Capgemini invent

Project 5 Data Science / OpenClassrooms

Presented by:
Marc Felix DEGNI
Apprentice Data Scientist Consultant OC/CC

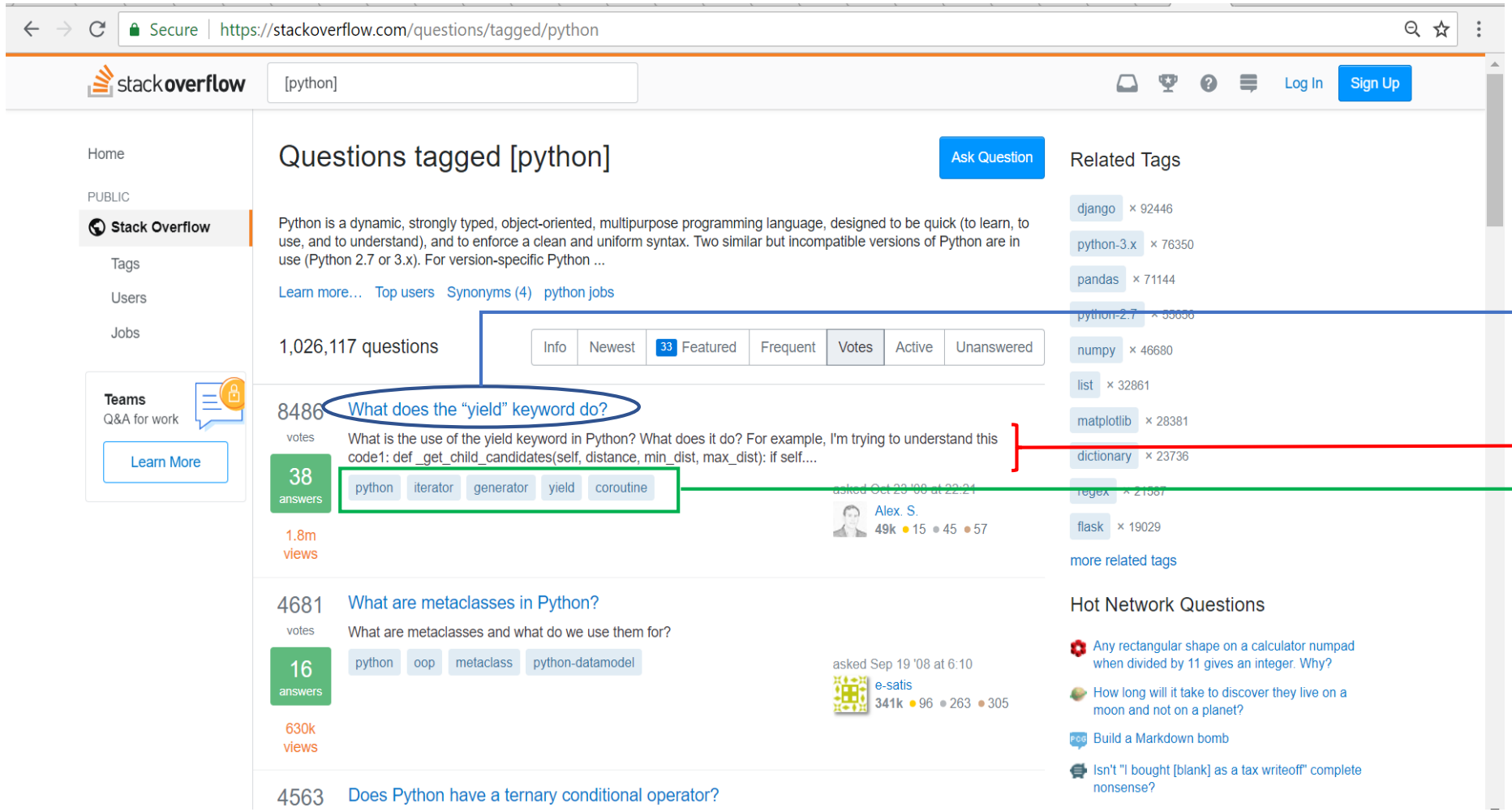
AutoTag Suggession StackOverFlow

OPENCLASSROOMS

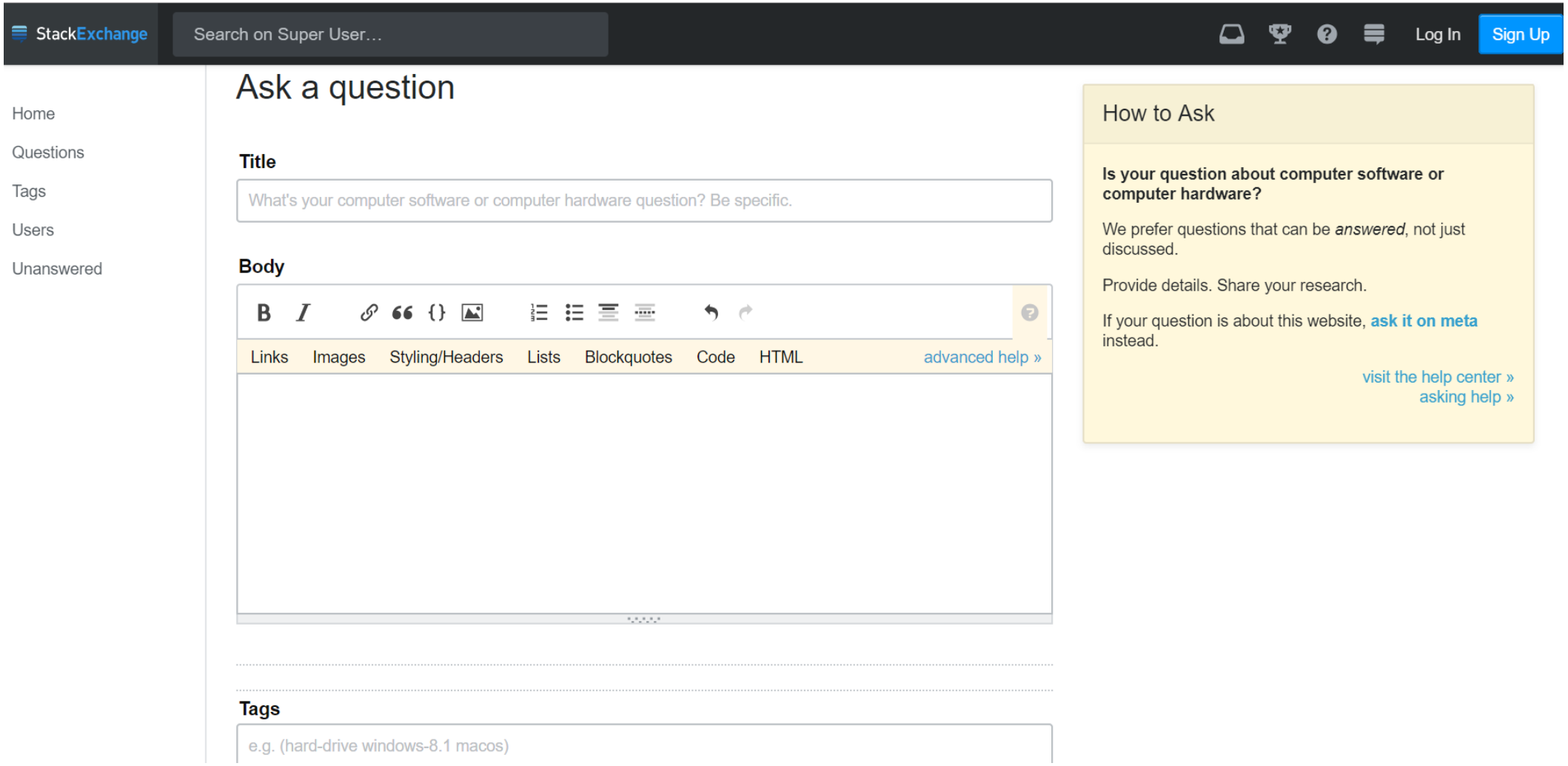
stackoverflow

05/11/2018

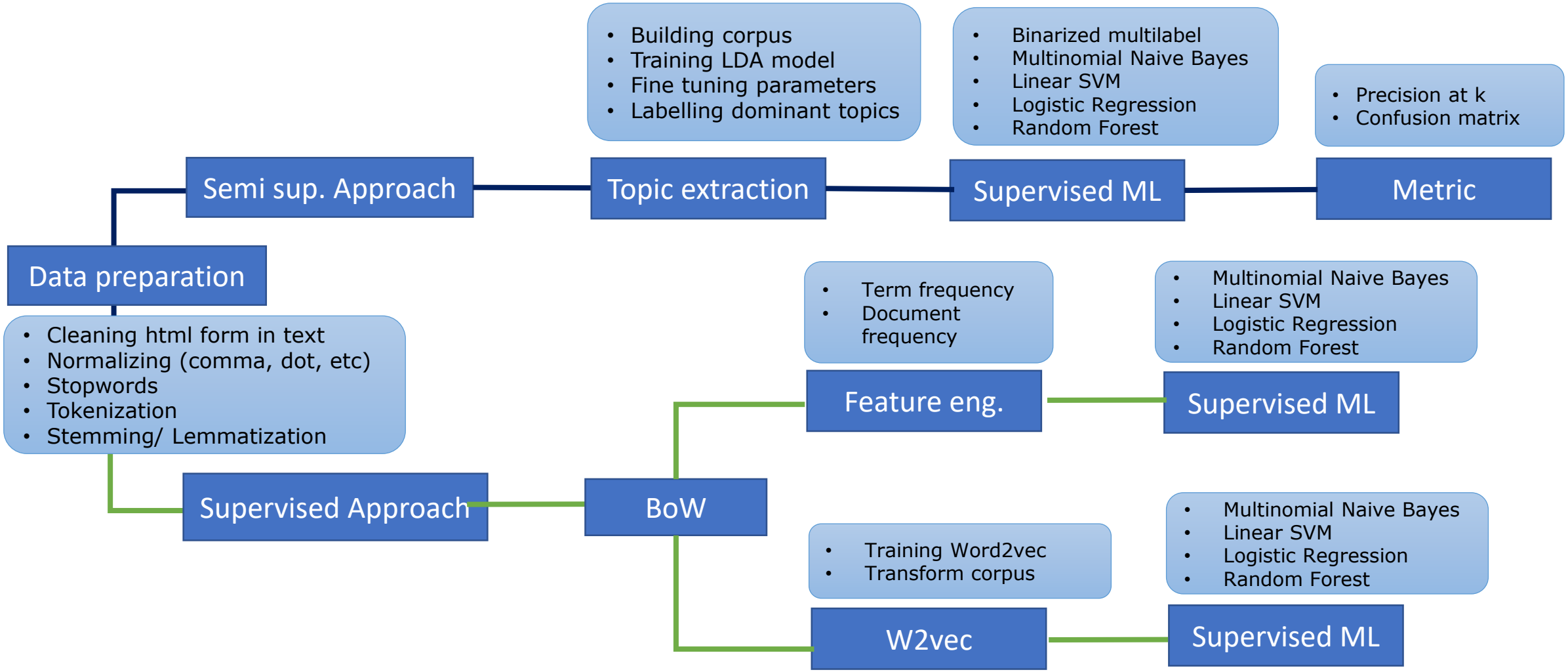
Data description



Data cleansing



Data approach



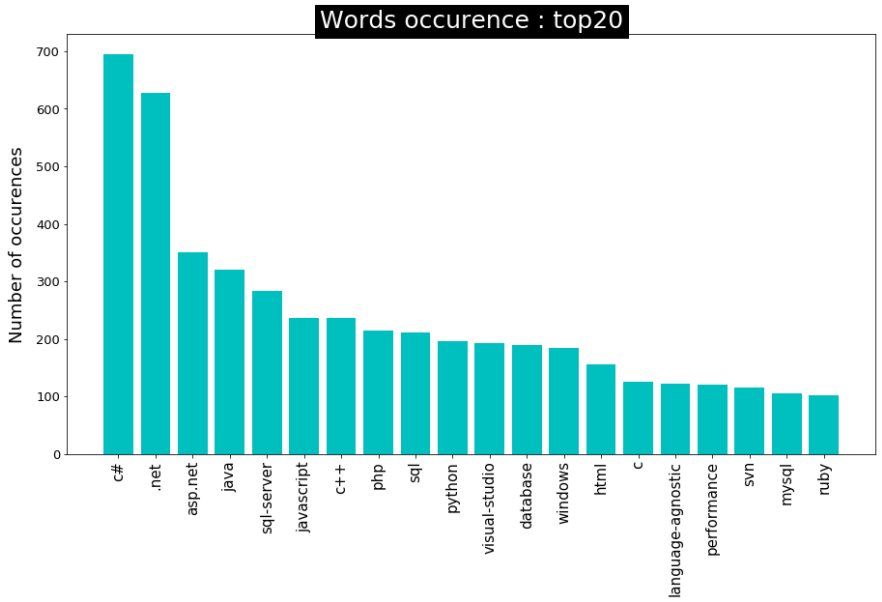
Data cleansing

	Body	Title
0	I want to use a track-bar to change a form'...	Convert Decimal to Double?
1	I have an absolutely positioned <code>div</...	Percentage width child element in absolutely p...
3	Given a <code>DateTime</code> representing ...	How do I calculate someone's age in C#?
4	Given a specific <code>DateTime</code> valu...	Calculate relative time in C#
6	Is there any standard way for a Web Server ...	Determine a User's Timezone

token_tag
[c#, floating-point, type-conversion, double, ...
[html, css, css3, internet-explorer-7]
[c#, .net, datetime]
[c#, datetime, time, datediff, relative-time-s...
[javascript, html, browser, timezone, timezone...

Questions	token_questions	tokens_clean	tokens_clean_lemma
Convert Decimal to Double? I want to use a track-bar to change a form's opacity. \r\n\r\n This is my code: \r\n\r\n<pre> <code>decimal trans = trackBar1.Value / 5000;\r\nthis.Opacity = trans;\r\n</code> </pre>\r\n\r\n When I build the application, it gives the following error: \r\n\r\n<blockquote>\r\n Cannot implicitly convert type <code>'decimal'</code> to <code>'double'</code>. \r\n</blockquote>\r\n\r\n I tried using <code>trans</code> and <code>double</code> but then the control doesn't work. This code worked fine in a past VB.NET project. \r\n	['convert', 'decimal', 'to', 'double', 'i', 'want', 'to', 'use', 'a', 'track', 'bar', 'to', 'change', 'a', 'form', 's', 'opacity', 'this', 'is', 'my', 'code', 'when', 'i', 'build', 'the', 'application', 'it', 'gives', 'the', 'following', 'error', 'cannot', 'implicitly', 'convert', 'type', 'to', 'i', 'tried', 'using', 'and', 'but', 'then', 'the', 'control', 'doesn', 't', 'work', 'this', 'code', 'worked', 'fine', 'in', 'a', 'past', 'vb', 'net', 'project']	['convert', 'decimal', 'double', 'want', 'use', 'track', 'bar', 'change', 'form', 'opacity', 'code', 'build', 'application', 'gives', 'following', 'error', 'implicitly', 'convert', 'type', 'tried', 'using', 'control', 'doesn', 'work', 'code', 'worked', 'fine', 'past', 'vb', 'net', 'project']	['convert', 'decimal', 'double', 'want', 'use', 'track', 'bar', 'change', 'form', 'opacity', 'code', 'build', 'application', 'give', 'follow', 'error', 'implicitly', 'convert', 'type', 'try', 'use', 'control', 'doesn', 'work', 'code', 'work', 'fine', 'past', 'vb', 'net', 'project']

Cleaning → **Tokenization** → **Stopword** → **Lemmatization**



Topic extraction (Part I)

Latent Dirichlet Allocation: an unsupervised algorithm for topic modelling

Text samples

Document 1

[...] People spend more time on Netflix than on [...]

Document 2

[...] Black Mirror is NOT a show for the family [...]

Document 3

[...] Teenagers watching TV shows learn better English [...]

Document 4

[...] House of cards teaches the fundamentals of the mechanics of power [...]

Automatic topic extraction

- "people", "teenagers", "family"
- "Netflix", "TV shows", "season"
- "learn", "teaches", "English"

Topic allocation

Document 1: 70% | 20% | 10%

Document 2: 50% | 50%

Document 3: 75% | 25%

Document 4: 15% | 15% | 70%

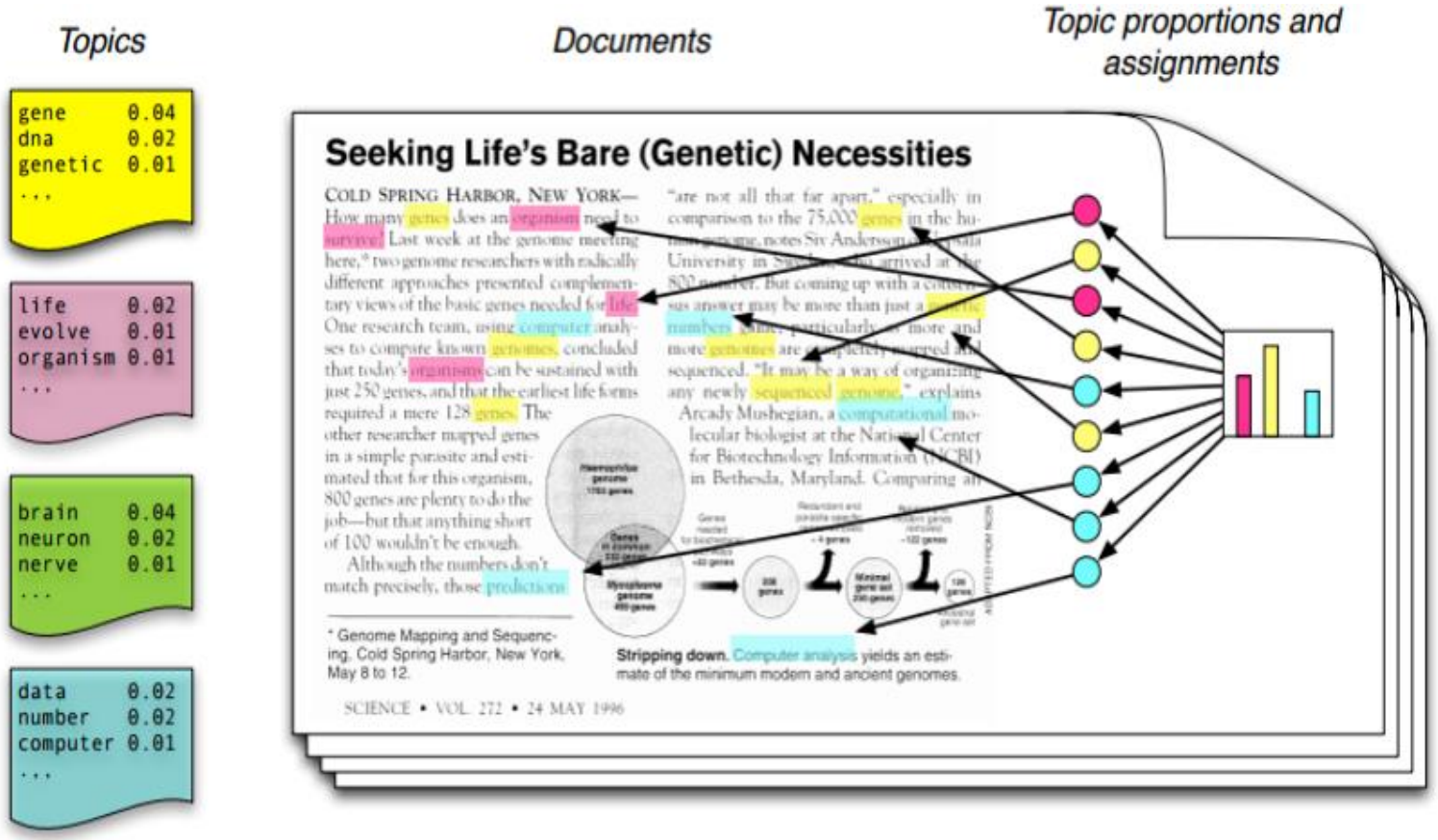
Topic interpretation

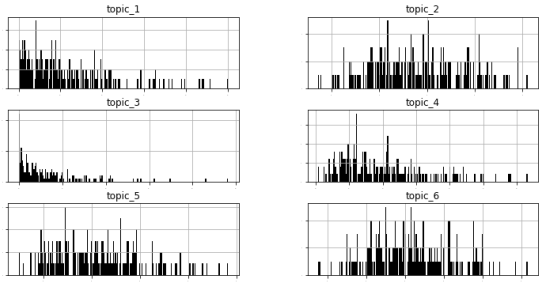
- Persons
- TV shows
- Knowledge

Tasks performed by a LDA model

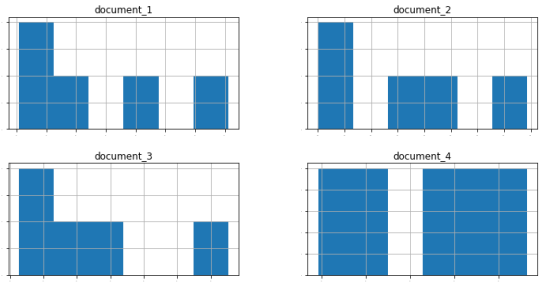
Topic extraction (Part 2)

LDA optimizes the probability that documents occur knowing topics are distributed among them





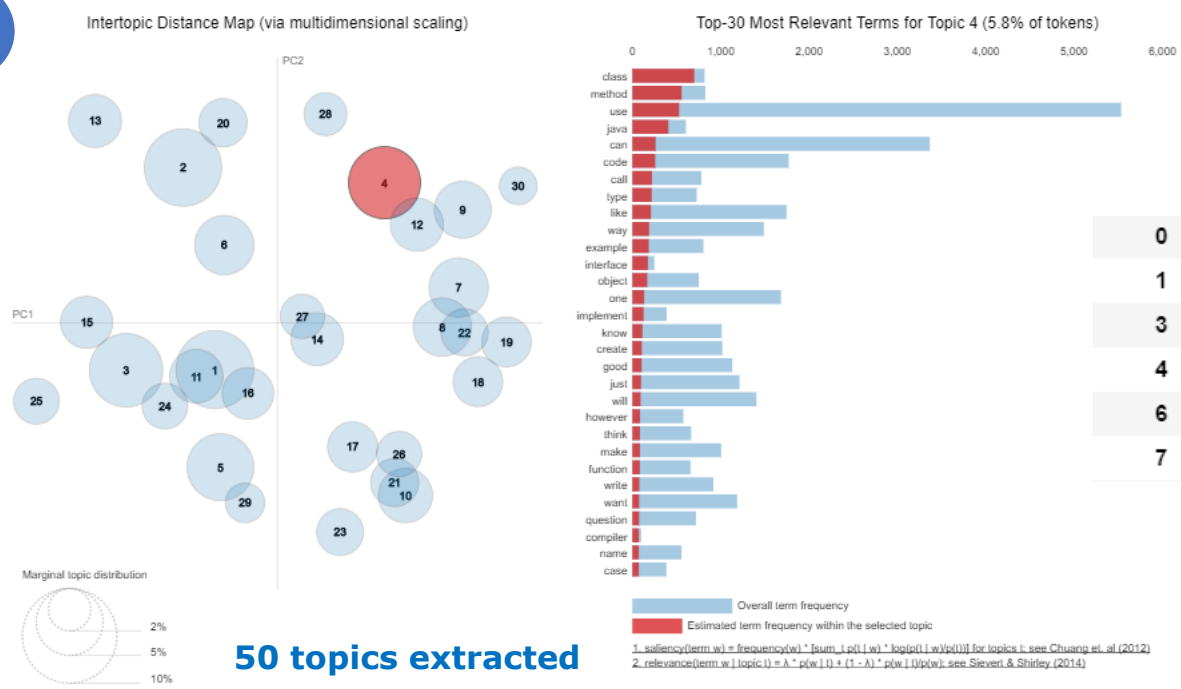
Topics are represented by different word probabilities



Similarly, documents are represented by different topic probabilities

Results approach and performance

1



50 topics extracted

2

Dominant topic labelled

	Dominant_Topic	token_tag
0	4.0	[c#, floating-point, type-conversion, double, decimal]
1	1.0	[html, css, css3, internet-explorer-7]
2	2.0	[c#, .net, datetime]
3	27.0	[c#, datetime, time, datediff, relative-time-span]
4	20.0	[javascript, html, browser, timezone, timezoneoffset]

3

How to deal with multilabel ? Binarizing !!!

Multilabel (Targets)

0	[c#, floating-point, type-conversion, double, decimal]
1	[html, css, css3, internet-explorer-7]
3	[c#, .net, datetime]
4	[c#, datetime, time, datediff, relative-time-span]
6	[javascript, html, browser, timezone, timezoneoffset]
7	[.net, math]

Labels Binarized

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

4

Supervised method (LR/ MNB)

	Probabilities
c#	0.196276
.net	0.120634
c++	0.111925
php	0.074453
c	0.070478

	Probability
c#	0.196276
.net	0.120634
c++	0.111925
php	0.074453
c	0.070478

Total supervised approach

Bag of word : TF-IDF

Bag of words

- Consider a corpus of documents, each document is a text
- Each document contains a certain number of words for which we can do tokenization
- Each word will be a token
- Bag-of-word: representing without any order the words in a document

Example:

- "It is super cool to be a Data Scientist"
- - "super" "it "scientist" "be" "to" "cool" "is" "a" "data"

Term Frequency principle

- After creating a bag of words, it is necessary to think about the importance of each word in the text
- $tf(t,d)$ can be read as the term frequency of the term t in document d
- Formula: $$tf(t,d) = \frac{n_{t,d}}{\sum_{i \in d} n_{i,d}}$$
- With $n_{i,d}$ the number of appearance of word i in document d
- Thus $\sum_{i \in d} n_{i,d}$ is the total number of words in the document d

Example:

For the next slides we will use the following examples:

- Doc 1: "It is super cool to be a Data Scientist"
- Doc 2: "Data Science is done with data by Data Scientists"
- Doc 3: "Poker is a cool game"

By using stopwords & lemmatization, only the following words remain: "super", "cool", "data", "science", "do", "poker", "game"

Bag of word : TF-IDF

Principle of TF-IDF

- Continuity of the previous slide
- It is more likely to have an insight on words which are really important
- The weights of the words which are extremely frequent like “the” or “of” are diminished
- It deals with how much information the word gives for the query
- Formula of Inverse Document Frequency:

$$idf(t) = \log \left| \frac{n}{\{d \in D : t \in d\}} \right|$$

- With n the total number of documents, D the set of all documents, d a given document in D, t is a given term. Here, the natural logarithm (base e) is used
- Thus, $\{d \in D : t \in d\}$ is the number of documents in which the term t appears
- Once Term Frequency and Inverse Document Frequency has been found, it is possible to compute TF-IDF:

$$tfidf(t, d) = tf(t, d) \times idf(t)$$

TF-IDF on data

	aa	aad	aardvark	ab	abandon	abap	abbreviate	abbreviation	abbriviate	abc	...	éàè	öäü	šturc	caught
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0

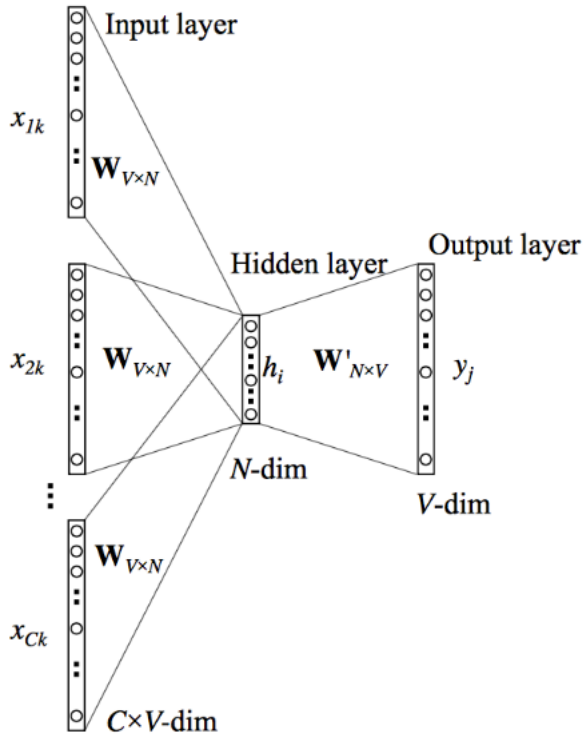
6 rows × 10757 columns

Supervized model

Model	BoW	Metrics
Multinomial NB	TF-IDF	27,12%
Linear SVM	TF-IDF	24,37%

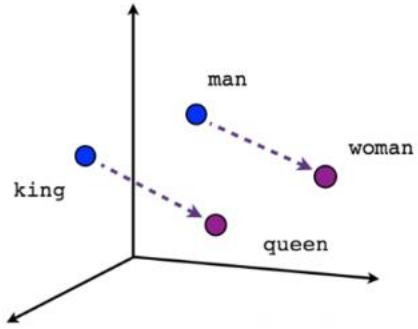
Bag of words : Word Embedding (Word2vect)

Intuition behind Word2Vec

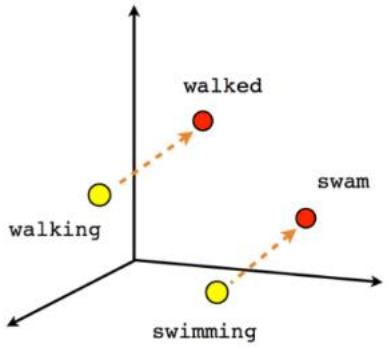


Représentation du modèle CBOW

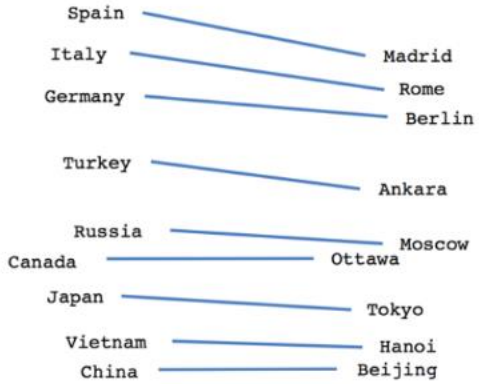
Example of word vectorized



Male-Female



Verb tense



Country-Capital

Example of word similar

```
model.most_similar('convert')
```

executed in 16ms, finished 14:54:44 2018-09-17

```
[('hashcode', 0.9891046285629272),
 ('hash', 0.9856103658676147),
 ('iterate', 0.9854134321212769),
 ('aq', 0.9842157959938049),
 ('timezone', 0.9826910495758057),
 ('chicago', 0.9822856783866882),
 ('represent', 0.9810150861740112),
 ('decrypt', 0.9808613657951355),
 ('concatenate', 0.980566143989563),
 ('representation', 0.9798986315727234)]
```

Supervized model

Model	BoW	Metrics
Multinomial NB	word2vec	8,90%
Linear SVM	word2vec	2,62%

DEPLOY MODEL ON HEROKU CLOUD

- Creating API with microframework (FLASK)
- Deploy on heroku cloud

Link for demonstration

https://autotag-suggestion-nlp.herokuapp.com/

CONCLUSION

- ***The reduction of the sparsity*** of the TF-IDF matrix by a principal component analysis or other dimension reduction method. The algorithms used will then have less computational problems and can focus their intelligence on the words having weight in at least one document.

- Another improvement would be ***the deletion of words with a weight below a fixed threshold***, because these features will be common to each document and therefore will not bring any value.

- With the word2vec, instead of using a MeanEmbedding which aggregates the values of the weights of the words present in the questions, one could combine ***word2vec and TF-IDF method.***