



ANALYSIS SYSTEM

NETWORK SECURITY

CREATE IDEA

SETTING SYSTEM

CONTROLLING AND SEARCHING

AI-basierte

Geschäftsmodelle und technologische Grundlagen

Recommendation Engines

Oder: Woher weiß Netflix was ich schauen möchte?

E-COMMERCE

KEY WORD

SEARCH BUG

APPLICATION MAKER

SPEED ACCES

Vorstellung - Dr. Anna Bernhard



Station 1 (2006-2011):

Studium der Physik an der Goethe-Universität Frankfurt



Station 2 (2012-2015):

Promotion in Astrophysik an der TU München
Thema der Dissertation: Origin of IceCube's Astrophysical Neutrinos



Station 3 (2015-2017):

Data Analyst bei thyssenkrupp



Station 4 (2017-heute):

Senior Data Scientist bei MAN Truck & Bus



Seit 2018 Dozentin an der FOM

für „Wissenschaftliche Methodik“ im
Masterstudiengang „Business Consulting & Digital
Management“



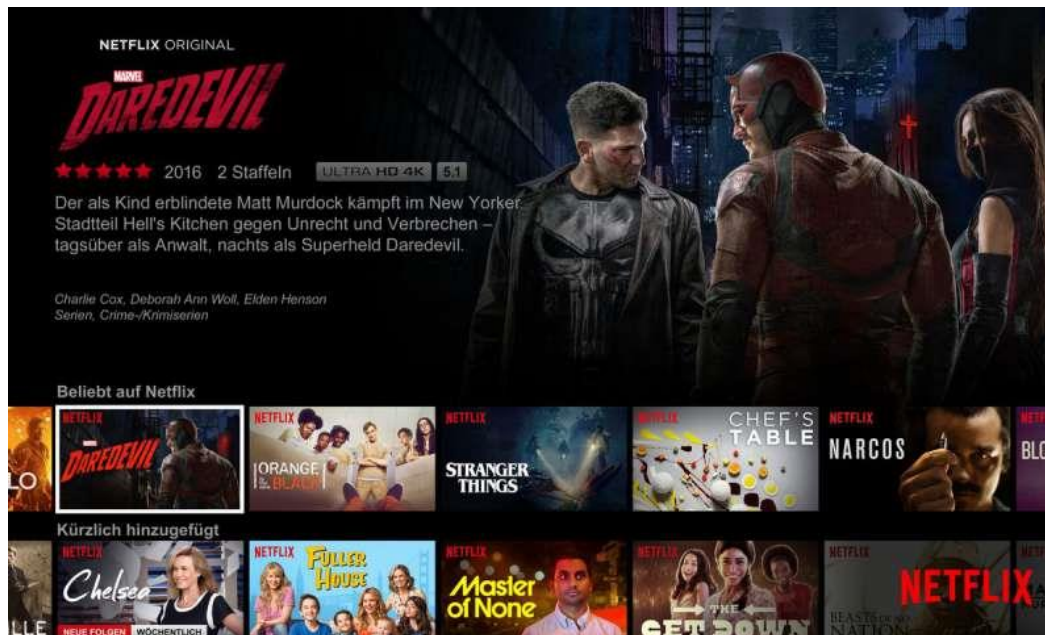
Lernziele

- ✓ Sie haben das **Konzept einer Recommendation Engine verstanden** und können es anhand von Beispielen erklären.
- ✓ Sie können den **Unterschied zwischen inhaltsbasierten Systemen und kollaborativen Filtern** erläutern.
- ✓ Sie können **bekannte Probleme** nennen, die **bei Recommendation Engines** auftreten.
- ✓ Sie können **Beispiele für Distanzmaße** aufzählen und diese kurz erklären.
- ✓ Sie kennen **Beispiele für Gruppierungsmethoden**.

Woher weiß Netflix eigentlich was ich schauen möchte?



- Oder Amazon was ich kaufen möchte?



- **Netflix Prize** (2006) Wettbewerb mit 1M \$ dotiert
- Siegerlösung 2009 von „BellKor's Pragmatic Chaos“¹ Team
- Das Geschäftsmodell beruht dabei auf sogenannten **Recommendation Engines**

¹ Andreas Töscher & Michael Jahr (2009-09-21). ["The BigChaos Solution to the Netflix Grand Prize"](#) Und R. Bell; Y. Koren; C. Volinsky (2007). ["The BellKor solution to the Netflix Prize"](#) .

Was ist eine Recommendation Engine?

- “Empfehlungsdienste” als spezielle Filter, die dem Kunden helfen für ihn relevante Inhalte aus großen Datenmengen zu finden
z.B. Wie findet man interessante Songs aus aller Musik der Welt?
- Recommender benutzen Präferenzen um Vorhersagen zu treffen
 - Input ist das Feedback über gefallen oder nicht gefallen
Explizit: z.B. Sterne oder Daumen
Implizit: z.B. Anzahl gehörter Songs
- Output ist eine Liste von vorgeschlagenen Inhalten basierend auf dem Feedback

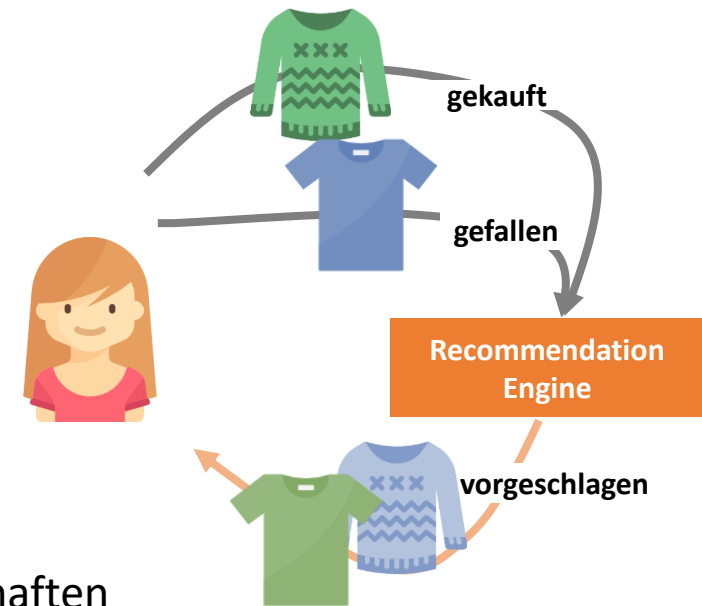


Quelle: Rina Piccolo, <https://www.rinapiccolo.com/piccolo-cartoons/>

- **Zwei Typen an Recommendern**
 1. Content-based
 2. Collaborative filtering

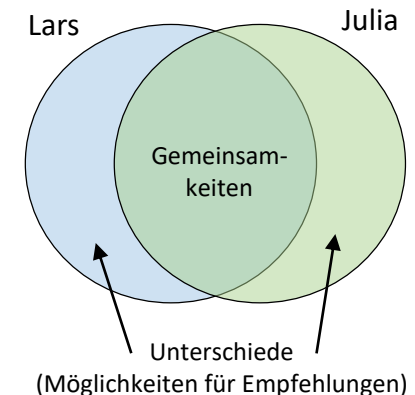
Inhaltsbasierte Systeme

- **Inhaltsbasierte Systeme benutzen Eigenschaften der Artikel um Ähnlichkeiten zu bereits gekauften Artikeln zu finden**
 - z.B. Filme: Schauspieler, Regisseur, Orte, Thema Bücher: Autor, Thema, Verlag, Seitenanzahl
 - Der Geschmack des Nutzers definiert dabei durch Bewertungen oder Kaufentscheidungen die Gewichte der Eigenschaften
- **Inhaltsbasierte Systeme sind domänenspezifisch, da die Eigenschaften nicht produktübergreifend sind**
 - z.B. Ein Rosamunde Pilcher Fan bekommt keine gelben Gummistiefel vorgeschlagen
- **Beispiele für inhaltsbasierte Systeme:**
 - Jemand mag 1980's Action Filme mit Chuck Norris -> *Delta Force*
 - Jemand mag abstrakten Rock aus den 70's -> *Dark Side of the Moon*



Collaborative Filtering

- **Kollaboratives Filtern benutzt Präferenzen von ähnlichen Nutzern, um neue Produkte vorzuschlagen**
 - Dieses Vorgehen ist ähnlich zu Empfehlungen im Freundeskreis
- **Dieser Ansatz ist *nicht* Domänenspezifisch**, da das System nichts über die eigentlichen Produkte “weiß”, sondern rein nach Nutzerinteressen filtert
 - Dadurch können neue Produktkategorien vorgeschlagen werden
- **Kollaboratives Filtern besteht meistens aus zwei Schritten:**
 1. Suche nach Nutzern, die das gleiche Verhaltensmuster wie der aktive Nutzer haben.
 2. Verwendung der Verhaltensmuster um eine Vorhersage für diesen Nutzer zu treffen.





Übung 1

Sie bauen eine Plattform für Computerspiele auf. Welche Daten sollten Sie von ihren Kunden erheben um inhaltsbasierte Empfehlungen geben zu können?

- a. Demographische Daten der Nutzer
- b. Eigenschaften der installierten Spiele des einzelnen Nutzers
- c. Login-Daten der Nutzer



Übung 1

Sie bauen eine Plattform für Computerspiele auf. Welche Daten sollten Sie von ihren Kunden erheben um inhaltsbasierte Empfehlungen geben zu können?

- a. Demographische Daten der Nutzer
- b. Eigenschaften der installierten Spiele des einzelnen Nutzers
- c. Login-Daten der Nutzer

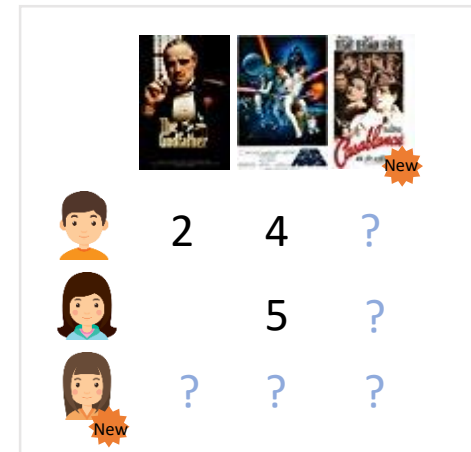
Richtig ist **b.**

Bei inhaltsbasierten Systemen werden Empfehlungen anhand der Eigenschaften der einzelnen Produkte abgeleitet, z.B. ein Computerspiel gehört zur Kategorie Fantasy und kann mit mehreren Spielern gespielt werden. Andere Spiele, welche die gleichen Kategorien erfüllen werden vorgeschlagen.

Bekannte Probleme von Recommendern

1. Das “Cold Start Problem”

- Limitiert kollaboratives Filtern
- Neue Nutzer haben noch keine Interessen hinterlegt und können somit keine sinnvollen Empfehlungen erhalten
- Neue Produkte haben ebenfalls noch keine Historie und können nicht empfohlen werden
- Workaround: Zuerst inhaltsbasierte Systeme nutzen, dann umsteigen, Fragebogen

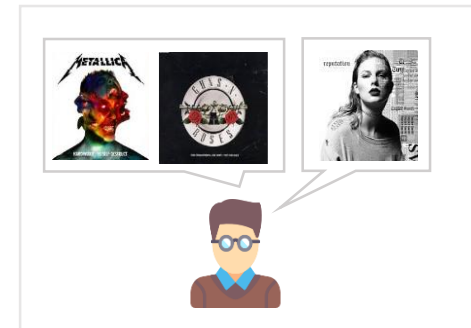


The diagram shows a 3x3 matrix representing user ratings for three movies. The movies are represented by their posters: *Indiana Jones*, *Avatar*, and *Challenger* (marked with a 'New' star). The users are represented by icons: a man, a woman, and a woman (marked with a 'New' star). The ratings are as follows:

	Indiana Jones	Avatar	Challenger (New)
Man	2	4	?
Woman 1		5	?
Woman 2 (New)	?	?	?

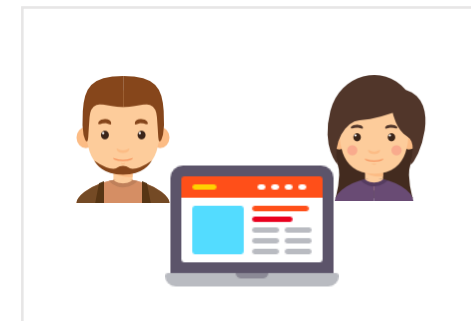
2. Individueller Geschmack ist nicht immer vorhersagbar

- Ein Nutzer mag vielleicht *Metallica*, *Guns N' Roses* und *Slayer*
Anders als andere Nutzer, mag dieser vielleicht aber auch *Taylor Swift*
- Dieses Problem kann evtl. durch mehr Input Daten gelöst werden



3. Ein Account kann von mehreren Personen genutzt werden

- Dadurch können Präferenzen nicht mehr einzelnen Nutzern zugeordnet werden



Recommendation Engine - Algorithmen

Für die Implementierung von Recommendation Engines gilt es zwei Probleme zu lösen:

Ähnlichkeiten finden



Gruppen bilden



Ähnlichkeiten bestimmen - Distanzmaße

Die zentrale Frage bei Recommendation Engines ist, wie man Ähnlichkeiten bestimmt

1. Jaccard Koeffizient:

Ähnlichkeit wird bestimmt, indem man die Anzahl der gemeinsamen Elemente (Anzahl der Nutzer, die Artikel A und B kauften) durch die Größe der Vereinigungsmenge (Anzahl der Nutzer, die Artikel A oder B kauften) teilt:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

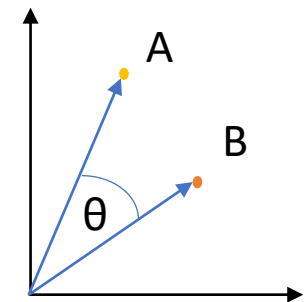
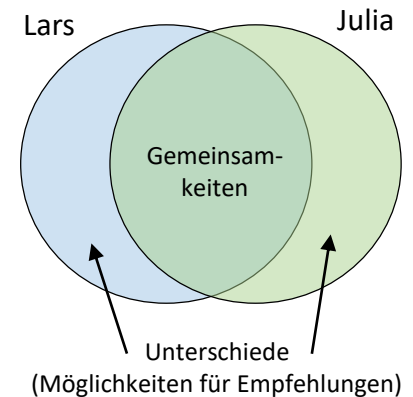
Benutzt man, wenn man keine numerischen Werte hat, sondern nur sagen kann ob der Artikel gekauft wurde

2. Kosinus Ähnlichkeit:

Ähnlichkeit bezeichnet hier den Kosinus des Winkels zwischen 2 Vektoren A und B:

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}$$

Je näher die Vektoren, desto kleiner der Winkel und desto größer der Kosinus





Übung 2

Sie haben einen Onlineshop für Zierfische. Daraus können Sie Daten ableiten, welche Fische besonders viel gekauft wurden und welche weniger. Welche Metrik sollten Sie anwenden, um die Ähnlichkeit zwischen 2 Kunden herauszufinden?

- a. Jaccard-Koeffizient
- b. Cosinus Ähnlichkeit



Übung 2

Sie haben einen Onlineshop für Zierfische. Daraus können Sie Daten ableiten, welche Fische besonders viel gekauft wurden und welche weniger. Welche Metrik sollten Sie anwenden, um die Ähnlichkeit zwischen 2 Kunden herauszufinden?

- a. Jaccard-Koeffizient
- b. Cosinus Ähnlichkeit

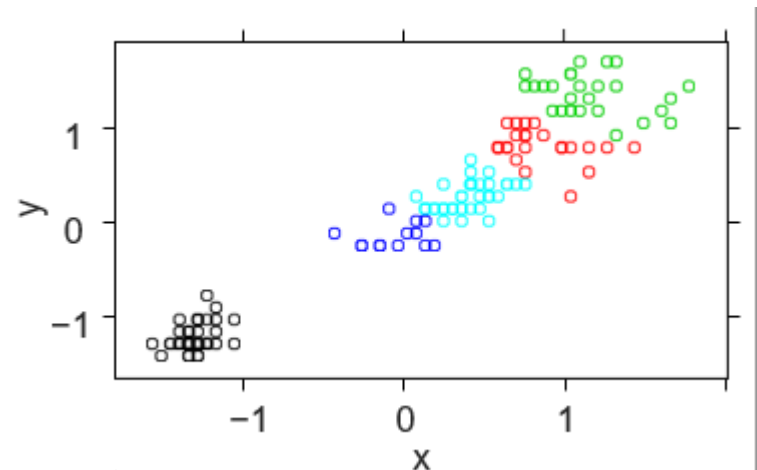
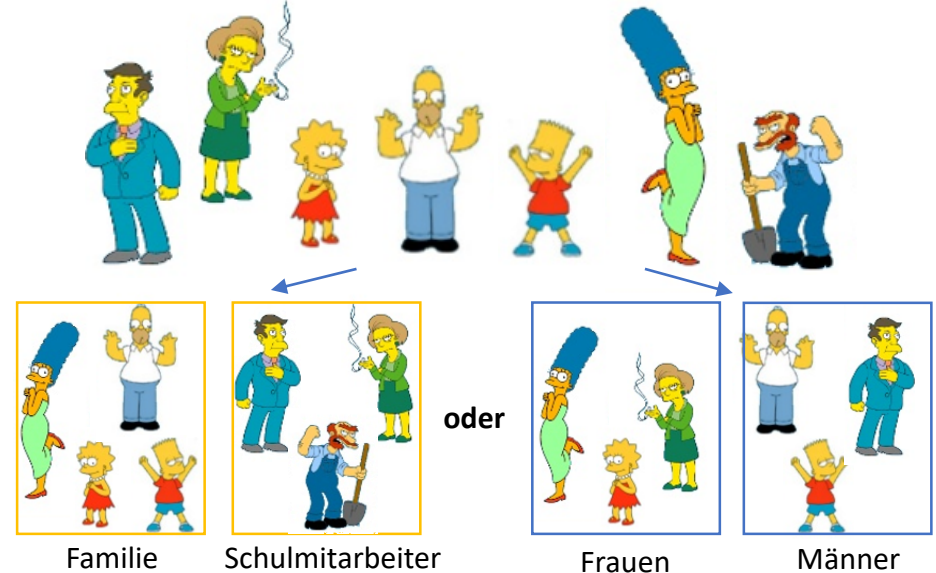
Richtig ist **a**.

Den Jaccard-Koeffizient benutzt man, wenn man keine numerischen Werte über Gefallen/Nicht-Gefallen (z.B. Sterne oder Punkte) zur Verfügung hat, sondern rein anhand der Schnittmengen Ähnlichkeiten feststellen möchte.

Gruppen bilden

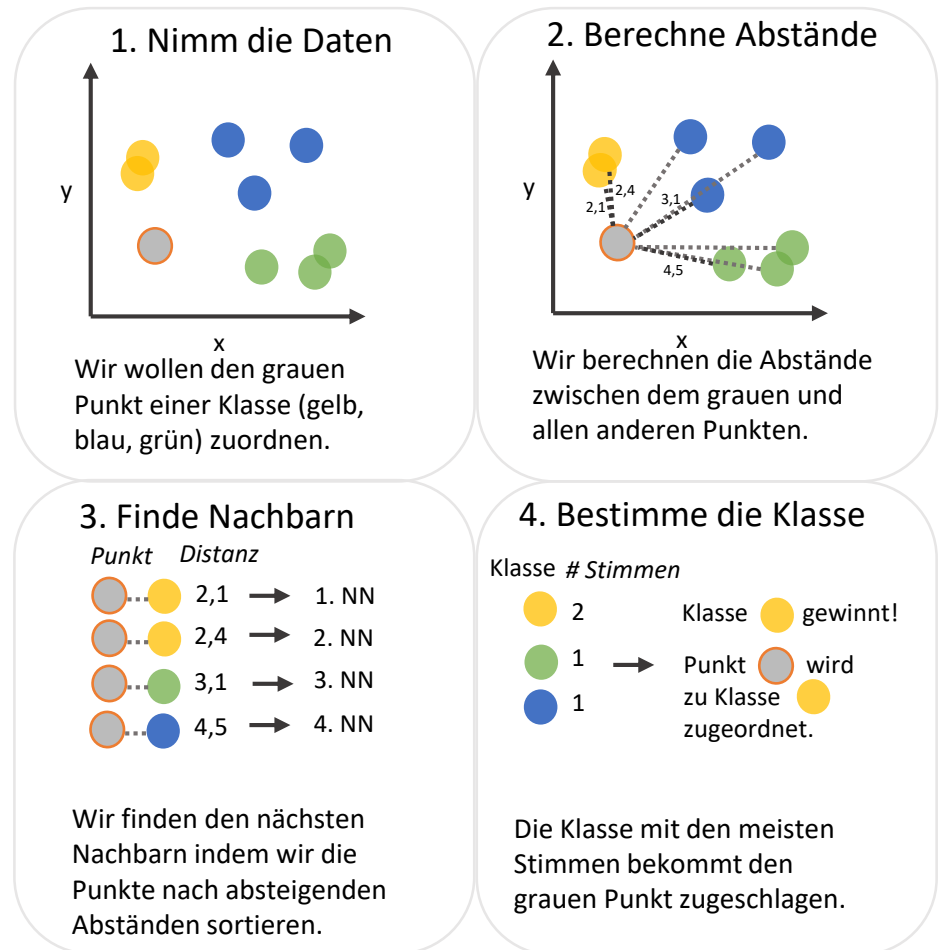
- **Klassifizierung** und **Clustering** möglich
- **Klassifizierung** sortiert Variablen in *bereits festgelegte* Klassen (pre-labeled data)
- **Clusterverfahren** werden benutzt, um Variablen oder Nutzer zu „natürlichen“ Gruppen (Clustern) zusammenzufassen (unlabeled data)
 - Idealerweise sind die Gruppen innerhalb der Cluster homogen, zwischen den Clustern heterogen
- Beispiele für Methoden:
 - **KNN (Klassifizierung)**
 - Kmeans (Clustering)
 - Matrix Factorization (Dimensionsreduktion)

Wie würden Sie hier gruppieren?



KNN (K-Nearest Neighbor)

- Klassifikationsverfahren unter Berücksichtigung der k -nächsten Nachbarn
- Sortiert die Werte nach den Distanzen zu den nächsten Nachbarn und ordnet die Klassen durch Mehrheitsentscheidung zu
- Basierend auf historischen Daten mit vorhandener Klassenzuordnung
- Kann mit verschiedenen Distanzmaßen verwendet werden
- k sollte normalerweise ungerade sein, um Patt Situationen zu verhindern





Ausblick: Erstellen einer eigenen einfachen Recommendation Engine mit python

- **Book-Crossings** ist ein Datenset erstellt von Cai-Nicolas Ziegler mit 1,1 Millionen Bewertungen von 270.000 Büchern von 90.000 Nutzern. Die Bewertungen gehen von 1 bis 10.



Hausaufgabe: Bitte laden Sie das Datenset, bestehend aus 3 Tabellen (Ratings, Books Info und Users Info) hier herunter:

<http://www2.informatik.uni-freiburg.de/~ciegler/BX/>

- Laden Sie die Daten in pandas:

```
In [8]: import pandas as pd
import numpy as np
from scipy.sparse import csr_matrix
import sklearn
from sklearn.decomposition import TruncatedSVD

book = pd.read_csv('BX-Books.csv', sep=';', error_bad_lines=False, encoding="latin-1")
book.columns = ['ISBN', 'bookTitle', 'bookAuthor', 'yearOfPublication', 'publisher', 'imageUrlS', 'imageUrlM', 'imageUrlL']
user = pd.read_csv('BX-Users.csv', sep=';', error_bad_lines=False, encoding="latin-1")
user.columns = ['userID', 'Location', 'Age']
rating = pd.read_csv('BX-Book-Ratings.csv', sep=';', error_bad_lines=False, encoding="latin-1")
rating.columns = ['userID', 'ISBN', 'bookRating']
```

- Benutzen Sie KNN aus dem Paket scikit-learn um Vorhersagen zu treffen :

```
In [36]: from sklearn.neighbors import NearestNeighbors

model_knn = NearestNeighbors(metric = 'cosine', algorithm = 'brute')
model_knn.fit(us_canada_user_rating_matrix)

Out[36]: NearestNeighbors(algorithm='brute', leaf_size=30, metric='cosine',
metric_params=None, n_jobs=1, n_neighbors=5, p=2, radius=1.0)
```

- Das genaue Vorgehen schauen wir uns das nächste Mal im Detail an 😊