

Data Science 2

Discriminantanalyse

Wim De Keyser

Geert De Paepe

Jan Van Overveld

Quote van de week

“I think the driving force for cultural evolution is this desire for groups to be splitting off and separating and forming subgroups insofar as the environment will allow it.”

Mark Pagel (1954-)



Agenda

1. Herhaling enkele begrippen
2. Inleiding discriminatanalyse
3. Karakteristieken
4. Discriminantanalyse met Python
5. In de praktijk



Herhaling enkele begrippen

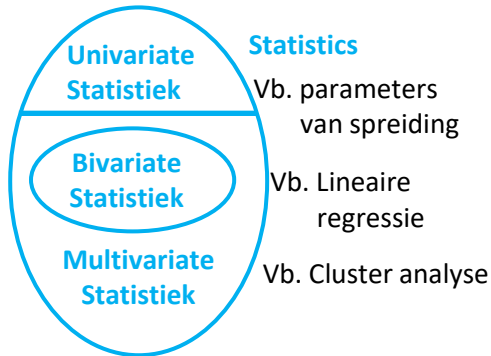
Univariate – Bivariate - Multivariate



Indeling statistische technieken kan o.a. op basis van aantal beschouwde variabelen.

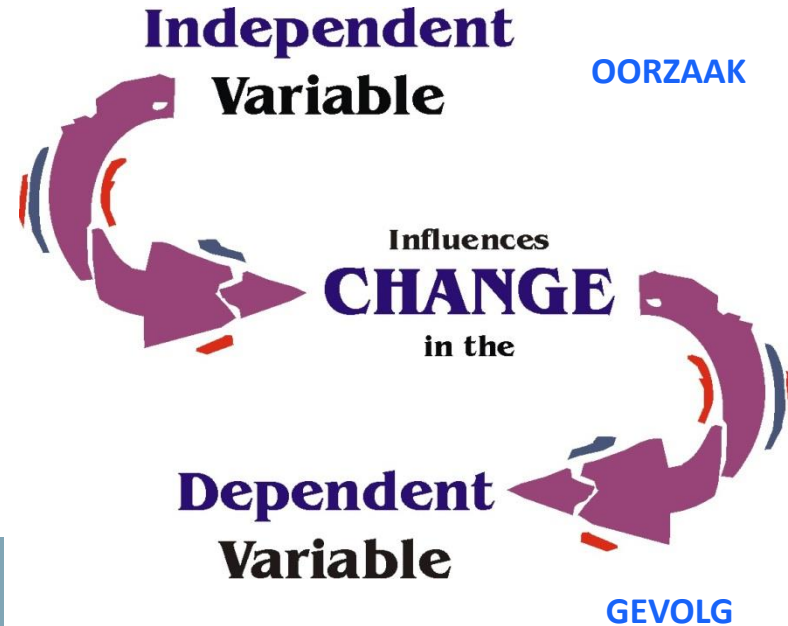
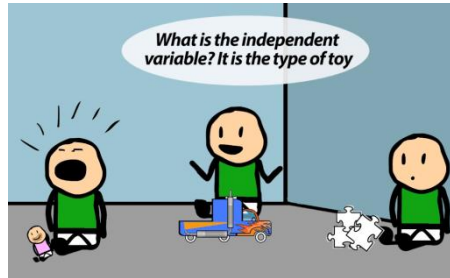
Statistische technieken toepasbaar op

- 1 variabele = **Univariate** statistiek
- 2 variabelen = **Bivariate** statistiek
- meerdere variabelen = **Multivariate** statistiek



Afhankelijke en onafhankelijke variabelen

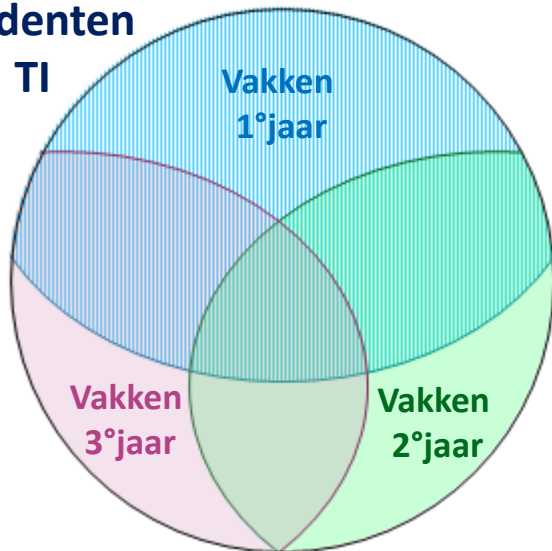
Een **afhankelijke variabele** is de variabele waarover –met behulp van een statistische techniek- een voorspelling of uitspraak wordt gedaan, terwijl een **onafhankelijke variabele** een variabele is die gebruikt wordt om voorspellingen of uitspraken op te baseren.



Wederzijds uitsluitende deelgroepen

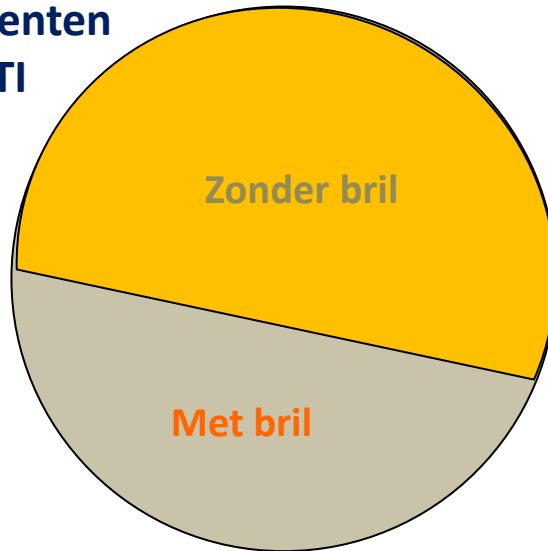
Overlappende
deelgroepen

Studenten
TI

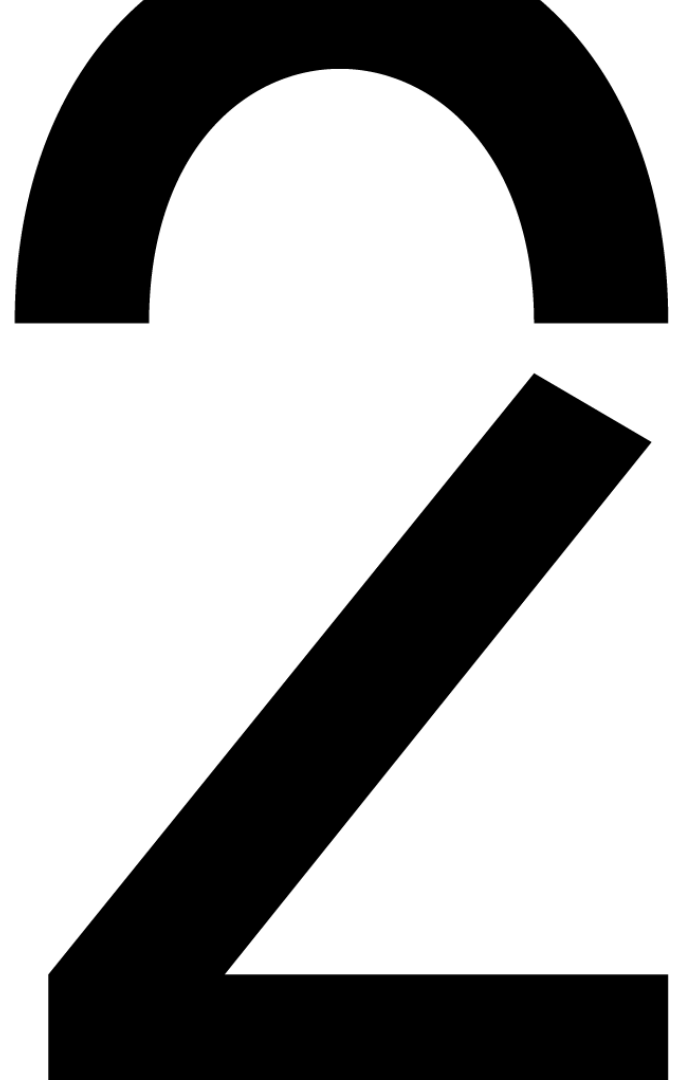


Wederzijds uitsluitende
deelgroepen

Studenten
TI

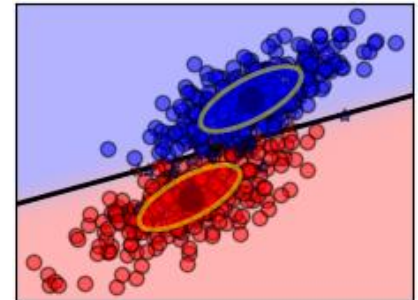


Inleiding discriminantanalyse



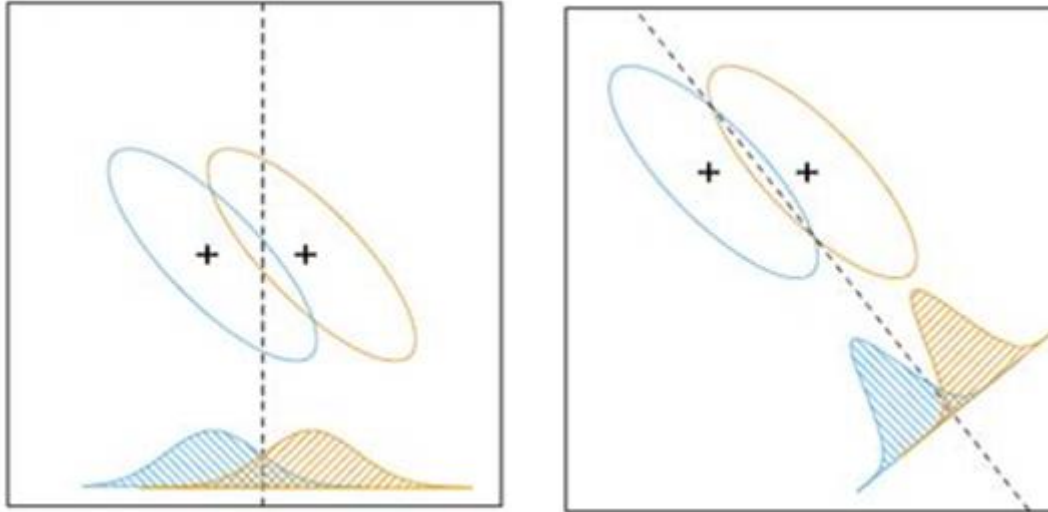
Inleiding discriminantanalyse

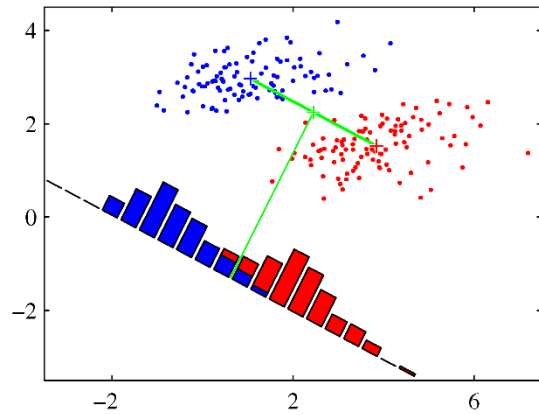
- Behoort tot de **multivariate statistiek**
- **Doel:** voor een nieuw gegeven waarneming, te bepalen tot welke van een aantal gegeven groepen van waarnemingen deze het best thuis hoort.
- De **afhankelijke variabele** is de groep. De **onafhankelijke variabelen** zijn de gegevens die gebruikt worden om tot de groep te komen



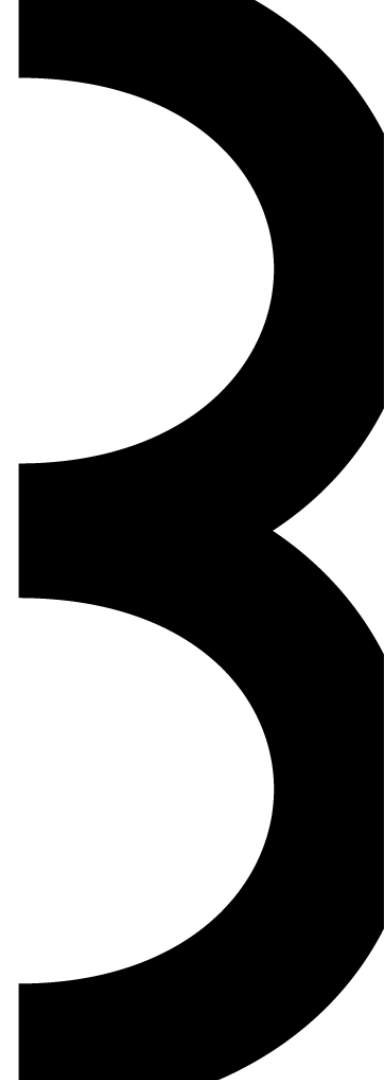
Inleiding discriminantanalyse

- **Achtergrond:** Ronald A. Fisher formulated the Linear Discriminant in 1936, and it also has some practical uses as classifier. The original *Linear discriminant* or **Fisher Linear Discriminant Analysis** was described for a 2-class problem, and it was then later generalized as “**multi-class Linear Discriminant Analysis**” or “*Multiple Discriminant Analysis*” by C. R. Rao in 1948 for his research “*The utilization of multiple measurements in problems of biological classification*”.





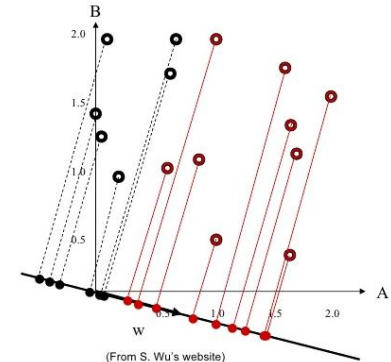
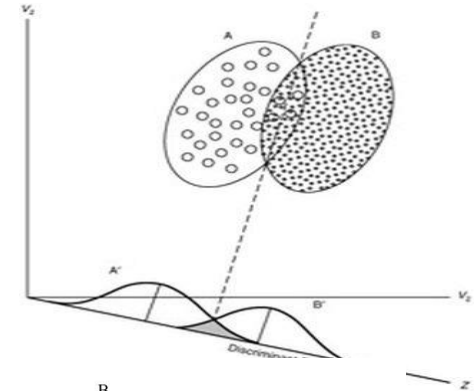
Karakteristieken



Karakteristieken Discriminantanalyse

Twee verschillende doelen:

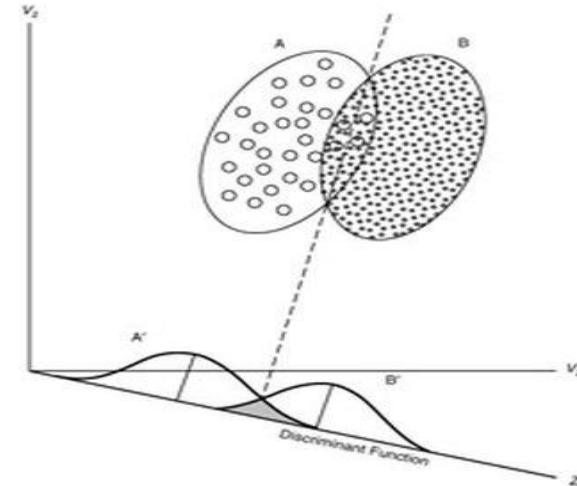
- Verschillen bepalen tussen –wederzijds uitsluitende- groepen (***descriptieve – beschrijvende- discriminantanalyse***)
 - groep wordt weergegeven door de afhankelijke variabele
 - patronen vinden in de waarden van de onafhankelijke variabelen
- Bepalen tot welke groep een nieuwe waarneming behoort. (***predictieve – voorspellende- discriminantanalyse***)



(From S. Wu's website)

Descriptieve discriminantanalyse

- **Doel:** de relatieve belangrijkheid van kenmerken om de verschillende groepen van elkaar te onderscheiden bepalen
- **Concreet:** kenmerken die meer en/of beter differentiëren tussen de verschillende groepen krijgen een groter gewicht.
- De onafhankelijke variabelen, worden de ***discriminanten*** (of de criteria) genoemd.
- Het resultaat is een ***discriminantfunctie***.
 - Hierbij wordt de waarde voor elk van de discriminanten vermenigvuldigd met een gewicht en bij elkaar opgeteld (lineaire descriptieve discriminant analyse).

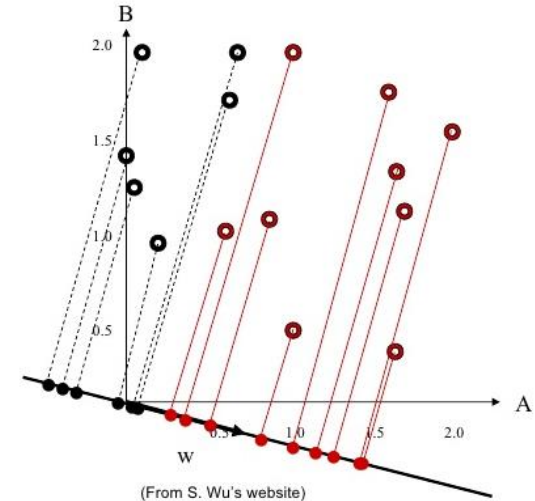


Descriptieve discriminantanalyse

- Indien de afhankelijke variabele slechts 2 waarden kan aannemen, dan is er één discriminantfunctie. Zijn er meerdere mogelijke waarden (meerdere groepen), dan zijn er meerdere discriminantfuncties
 - Het aantal discriminantfuncties is gelijk aan het minimum van:
 - het aantal waarden voor de afhankelijke variabele vermindert met één
 - het aantal onafhankelijke variabelen
 - Deze discriminantfuncties zijn onderling niet gecorreleerd.
- De gewichten in een discriminantfunctie weerspiegelen enkel de belangrijkheid van de verschillende discriminaten wanneer het gaat om gestandaardiseerde coëfficiënten

Predictieve discriminantanalyse

- **Doel:** door gebruikt te maken van de discriminantfunctie een **discriminantfunctiescore** berekenen om te bepalen tot welke groep een nieuwe waarneming behoort
- **Concreet:** de som wordt gemaakt van de waarden van de waarneming die overeenstemmen met de discriminanten vermenigvuldigd met de overeenstemmende gewichten.
- In feite worden de kansen bepalen dat een nieuwe waarneming in de verschillende mogelijke klassen thuis hoort. Hierbij wordt gesteund op de wet van Bayes.



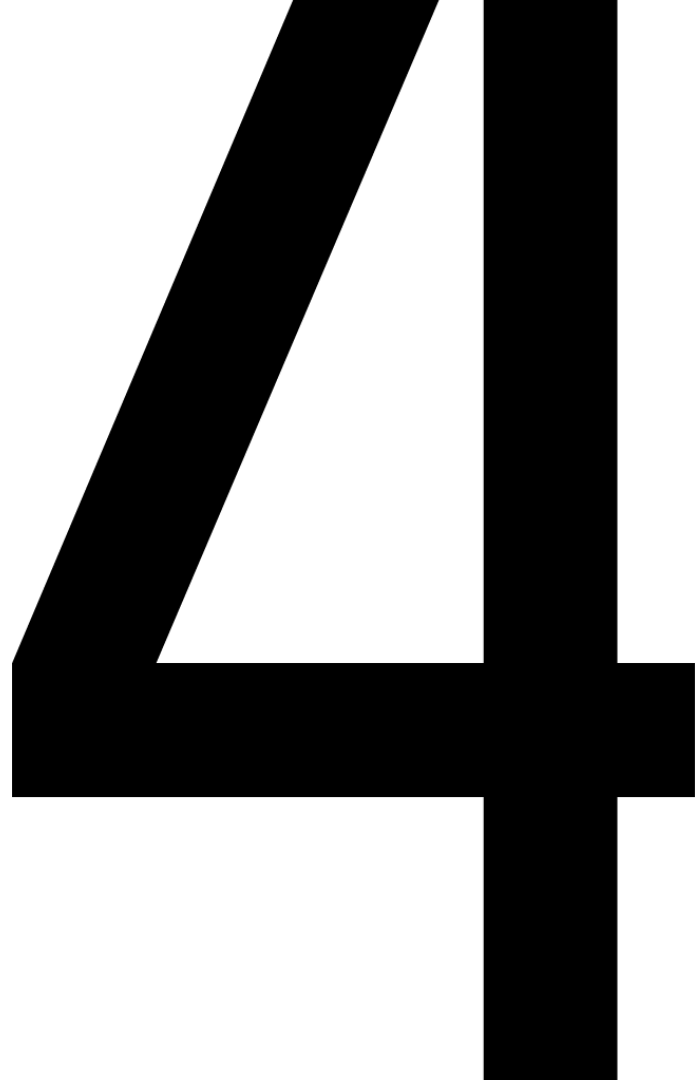
$$P(y = k|x) = \frac{P(x|y = k)P(y = k)}{P(x)} = \frac{P(x|y = k)P(y = k)}{\sum_l P(x|y = l) \cdot P(y = l)}$$

Veronderstellingen mbt de data

Meerdere veronderstellingen mbt de data moeten worden gemaakt (bepaalde kunnen ook statistisch getoetst worden)

- Er is **geen afhankelijkheid** tussen de onafhankelijke variabelen
- Tussen elk paar van onafhankelijke variabelen is er per waarde van de afhankelijke variabele **lineariteit**
- **Geen hoge correlaties** tussen de onafhankelijke variabelen.
- **Multivariate normaliteit.** Dit stelt dat de verdeling van een lineaire combinatie van de onafhankelijke variabelen, overeenkomt met een normale verdeling. In de praktijk wordt deze veronderstelling vervangen door de veronderstelling dat de verdeling van de gegevens van elke van de onafhankelijke variabelen een normale verdeling is.
- **Homogeniteit** van de variantie-covariantiematrices

Discriminantanalyse met Python



Discriminantanalyse met Python

We voeren een discriminantanalyse uit op de biopsy dataset. Hierbij gebruiken we enkel de waarden in de kolommen V1, V2 en V3 om de waarde in de kolom 'class' te voorspellen.

| | ID | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | class |
|---|---------|----|----|----|----|----|-----------|----|----|----|-----------|
| 0 | 1000025 | 5 | 1 | 1 | 1 | 2 | 1.000000 | 3 | 1 | 1 | benign |
| 1 | 1002945 | 5 | 4 | 4 | 5 | 7 | 10.000000 | 3 | 2 | 1 | benign |
| 2 | 1015425 | 3 | 1 | 1 | 1 | 2 | 2.000000 | 3 | 1 | 1 | benign |
| 3 | 1016277 | 6 | 8 | 8 | 1 | 3 | 4.000000 | 3 | 7 | 1 | benign |
| 4 | 1017023 | 4 | 1 | 1 | 3 | 2 | 1.000000 | 3 | 1 | 1 | benign |
| 5 | 1017122 | 8 | 10 | 10 | 8 | 7 | 10.000000 | 9 | 7 | 1 | malignant |
| 6 | 1018099 | 1 | 1 | 1 | 1 | 2 | 10.000000 | 3 | 1 | 1 | benign |
| 7 | 1018561 | 2 | 1 | 2 | 1 | 2 | 1.000000 | 3 | 1 | 1 | benign |



Descriptieve Discriminant analyse met Python

```
>>> from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
>>> biopsy = pd.read_csv('biopsy.csv')
>>> X = biopsy[['V1','V2', 'V3']]
>>> y = biopsy['class']
>>> lda = LinearDiscriminantAnalysis()
>>> lda.fit(X,y)
```

waarbij:

- X een array-achtige data structuur is met de **onafhankelijke variabelen**
- y een array-achtige data structuur is met de **afhankelijke variabele**

```
>>> lda.classes_ # values of the dependent variable
>>> lda.coef_ # NOT the coefficients of the discriminant function (see next slide)
>>> lda.priors_ # probability of an observation coming from a particular group
>>> lda.explained_variance_ratio_ # how much of the variance is explained by
# each of the discriminant functions (only useful when there are several discriminant functions)
```

Descriptieve Discriminant analyse met Python

Om de coëfficiënten van de discriminant functie(s) te bekomen, maak je best gebruik van volgende functie:

```
def LDA_coefficients(X,lda):
    nb_col = X.shape[1]
    matrix= np.zeros((nb_col+1,nb_col), dtype=int)
    Z=pd.DataFrame(data=matrix,columns=X.columns)
    for j in range(0,nb_col):
        Z.iloc[j,j] = 1
    LD = lda.transform(Z)
    nb_funct= LD.shape[1]
    resultaat = pd.DataFrame();
    index = ['const']
    for j in range(0,LD.shape[0]-1):
        index = np.append(index,'C'+str(j+1))
    for i in range(0,LD.shape[1]):
        coef = [LD[-1][i]]
        for j in range(0,LD.shape[0]-1):
            coef = np.append(coef,LD[j][i]-LD[-1][i])
        result = pd.Series(coef)
        result.index = index
        column_name = 'LD' + str(i+1)
        resultaat[column_name] = result
    return resultaat
```

Descriptieve Discriminant analyse met Python


Visualisatie van de discriminantanalyse bestaande uit de eerste 3 gemeten waarden van de biopsie (V1, V2 en V3):

```
# map the independent variables based on the discriminant functions of the model to their N  
# discriminant values
```

```
>>> LD = lda.transform(X)
```

```
# combine with the original dependent variable
```

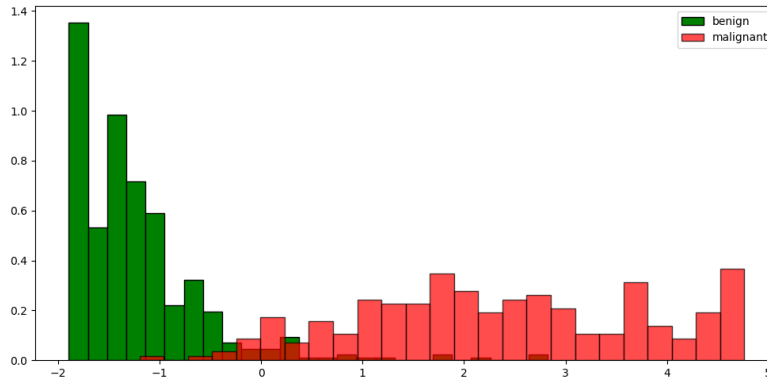
```
>>> LD_df = pd.DataFrame(zip(LD[:,0], biopsy['class']),  
                          columns=['LD1', 'Target'])
```



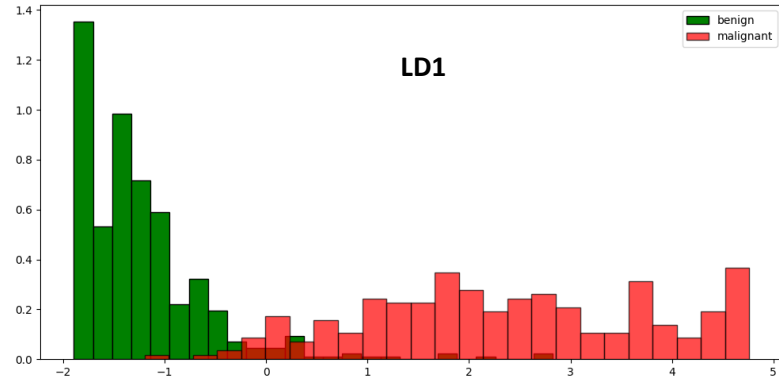
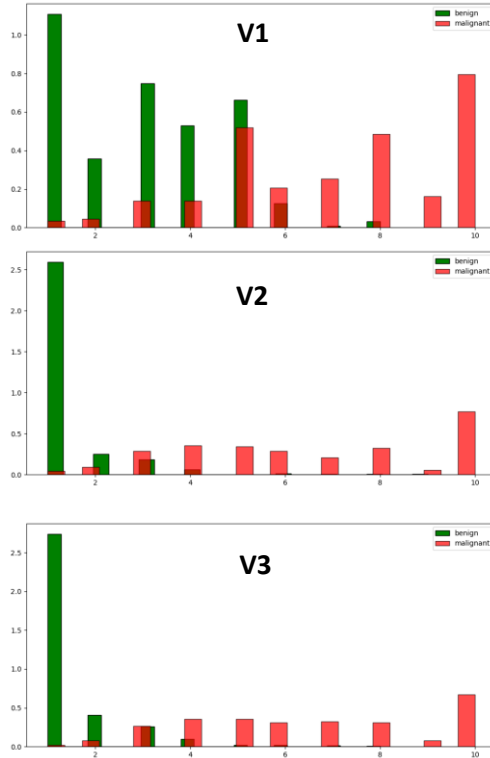
| | V1 | V2 | V3 | LD1 | Target |
|---|----|----|----|-----------|-----------|
| 0 | 5 | 1 | 1 | -0.809612 | benign |
| 1 | 5 | 4 | 4 | 0.539399 | benign |
| 2 | 3 | 1 | 1 | -1.325530 | benign |
| 3 | 6 | 8 | 8 | 2.596039 | benign |
| 4 | 4 | 1 | 1 | -1.067571 | benign |
| 5 | 8 | 10 | 10 | 4.011297 | malignant |

Descriptieve Discriminant analyse met Python

```
>>> plt.figure()
>>> LD_df.hist(column=['LD1'], by='Target', bins=25, density=True, edgecolor='black',
               color='cyan', sharex=True, sharey=True, figsize=(10,10), layout=(2,1))
>>> fig, ax = plt.subplots(figsize=(10,5))
>>> LD_df['LD1'][LD_df['Target'] == 'benign'].hist(ax=ax, bins=25, density = True,
           edgecolor='black', color='green', label='benign')
>>> LD_df['LD1'][LD_df['Target'] == 'malignant'].hist(ax=ax, bins=25, density = True,
           edgecolor='black', color='red', alpha=0.7, label='malignant')
>>> ax.legend()
>>> ax.grid(False)
>>> plt.show()
```



Descriptieve Discriminant analyse met Python



Predictieve discriminant analyse met Python

De bekomen resultaten kunnen we gebruiken om te vergelijken met de geobserveerde klassen om zo een idee te vormen van de kwaliteit van de discriminantanalyse:

```
>>> predicted = pd.Series(lda.predict(X), name='predicted')
>>> actual = biopsy['class'].rename('actual')
>>> pd.crosstab(index=actual, columns=predicted, margins='all', margins_name='total')
```

```
#output:      predicted  benign  malignant  total
actual
benign        448         10        458
malignant      33        208        241
total         481        218        699
```

Dit is een **confusion matrix**
(zie Evaluatie metriecken)

```
>>> lda.score(X,y)
```

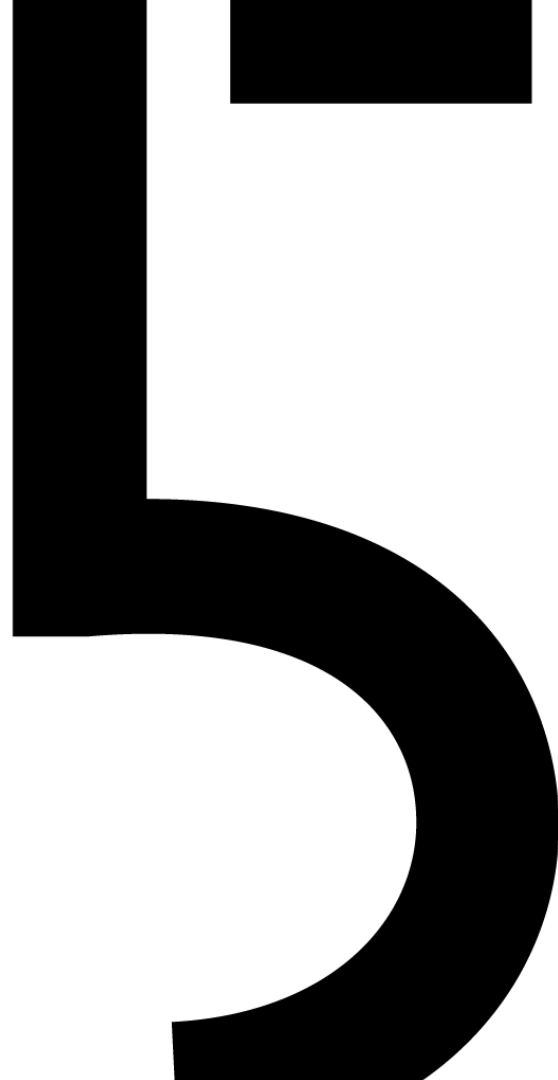
```
#output:
```

```
0.9384835479256081
```

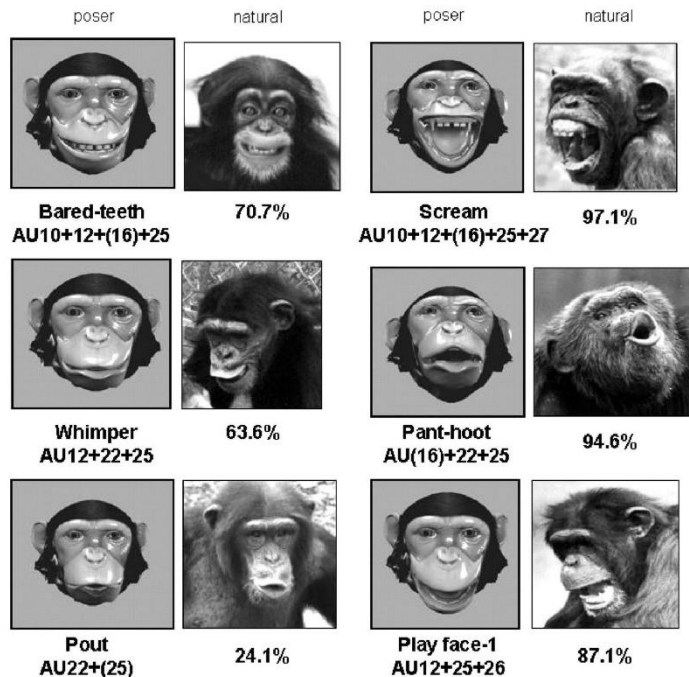
Dit is de **accuracy**: volgens de discriminantanalyse komt 6.15% in de andere klas terecht dan waar ze werkelijk in zitten, terwijl 93,85% correct wordt geklasseerd. $Accuracy = (448+208)/699$



In de praktijk



Gezichtsuitdrukkingen herkennen



Understanding chimpanzee facial expression: insights into the evolution of communication

Lisa A. Parr, Bridget M. Waller

Social Cognitive and Affective Neuroscience, Volume 1, Issue 3, 1 December 2006, Pages 221–228, <https://doi.org/10.1093/scan/nsl031>

Published: 01 December 2006 [Article history](#) ▼

An illustration of prototypical chimpanzee facial expressions. These are listed in pairs. The example on the left side of the pair shows the Poser animated expression, while the example on the right shows a naturalistic chimpanzee expression. Under the Poser expression is the prototypical AU configuration as identified by the **Discriminant Functions Analysis**, and under the naturalistic expression is the percentage agreement between AU configuration and a priori classification for that category.

Verwerken van satellietbeelden

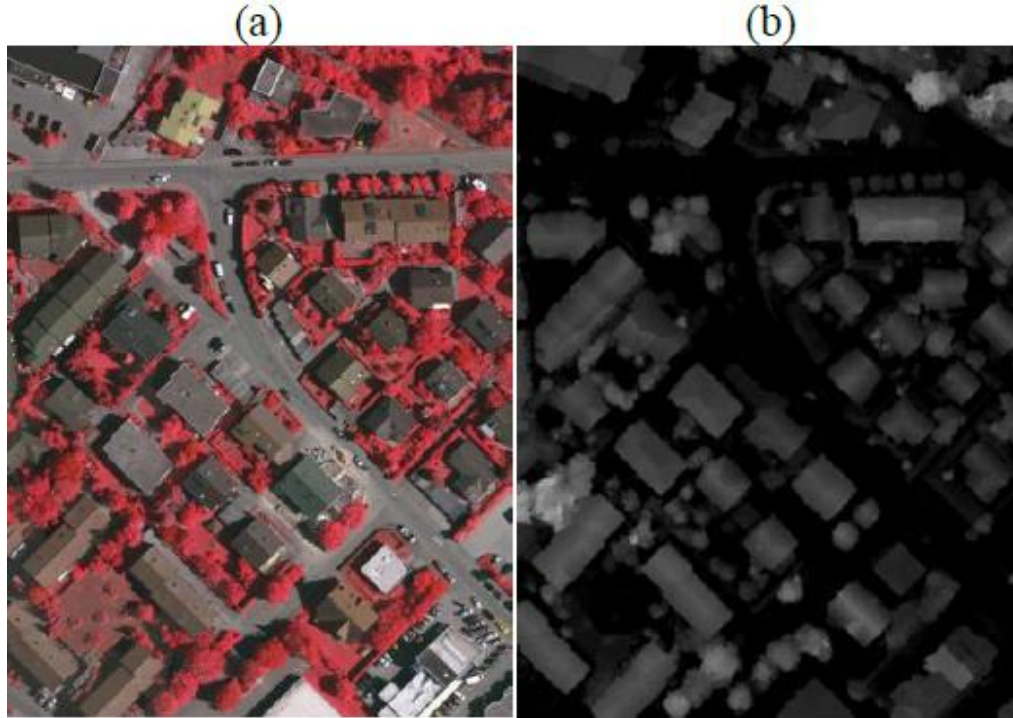


Figure 1. The sample image "area30" in the dataset (a) and the projection of its normalized DSM (b)

| Class | Threshold {0,1,2, ..., 255} | Best Binary Classification Score |
|--------------|--------------------------------|-------------------------------------|
| "Road" | 127 | 0.7984 |
| "Building" | 152 | 0.7447 |
| "Vegetation" | 107 | 0.6438 |
| "Tree" | 88 | 0.8009 |

<https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-3-W4/429/2018/isprs-archives-XLII-3-W4-429-2018.pdf>



Vragenlijst



Vragenlijst



- Download het bestand *vragenlijst 21-22.xlsx* van Canvas
- Exporteer het excel-bestand als een csv bestand
- Plaats *vragenlijst 21-22.csv* in je Python workspace
- Lees de data in en plaats het in het dataframe

studenq

```
>>> import pandas as pd
```

```
>>> studenq = pd.read_csv('vragenlijst 21-22.csv', delimiter=';',  
decimal='.')
```

Vragenlijst



1.a Kan je de schrijfhand van volgende student voorspellen op basis van de opgegeven lengte (182), stukken fruit (2), schoenmaat (44), afstand tot KdG (22)? Gebruik enkel de studenten die Links of Rechts als schrijfhand hebben opgegeven.

Vragenlijst



1.b Stel een confusion matrix op op basis van de voorspelde en de effectieve waarden voor de data die je gebruikt hebt om de discriminantanalyse uit te voeren (zie 1.a)

Vragenlijst

1.c Bereken de gebruikelijke evaluatie metrieken voor een binaire classifier



1.d Teken de ROC-curve. Verklaar –indien van toepassing– waarom de ROC-curve niet wordt getekend

Vragenlijst

2. Wat is de accuracy van de discriminantanalyse waar de lengte en de schoenmaat het aantal broers en zussen voorspelt?



Oefeningen







Vragenlijst

Oplossingen

Antwoorden vragenlijst



1.a Kan je de schrijfhand van volgende student voorspellen op basis van de opgegeven lengte (182), stukken fruit (2), schoenmaat (44), afstand tot KdG (22)? Gebruik enkel de studenten die Links of Rechts als schrijfhand hebben opgegeven.

```
>>> subset = studenq[['lengte', 'stukken fruit', 'schoenmaat', 'afstand tot KdG',  
                    'schrijfhand']].copy()  
>>> subset = subset[subset.schrijfhand != 'Beide']  
>>> subset.dropna(inplace=True)  
>>> subset = subset.reset_index() # zonder dit een impact op evaluatie metrieken  
>>> X = subset[['lengte', 'stukken fruit', 'schoenmaat', 'afstand tot KdG']]  
>>> y = subset['schrijfhand']  
>>> lda = LinearDiscriminantAnalysis()  
>>> lda.fit(X,y)  
>>> lda.predict([[182,2,44,22]])
```

Antwoorden vragenlijst



1.b Stel een confusion matrix op op basis van de voorspelde en de effectieve waarden voor de data die je gebruikt hebt om de discriminantanalyse uit te voeren (zie 1.a)

```
>>> predicted = pd.Series(lda.predict(X), name='predicted')
>>> actual = subset['schrijfhand'].rename('actual')
>>> conf_mat = pd.crosstab(index=actual, columns=predicted)
# De confision matrix is niet volledig!!
>>> conf_mat = square_conf_mat(conf_mat)
#Persoonlijke functie: bekomen van een vierkante confusion matrix
>>> accuracy(conf_mat) #Persoonlijke functie
```

Antwoorden vragenlijst

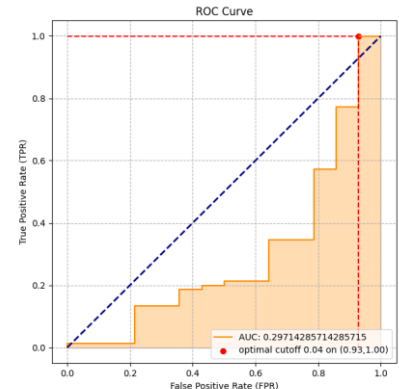


1.c Bereken de gebruikelijke evaluatie metrieken voor een binaire classificator

```
>>> overviewmetrics(conf_mat,1) #Persoonlijke functie  
>>> positiverates(conf_mat) #Persoonlijke functie
```

1.d Teken de ROC-curve. Verklaar –indien van toepassing– waarom de ROC-curve niet wordt getekend

```
>>> y_score = lda.predict_proba(X)[: ,0]  
>>> plot_roc(actual, y_score, pos_label='Rechts')  
#Persoonlijke functie
```



Antwoorden vragenlijst



2. Wat is de accuracy van de discriminantanalyse waar de lengte en de schoenmaat het aantal broers en zussen voorspelt?

```
>>> subset2 = studenq[['lengte','schoenmaat', 'siblings']].copy()
>>> subset2.dropna(inplace=True)
>>> subset2 = subset2.reset_index() # zonder dit impact op evaluatie metriecken
>>> X2 = subset2[['lengte','schoenmaat']]
>>> y2 = subset2['siblings']
>>> lda2 = LinearDiscriminantAnalysis()
>>> lda2.fit(X2,y2)

>>> predicted2 = pd.Series(lda2.predict(X2), name='predicted')
>>> actual2 = subset2['siblings'].rename('actual')
>>> conf_mat2 = pd.crosstab(index=actual2, columns=predicted2 )
>>> conf_mat2 = square_conf_mat(conf_mat2)
        #Persoonlijke functie: bekomen van een vierkante confusion matrix
>>> accuracy(conf_mat2) # persoonlijke functie
```