Projet d'Intégration

Etude de la représentativité des participants au grand débat national par rapport à la population française

Réalisé par

Adrien HANS

Ecole Centrale de Lille – INRIA Lille

Parcours « Sciences des données et Intelligence Artificielle »

Filière métier « Responsable de l'Innovation et du Développement »

Année scolaire 2019/2020

Date de remise : lundi 9 mars

Remerciements

Je tiens à remercier Pascal Denis, Mikaela Keller et Aurélien Bellet pour leur accompagnement et leurs précieux conseils.

Je tiens également à remercier Pierre Chainais et Sire de Marc Ebode pour leurs encouragements à réaliser ce projet d'intégration.

Table des matières

I. Introduction

Le 15 janvier 2019, dans le contexte du mouvement des Gilets Jaunes en France et pour répondre à cette crise sociale grave pendant son quinquennat, le Président de la République Française Emmanuel Macron a décidé de mettre en place un « Grand Débat National ». Cet exercice, permettant aux français de s'exprimer librement sur des sujets sociétaux en dehors des élections, vise particulièrement à répondre à la crise « démocratique », et, selon le pouvoir exécutif : « Cette concertation d'ampleur nationale a pour objectif de redonner la parole aux Français sur l'élaboration des politiques publiques qui les concernent »[1].

Cette proposition peut en effet sembler être une réponse intéressante à la crise démocratique qui secoue notre pays : Abstention massive à toutes les élections, crises sociales répétées, sentiment d'éloignement des élus…. Ou encore une réponse à la demande de beaucoup de gilets jaunes de l'instauration un « RIC », « Référendum d'Initiative Citoyenne ».

Cependant, alors que cet exercice démocratique pourrait être de plus en plus développé, notamment à mesure que les techniques d'intelligence artificielle et de NLP (Natural Language Processing – Traitement automatique du langage naturel) évoluent et que les perspectives d'utilisation deviennent de plus en plus claires, il serait intéressant de déterminer la pertinence effective d'un tel exercice. En effet, l'utilisation politique et partisane est une des grandes limites de cet exercice, car nous ne sommes à priori pas en mesure de nous en prémunir, à la fois dans l'analyse qui est faite des contributions et dans la représentativité ou non des contributeurs par rapport à l'ensemble de la population française.

Ainsi, si dans une élection chaque citoyen est en capacité de voter et de ne voter qu'une seule fois, aucune garantie n'est émise s'agissant de ce grand débat national. En effet, chacun peut y contribuer plusieurs fois et aucune identification n'est demandée pour y participer. On peut alors se poser la question, si cet exercice se répète, et surtout si l'analyse qui est faite des contributions sert de base à l'établissement ou la poursuite d'une politique, de la représentativité des participants par rapport à la population française ou bien de la tentative de certains partis français ou encore de puissances étrangères d'influencer les résultats de ce débat.

J'ai alors décidé de me poser les questions suivantes :

- Les participants de ce grand débat sont-ils représentatifs politiquement de la population française ?
- Comment essayer d'étudier cette représentativité ?

Cependant, aucune donnée personnelle en dehors du code postal du participant n'est associée à chaque contribution. Je me suis donc inspiré d'autres travaux mis en place dans un contexte d'avancée des techniques de traitement automatique du Langage naturel, d'abord avec l'« author profiling »[3] [4], c'est-à-dire la découverte d'attributs personnels, tels que le genre, l'âge… à partir de textes écrits, mais aussi dans le domaine de la politique, avec le political scaling [2] [5]. En revanche, les données du grand débat national n'étant pas annotées et ne les ayant pas annotées moi-même étant donné la complexité en termes d'hypothèses notamment au niveau politique et le fait que le temps nécessaire pour réaliser cela aurait été beaucoup trop important par rapport à celui disponible pour ce projet.

[1] https://www.gouvernement.fr/grand-debat-national-4-themes-retenus-pour-animer-la-concertation

Ainsi, l'étude de cette représentativité s'est faite à partir de plusieurs points, notamment la recherche et l'utilisation d'autres de bases de données ayant des informations de type politique associées à des textes, ensuite l'analyse des données de ces bases, puis la mise en place de techniques de NLP et de Machine Learning en entrainant des modèles sur les bases de données trouvées et en les appliquant sur les données du grand débat dans le but de découvrir des attributs personnels intéressants pour chaque contributeur afin d'essayer de résoudre mes problématiques.

D'abord, je présenterai les différentes bases de données utilisées et des analyses descriptives réalisées sur celles-ci. Ensuite, nous verrons quelles sont les techniques de Machine Learning disponibles pour un tel exercice puis les résultats qui découlent de leur application dans le cadre de mon projet d'intégration. Enfin, je présenterai quels travaux peuvent être envisagés pour compléter ce projet.

II. Présentation et analyses des bases de données utilisées

Dans, cette partie, je précise tout d'abord quelles sont les bases de données utilisées et comment celles-ci ont été collectées et nettoyées. Ensuite, nous verrons dans un premier temps les analyses descriptives des données ouvertes du grand débat national, puis, dans un second temps, des analyses similaires sur les autres données en fonction des informations disponibles.

a. Présentation et nettoyage

i. Présentation des bases de données et des processus de collecte

Comme je l'ai expliqué, étant donnée le peu d'informations fournies dans les données ouvertes du grand débat, il a fallu chercher d'autres bases de données pouvant servir dans l'essai de la résolution de ma problématique. Trois bases de données principales ont été utilisées.

Tout d'abord, une base de données lié à un article [1], que nous appellerons la base #Elysee2017, contenant une liste de comptes twitter, dont le parti politique et le sexe ont été annoté à la main. Cette base de données semblait donc utile pour appliquer des modèles de type NLP afin d'essayer de classifier le parti politique de chaque contribution au grand débat national.

Ensuite, et dans la même idée, puisque nous connaissons évidemment les partis politiques de nos élus, j'ai cherché une base de données indiquant les comptes twitter de chaque parlementaire (députés et sénateurs)[2].

Enfin, j'ai utilisé une base de données issu de « Entendre la France », un débat du même type que le grand débat national et organisé en marge de celui-ci, mais dont les contributions étaient annotées, notamment en fonction du genre, de l'âge, ou encore de la position du contributeur par rapport au mouvement des gilets jaunes.

Pour les bases de données Twitter, un « scraper » permettant de récupérer tous les tweets et les données associées (nombre de retweets, nombre de « likes », date de la publication…) a été développé avec twint[3], c'est-à-dire sans passer par l'API officielle de Twitter, afin de ne pas avoir de limites dans le nombre de tweets à intercepter, ni en termes de vitesse, ni en termes de volume. Ce scraper a été, étant donné le très grand nombre de profils à capter, développé en multithread pour augmenter les possibilités d'action et réduire le temps de calcul. Les bases de données étant importantes, que ce soit en termes d'utilisateurs ou de tweets à récupérer, le temps requis pour « scraper » toutes ces données aurait été bien trop important si je n'avais pas utilisé cette fonctionnalité.

Pour la base de données twitter liée aux parlementaires, tous les tweets ont été récupérés. Le nombre de comptes twitter étant relativement limité, cela était possible et pouvait démontrer un intérêt pratique pour la classification ultérieure, puisque plus le jeu de données est important, meilleure pouvait être la caractérisation de chaque parti, ou en tout cas je ne faisais pas de choix possiblement difficile à justifier sur la pertinence ou non de chaque écrit.

Pour la base de données #Elysée2017, le nombre de tweets a été limité aux 100 les plus populaires pour chaque compte. Ceci est justifié par des tests empiriques de ma part, c'est à dire que regarder quelques tweets parmi les plus populaires était bien plus instructif pour connaître le parti de la personne considérée que des tweets pris totalement au hasard. De plus, et surtout, ceci était justifié par l'annotation même de la base, car les chercheurs à l'origine de cette base de données annotée à

[2] https://github.com/regardscitoyens/twitter-parlementaires
[3] https://github.com/twintproject/twint

la main ont établi le parti politique de chaque utilisateur grâce à leurs biographies et aux 10 tweets les plus populaires de chacun.

La base de données 'Entendre la France' a quant à elle été téléchargée simplement sur le site internet de l'organisation[4], tout comme la base des contributions du grand débat national.[5]

ii. Nettoyage

1. Base de données des contributions du grand débat national

Le but de ce rapport n'est pas de détailler chaque aspect de mon travail, mais avant de pouvoir analyser ou bien traiter les données ouvertes du grand débat national, il a fallu réaliser un grand travail de nettoyage, certaines données étant totalement fausses, absentes, ou bien parce que les bases de données n'étaient pas sous la bonne forme. Par exemple, et ceci en est un parmi beaucoup d'autres exemples de nettoyage réalisés au cours de ce projet, dans la base des contributions du thème « Organisation de l'état et des services publics », les colonnes intitulées « authorId », « authorType » et « authorZipCode » de la ligne 6096 contenaient des réponses aux questions du thème tandis qu'elles correspondent normalement respectivement à un code d'identification de l'auteur de la contribution, à un type d'auteur et au code postal du participant. Alors, cette ligne étant quasiment inutilisable, j'ai décidé de la supprimer pour plus de facilité et de logique, puisque je n'étais pas sûr de la question à laquelle ce participant voulait répondre dans chacune de ces propositions.

Il a fallu aussi, parfois, recouper les données ouvertes du grand débat avec d'autres bases données, ce qui a constitué là aussi de grands travaux de nettoyage, certaines données étant manquantes. Aussi, les données issues des participants résidents en Outre-Mer n'ont pas pu être considérées, elles ont donc été supprimées de la base.

2. Base de données twitter #Elysée2017

La base de données associée à #Elysée2017 est composée de 22853 profils twitter annotées à la main, dont 3401 pour lesquels le parti politique n'était pas étiqueté et qui n'ont donc pas été considérés. Pour chaque contributeur annoté politiquement, l'étiquetage se faisait soit en ne considérant qu'un seul parti parmi les cinq présentés ci-après, soit via une annotation double de deux partis pour certains profils. Pour plus de cohérence, j'ai décidé, les annotations doubles étant en plus très minoritaires, de n'utiliser que les annotations simples d'un unique parti.

Il est à noter que les données étaient annotées suivant si le compte twitter était celui d'un média ou non. Etant donné mes problématiques et les problèmes à résoudre, j'ai décidé de ne considérer que les « citoyens » et donc de ne pas considérer les médias.

La taille de la base de données était initialement de 22853 comptes twitter dont 19452 annotés politiquement. Cependant, comme je viens de le dire, les médias n'ont pas été considérés, et certains comptes n'ont pas pu être retrouvés, en général à cause de deux principales raisons :

- Le compte twitter n'existe plus. Ceci est une possibilité forte puisque la base de données a été annotée il y a environ 3 ans, lors de l'élection présidentielle de 2017.
- Le compte twitter a été mis en « privé », ce qui veut dire que le compte existe mais qu'il n'est pas accessible pour quelqu'un dont le titulaire du compte n'a pas expressément accepté qu'il pouvait y avoir accès.

[4] https://www.entendrelafrance.fr/
[5] https://granddebat.fr/pages/donnees-ouvertes

Ainsi, à la fin, 14211 comptes associés à la base #Elysée2017 ont pu être récupérés via la méthode de « scraping » explicitée précédemment.

3. Base de données des parlementaires

Dans la base de données des parlementaires, le fichier contenant les comptes twitter de chaque parlementaire était associé au nom de celui-ci mais pas à son appartenance politique.[6] Il a donc fallu utiliser une base de données disponible sur les sites de l'assemblée nationale[7] et du sénat[8] afin d'apporter l'information de la couleur politique associée à chaque compte. A la fin, 315 comptes twitter associés à des sénateurs et 532 associés à des députés ont pu être récupérés, par la même méthode que celle utilisée pour la base de données #Elysee2017.

b. Analyses descriptives des données ouvertes du grand débat national

Dans cette partie, je présente les premières étapes d'analyses des données ouvertes du grand débat national. Pour ce travail d'analyse, j'ai développé plusieurs fonctions afin d'afficher des statistiques sur différents aspect de ces données, comme le nombre de participations à chaque thème, le nombre de questions ouvertes (avec des réponses sous forme d'un texte libre) ou bien fermées (réponses sous la forme d'un QCM), la taille des titres…

En outre, des informations ont été inférées grâce aux codes postaux, comme des cartes des contributions ou bien des affichages liés au résultat du premier tour des élections présidentielles de 2017. Pour établir ces résultats, il a fallu lier les codes postaux à une autre base de données, afin d'avoir les coordonnées géographiques pour établir les cartes des contributions, et aux codes INSEE, puis aux codes géographiques et enfin aux résultats fournis par le ministère de l'intérieur pour les élections.

On présente ci-après quelques résultats de ces analyses pour tous les thèmes, toutes les analyses étant disponibles sur le github du projet, dont le lien est disponible en annexe.

i. Nombre de contributions pour chaque thème

	theme	nb_contributions	nb_questions
0	La fiscalité et les dépenses publiques	113521	8
1	Organisation de l'état et des services publics	67301	33
2	Démocratie et citoyenneté	65758	37
3	La transition écologique	35982	16

Figure 1- Nombre de contributions et de questions pour chaque thème

Les propositions faites pendant ce grand débat pouvaient se faire selon quatre thèmes :

- « La fiscalité et les dépenses publiques »
- « Organisation de l'état et des services publics »
- « Démocratie et citoyenneté »
- « La transition écologique »

[6] https://github.com/regardscitoyens/twitter-parlementaires
[7] http://data.assemblee-nationale.fr/acteurs/deputes-en-exercice
[8] https://data.senat.fr/les-senateurs/

Nous pouvons observer, dans le tableau présenté ci-dessus, que le thème ayant eu le plus de contributions est le thème « La fiscalité et les dépenses publiques », tandis que le thème ayant eu le moins de participants est le thème « La transition écologique ».

Nous pouvons aussi afficher le nombre maximal de contributions pour chaque thème pour un auteur unique :

	theme	maximum contributions
0	La fiscalité et les dépenses publiques	104
1	Organisation de l'état et des services publics	54
2	Démocratie et citoyenneté	57
3	La transition écologique	38

Figure 2- Nombre maximum de contributions par auteur pour chaque thème

Nous observons que certains participants ont beaucoup contribué sur certains thèmes. Par exemple, un contributeur a émis 104 contributions sur le thème « La fiscalité et les dépenses publiques » et un autre 57 fois sur le thème « Démocratie et citoyenneté ».

Cela a été un problème lorsqu'il a fallu appliquer des modèles de Machine Learning, car je ne savais que faire de toutes ces contributions pour un seul auteur : faut-il n'en considérer qu'une seule ? Et si oui comme la sélectionner ? En prenant la première ? La dernière ? Aléatoirement ? J'ai donc décidé de considérer toutes les contributions de chaque auteur en les concaténant, pour ne pas perdre de l'information.

Le fait que certains participants aient contribué beaucoup de fois pour certains thèmes a motivé le fait d'étudier la représentativité des contributions et des contributeurs, car on peut se demander si la personne ayant contribué 104 fois pour le thème « La fiscalité et les dépenses publiques » n'essaient pas de « bourrer les urnes » du grand débat national avec ses multiples contributions.

ii. Proportions de personne ayant contribués à plusieurs thèmes

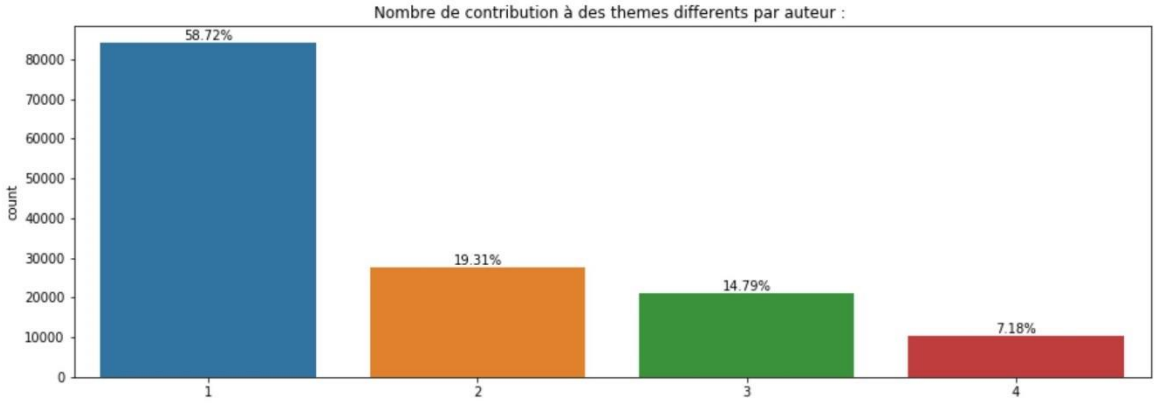


Figure 3 - Part de contributions aux différents thèmes par participant

Nous pouvons observer que la plupart des contributeurs n'ont contribué qu'à un seul et unique thème (58.72%) et que très peu ont contribués à tous les thèmes (7.18%). Cela montre que certains contributeurs ont montré moins d'intérêt pour certains thèmes, de la même façon que nous venons de voir que le thème « la fiscalité et les dépenses publiques » avait quasiment le double de

réponses des thèmes « Organisation de l'Etat et des services publics » et « Démocratie et citoyenneté » et quatre fois plus que le thème « La transition écologique ».

iii. Types des auteurs

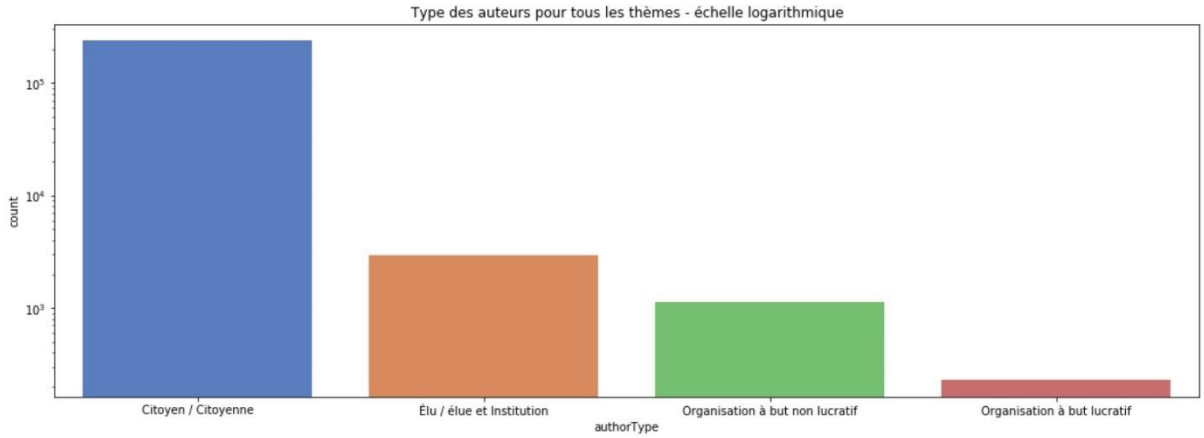


Figure 4 - Types des participants

L'échelle de ce graphe étant logarithmique, les contributions au grand débat ont en grande partie été émises par des citoyens et des citoyennes. Cela encourage le fait de vouloir essayer de déterminer la représentativité des contributeurs, certains types d'auteurs autre que les citoyens pouvant être connotés politiquement. Cependant, étant donné la très faible part d'« Elu / Elue et institution », d'« Organisation à but non lucratif » ou bien d'« Organisation à but lucratif » par rapport à « Citoyen /Citoyenne », j'ai décidé de tout considérer, puisque mes problématiques se posent sur l'ensemble des données du grand débat national.

iv. Tailles moyennes des réponses aux questions dites ouvertes

Dans le but de comparer les données de type « twitter » et les données issues des contributions, nous pouvons afficher les tailles des réponses aux questions ouvertes, c'est-à-dire en ne considérant pas les questions fermées, pour lesquelles les réponses sont souvent « oui » ou « non », influençant grandement la moyenne de ces tailles de réponses.

Je présente ci-dessous l'histogramme des tailles des réponses aux questions ouvertes pour tous les thèmes, en limitant l'affichage à 1000 caractères pour plus de clarté :

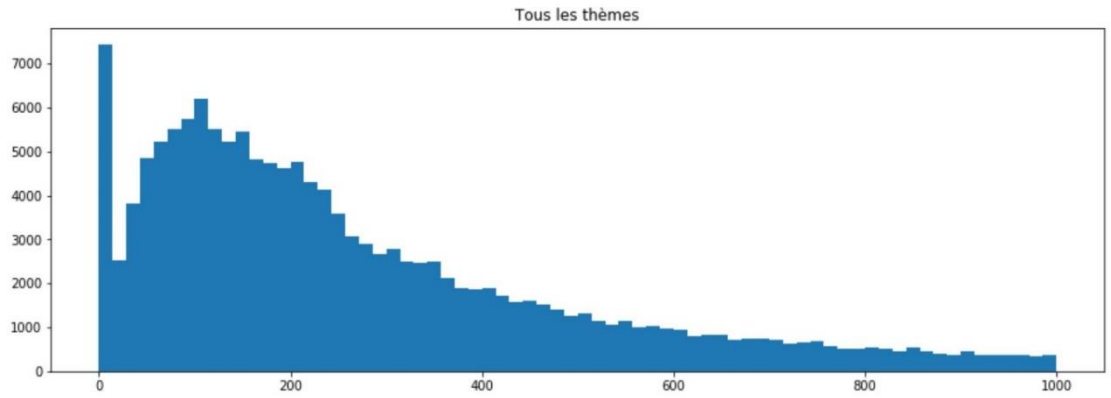


Figure 5 - Tailles des contributions sur les questions ouvertes pour tous les thèmes du grand débat national

Comme nous pouvons l'observer, la taille des réponses aux contributions est assez grande devant le nombre de caractères présents dans un tweet. La taille moyenne est en effet de quasiment 500 caractères, avec environ 70% des tailles des réponses supérieures à la taille d'un tweet de 140 caractères, qui est la limite du nombre de caractères sur twitter avant novembre 2017[9], et environ 45% supérieures à la taille d'un tweet de 280 caractères, qui est la limite après novembre 2017.

Nous voyons donc bien que les données twitter et les contributions du grand débat national ne sont pas du même type, et qu'il faudra étudier le passage d'un type de données à un autre par la suite.

v. Résultats aux élections :

Avec l'idée d'essayer de commencer à répondre à mes problématiques de représentativité des contributeurs du grand débat national, j'ai croisé les données des codes postaux avec ceux des résultats aux élections présidentielles de 2017 fournies par le ministère de l'intérieur (en passant par les codes INSEE et les codes géographiques). Pour avoir des résultats les plus cohérents possibles, j'ai fait l'hypothèse que chaque contributeur était représentatif politiquement de sa ville, en lui associant les proportions de votes de sa commune pour chaque candidat. C'est-à-dire que si dans une ville les résultats au premier tour des élections présidentielles étaient de 30% pour Emmanuel Macron, 20% pour Marine Le Pen, 10% pour François Fillon… alors un habitant de cette ville ayant participé au grand débat national serait considéré dans l'étude suivante comme étant votant à hauteur de 30% pour Emmanuel Macron, 20% pour Marine Le Pen, 10% pour François Fillon…

a) Résultats du premier tour des élections présidentielles de 2017

Je rappelle ci-après les résultats du premier tour des élections présidentielles de 2017 afin de les comparer aux analyses faites sur les contributions du grand débat via les codes postaux :

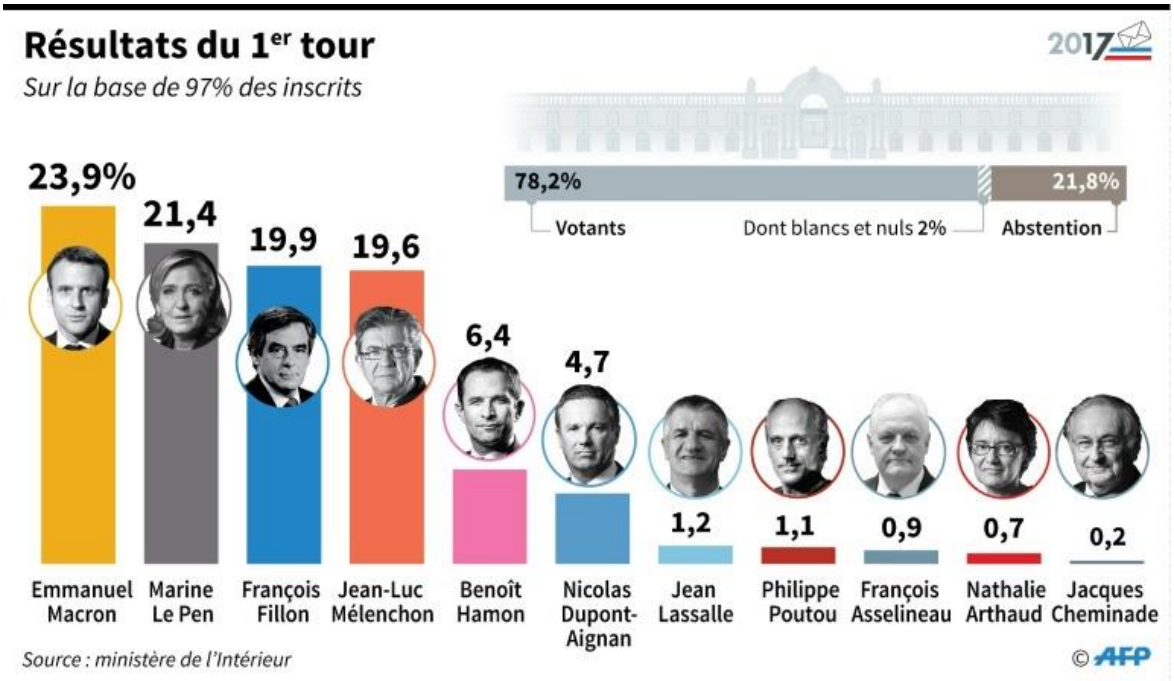


Figure 6 - Résultats du premier tour de l'élection présidentielle de 2017[10]

[9] https://fr.wikipedia.org/wiki/Twitter
[10] https://www.challenges.fr/politique/presidentielle-les-resultats-globaux-hors-francais-de-l-etranger_468866

b) En considérant toutes les contributions

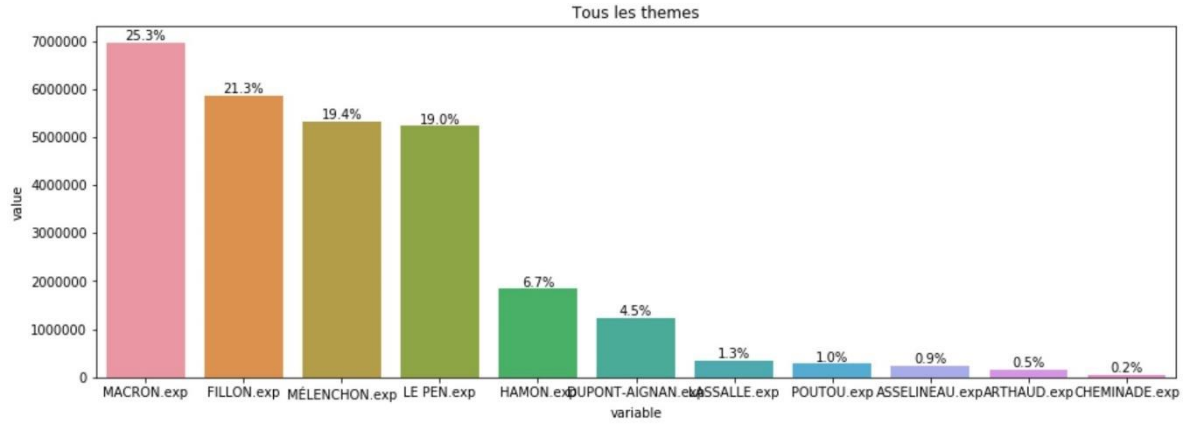


Figure 7 - Résultats aux élections en considérant toutes les contributions

On observe déjà en considérant toutes les contributions, c'est-à-dire que chaque contributeur est considéré autant de fois qu'il a contribué au grand débat via ses propositions, que les résultats sont différents des résultats du premier tour de l'élection présidentielle.

c) En considérant une seule contribution par auteur :

On peut maintenant réaliser le même type d'affichage en considérant une seule fois chaque participant du grand débat national, même s'il a contribué à plusieurs thèmes par exemple.

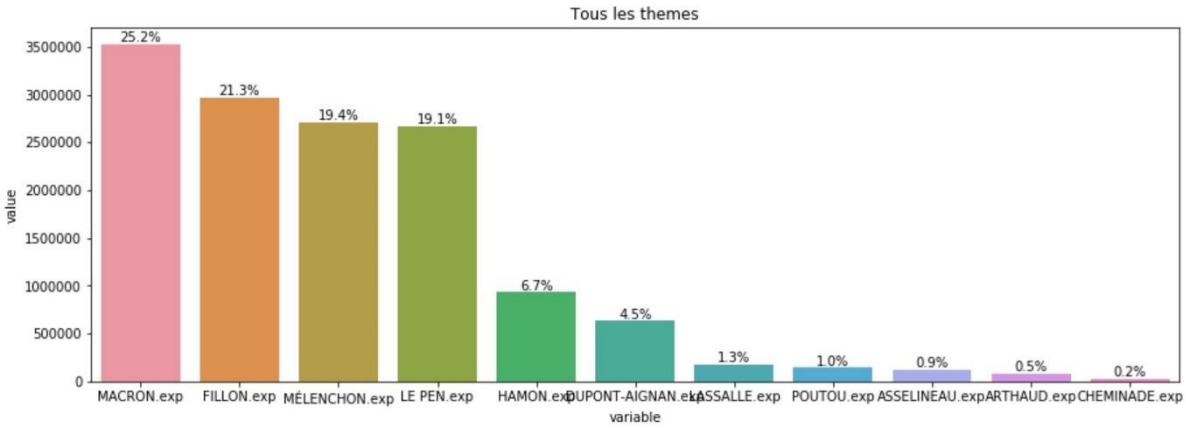


Figure 8 - Résultats aux élections en considérant une seule contribution par auteur

On retrouve des résultats relativement similaires ici par rapport à si l'on considère toutes les contributions en ne se préoccupant pas du fait que le même participant a déjà été pris en compte, ce qui s'explique par le fait que la grande majorité des contributeurs n'a contribué qu'une seule fois.

On observe alors de façon générale qu'avec les hypothèses faites qui sont certes très fortes, les résultats du premier tour des élections présidentielles auraient été assez différents :

- Le score d'Emmanuel Macron, candidat de La République en Marche, arrivé en tête au premier tour, arrive là aussi en tête mais avec environ 1,5 points de pourcentage en plus par rapport aux « vrais » résultats dans les deux cas.

- Le score de François Fillon, candidat des Républicains, arrivé troisième au premier tour des élections présidentielles de 2017, arrive ici deuxième et son score augmente lui aussi d'environ 1,5 points de pourcentage.
- Jean-Luc Mélenchon, candidat de La France Insoumise, a un score relativement stable mais arrive ici troisième, devant Marine Le Pen, candidate du Front National, pourtant qualifiée pour le second tour, et qui voit elle son score diminué de 2,5 points environ.

Ces résultats peuvent s'expliquer en partie par la sous-représentation de petites ou moyennes villes en termes d'habitants, par rapport aux grandes villes. Cependant, même s'ils sont obtenus en supposant des hypothèses très fortes, surtout si l'on considère que les élections ne sont elles même peut être pas totalement représentatives de la population française étant donné le taux d'abstention élevé à chaque élection, ces résultats ont tout de même encouragé ma volonté à développer cette idée d'étudier la représentativité des contributeurs, puisqu'ils ne sont pas totalement en accord avec les résultats « réels ».

vi. Heatmap des contributions

Avec la même démarche de croisement entre les codes postaux et d'autres données, j'ai associé chaque code postal à ses coordonnées GPS afin d'afficher une carte des contributions sous forme de heatmap, avec des clusters donnant le nombre de contributions par « zone ». Voici le résultat global sur toutes les contributions :

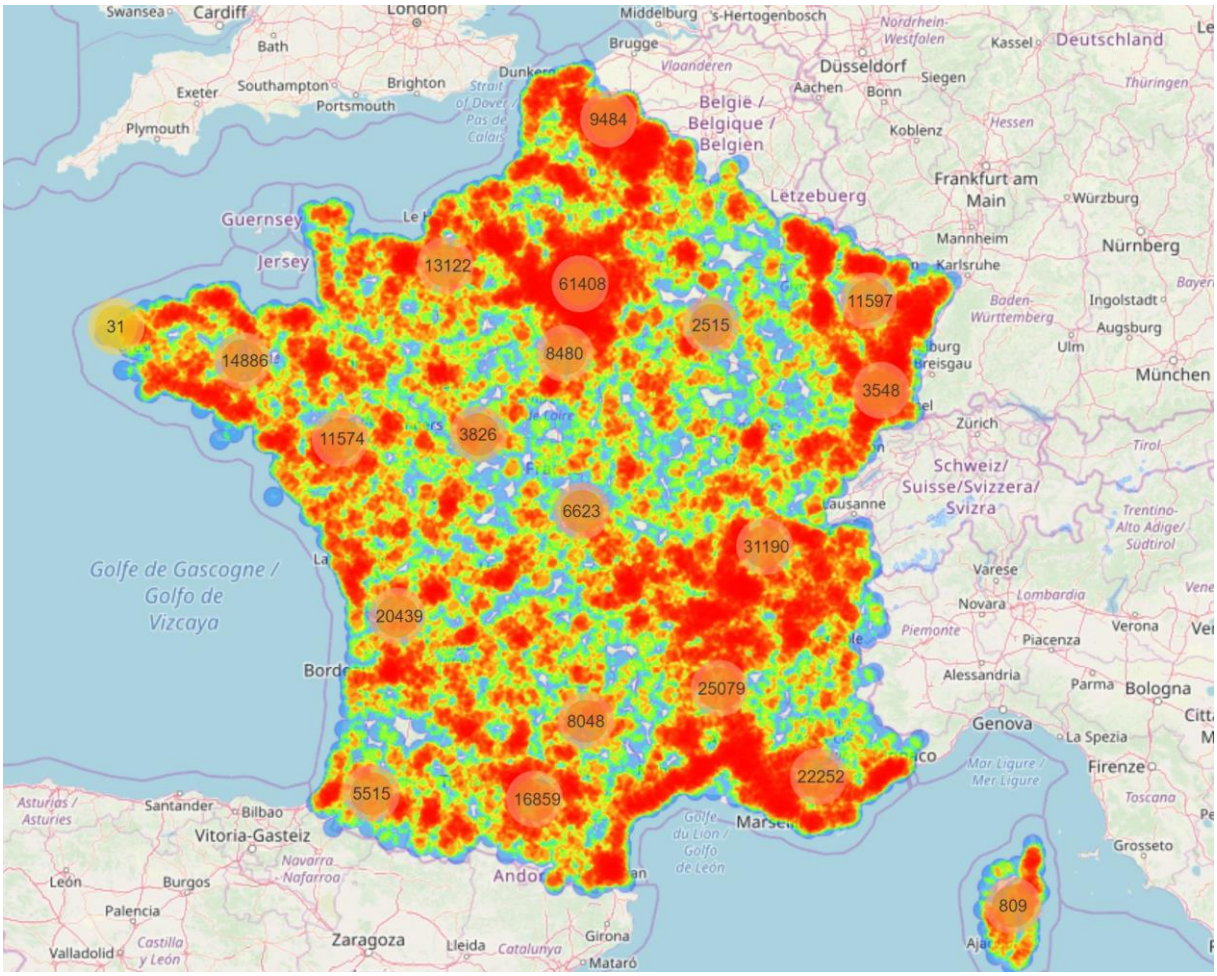


Figure 9 - Heatmap des contributions

On peut retrouver les zones les plus habitées, ce qui confirme que les petites et moyennes villes sont certainement sous-représentées.

c. Analyses descriptives des autres bases de données utilisées

Après avoir pu explorer certains aspects des contributions du grand débat national, nous allons, dans cette partie, présenter, à l'instar des analyses de la partie précédente, des analyses relatives aux autres bases de données utilisées dans mon travail, c'est-à-dire la base de données twitter liée à #Elysée2017, celle liée aux parlementaires, et enfin la base de données « Entendre la France ».

i. Base de données twitter #Elysee2017

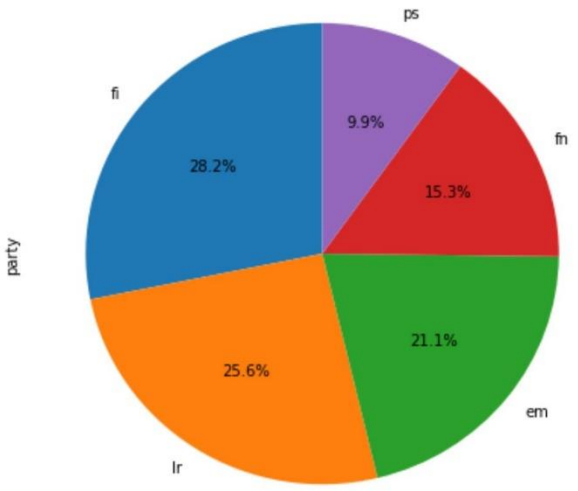


Figure 10- Proportion des cinq partis politiques dans la base #Elysée2017

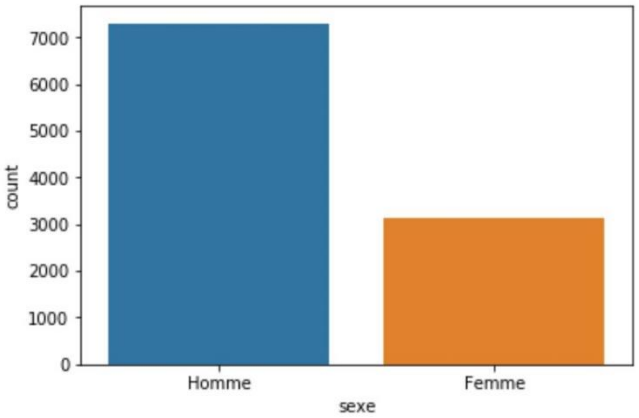


Figure 11 - Proportion de chaque sexe dans la base #Elysée2017

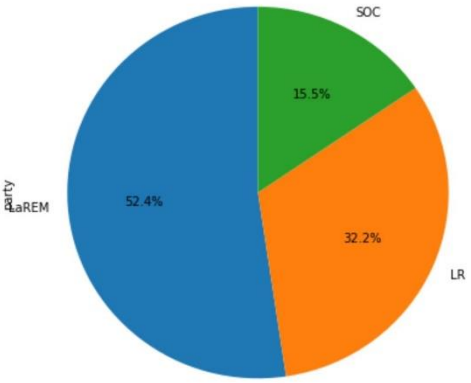
Nous pouvons voir que les profils considérés ont été annotés selon cinq partis politiques :

- La France Insoumise : « fi »[11]
- Le Parti Socialiste : « ps »[12]
- La République En Marche : « em »[13]
- Les Républicains : « lr »[14]
- Le Front National : « fn »[15]

Les proportions de chaque parti dans la base de données sont sensiblement du même ordre, tous les partis ont donc été considérés par la suite. Nous pouvons par ailleurs observer qu'il y a plus d'hommes que de femmes dans la base de données.

[11] https://lafranceinsoumise.fr/
[12] https://www.parti-socialiste.fr/
[13] https://en-marche.fr/
[14] https://republicains.fr/
[15] https://rassemblementnational.fr/

ii. Twitter des parlementaires



Tous les partis présents au parlement français étaient aussi présents dans cette base de données, cependant, certains partis étant très faiblement représentés, seuls trois partis ont été considérés : La République en Marche (« LaREM »), Les Républicains (« LR »), le Parti Socialiste (« PS »). J'aurais aussi pu essayer de grouper les partis faiblement représentés entre eux, mais il aurait fallu faires des hypothèses politiques sans doute assez fortes, ce que j'ai décidé de ne pas faire.

Figure 12 - Proportion des partis politiques dans la base de données liée aux parlementaires

Comme nous pouvons l'observer, dans les partis considérés, la base de données est majoritairement constituée de parlementaires de La République en Marche (« LaREM ») avec 52.4% contre 15.5 % pour ceux du Parti Socialiste (« SOC ») et 32.2% pour ceux des Républicains (« LR »).

iii. Entendre La France

La base de données Entendre La France a été constituée en même temps que le grand débat national, dans une démarche citoyenne, afin d'obtenir une base de données non contrôlée par l'Etat et les pouvoirs publics. De plus, cette démarche d'un « contre » grand débat a été assortie de la possibilité pour les contributeurs de fournir certaines informations personnelles, comme leur sexe, leur position par rapport aux gilets jaunes, leur tranche d'âge… Voici des graphes représentants certaines de ces informations, et en quelle proportion chaque classe est représentée :

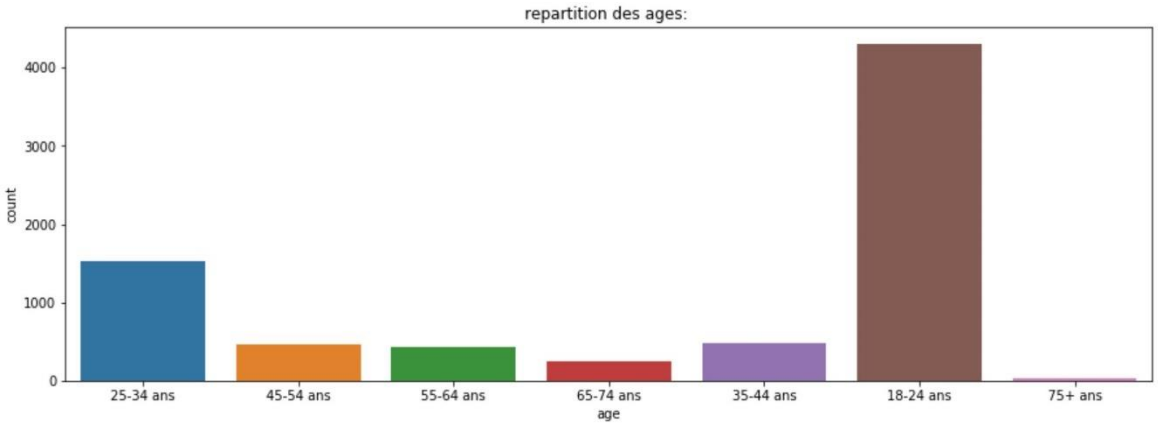


Figure 13 - Répartition des tranches d'âges dans la base de données 'Entendre la France'

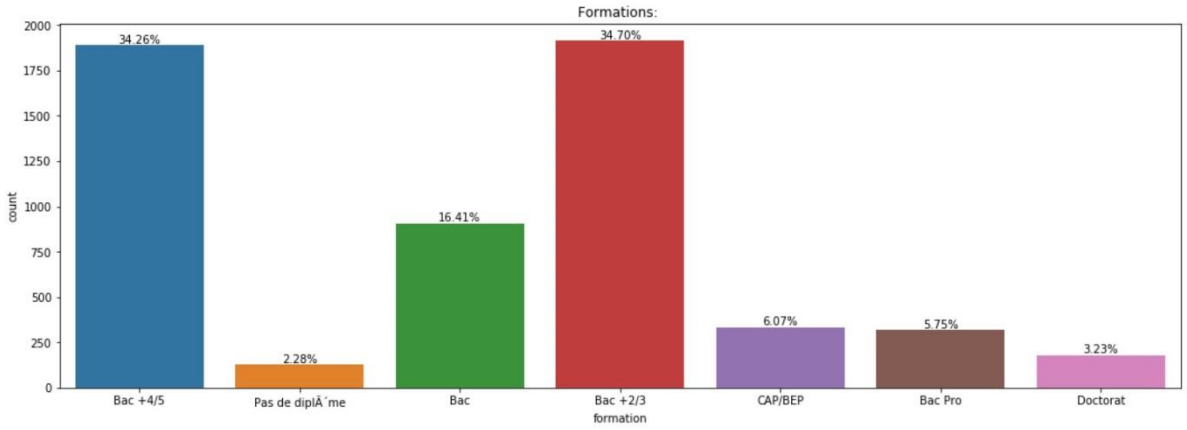


Figure 14- Répartition des niveaux de formation dans la base de données 'Entendre la France'

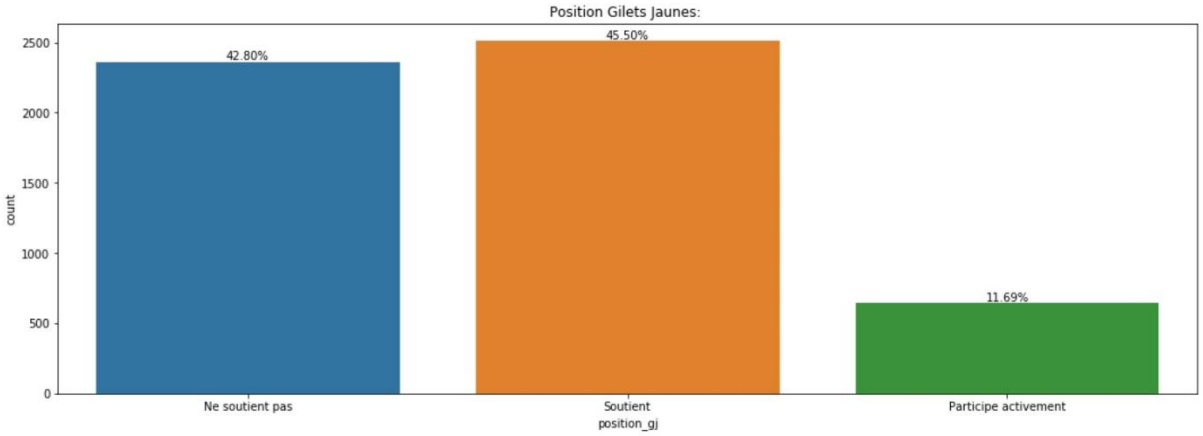


Figure 15 - Répartition des positions par rapport aux gilets jaunes dans la base de données 'Entendre la France'

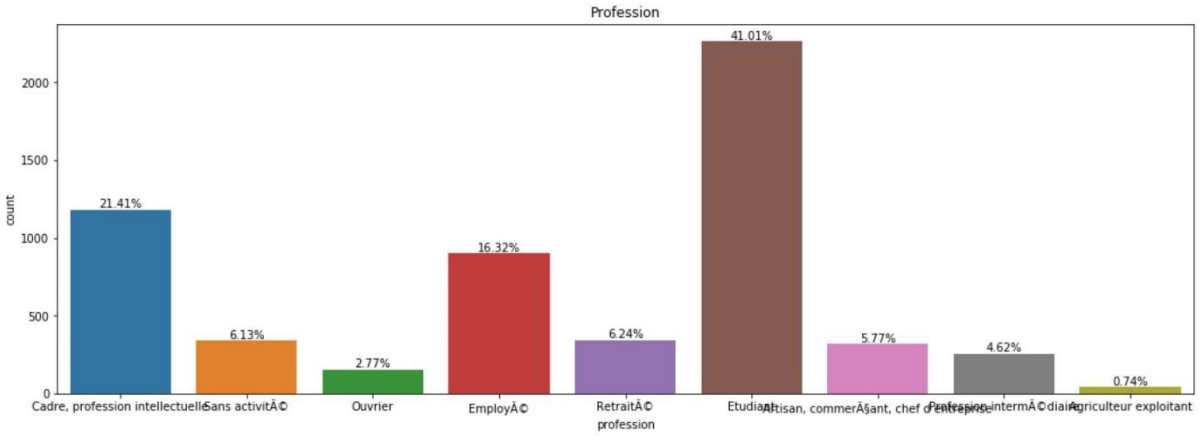


Figure 16 - Répartition des professions dans la base de données 'Entendre la France'

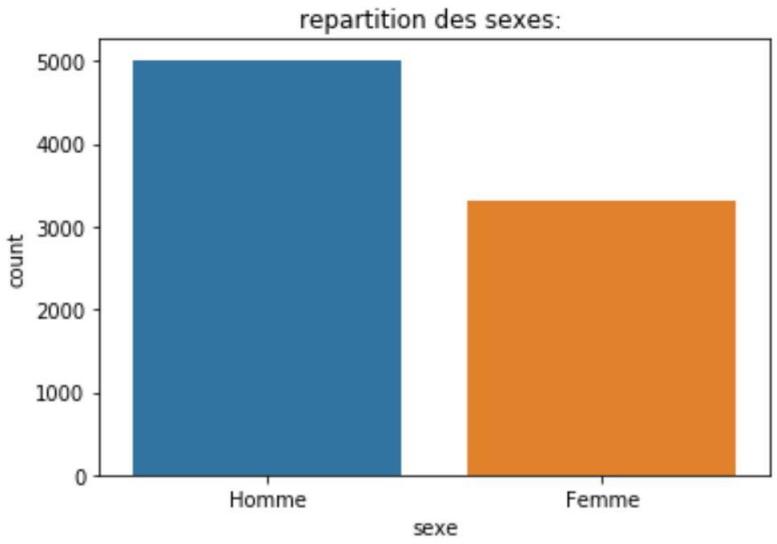


Figure 17 - Répartition de chaque sexe dans la base de données 'Entendre la France'

Nous pouvons observer que la base de données 'Entendre la France' est majoritairement constitué d'hommes, avec un fort taux de soutien envers les gilets jaunes, et que ce sont pour la plupart des jeunes. Cette base de données, en tout cas les personnes ayant participé à son établissement, représente bien une catégorie de la population connectée et en opposition (plus ou moins active) au pouvoir politique, ce qui est conforme à ce que nous pouvions attendre, étant donné que cette base de données a été constitué dans une certaine démarche d'opposition au grand débat national organisé par le gouvernement français.

III. Applications de modèles de Machine Learning grâce aux autres bases de données

Après avoir développé des analyses descriptives des bases de données utilisées, nous allons maintenant tacher de présenter des travaux de Machine Learning autour de ces données.

Etant donné les bases de données que j'ai trouvé, c'est-à-dire #Elysée2017, la base liée aux parlementaires français et la base du débat « Entendre la France », j'ai voulu essayer d'appliquer des modèles de NLP et de Machine Learning afin d'entrainer des algorithmes visant à classifier le parti politique, les opinions vis-à-vis du mouvement des gilets jaunes… des individus suivant le texte des contributions ou les tweets qu'ils avaient écrit avant d'appliquer ces modèles sur les contributions du Grand Débat National afin de retrouver ces opinions pour les participants et donc d'étudier la représentativité des contributeurs de cette consultation nationale.

Pour cela, il a fallu développer une méthode de travail, allant du nettoyage et du pré-traitement des textes écrits, d'une représentation mathématique de ces textes et enfin à l'apprentissage par des algorithmes de Machine Learning. Les Algorithmes ont ensuite été appliqués sur les contributions pour obtenir les résultats souhaités. Enfin, il a fallu étudier la bonne cohérence du fait d'apprendre et d'appliquer des algorithmes sur des données qui ne sont pas du même type et issues de la même base, c'est-à-dire passer de données textuelles de type « twitter » à des données de type « contributions », qui sont assez différentes, notamment en termes de tailles, comme nous l'avons vu précédemment.

a. Méthode utilisée

Dans cette partie, nous détaillons l'ensemble des étapes de travail dans une étude de traitement automatique du langage et d'application d'algorithmes et de méthodes de Machine Learning afin d'essayer d'obtenir des résultats convenables.

i. Pré-traitement

Avant d'appliquer des modèles, il faut nettoyer le texte à analyser. Ce nettoyage consiste en plusieurs points. D'abord, on gère les accents en remplaçant la lettre avec accent par la même lettre sans accent (« é » devient « e »)[16], ceci permet d'éviter de considérer comme étant différents des mots correctement orthographiés et aux accents manquants, et surtout des mots en début de phrase qui débutent avec une lettre accentuée, parfois absentes sur des majuscules. J'avais remarqué ceci en essayant de faire du topic modelling simple sans TF-IDF sur le thème « La transition écologique », avec les mots (après stemming) « ecolog » et « écolog » qui étaient considérés comme étant différents, comme illustré sur la figure ci-dessous.

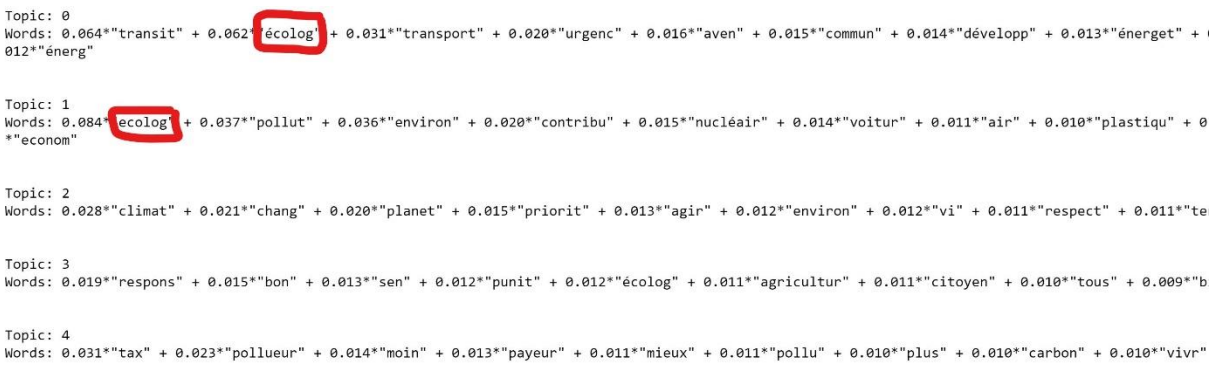


Figure 18 - Essai de topic modelling simple sur les données du thème "La transition écologique"

[16] 5.2.2 Missing accents in words: http://www.diva-portal.se/smash/get/diva2:1317274/FULLTEXT01.pdf

Puis, le texte est mis en minuscule, afin que 'Manger' et 'manger' soient considérés comme étant les mêmes mots. Ensuite, on enlève les « stop words » des textes. Les « stop words » sont des mots qui sont tellement communs qu'il est inutile de les utiliser, car ils n'apporteraient aucune plus-value dans la compréhension du texte. Les « stop words » typiques sont : « un », « le », « de », « du »… Les stop words utilisés ici sont ceux par défaut de l'implémentation française de Spacy.[17] En outre, on enlève les caractères spéciaux, les liens, et on enlève les '#' tout en gardant les mots qui les suivent directement, car ils peuvent être importants et contenir une information précieuse sur le thème, le style ou le ton d'un tweet.

Pour les prédictions faites en dehors de celle concernant le sexe des auteurs des textes, une lemmatisation a été effectuée. Une lemmatisation consiste à convertir un mot sous sa forme canonique, que l'on désigne sous le terme de lemme. Par exemple, « petite », « petits » et « petites » seront transformé en « petit ». Cela évite de considérer les quatre mots comme étant différents alors qu'ils ont exactement le même sens. En revanche, il me semblait logique et important ne pas effectuer de lemmatisation dans la prédiction du genre car dans ce cas la forme au féminin d'un mot est évidemment très importante.

ii. Vectorisation et TF-IDF

La technique de « bag of words » (sacs de mots) consiste à transformer chaque document (ici les contributions) en un vecteur de la même taille que le dictionnaire, dont la composante *i* indique le nombre d'occurrences du *i-ème* mot du dictionnaire dans le document. Un exemple peut être donné ci-dessous :

Document 1 : « voici un chat »
Document 2 : « voici un chien »
Document 3 : « je suis un humain »

La matrice représentant ces trois documents sera alors, en considérant tous les mots dans le dictionnaire :

	voici	*un*	*chat*	*chien*	*je*	*suis*	*humain*
Document 1	*1*	*1*	*1*	*0*	*0*	*0*	*0*
Document 2	*1*	*1*	*0*	*1*	*0*	*0*	*0*
Document 3	*0*	*1*	*0*	*0*	*1*	*1*	*1*

Cependant, au lieu de considérer chaque contribution (document) dans un vecteur en spécifiant si un mot est présent (en indiquant le nombre de fois n où le mot est présent dans le document) ou s'il ne l'est pas (en indiquant la valeur 0), on utilise ensuite une technique de TF-IDF (« Term Frequency – Inverse Document Frequency) qui permet de mesurer l'importance d'un mot dans un document au sein d'un corpus.

La représentation sous TF-IDF se calcul comme suit :

$$TF - IDF_{i,j} = tf_{i,j} * idf_i$$

Avec *i* l'index du mot et *j* l'index du document et :

$$tf = \frac{Nombre\ d'apparitions du\ mot\ dans\ le\ document}{Nombre\ de\ mots\ dans\ le\ document}$$

Il s'agit donc de la fréquence du mot dans le document.

[17] https://spacy.io/models/fr

$$idf = \log(\frac{Nombre\ total\ de\ documents\ dans\ le\ corpus}{Nombre\ de\ documents\ où\ le\ mot\ apparaît})$$

Cette technique permet de ne pas considérer avec la même importance les termes fréquents dans le corpus et ceux qui ne le sont pas. Par exemple, si le corpus traité porte sur les voitures, l'importance du mot « moteur », sûrement très présent dans le corpus, ne doit pas être la même que dans un corpus portant sur les vélos.

iii. *n*-gramme :

Un *n*-gramme est une sous-séquence de *n* termes dans un document.

En effet, on peut, dans un texte, considérer chaque unique mot (unigramme), comme dans l'exemple de sac de mots présenté ci-dessus, mais on peut aussi considérer les n mots se suivant. Par exemple, dans la phrase « voici un chien » on peut considérer chaque mot unique : « voici », « un », « chien » ou bien les bigrammes (2-grammes) : « voici un », « un chien ». Cela permet principalement de mieux comprendre le sens des phrases dans les cas où les mots considérés seuls peuvent induire des contre-sens, puisqu'avec les sacs de mots l'ordre des mots n'est pas pris en compte. Les exemples les plus évidents sont les négations, comme « ne suis » ou « suis pas ».

Dans mes classifications, j'ai considéré à la fois les unigrammes et les bigrammes. Ce choix a été fait après des tests de classification en évaluant les résultats de l'utilisation de n-grammes avec plusieurs possibilités pour *n*.

iv. Critères de classification

Je présente ci-dessous les critères de classification utilisés plus bas lors de l'application d'algorithmes.

- Accuracy[18] (Justesse), désigne la proportion de prédictions correctes effectuées par le modèle.

$$Accuracy = \frac{Nombre\ de\ prédictions\ correctes}{Nombre\ total\ de\ prédictions}$$

- Précision[19], désigne la proportion de classifications d'une classe *i* qui sont effectivement correctes.

$$Précision = \frac{Nombre\ de\ classifications\ correctes\ pour\ la\ classe\ i}{Nombre\ total\ de\ classifications\ pour\ i} = \frac{VP}{VP + FP}$$

Avec : VP le nombre de Vrais Positifs et FP le nombre de Faux Positifs.

- Recall (Rappel), désigne la proportion de résultats positifs réels pour une classe *i*.

$$Rappel = \frac{Nombre\ de\ classifications\ correctes\ pour\ la\ classe\ i}{Nombre\ total\ d'éléments\ dans\ la\ classe\ i} = \frac{VP}{VP + FN}$$

Avec FN le nombre de Faux Négatifs.

[18] https://developers.google.com/machine-learning/crash-course/classification/accuracy?hl=fr
[19] https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall?hl=fr

- AUC-ROC[20] : la courbe ROC (Receiver Operating Characteristic) est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Elle trace le taux de vrais positifs en fonction du taux de faux positifs.

$$Taux\ de\ Vrais\ Positifs\ (TVP) = \frac{VP}{VP + FN}$$

$$Taux\ de\ Faux\ Positifs\ (TVP) = \frac{FP}{FP + VN}$$

AUC (Area Under Curve), mesure l'intégralité de l'aire située sous la courbe ROC.

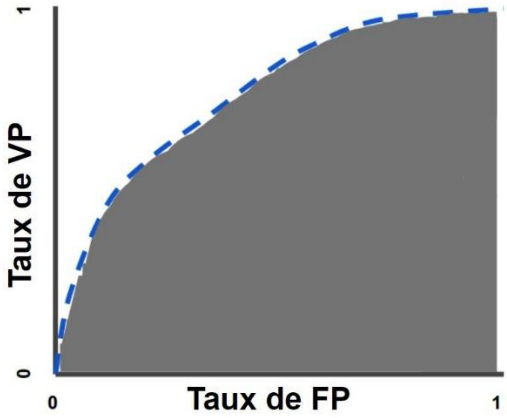


Figure 19-Mesure de l'AUC

Cette évaluation permet de déterminer la capacité, pour un classifieur, à différencier les différentes classes, et donc de mesurer la qualité des précisions du modèle.

[20] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=fr

v. Algorithmes utilisés :

Six principaux algorithmes ont été testé lors de mes expérimentations. En dehors des paramètres explicitement désignés, ce sont les paramètres par défaut de scikit-learn qui ont été utilisés :

- Un modèle linéaire de classification optimisé par descente de gradient stochastique, dont l'implémentation utilisée est celle de scikit-learn : SGDClassifier.[21] On peut l'utiliser en classification en multi-classes en combinant plusieurs classifieurs binaires et en utilisant une classification selon « One vs All », c'est-à-dire une classe contre toutes les autres.

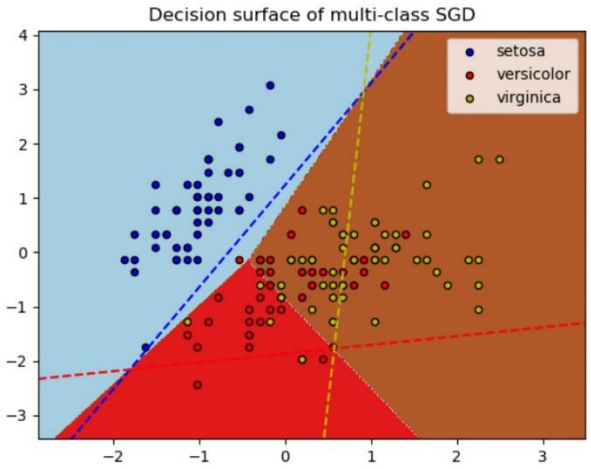


Figure 20 - Utilisation de SGDClassifier en multi-classes[22]

Les paramètres suivants ont été utilisés lors de mes expérimentations :

- loss='hinge'
- penalty='l2'
- alpha=1e-3
- max_iter=5
- tol=None

- Linear Support Vector Classification, un type de SVM[23] pour la classification, dont l'implémentation utilisée est celle de scikit-learn : LinearSVC.[24]

SVM (Support Vector Machine) consiste en une représentation des données d'entraînement sous forme de points dans un espace où les classes sont séparées par une marge aussi large que possible. De nouveaux exemples sont ensuite mappés dans ce même espace et sont classifiés dans une classe en fonction du côté de la marge dans lequel ils sont représentés.

[21] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html
[22] https://scikit-learn.org/stable/modules/sgd.html
[23] https://scikit-learn.org/stable/modules/svm.html#svm-classification
[24] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

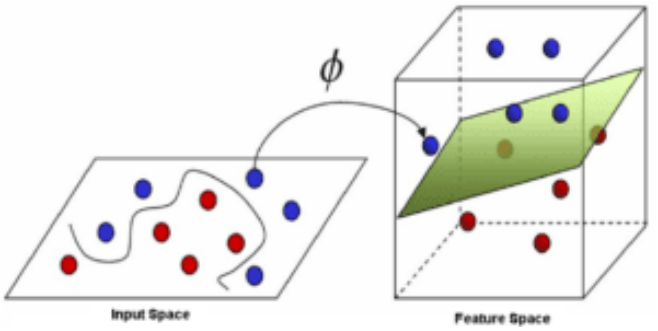


Figure 21 - SVM[25]

- Multinomial Naive Bayes, dont l'implémentation est MultinomialNB.[26]

L'algorithme Naive Bayes est un classifieur probabiliste inspiré par le théorème de Bayes, sous l'hypothèse que les attributs sont indépendants. Il permet de déterminer la classe d'un exemple suivant celle où le *posterior* est maximal, avec pour formule pour celui-ci :

$$P(c|x) = \frac{p(x|c)P(c)}{p(x)}$$

Avec *c* la classe et *x* l'exemple considéré.

MultinomialNB implémente l'algorithme Naive Bayes pour les données distribuées de façon multinomiale, et est une des variantes de Naive Bayes classiques utilisées dans la classification de texte, où les données sont généralement représentées comme des chiffres dans des vecteurs de mots.

- Régression Logistique[27], dont l'implémentation utilisée est celle de scikit-learn : LogisticRegression.[28]

La régression logistique fonctionne comme un modèle de régression binomiale, son but est de trouver le meilleur modèle pour décrire la relation entre la variable d'intérêt (le label) binomiale et un ensemble de variables indépendantes parmi les attributs. Une frontière de décision est ensuite construite afin de classifier les nouveaux exemples. L'avantage de cette méthode est qu'elle est très interprétable, puisqu'elle permet de déterminer les facteurs quantitatifs menant à une classification.

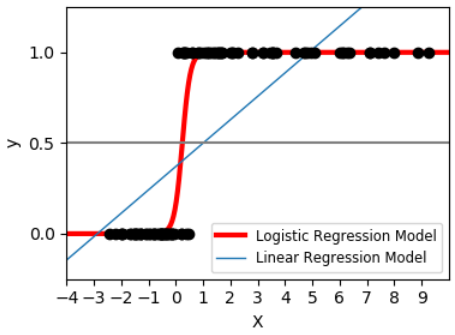


Figure 22 - Illustration de la régression logistique[29]

[25] https://dataanalyticspost.com/Lexique/svm/
[26] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
[27] https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
[28] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[29] https://scikit-learn.org/stable/auto_examples/linear_model/plot_logistic.html

- Arbres de Décision[30], dont l'implémentation de scikit-learn est : DecisionTreeClassifier.[31]

Comme son nom l'indique, le classifieur par arbres de décision prend une décision quant à la classification des exemples avec un modèle arborescent. Il divise l'échantillon en deux ou plusieurs ensembles homogènes (feuilles) en fonction des variables d'entrées disponibles différenciant le plus significativement chaque classe. Pour choisir un différenciateur (prédicteur), l'algorithme prend en compte tous les attributs et effectue une division binaire sur ceux-ci, c'est-à-dire qu'il les divise selon leur valeurs pour des données catégorielles, et selon un seuil pour les données continues. Il choisit ensuite le différenciateur avec la meilleure accuracy, et ce récursivement jusqu'à ce que toutes les données soient séparées dans les feuilles en fonction de leur classe, ou jusqu'à atteindre la profondeur maximale de l'arbre.

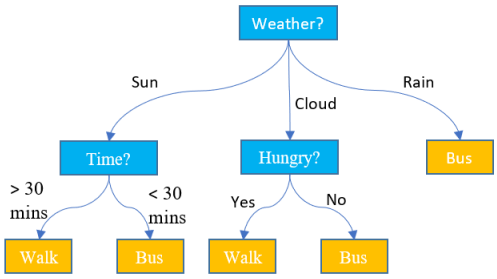


Figure 23 - Exemple d'arbre de décision[32]

En dehors des paramètres par défaut, le paramètre suivant a été utilisé lors de mes expérimentations :

- max_depth=5

- Random Forest[33], dont l'implémentation utilisée est aussi celle de scikit-learn : RandomForestClassifier.[34]

Pour la classification, Random Forest est une méthode d'Ensemble Learning qui fonctionne en construisant une multitude d'arbres de décision de façon aléatoire et en classifiant le nouvel exemple en fonction des décisions prises par ces arbres. Random Forest permet de corriger le fait que les arbres de décision aient tendance à overfitter.

[30] https://scikit-learn.org/stable/modules/tree.html#tree
[31] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[32] https://www.displayr.com/what-is-a-decision-tree/
[33] https://scikit-learn.org/stable/modules/ensemble.html#forest
[34] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

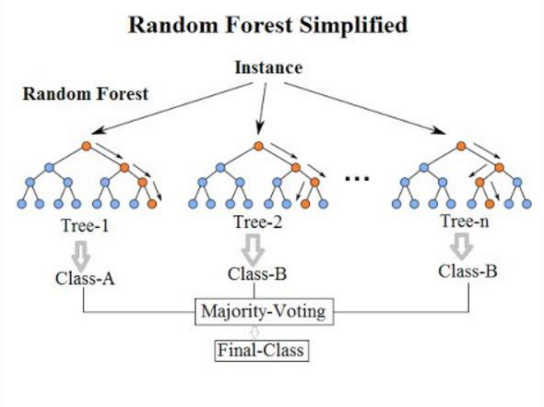


Figure 24 - Illustration simplifiée de Random Forest[35]

En dehors des paramètres par défaut, les paramètres suivants ont été utilisés lors de mes expérimentations :

- n_estimators=200
- max_depth=3

vi. Jeux de données non-équilibrés

La plupart des jeux de données utilisés n'étaient pas équilibrés, c'est-à-dire que les classes n'étaient pas représentées dans les mêmes proportions. Cela pose un problème lors d'une classification, surtout lorsque la classification est faite en « multi-classes », c'est-à-dire que le problème n'est pas à deux classes (ce serait alors une classification binaire) mais à plus de deux classes.

Lorsque ce problème est rencontré, plusieurs solutions sont possibles :

- L'utilisation d'un algorithme prenant en compte cela, notamment SVM avec un paramètre 'class_weight'='balanced' dans son implémentation dans scikit-learn.
- Le sur-échantillonnage des classes minoritaires ou le sous-échantillonnage des classes majoritaires, ce qui consiste à tirer n échantillons dans les classes minoritaires de façon aléatoire, *n* étant le nombre d'individus dans la classe majoritaire. Cette solution peut être privilégiée lorsque des classes ne sont pas « trop » minoritaires, c'est-à-dire que leurs proportions sont du même ordre.
- La suppression pure et simple des classes minoritaires si celles-ci sont trop minoritaires et que réaliser un sur-échantillonnage serait absurde, car trop d'exemples seraient répétés, et alors l'algorithme serait évalué sur des exemples qu'il aurait déjà 'vu' lors de la phase d'apprentissage.

[35] https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

b. Application de modèles via les données twitter

Après avoir explicité la méthode générale à mettre en place dans une démarche de classification en utilisant des modèles de NLP et de Machine Learning, nous allons maintenant les appliquer, en entrainant d'abord sur les données étiquetées que j'ai pu trouver, puis en appliquant les algorithmes entrainés sur les données ouvertes du grand débat national.

Nous présentons d'abord les résultats pour les classifications de parti avec la base #Elysée2017, puis avec la base des parlementaires. Ensuite, nous étudierons le passage d'une base de données de type « twitter » à une base de données de type « contribution ». Enfin, nous appliquerons les mêmes méthodes pour la classification de la position du contributeur par rapport au mouvement des gilets jaunes.

i. Classification du parti politique avec #Elysée2017

Nous utilisons ici la base #Elysée2017 afin d'essayer d'extraire le parti politique de chaque contributeur du grand débat national. Voici les résultats génériques lors de l'entrainement et de l'évaluation sur l'accuracy de chaque algorithme avec une validation croisée de trois folds (trois est peu mais plus aurait été trop long étant donné la puissance de mon ordinateur personnel), avant sur-échantillonnage des classes minoritaires, car, comme on l'a vu, la base de données n'est pas équilibrée :

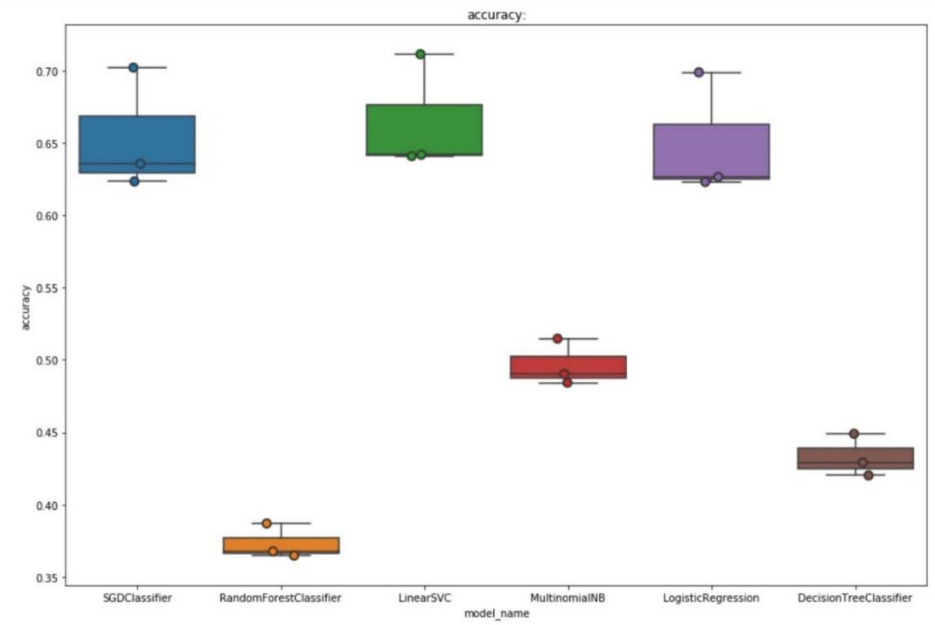


Figure 25 - Graphe des accuracy sur 3-folds CV avec #Elysée2017 sans sur-échantillonnage

```
Accuracy:
model_name
DecisionTreeClassifier    0.432722
LinearSVC                 0.664817
LogisticRegression        0.649565
MultinomialNB             0.496447
RandomForestClassifier    0.373032
SGDClassifier             0.653819
```

Figure 26 - Tableau des moyennes d'accuracy de chaque algorithme pour l'entrainement sur la base #Elysée2017 sans sur-échatillonnage

Dans les mêmes conditions mais après sur-échantillonnage des classes minoritaires :

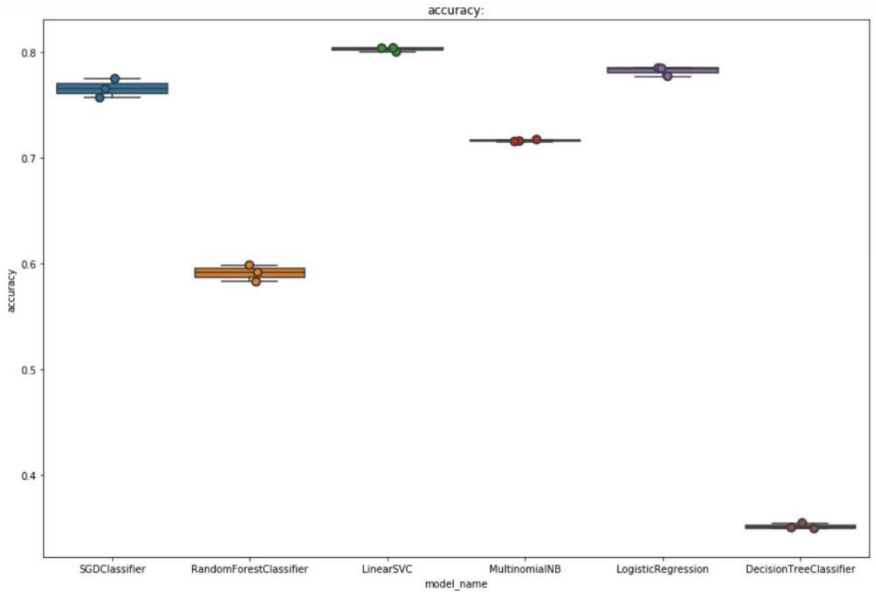


Figure 27 - Graphe des résultats Accuracy de chaque algorithme sur #Elysée2017 avec sur-échantillonnage

```
Accuracy:
model_name
DecisionTreeClassifier    0.351654
LinearSVC                 0.802438
LogisticRegression        0.782097
MultinomialNB             0.716127
RandomForestClassifier    0.591083
SGDClassifier             0.765517
Name: accuracy, dtype: float64
```

Figure 28 - Tableau des accuracy pour chaque algorithme avec #Elysee2017 après sur-échantillonnage

On observe comme attendu qu'après sur-échantillonnage les résultats de classification sont bien meilleurs qu'avant. Nous allons donc nous concentrer sur les derniers résultats avec cette propriété.

Trois algorithmes se détachent plus particulièrement par rapport aux autres dans leurs résultats de classification :

- LinearSVC avec 80% d'Accuracy
- Régression Logistique avec 78% d'Accuracy
- SGDClassifier avec 77% d'Accuracy

C'est donc ceux-là qui sont considérés dans la suite de la classification. Il est à noter que ces résultats sont relativement satisfaisant pour un problème à cinq classes, car un modèle aléatoire n'aurait eu que 20% d'accuracy.

Nous pouvons tout de même noter que les performances des arbres de décision sont très faibles, certainement à cause d'overfitting, et que Random Forest corrige cela mais pas assez pour rivaliser avec les algorithmes ci-dessus ayant les meilleures performances.

On peut aussi afficher les matrices de confusion associées à ces algorithmes en entrainement sur les données sur-échantillonnées. Pour plus de clarté, j'ai choisi de n'afficher ici que celle obtenue avec LinearSVC, avec le rapport de classification, mais l'ensemble des matrices de confusion peut aussi être retrouvé sur le github du projet dont le lien se situe en annexe.

```
------------ LinearSVC ---------------------------------------
              precision    recall  f1-score   support

          ps       0.86      0.90      0.88       970
          em       0.72      0.65      0.68       949
          lr       0.82      0.81      0.81       944
          fi       0.78      0.78      0.78       948
          fn       0.84      0.89      0.87       927

    accuracy                           0.81      4738
   macro avg       0.80      0.81      0.80      4738
weighted avg       0.80      0.81      0.80      4738
```

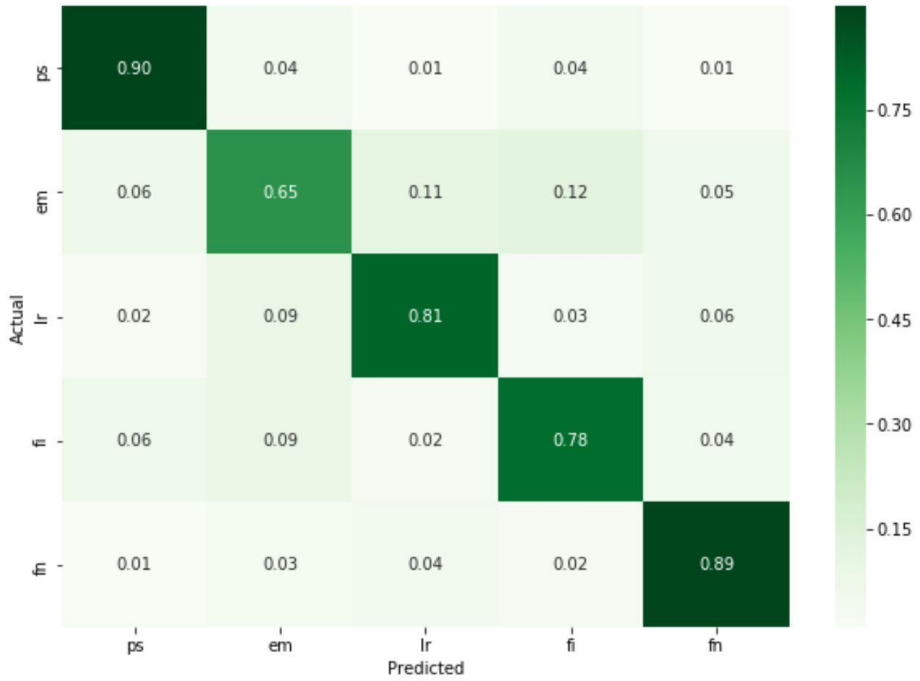


Figure 29 - Matrice de confusion liée aux données #Elysée2017 sur-échantillonnées en entrainement pour la classification de partis politiques avec LinearSVC

Même si ce n'est pas le cas pour tous les partis, on peut remarquer certains résultats intéressants concernant cette matrice de confusion :

Le Parti Socialiste est le parti le mieux classifié, avec 90% d'accuracy. Surtout, on peut remarquer que les partis les plus prédits pour le parti socialiste, en dehors de lui-même, est la République en Marche et la France Insoumise, à 4% chacun. Cela semble cohérent par rapport à l'échiquier politique français. De même, Les Républicains, à 81% d'Accuracy, est le plus « mal prédit » sur le Front National et La République en Marche, avec respectivement 6% et 9%. En revanche, La République en Marche est elle assez mal prédite, avec 65% d'accuracy, et mal classifiés comme Les Républicains à 11%, et surtout, plus surprenant, comme France Insoumise à 12%.

Ainsi, certains résultats semblent cohérents pas rapport à la vie réelle et la vie politique française, mais d'autres semblent assez erratiques.

J'ai ensuite appliqué les meilleurs algorithmes sur les contributions. Voici ces résultats de classification du parti politique lié à chaque participant au Grand Débat National :

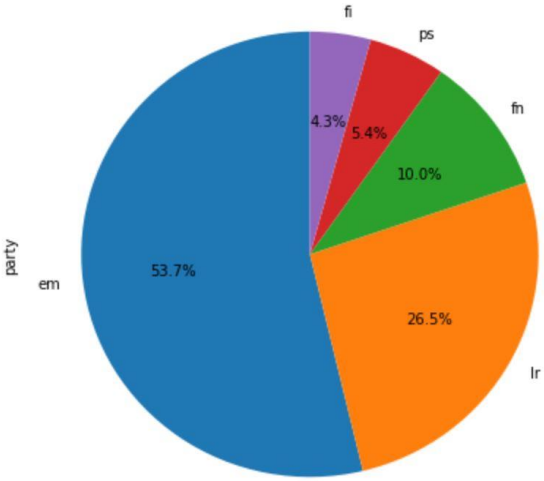


Figure 30- Résultat de classification avec LinearSVC et #Elysée2017

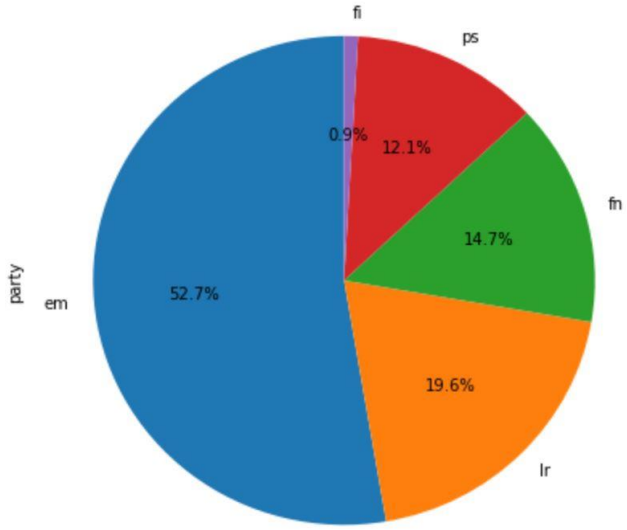


Figure 31- Résultat de classification avec Régression Logistique et #Elysée2017

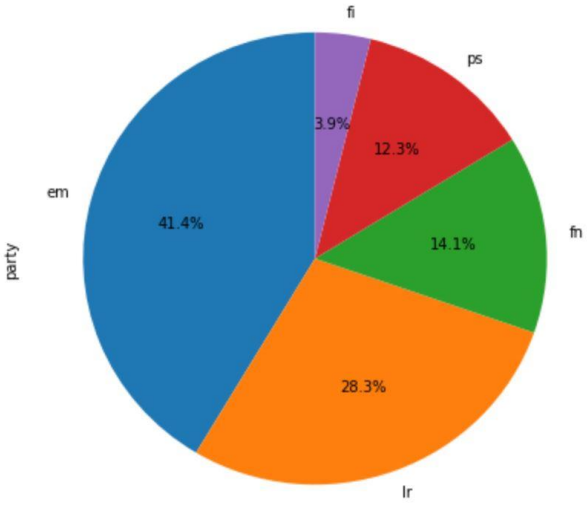


Figure 32- Résultat de classification avec #Elysée2017 et SGD

On note d'abord que ces résultats ne semblent pas totalement absurdes dans leurs proportions. De plus, les proportions de classification sont toutes du même ordre pour tous les algorithmes pour chaque parti. La République en Marche est en tête pour tous les algorithmes, suivie des Républicains et du Front National. Cependant, on observe que la proportion La République En Marche parait très importante et que l'« ordre » des partis par rapport aux résultats du premier tour des élections présidentielles de 2017 n'est pas respecté.

En outre, il est difficile de mesurer la réelle bonne classification de chaque contribution étant donné que les participations au grand débat national n'étaient malheureusement pas annotées. De plus, étant donné le temps pouvant être consacré à ce projet, il aurait été bien trop fastidieux, et difficile compte tenu des hypothèses politiques à faire, de les annoter à la main.

En revanche, afin d'essayer de mesurer si les « meilleures » classifications étaient correctes ou non, j'ai aussi affiché le nuage de mots associés aux 100 classifications les plus 'certaines' (par decision_function[36] et distance par rapport à chaque parti) pour chaque algorithme et chaque parti. Ces nuages de mots ont été réalisés pour chaque algorithme considéré. Cependant, pour plus de clarté, seuls ceux réalisés par classification avec Régression Logistique sont affichés ici.

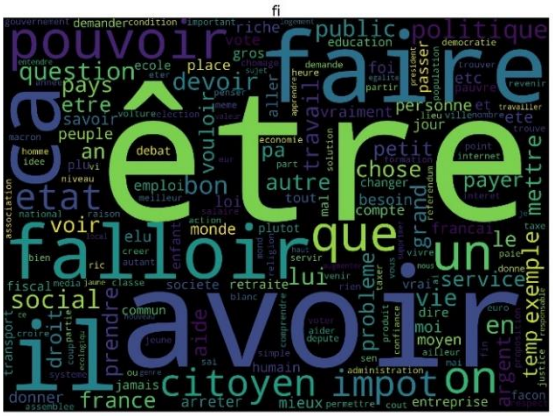


Figure 34- Nuage de mots lié à la classification "fi" avec la base #Elysée2017

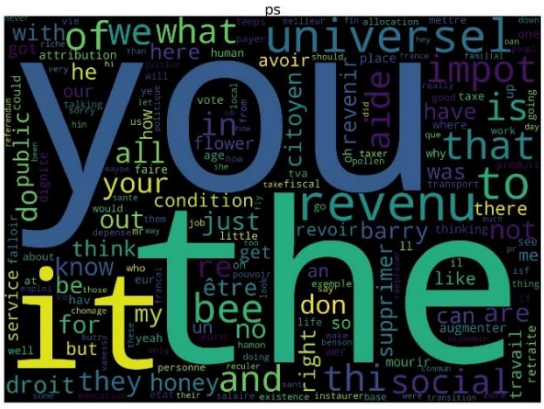


Figure 33 - Nuage de mots lié à la classification "ps" avec la base #Elysée2017

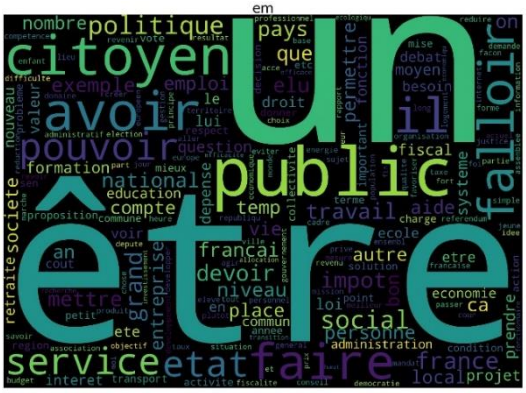


Figure 35 - Nuage de mots lié à la classification "em" avec la base #Elysée2017

Figure 36 - Nuage de mots lié à la classification "lr" avec la base #Elysée2017

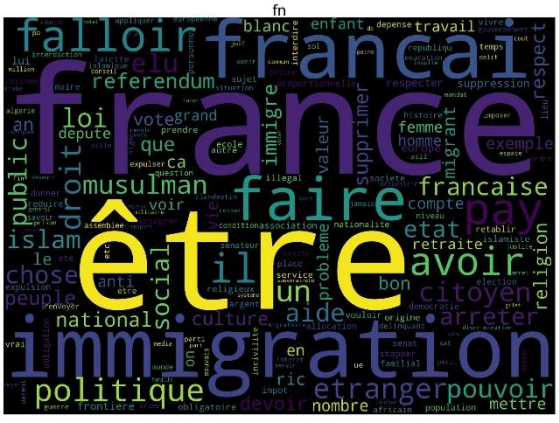


Figure 37 - Nuage de mots lié à la classification "fn" avec la base #Elysée2017

[36] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression.decision_function

Comme nous pouvons le voir, les nuages de mots rendent compte de certaines caractéristiques de partis qui semblent être cohérentes et intéressantes. Les mots « revenu » et « universel » sont bien présents dans la classification par rapport au Parti Socialiste, et, en effet, une des principales propositions du candidat socialiste à l'élection présidentielles de 2017 était bien le revenu universel. De plus, les mots « français », « France » et « immigration » sont bien présents, et occupent une place assez importante, dans la classification relative au Front National, et cela semble cohérent pour des raisons évidentes.

Cependant, ces nuages de mots rendent compte d'un possible problème de pré-traitement, avec les mots « être » ou « avoir » qui ont une place bien trop importante et qui n'ont surtout pas été filtrés. Le fait que ces mots n'aient pas été filtrés avec les stop words s'explique par une raison simple : l'utilisation des stop words se faisait avant la lemmatisation par spacy dans ma fonction de pré-traitement des données textuelles. Ainsi, toutes les conjugaisons de « être » ou « avoir » n'étant évidemment pas présents dans les stop words de spacy, si les infinitifs de ces deux verbes étaient renvoyés par la fonction de lemmatisation ils ne pouvaient pas être filtrés à la fin. Cependant, on pourrait discuter du fait de les supprimer ou non, car étant donné que j'utilise les bigrammes en plus des unigrammes, ces verbes à l'infinitif peuvent rendre compte de certaines caractéristiques intéressantes par ailleurs.

Ainsi, pour la classification de partis avec la base #Elysée2017, nous avons donc pu observer que, malgré de bons résultats en entrainement pour cinq classes, les proportions de classification étaient difficilement mesurables sur la base de contributions, principalement parce que celles-ci n'étaient malheureusement pas annotées.

Cependant, comme nous avons pu l'observer malgré certains soucis de pré-traitement possibles, les 100 classifications les plus « certaines » semblaient cohérentes via l'affichage des nuages de mots. On aurait pu, alors, essayer d'apprendre sur ces nouvelles données, qui sont bien elles des contributions et pas des données de type twitter, avec comme nous l'avons vu une taille différente et des messages qui peuvent être bien différents dans leur nature.

Nous allons maintenant réaliser le même type d'exercice de classification que nous venons d'étudier sur la base #Elysée2017, mais cette fois-ci sur la base des parlementaires.

ii. Classification du parti politique avec la base liée aux parlementaires

A l'instar de ce qui a été réalisé sur la base #Elysée2017, Voici les résultats génériques lors de l'entrainement et de l'évaluation sur l'accuracy de chaque algorithme avec une validation croisée de 10 folds, avant sur-échantillonnage des classes minoritaires, car, comme on l'a vu, la base de données n'est pas équilibrée :

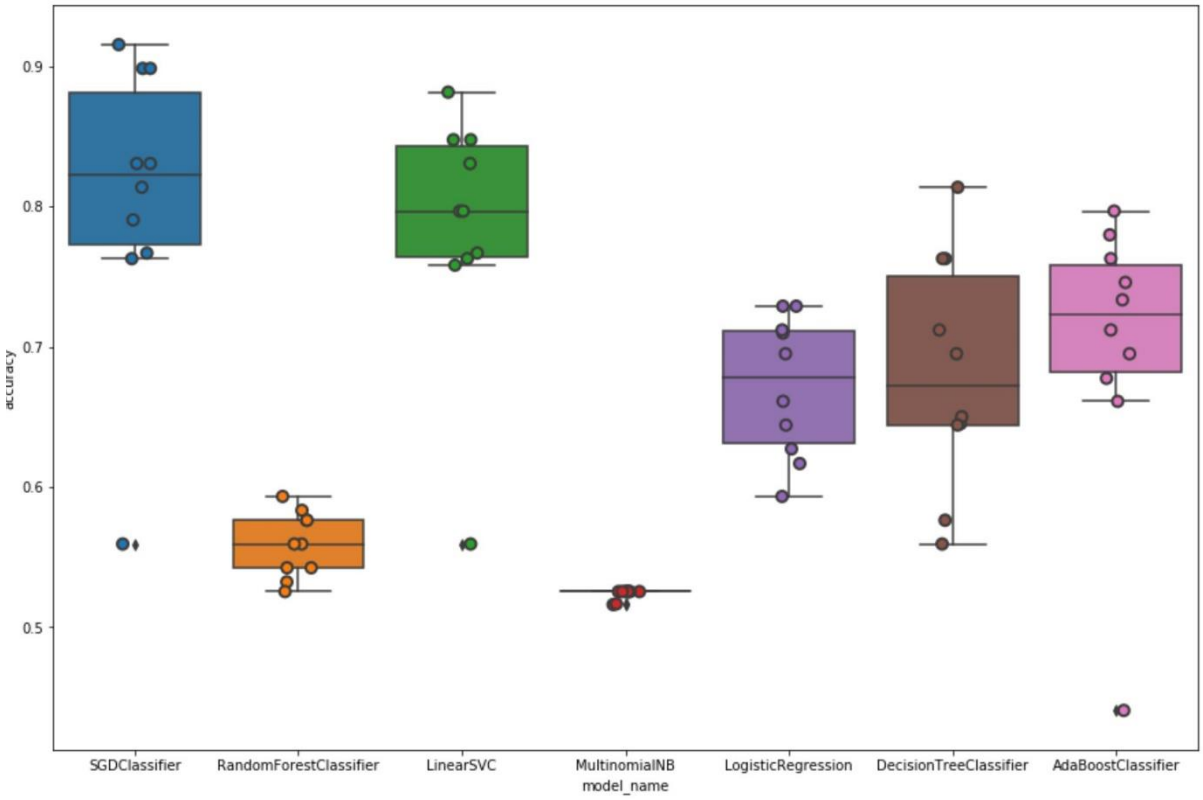


Figure 38 - Accuracy sur 10 folds des différents algorithmes pour la classification de partis sur la base des parlementaires avant sur-échantillonnage

```
model_name
AdaBoostClassifier        0.700397
DecisionTreeClassifier    0.682059
LinearSVC                 0.784677
LogisticRegression        0.671617
MultinomialNB             0.523619
RandomForestClassifier    0.559017
SGDClassifier             0.806546
```

Figure 39 - Accuracy moyenne pour les différents algorithmes pour la classification de partis sur la base des parlementaires avant sur-échantillonnage

Voici les résultats après sur-échantillonnage :

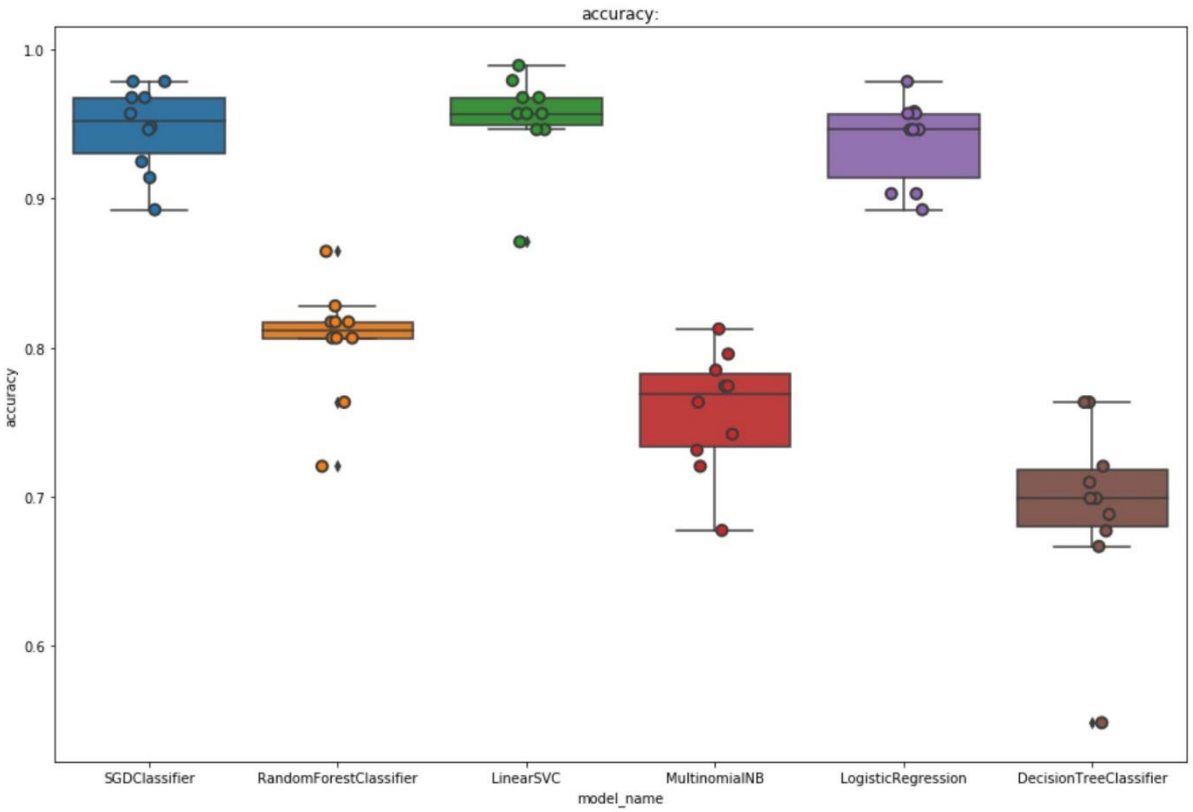


Figure 40 - Accuracy sur 10 folds des différents algorithmes pour la classification de partis sur la base des parlementaires après sur-échantillonnage

```
Accuracy:
model_name
DecisionTreeClassifier    0.693515
LinearSVC                 0.953831
LogisticRegression        0.938844
MultinomialNB             0.757594
RandomForestClassifier    0.804738
SGDClassifier             0.947480
```

Figure 41 - Accuracy moyenne pour les différents algorithmes pour la classification de partis sur la base des parlementaires après sur-échantillonnage

Nous observons comme attendu que les résultats de classification sont bien meilleurs après sur-échantillonnage qu'avant. On observe dans le second cas de très bons résultats d'entrainement, surtout pour trois classes, avec trois algorithmes qui se détachent plus particulièrement des autres :

- LinearSVC avec 95% d'accuracy
- Régression Logistique avec 94% d'accuracy
- SGDClassifier avec 95% d'accuracy

Ce sont donc ces modèles qui ont été utilisés par la suite pendant l'application sur les contributions au grand débat national. Il est à noter qu'un modèle de classification aléatoire sur trois classes n'aurait, en moyenne, que 33% d'accuracy.

On peut aussi noter les mêmes mauvaises performances que sur la base #Elysée2017 des arbres de Decision et de Random Forest, sûrement pour les mêmes raisons d'overfitting.

On peut, comme dans la partie précédente, présenter la matrice de confusion liée à la base de données twitter des parlementaires, avec LinearSVC :

```
------------ LinearSVC --------------------------------------
              precision    recall  f1-score   support

       LaREM       0.93      0.92      0.93       101
          LR       0.98      0.95      0.97       104
         SOC       0.94      0.98      0.96       103

    accuracy                           0.95       308
   macro avg       0.95      0.95      0.95       308
weighted avg       0.95      0.95      0.95       308
```

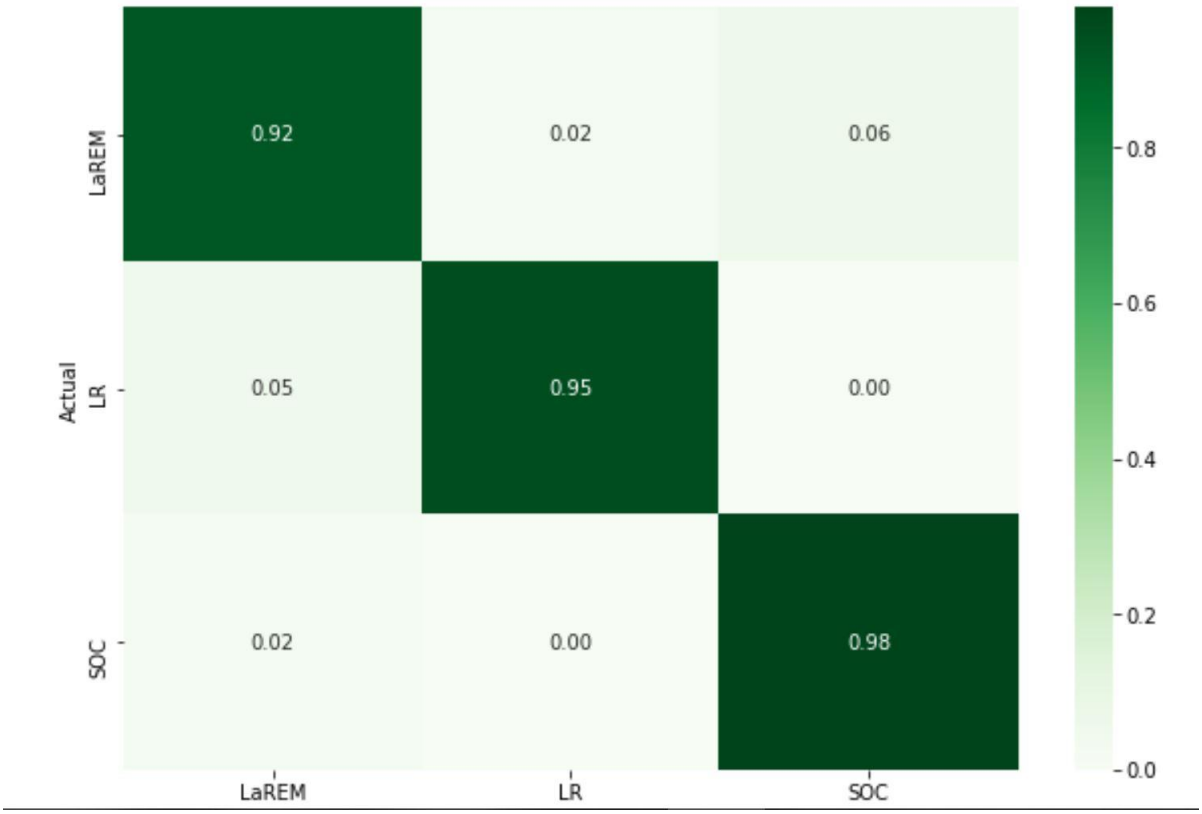


Figure 42 - Matrice de confusion à l'entrainement sur LinearSVC et la base de données des parlementaires à l'entrainement après sur-échantillonnage

De même qu'avec la matrice de confusion présentée précédemment sur la base #Elysée2017, certains résultats visibles dans cette matrice de confusion montrent des proximités entre partis qui sont celles de la vie réelle. Ainsi, Le Parti Socialiste, à 98% d'accuracy, n'est pas du tout classifié comme étant Les Républicains mais à 2% comme étant La République En Marche, de même, LR, à 95% d'accuracy, n'est pas du tout classifié comme étant SOC mais à 5% LaREM, et, enfin, LaREM, à 92% d'accuracy, est classifié à 6% comme SOC et 2% comme LR.

Ces résultats sont exactement ceux qui pourraient correspondre à l'échiquier politique français.

Voici les résultats en termes de proportions pour chaque parti attribué aux contributions après application sur les données du grand débat national :

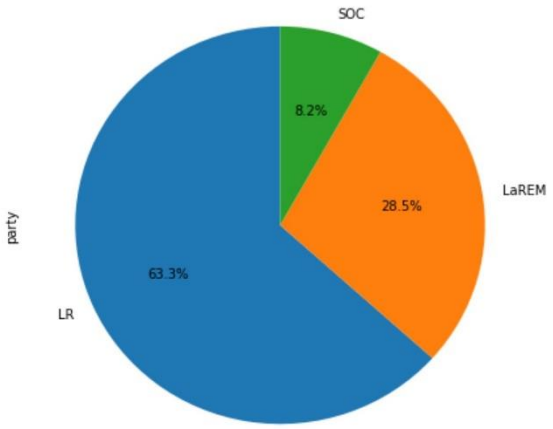


Figure 43 – Classification après entrainement avec LinearSVC sur la base des parlementaires

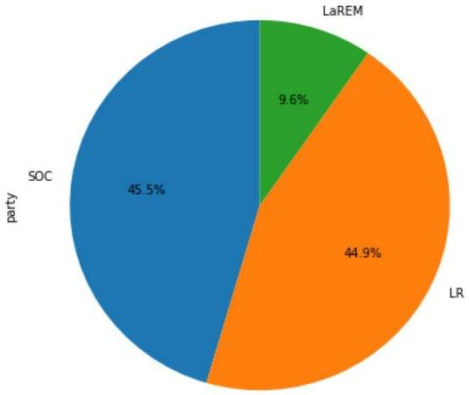


Figure 44 – Classification après entrainement avec LogisticRegression sur la base des parlementaires

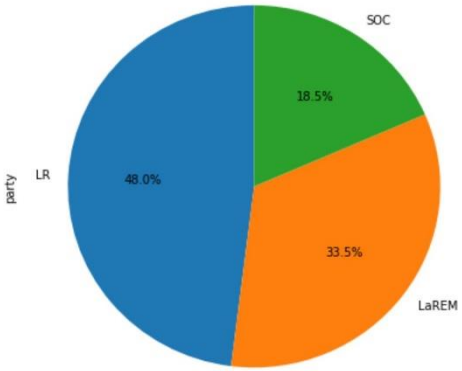


Figure 45 – Classification après entrainement avec SGDClassifier sur la base des parlementaires

Nous observons des résultats de classifications semblant incohérents. D'une part, les résultats de classification ne sont pas les mêmes pour tous les algorithmes, et d'autre part les proportions semblent assez absurdes, notamment avec un parti très minoritaire par rapport aux deux autres différent pour chaque algorithme, par exemple « SOC » à 7.3% avec LinearSVC et « LaREM » à 7.3% avec la régression logistique.

Comme nous l'avons vu dans le même exercice de classification réalisé avec la base #Elysée2017, il est assez difficile de mesurer la réelle performance d'un tel modèle étant donné que les données de base n'étaient pas annotées. En revanche, ici encore, on peut afficher les nuages de mots associés aux 100 classifications les plus certaines afin d'étudier la cohérence de mes classifications. Seules les classifications réalisées par Régression Logistique sont affichées ici.

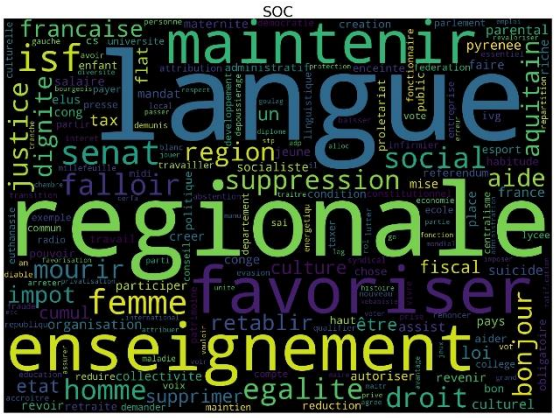


Figure 47 - Nuage de mots lié à la classification "SOC" avec la base liée aux parlementaires

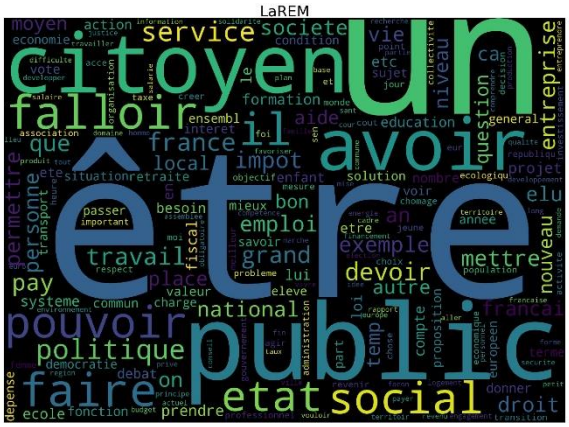


Figure 46 - Nuage de mots lié à la classification "LaREM" avec la base liée aux parlementaires

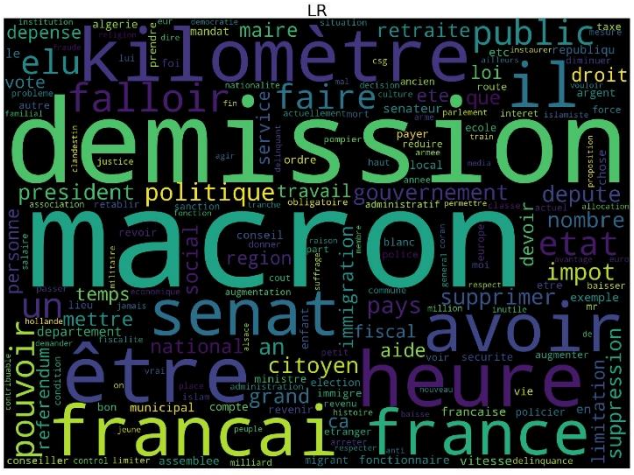


Figure 48 - Nuage de mots lié à la classification "LR" avec la base liée aux parlementaires

Tout d'abord, nous pouvons observer les mêmes problèmes de pré-traitement que précédemment, et qui amènent aux mêmes conclusions suivant l'utilisation des stop words avant ou après lemmatisation, donc nous ne nous attardons pas sur ce point.

Ensuite, ces nuages de mots obtenus après entrainement sur la base des parlementaires et application sur les données du grand débat national permettent eux aussi de voir que certaines caractéristiques ou revendications de partis peuvent être retrouvées. Ainsi, on peut observer les mots « kilomètre » et « heure » dans le nuage de mots lié à la classification « Les Républicains », or, on peut se dire que cela correspond à la demande de retour au 80km/h sur les routes de France, qui était en effet très forte de la part des élus de ce parti. On pourrait aussi se dire que si « isf » est présent dans le nuage de mots pour la classification du parti socialiste, c'est parce que les élus socialistes étaient opposés à sa suppression par la majorité.

Nous pourrions, je pense, étudier plus en détail ces nuages de mots et trouver plus de cohérence avec les revendications effectives des différents partis, en faisant des hypothèses plus fortes ou en étudiant plus en détail le paysage politique français actuel, par exemple avec le mot « France » très présent dans le nuage de mots des républicains et pas dans celui lié au parti socialiste ou bien à l'inverse les mots « égalité », « femme » et « homme » présent dans le nuage socialiste et pas dans les autres.

Cependant, j'ai décidé de présenter les résultats qui me semblaient les plus évidents en termes de cohérence sans faire d'hypothèses politiques très fortes.

Ainsi, nous avons pu observer, à l'instar de la classification liée à la base de données #Elysée2017, que malgré des résultats en entrainement assez satisfaisant, il était assez difficile d'établir les réelles performances des modèles après application sur les contributions. Cependant, les nuages de mots liés aux 100 classifications les plus « certaines » permettent aussi d'observer des caractéristiques intéressantes et ne semblent pas faire de contre sens. Ils auraient donc pu être utilisés pour un entrainement ultérieur.

Nous allons maintenant essayer d'étudier le passage de données de type « twitter », en entrainement, à des données de type « contributions.

c. Justification du passage d'un type de données à un autre

Puisque les données de type 'twitter' et celles des contributions ne sont pas les mêmes, comme nous l'avons vu par exemple dans la partie II. B. iv. de ce rapport en comparant les tailles des réponses aux questions ouvertes du grand débat aux tailles limites de tweets, il faut tout de même évaluer le fait d'entrainer sur une base de données twitter et de l'appliquer sur les contributions. Afin de justifier cela, j'ai utilisé un label présent à la fois dans les bases de données twitter à ma disposition (celle des parlementaires et celle relative à #Elysée2017) et dans la base de données 'Entendre la France' : le genre des individus.

J'ai alors évalué des modèles sur les bases de données twitter, en utilisant la base #Elysée2017.

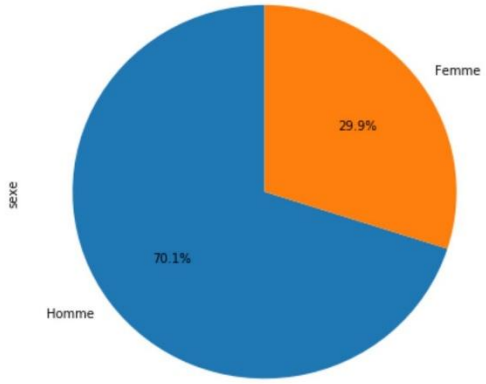


Figure 49 - Rappel des proportions de chaque sexe dans la base #Elysée2017

Je rappelle ci-contre les proportions de chaque sexe dans la base de données liée à #Elysée2017. Comme nous pouvons le voir, la base n'est pas équilibrée, avec 70,1% d'Hommes et 29.9% de Femmes.

D'une part sur les jeux de données non équilibrés, voici les résultats de classification à l'entrainement :

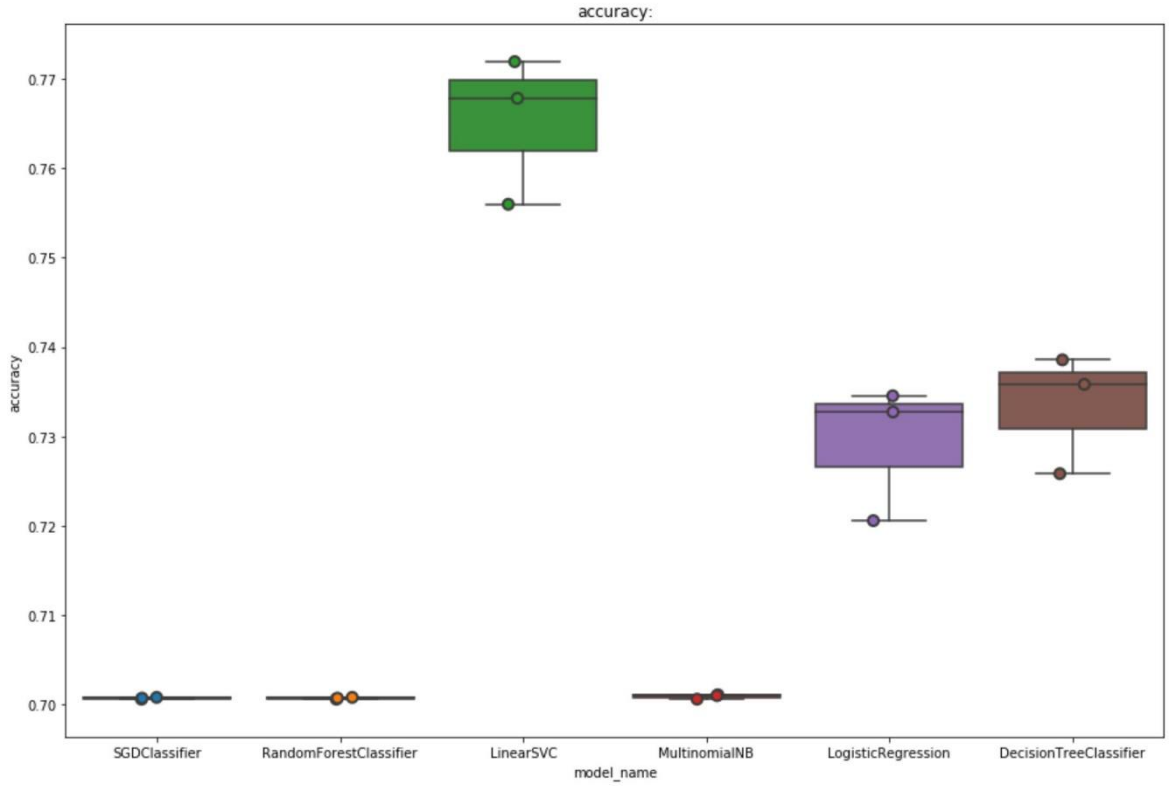


Figure 50 - Score d'Accuracy pour chaque algorithme sur 3-CV pour la classification du sexe sur la base #Elysée2017 à l'entrainement sans sur-échantillonnage de la classe minoritaire

Accuracy:

model_name	
DecisionTreeClassifier	0.733391
LinearSVC	0.765263
LogisticRegression	0.729261
MultinomialNB	0.700845
RandomForestClassifier	0.700653
SGDClassifier	0.700653

Figure 52 - Tableau des accuracy moyennes pour chaque algorithme sur 3-CV pour la classification du sexe à l'entrainement sur la base #Elysée2017 sans sur-échantillonnage de la classe minoritaire

ROC AUC:

model_name	
DecisionTreeClassifier	0.578946
LinearSVC	0.768073
LogisticRegression	0.778374
MultinomialNB	0.685015
RandomForestClassifier	0.713844
SGDClassifier	0.776652

Figure 51 - Tableau des accuracy moyennes pour chaque algorithme sur 3-CV pour la classification du sexe à l'entrainement sur la base #Elysée2017 sans sur-échantillonnage de la classe minoritaire

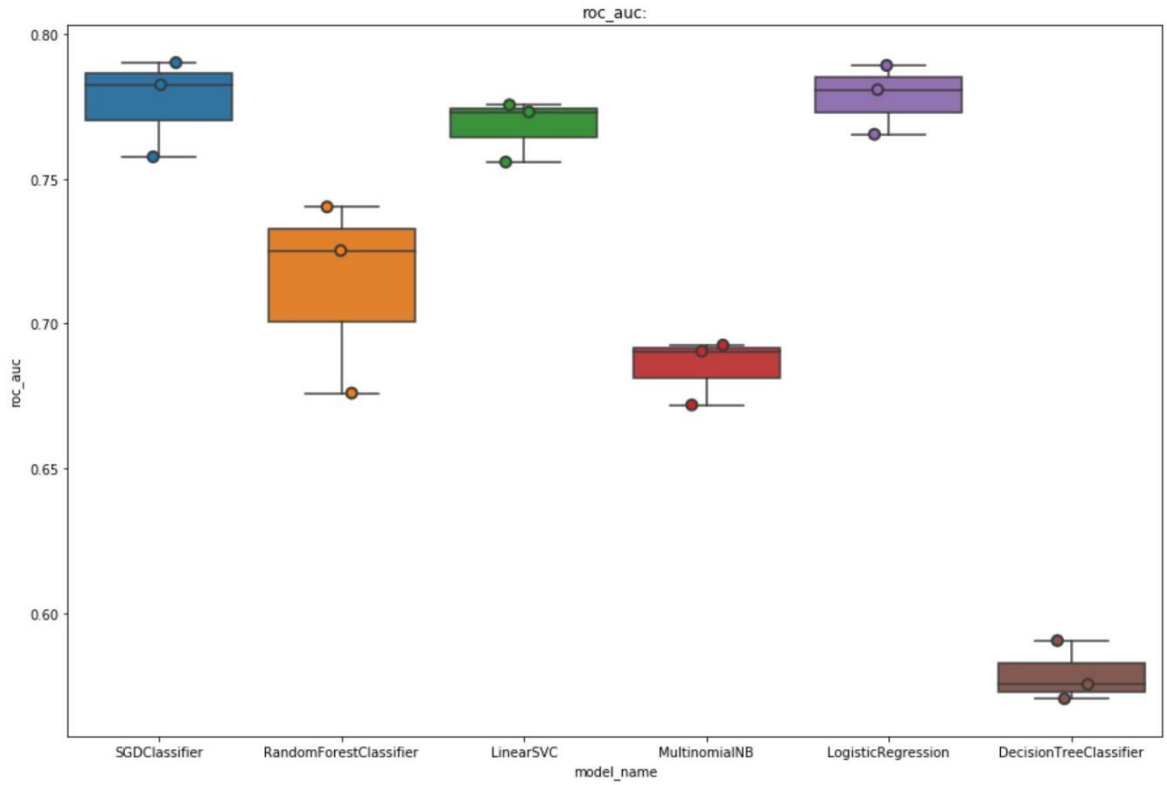


Figure 53 - Score d'AUC pour chaque algorithme sur 3-CV pour la classification du sexe sur la base #Elysée2017 à l'entrainement sans sur-échantillonnage de la classe minoritaire

D'autre part sur les jeux de données équilibrés via sur-échantillonnage de la classe minoritaire, en l'occurrence les femmes. Voici les résultats de classification à l'entrainement :

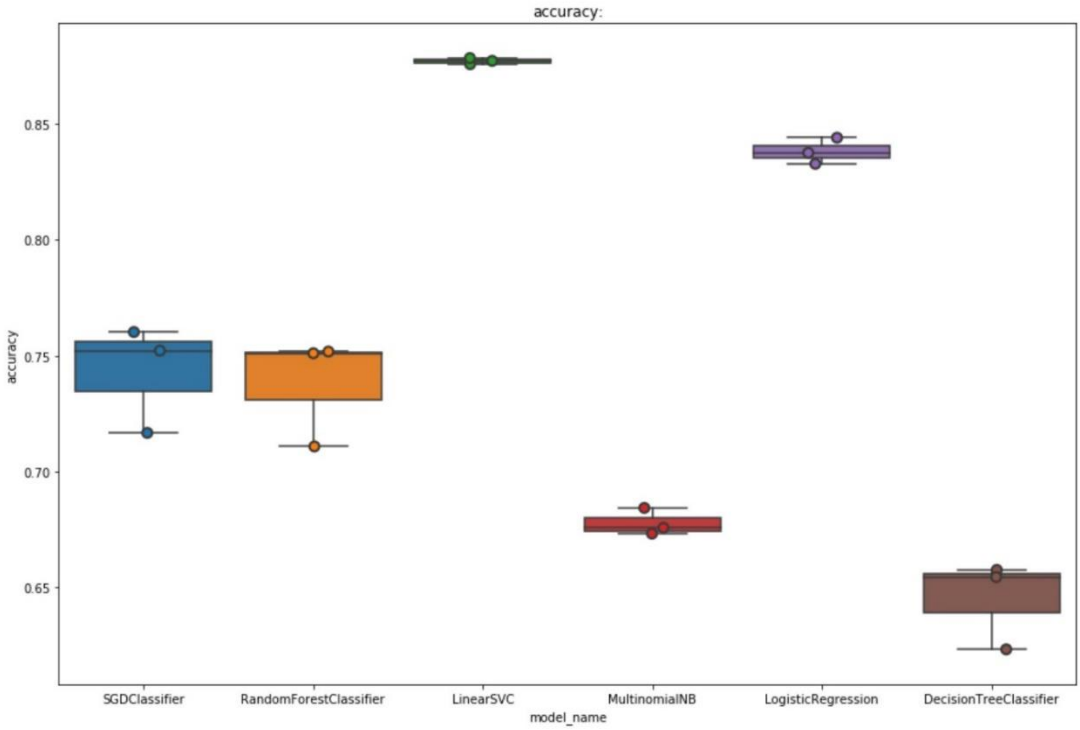


Figure 54 - Score d'Accuracy pour chaque algorithme sur 3-CV pour la classification du sexe sur la base #Elysée2017 à l'entrainement avec sur-échantillonnage de la classe minoritaire

```
Accuracy:
model_name
DecisionTreeClassifier    0.645174
LinearSVC                 0.877021
LogisticRegression        0.838039
MultinomialNB             0.677788
RandomForestClassifier    0.737875
SGDClassifier             0.743008
```

Figure 56 - Tableau des accuracy moyennes pour chaque algorithme sur 3-CV pour la classification du sexe à l'entrainement sur la base #Elysée2017 avec sur-échantillonnage de la classe minoritaire

```
ROC AUC:
model_name
DecisionTreeClassifier    0.664482
LinearSVC                 0.945317
LogisticRegression        0.910624
MultinomialNB             0.874803
RandomForestClassifier    0.814192
SGDClassifier             0.836610
```

Figure 55 - Tableau des scores AUC moyens pour chaque algorithme sur 3-CV pour la classification du sexe à l'entrainement sur la base #Elysée2017 avec sur-échantillonnage de la classe minoritaire

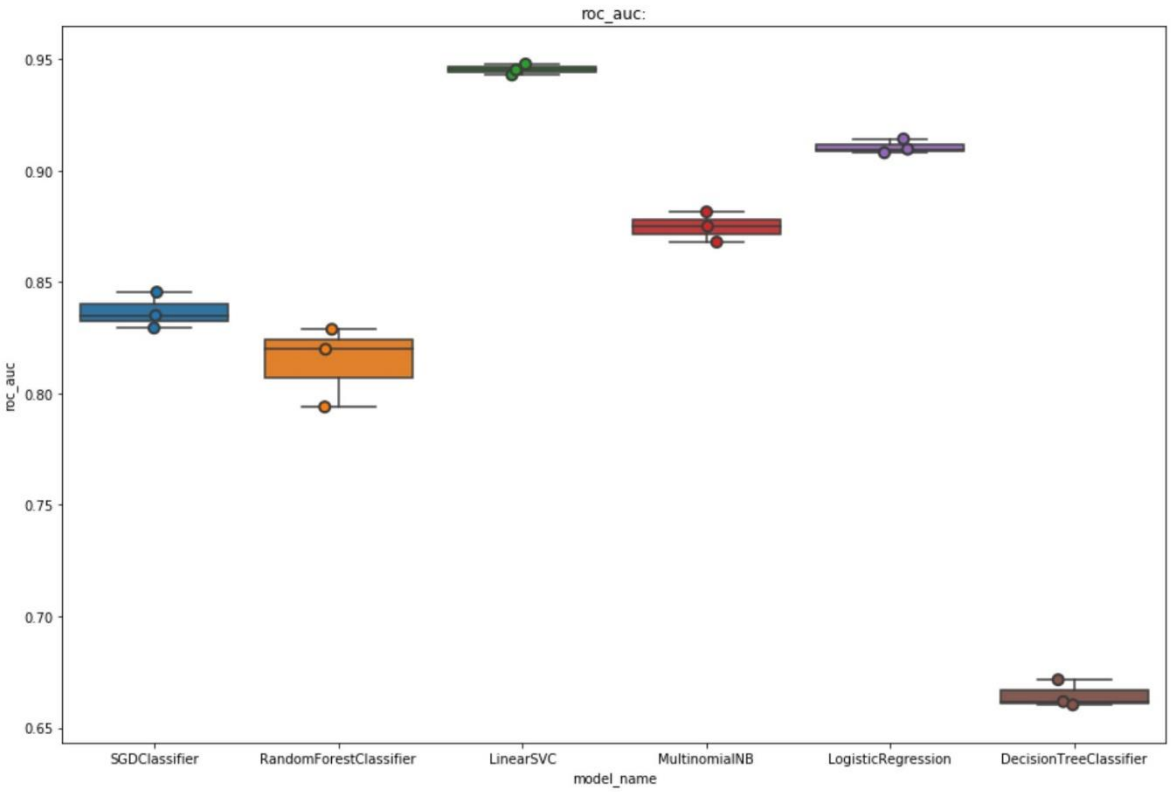


Figure 57 - Score d'AUC pour chaque algorithme sur 3-CV pour la classification du sexe sur la base #Elysée2017 à l'entrainement avec sur-échantillonnage de la classe minoritaire

Nous observons encore une fois qu'avec sur-échantillonnage les entrainements sont bien meilleurs autant en score AUC qu'en score en Accuracy.

Ensuite, j'ai appliqué chercher à optimiser les meilleurs modèles, en les optimisant plusieurs fois itérativement via GridSearch et validation croisée à 3 folds. Les résultats ont donc été améliorés, et un principal paramètre a été ajusté pour les algorithmes LinearSVC et la Régression Logistique :

- Régression Logistique : C = 15
- LinearSVC : C = 1.1

Enfin, j'ai appliqué les modèles, après entrainement sur l'ensemble de chaque jeu de données twitter, sur la base entendre la France pour valider le passage d'un type de base de données ('twitter') à un autre ('contribution'). Les résultats sont les suivants :

- 59% d'Accuracy avec LinearSVC
- 60% d'Accuracy avec Régression Logistique

Ces résultats, appliqués après sur-échantillonnage de la classe minoritaire sur la base 'Entendre la France' semblent assez faibles, et ne semblent pas vraiment pouvoir justifier le passage d'un type de données à un autre. Cela peut donc expliquer les incohérences dans les proportions de partis issues des classifications après application sur les contributions des modèles entrainés sur les données de type twitter.

Nous allons maintenant utiliser, en utilisant une méthode similaire, la base de données «Entendre la France » afin de classifier chaque contribution selon sa position par rapport au mouvement des gilets jaunes.

d. Application de modèles via les données 'Entendre la France'

L'avantage de l'utilisation de la base de données du débat organisé par Entendre la France contient à la fois des contributions qui ont sensiblement la même forme que les contributions du grand débat national, et qu'en plus chaque participant est caractérisé par des données personnelles qui sont intéressantes pour répondre à mes problématiques de représentativité des participants au grand débat. L'information que j'ai utilisée est la position des participants par rapport au mouvement des gilets jaunes.

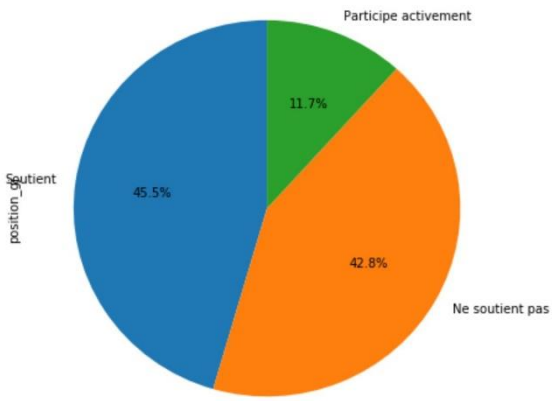


Figure 58 - Proportions de chaque position par rapport au mouvement des gilets jaunes dans la base 'Entendre la France'

Les proportions de chaque position par rapport au mouvement des gilets jaunes, c'est-à-dire « Soutient », « Ne soutient pas » et « Participe Activement » parmi les personnes ayant participé au débat organisé par « Entendre la France » sont rappelées ci-contre.

Etant donné que les différentes classes ne sont pas équilibrées, des entrainements sans et avec sur-échantillonnage des classes minoritaires ont été considérés. J'expose les résultats de ces entrainements ci-après.

i. Entrainement

Avant sur-échantillonnage :

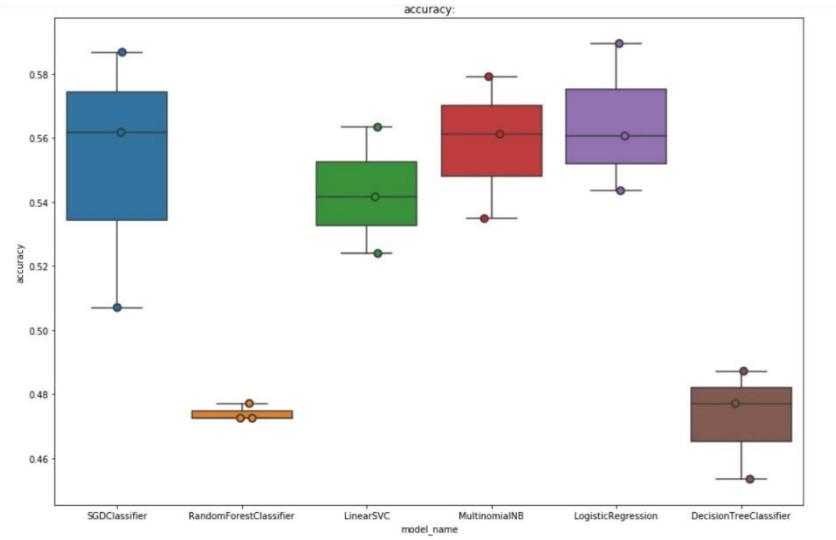


Figure 59 - - Score d'Accuracy en entrainement pour les différents modèles sur la classification de la position par rapport au moouvement des gilets jaunes, sans sur-échantillonnage

```
Accuracy:
model_name
DecisionTreeClassifier    0.472626
LinearSVC                 0.542962
LogisticRegression        0.564536
MultinomialNB             0.558371
RandomForestClassifier    0.474076
SGDClassifier             0.551841
```

Figure 60 - Scores d'accuracy moyens des différents algorithmes considérés pour la classification par rapport au mouvement des gilets jaunes, sans sur-échantillonnage

Après sur-échantillonnage :

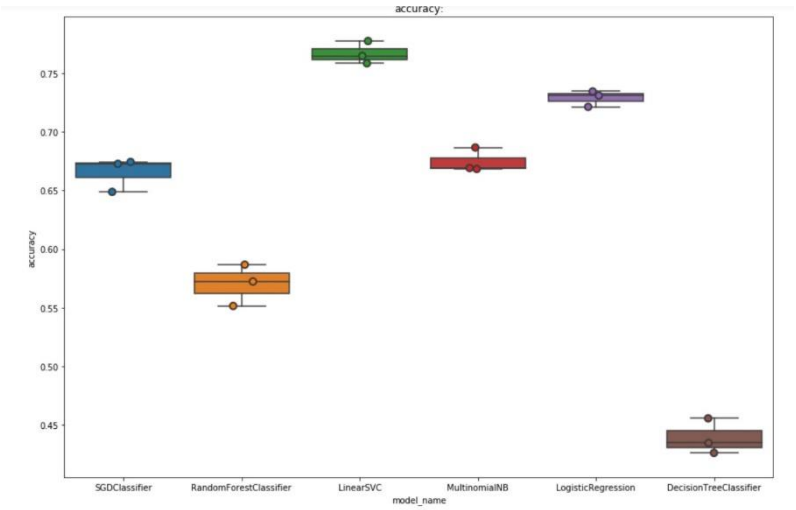


Figure 61 - Scores d'accuracy des différents algorithmes pour la classification sur la position par rapport au mouvement des gilets jaunes après sur-échantillonnage

```
Accuracy:
model_name
DecisionTreeClassifier    0.438777
LinearSVC                 0.766796
LogisticRegression        0.728819
MultinomialNB             0.674632
RandomForestClassifier    0.570112
SGDClassifier             0.665199
```

Figure 62 - Scores d'accuracy moyens pour les algorithmes après sur-échantillonnage

Nous observons là-aussi des résultats relativement satisfaisant compte tenu qu'un modèle aléatoire n'obtiendrait que 33% d'accuracy sur ce problème de classification à trois classes.

ii. Applications des modèles

Voici les résultats, en proportion, des classifications relatives à chaque position par rapport au mouvement des gilets jaunes, en utilisant, à l'instar des classifications précédentes, les meilleurs algorithmes à l'entrainement, qui sont toujours LinearSVC, la régression logistique et SGDClassifier, en utilisant les entrainements sur les données avec sur-échantillonnage des classes minoritaires :

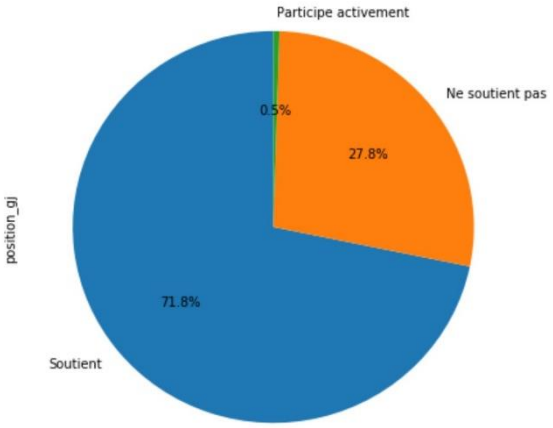


Figure 64 – Application après entrainement avec LinearSVC

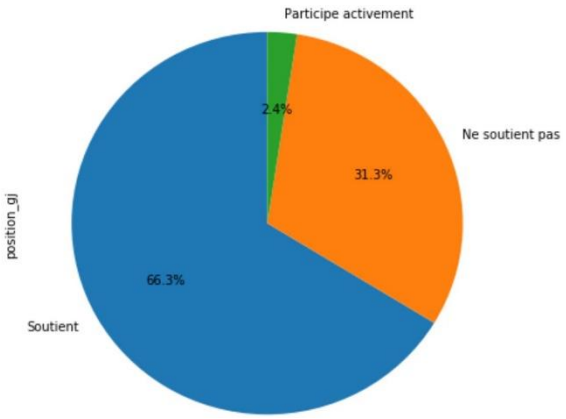


Figure 63 – Application après entrainement avec LogisticRegression

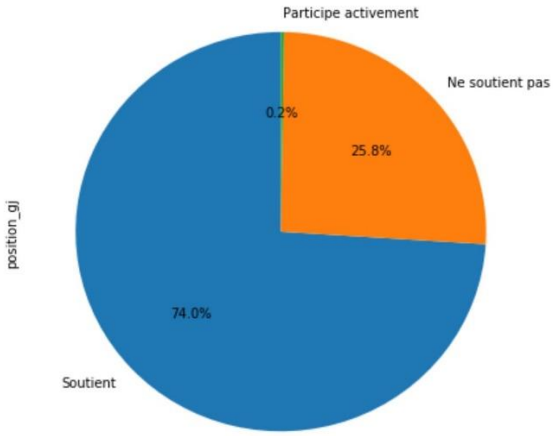


Figure 65 -Application après entrainement avec SGDClassifier

Ces résultats, en tout cas dans leurs proportions, ne sont pas, on le voit, très satisfaisant. La classe la plus classifiée est celle relativement au « Soutient » des gilets jaunes, avec autour de 70% pour chacun des algorithmes étudiés, contre autour de 25% pour « Ne soutient pas », et une proportion quasi nulle de « Participe activement ».

A l'instar de ce qui a été réalisé plus haut avec les classifications liées au parti politique, nous pouvons afficher les nuages de mots pour les 100 classifications les plus « certaines » pour chaque position. Des nuages de mots ont été développés pour chaque algorithme, mais nous représentons ici seulement ceux réalisés pour la classification avec Régression Logistique :

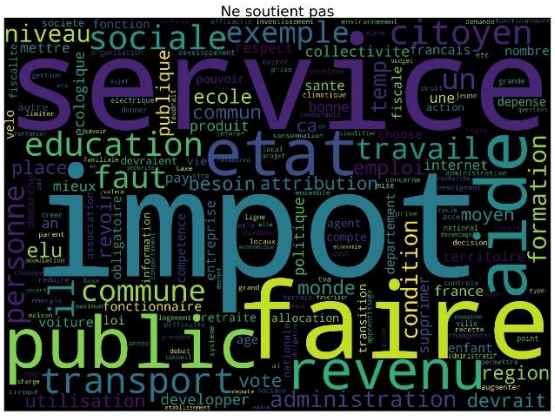


Figure 67 - Nuage de mots pour "Ne soutient pas" par rapport au mouvement des gilets jaunes

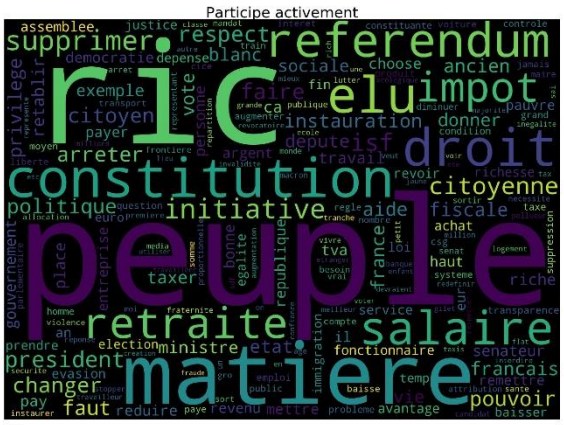


Figure 66 - Nuage de mots pour "Participe activement" par rapport au mouvement des gilets jaunes

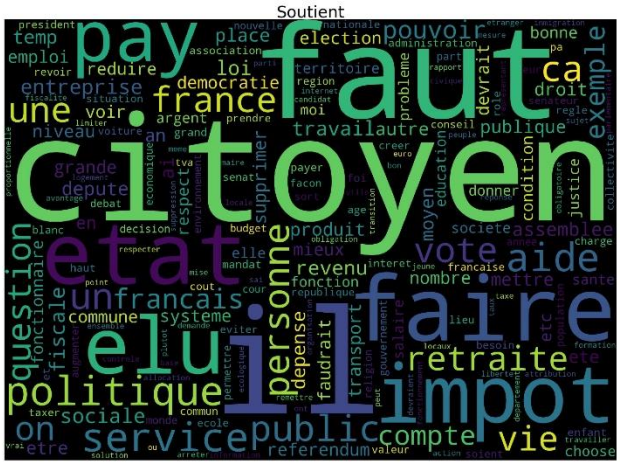


Figure 68 - Nuage de mots pour "Soutient" par rapport au mouvement des gilets jaunes

Certains résultats sont difficilement analysables, étant donné qu'il faudrait comprendre complétement le mouvement des gilets jaunes pour pouvoir être en mesure de commenter chaque mot présent dans ces représentations. Cependant, nous pouvons nous satisfaire que certains mots sont en parfaite cohérence avec ce que nous pouvions attendre, notamment les mots « peuple » mais surtout « ric » ou « referendum », « initiative » et « citoyenne » pour la classe associée à « participe activement », puisque la grande mesure réclamée par le mouvement des gilets jaunes était bien le RIC (« Référendum d'Initiative Citoyenne »).

Comme nous avons pu le noter dans la partie traitant des classifications de parti, il est ici aussi difficile de mesurer de façon certaine les performances de mes modèles, mais nous pouvons tout de même nous satisfaire de ces premiers résultats, notamment parce que les nuages de mots sont cohérents, et donc que même si certaines erreurs ont été faites sur une partie non négligeable des contributions, la caractérisation « intrinsèque » de chaque position par rapport au mouvement des gilets jaunes semble avoir été « comprise » par les modèles.

iii. Comparaison avec les classifications de partis politiques

Toujours dans le but d'étudier la réelle cohérence des résultats de mes modèles, les classifications de la position par rapport au mouvement des gilets jaunes peuvent être croisées avec les classifications par rapport aux partis politiques faites précédemment afin de savoir, pour chaque position par rapport au mouvement des gilets jaunes prédite, quel parti politique avait été attribué. Ces résultats sont présentés ci-dessous.

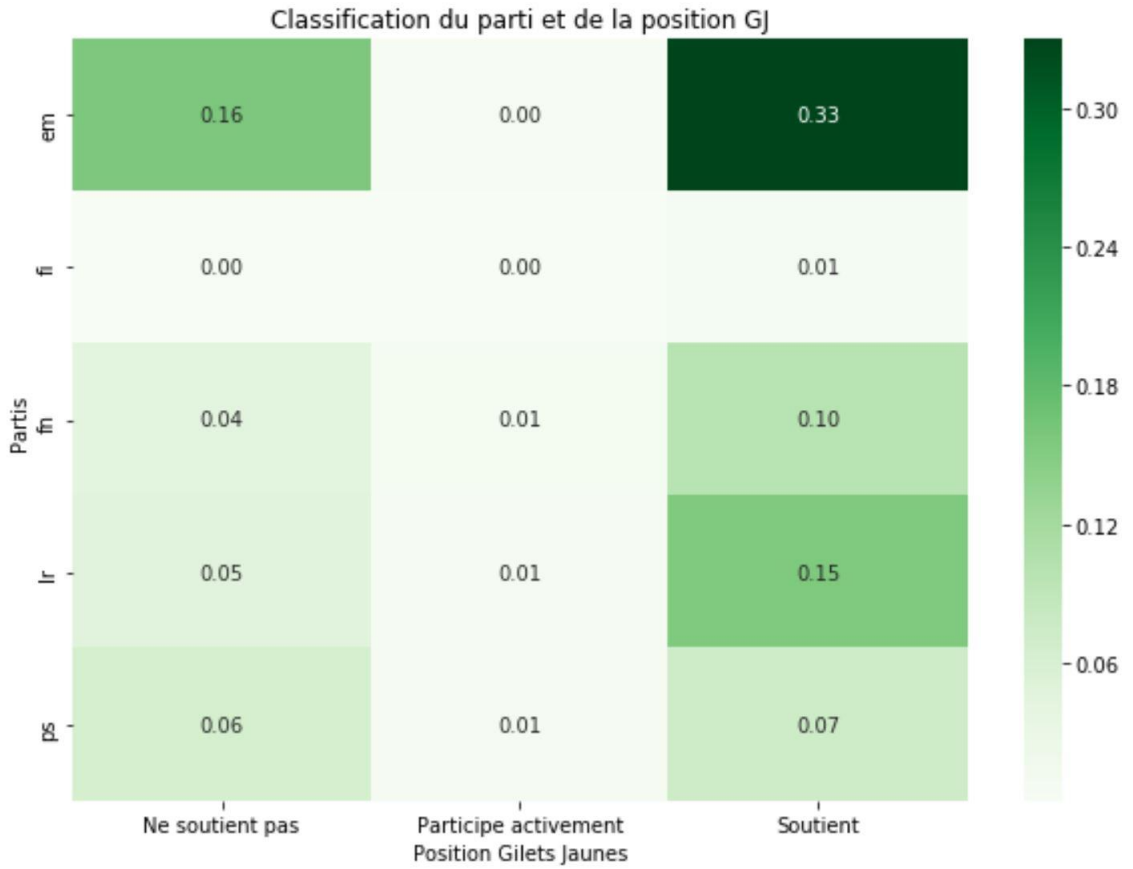


Figure 69 - Tableau croisé des classifications de de position par rapport au mouvement des gilets jaunes et de partis politiques sur la base #Elysée2017 avec la Régression Logistique

Ces résultats ont été envisagés pour étudier la cohérence des classifications. Cependant, on observe que ce qui a été obtenu n'est pas concluant. En effet, comme on pouvait s'en douter, les classes ayant été le plus attribué pour le parti et pour la position par rapport aux gilets jaunes sont donc les plus classifiées ensemble. Ainsi, la classification « La République en Marche » et « Soutient » par rapport aux gilets jaunes totalise un tiers des contributions alors que ce n'est pas du tout logique comparé à la réalité. La case « République en Marche » / « Ne soutient pas » est la deuxième en termes de nombre de contributions, et semble évidemment plus logique, mais on ne peut pas, par honnêteté intellectuelle, dire que les modèles sont pertinents simplement avec ce résultat.

e. Conclusion sur l'application des modèles de Machine Learning

Pour conclure sur ces classifications via applications de modèles de Traitement automatique du Langage Naturel et de Machine Learning, nous pouvons dire qu'en entrainement, même si les résultats auraient pu être encore améliorer via GridSearch par exemple, les résultats en accuracy étaient relativement bons.

En revanche, il est difficile, étant donné le peu d'annotations des données du grand débat national, d'établir de façon précises les performances réelles des modèles utilisés et produits, mais les nuages de mots affichés pour les 100 plus « certaines » classifications semblent cohérents par rapport aux caractéristiques et revendications habituelles des partis politiques français, et aussi des positions par rapport au mouvement des gilets jaunes.

Il aurait donc pu être intéressant d'utiliser pour un entrainement ultérieur les revendications et les contributions issues de ces classifications les plus certaines, afin notamment d'éviter le passage en entrainement et en application des modèles des données de type « twitter » et de type « contribution ».

IV. Travaux pouvant compléter ce projet

Compte tenu du temps et des moyens alloués à ce projet que j'ai réalisé seul, même si encadré, plusieurs autres tâches peuvent être réalisées afin d'améliorer l'étude de la pertinence des modèles et des résultats obtenus, ou bien en vue d'améliorer les modèles eux-mêmes. Je liste de façon non-exhaustive les tâches qui auraient ainsi pu être effectuées et qui ont pour certaines été sérieusement envisagées, avec des débuts d'expérimentation.

a. Interface graphique

Contrairement à ce qui avait été prévu, l'implémentation d'une interface graphique interactive n'a pas été réalisée. En revanche, les solutions techniques à mettre en œuvre ont été envisagées et quelques tests ont été réalisés. Je pense que le plus pertinent et plus « facile » à mettre en œuvre aurait été d'implémenter un dashboard Dash[37], et donc de « convertir » toutes les figures et tous les affichages actuels dans Plotly[38].

De plus, dans cette optique de réaliser une interface interactive pour présenter mon travail, des cartes de contributions ont été réalisées avec Folium[39], qui permet de mettre en place des cartes directement interactives. Pour cela, il a fallu convertir les codes postaux en coordonnées géographiques pour pouvoir les afficher sous forme de « markers » et de faire une « heatmap » des contributions. Un exemple de ce type de cartes réalisé lors de ce projet peut être retrouvée dans la partie I.b.vi. de ce rapport. On aurait aussi, je pense, pu essayer d'afficher une « heatmap », donc représentant les endroits dont le plus de contributions avaient été émises, mais en pondérant ces participations par rapport aux tailles de population de chaque ville, département ou région.

b. Word-Embeddings

Contrairement à ce que j'avais prévu, je n'ai malheureusement pas pu mettre en place une représentation des textes (tweets ou contributions) grâce aux word-embeddings. Les word-embeddings n'aurait peut-être pas été utiles lorsqu'il s'agissait d'étudier la bonne justification du passage d'un type de données à un autre comme ce qui a été réalisé en utilisant le sexe de chaque utilisateur comme donnée étiquetée, car, à l'instar de la lemmatisation, cela aurait pu « enlever » les informations relatives au genre des mots. En revanche, cela aurait été certainement utile lorsqu'il s'agissait de classifier le parti politique et la position par rapport au mouvement des gilets jaunes.

Les word-embeddings permettent de représenter chaque mot par un vecteur de nombres réels avec des distances correspondant aux « distances » entre les mots dans le langage courant. Par exemple, si des mots sont souvent employés ensemble, la distance entre les représentations vectorielles seront faibles. Alors, et par exemple, le mot « chien » sera plus proche du mot « chat » que ne le sera le mot « ordinateur ».

Plusieurs implémentations sont disponibles, et celle envisagée était fastText[40], implémentée par Facebook.

[37] https://dash.plot.ly/
[38] https://plot.ly/python/
[39] https://python-visualization.github.io/folium/
[40] https://fasttext.cc/

c. Affichage des données et PCA

Un essai d'Analyse en Composantes Principales (ACP ou PCA – Principal Components Analysis) a été réalisée, comme vous pouvez le voir ci-après, cependant les résultats n'ont vraiment pas été concluants. Cette ACP permet de représenter une matrice sous une dimension moins importante tout en gardant une représentation adéquate des données. L'objectif était de montrer que mon modèle permettait d'analyser et d'afficher les proximités entre partis (par exemple droite et gauche d'un point de vue politique).

Avec plus de temps et plus de moyens, les paramètres auraient pu être optimisés pour essayer soit d'afficher les classes selon des régions distinctes (dans la mesure du possible et selon les performances des classifieurs utilisés) ou bien d'utiliser cet Analyse en Composantes Principales pour représenter les données d'une manière différente et utiliser cette nouvelle représentation comme base d'apprentissage.

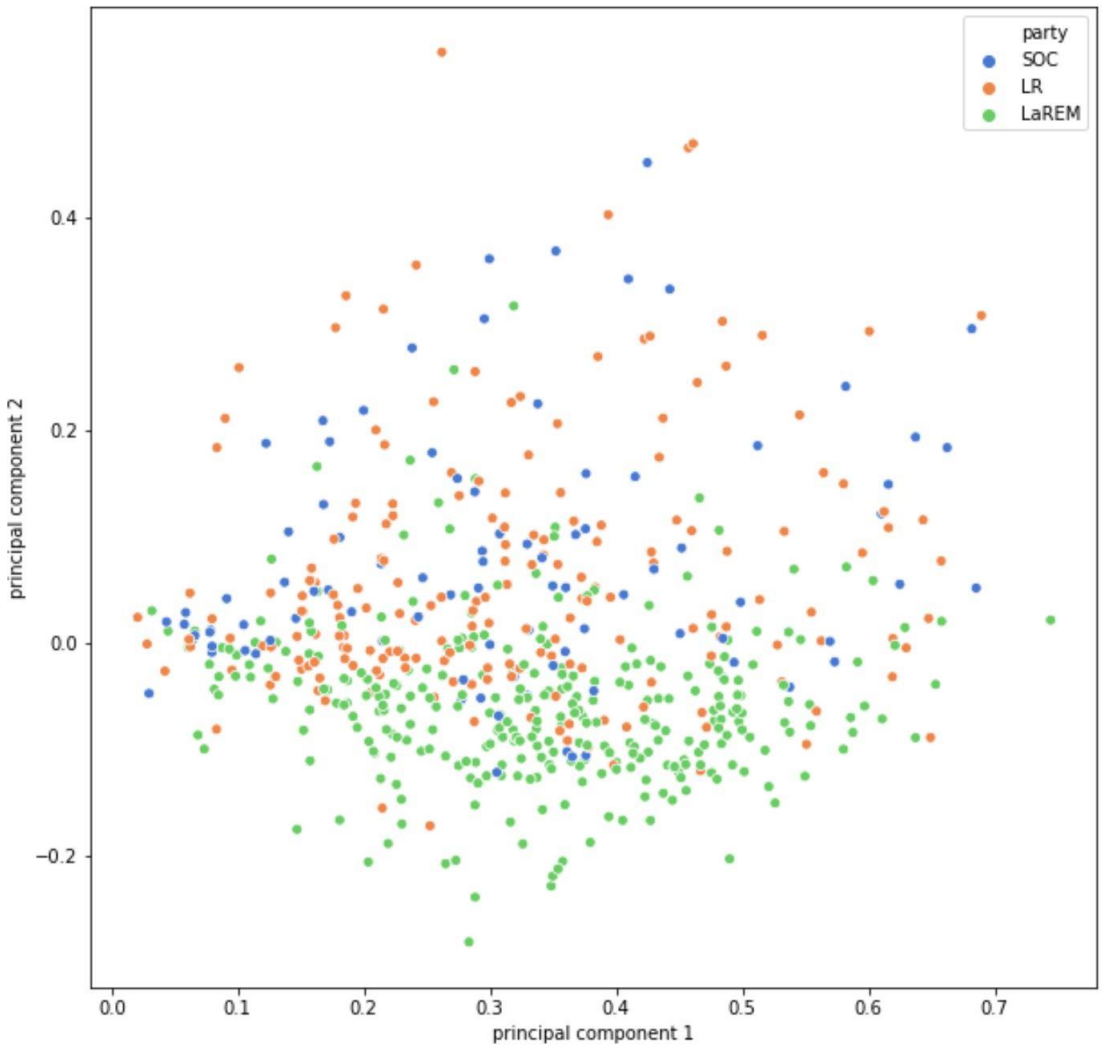


Figure 70 - Analyse en composantes principales sur la base de données liée aux parlementaires

Comme nous pouvons le voir sur l'essai d'analyse en composantes principales sur la base liée aux parlementaires, certains partis semblent se détacher, notamment La République en Marche (« LaREM ») par rapport aux deux autres, mais les partis sont encore trop liés entre eux pour établir de réelles conclusions et être tout à fait satisfait du résultat.

d. Mise en relation avec d'autres données de l'INSEE

Contrairement à ce qui était initialement prévu, je n'ai pas associé les données issues du grand débat national à des données INSEE issus du recensement de la population. Je voulais, au départ, mettre chaque contribution en relation avec des données relevant du niveau social, du niveau d'éducation, de la classe d'âge, du taux d'emploi…. de la ville et du code postal, et ainsi réaliser une étude similaire à celle effectuée pour les résultats du premier tour de l'élection présidentielle afin de confronter les valeurs obtenues aux moyennes nationales.

Cette initiative ne m'a pas semblé, finalement, très pertinente compte tenu du travail réalisé ici et aurait mobilisée trop de moyen par rapport à la plus-value concrète qu'elle aurait pu apporter.

Cependant, cette analyse pourrait compléter le travail entrepris, en la mettant en relation, à l'instar de ce que j'ai réalisé avec les opinions par rapport au mouvement des gilets jaunes, avec les classifications par rapport aux partis politiques et en les comparant à différentes études d'instituts de sondage, permettant justement de comparer les indicateurs économiques considérés avec les votes pour certains partis. On pourrait alors essayer d'étudier la cohérence de mes modèles sur tout ou partie des classifications mises en place.

e. Affichage d'une classe contre toutes les autres (« OnevsRest »)

Au lieu d'essayer d'afficher la proportion de chaque classification de parti de façon globale, j'aurais pu essayer de classifier chaque contribution selon qu'elle soit plutôt « pour » ou plutôt « contre » chaque parti, en essayant de la classifier de façon binaire, en posant en quelque sorte la question suivante : « Cette contribution est-elle représentative de ce parti ou des autres ? ».

En effet, d'une part, si un représentant politique est en général forcément pour un parti et pour un seul, un citoyen peut être représentatif de plusieurs partis, et, d'autre part, un citoyen peut tout à fait passer d'un parti à un autre.

De plus, la question ne se pose quasiment pas dans le cas de la base liée aux parlementaires, mais étant donné que la base des comptes twitter issue de l'article lié à « #Elysée2017 » a été annotée lors de l'élection présidentielle de 2017 et que les tweets utilisés dans mon travail sont les 100 les plus populaires de chaque compte, mais écrits depuis la création de leur compte, et donc aussi après 2017. Ainsi, pour les raisons évoquées plus haut et puisque quelqu'un peut tout à fait changer d'opinion politique, on peut se dire que certains comptes twitter ne sont actuellement plus forcément annotés de façon correcte.

f. Considérations plus techniques :

Plusieurs propositions un peu plus techniques que les précédentes peuvent être faites pour améliorer ou compléter le travail déjà réalisé.

D'une part, il aurait sans doute été plus pertinent d'afficher le score AUC-ROC plutôt que l'Accuracy, surtout pour les évaluations faites en présences de plusieurs classes. D'autre part, un meilleur pré-traitement aurait pu être mis en place, car comme nous l'avons vu avec les nuages de mots, certains mots toujours considérés ne devraient sans doute pas avoir une si grande importance. Enfin, une optimisation plus systématique par GridSearch aurait pu être mise en place, comme ce qui a pu être fait en partie III.c. lors de l'étude du passage d'un type de base de données à un autre. Cependant, cela demanderait plus de puissance et beaucoup de temps.

V. Conclusion

Tout d'abord, le travail prévu était réellement très ambitieux. Les données disponibles n'étaient pas nombreuses, et surtout les données ouvertes du Grand Débat National étaient assez peu annotées. Il a ainsi fallu aller chercher, obtenir et nettoyer d'autres bases de données ce qui a constitué une grande partie du travail, mais qui n'étaient en plus majoritairement pas du même type.

Nous avons pu observer que ma problématique ne pouvait pas réellement être résolue étant donné les moyens et le temps mis à ma disposition, notamment parce que même si un parti politique ou une position par rapport au mouvement des gilets jaunes ont pu être attribué à chaque contribution, aucun moyen sérieux ou adéquat ne peut en l'état attester des bonnes performances de mes modèles. Cependant, si certains résultats étaient relativement incohérents, je pense notamment au tableau croisant les classifications de l'opinion politique et de celle vis-à-vis du mouvement des gilets jaunes, ou encore de certaines proportions de partis sur la base des parlementaires, certains résultats sont très intéressants, surtout d'un point de vue politique.

Par exemple, les matrices de confusion pour certains algorithmes, retrouvant assez bien la proximité entre certains partis, qui pousserait, dans un futur travail, à essayer de représenter les différents individus via une Analyse en Composantes Principales (PCA) en deux dimensions, et surtout avec les nuages de mots exposants les mots les plus représentatifs pour les 100 classifications de chaque parti dont les algorithmes étaient les plus « sûrs », puisqu'on retrouve assez bien les termes représentants chaque parti, comme par exemple « revenu » et « universel » dans la classification du parti socialiste alors que la proposition phare du candidat à l'élection présidentielle 2017 Benoit Hamon était bien celle du revenu universel, ou encore le terme « immigration » apparaissant de façon très marquée pour le front national, ce qui est, pour des raisons évidentes, cohérent.

L'aspect qui n'a pas pu être réalisé ici mais qui résoudrait la plupart des soucis serait une annotation à la main d'au moins quelques contributions, celles-ci pouvant alors servir de base d'apprentissage. D'ailleurs, avec plus de temps, il aurait aussi été intéressant de prendre pour base d'apprentissage les mêmes contributions que celles servant à réaliser les nuages de mots, ceux-ci semblant relativement cohérents.

Enfin, il serait tout de même intéressant d'établir plus d'outils afin d'évaluer la représentativité politique et sociétale des contributeurs, ainsi que la pertinence réelle des analyses faites sur les contributions d'un point de vue démocratique, car comme expliqué en introduction, si cet exercice de grande consultation nationale est très intéressante et dans l'air du temps avec le développement de nouvelles techniques et technologies d'analyses, elle pourrait être très dangereuse si un pouvoir gouvernemental s'en emparait comme justification de sa politique alors que, possiblement, ni les participants ne représenteraient la population française, ni les analyses faites ne seraient conformes à ce qui est proposé par le peuple français.

VI. Annexes

a. Base technique :

Ce projet a été réalisé sur ma machine personnelle sous windows 10 Professionnel, ayant les caractéristiques suivantes :

- Mémoire installée (RAM) : 16 GO
- Processeur : Intel® Core™ i7-10510U CPU @ 1.80GHz 2.30GHz
- Type du système : système d'exploitation 64 bits, processeur x64

Etant donné ces spécifications, le coût computationnel important de certaines opérations s'est fait ressentir.

Ce projet peut être trouvé sur mon GitHub : https://github.com/adrihans/Projet_Grand_Debat

Les notebooks utilisés sont les suivants :

- Cleaning.ipynb

Notebook utilisé pour nettoyer les données.

- Concatenantion.ipynb

Notebook utilisé pour croiser les données et les concaténer, comme les données de code postal des contributions avec les données de résultats aux élections par exemple.

- Statistiques_tailles_contributions.ipynb

Notebook utilisé pour produire et afficher les analyses descriptives des données de contribution du grand débat national.

- Description_Analyse_Entendre_la_France.ipynb

Notebook utilisé pour afficher les analyses descriptives de la base de données « Entendre la France ».

- General_NLP.ipynb

Notebook utilisé pour appliquer les modèles de NLP et de Machine Learning afin de classifier les contributions selon le parti politique de leurs auteurs, en entrainant les modèles sur les bases de données twitter.

- NLP_Entendre_La_France.ipynb

Notebook utilisé afin d'appliquer les modèles de NLP et de Machine Learning afin de classifier les contributions selon la position par rapport au mouvement des gilets jaunes de leurs auteurs en entrainant les modèles sur la base de données « Entendre la France ».

- NLP_Sexe_Elysee.ipynb

Notebook utilisé afin d'appliquer les modèles de NLP et Machine Learning afin de vérifier et d'étudier le passage de données de type « twitter » avec la base de données #Elysée2017, au type « contribution » avec la base de données « Entendre la France ».

- Affichage_Résultats_GJ_partis.ipynb

Notebook utilisé afin de produire et d'afficher les tableaux croisés des classifications partis/Gilets jaunes des contributions du grand débat national.

Les fichiers .py utilisés sont les suivant :

- Affichage_stats.py

Fichier Python contenant les fonctions permettant d'afficher les analyses descriptives.

- NLP_plots_models.py

Fichier python contenant les fonctions permettant de réaliser les applications de modèles de Machine Learning et d'en afficher les résultats.

- Scraping.py

Fichier python permettant de réaliser le scraping des données twitter de façon simple.

- Multithread_scrap.py

Fichier python permettant de réaliser le scraping des données de la base #Elysée2017 en multithread.

- Multithread_scrap_deputes.py

Fichier python permettant de réaliser le scraping des données de la base twitter relative aux députés en multithread.

- Multithread_scarp_senateurs.py

Fichier python permettant de réaliser le scraping des données de la base twitter relative aux sénateurs en multithread.

Fichiers dont l'utilisation et le développement ont été abandonnés :

- Dashboard.py

Notebook d'essai de réalisation d'un dashboard.

- Word_Embedding.ipynb

Notebook d'essai d'utilisation de word embeddings

- Topic_Modelling_and_LDA.ipnyb

Notebook d'essai de réalisation de Topic Modelling et LDA.

- Tweets_elysee2017fr_NLP_sentiment_analysis.ipynb

Notebook réalisé pour essayer d'étudier des analyses de sentiments sur la base #Elysée2017, l'idée de base, abandonnée, était de prendre en compte les questions binaires du grand débat national, qui ne le sont dans les faits pas actuellement. On aurait pu, alors, essayer de réaliser une analyse de sentiments sur les phrases ou groupe de mots contenant les thèmes abordés dans ces questions à choix multiples (ou réduits)

VII. Bibliographie :

[1] Fraisier Ophélie, Cabanac Guillaume, Pitarch Yoann, Besançon Romaric, Boughanem Mohand. 2018. #Élysée2017fr: the French Presidential Election on Twitter. In International Conference on Weblogs and Social Media.

[2] Simone Paolo Ponzetto, Heiner Stuckenschmidt, 2019, Political Text Scaling Meets computational Semantics, https://arxiv.org/pdf/1904.06217.pdf

[3] Dominique Estival Tanja Gaustad, Son Bao Pham, Will Radford, Ben Hutchinson, Author Profiling for English Emails, 2007, http://dominique-estival.net/PACLING07Final.pdf

[4] Helena Gomez-Adorno, Improving Feature Representation Based on a Neural Network for Author Profiling in Social Media Texts, 2016, https://www.hindawi.com/journals/cin/2016/1638936/

[5] Goran Glavas, Federico Nanni, Simone Paolo Ponzetto, Unsupervised Cross-Lingual Scaling of Political Texts, 2017, https://www.aclweb.org/anthology/E17-2109.pdf