DATA SCIENCE CONSULTING
Session 5

March 6th, 2023

AGENDA

1. Agile Methodology

2. Topic Extraction

a. LSI/LSA review

b. LDA

3. Sentiment analysis

a. Rule-based methods

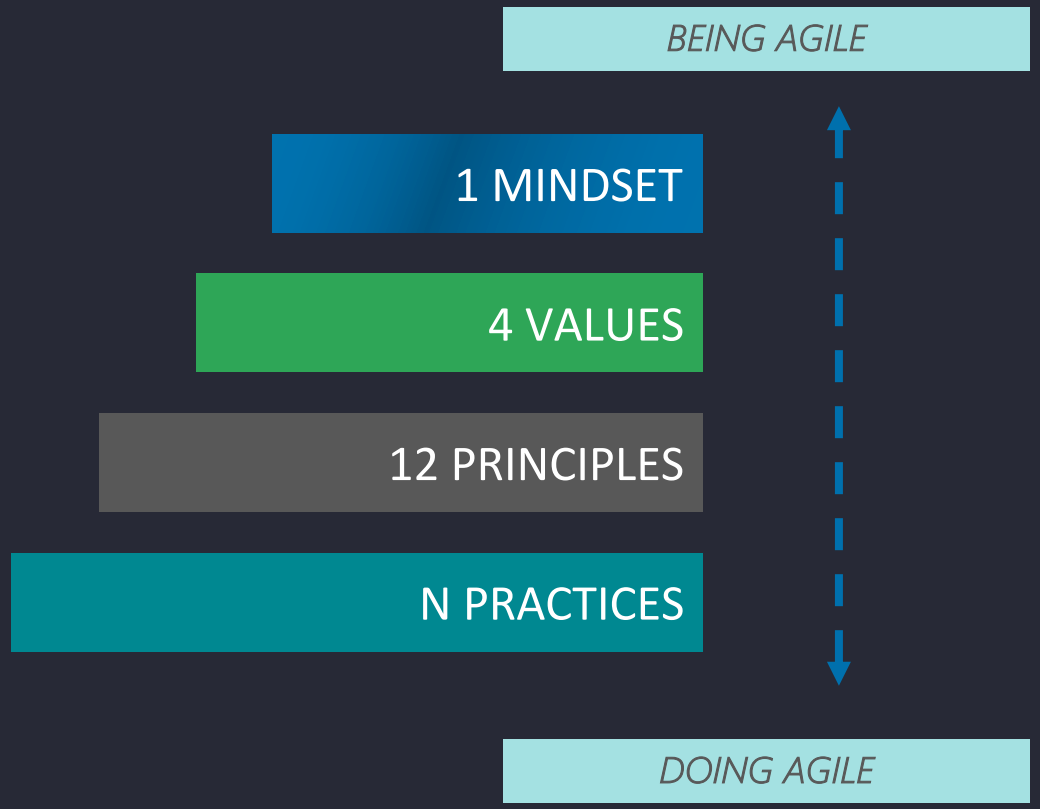
b. Learning based methods

TO BE COMPETING IN THIS DIGITAL ERA, WE NEED AGILE* …

But … what is Agile?

Agile refers to the MINDSET & BEHAVIOUR that supports an iterative & incremental approach to manage changes in design, build, deployment and adoption of products in a highly flexible and interactive manner.

Typically, it involves self-governing, cross-functional teams working on the product.



AGILE METHODOLOGY IS THE CURRENT PRODUCTION SYSTEM PARADIGM

(ESPECIALLY FOR IT/DATA PROJECTS)

Mass production

1908 — **Taylorism**

1911 — **Fordism**

- Control of function
- Automation
- Commoditization

Mass customization

1962 — **Toyotism**

1990 — **Lean Manufacturing**

- Control of information
- Scalable technology & communication
- Just-in-time production

Mass collaboration

2001 — **Agile Manifesto**

Today — **Agile@Scale** (Google, Yahoo, Facebook, etc.)

- Widespread social computing
- Iterative creation of value
- Network collaboration

AGILE MAIN BENEFITS IN CONSULTING

1. Arbitrate on value

*Agile teams arbitrate on **value** rather than cost*

2. Deliver business value faster

*Agile aims at **delivering business value as fast as possible** with quality minimum requirements*

3. Focus on products

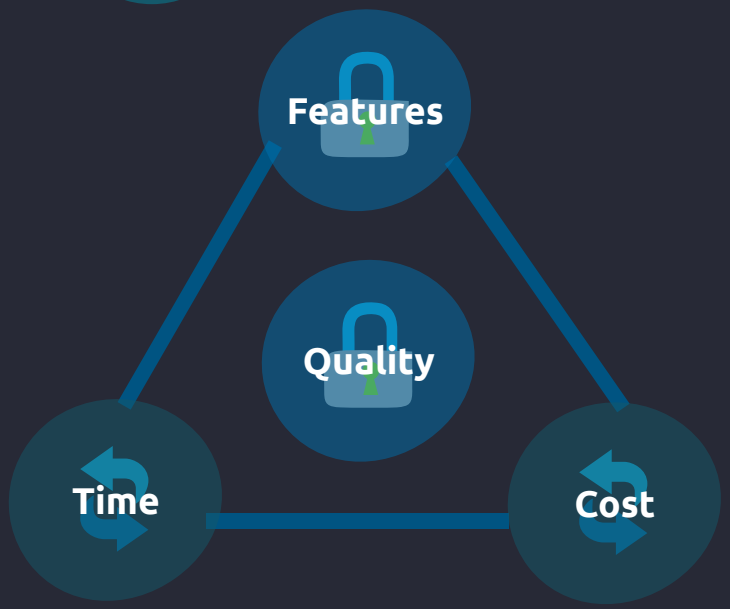
*Shifting from a project to a **product-based approach**, necessary for adaptability and fast value delivery*

AGILE MAIN BENEFITS IN CONSULTING
1/3 COMPARING AGILE VS WATERFALL

Legend: fixed, adjustable

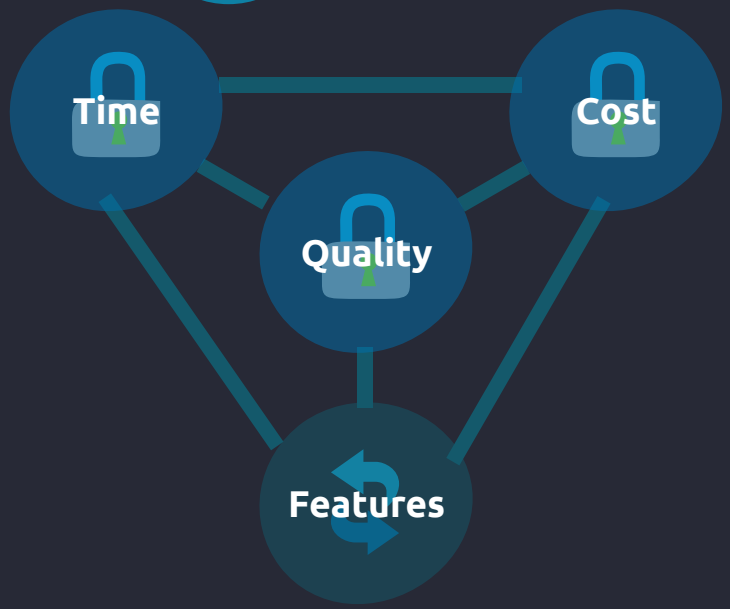
A paradigm shift is happening, from a cost to a value-based approach

WATERFALL



In this traditional approach, the scope is set without collaboration. Costs/times must remain adjustable to deliver the entire scope. Quality is often adjusted in this approach.

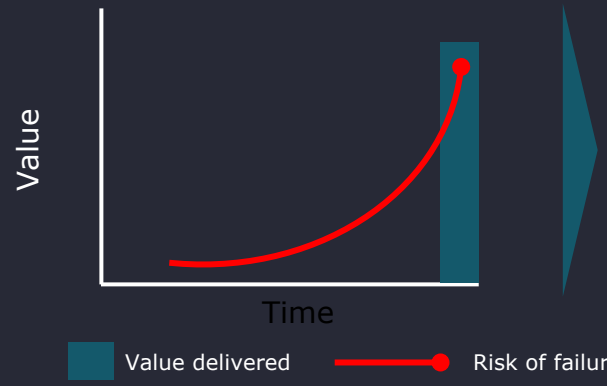
AGILE



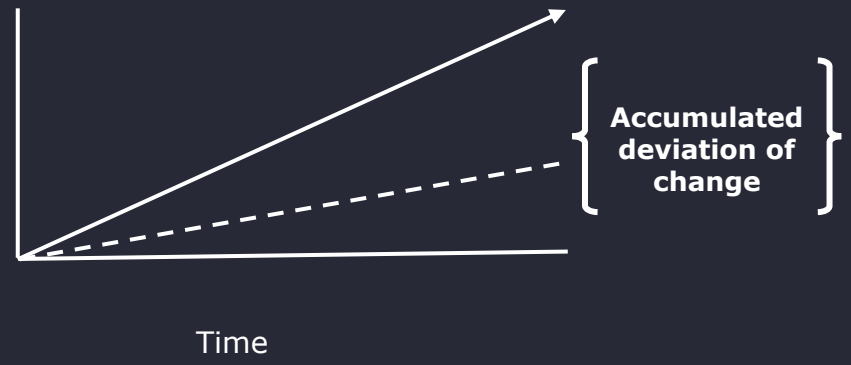
In the Agile approach, the scope is not fixed but prioritized in order to deliver the most value. Costs/Time are fixed given the short cycle operation with a given number of resources. Quality is often improved in this approach

AGILE MAIN BENEFITS IN CONSULTING
2/3 DELIVER VALUE EARLIER AND AT LESS OVERALL RISK

Waterfall: deliver the whole product at once

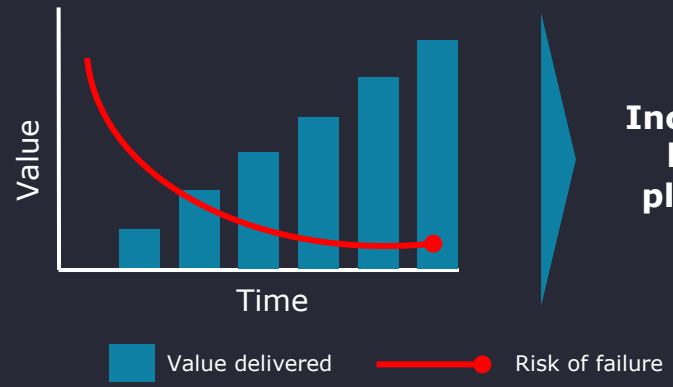


One-time planning



The **longer the time** from planning to value delivery (release) **the higher the risk of failing** to meet customer expectations

Agile: deliver increment by increment



Increment based planning

The **shorter the time** from planning to value delivery, **the lower the risk of failing** to meet customer expectations over time

AGILE MAIN BENEFITS IN CONSULTING
3/3 FOCUS ON PRODUCTS

Agile enables shifting from a project to a product-based approach, necessary for adaptability and fast value delivery

A product-based approach…

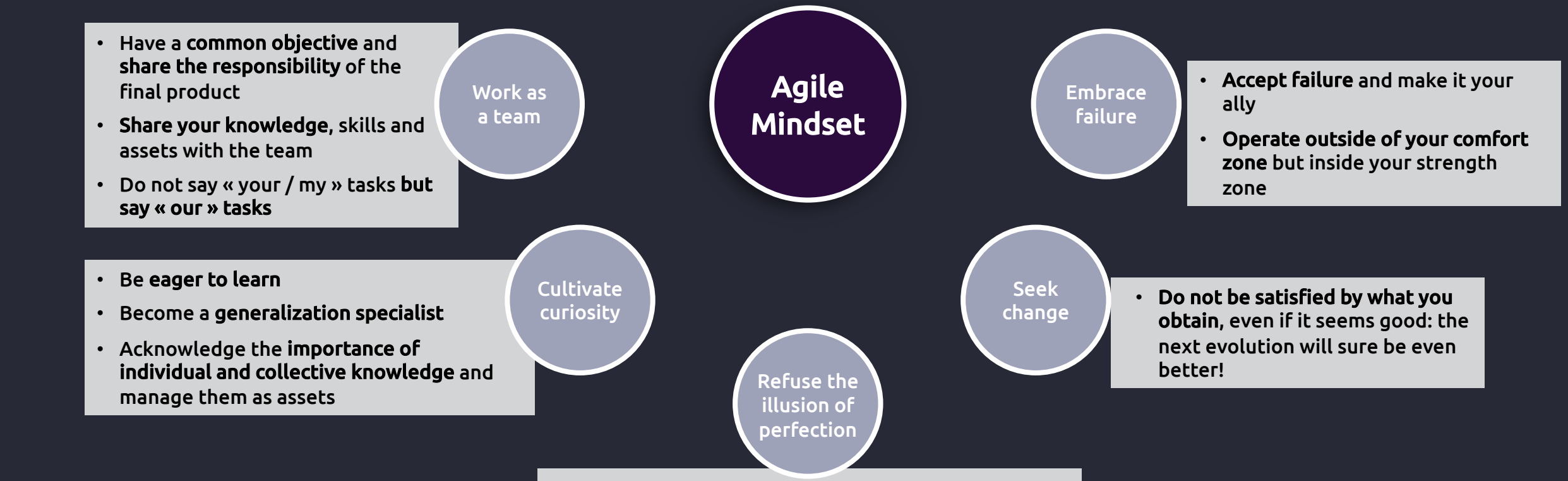
… gives **responsibility to the business**

… enables **continuous improvement & adapts to business needs**

… provides **concrete business value with every IT development**

… enables an **organizational structure** for **faster value delivery**

BEING AGILE = HAVING AN AGILE MINDSET



- Have a **common objective** and **share the responsibility** of the final product
- **Share your knowledge**, skills and assets with the team
- Do not say « your / my » tasks **but say « our » tasks**

- **Accept failure** and make it your ally
- **Operate outside of your comfort zone** but inside your strength zone

- Be **eager to learn**
- Become a **generalization specialist**
- Acknowledge the **importance of individual and collective knowledge** and manage them as assets

- **Do not be satisfied by what you obtain**, even if it seems good: the next evolution will sure be even better!

- **Perfectionism is an impediment** to innovation
- Agile computing is **based on iterative development** which enables the search for solutions thanks to collaboration
- **Teams are cross-functional** and self-managed

THE AGILE MANIFESTO (2001) LIES ON FOUR PRINCIPLES

An organization or a project that is based on the Agile Manifesto must always value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

AGILE LANDSCAPE – THE TUBE MAP

Structure

Culture

Portfolio

Project

Day-to-day

THE “AGILE MANIFESTO” OUTLINED SCRUM BY DEFINING CONCRETE ELEMENTS AND A MINDSET OF CONTINUOUS IMPROVEMENT

VALUES AND PRINCIPLES

SCRUM

ROLES

- There are three key roles within the Scrum framework
- Each role with a defined set of responsibilities
- Only through collaboration across the roles can an agile engagement be successful

CEREMONIES

- Regular collaborative meetings or "ceremonies" are an important part of agile development
- They are clearly time-boxed and help the team to drive the required transparency for success

ARTIFACTS

- In Scrum, artefacts are “information radiators” and serve to capture the shared understanding of the team at particular points in time. They are at the number of three :
  - Sprint Backlog
  - Product Backlog
  - Product Increment

FUNDAMENTALLY AN AGILE (SCRUM) TEAM CONSISTS OF A PRODUCT OWNER, SCRUM MASTER AND TEAM MEMBERS

ROLES

PRODUCT OWNER

Takes inputs of what the product should be, translating into a product vision and the product backlog

| What is being made

A leader with vision, authority, and availability who decides the focus area for the team based on deep knowledge of the risks and rewards

TEAM

Develops the product envisioned by the Product Owner. Often a wide mix of skills and roles!

| By whom it is made

A small, autonomous, cross-functional team with a strong sense of purpose and in constant communication with each other

SCRUM MASTER

Does whatever it takes to make the team successful! Facilitates, protects the team and removes blockers

| How it is made

Half coach, and half team facilitator, a servant leader who serves the team by focusing on process and removing obstacles

COLLECTIVELY, THE ELEMENTS OF A SPRINT ARE DESIGNED TO ENABLE SCRUM TEAMS TO BE HIGHLY FLEXIBLE AND RESPONSIVE TO CHANGE

EVENTS

1 *Sprint Planning*

Team commits to an achievable goal for the sprint and selects stories that will make it happen

2 *Daily Scrum / Stand Up*

Team meets for up to 15 minutes to share progress, answer questions and present any roadblocks

3 *Sprint Review (Demo)*

Team demos a “potentially shippable” product generated from the sprint to any interested stakeholders

4 *Retrospective*

Team looks back at the process used last sprint, discusses what went well & what didn’t, as informed by reporting, decides on a single process improvement to pursue next sprint

= Product Owner

= Team

= Scrum Master

Sprint Length

The core of Scrum - a fixed length of time that lasts no more than a month and doesn’t vary over the life of a project. It should include periodic reporting of progress (Between 1 and 4 weeks)

SCRUM TEAMS WORK IN A SPRINTLY CADENCE, PUNCTUATED BY STAKEHOLDER FEEDBACK THAT INFORMS PRIORITIES FOR FUTURE SPRINTS

EVENTS

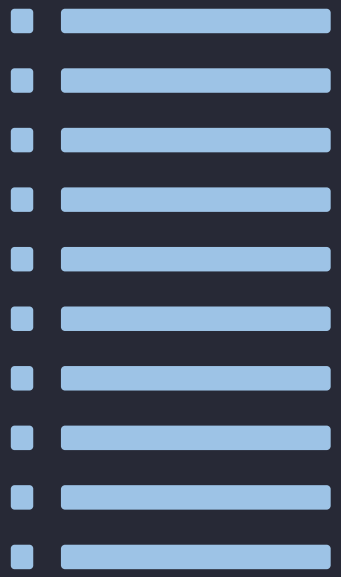
Key elements

- Focused bursts of activity (1-4 week sprints)
- Each sprint delivers something valuable
- Frequent feedback to refine work items
- Retrospective after every sprint

ARTIFACTS TO DETAIL PRODUCT BEING DEVELOPED, ACTIONS TO PRODUCE IT, & ACTIONS PERFORMED DURING THE PROJECT

ARTIFACTS

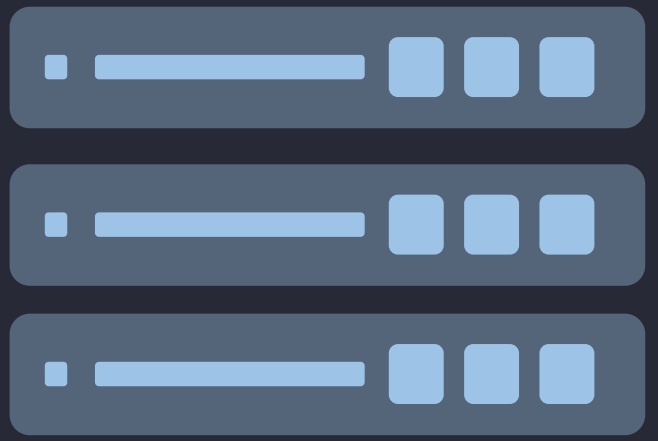
PRODUCT BACKLOG



What to do

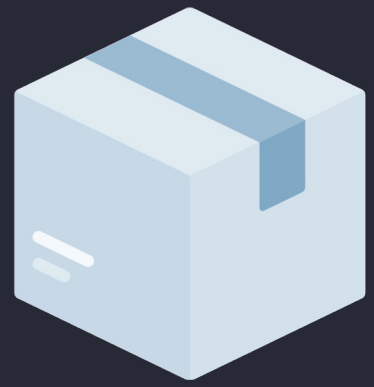
DEFINITION OF READY

SPRINT BACKLOG



What to take on now & how

PRODUCT INCREMENT



What gets done

DEFINITION OF DONE

ANIMAGILE

We use story points (with modified Fibonacci model) to calculate agile team capacity for every team

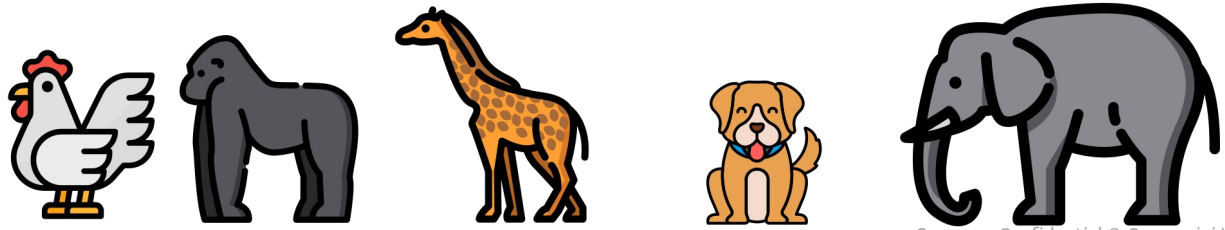
- A Story point is a singular number that represents:
  - **Volume :** How much is there?
  - **Complexity :** How hard is it?
  - **Knowledge :** Do we have all competencies to build the item
  - **Uncertainty :** What's not known?
- Story Points are relatives. They are not connected to any specific unit of measure.
- Compared with other Stories, an 8-point Story should take relatively four times longer than a 2-point Story.

Exercise :

Use estimating poker to relatively estimate the mass of a set of animals.

- **Step 1 :** In groups, identify the smallest animal and mark it as **1**
- **Step 2 :** Estimate the remaining animals using values **1, 2, 3, 5, 8, 13, 20, 40, 100**

1	2	3	5	8	13	20	40	100



PREPARE

HOW SCRUM, DATA SCIENCE & CONSULTING WORK TOGETHER?

The Scrum methodology has been originally created for **software development** made by **internal teams**

Data Science

Data Science is **closer to Research & Development** than software development

It is **difficult to predict the workload and the tasks** to deliver the user story

Conduct PoC before launching MVP developments and foresee potential difficulties

Consulting

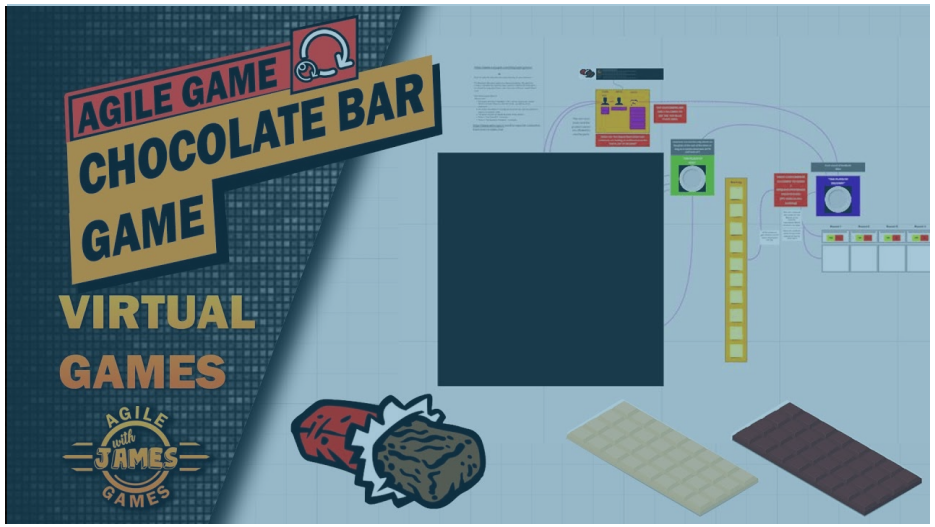
Consulting remains a premium service **based on commitment**

Flexibility allowed by the methodology can be **difficult to apply** with a client

Awareness to the client's team and If possible, **integrate client developers into the delivery team** to make it realize the work done

CHOCOLATE BAR GAME

Playing Time	Players Required	Objectives
5 mins	4+	The goal is to create a chocolate bar as if you were taking instructions from the product owner. Development teams choose their product manager who can also be the product owner.



1. PO : instruct the team members to create a chocolate bar (different components possible)
2. After each iteration the project manager provides the team with customer feedback. Customers give 👍 or 👎
3. Teamwork involves recording customers' responses for changes before the next iteration
4. Iterate on adding or removing ingredients from the chocolate bar

AGENDA

1. **Agile Methodology**

2. **Topic Extraction**

    a. LSI/LSA review

    b. LDA

3. **Sentiment analysis**

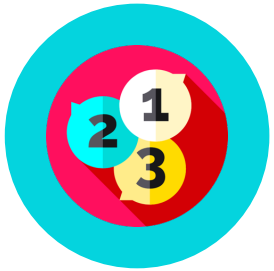
    a. Rule-based methods

    b. Learning based methods

Data pipeline

Data Collection

Data Cleaning



Word Embedding

Topic Extraction

Sentiment Analysis

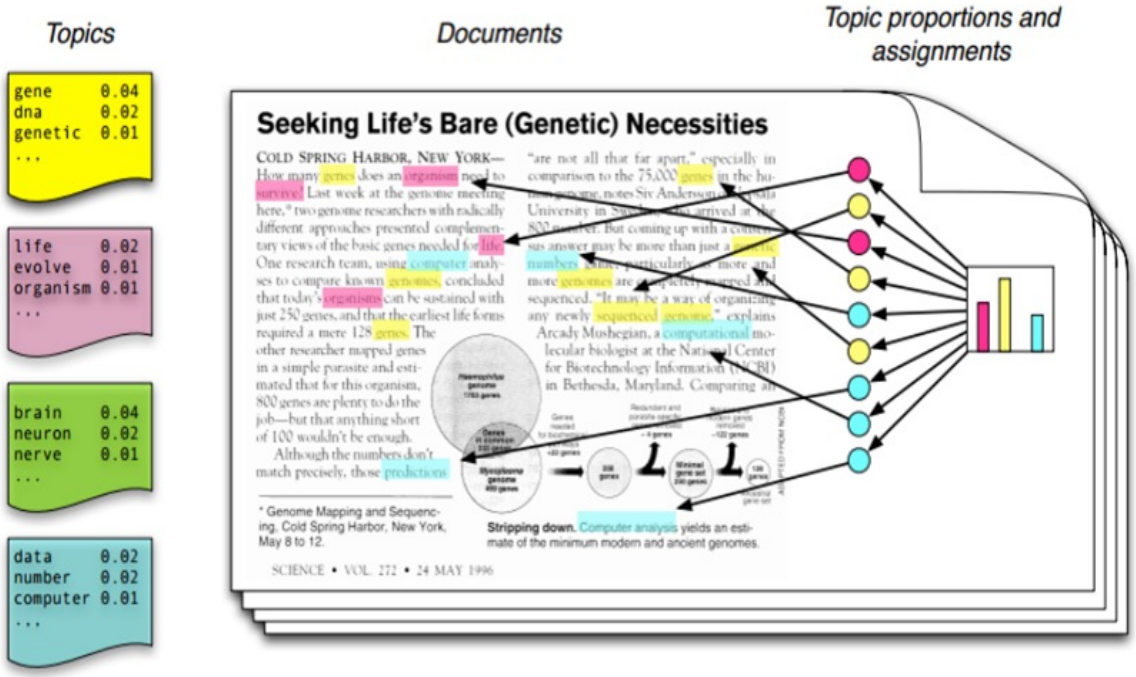
Topic extraction: Introduction

Context and objectives:

1. Companies now possess a lot of text data such as mails, documents and process information
2. Find hidden topics among a corpus of documents

Application:

- *Email classification by attributing a category to any incoming mail*
- Identify main topics of discussion on social media
- Find emerging topics across client feedbacks on a certain product



Agenda

1. Agile Methodology
2. Topic Extraction
    a. LSI/LSA review
    b. LDA
3. Sentiment analysis
    a. Rule-based methods
    b. Learning-based methods

Latent Semantic Analysis (1/4)

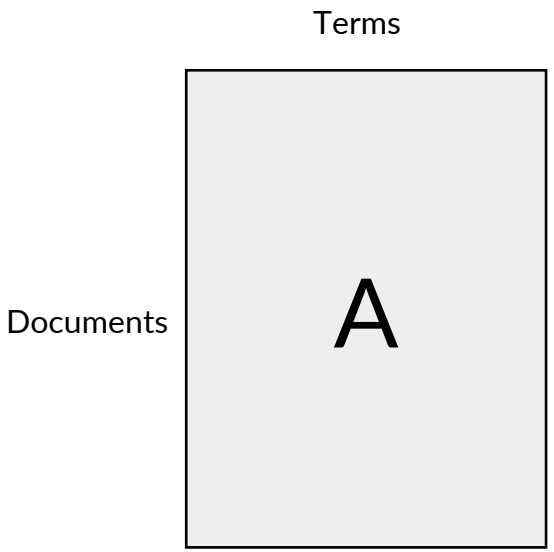
Latent Semantic Analysis or Indexing (LSA or LSI) is a technique for creating a **vector representation** of a document. Having a vector representation of a document gives you a way to compare documents for their similarity by calculating the **distance between the vectors**.

Two main steps:

- Term-Document matrix (A), using for example TF-IDF.
- Singular Value Decomposition (SVD) of A.

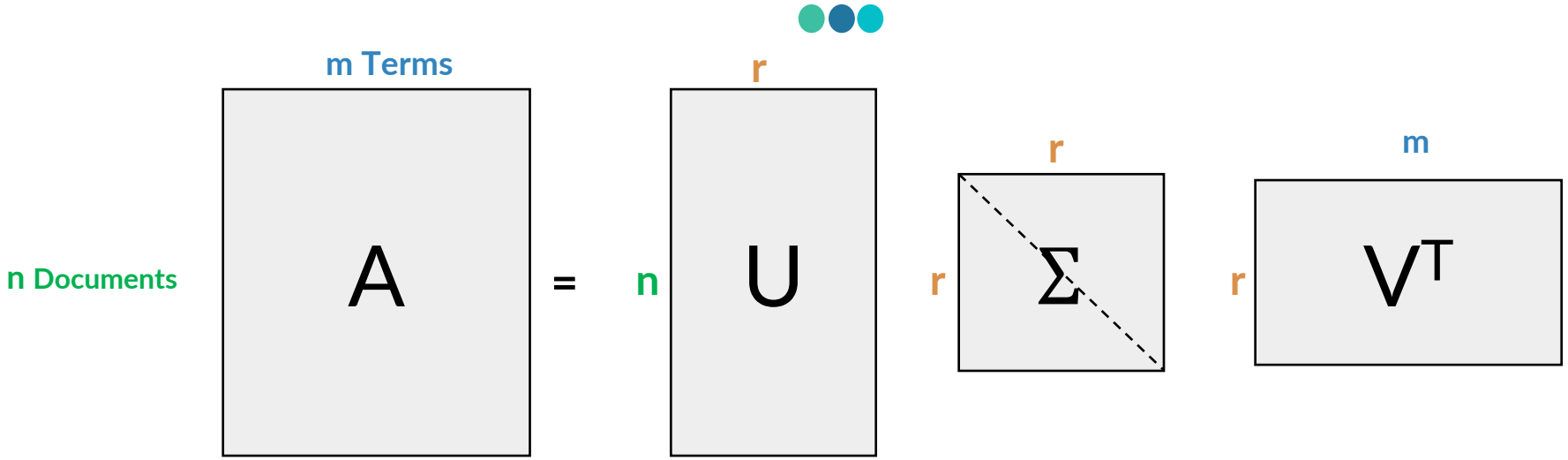
Latent Semantic Analysis (2/4)

Let's start with A, the TF-IDF matrix of our corpus



- The matrix A^TA contains all the correlations between terms over the set of documents
- The matrix AA^T contains all the correlations between documents over the terms

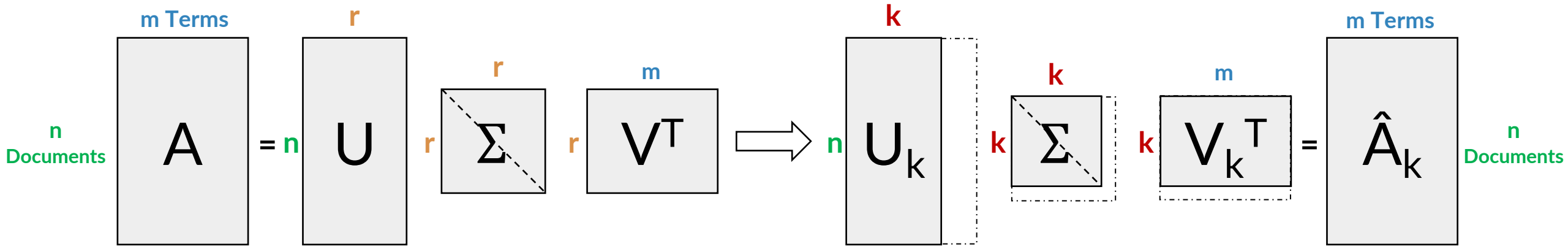
SVD details



$$A_{[n \times m]} = U_{[n \times r]} \Sigma_{[r \times r]} (V_{[m \times r]})^T$$

In this decomposition we see that documents are constituted of topics and topics are made of words

Latent Semantic Analysis (3/4)



We perform a SVD
U and V are orthogonal matrices
Σ is a diagonal matrix

We only want to keep the k most relevant topics. Choosing k depends on topic importance, size of A, interpretability …

We obtain a topic distribution in the documents and topic composition in term of words

Latent Semantic Analysis (4/4)

Key take away

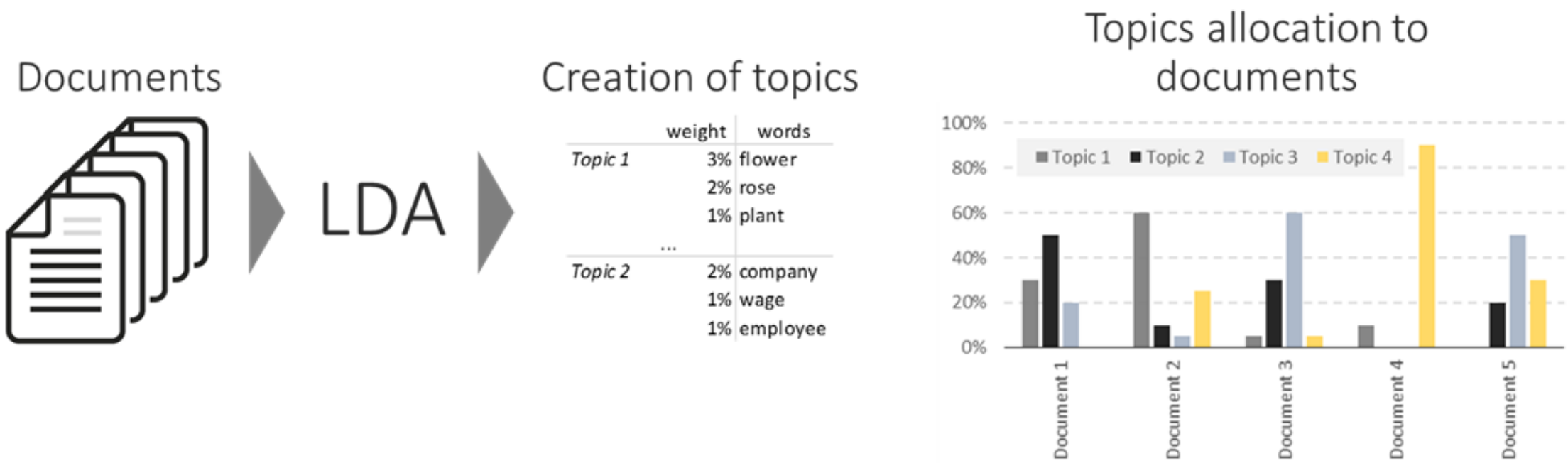
- Gives you a first approach to topic extraction
- Topics require human interpretation and validation
- LSA follows a principle of topic-based documents and word-based topics

Agenda

1. Agile Methodology
2. Topic Extraction
    a. LSI/LSA review
    b. LDA
3. Sentiment analysis
    a. Rule-based methods
    b. Learning based methods

Topic extraction: LDA

- LDA is an unsupervised generative model, that takes documents as input, apply posterior inference to find topics.
- The model determines in what percentage each document talks about each topic
- A topic is represented as a probability distribution of words , for example:



Topic extraction: LDA

LDA: Some definition

Latent Dirichlet Allocation

This refers to everything that we don't know *a priori* and is hidden in the data. Here, the topics are hidden in the documents, and we need to find them

It is a "distribution of distributions". Here, in the context of topic modeling, the Dirichlet is the distribution of topics in documents and distribution of words in the topic.

Once we have Dirichlet, we will allocate topics to the documents and words of the document to topics.

$$Dir(\theta, \alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{n} \theta_k^{\alpha_{k-1}}$$

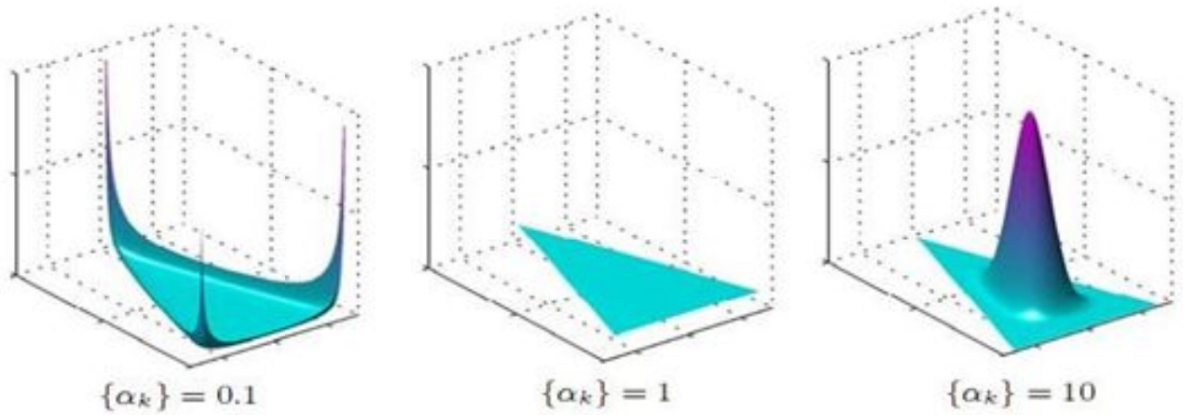
$$\begin{cases} \sum_{k=1}^{n} \theta_k = 1 \\ \forall i, \theta_i > 0, \alpha_i > 0 \\ \mathrm{B}(\alpha) = \dfrac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)} \end{cases} \quad \text{with} \quad \Gamma(n) = (n-1)!$$

Topic extraction: LDA

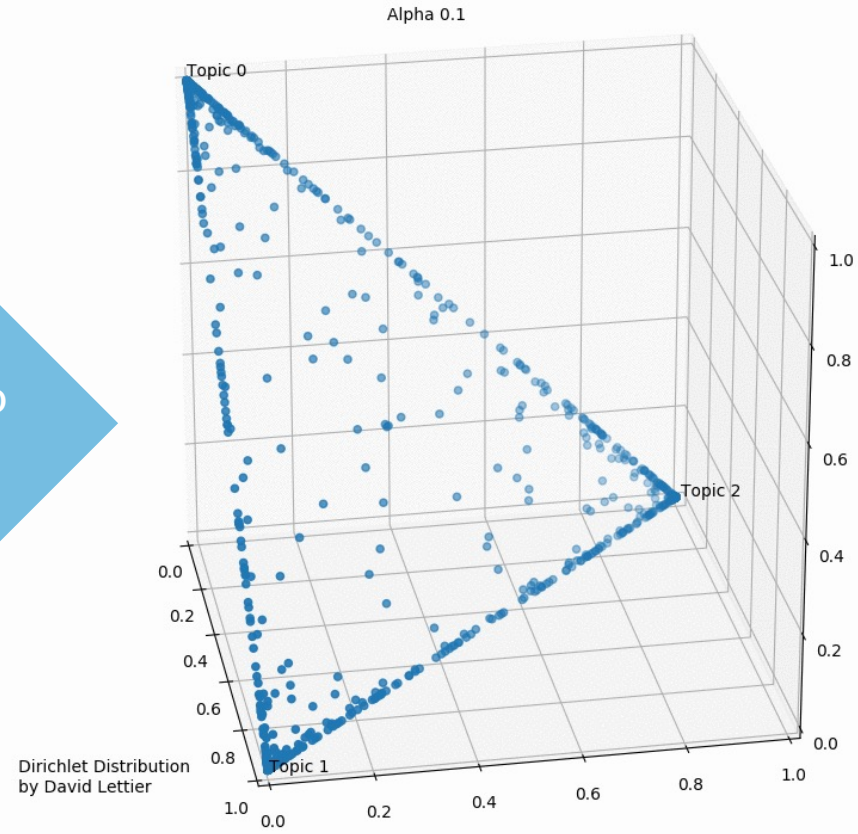
Dirichlet distribution:

$$Dir(\theta, \alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{n} \theta_k^{\alpha_{k-1}}$$

With n = 3, the range of solutions can be visualized as the following:



Apply to LDA



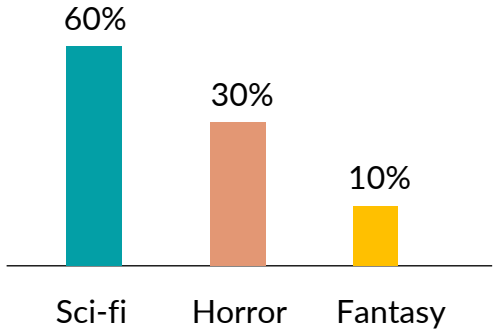
The smaller α is, the fewer topics are attributed per document

Topic extraction: LDA

Now let's go back to LDA

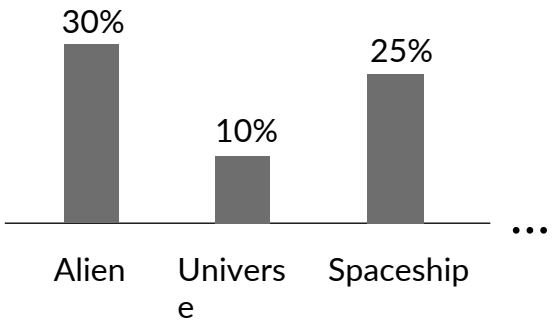
LDA assume that:

- A document is a (Dirichlet) distribution over topics

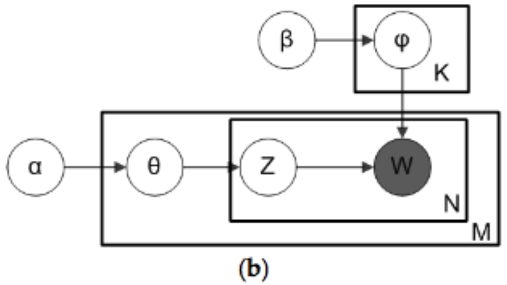


- A topic is a (Dirichlet) distribution over words

Topic: Sci-fi



Topic extraction: LDA



(b)

Now in In terms of probability…

$$p(W, Z, \Theta) = \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N_d} p(z_{d_n} \mid \theta_d) p(w_{dn} \mid z_{dn})$$

Generate topic probability

Select word from topic

For each document

For each word

Select a topic

- $\boldsymbol{\theta_d \sim Dir(\alpha)}$ → α is a user setting parameter
- $p(w_{dn}|z_{dn}) = \Phi_{z_{dn}w_{dn}} \sim \boldsymbol{Dir(\beta)}$ → *Probability of* w_{dn} *in the topic* z_{dn} and β set by user
- $p(z_{dn}|\theta_d) = \theta_{dz_{dn}}$ → z_{dn} component of the vector θ_d

Known

W: Dictionnary

w_{dn}: n-th word in document d

Unknown

Z: Topics of all words in all documents

z_{dn}: Topic of the n-th word in document d

Unknown

Θ: Distribution of topics for all documents

θ_d: Distribution over topic for document d

Topic extraction: LDA

How to find the unknown distributions?

We need to find:

θ: Probability distribution of topics in documents
Φ: Probability distribution of words in topics
Z: Topic of all words in document

How to estimate Z , Φ and θ ?

- With **Gibbs sampling**: Iterative algorithm based on MCMC method. After n iterations, the distributions will converge to the theoretical distributions.
- If we manage to estimate Z, it will be quite easy to estimate Φ and θ.

Topic extraction: LDA

Gibbs Sampling algorithm

Input: words $\mathbf{w} \in$ documents $\mathbf{d}$
Output: topic assignments $\mathbf{z}$ and counts $n_{d,k}, n_{k,w}$, and n_k
begin
  randomly initialize $\mathbf{z}$ and increment counters
  foreach *iteration* **do**
    for $i = 0 \rightarrow N-1$ **do**
      $word \leftarrow w[i]$
      $topic \leftarrow z[i]$
      $n_{d,topic}$-=1; $n_{word,topic}$-=1; n_{topic}-=1
      for $k = 0 \rightarrow K-1$ **do**
        $p(z = k|\cdot) = (n_{d,k} + \alpha_k)\frac{n_{k,w}+\beta_w}{n_k+\beta \times W}$
      end
      $topic \leftarrow$ sample from $p(z|\cdot)$
      $z[i] \leftarrow topic$
      $n_{d,topic}$+=1; $n_{word,topic}$+=1; n_{topic}+=1
    end
  end
  return $\mathbf{z}$, $n_{d,k}, n_{k,w}, n_k$
end

Algorithm 1: LDA Gibbs Sampling

Outputs

- the matrix Z (Topic of all words in a document)
- $n_{d,k}$ the number of words associated to the topic k in the document d
- $n_{k,w}$, how many times the word w is associated to the topic k
- n_k , how many times the topic k has been associated to a word

So we can find Φ and θ:

$$\theta_{d,k} = \frac{n_{d,k}+\alpha}{\sum_{|Z|} n_{d,k}+\alpha} \; ; \Phi_{k,w} = \frac{n_{k,w}+\beta}{\sum_{|W|} n_{k,w}+\beta}$$

Topic extraction: LDA

Gibbs Sampling algorithm

Gibbs conditionnal probability equation:

$$p(z_{d,n} = k | z_{-d,n}) = \frac{1}{M} \underbrace{\frac{n_{d,k} + \alpha}{\sum_{i=1}^{K} n_{d,i} + \alpha}}_{=\theta_{d,k}} * \underbrace{\frac{n_{k,w_{d,n}} + \beta}{\sum_{|W|} n_{k,w} + \beta}}_{=\Phi_{z_{d,n} w_{d,n}}}$$

Probability that the n-th word of the doc d is assigned to the topic k knowing all the topics of all the other words in the doc

How many times topic k occurs in this document

How many times the word $w_{d,n}$ occurs in this topic

Topic extraction: LDA

Gibbs Sampling algorithm simplified application

Do it again with another word in the updated table

1 Assign topic to each word of every doc (first randomly)

Etruscan	trade	price	temple	market
3	2	1	3	1

2 Create the word/topic matrix

Z	1	2	3
Etruscan	1	0	35
trade	10	7	1
price	0	0	34
temple	1	2	67
market	3	3	1
…			

3 If we focus on the word « trade »

Etruscan	trade	price	temple	market
3	??	1	3	1

W	1	2	3
Etruscan	1	0	35
trade	10	6	1
price	0	0	34
temple	1	2	67
market	3	3	1
…			
Total	196	203	180

- 2 topic 1
- 0 topic 2
- 2 topic 3

4 topics in total

$$p(z_{trade} = 1 | z_{-d,n}) = \frac{1}{M}\left(\frac{2+0.01}{4+0.01} * \frac{10+0.1}{196+0.1}\right) = \frac{0.02}{0.02+10^{-5}+10^{-3}} = 0.95$$

$$p(z_{trade} = 2 | z_{-d,n}) = \frac{1}{M} * 10^{-5} = 0.01$$

$$p(z_{trade} = 3 | z_{-d,n}) = \frac{1}{M} * 10^{-3} = 0.04$$

4 Update table

W	1	2	3
trade	11	6	1

Etruscan	trade	…
3	1	…

Multinomial law

$$p(z_{d,n} = k | z_{-d,n}) = \frac{1}{M} \frac{n_{d,k} + \alpha}{\sum_{i=1}^{K} n_{d,i} + \alpha} * \frac{n_{k,w_{d,n}} + \beta}{\sum_{|W|} n_{k,w} + \beta}$$

Assume that:
- $\alpha = 0.01$
- $\beta = 0.1$

Hands on: Implement our own LDA

Topic extraction: LDA

Model evaluation

LDA score must be correlated to human ratings so it's hard to evaluate…
« Human topic rankings serve as the gold standard for coherence evaluation »

Different approaches

- Human judgements
  - What is a topic
- Intrinsic Evaluation Metrics
  - Capturing model semantics
  - Topics interpretability
- Extrinsic Evaluation Metrics
  Is model good at performing predefined tasks, such as classification

Conclusion :

- No universal definition of what good mean
- A good topic model is use-case dependent: interpretation, prediction

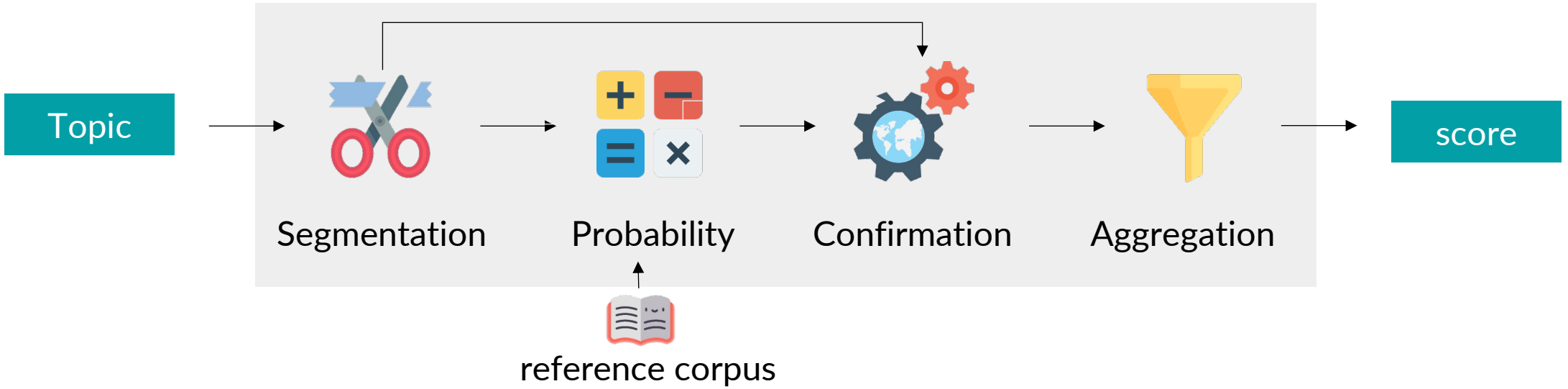
Topic extraction: LDA

Model evaluation : topic coherence

Topic Coherence Metric : How well a topic is "supported" by a text set (called reference corpus) ?

- Used statistics and probabilities drawn from reference corpus to give coherence score to a topic
- Depends not only on the topic itself but also on the dataset used as reference.

LDA coherence score: A pipeline in 4 steps



Topic extraction: LDA

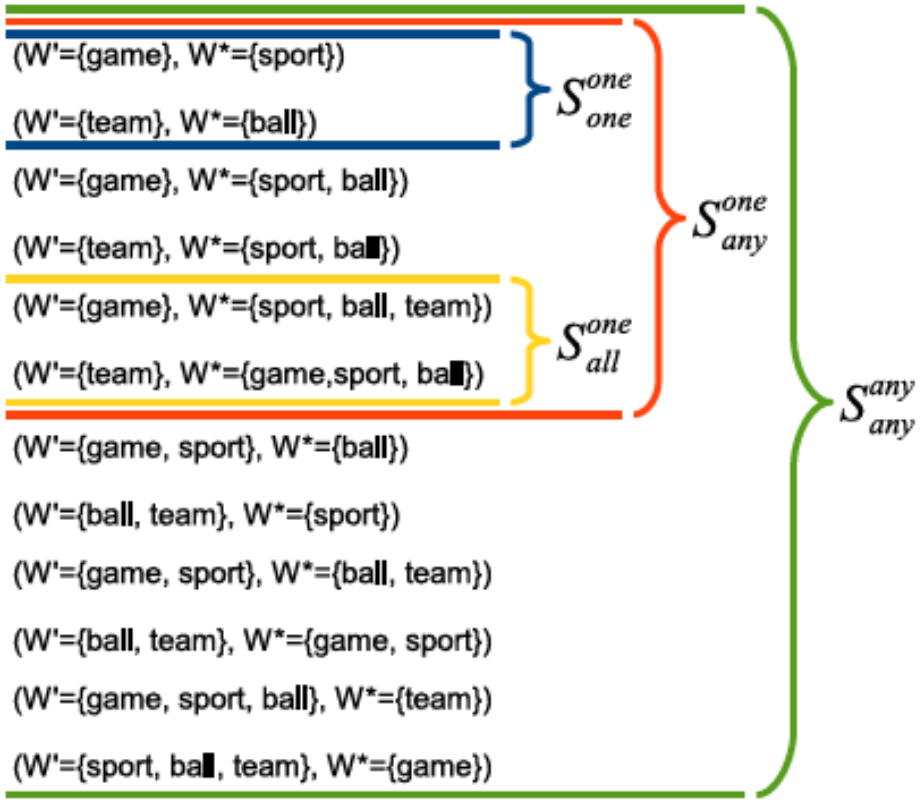
Model evaluation

1- Segmentation

How to divide a word set into smaller pieces?
Strategies by pairs of pairs , pairs of triples etc ..

Ex: **Topic word composition W = { (game, ball, sport, team) }**

- $S_{one}^{one} = \{(w_i, w_j)/w_i, w_j \in \mathrm{W}; \mathrm{i} \neq j\}$
- $S_{any}^{one} = \{(w_i, w)/w_i \in \mathrm{W}; w \in W^+; w_i \notin \mathrm{w}\}$
- $S_{pre}^{one} = \{(w_i, w_j)/\ w_i, w_j \in \mathrm{W}; \mathrm{i} > j\}$
- And many other… (check the ref article more details)



Topic extraction: LDA

Model evaluation

2- Probability calculation

We want to estimate the probability of a single word – **Boolean method**

- Boolean **document** (P_{bd})
- Boolean **paragraph** (P_{bp})
- Boolean **sentenced** (P_{bs})
- Boolean **sliding windows** (P_{sw})

$$\frac{\text{Number of unity in which the word occurs}}{\text{Total number of the chosen unity}}$$

Unity = document/paragraph/sentence/sliding window

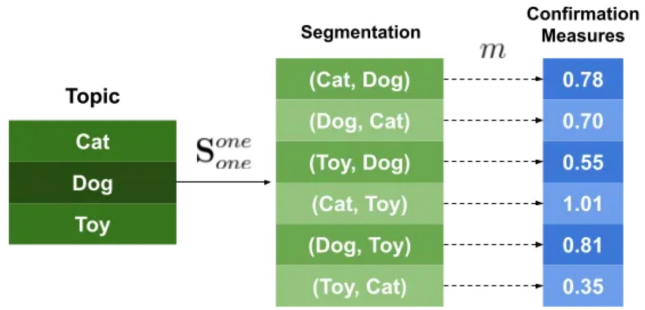
Topic extraction: LDA

Model evaluation

3. Confirmation measure : core of topic coherence

- Calculated over the pairs S using probabilities calculated P.
- Computes "how well" subset W° supports the subset W' in each pair.

- $m_d(S_i) = \mathrm{P}(\mathrm{W}'|\mathrm{W}^\circ) - \mathrm{P}(\mathrm{W}')$ Difference measure
- $m_r(S_i) = \frac{\mathrm{P}(\mathrm{W}'|\mathrm{W}^\circ)}{\mathrm{P}(\mathrm{W}')*\mathrm{P}(W^\circ)}$ Ratio measure
- And many others… (check reference for more details)

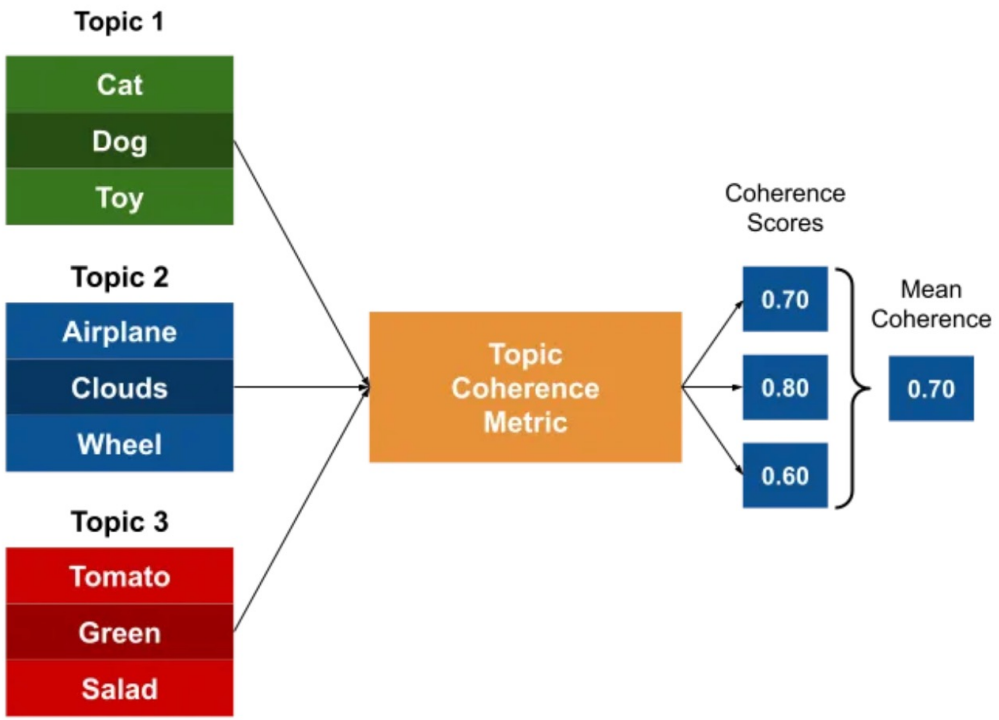


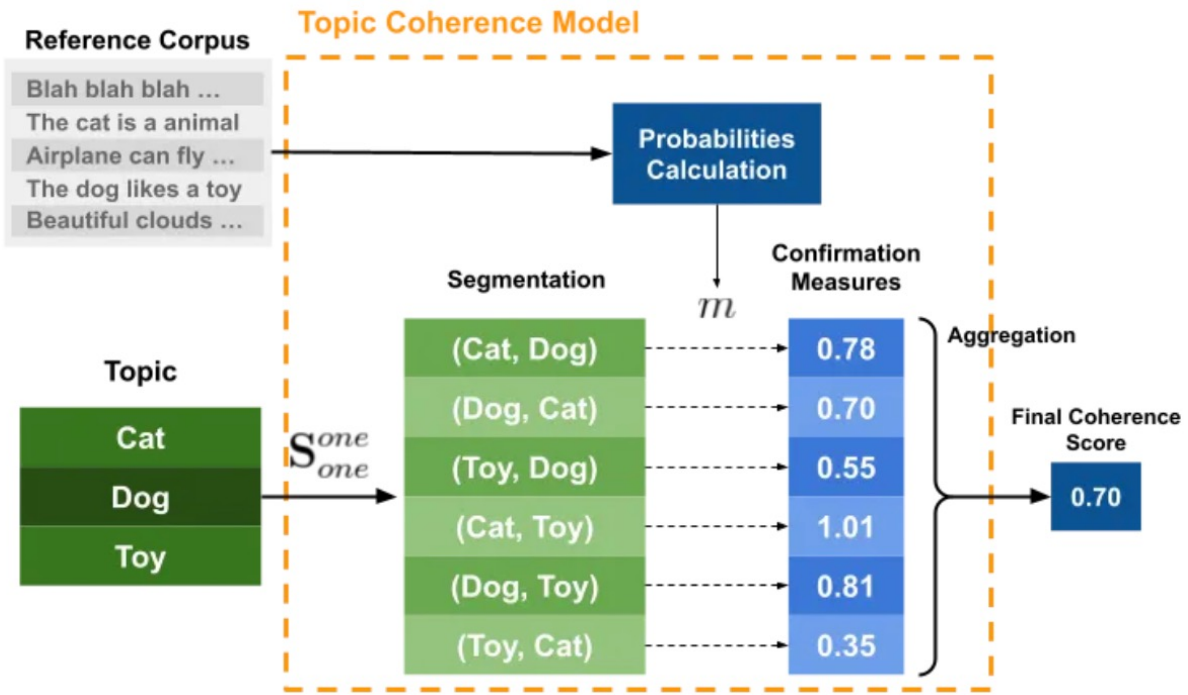
4. Aggregation Strategy

Average, median, minimum, maximum etc.

Topic extraction: LDA

Topic Coherence Measure of multiple topics





Topic extraction: LDA

LDA quick recap

Input:

- Number of topics
- α: mixture of topic per document
- β: mixture of word per topic
- Number of iterations

LDA Model

$$p(W, Z, \Theta) = \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N_d} p(z_{d_n} \mid \theta_d) p(w_{dn} \mid z_{dn})$$

Gibbs

$$\Phi_{k,w} = \frac{n_{k,w}+\beta}{\sum_{|W|} n_{k,w}+\beta} \; ; \; \theta_{d,k} = \frac{n_{d,k}+\alpha}{\sum_{|Z|} n_{d,k}+\alpha}$$

Output:

- Topic distribution per document Θ
- Word distribution per topic Φ

Agenda

1. Agile Methodology

2. Topic Extraction

    a. LSI/LSA review

    b. LDA

3. Sentiment analysis

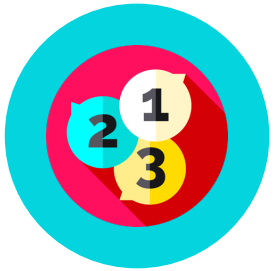
    a. Rule-based methods

    b. Learning based methods

Data pipeline

Data Collection

Data Cleaning



Word Embedding

Topic Extraction

Sentiment Analysis

Sentiment analysis

Goal: The process of determining whether a piece of text's sentiment is positive, negative or neutral.

Many business applications:

- Analyzing customer feedback on a particular Amazon product
- Gather people's opinions on social media about new political reform and get the overall general sentiment
- Obtain the financial market sentiment of a stock or credit, based on extracting the sentiment of certain news

Sentiment analysis: Why so important ?

Situation: Companies are receiving information more than ever before recently thanks to the social media channels, surveys and online reviews.

➡ Sentiment analysis will help in **saving time** and **effort** and **resources** analyzing this information that is unstructured as well by automating processes.

Scalability

- Huge amount of Data gathered
- Machine learning models can tag a sentiment to thousands of texts in seconds

Real-Time

- In some cases , there is a need for issues to be raised instantaneously e.g., a complaint from an important customer

Consistency

- Reducing human error by consistently tagging the same piece of text in the same sentiment
- Avoid different interpretations across colleagues

Sentiment analysis: How it works

Automated Systems

- Classifier will determine the sentiment via two stages: Training and prediction
- Training process: each text has a tag (negative, neutral or positive). A feature extractor transforms the text into vectors and tags to create and generate the model
- Prediction stage: New unseen text from the model will be transformed to sentiment predictions

Rule-Based Systems

- Rely on custom rules to classify text data
- Based on techniques like tokenization, parsing , stemming etc.
- Ability to customize and tailor-made algorithms based on the context
- Higher demands of maintenance as this type of rule-based model needs updating in order to improve results in the future

Hybrid Systems

- Widely-used approach for sentiment analysis
- Combines both automated and rule-based systems
- *Customization + Machine learning model*
- Example: Updating wordlist based on Word2Vec

Sentiment analysis: A hard task ?

YES !

1. Emotions are hard to understand by a machine, even we as humans can get confused
2. A piece of writing can contain multiple sentiments, for example :

> *“ Your work has a great structure, but I think it could have been better”*

3. Figurative speech complicated to be understood by computers.

> *“The best I can say about the movie is that it was interesting.”*

4. With the rise of social media , emoticons, slangs and acronyms participate heavily in determining the sentiment (“LOL”, “OMG”, …)

Agenda

1. Agile Methodology
2. Topic Extraction
    a. LSI/LSA review
    b. LDA
3. Sentiment analysis
    a. Rule-based methods
    b. Learning based methods

Sentiment analysis: Vader

Vader (Valence Aware Dictionary and sEntiment Reasoner)

Rule-based sentiment analysis model

VADER-lexicon:

Word	Sentiment rating
tragedy	-3.4
rejoiced	2.0
insane	-1.7
disaster	-3.1
great	3.1

Labeled thanks to Amazon Mechanical Turk

- Score of each word between -4 and 4
- Sum each word and normalize it to have a score between -1 and 1.

$$f(x) = \frac{x}{\sqrt{x^2 + \alpha}}$$

+ 5 simple heuristics:

- Punctuation (Cool!!! VS Cool)
- Capitalization (Amazing VS AMAZING)
- Degree modifiers (it's cute VS it's sort of cute)
- Polarity (I like it but I don't want it)
- Handles Emojies , slangs and emoticons

Limits:

- Heuristics sometimes fail
- Out of Vocabulary (OOV) words will be neutral

Advantages of Vader

- ✓ It works exceedingly well on social media type text, yet readily generalizes to multiple domains
- ✓ It **doesn't require any training data** but is constructed from a generalizable, valence-based, human-curated gold standard sentiment lexicon
- ✓ It is fast enough to be used online with streaming data, and
- ✓ It does not severely suffer from a speed-performance tradeoff.

NLP Rule-based libraries

	Pros	Cons
TextBlob	• It is an easy-to-use toolkit that works really well for common, more 'traditional' NLP tasks	• Slower than spacy but faster than NLTK • Does not provide features like dependency parsing, word vectors like with Spacy
NLTK	• 'Mother' of all NLP libraries • Works perfectly for educational purposes (datasets) • Standard for many NLP tasks: stop words, stemming..	• Difficult to learn and use • Slow • Only splits text by sentences, without analyzing the semantic structure

Agenda

1. Agile Methodology
2. Topic Extraction
    a. LSI/LSA review
    b. LDA
3. Sentiment analysis
    a. Rule-based methods
    b. Learning based methods

Sentiment Analysis: Learning based methods

Raw data → **Engineering approach** → **Input matrix** → **Machine Learning Model** → **Sentiment**

	Sentence	label
0	a stirring , funny and finally transporting re imagining of beauty and the beast and 1930s	1
1	apparently reassembled from the cutting room floor of any given daytime soap	0
…	…	…
,999	the movie is undone by a filmmaking methodology that 's just experimental enough	1

Feature Based:
- CountVectorizer
- TF-IDF

Embedding Based:
- FastText
- Flair
- BERT

	Features					label
	funny	stirring	apparently	…	movie	
0	2,2	0,2	0	…	0	1
1	0	0	3	…	0	0
…			…			
1,999	0	0	0	…	0,5	1

	Sentence Embeddings				label
	0	1	…	767	
0	-0.215	-0.1402	…	0.201	1
1	-0.172	-0.144	…	0.371	0
…	…	…	…	…	…
1,999	0.124	0.014	…	0.274	1

- Logistic Regression
- Decision Tree
- Random Forest
- XGBoosting
- Neural Network

Useful NLP libraries

	Pros	Cons
flair	• Open-source library designed to reach the state of the art in NER • It supports a good number of languages and is always looking to add new ones • Gives access to their custom embeddings • Easy to use • Stack and combine different word embeddings	• Known to be slow
spaCy	• Excellent documentation • Fastest NLP framework • Provides built in word vectors • Easy to learn • Support is active and releases are ongoing	• Accuracy is too limited • Doesn't support many languages , there are 7 languages at date
Transformers	• State of the art attention-based models • A wide collection of pre-trained model for over 100 languages • Covers a wide variety of NLP tasks: text classification, information extraction, question answering, summarization, etc.) • Best-in-class documentation and large communities	• Covers only deep transformers models, • Could be too complex for simple task

Reference

Topic extraction

- LDA: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf
- Gibbs sampling: https://u.cs.biu.ac.il/~89-680/darling-lda.pdf
- Coherence measure: https://svn.aksw.org/papers/2015/WSDM_Topic_Evaluation/public.pdf
- DMM: http://dbgroup.cs.tsinghua.edu.cn/wangjy/papers/KDD14-GSDMM.pdf

Sentiment analysis

- VADER: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/download/8109/8122
- VADER github: https://github.com/cjhutto/vaderSentiment
- FLAIR: https://arxiv.org/pdf/1909.09586.pdf
  https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Hands on: Introduction on NLP package

Final Restitution : 13/03 at Capgemini

147 quai du président roosevelt 92130 issy-les-moulineaux

Participants

Project team

Client team

- Business director
- Data Science director

Sponsor

- Head of consumer gas & electricity (n-1 to Stéphane Michel, DG gas, renewables and power, exco member)

Content

Objectives

- Context
- Need
- Analysis scope

Methodology

- Data (what, how)
- NLP methodologies used

Analysis Results

- Explain your conclusions, and the limits associated with them

Recommendations

- Based on the results of the analysis (including the KPIs)

Bonus – Next Steps

- High level roadmap
- Estimate the operational cost of implementation

Format

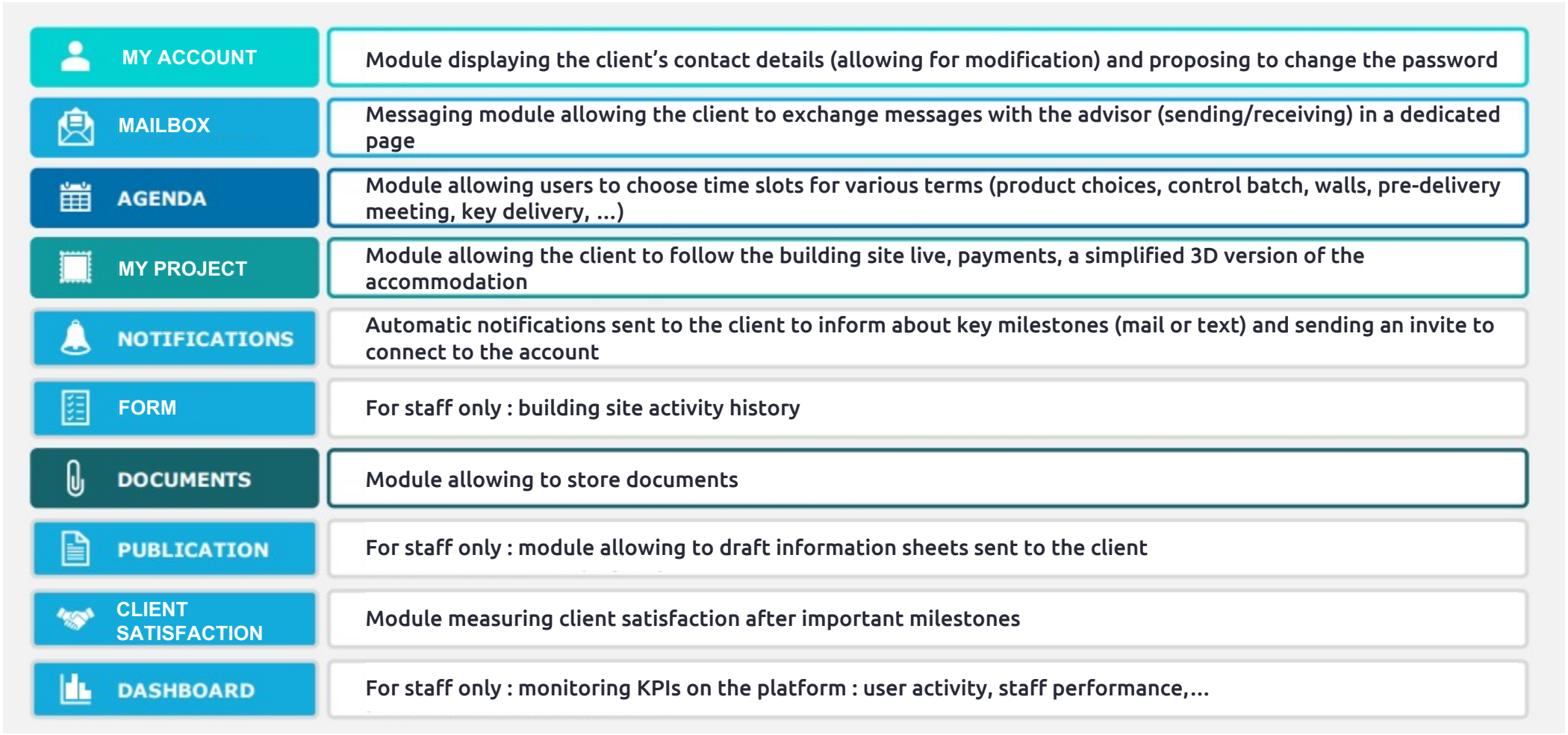
25mn presentation + 10mn questions

Course evaluation

Did you like that course ? It’s time to share your feedbacks !

HANDS ON #2: PRODUCT BACKLOG FEEDING

Example



PRIORITIZATION : A KEY SUCCESS FACTOR

Why prioritize product backlog items?

- **Focus on** the **must have items** to develop in order to deliver the **highest value to customer**
- **Consolidate technical environment**
- **Reduce risk**: key performance parameters are analyzed so problem areas surface early
- **Keep on schedule**

Business Priorities

Customer attractivity levers

*Example: **retention, acquisition**, treatment of dissatisfaction, cross-selling*

Operational efficiency levers

*Example: **automatization & robotics,** selfcare, dematerialization, reengineering of the FO/BO processes*

S P R I N T

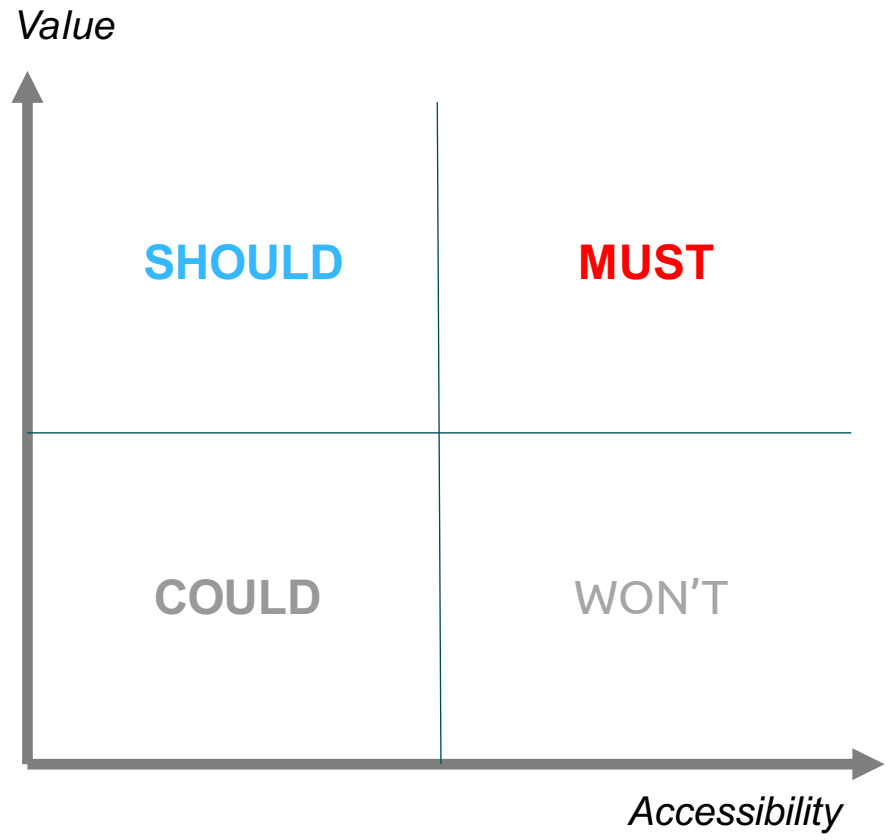
Technical Priorities

- **Development time**
- **Development cost**
- **Implementation risk**
- **Usability of features for other User Stories**

HANDS ON #3: PRIORITIZATION OF PRODUCT BACKLOG ITEMS

Example

1. **Measure Desirability and delay low-value features**
   - The concept attractiveness mapping can help assess Desirability (seenext slide)
   - You may arbitrarily score features based on user input
2. **Measure Viability and delay low-revenue features**
   - Build a Business Plan around the features
   - Features with the highest revenue are likely to be prioritary
   - You may use user input to define your BP hypotheses
3. **Measure Feasibility and exclude unachievable features**
   - List the prerequisites to build the features.
   - Asses the cost of building the features to enrich the BP.
4. **Rank features in a value/accessibility matrix**
   - Use the *Weighted Shortest Job First* methodology to select top performer
   - Top performers are features with the highest cost of delay.

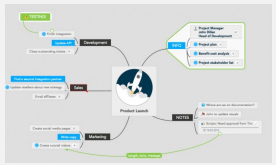


DIGITAL TOOLS YOU CAN USE FOR YOUR AGILE PROJECTS

Define your product

- Mindmeister
- JIRA
- Excel

Mindmapping



Backlog management

Make a prototype

- InVision
- Napkin
- POP
- Weebly

Test it with your user & get a feedback

- A/B testing
- Google Analytics
- SurveyMonkey
- Kahoot
- Typeform

Pitch your idea

- Prezi
- Sway Microsoft
- Vyond
- Pecha Kucha

Plan and manage the work break down

- Basecamp
- Trello
- Post-it on the wall

Communicate fast and efficiently with your team

- Microsoft Teams
- Slack

