Catégorisation automatique des questions sur le site stack-overflow

Présenté par SEKPONA Kokou Sitsopé

Introduction

Pour poser une question sur ce site, il faut entrer plusieurs tags afin de retrouver facilement la question par la suite.
En tant que volontaire, nous voulons resoudre ce problème qui gène plus les nouveux utilisateurs.
Pour ce faire, nous développons un système de suggestion de tags pour le site. Celui-ci est un algorithme de machine learning qui assignera automatiquement plusieurs tags pertinents à une question.

Nettoyage et Analyse explratoire

Feature extractions

Modélisation

Nettoyage et Analyse exloratoire

Nettoyage

Recupération du Corpus

Nous selectionons les questions les plus vues, mises en favori ou jugées pertinentes par les internautes, ayant reçu une réponse et ayant au moins 5 tags. Avec ce bout de code fourni par openclassroms:

SELECT TOP 500000 Title, Body, Tags, Id, Score, ViewCount,
FavoriteCount, AnswerCount
FROM Posts
WHERE PostTypeId = 1 AND ViewCount > 10 AND FavoriteCount > 10
AND Score > 5 AND AnswerCount > 0 AND LEN(Tags) - LEN(REPLACE(Tags, '<',')) >= 5

	Title	Body	Tags	Id	Score	ViewCount	FavoriteCount	AnswerCount
0	`Sudo pip install matplotlib` fails to find fr...	<p>I already have <code>matplotlib-1.2.1</code...	<python><numpy><matplotlib><homebrew><osx-mave...	20572366	46	23384	20	1
1	mysqli or PDO - what are the pros and cons?	<p>In our place we're split between using mysq...	<php><mysql><pdo><mysqli><database-abstraction>	13569	342	146246	284	13
2	C char array initialization	<p>I'm not sure what will be in the char array...	<c><arrays><char><initialization><buffer>	18688971	147	748161	80	6
3	How to load plugins in .NET?	<p>I'd like to provide some way of creating dy...	<.net><windows><plugins><add-in><extensibility>	14278	27	14862	15	8
4	Increasing camera capture resolution in OpenCV	<p>In my C/C++ program, I'm using <a href="htt...	<c><image><opencv><webcam><resolutions>	14287	52	78366	26	15

Notre dataset resemble à ceci: 27977 lignes et 8 colonnes

Conversion des Tags et du text (sans caratères HTML)

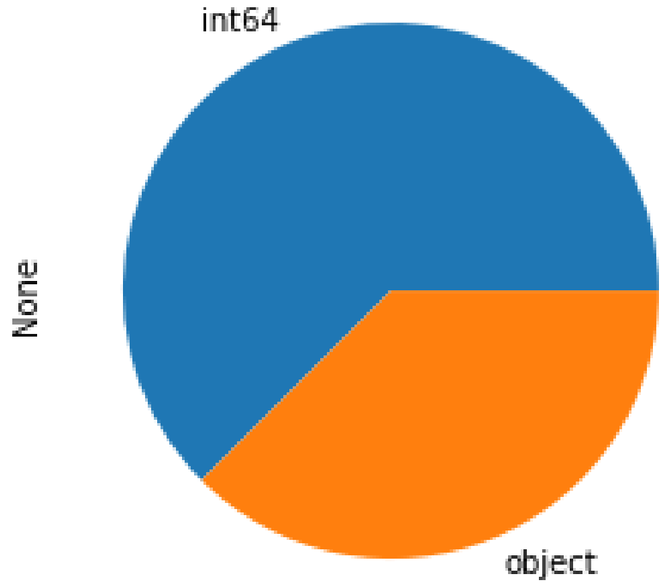
Nous traintons les text avec beautifulsup en nettoyant les caractères html

Les types

Notre dataset est constitué :

- majoritairement des types entiers: Ce sont les id, score, aswercount ...
- Aussi des types objects. Ce sont ces types qui nous sont utiles dans ce projet

Nous n'allons garder que le titre, le contenu de la question et les tags associés.

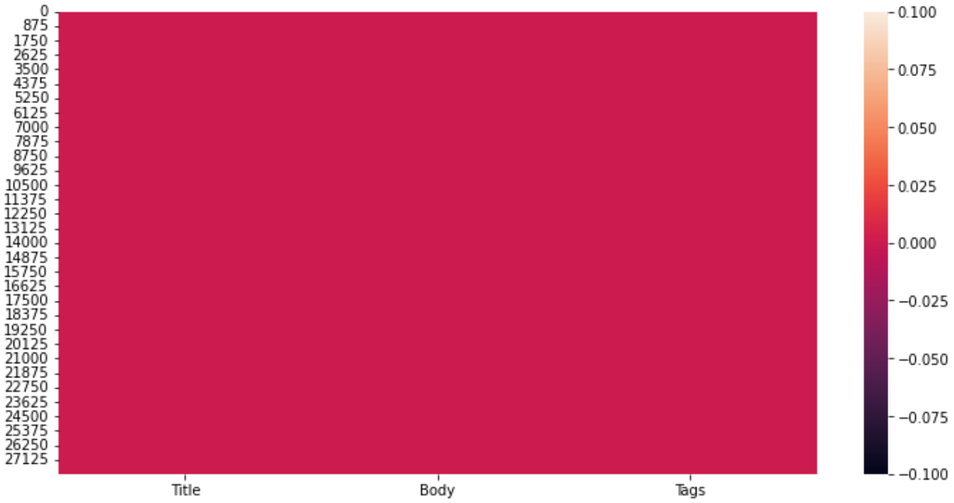


Visualisation

Par visualisation de la dataset, on peux voir que certains tags ont bien été attribués aux questions

Valeurs manquantes et doublons

Valueurs manquantes



Pas de valeurs manquantes

Doublons

```
df.duplicated(subset=['Tags']).sum()
```

```
488
```

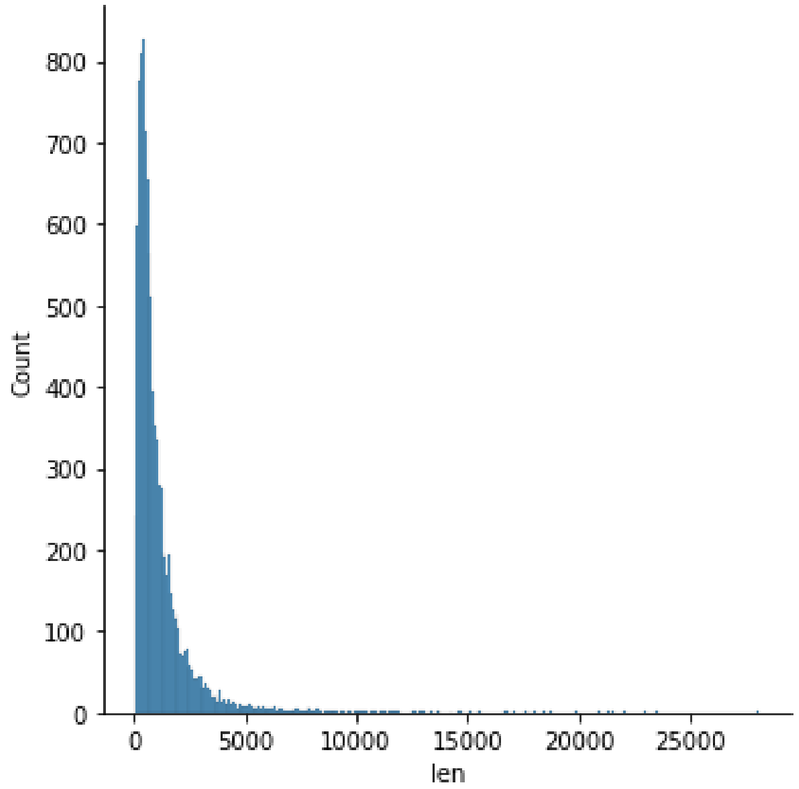
Les tags se dupliquent, ce qui est possible car plusieurs questions differentes peuvent avoir les memes tags

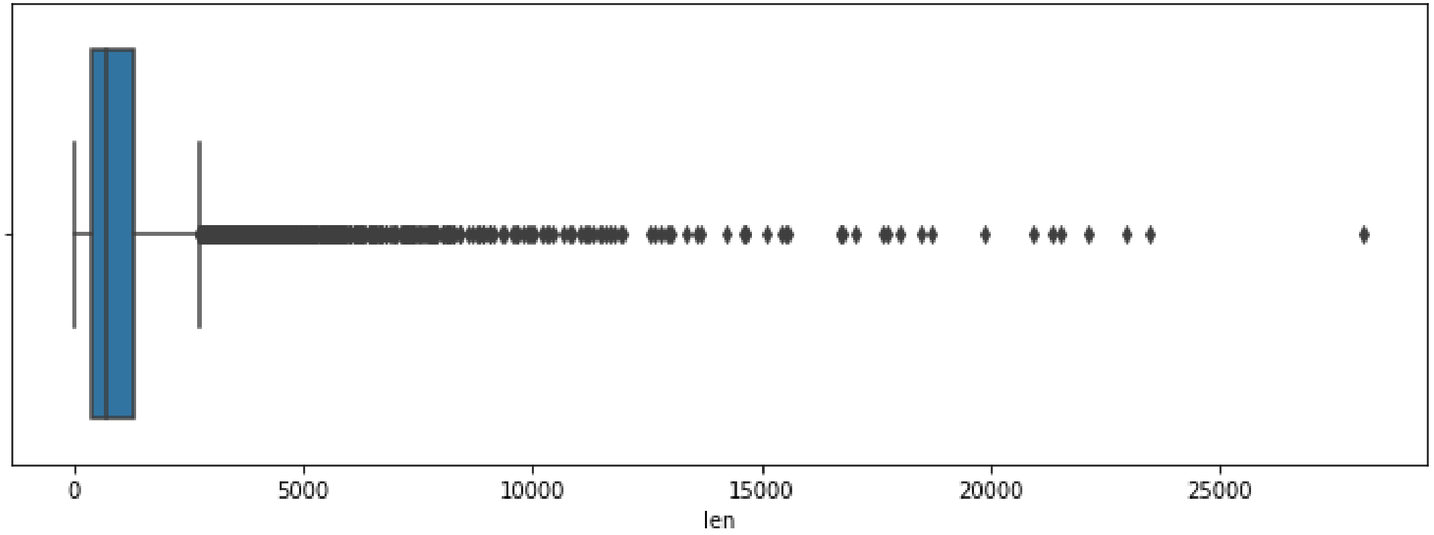
Nous constatons que nous avons 18310 tags differents, 27977 questions differentes

Creation et Nettoyage du Corpus

Visualisation et selection

Par representation des longeurs de chaque question, il est clair qu'il y a des valeurs aberrantes: Ce sont des questions qui contiennent du code et les erreurs obtenus dans la console , ce qui n'est pas trop utile pour le corpus. Aussi, laplupartt des textes ont des longueurs inférieurs à 5000. Nous n'allons conserver que les questions ayant au plus 5000 mots.





Traitement du Corpus

Tokenisation

Nous transformons nos documents en tokens

Lemmatisation/ Stemming

Nous essayons les methodes lemmatisation et stemming. Nous retenons la lemmatisation car avec le stemming les mots n'ont plus de sens, il ya les problèmes de under stemming et de overstemming.

Autres correctif: Stopwords, Mots rares...

Nous ne gardons que les mots anglais, Nous supprimons les stopwords, Nous supprimons les mots rares qui n'apparaissent dans tout le corpus qu'une seule fois.

Une fois ces opérations faites, nous affichons le corpus ici sous un wordcloud qui nous aide à visualiser les mots les plus fréquents: Code, C, Data...

Nettoyage du corpus et preprocessing pour Tag

Tokenisation

Nous transformons les
en tokens

Lemmatisation

Nous utilisons la lemmatisation permettant de ne garder que les radicales ou l'infinitif des mots mais ayant un sens.
Ce la nous permettra de ne pas repeter certains tags.

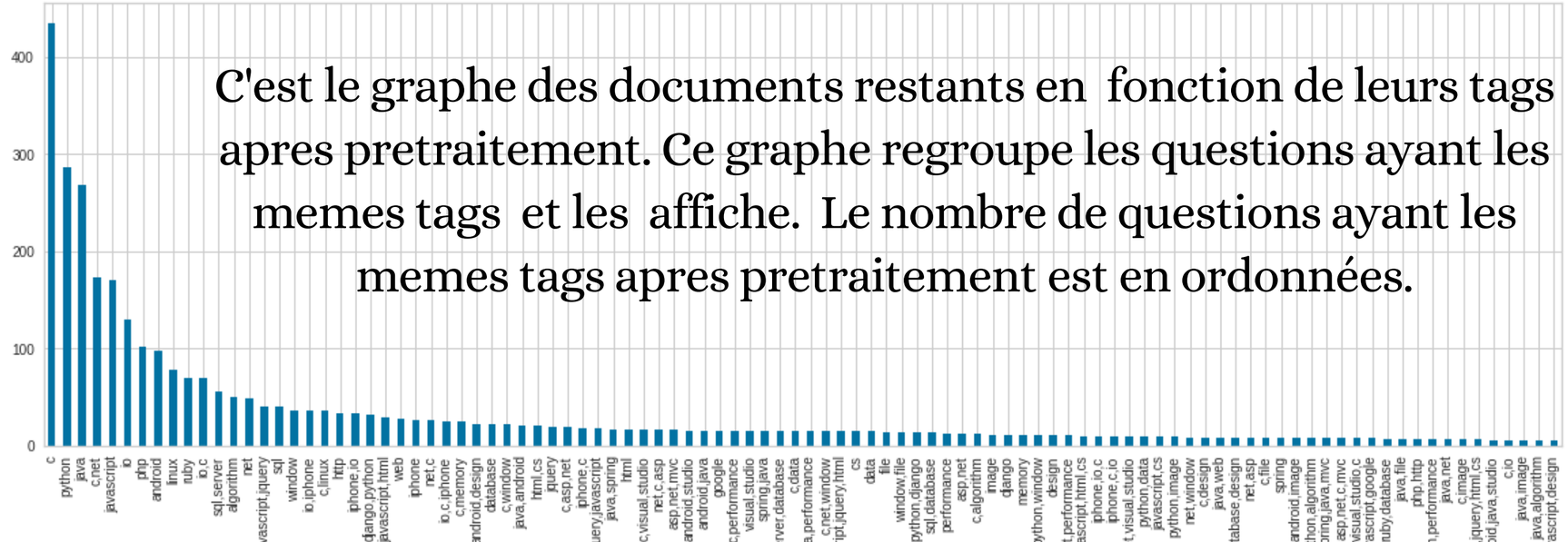
Selection de tags

Nous selectionnons les 40 Tags les plus fréquents. Au final il ne reste que 34 apres nettoyage.(Puisque certains tags sont des nombres...)
Ce serra les tags que nous allons prédire

Apres ces étapes passés par chaque tags, nous vérifions si chaque ligne contient au moins 1 de ces Tags, si c'est le cas, nous la gardons, autrement, nous la supprimons de la dataset

Filtre de la Dataframe

['c', 'net', 'python', 'java', 'android', 'javascript', 'io', 'sql', 'asp', 'jquery', 'html', 'iphone', 'php', 'database', 'window', 'server', 'ruby', 'linux', 'django', 'spring', 'studio', 'visual', 'performance', 'data', 'web', 'image', 'mvc', 'design', 'cs', 'http', 'google', 'file', 'algorithm', 'memory']. **Au total 34 Tags. Ce sont les plus fréquents. Ce sont les tags à prédire**



C'est le graphe des documents restants en fonction de leurs tags apres pretraitement. Ce graphe regroupe les questions ayant les memes tags et les affiche. Le nombre de questions ayant les memes tags apres pretraitement est en ordonnées.

Plus de 400 phrases on le Tags C, pres de 300, le Python, pres de 290, le java , ensuite le c, net...

Notre DataFrame finale a 4435 Lignes

Partie 2: Features extraction

Nous allons utiliser pluseurs methodes d'extraction de features: Use, SBert, Tfidf, Countvectorizer, Doc2vec

Bag-of-words

Nous utilions 2 methodes:

- **Tfidf**
- **CountVectorizer**

Doc2vec

Doc2vec nous permettra également de convertir nos données en features

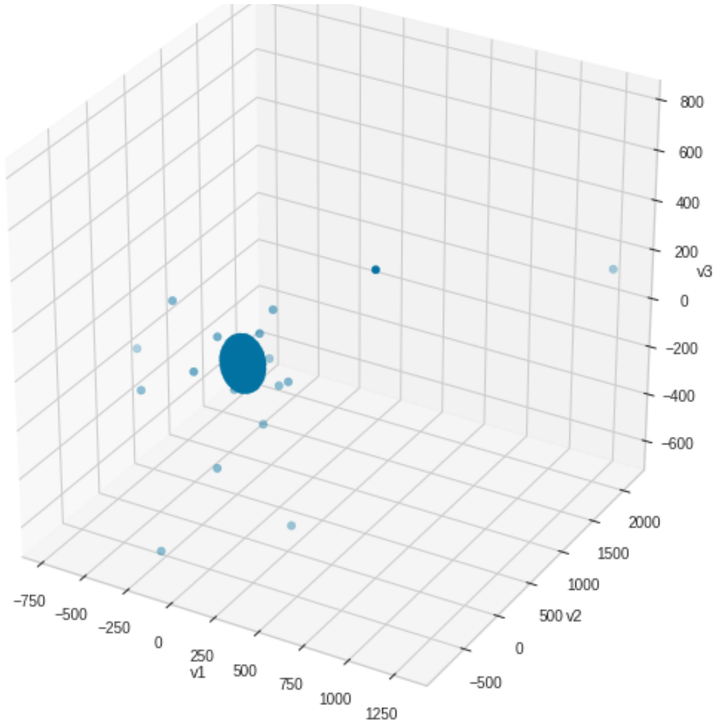
SBert

Nous utiisons Sbert pour extraire les features dans nos données

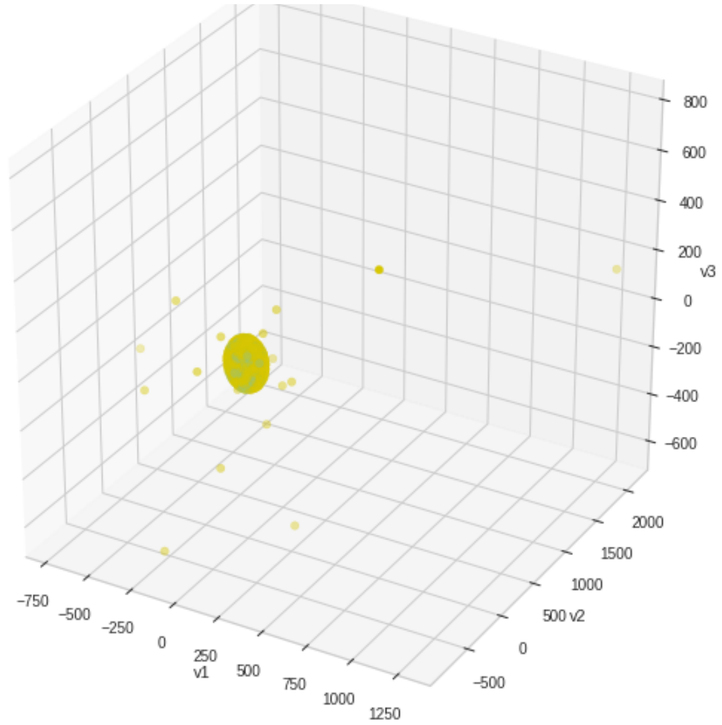
Use

Nous utilisons également le Universal Sentence Encoder

Visualisation pour Tfidf

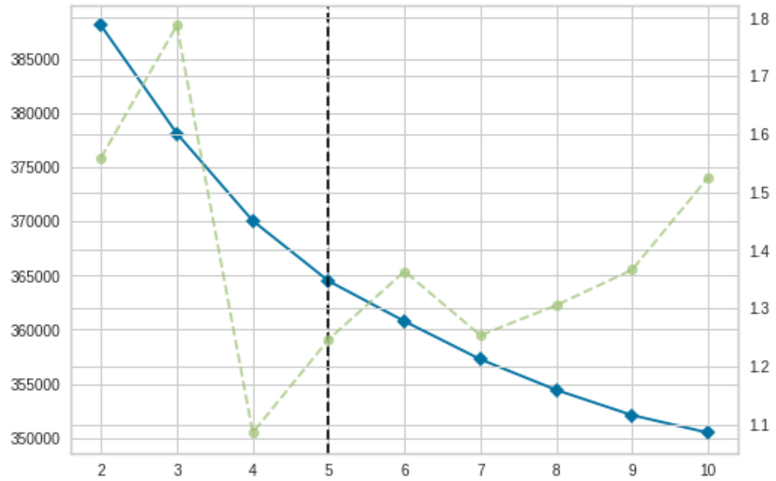


Representation Graphique en 3D avec TSNE

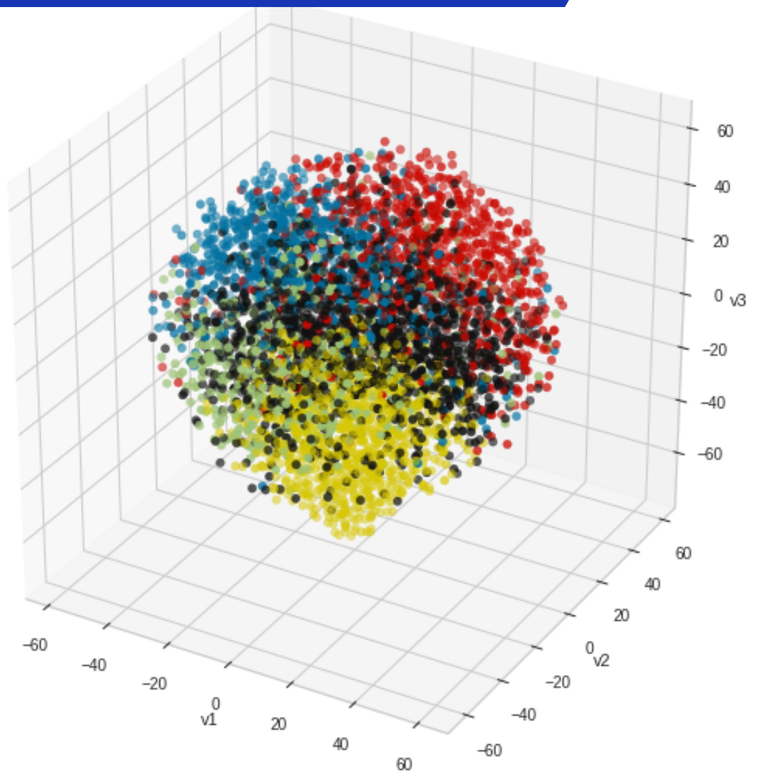


TSne et Clustering: Ici le model n'a pas pu detecter differents cluster . Il y a apparement 1 seul cluster

Visualisation de la sortie de Doc2vec

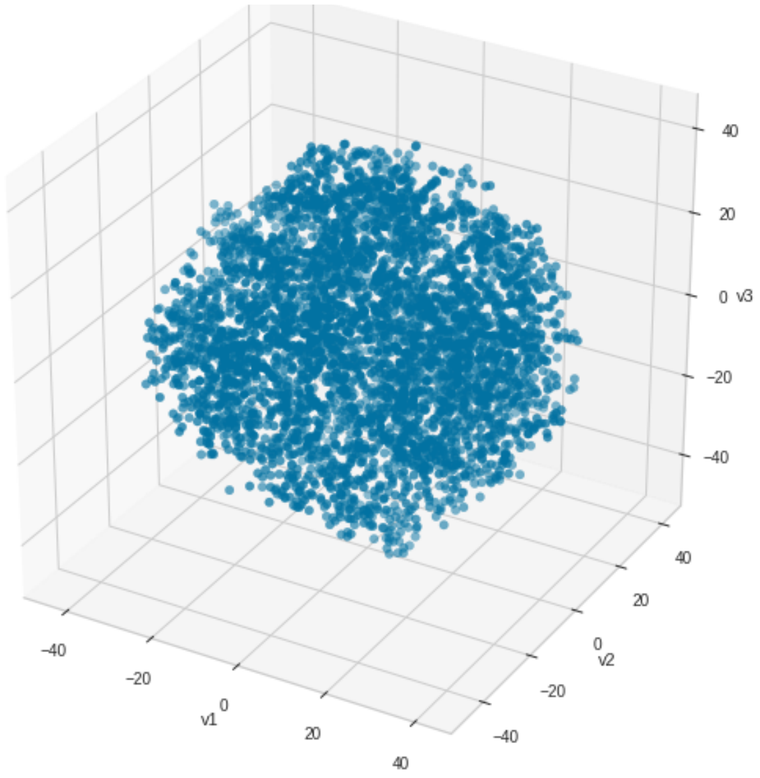


Nous obtenons le nombre de cluster égal à 5 que nous donnons au model KMeans.

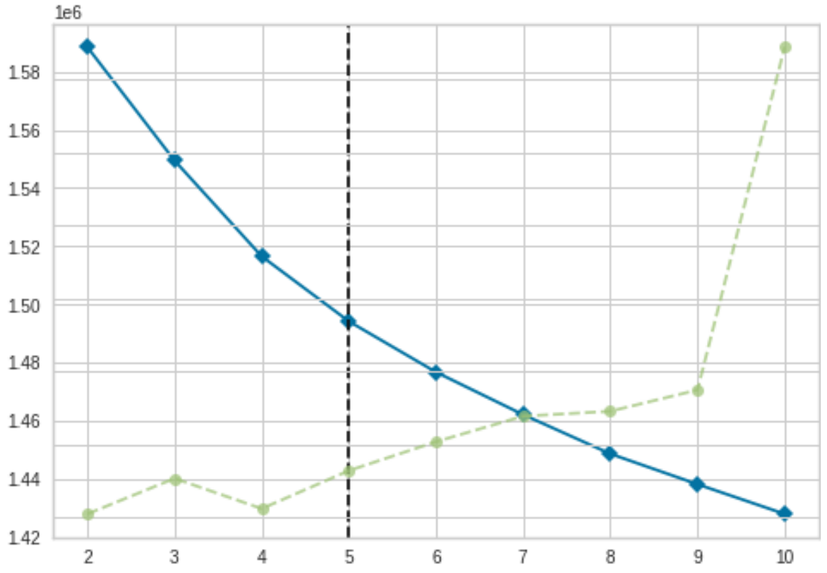


Avec Tsne, colorié suivant les clusters.

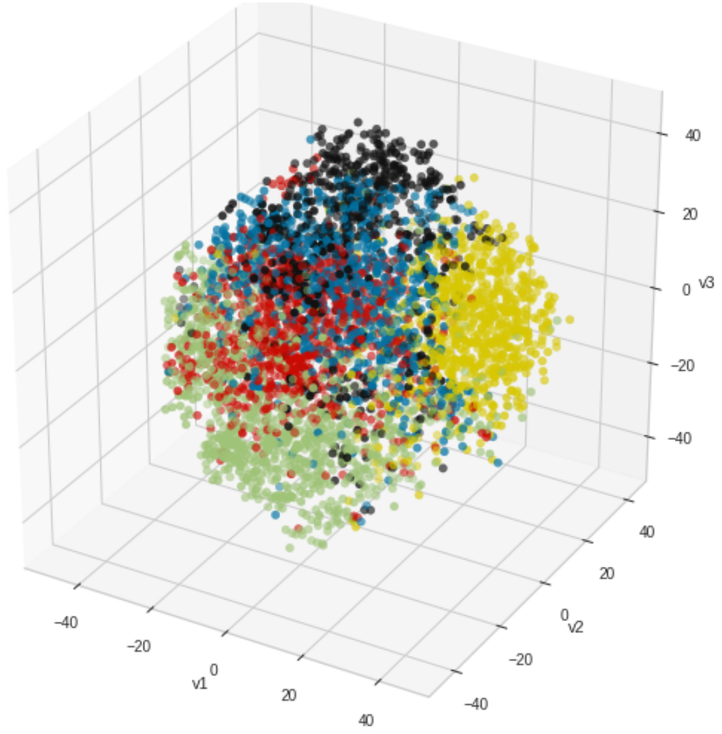
Visualisation de la sortie de SBert



Avec Tsne,Sans KMeans

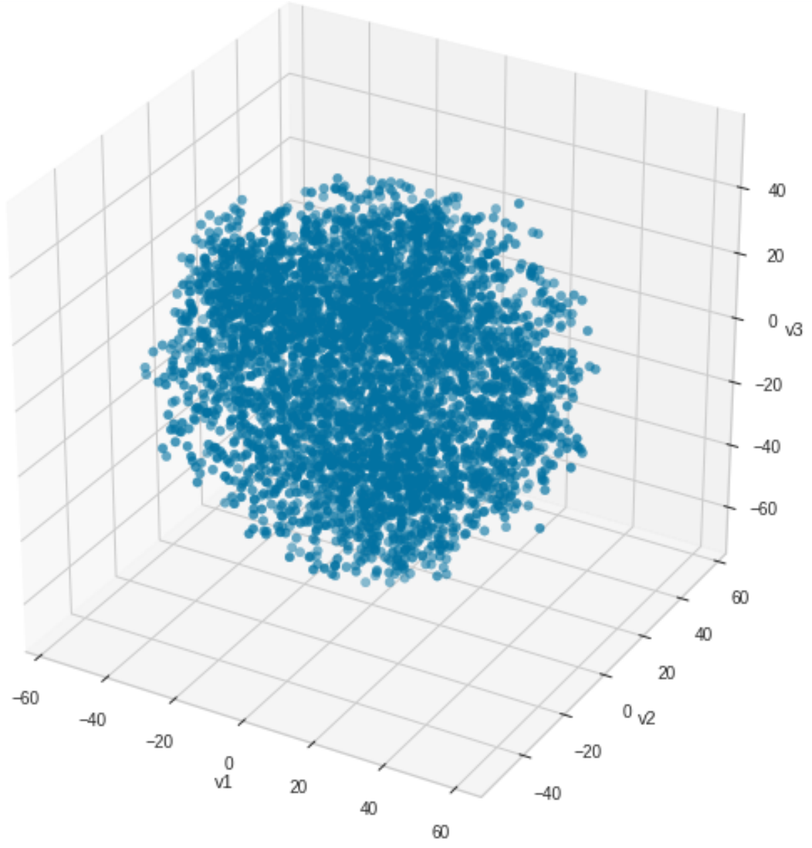


Nous obtenons le nombre de cluster égal à 5 que nous donnons au model KMeans.

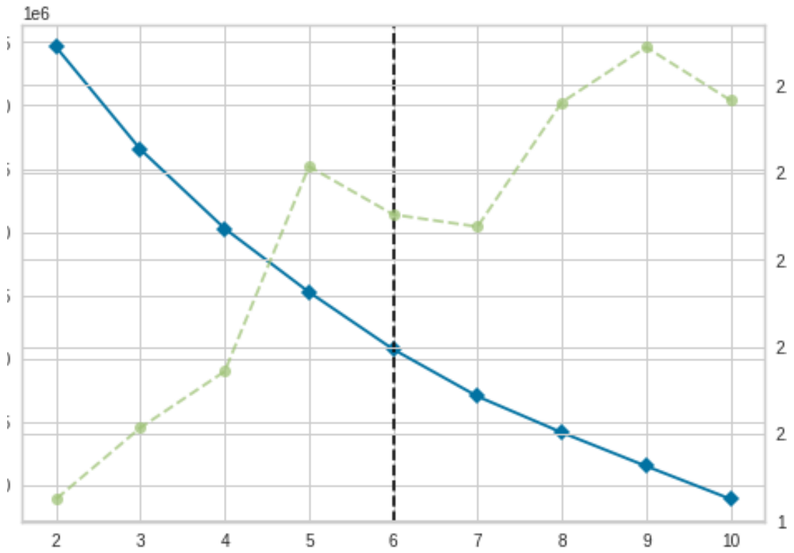


Avec Tsne, colorié suivant les clusters

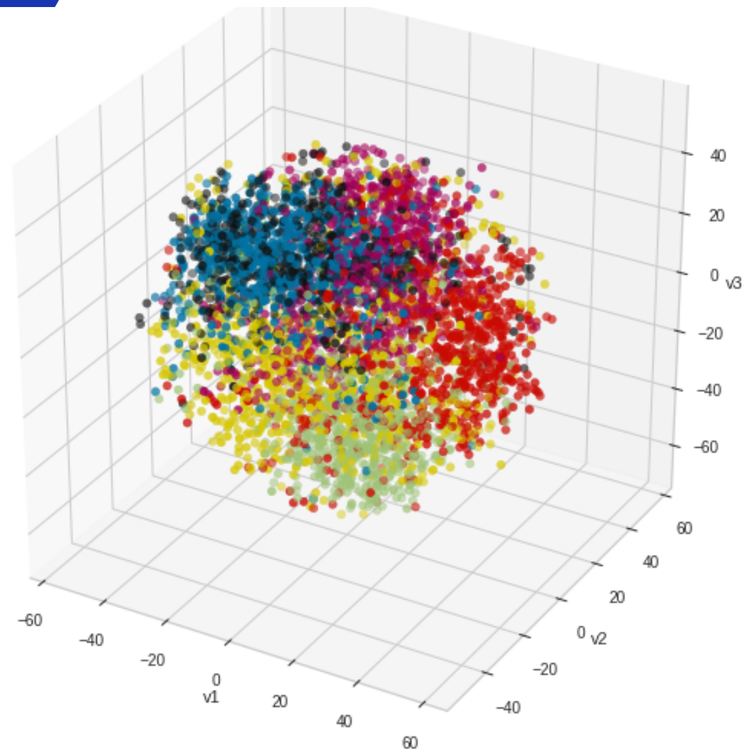
Visualisation de la sortie de USE



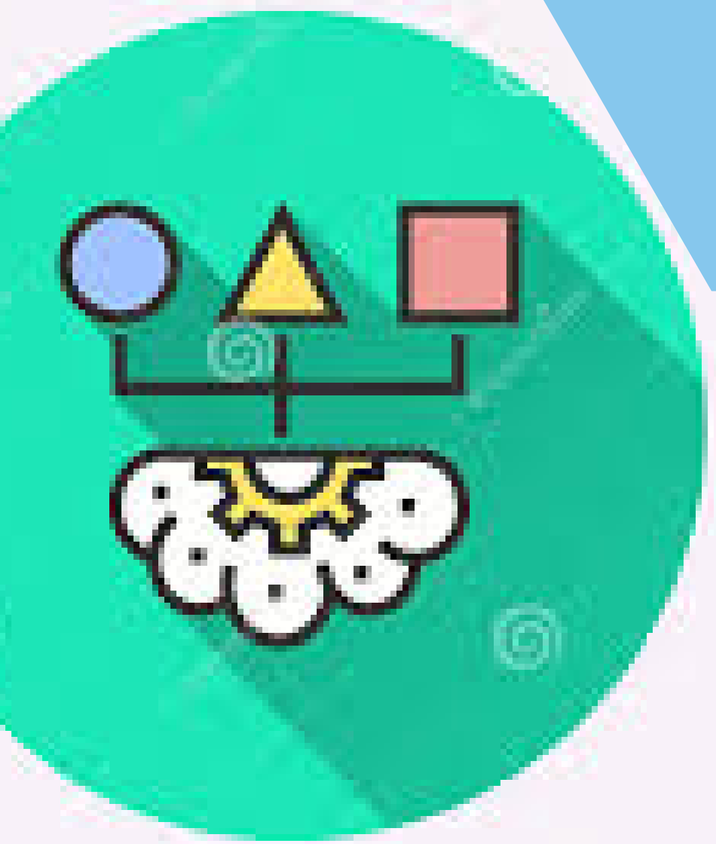
Avec Tsne,Sans KMeans



Nous obtenons le nombre de cluster égal à 6 que nous donnons au model KMeans.



Avec Tsne, colorié suivant les clusters



Partie 3: Modelisation

Nous allons modelliser par une approche non supervisée du non de LDA et plusieurs methodes d'approche supervisée: SVC, RandomForestClassifier,...

Modelisation Non supervisée

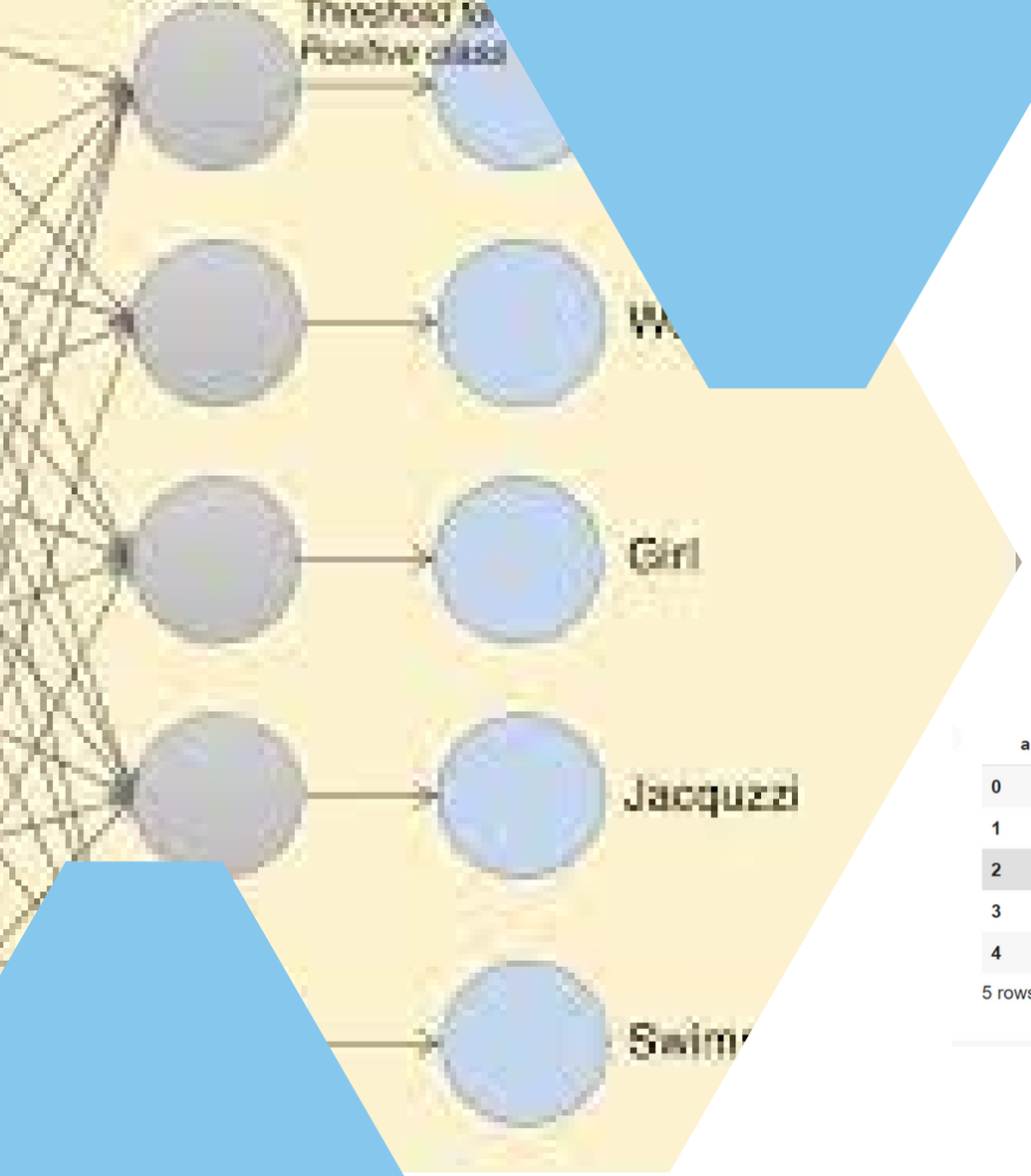
Par LDA

Modelisation supervisée

Par plusieurs methodes de classification

Choix du model

Selon la précision, la rapidité,...



Encodage des Tags avec MultiLabelEncoder

Les tags etant des text, nous devons les modeliser avec le multilabelencoder pour avoir une matrice qui serra l"etiquette pour les models d'apprentissage supervisé.

	algorithm	android	asp	c	cs	data	database	design	django	file	...	php	python	ruby	server	spring	sql	studio	visual	web	window
0	0	0	0	0	0	0	1	0	0	0	...	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	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	1
3	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 34 columns

Nous convertissons le resultat en ce dataFrame qui nous servira de Target dans l'apprentissage supervisé.

Approche Non supervisé:LDA

Avec LDA, nous arrivons à grouper les mots suivant differents sujets.
Quelques exemples:

Sujet 0: flag explain server difference perform dom valid tell evaluate current
Sujet 1: iso way sleep pass net best date time view asp
Sujet 2: explanation big notation pointer use little local difference degree anybody
Sujet 3: like use way want code work need file know new
Sujet 4: use difference python better reading people recently know specifically traditional
Sujet 5: busy process causing throughput drive device disk use file need
Sujet 6: benefit strong deal way recall type cause array scatter template
Sujet 7: use net make way bar want problem method text button

le sujet 5 a tendance à parler d'ordinateur donc proche de windows
Le Sujet 7 parle plus du developement web

L'évaluation de ce model donne un Coherence score de 0.471428280242675 7. Mais c'est plus difficle de reconnaitre les sujet avec cette methode. Nous allons essayer une approche supervisée.

Approche Supervisée

Avec Tfidf

Model	Précision	Taux de prédiction incorrect	Jaccard score
DummyClassifier	0	0.051	0
Binary Relevance	0.036	0.0497	0.0008
Classifier Chain	0.055	0.04921	0.016
Label Powerset	0.055	0.049	0.0161
KneighborsClassifier	0.1172	0.045	0.13
SVC	0.181	0.040	0.1754
LogisticRegressor	0.092	0.045	0.052

Model DeepLearning

Perte: 0.18500

Precision : 0.2501

Le model de deep learning au lieu d'etre évalué par les metrics utilisés pour les autres, a par defaut la methode evaluate lui permettant de connaitre la precision et la perte du model..

Approche Supervisée

Avec Doc2vec

Model	Précision	Taux de prédiction incorrect	Jaccard score
DummyClassifier	0	0.05198	0
RandomForestClassifier	0.025	0.052	0.00054
LogisticRegressor	0.0016	0.05213	0.0008
KneighborsClassifier	0.0083	0.055	0.003
SVC	0.026	0.059	0.010

Model de deep learning

Perte: 0.1875

Precision : 0.2700

Approche Supervisée

Avec SBert

Model	Précision	Taux de prédiction incorrect	Jaccard score
DummyClassifier	0	0.0519852	0
RandomForestClassifier	0	0.05203	0
KneighborsClassifier	0.0091	0.053	0.0024
SVC	0.013	0.060	0.008

Model Deep learning

Perte: 0.187568

Precision : 0.27

Approche Supervisée

Avec USE

Model	Précision	Taux de prédiction incorrect	Jaccard score
DummyClassifier	0	0.05152	0
RandomForestClassifier	0.1	0.045	0.06270
KneighborsClassifier	0.014	0.054	0.0058
SVC	0	0.0515203	0

Model Deep leaning

Perte: 0.18569
recision : 0.25658

Deep Learning avec APC sur TFidf en gardant 80% de la variance pour ne perdre que 20% d'information et reduire la dimension de la

Perte: 0.185280
Precision : 0.250

Le resultat est presque le meme et la précision légèrement a diminué.

Nous gardons le model de deep learning sur les donnés tfidf il est rapide à transformer les documents en vecteurs plus que tous les autres models mais aussi ce model obtient presque la meme précision sur sa sortie que sur tous les autres(sbert, use,)

Pour l'API nous allons garder le model SVC, car le reseau de neuronnes n'obtient pas de bonnes predictions sur les données de test. SVC est mieux.

Construction de L'API

Nous utilisons FastAPI pour construire l'api que nous avons déployé sur Heroku.

Phrase: hibernate test spring way bean row table find create fixture trying tell whenever insert afterwards immediately equivalent hook could add maybe another testing data rail exist set thanks integration run memory take blank constraint however good new check particular value method need creation

Merci !