ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

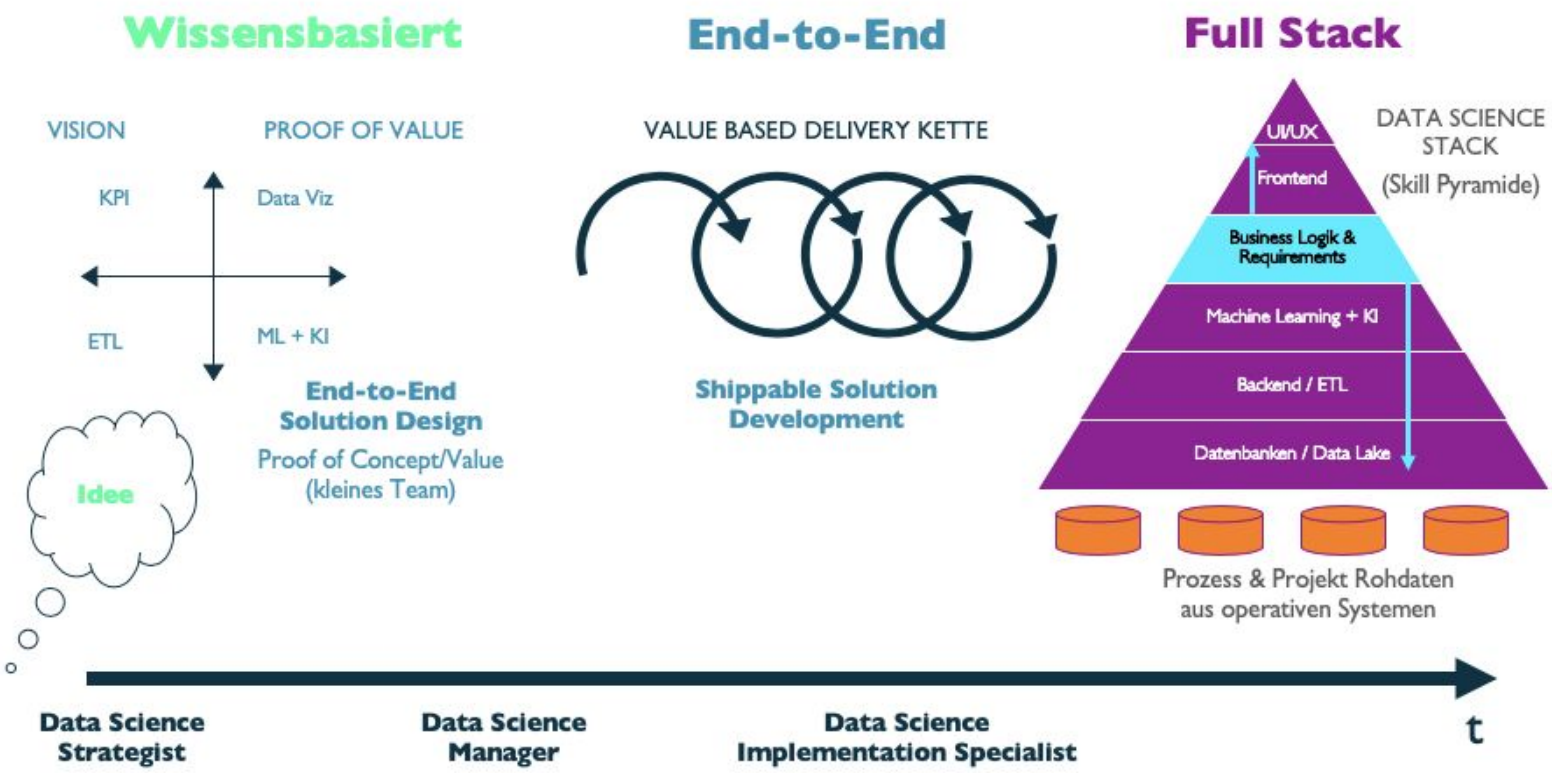
Recommender System

Prototyp 1.0 Naive Bayes
Open Data Day 2019

www.alessio-analytics.com
info@alessio-analytics.com

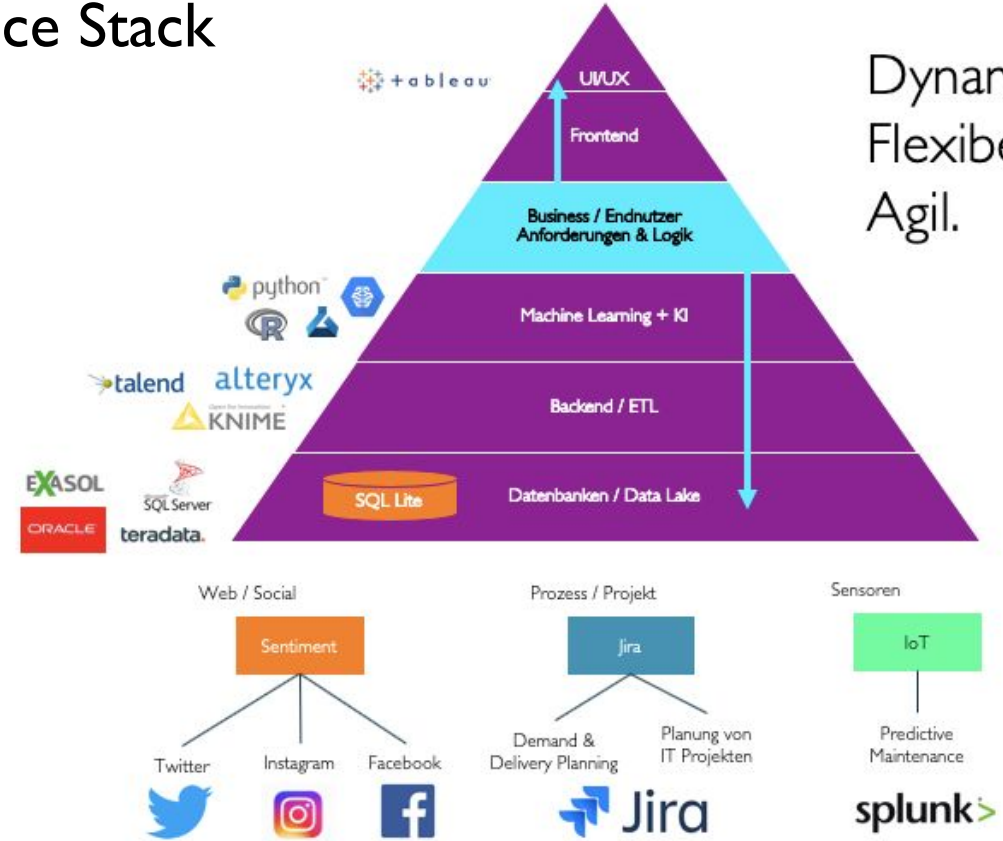
Fullstack Data Science = Full Stack Mehrwert

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS



Alessio Analytics
Data Science Stack

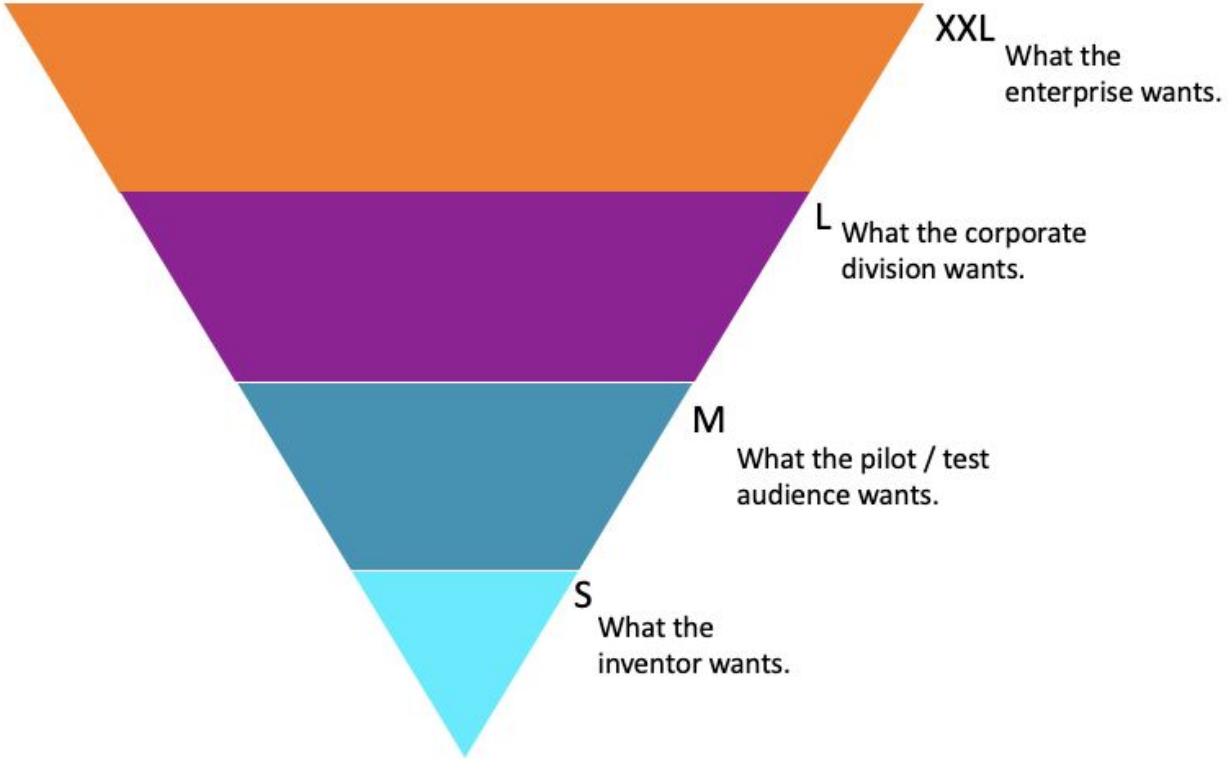
ALESSIO ANALYTICS
FROM DATA TO INSIGHTS



Dynamisch,
Flexibel,
Agil.

Alessio Analytics
Value Stack

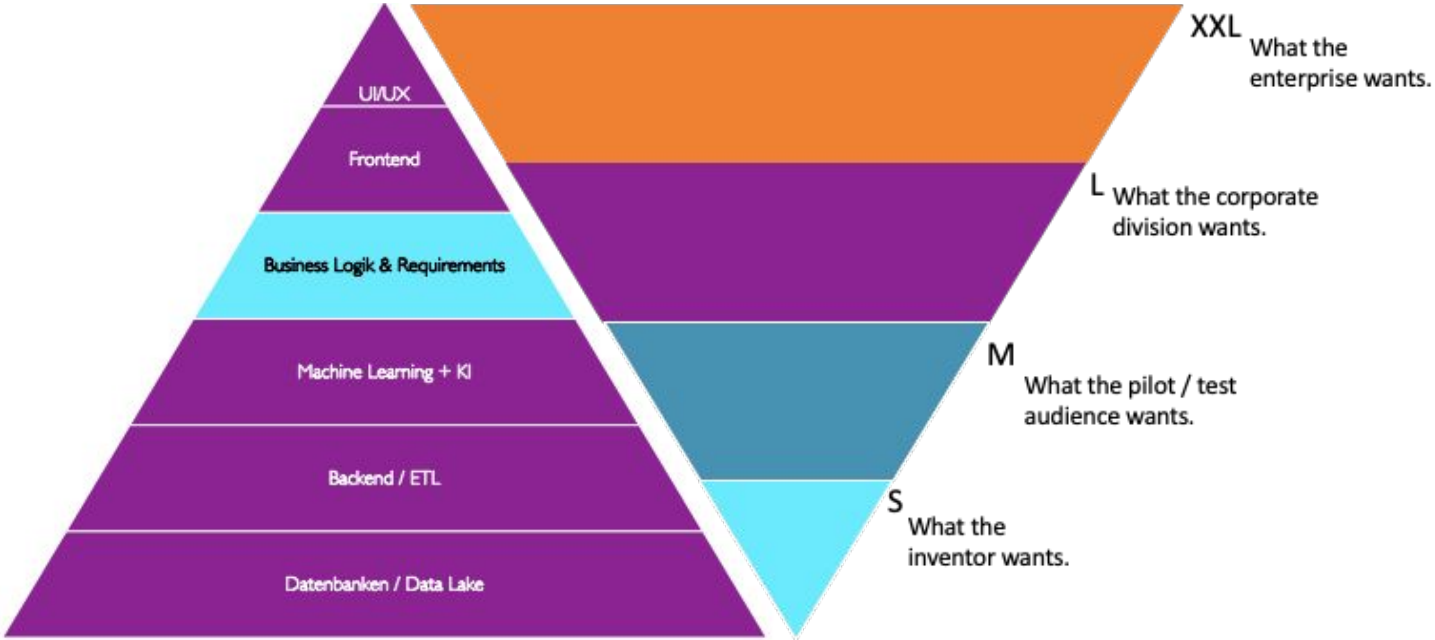
ALESSIO ANALYTICS
FROM DATA TO INSIGHTS



Alessio Analytics
Digital Value Model

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

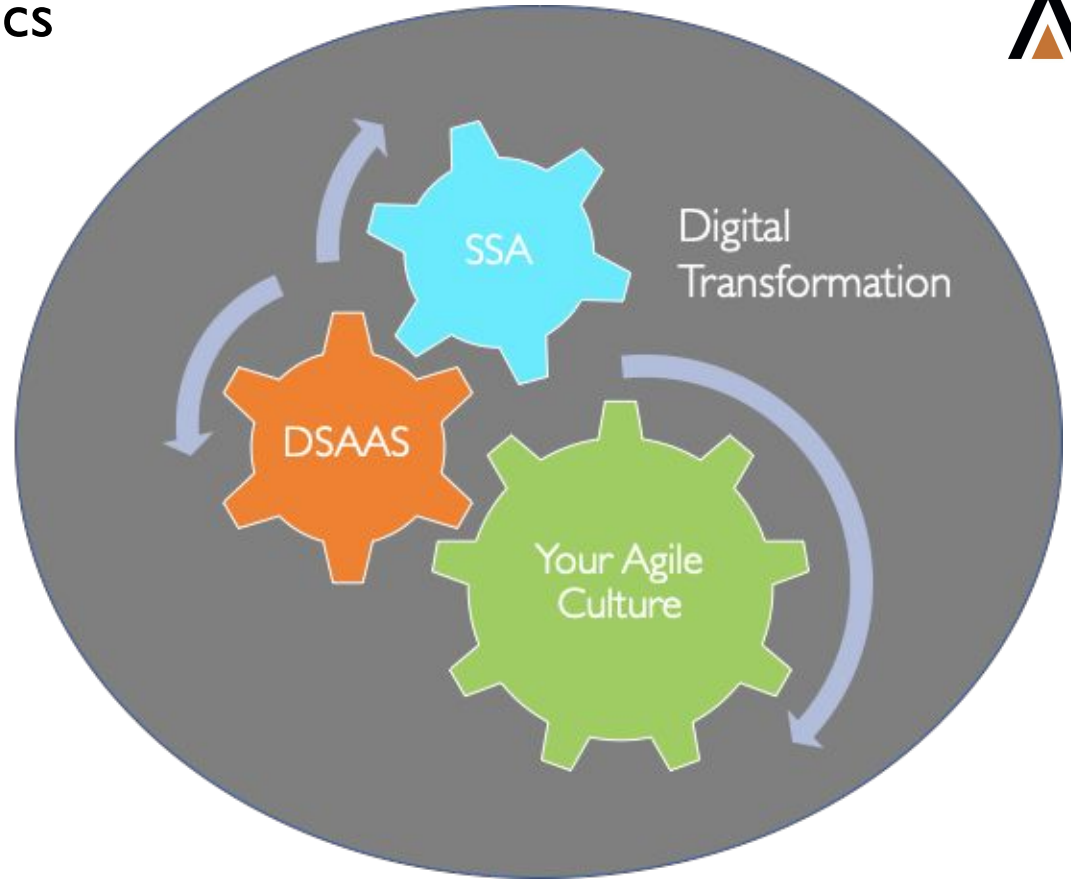
Alessio Analytics
Value Stack



Alessio Analytics
Data Science Stack

Alessio Analytics Service Model

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS



ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

NETFLIX

amazon

zalando

Theorie

Naive Bayes

Ausgangslage:

	🌭	🍷	🥨	🍬	🥥
👩🏿‍🦱	0	1	1	0	0
🧑🏼‍🦳	0	1	0	1	0
👩🏼‍🦳	0	0	1	1	1
🧔🏻	1	0	0	1	0

Gewünschtes Ergebnis

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

	🌭	🍷	🥨	🍬	🥥
👩🏿‍🦱	0,32	0,43	0,30	0,82	0,23
👩🏼‍🦳	0,15	0,60	0,53	0,62	0,23
👩🏼‍🦰	0,28	0,53	0,64	0,77	0,28
🧔🏿	0,26	0,50	0,47	0,69	0,17

Umsetzung 1

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

	🌭	🍷	🥨	🍬	🥥
👩🏿‍🦱	0,32	0,43	0,30	0,82	0,23
👱🏼‍♀️	0,15	0,60	0,53	0,62	0,23
👩🏼‍🦳	0,28	0,53	0,64	0,77	0,28
🧔🏿	0,26	0,50	0,47	0,69	0,17

	🌭	🍷	🥨	🍬	🥥
🥥 👱🏼‍♀️	0	1	0	1	0

Umsetzung 2

	🌭	🍷	🥨	🍬	🥥
👩🏿‍🦱	0,32	0,43	0,30	0,82	0,23
👱🏼‍♀️	0,15	0,60	0,53	0,62	0,23
👱🏻‍♀️	0,28	0,53	0,64	0,77	0,28
🧔🏿	0,26	0,50	0,47	0,69	0,17

	🌭	🍷	🥨	🍬	🥥
🥥👱🏼‍♀️	0	1	0	1	0

P(🥥 | 🍷 & 🍬)

Bayes Theorem

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

P(🥥| 🍷 & 🍬) =

P(🍷 & 🍬 | 🥥)*P(🥥)/P(🍷 & 🍬)

Bayes Theorem

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

P(🥥| 🍷 & 🍬) =

P(🍷 & 🍬 | 🥥)*P(🥥)/P(🍷 & 🍬)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Vorgehen

	🌭	🍷	🥨	🍬	🥥
👩🏾	0	1	1	0	0
👱🏼‍♀️	0	1	0	1	0
👩🏼‍🦳	0	0	1	1	1
🧔🏿	1	0	0	1	0

P(🥥| 🍷 & 🍬) =

P(🍷 & 🍬 | 🥥)*P(🥥)/P(🍷 & 🍬)

Vorgehen

	🌭	🍷	🥨	🍬	🥥
👩🏾	0	1	1	0	0
👱🏻‍♀️	0	1	0	1	0
👩🏼‍🦳	0	0	1	1	1
🧔🏿	1	0	0	1	0

P(🥥| 🍷 & 🍬) =

P(🍷 & 🍬 |🥥)*P(🥥)/P(🍷 & 🍬)

Ergibt sich aus absoluten Häufigkeiten

Vorgehen

	🌭	🍷	🥨	🍬	🥥
👩🏾	0	1	1	0	0
👨🏼	0	1	0	1	0
👱🏻‍♀️	0	0	1	1	1
🧔🏻	1	0	0	1	0

P(🥥| 🍷 & 🍬) =

P(🍷 & 🍬 |🥥)*P(🥥)/P(🍷 & 🍬)

Ergibt sich aus absoluten Häufigkeiten

Eine von 4 Personen hat eine Kokosnuss gekauft, also beträgt die Wahrscheinlichkeit für den Kauf bei ¼ bzw 25%

Warum ist Bayes Naiv?

Bayes Theorem

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

P(🥥 | 🍷 & 🍬) =

P(🍷 & 🍬 | 🥥)*P(🥥)/P(🍷 & 🍬)

P(🍷 & 🍬) kann auf unterschiedliche Art und Weise berechnet werden.
Unter der Annahme der Unabhängigkeit von 🍷 und 🍬 wird P(🍷 & 🍬) wie folgt berechnet

P(🍷 & 🍬) = P(🍷) * P(🍬)

Unabhängigkeit

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

P(🍷 & 🍬) = P(🍷) * P(🍬)

Beispiel Korrelation

	🍬	📒	📝
👩🏾‍🦱	0	1	1
👨🏼‍🦲	1	1	1
👩🏼‍🦳	1	0	1
👨🏾‍🦲	1	0	0

Beispiel Korrelation

	🍬	📒	✏️
👩🏾‍🦱	0	1	1
👨🏼	1	1	1
👩🏼‍🦳	1	0	1
🧔🏻	1	0	0

Es besteht eine Korrelation zwischen

den Käufen von 📒 und ✏️

Vergleich der Ergebnisse

	🍬	📒	✏️
👩🏾‍🦱	0	1	1
👱🏼‍♀️	1	1	1
👧🏼	1	0	1
🧔🏿	1	0	0

Empirische Wahrscheinlichkeit:

P(📒 & ✏️) = 0.5

Naive Bayes:

P(📒 & ✏️) = P(📒) * P(✏️) = 0.5 * 0.75 = 0.375

Analog für P(🍷&🍬|🥥)

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

P(🍷&🍬|🥥)=P(🍷|🥥)*P(🍬|🥥)

Resumee Unabhängigkeit

Die annahme der Unabhängigkeit von Features bei dem Naive Bayes Algorithmus:

- verbessert die Laufzeit drastisch
- führt jedoch zu Ungenauigkeiten.

Resumee Naive Bayes Theorie

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

- ➢ Ausgangslage des Naive Bayes Algorithmuses ist die Kaufhistorie vieler verschiedener Kunden
- ➢ Es ist möglich jedem Kunden-Produkt-Paar eine Kaufwahrscheinlichkeit zuzuordnen
- ➢ Die Wahrscheinlichkeiten ergeben sich aus den absoluten Häufigkeiten in der Datenmenge.
- ➢ Annahmen und Randbedingungen beeinflussen die Laufzeit

➔ in der Praxis werden vorgefertigte Pakete (wie z.B. von Sklearn) verwendet

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

Praxis

Real World Implementation in Python3
mit Sklearn, Pandas und Numpy

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

Für Hilfe bei der Implementierung eines ML-basierten Recommender Systems wende Dich an:

info@alessio-analytics.com

Resumee Naive Bayes & Superstore Dataset

ALESSIO ANALYTICS
FROM DATA TO INSIGHTS

- ➢ Ziel, jedem Kunden-Produkt-Paar eine Kaufwahrscheinlichkeit zuzuordnen, ist erfüllt
- ➢ Python Pakete machen die Implementierung relativ simpel
- ➢ Die **Modellevaluation** zeigt, dass das **Superstore Dataset nicht ausreicht** für ein performantes Recommender System auf Basis von Naive Bayes:
  - in der Regel stehen **Sales-Daten in großer Menge (100k+ Zeilen)** zur Verfügung
  - in Real-World-Szenarien sind somit **bessere Ergebnisse zu erwarten**

Fazit:

- ➔ **Das für den Prototyp 1.0 entstandene Framework ist für den Einsatz verschiedener Machine-Learning Verfahren und Algorithmen geeignet.**
- ➔ **Tabellarische Datasets sind für den “Plug-And-Play” Einsatz geeignet.**