Groupe 7 : Prédiction

BOLMIER Geoffrey , CRENNER Noémie, DELPUECH Floric, HUGUES Valentin, IZARD Damien, NJAMO KAPDEM Raphaël, OLLIVIER Cassandre, PLISSONNEAU DUQUENE Antoine, POUCHAN Marianne, SARTORI Célia

GROUPE [n°7]
Projet inter-promotion
Année universitaire : 2017/2018

S O M M A I R E

1. Objectif du groupe

L'objectif de notre groupe est de prédire les catégories auxquelles appartiennent les articles de différents journaux.
Comme il s'agit d'un problème de classification, nous avons procédé à l'aide de deux méthodes d'apprentissage : supervisée et non-supervisée. En effet, nous distinguons deux catégories différentes :

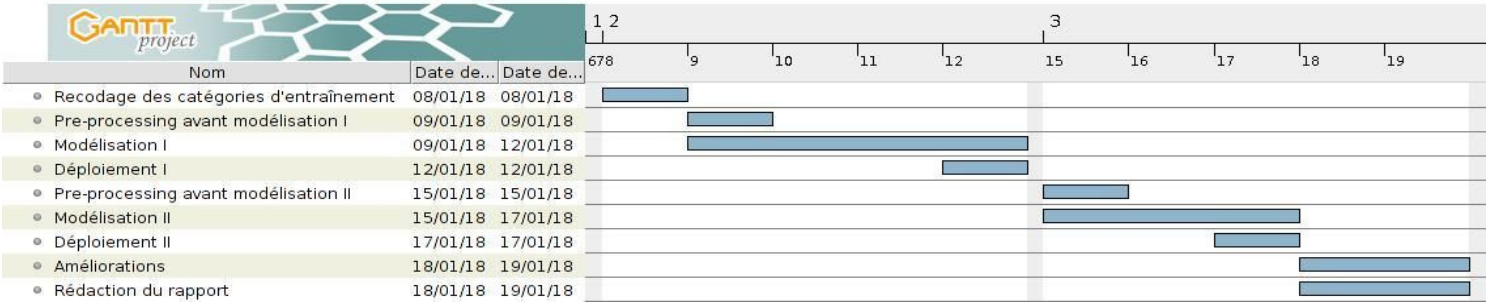
- Un ensemble de catégories thématiques (International, Economie, Sport, etc..) : chaque article peut appartenir à plusieurs catégories. Il s'agit d'un problème multi-classes, multi-labels. Pour cela nous utilisons un dataset labellisé pendant la phase d'apprentissage, c'est la méthode d'apprentissage supervisée.

- Un ensemble de catégories sémantiques : chaque article appartient juste à une seule catégorie. Il s'agit d'un problème mono-label. Pour cela nous utilisons un dataset non-labellisé dans lequel les articles sont enrichit d'informations sémantiques, c'est la méthode d'apprentissage non-supervisée.

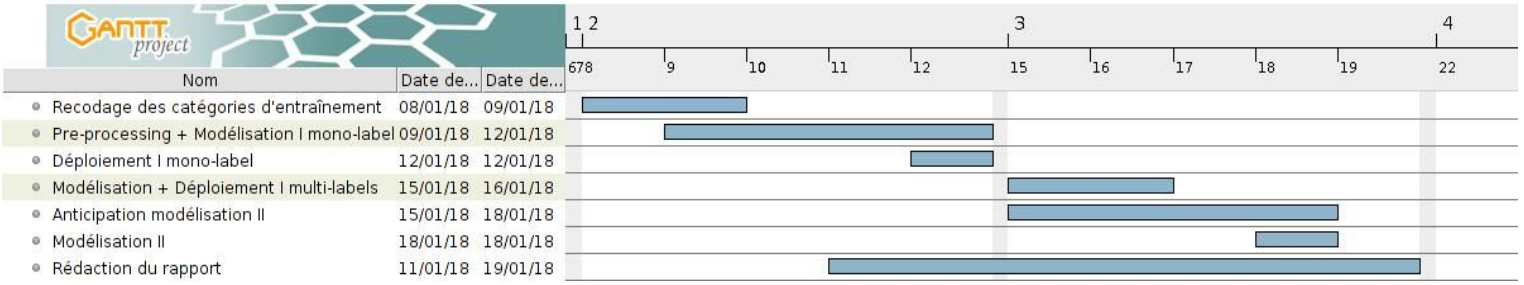
2. Organisation

2.1.Organisation temps de travail

Lors du travail préparatoire, nous avons fait un Gantt de notre travail prévisionnel.



Cependant à la fin du projet nous avons eu un Gantt :



Les différences s'expliquent par plusieurs raisons. Tout d'abord un retard dans la remise des articles du 1er robot car nous les avons obtenus juste le lundi soir à 17h. Pour nous adapter nous avons utilisé les articles que Geoffrey avait utilisé lors de son stage de L3 (une petite base de 3000 articles).
Nous n'avons pas pu faire de déploiement car il y a eu quelques problèmes avec la base de données.

Pour réaliser le travail nous nous sommes séparés en plusieurs groupes homogénéisés L3/M1/M2.

2.2 Organisation du travail dans le groupe

	Geoffrey	Noémie	Floric	Valentin	Damien	Raphael	Cassandre	Antoine	Marianne	Célia
Recodage	X	X	X	X	X	X	X	X	X	X
Pré-processing I										
TF-IDF	X								X	
Hashing	X						X			
Démarche de modélisation										
Naive Bayes					X			X	X	
Logistic Regression (SGD)	X					X				X
Linear SVM OneVSOne		X		X			X			
Linear SVM OneVSAll		X		X			X			
No Linear SVM OneVSAll				X						
Random Forest		X			X		X			
KNN			X			X				X

LGBM								X		
LDA (Linear Discriminant Analysis)					X					
Bagging / Staking			X					X		
Deep Learning	X		X					X		
Apprentissage non supervisé										
K means	X			X						X
CAH						X				
Rapport	X	X	X	X	X	X	X	X	X	X

3. Déroulement de la mission principale

Pour la v0, nous avons utilisé un filtrage basique afin de faire le classifieur sur 2000 articles. Nous avons alors mis 3 articles sur le serveur car les autres groupes n'étaient pas prêts. Nous avons utilisé la méthode mono label car le groupe bases de données n'était pas prêt.

3.1. Recodage

Nous avons commencé le projet par le recodage des catégories afin de regrouper les articles en 7 catégories.
Pour les choisir nous avons utilisées celles de Google News car elles catégorisent automatiquement les articles trouvés par ses robots sur le web.

Les catégories sont les suivantes : « international », « france », « economie », « sciences/high-tech », « arts et culture », « sports », « sante » auxquelles nous avons rajouté la catégorie « politique fr ».
Cependant par la suite nous avons retiré cette dernière car lors des prédictions certains articles de « france » se confondaient avec « politique fr », ce qui faussait les résultats.

Les thèmes/catégories sont numérotés de 0 à 6 de la façon suivante :

- international : 0
- france : 1
- economie : 2
- sciences_high_tech : 3
- arts_et_culture : 4
- sports : 5
- sante : 6

3.2. Démarche de modélisation

Après recodage et suppression des doublons grâce aux titres des articles, nous avons obtenus une base d'apprentissage de 17 031 articles. On a mis de côté 20% des articles afin de comparer les modèles (jeu de test final). Sur les 80% restants on a tuné les paramètres des différentes méthodes avec la méthode Grid Search.

3.2.1. Méthode Grid Search

La méthode Grid Search permet de savoir quels sont les paramètres optimaux pour une méthode donnée.
Nous avons une base d'apprentissage de 17 000 articles. Nous avons pris 80% des articles pour faire le jeu d'apprentissage et les 20% restants pour le jeu de test. Le grid search à été appliqué sur ces 80%. Nous avons utilisé 3 splits pour effectuer le grid search.

3.2.2. Méthode Random Search

La méthode Random Search implémente une recherche aléatoire sur les paramètres que nous mettons en entrée pour une méthode. Chaque paramètre est échantillonné à partir d'une distribution des valeurs de paramètres possibles.

Nous l'avons utilisé pour trouver le meilleur paramètre "C" dans le classifieur SVM OneVsOne. "C" étant un paramètre continu il a fallu spécifier une distribution continu pour que la randomisation soit la meilleur possible.

3.3. Pre-Processing

3.3.1. Méthode TF-IDF

La méthode TF-IDF est une méthode qui permet de calculer la fréquence des mots.
Cela signifie "Terme Frequency - Inverse Document Frequency".
Le TF permet de calculer la fréquence d'un mot dans un document, alors que IDF donne l'inverse du nombre de documents dans lequel apparaît le mot en question.
Au final, le TF-IDF est un score de fréquence de mot qui met en évidence des mots qui sont intéressants pour la prédiction. En effet, si le poid est élevé, cela signifie que le mot est important. Par exemple si le tf-idf du mot "chignon" est de 0.1 alors que celui de "sombrero" est de 0.8 donc le mot qui va nous intéresser est celui qui a le score le plus élevé.

3.3.2. Méthode Hashing

La méthode Hashing permet de réduire les dimensions du TF-IDF.
Dans notre cas le Hashing n'a pas été très utile car les résultats n'ont pas été meilleurs par rapport au TF-IDF. En effet, avec le Hashing nous perdons de l'information du fait de la réduction de dimension.

3.4. Apprentissage supervisé

Nous commençons l'apprentissage par la représentation des contraintes, et ensuite nous verrons qu'il existe deux types d'approches pour catégoriser les articles : une approche classique et une approche deep-learning.
Comme nos articles d'entrainement ne sont pas multi-labélisés, ils n'ont qu'une catégorie. C'est pourquoi nous entrainons un classifieur multi-classes, puis le modifions afin qu'il devienne multi-label.
Pour cela, nous voulons obtenir un classifieur qui prend le texte d'un article en entrée et qui renvoie les différentes catégories auxquelles il correspond avec le plus de précision possible.

3.4.1. Approche classique

- **Hashing**

Hashing en 20 dimensions

Model	Score en %	Confusion	Precision
Logistic Regression with Sarcastic Gradient Descent	34.41	[216 114 37 76 108 5 1] [136 310 47 109 237 14 1] [62 71 110 31 86 5 1] [146 161 37 119 186 4 0] [78 141 24 57 400 10 0] [46 46 10 24 60 17 0] [8 23 9 11 12 0 0]	Them Precision Recall f1-score Support 0 0.31 0.39 0.35 557 1 0.36 0.36 0.36 854 2 0.40 0.30 0.34 366 3 0.28 0.18 0.22 653 4 0.37 0.56 0.44 710 5 0.31 0.08 0.13 203 6 0.00 0.00 0.00 63 avg / total 0.33 0.34 0.33 3406
Linear SVM	34.50	[175 180 33 71 98 0 0] [102 426 36 83 207 0 0] [46 122 99 26 73 0 0] [113 252 32 93 163 0 0] [54 209 19 46 382 0 0] [36 86 6 18 57 0 0] [4 33 8 7 11 0 0]	Them Precision Recall f1-score Support 0 0.33 0.31 0.32 557 1 0.33 0.50 0.39 854 2 0.42 0.27 0.33 366 3 0.27 0.14 0.19 653 4 0.39 0.54 0.45 710 5 0.00 0.00 0.00 203 6 0.00 0.00 0.00 63 avg / total 0.31 0.34 0.32 3406

Hashing en 100 échantillons

Model	Score en %	Confusion	Precision
Logistic Regression with SGD	50.70	[212 120 23 97 89 17 0] [39 481 63 103 153 13 2] [11 54 200 52 40 9 0] [71 97 36 310 115 17 6] [29 102 41 80 436 22 0] [17 29 5 27 44 81 0] [1 11 7 28 9 0 7]	Them Precision Recall f1-score Support 0 0.56 0.38 0.45 558 1 0.54 0.56 0.55 854 2 0.53 0.55 0.54 366 3 0.44 0.48 0.46 652 4 0.49 0.61 0.55 710 5 0.51 0.40 0.45 203 6 0.47 0.11 0.18 63 avg / total 0.51 0.51 0.50 3406
Linear SVM	51.35	[277 126 21 53 75 6 0] [61 530 55 80 120 7 1] [23 67 196 41 36 3 0] [118 123 37 262 102 6 4] [49 122 37 65 425 12 0] [29 45 4 22 51 52 0] [4 13 6 27 6 0 7]	Them Precision Recall f1-score Support 0 0.49 0.50 0.50 558 1 0.52 0.62 0.56 854 2 0.55 0.54 0.54 366 3 0.48 0.40 0.44 652 4 0.52 0.60 0.56 710 5 0.60 0.26 0.36 203 6 0.58 0.11 0.19 63 avg / total 0.52 0.51 0.51 3406

Hashing en 1000 échantillons

Model	Score en %	Confusion	Precision
Logistic Regression with SGD	71.17	[424 40 8 41 40 4 0] [40 640 50 34 83 3 4] [24 34 268 17 17 6 0] [143 50 28 373 52 3 4] [48 76 24 23 530 7 2] [8 12 5 8 10 160 0] [1 12 0 1 8 3 0 29]	Them Precision Recall f1-score Support 0 0.62 0.76 0.68 557 1 0.74 0.75 0.75 854 2 0.70 0.73 0.72 366 3 0.73 0.57 0.64 653 4 0.72 0.75 0.73 710 5 0.87 0.79 0.83 203 6 0.74 0.46 0.57 63 avg / total 0.72 0.71 0.71 3406
Linear SVM	71.34	[396 56 8 53 38 5 1] [27 680 38 31 69 3 6] [18 45 254 20 24 4 1] [120 69 20 391 45 4 4] [40 92 15 34 520 6 3] [6 13 3 9 10 162 0] [1 12 0 20 3 0 27]	Them Precision Recall f1-score Support 0 0.65 0.71 0.68 557 1 0.70 0.80 0.75 854 2 0.75 0.69 0.72 366 3 0.70 0.60 0.65 653 4 0.73 0.73 0.73 710 5 0.88 0.80 0.84 203 6 0.64 0.43 0.51 63 avg / total 0.72 0.71 0.71 3406

- **TF-IDF**

Notre texte est considéré comme une liste de mots non ordonnée auxquels on attribue des poids. Pour cela nous avons utilisé la méthode du tf-idf.

Model	Score en %	Confusion	Precision
Naive bayes	72.75	[441 26 6 35 45 2 2] [36 644 49 39 69 1 16] [10 36 275 21 23 1 0] [183 37 23 356 42 3 9] [41 75 18 23 539 8 6] [7 7 3 2 10 174 0] [0 5 0 6 3 0 49]	precision recall f1-score support 0 0.61 0.79 0.69 557 1 0.78 0.75 0.76 854 2 0.74 0.75 0.74 366 3 0.74 0.55 0.63 653 4 0.74 0.76 0.75 710 5 0.92 0.86 0.89 203 6 0.60 0.78 0.68 63 Avg/total 0.74 0.73 **0.73** 3406
Logistic regression	78.74	[454 26 4 44 22 7 0] [21 715 35 28 48 1 6] [12 34 286 16 17 1 0] [103 51 15 443 32 3 6] [40 67 8 27 560 6 2] [5 6 3 2 6 181 0] [1 8 0 9 2 0 43]	Them precision recall f1-score support 0 0.71 0.82 0.76 557 1 0.79 0.84 0.81 854 2 0.81 0.78 0.80 366 3 0.78 0.68 0.73 653 4 0.82 0.79 0.80 710 5 0.91 0.89 0.90 203 6 0.75 0.68 0.72 63 Avg/ total 0.79 0.79 **0.79** 3406
Linear SVM one vs one	78.51	[440 31 4 57 19 6 0] [14 719 29 37 47 0 8] [8 42 278 16 21 1 0] [96 47 13 468 20 2 7] [36 79 8 32 542 11 2] [6 5 1 2 5 184 0] [1 8 0 9 2 0 43]	Them precision recall f1-score support 0 0.73 0.79 0.76 557 1 0.77 0.84 0.81 854 2 0.83 0.76 0.80 366 3 0.75 0.72 0.73 653 4 0.83 0.76 0.79 710 5 0.90 0.91 0.90 203 6 0.72 0.68 0.70 63 avg / total 0.79 0.79 **0.78** 34

Linear SVM one vs all	78.62	[438 28 5 55 24 7 0] [18 18 33 30 48 3 4] [13 39 283 16 14 1 0] [96 54 12 450 29 4 8] [35 70 7 29 557 10 2] [3 5 1 2 3 189 0] [1 9 0 8 2 0 43]	Them precision recall f1-score support 0 0.73 0.79 0.75 557 1 0.78 0.84 0.81 854 2 0.83 0.77 0.80 366 3 0.76 0.69 0.72 653 4 0.82 0.78 0.80 710 5 0.88 0.93 0.91 203 6 0.75 0.68 0.72 63 avg / total 0.79 0.79 **0.79**
No Linear SVM one vs all	72.94	[404 66 7 27 51 3 0] [14 721 30 31 55 3 0] [5 66 257 17 20 1 0] [148 81 18 342 53 10 1] [25 79 8 12 578 8 0] [0 7 1 4 16 1750] [4 12 3 33 3 0 8]	Them Precision Recall f1-score Support 0 0.67 0.72 0.70 558 1 0.70 0.84 0.76 854 2 0.79 0.70 0.74 366 3 0.73 0.52 0.61 653 4 0.74 0.81 0.78 710 5 0.88 0.86 0.87 203 6 0.89 0.13 0.22 63 Avg /total 0.74 0.73 **0.72** 3407
Random forest	75.16	[455 49 1 11 35 6 0] [20 745 12 12 62 3 0] [12 76 245 10 22 1 0] [140 82 10 374 45 2 0] [33 101 5 10 554 7 0] [9 12 4 1 18 159 0] [1 13 0 20 1 0 28]	Them precision recall f1-score support 0 0.68 0.82 0.74 557 1 0.69 0.87 0.77 854 2 0.88 0.67 0.76 366 3 0.85 0.57 0.69 653 4 0.75 0.78 0.77 710 5 0.89 0.78 0.83 203 6 1.00 0.44 0.62 63 Avg/total 0.77 0.75 **0.75** 3406

20-NN	73.51	[460 31 1 46 17 2 0] [39 693 35 38 36 4 9] [16 44 257 34 14 1 0] [170 56 16 386 19 0 6] [57 86 15 38 498 12 4] [12 6 1 4 2 178 0] [1 14 1 12 3 0 32]	Them precision recall f1-score support 0 0.61 0.83 0.70 557 1 0.75 0.81 0.78 854 2 0.79 0.70 0.74 366 3 0.69 0.59 0.64 653 4 0.85 0.70 0.77 710 5 0.90 0.88 0.89 203 6 0.63 0.51 0.56 63 Avg/ total 0.75 0.74 **0.73** 3406
LGBM	80.21	[447 30 3 47 22 7 1] [21 716 25 37 48 6 1] [15 45 274 15 15 2 0] [61 52 12 494 30 3 1] [28 74 9 16 573 9 1] [4 7 3 0 4 185 0] [0 10 0 9 1 0 43]	Them precision recall f1-score support 0 0.78 0.80 0.79 557 1 0.77 0.84 0.80 854 2 0.84 0.75 0.79 366 3 0.80 0.76 0.78 653 4 0.83 0.81 0.82 710 5 0.87 0.91 0.89 203 6 0.91 0.68 0.78 63 avg / total 0.80 0.80 **0.80** 3406
LDA(Linear Discriminant Analysis)	62.	[347 53 25 88 37 7 0] [57 576 61 74 77 2 7] [13 49 237 35 29 2 1] [103 78 32 355 63 8 14] [72 109 47 63 393 16 10] [8 12 3 5 12 163 0] [2 7 1 6 4 0 43]	Them Precision Recall f1-score Support 0 0.58 0.62 0.60 557 1 0.65 0.67 0.66 854 2 0.58 0.65 0.61 366 3 0.57 0.54 0.56 653 4 0.64 0.55 0.59 710 5 0.82 0.80 0.81 203 6 0.57 0.68 **0.62** 63 avg / total 0.62 0.62 **0.62** 3406

Le score en pourcentage correspond à la précision c'est à dire le nombre de prédictions correctes. C'est sur ce facteur que nous déciderons quel modèle nous allons garder.

La matrice de confusion permet d'évaluer si la classification est de bonne qualité. Elle se construit en comparant respectivement les données de références et les données obtenues lors de la classification. Elle permet de calculer plusieurs indicateurs comme la précision.

Pour évaluer nos classifications, nous utilisons 2 autres indicateurs le rappel et le F1-Score. Le rappel correspond aux taux de vraie positif sur le taux de vraie positif et de faux négatif. C'est la proportion d'éléments bien classés par rapport au nombre d'éléments de la classe à prédire. Le F1-Score mesure la performance du système.

Suite à ces résultats nous avons choisi de garder le modèle Linear LGBM one vs all, car il s'agit de celui qui prédit le meilleur score.
Lors de la V1, par souci de performance nous avons choisi le modèle Naive Bayes.

3.4.2. Bagging et stacking

En vue d'améliorer notre score et la stabilité de notre modélisation, nous avons mis en place plusieurs algorithmes d'agrégation de modèles.

Bagging:
Le Bagging consiste à sous-échantillonner le jeu d'apprentissage dans un premier temps. Ensuite on applique à chaque sous-échantillon un algorithme voulu de machine learning, une fois les modèles entraînés, on applique un système de votant majoritaire. Cette méthode a pour avantage d'avoir une meilleure stabilité sur nos résultats.

Stacking:
Le Stacking est une méthode (Meta-modèle) qui permet d'appliquer plusieurs classifieurs de machine learning afin d'enrichir les données. Ensuite le principe est de donner un poids à chaque classifieurs selon son niveau de performance. Cela permet donc d'utiliser les meilleurs modèles pour une situation donnée: Agrégation par xgboost

Cette méthode qui va permettre de prédire la catégorie d'un article en enrichissant les données via les sorties des autres classifieurs. Au final, ce meta-classifieur va agréger les modèles en prenant le meilleur modèle pour chaque situation.

Score = 0.793 Bagging

3.4.3. Deep Learning

.
Nous avons essayé de voir si des méthodes de deep learning pouvait nous permettre d'améliorer notre score. Plusieurs méthodes ont été utilisées:

- Embedding layer /LSTM: Une première couche d'embedding avec la librairie Word2Vec permettant de rattacher chaque mot à un concept, et cela sous forme d'un vecteur. Ensuite, une seconde couche appelée LSTM a pour principe de garder en mémoire les mots vectorisés précédents ce qui, dans une démarche de reconnaissance de langage naturel, où une phrase (ou un article) a du sens non uniquement par la présence des mots le composant mais aussi par leur ordre, est très adapté pour notre problème.

-Embedding layer/GRU: Similaire à la première méthode, à l'exception de la couche LSTM remplacée par une couche Gated Recurrent Unit (GRU) qui est une alternative au Long Short Term Memory (LSTM), cette méthode permet de prendre en compte moins de paramètres. Il est admis que c'est une méthode ayant fait ses preuves dans un premier temps sur des données de type sons, mais également sur des données textuelles.

-Embedding layer/Convolutional layer/LSTM:

Une couche dense à 7 neurones en sortie du réseau permet ensuite au modèle de prendre la décision sur la catégorie à prédire.

Malheureusement, les résultats n'ont pas dépassé les méthodes de boosting, et au vue de la complexification du réseau à produire (Ajout de multiples couches denses) pour espérer améliorer les résultats, nous avons décidé d'abandonner l'idée du deep learning.

3.4.4. Solution choisie: Light GBM

Finalement, nous avons choisi la solution nous donnant les meilleurs résultats et cela avec des performances satisfaisantes. Il s'agit d'une librairie de gradient boosting créée par Microsoft© ayant des résultats similaire voire légèrement supérieurs à celles de la librairie XGboost mais avec un rapidité d'exécution bien supérieure. Cette solution nous a donné 80% en f1-score en mono-label. Ensuite avec la version multi-label, nous avons pu constater que 90% du temps, la catégorie étiquetée (y) faisait partie des catégories prédites en multi-label.

3.5. Apprentissage non supervisé

Pour effectuer un clustering qui va nous permette de définir un nouvel ensemble de catégories sémantiques nous nous sommes servis du travail du groupe 6.
Nous avons obtenu les informations sémantiques au niveau des documents sous forme de taux : la positivité, la négativité et différentes émotions telles que la joie, la peur, la tristesse, la colère, la surprise, le dégoût et la subjectivité (cf. image).


```
"article":
   {
        "id_article":4,
        "rate_positivity": 0.5,
        "rate_negativity": 0.5,
        "rate_joy": 0.5,
        "rate_fear": 0.5,
        "rate_sadness": 0.5,
        "rate_angry": 0.5,
        "rate_surprise": 0.5,
        "rate_disgust": 0.8
        "rate_subjectivity" : 0.4,
        "is_positive" : True
    },
```


L'objectif est de trouver des catégories différentes des premières catégories prédites lors de l'apprentissage supervisé (ex : Article sarcastique, subjectif, joyeux,...).

Au départ, nous avions un grand nombre de dimensions caractérisant les vecteurs tf-idf des articles. Nous avons donc décidé de faire une ACP afin de réduire le nombre de dimensions. Nous avons choisis de réduire à 9 dimensions pour que les informations sémantiques aient autant d'importance que les vecteurs tf-idf des articles. Nous avons également décidé de prendre en compte les catégories initiales des articles sous forme de vecteur binaire selon la ou les catégories correspondantes aux articles.

Ainsi, nous avons pu agréger les vecteurs tf-idf aux informations sémantiques et aux 6 catégories.

À partir de là, nous avons utilisé les méthodes classiques de clustering :

- K-moyennes (k-means)
- Classification Ascendante Hiérarchique (CAH)

3.5.1. Algorithme des k-means

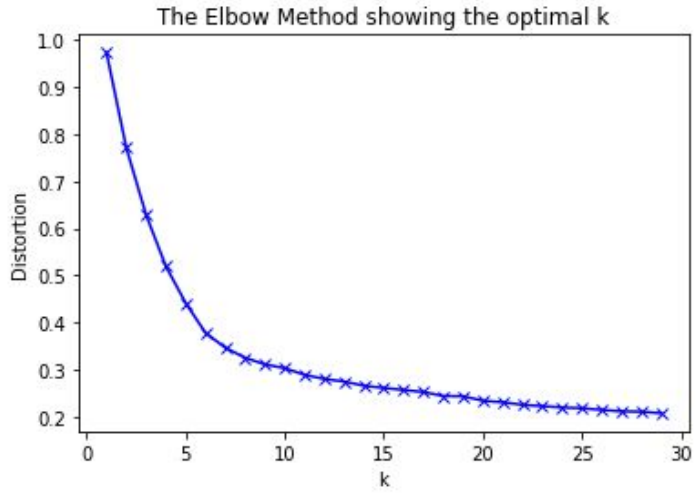
Pour réaliser l'algorithme nous disposons de 10634 articles. Ces articles sont répartis suivant les 5 journaux suivants : Le Monde, Le Figaro, Télérama, Nouvel Observateur et Libération.

Principe de l'algorithme:

Pour l'algorithme des K-means, on choisit K articles aléatoirement qui sont les premiers centroïdes.
On assigne chacun des articles à son centre le plus proche (distance minimale).
On trouve les nouveaux centres de clusters en re-calculant les barycentres des groupes.
On continue à assigner les articles aux centroïdes correspondants puis à calculer les barycentres des groupes jusqu'à ce que l'algorithme converge c'est-à-dire que les centres soient fixes.

Détermination du K optimal:

Nous avons testé l'algorithme avec à chaque fois un nombre K de clusters différents dans le but de trouver le nombre de clusters optimal, tel que les groupes soient différentiables au maximum.
Après plusieurs tentatives, nous avons remarqué que le cas optimal était atteint pour K=5.
Afin d'être sûrs de notre choix, nous avons utilisé la méthode du coude (elbow method) et nous avons ainsi confirmé le nombre de groupes K optimal, qui est de 5.



Ce graphe montre l'évolution de la variance intra-classe en fonction du nombre K de clusters choisis. On constate qu'avec cette valeur de K=5, on a un équilibre entre une variance intra-classe petite et une variance inter-classe grande.

Nous retrouvons des groupes de clusters assez équilibrés en terme d'articles, leur taille varie entre 1726 articles et 3038 articles par cluster.

Index	rate_angry	rate_surprise	rate_positivity	rate_joy	rate_negativity	rate_fear	rate_sadness
0	0.0846988	0.0685129	0.720169	0.0466447	0.27983	0.111137	0.103079
1	0.0766414	0.0521036	0.72516	0.0398803	0.271419	0.102968	0.0945134
2	0.0938314	0.0640407	0.704711	0.036845	0.295289	0.122807	0.100539
3	0.128016	0.0627689	0.69556	0.0289626	0.304439	0.147852	0.111527
4	0.109596	0.0655646	0.711739	0.0334911	0.287166	0.124709	0.101306

rate_disgust	rate_subjectivity	international	france	economie	sciences_high_tech	arts_et_culture	sports	sante
0.0548077	0.273063	-4.996e-15	-8.85403e-15	-5.46785e-15	0.0317508	1	0.0142123	0.00483822
0.0415462	0.221387	0.0337079	0.0806058	1	0.0341964	0.0669272	0.00439668	0.00341964
0.0518611	0.241052	6.07847e-15	1.02696e-15	0.000645995	0.463824	5.32907e-15	0.50323	0.0484496
0.0649219	0.213205	1	0.0548692	0.000363372	0.0403343	0.117006	0.00799419	0.000726744
0.0616078	0.222091	0.0130208	1	0.00625	0.0510417	0.256771	0.0078125	0.00572917

Nous avons créé un dataframe à partir des groupes créés par l'algorithme K-means. On peut y voir pour chaque cluster la moyenne pour chaque caractéristique. C'est à partir de ce tableau que nous avons pu interpréter les clusters formés.

<u>Interprétation:</u>

Cluster 1 - Articles d'arts et culture subjectifs - Des articles de catégorie Arts et culture (moyenne de 1) où l'on retrouve un taux de subjectivité beaucoup plus élevé par rapport aux autres clusters et donc propre à ce groupe. Ceci est logique car l'art et la culture sont subjectifs selon les avis.
Cluster 2 - Articles économique positifs - Des articles de catégorie Économie (moyenne de 1) où l'on observe un taux de positivité supérieur par rapport aux autres clusters (0,73).
Cluster 3 - Articles de sports, de sciences/high-tech associés à la peur - Des articles de catégorie Sports, Sciences/High-Tech où l'on observe un taux de peur élevé par rapport aux autre clusters.

Cluster 4 - Articles Internationaux associés à la colère - Des articles de catégorie Internationale (moyenne de 1) où l'on retrouve un taux de colère élevé par rapport aux autres clusters (0,13).
Cluster 5 - Articles France orientés positifs - Des articles de catégorie France (moyenne de 1) où l'on observe un fort taux de positivité (0,71) et de surprise (0,07).

3.5.2. Classification Ascendante Hiérarchique (CAH)

→ ***PRINCIPE***

La CAH est une méthode statistique visant à partitionner par exemple nos articles en différentes catégories. Elle est dite "Ascendante" parce qu'elle part des observations individuelles à un unique groupe. Elle est dite "Hiérarchique" car elle produit des classes de plus en plus vastes en terme d'individus. On cherche à ce que les articles regroupés au sein d'une même catégorie soient le plus semblables possibles tandis que les catégories soient le plus dissemblables possibles.

Notre CAH rassemblera les 11574 articles selon un critère de ressemblance basé sur la matrice des distances de saut de ward; deux à deux entre eux. Plus deux articles sont identiques, plus leur distance est proche de 0. Cette tâche sera exécutée de manière itérative afin de produire un arbre de classification. En découpant cet arbre à une certaine hauteur choisie, on produira la partition voulue.

→ **APPLICATION**

On applique la CAH sur

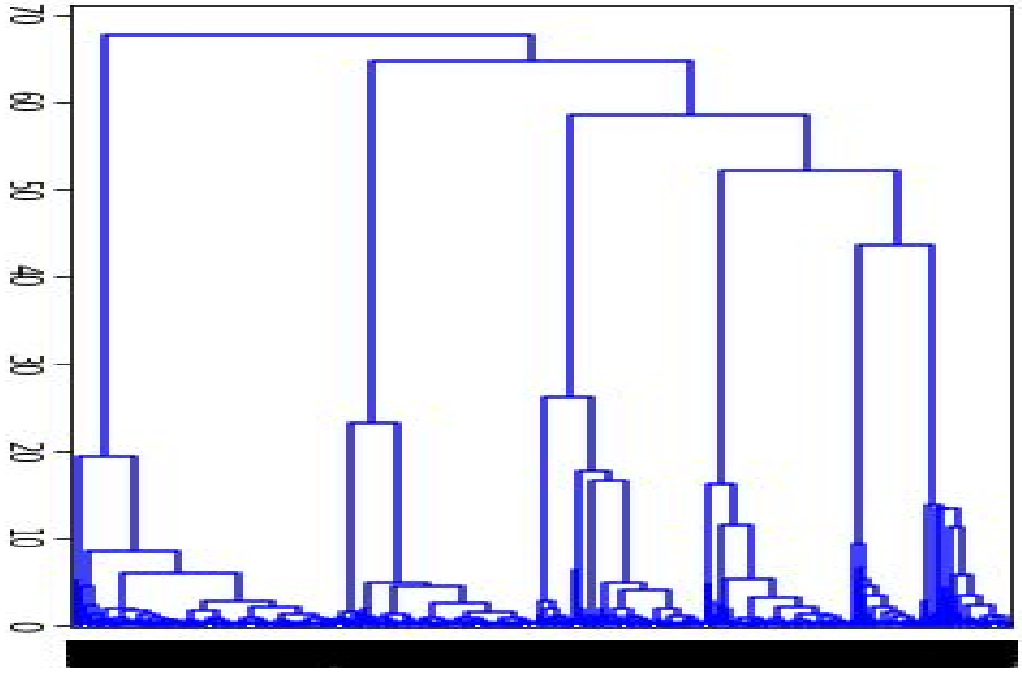
. 11.574 articles décrits par 16 variables.

. 16 variables Y1 : rate_angry
Y2 : rate_surprise
Y3 : rate_positivity
Y4 : rate_joy
Y5 : rate_negativity
Y6 : rate_fear
Y7 : rate_sadness
Y8 : rate_disgust
Y9 : rate_subjectivity
Y10 : international
Y11 : france
Y12 : economie
Y13 : sciences_high_tech

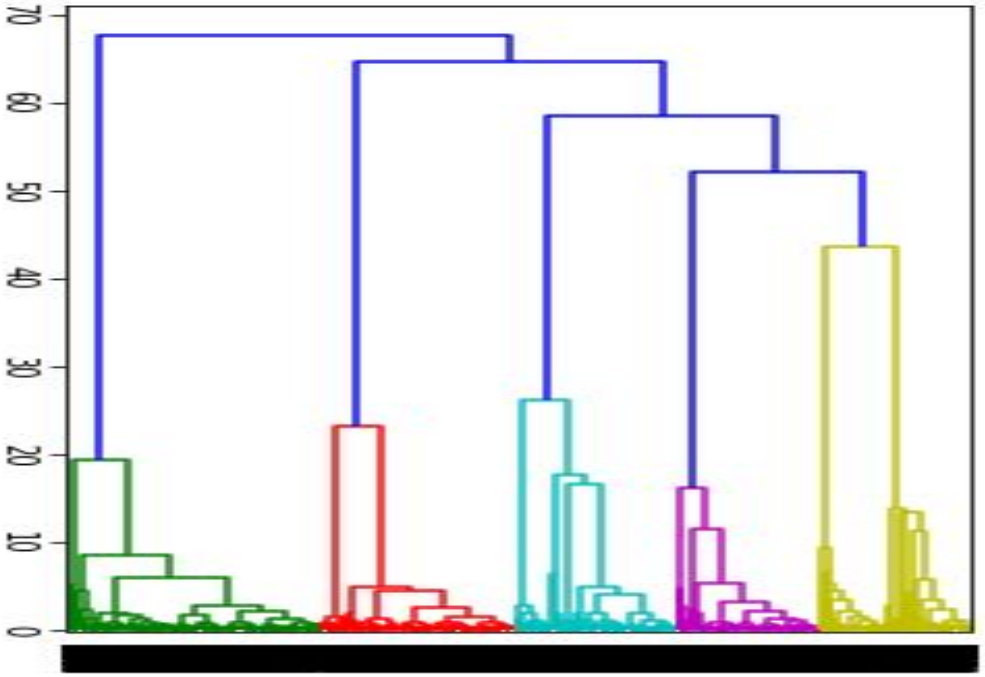
Y14 : arts_et_culture
Y15 : sports
Y16 : sante

Les neuf (Y1 à Y9) première variables sont explicitées pas leur taux dans le corpus et les sept dernières (Y10 à T16) par les TF-IDF. Les deux associés, nous donne une matrice numérique de dimension 11574X16.

Dans un premier temps on regarde la structure de l'arbre de classification (Dendrogramme) afin d'en tirer le nombre de classes à considérer en fonction du regroupement des articles .On obtient le dendrogramme suivant :



En coupant le dendrogramme obtenu à la hauteur k=50, on décide alors de garder cinq classes comme l'illustre la figure ci-dessous:



4. Difficultés rencontrées et solutions apportées

Durant le projet nous avons pu rencontrer quelques problèmes.
En effet, les articles nous ont été donnés le deuxième jour du projet. Pour pouvoir nous entrainer nous avons alors utilisé, le premier jour, les articles qui ont servi à Geoffrey durant son stage de L3.
On devait recevoir un filtrage basique des textes que nous n'avons pas eu avant la fin de la première semaine (juste un exemple de 3 json le jeudi). Afin de pouvoir avancer nous nous sommes servies du filtrage effectué par Geoffrey lors de son stage.

5. Conclusion

Nos objectifs ont été atteint : nous avons réussi à mettre en place des méthodes de classification supervisée en multi-label ainsi que des méthodes non supervisées avec des données sémantiques malgré les imprévus.

Pour les L3, ce projet a été très enrichissant car cela leur a permis une première approche au machine learning autant dans la méthode que dans le fonctionnement.
Les M1 et M2 ont pu mettre en pratique les connaissances acquises lors des cours de machine learning, et on ainsi pu bien aider les L3 dans l'apprentissage des méthodes.