Projet 5 : Catégorisez automatiquement des questions

Claire Gayral

Décembre 2021 - Janvier 2022

Introduction

Introduction - Plan

1. Les données textuelles
   - Pré-traitements données textuelles
   - Exploration sur les tags
2. Classification Non Supervisée
3. Classification Supervisée
   - Modèles de classification
   - Résultats des classifications
4. Conclusion

Import des données

Les données d'entrée :

- Publications sur StackOverFlow
- 3 parties : titre, corps et tags
- Sélection des publications avec tags parmis les 10 000 premières

Requête SQL sur https://data.stackexchange.com

```
 SELECT Id, Title, Tags, Body
FROM posts
WHERE Id < 100000 AND Tags <> ''
```

Les données - Prétraitements

`<p>How do I forcefully unload a <code>ByteArray</code> from memory using ActionScript 3?</p>`

1. Format, ponctuation, filtre versions des langages de programmation

   `how, do, i, forcefully, unload, a, bytearray, from, memory, using, actionscript'`

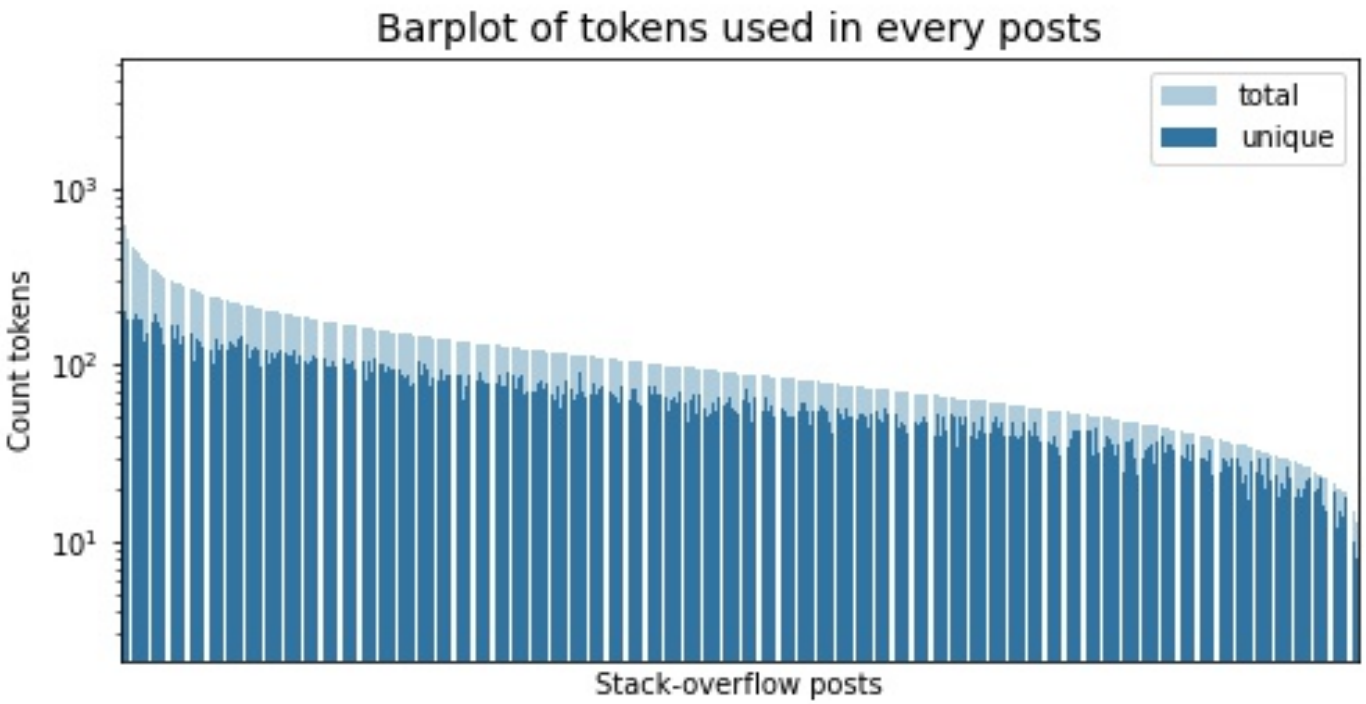
2. Stop words

   `forcefully, unload, bytearray, memory, actionscript`

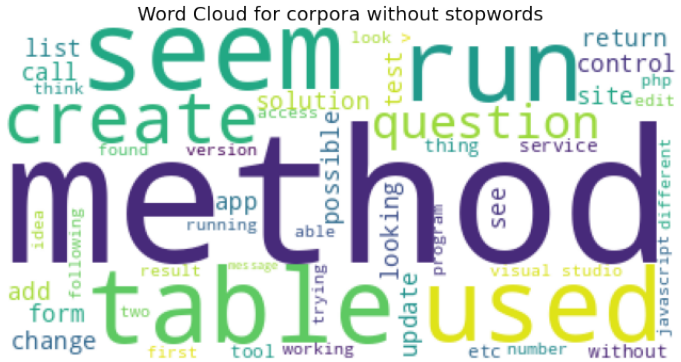
3. Lemmatisation

   `forc, unload, bytearray, memori, actionscript`

Les données textuelles - Répartition des mots

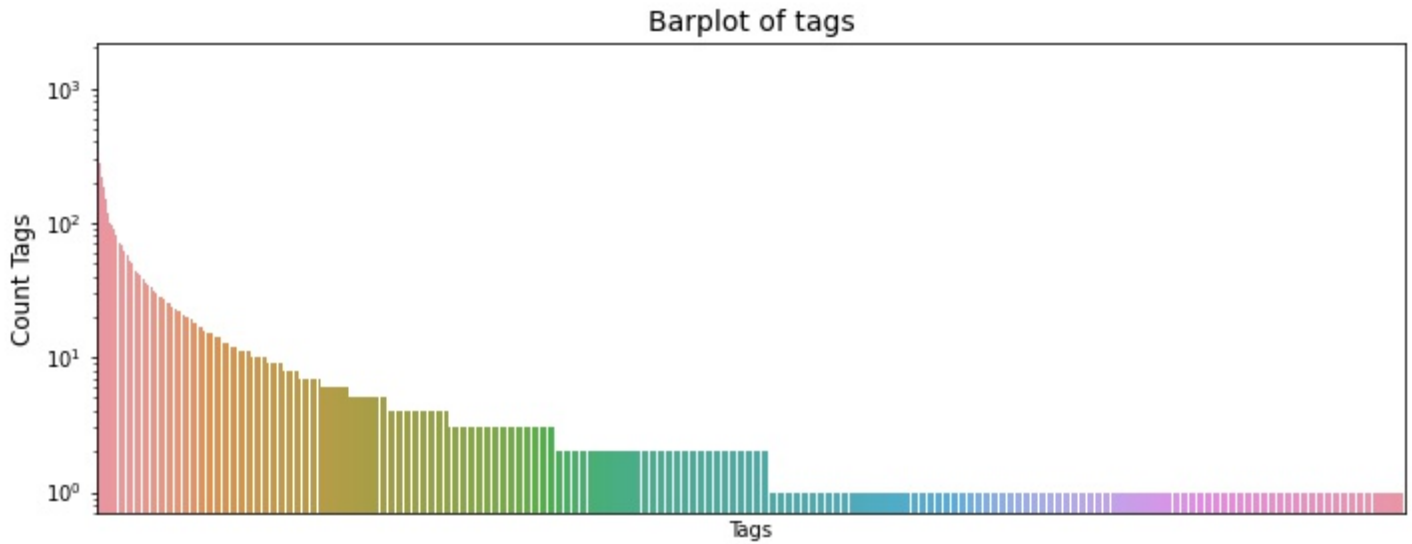


Les données textuelles - Représentation des tokens

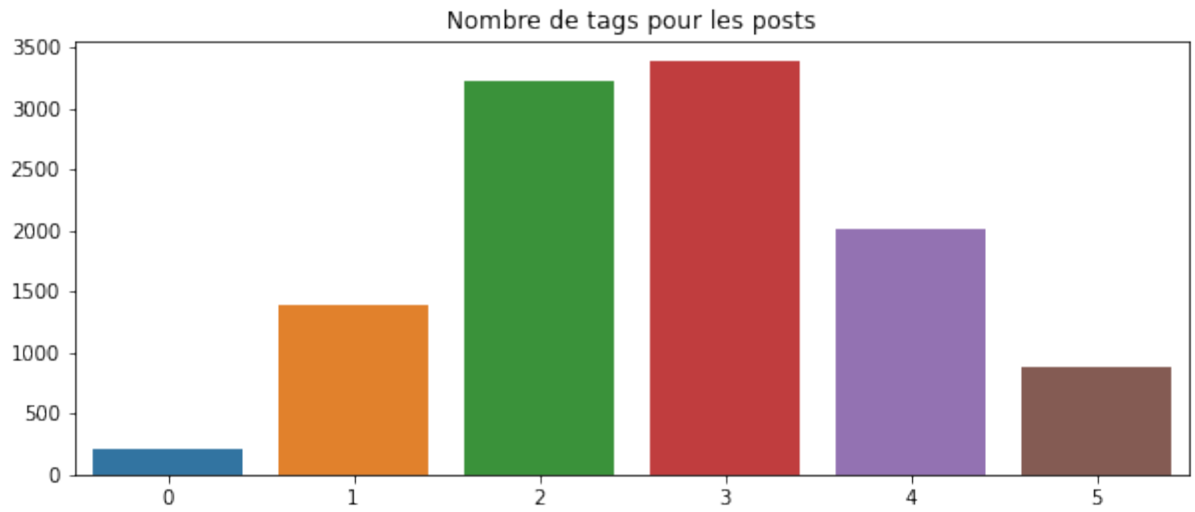


Les tags - Pré-traitements

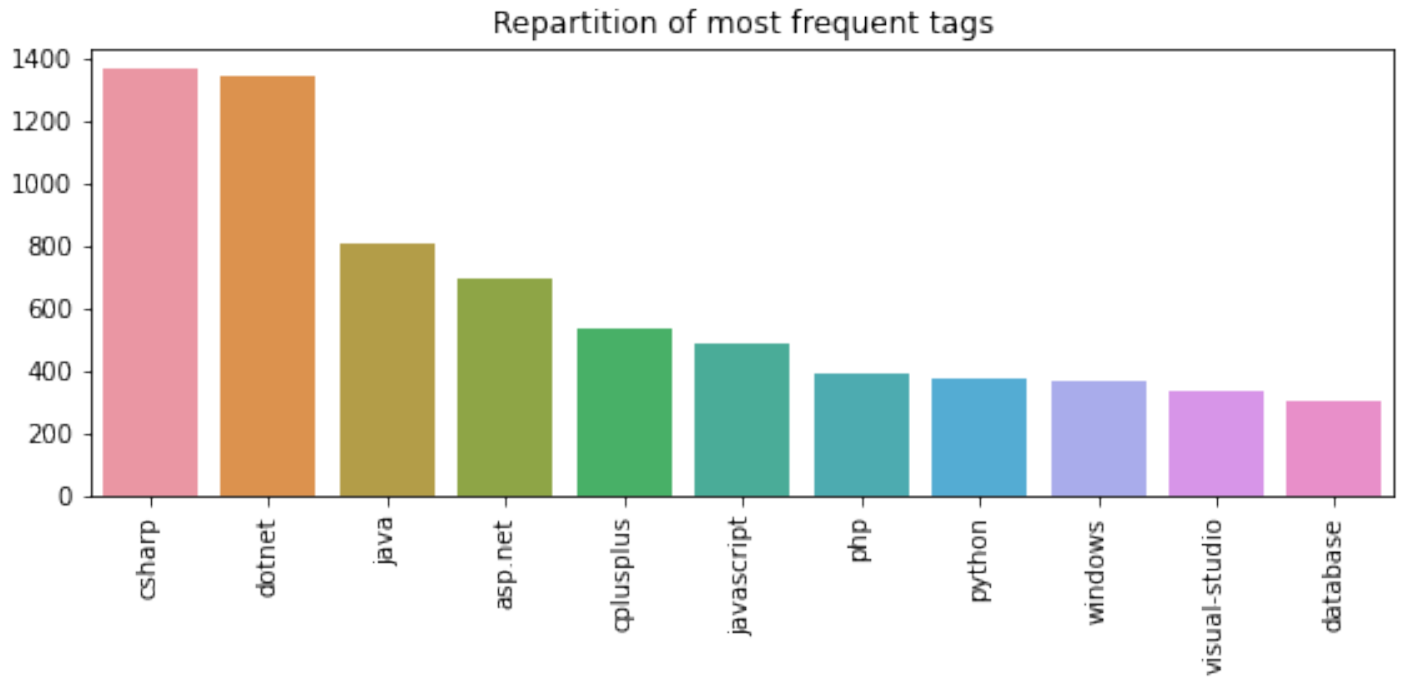
- `<c#><.net><wcf><web-services><soa>`
- Filtre nom de langages :
  `c♯, C♯, c♯-2.0, c♯-3.0, c♯-4.0` $\rightarrow$ `csharp`
- Data Frame en one hot encoding



Les tags - Nombre de tags par publication



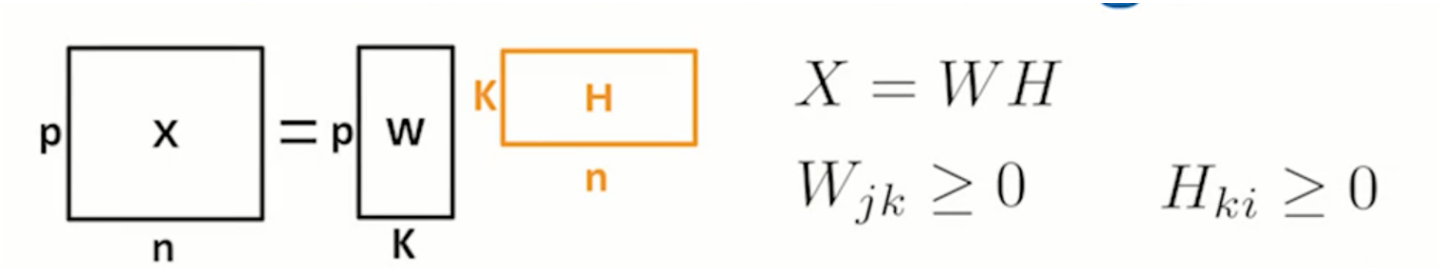
Tags - Création d'une variable univariée




```
↪ y = tags["csharp"]
```

Tags - NMF 1

La NMF :

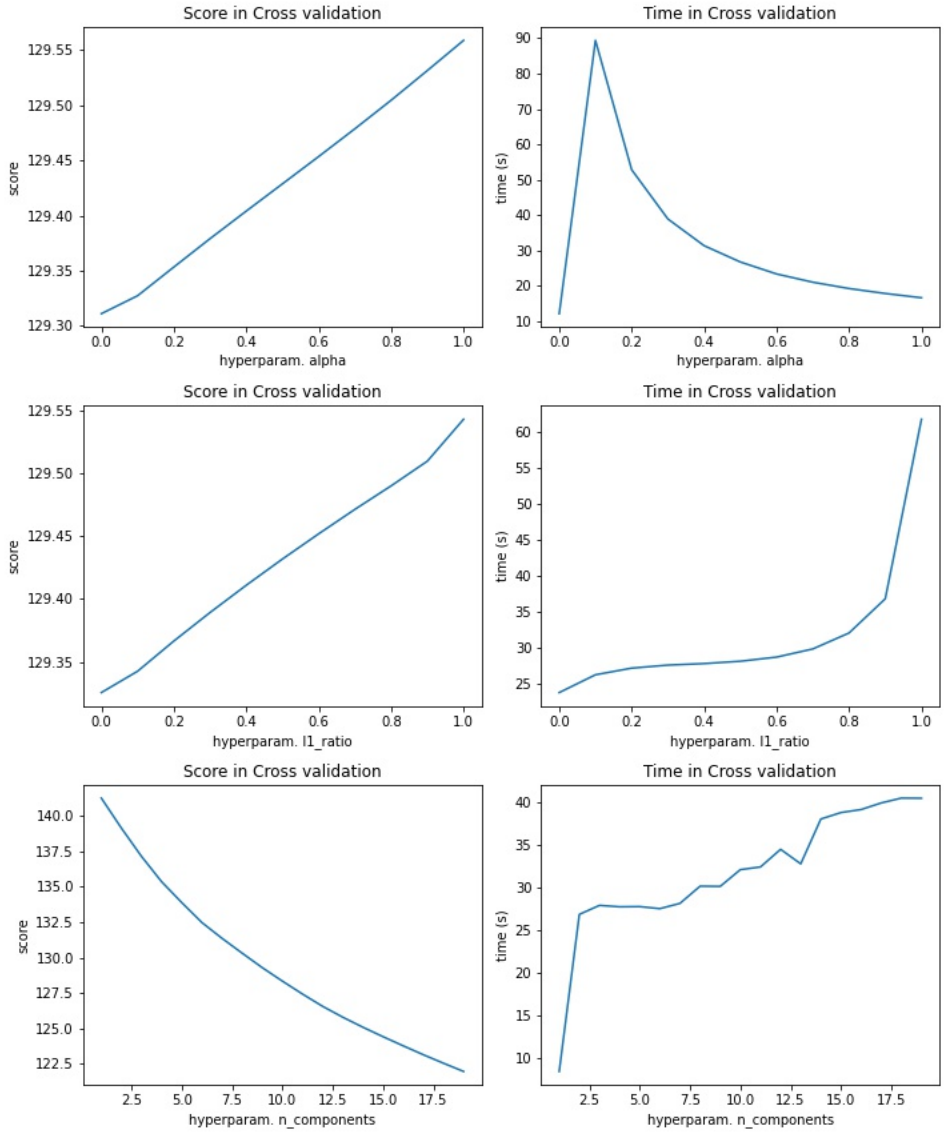


source

Modélisation :

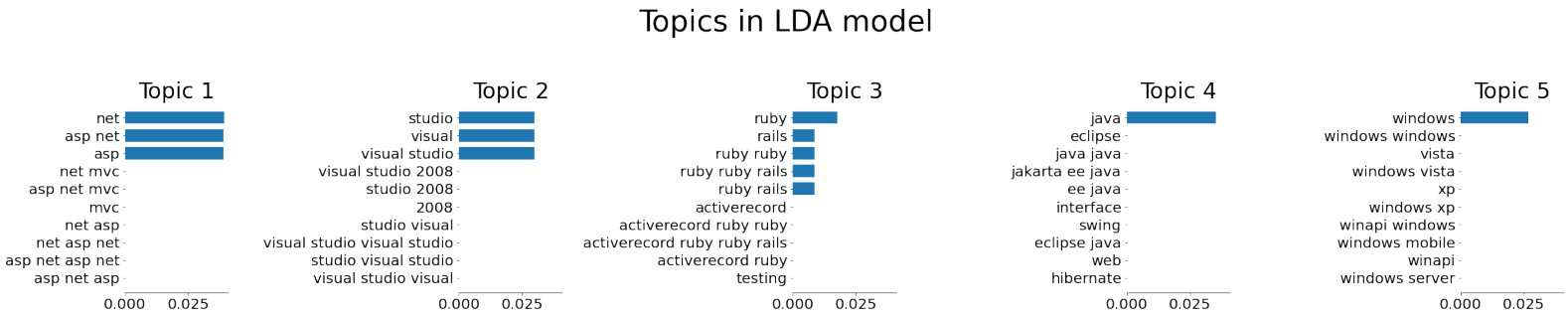
- Sur d'autres tags (`Id > 10 000`)
- Le choix des hyper-paramètres :
  - Séparation en train - validation
  - NMF en changeant : `n_componants`, `alpha`, `l1_ratio`
  - Choix des meilleurs paramètres (minimisent le score)

Tags - NMF 2

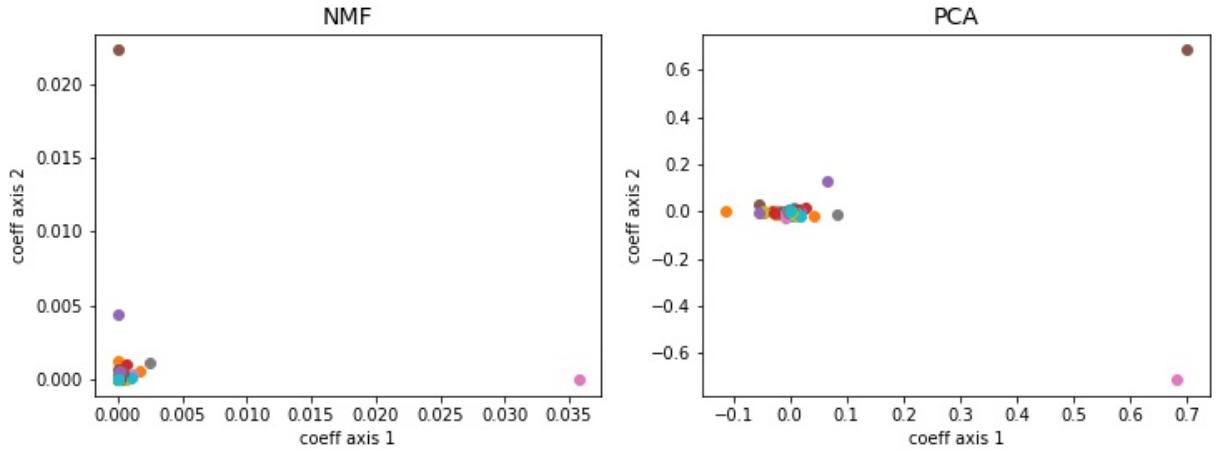


Tags - NMF 3

Les premiers topics de la NMF :



Projection sur les deux premières composantes :

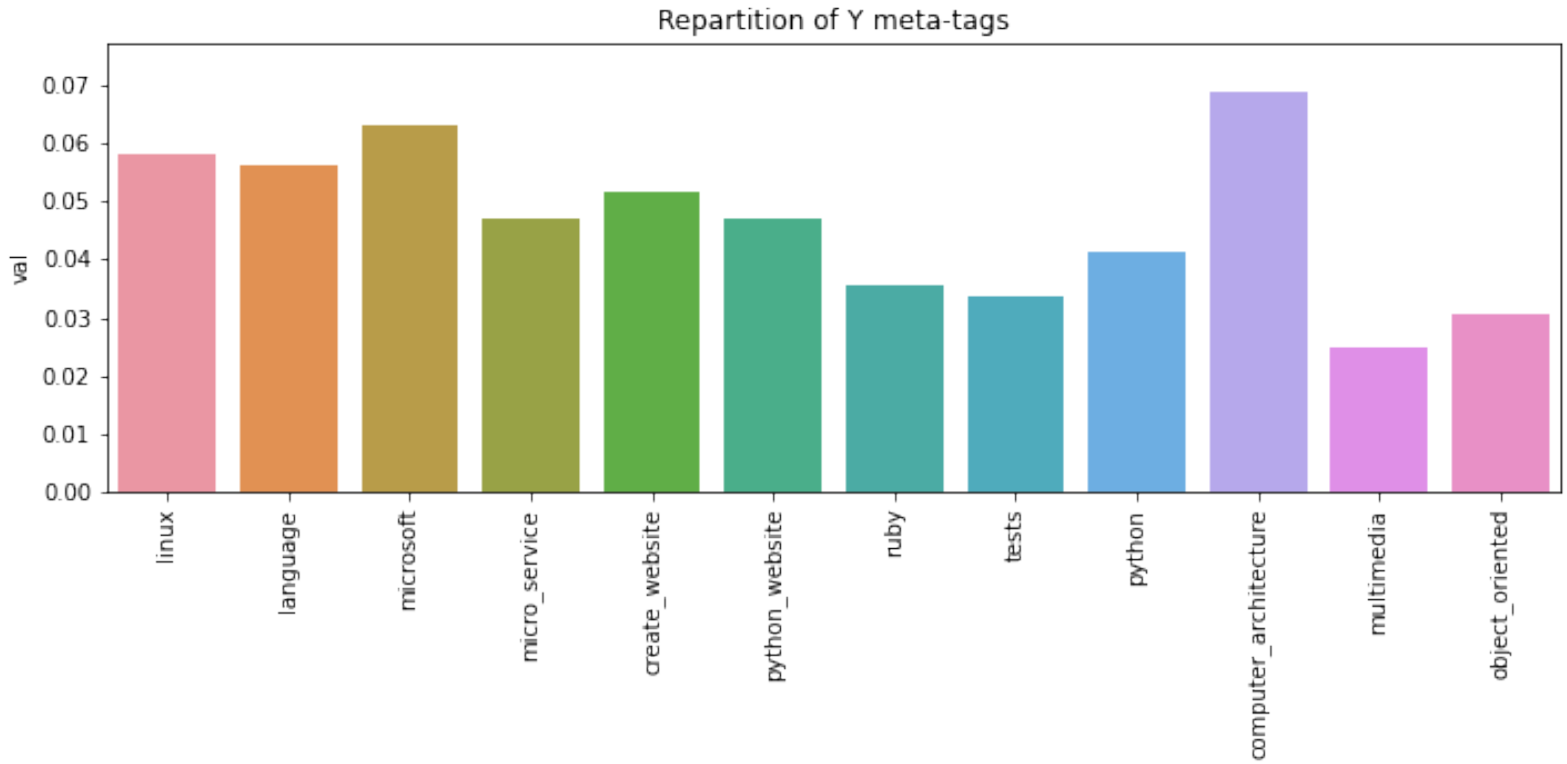


Tags - Clustering hierarchique

↪ 12 clusters nommés à partir des tag :

`linux, language, microsoft, micro_service, create_website, python_website, ruby, tests, python, computer_architecture, multimedia, object_oriented`

Tags - Répartition des clusters

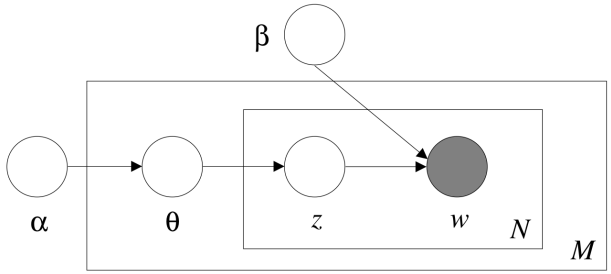


LDA et NMF

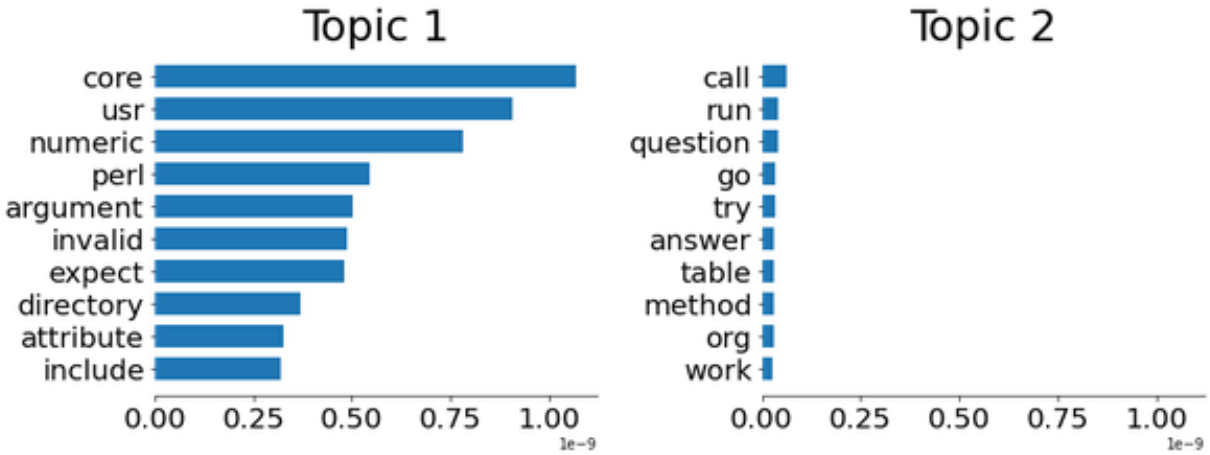
NMF :

$$X = W \times H$$

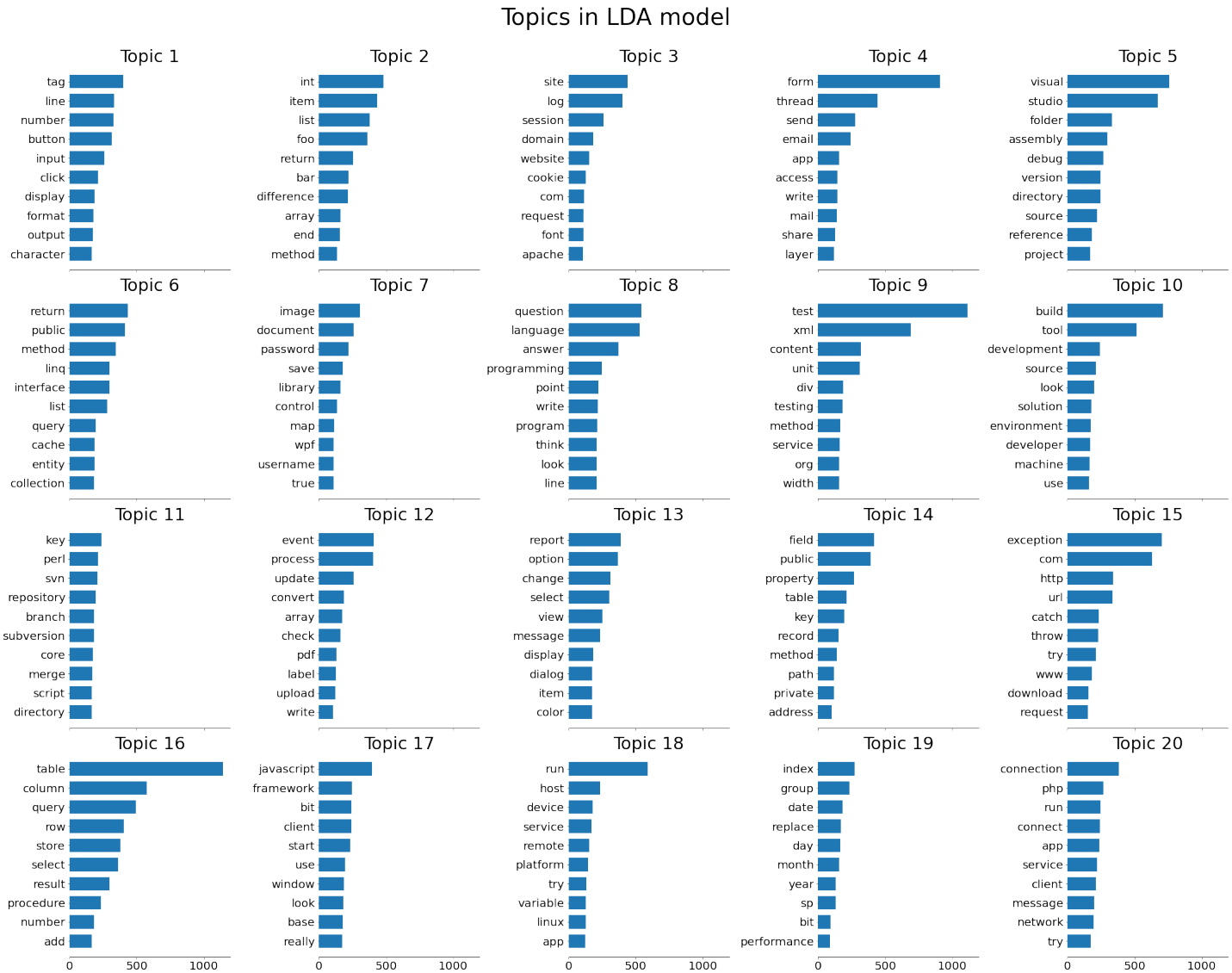
LDA :



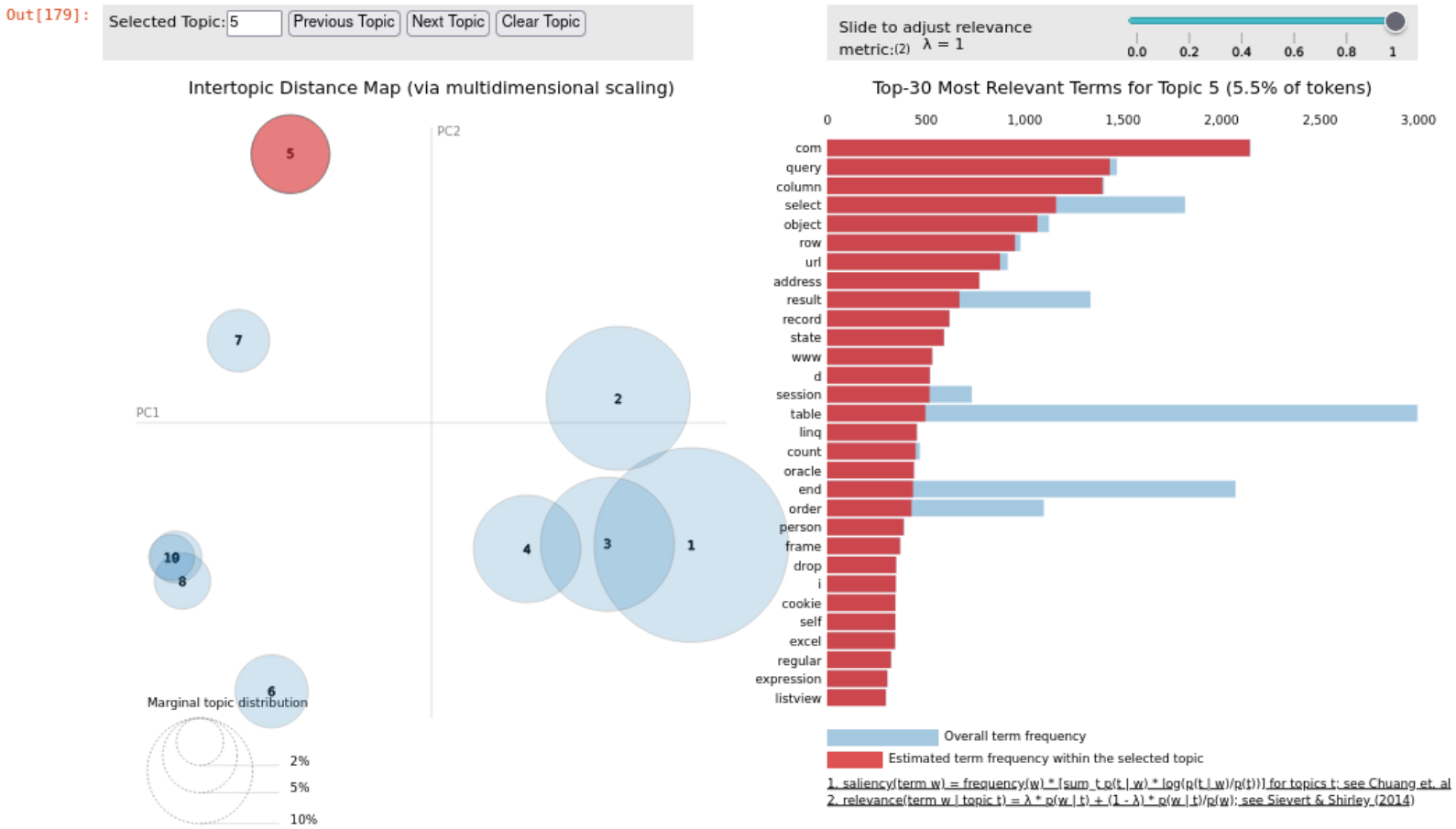
Résulats de la NMF :



LDA sur le corpus



LDA sur le corpus



Les différents modèles utilisés

- Naive Bayes : $P[A|B] = \frac{P[A \cap B]}{P[A \cup B]}$
- Gradient Boosting : min(loss logistique) ou min(loss exponentielle)
- Random Forest :

Les métriques de classification

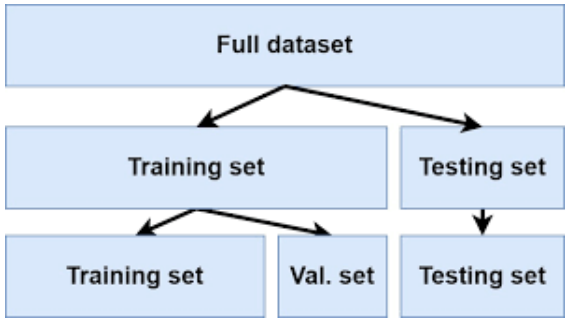
Classification binaire

- Accuracy :
- Cross-entropy de classification :

Classification multi-classe

- Accuracy :
- Cross-entropy de classification (log-loss) :

La séparation des données



Séparation des publications

- train/test
- Puis train = train/validation par validation croisée
- Corpus pré-traité

Classification binaire

- y = tag le plus courant, *csharp*
- Accuracy optimale à 1, à maximiser

Classification multi-classe

- Y = méta-tags issus de la classification hiérarchique
- Accuracy moyenne, optimale à 1, à maximiser

Résultats de la classification binaire :

Classification binaire Naive Bayes (NB) :

Loi	α	Accuracy	log-loss	AUC	Temps
Bernoulli	100	0.876	4.27	0.49	0.12s
Complement	125	0.862	4.78	0.52	0.05s
Multinomial	600	0.877	4.25	0.5	0.05s

Classification binaire Gradient Boosting :

Loss	n estimators	Accuracy	log-loss	AUC	Temps
Deviance	100	0.88	4.16	0.524	13.9s
Exponential	100	0.878	4.22	0.501	13.5s

Résultats de la classification multi-classe :

Modèle	Accuracy	log-loss	AUC	Temps
Complement NB	0.615	2.33	0.574	0.25s
Arbre décision				
gini	0.615	3.1	0.71	1.14s
entropie	0.609	3.07	0.73	1.05s
Gradient Boosting				
deviance				s
exponential				s

$\hookrightarrow$ meilleur modèle =

Analyse sur le meilleur modèle et API

Conclusion

Résumé

- Une analyse sur 3 échelles
- Deux façons de modéliser le problème
- Meilleur modèle = interprétabilité + facilité de calculs

Améliorations et suite :

- Finir de faire un script avec le meilleur modèle en PEP8
- Utiliser la table "orders" pour caractériser les clusters de clients

Merci pour votre écoute !