Loes Modderman

Samenvatting Udacity Course

[Intro to Machine Learning 2](#_Toc510605222)

[Machine Learning methodes 4](#_Toc510605223)

[Datasets 8](#_Toc510605224)

[Classificatie vs. regressie 8](#_Toc510605225)

[Uitschieters 9](#_Toc510605226)

[Clustering 9](#_Toc510605227)

[Text Learning 10](#_Toc510605228)

[Principal Component Analysis 10](#_Toc510605229)

[Validation 11](#_Toc510605230)

[Evaluation metrics 11](#_Toc510605231)

# Intro to Machine Learning

## Features & labels

In een beschrijvende dataset zijn de features de kenmerken en de labels zijn de bijbehorende resultaten van deze kenmerken.

### Feature scaling

Features kun je scalen (schalen) om ze makkelijker met elkaar te kunnen vergelijken. Zo krijgen ze vergelijkbare ranges, tussen 0 en 1. Dit wordt ook wel normalization genoemd.

Je kunt ook standardiseren door gebruik te maken van het gemiddelde en de standaarddeviatie.

### Feature selection

#### Een nieuwe feature toevoegen

* Gebruik je eigen intuïtie
* Codeer de nieuwe feature
* Visualiseer het resultaat
* Herhaal

#### Features vs. Informatie

Figuur 1 Regularization

Features en informatie zijn twee verschillende dingen. Features proberen aan informatie te komen. En informatie is hetgeen wat we willen. Wel willen we hiervoor zo min mogelijk features gebruiken om dus aan zoveel mogelijk informatie te komen.

#### Regularization

Regularization is een methode voor feature selection waarbij je vooraf bepaald hoeveel features jouw model nodig heeft om optimaal te werken. Dit model wordt bepaald met de Lasso regressie: met als een parameter en als coefficiënten van de regressie. Dit model is visueel weergegeven in Figuur 1.

### Measurable vs. Latent features

Measurable features zijn features die je letterlijk kunt meten. Latent features zijn features die je niet direct kunt meten, maar die de measurable features kunnen beschrijven.

## Supervised & Unsupervised learning

Er zijn verschillende methodes om machine learning toe te passen. Enkele hiervan zijn de supervised en de unsupervised learning methodes. Supervised wordt gebruikt op datasets waarbij er een label is meegegeven. De machine learning (ML) methode heeft een voorbeeld en past zichzelf, naarmate het leert, aan het voorbeeld aan. Men kan met deze methode regressie uitvoeren en classificeren. Unsupervised learning wordt gebruikt wanneer er alleen features beschikbaar zijn, maar geen labels. Hier is dan sprake van clustering.

## Decision surface

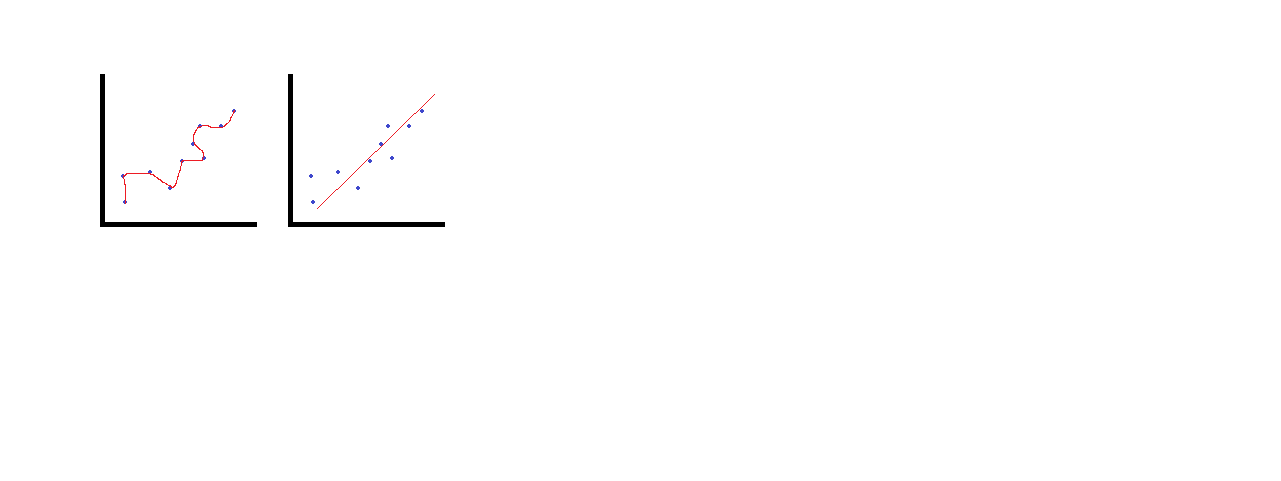
Een decision surface is een oppervlak dat bepaald welk label een datapunt krijgt. Welk oppervlak dit is wordt bepaald door een lijn die onderscheid maakt tussen verschillende klassen. Deze lijn wordt op zo’n manier berekend, dat het nieuwe data in kan delen gebaseerd op het gegeven van de oude data. Dit wordt bepaald m.b.v. verschillende ML methodes.

# Machine Learning methodes

Er zijn verschillende ML methoden. In dit hoofdstuk worden enkel de supervised methoden besproken. Bij al deze methoden horen enkele algemene zaken waar altijd rekening mee gehouden moet worden, zoals overfitting en bias & variance.

### Overfitting

Dit fenomeen treedt plaats als je je model zo traint op de trainingsset, dat het perfect erop werkt. Later, in de testfase, blijkt dat het model toch niet zo goed werkt. Het lijkt dan alsof je model perfect is, terwijl het in werkelijkheid heel slecht werkt. In Figuur 2 (a) is weergegeven hoe overfitting eruit ziet tegen over een correcte regressie in Figuur 2 (b).



Figuur 2 v.l.n.r. (a) Overfitting (b) Correcte regressie

### Bias & Variance

Je moet proberen een balans te vinden tussen de bias en de variance.

|  |  |
| --- | --- |
| High bias | High variance |
| Negeert praktisch alle data | Reageert heel sensitive op alle data (kans op overfitting) |
| Grote error op de trainingsset (lage en een grote SSE) Beide worden in een later hoofdstuk toegelicht | Een veel grotere error op de testset dan op de trainingsset |
| Gebruikt weinig features | Gebruikt veel features |

Je balans is te vinden als je weinig features hebt, een hoge en een lage SSE.

## Naïve Bayes

Deze methode is gebaseerd op Bayes-theorie. Hierbij worden de volgende termen gebruikt:

* Sensitivity (kans op een positief resultaat als er sprake is van A)
* Specitivity (Kans op een negatief resultaat als er geen sprake is van A)
* Prior (kans op A)
* Joint Posterior = Prior \* Sensitivity (kans op A als het resultaat positief is & kans op niet A als het resultaat positief is)
* Normalizer =
* Posterior =

#### Voordeel

Naïve Bayes is gemakkelijk te implementeren.

#### Nadeel

Naïve Bayes gaat ervan uit dat de features onderling onafhankelijk zijn.

## Support Vector Machine

Support Vector Machine (SVM) maakt een hyperplane (of lijn) die de data scheidt op zo’n manier dat het de afstand tot het dichtstbijzijnde punt (de marge) maximaliseert. Dit noem je ook wel maximaxing robustness. Terwijl je dit allemaal doet, moet je ook de data correct classificeren met de hyperplane.b

### Parameters

Bij SVM is er sprake van parameters die je zelf kunt aanpassen

* Kernel: een kernel trick zorgt ervoor dat je niet-lineaire data toch lineair kunt scheiden.
* C: controleert de balans tussen een smooth decision boundary en het correct classificeren van training datapunten. Een hoge C geeft meer correcte trainingspunten. Een lage C geeft een meer straight forward lijn.
* Gamma: Definieërt tot hoe ver de invloed van een enkel training datapunt kan gaan. Een hoge gamma zorgt ervoor dat de punten die dihctbij liggen vooral de decision boundary bepalen. Een lage gamma zorgt ervoor dat de verry punten ook veel weging hebben. Dit kan echter wel voor een ‘wiggly’ decision boundary zorgen.

#### Voordeel

SVM werkt goed in complexe datasets waarin in duidelijke marge of scheiding in voorkomt.

#### Nadelen

SVM werkt niet goed in grote datasets, vanwege de lange training tijd. Verder werkt het niet heel goed als er sprake is van veel ruis. Dit maakt de methode namelijk gevoelig voor overfitting.

## Decision Tree

Met Decision Trees (DT) kun je beslissingen maken over non-lineaire data m.b.v. simpele lineaire decision surfaces.

### Parameters

Er is hierbij één parameter van belang op dit moment en dat is Min\_samples\_split. Deze parameter geeft aan of er genoeg samples zijn om te kunne blijven splitten. Hoe meer samples je aangeeft, hoe minder complex de decision line wordt en hoe hoger de accuracy. Hoe minder samples je aangeeft, hoe complexer de decision line wordt en hoe lager de accuracy is (met kans op overfitting).

### Entropy

De entropy is hoe een DT bepaalt waar het de data moet splitten. De definitie van entropy is een ‘measure of impurity’. ‘Purity’ betekent hier zuiver, oftewel alle data in de set van voorbeelden is dan goed geclassificeerd. Een DT bepaalt met de entropy waar een decision line het beste werkt op basis van purity.

Als alle voorbeelden van dezelfde klassen zijn, dan is de entropy 0. Als alle voorbeelden gelijk verdeeld zijn over alle beschikbare klassen, dan is de entropy 1.0.

### Information Gain

De information gain geeft aan hoeveel informatie er gewonnen wordt bij het maken van een decision line.

#### Voordelen

DT is gemakkelijk in het gebruik en is ook zeer goed grafisch weer te geven.

#### Nadeel

DT is gevoelig voor overfitting en dan met name voor data met een heleboel features.

## Andere algoritmes

Er zijn nog een heleboel andere algoritmes waar ik over kan uitweiden. Ik wil nog enkele kort toelichten.

### Adaboost

Een binair classificatie algoritme dat de werking van een ML algoritme een boost geeft. Het meest gebruikelijke algoritme hiervoor zijn decision treest met één level (decision stumps). Het algoritme begint met alles dezelfde weging geven:

Voor elke iteratie:

* Leer .
* Compute coefficient .
* Recompute weights van .
* Normaliseer wegingen van .

Het uiteindelijke model voorspeld d.m.v.:

Splitten op elke feature apart (decision stumps). Voor elke stump wordt de weighted\_error bepaald. De feature met de laagste weighted\_error, wordt de feature waarmee begonnen wordt met beslissen. Hierna wordt de coefficient berekend. Hierna worden alle wegingen van de overige features weer opnieuw berekend, enz.

### k-Nearest Neighbour

* Bij het kNN algoritme wordt er evenveel centrumpunten bepaald als dat er klassen zijn.
* Alle punten binnen een bepaalde range van zo’n centrumpunt, behoren nu tot die klasse.
* Voor deze nieuwe klasse, wordt een nieuw centrumpunt bepaald door het gemiddelde van alle data punten te berekenen.
* Herhaal stap 2 en 3, totdat de verandering van het centrumpunt onder een bepaalde grens komt of totdat een x aantal iteraties is bereikt.

### Random Forest

Random Forest algoritme kan gebruikt worden voor classificatie en regressie problemen. Dit algoritme maakt gebruik van DT. Hier volgt alleen de uitleg voor de classificatie methode. RF zorgt voor een robuuster model. Hoe meer trees in je forest, hoe kleiner de kans op overfitting van het model wordt.

RF werkt als volgt:

* Elke DC stelt andere regels op. Zo komt elke DT uit op een ander resultaat.
* Het resultaat met de meeste ‘votes’ (het resultaat van één DT), is het uiteindelijke antwoord.

# Datasets

## Datatypen

Er zijn verschillende soorten datatypen.

* Numerieke data bestaat uit numerieke waarden, zoals nummers
* Categorische data bestaat uit een eindig aantal discrete waarden (categorieën)
* Tijdreeksen is data die bestaat uit datums of een timestamp

## Accuracy vs. Trainingset grootte

De grootte van je trainingsset heeft invloed op de accuracy van je ML methode. Een kleine trainingsset geeft een lage accuracy en een grote trainingsset geeft een hoge accuracy.

# Classificatie vs. regressie

Dit geeft aan in hoeverre de verandering in de output (y) verklaard worden door de verandering in de input (x). De waarde is altijd tussen 0 en 1. 0 betekend dat er geen sprake is van regressie. 1 betekend dat er sprake is van een zeer goede relatie tussen x en y.

## Multivariate regression

Meerdere variabelen hebben invloed op het resultaat (output y)

|  |  |  |
| --- | --- | --- |
| Eigenschappen | Supervised classificatie | Regressie |
| type Output | Discreet (class labels) | Continu (nummer) |
| Wat probeer je te vinden? | Decision boundary | ‘best fit line’ |
| Evaluatie | Accuracy | ‘sum of squared error’ / R^2 |

# Uitschieters

Datapunten die ver buiten de andere datapunten voorkomen, noemen we uitschieters. Deze uitschieters kunnen voor een verkeerde regressielijn zorgen. Een sensor malfunction of een data entry error kunnen zorgen voor uitschieters. Deze willen we negeren, omdat het geen correcte data is. Een freak event kan ook zorgen voor uitschieters. Deze uitschieters bekijken we aandachtig.

Hoe groter/verder weg je uitschieter zit, hoe meer invloed het heeft op je regressielijn.

## Detection

1. Train
2. Remove ~ 10%
3. Train again
4. Evaluate and Repeat from step 2 if necessary.

## Rejection

1. Train
2. Remove points with largest residual error
3. Re-train

# Clustering

Clusteren is een manier om unsupervised learning waar te maken. Dit houdt in dat de data niet gelabeld is.

## K-Means

Meest gebruikte clustering algoritme is k-means. Hierbij worden centroides bepaald. Aan de hand van deze centroides wordt bepaald welke datapunten er het dichtste bij liggen. Met deze ‘groep’ datapunten wordt bepaald wat het nieuwe punt van de centroide wordt door de afstand (punt – centroide) in het kwadraat te minimaliseren. Deze stappen worden herhaald, zodat er verschillende clusters ontstaan.

k-means is een hill climbing algorithm, waardoor het mogelijk is dat er een lokaal minimum ontstaat.

Hoe meer centroides je hebt, hoe meer lokale minima je gaat vinden.

# Text Learning

Text learning is het bewerken van tekstdata om hiermee verdere analyses uit te voeren. Hierbij moet opgelet worden dat niet alle woorden even belangrijk zijn. Zo bevatten stopwoorden minder informatie minder vaak voorkomende woorden.

## Stemmer

Een stemmer zoekt de stam bij elk woord. Zo kun je sommige worden onder één stam plaatsen.

## Bag of Words

Een Bag of Words (BoW), ook wel een count vectorizer genoemd, telt alle woorden in de gehele tekst en maakt hier een count vector van.

## TfIdf

Tf = Term frequency (Zoals met BoW)

Idf = inverse document frequency (Een weging van hoevaak een woord voorkomt in het document)

# Principal Component Analysis

Principal component analysis (PCA) wordt voornamelijk gebruikt om binnen de data de dimensies te reduceren.

## Data dimensionality

PCA specialiseert zich alleen in verschuivingen en verdraaiïngen in een coördinatensysteem.

PCA maakt een nieuwe as d.m.v. de oude as te roteren en transleren. Het centrum van de nieuwe as wordt verplaatst naar het centrum van de data. .

De data hoeft niet perfect 1D te zijn om PCA toe te passen

## Maximal variance & Information loss

De informatie die je verliest wordt groter, naarmate de data verder van de x’-axis ligt. De information loss is de som van alle punten tot de x’-axis.

Minimaliseer de information loss door de maximale variantie te minimalizeren van de afstand van de ‘oude’ datapunten tot de nieuwe datapunten.

# Validation

De reden dat we een training en een test dataset gebruiken is dat het een schatting geeft van de werkzaamheid van het model op een onafhankelijke dataset (testset). Verder geeft het een extra check of je niet je model aan het overfitten bent.

## Cross validation

Bij cross validation wordt de data set verdeeld in *k* bins. Een van deze bins is op dit moment de test set. Alle andere bins zijn samen de trainingsset. Bepaal hiermee de accuracy van je model. Herhaal dit totdat elke bin een testset is geweest. Hiermee kun je de accuracy van je model maximaliseren.

## Parameter tuning

GridSearch (in sklearn) is een manier om de computer het werk te laten doen m.b.t. het optimaliseren van je model door telkens de parameters te veranderen. Hierbij kunnen de parameters worden uitgezocht om het model mee te optimaliseren.

# Evaluation metrics

Hiermee laat je zien hoe goed je ML methode het doet.

## Accuracy

De accuracy is het percentage van het aantal items in een class die correct gelabeld zijn. Accuracy kan alleen bepaald worden voor supervised ML methoden.

## Confusion matrices

Een confusion matrix laat in één oogopslag zien hoe het model het doet.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual class | |
| Positive | Negative |
| Predicted  class | Positive |  |  |
| Negative |  |  |

Hiermee kun je de volgende termen ook bepalen:

Recall(x) = de kans dat het model x voorspelt, als het ook x is

(true positive / (true positive + false negative)

Precision(x) = de kans het x is, als het model x voorspelt

(true positive / (true positive + false positive)