**Clustering - Rapport vf**

**Introduction :**

Les données que nous avons étudiées sont composées d’articles ayant tous un contenu, pour la plupart un titre et des tags. L’objectif de ce groupe est de créer des clusters sur ces articles.

**Méthode 1 (Version 0) :**

**Méthodologie**

Le premier problème rencontré a été que nos données ne sont pas labellisées. Nous ne pouvons donc pas faire de clustering supervisé sur ces données.

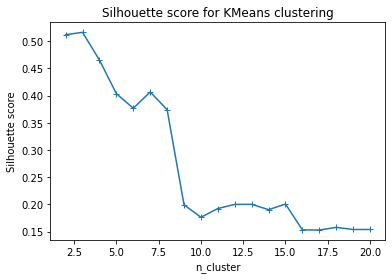
Nous avons opté pour une approche semi-supervisée. Pour cela nous avons utilisé les tags présents dans certains articles pour effectuer un premier clustering sur les articles ayant un tag. Pour ce faire nous avons calculé le Term Frequency Inverse Data Frequency (TF-IDF) sur la liste des tags. Les données n'étaient pas parfaitement propres et c’est pourquoi notre premier travail a été de faire du pré-processing sur les données. En effet, parmi les articles scrappés certains apparaissent en double, les titres ou les tags étaient souvent absents ou inexploitables (chaines vides, None,”Not Found”, List au format string).

Nous avons d’abord opté pour une approche semi-supervisée. Nous avions effectué une labellisation manuelle des articles par rapport à leurs tags et nous dégageons ainsi pour un tag, un label. Chaque document avait ainsi plusieurs labels, nous comparions celui qui apparaissait le plus de fois et le choisissons comme label final. Si deux labels apparaissent le même nombre de fois, nous regardons la similarité de chaque label avec le titre de l’article. Le problème rencontré est que tous les articles n’ont pas de tags ou de titres et parfois les tags ne sont pas exploitables (abréviation, tag qui réfèrent à plusieurs labels) et nous nous sommes orientés vers une nouvelle démarche.

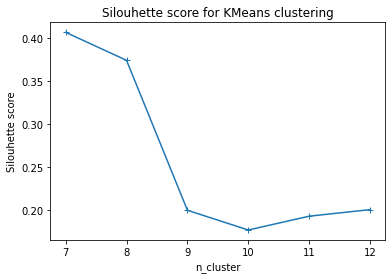
Ainsi, une fois les données plus propres nous avons calculé le TF-IDF des tags et nous avons entraîné et utilisé un algorithme de Kmeans afin d’obtenir des articles labellisés qui constitueront notre jeu de données d'entraînement. Puis, à partir de ces données labellisées nous avons entraîné un algorithme de Kmeans sur le TF-IDF, cette fois, du contenu des articles précédemment labellisés. Ce processus est semi-supervisé car nous avons nous même assigné des labels au jeu de données d'entraînement. Avec ce modèle nous avons cherché alors à prédire le contenu des articles n’ayant pas de tag, un thème.

**Résultats**

Avant de labelliser nos données “propres” on cherche d’abord le nombre de clusters optimal. Pour cela, nous utilisons la mesure de la silhouette qui consiste à calculer la différence entre la distance moyenne des documents du même cluster et la distance moyenne avec les points des autres clusters. On cherche la valeur maximisant ce score en partant du bas de la pente car sinon notre nombre de clusters n'évoluera jamais. (ici la pente se situe entre 0 et 9)



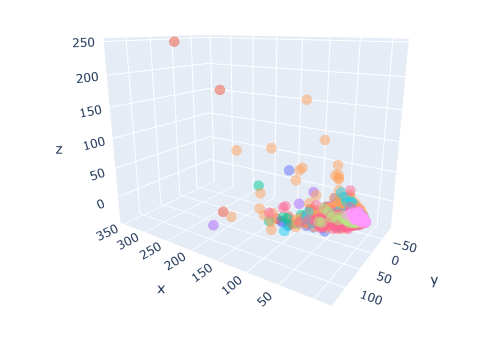
On effectue alors un zoom sur le bas de la pente (à partir de 9)pour trouver la valeur maximale du score de silhouette, voir graphique ci-dessous:

****

Le nombre optimal de clusters qui en ressort est 9.

On effectue alors notre clustering sur le TF-IDF du contenu des articles. A l’aide de l’algorithme Kmeans entraîné sur le jeu d'entraînement (articles avec tags labellisés) on prédit des labels sur le jeu de test, c'est-à- dire les articles n’ayant pas de tags.

Nous obtenons les représentations et répartitions suivantes:



| **numéro du cluster** | **nombre d’article** |
| --- | --- |
| 0 | 2758 |
| 1 | 844 |
| 2 | 479 |
| 3 | 297 |
| 4 | 639 |
| 5 | 1592 |
| 6 | 221 |
| 7 | 543 |
| 8 | 160 |

Les résultats sont assez satisfaisants d’un point de vue de la répartition des articles car elle est assez homogène dans chaque cluster. En revanche, la représentation 3D nous montre que les clusters ne regroupent pas vraiment les articles les plus proches. Pour améliorer nos résultats nous allons explorer d’autres modèles (aussi des métriques) et représentations de nos articles.

* **répartitions des articles: qualitatif niveaux résultats**
* **nuages de mots : nuages relativement proches**
* **représentation 3D pas assez hétérogènes**

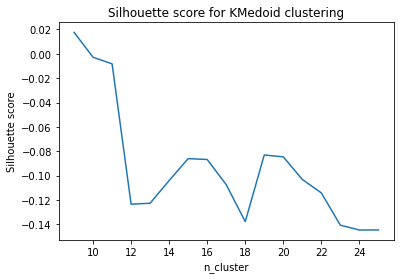
**Méthode 2 (Version 1) :**

**Méthodologie**

Pour cette version v1, la problématique est d’automatiser tout le processus pour trouver le nombre k, de clusters optimal. Il faudra également générer de manière automatique des noms de thèmes pour chaque cluster.

Nous avons d’abord cherché un nouveau modèle ainsi qu’une nouvelle représentation, plus robustes pour nos données. Notre choix après analyse (voir annexe: clustering\_model) de plusieurs modèle avec différents paramètres et représentations, s’est fixé sur le modèle des k-médoïdes en utilisant une métrique cosine(distance cosinus) sur le bow (Bag-of-Word) du groupe 3 sur lequel on a effectué une analyse en composantes principales (ACP). L’ACP permet de transformer des variables corrélées en nouvelles variables décorrélées les unes des autres

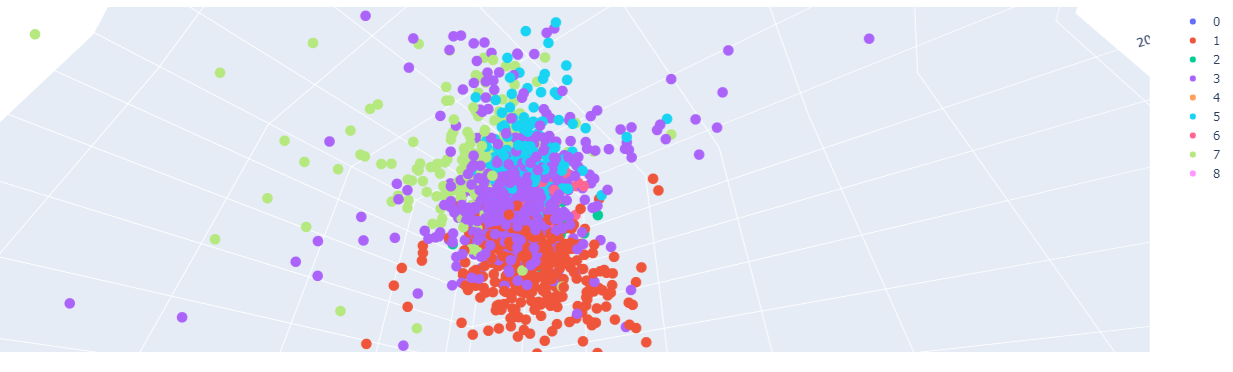
Pour trouver la valeur optimale de k nous utilisons le calcul sur le score de la silhouette. Ce dernier correspond à la différence entre la distance moyenne des documents du même cluster et la distance moyenne avec les points des autres clusters.



Le nombre optimal de clusters qui en ressort est 9.

**Résultats**

Notre modèle est le k-médoïdes avec pour metric cosine (n\_clusters=9, metric='cosine', random\_state=0 ) dont la représentation 3D est la suivante :



Cette représentation est assez bonne compte tenu que tous les articles se ressemblent, on obtient des clusters homogènes.

Dans chaque exécution du code, la numération des clusters différent, mais les tailles des clusters restent constants, la taille des clusters qu’on a obtenue est:

| **numéro du cluster** | **nombre d’article** |
| --- | --- |
| 0 | 3400 |
| 1 | 711 |
| 2 | 710 |
| 3 | 573 |
| 4 | 570 |
| 5 | 424 |
| 6 | 419 |
| 7 | 376 |
| 8 | 350 |

La répartition des articles dans chaque cluster est assez satisfaisante.

**Conclusion :**

Notre v1 est meilleure que la v0 en termes de répartition des mots dans chaque cluster, ainsi que dans la représentation 3D . Les résultats ne sont pas parfaits, nos documents sont très proches les uns des autres car ils abordent les mêmes sujets . C’est pourquoi notre représentation 3D montre un gros tas de documents et peu d’articles très différents des autres .

* **nuages de mots : nuages relativement proches**
* **représentation 3D relativement bonne**
* **répartitions des articles: qualitatif niveaux résultats**