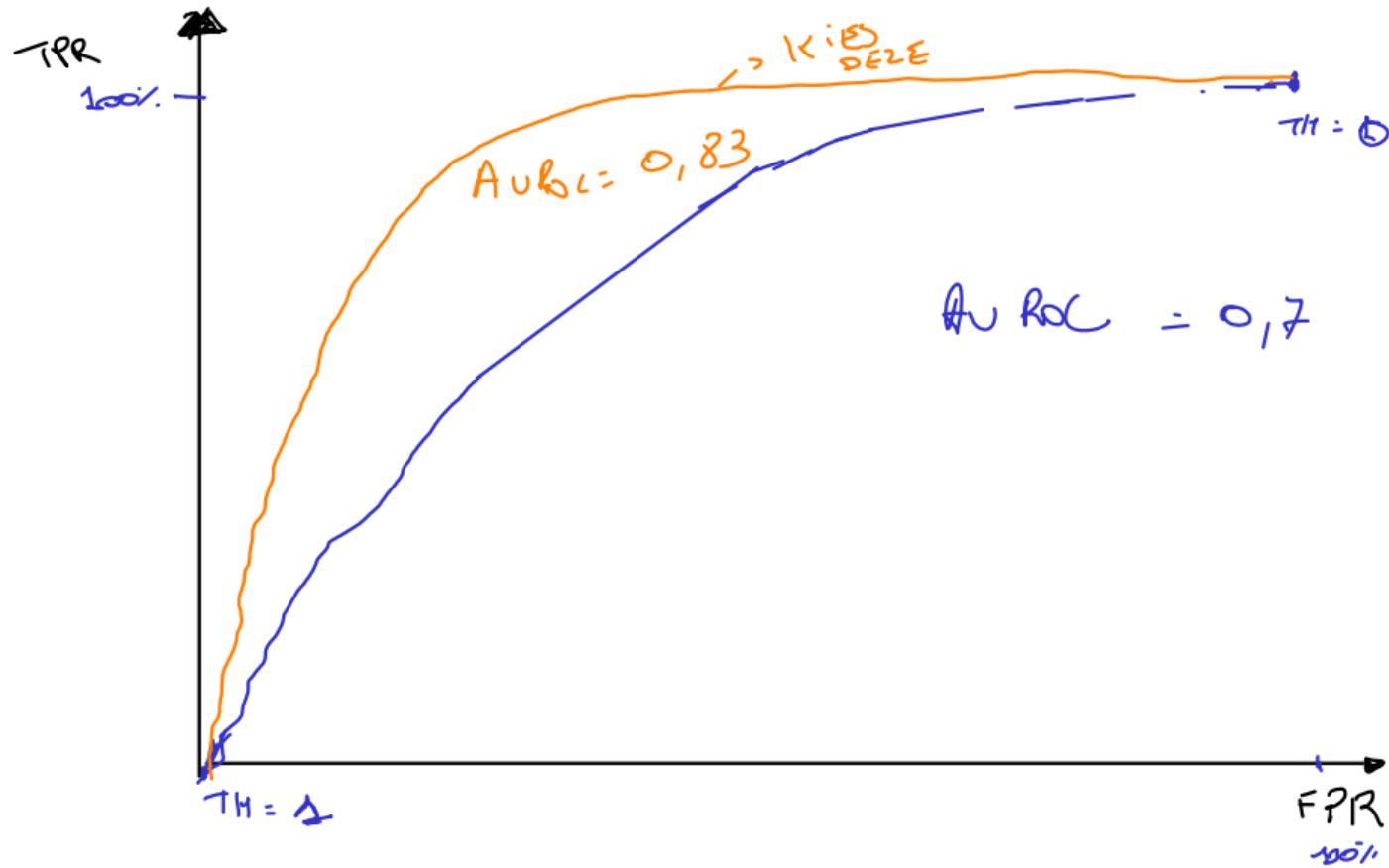
$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN} = \frac{4}{4 + 1} = 0.8$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} = \frac{3}{3 + 2} = 0.6$$



$0 \leq \text{THRESHOLD} \leq 1$

0,95



Evaluatie van een classifier

ROC curve en AUC (Area Under ROC Curve)

ROC bij verschillende threshold settings voor $p(y = 1|x)$ - threshold = 0.6

Example nr.	y_true	P(y=1 x)	→
1	1	0.93	1
2	0	0.55	0
3	0	0.30	0
4	0	0.53	0
5	1	0.81	1
6	1	0.69	1
7	0	0.42	0
8	1	0.28	0
9	1	0.96	1
10	0	0.51	0

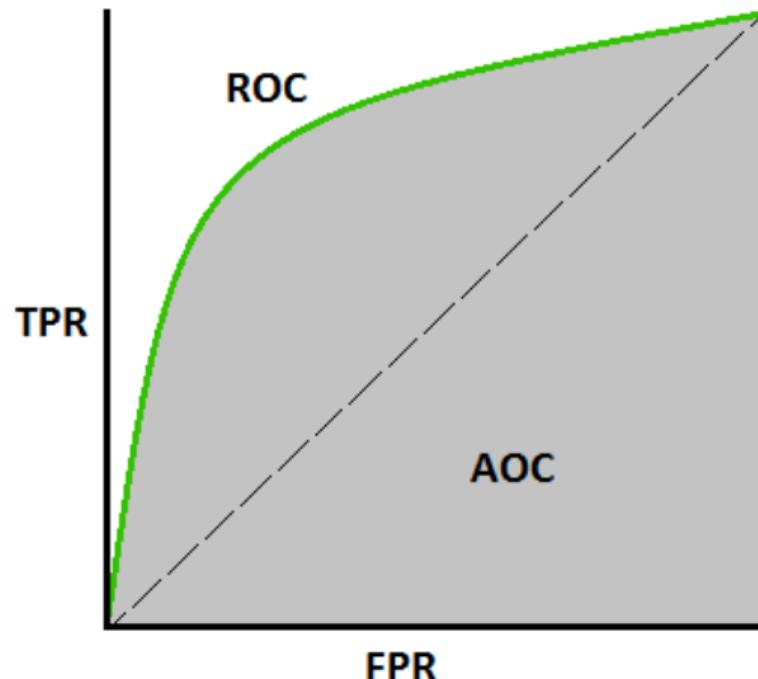
	Predicted (y=1)	Predicted (y=0)
Actual (y=1)	4	1
Actual (y=0)	0	5

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN} = \frac{4}{4 + 1} = 0.8$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} = \frac{0}{0 + 5} = 0$$

Evaluatie van een classifier

AuROC (Area under ROC)



Evaluatie van een classifier

Vuistregel

