Classification du parti politique sur les discours du Parlement Européen

Xinhao Zhang **Yingzi Liu** **Xiaohua Cui**
Université Sorbonne Nouvelle

Abstract

Ce projet a implémenté une tâche de classsification de texte par parti politique des discours du Parlement européen en français, en appliquant différents classificateurs. L'évaluation des modèles a permis d'identifier le meilleur classificateur et ses paramètres pour cette tâche. Les expériences ont montré que le classificateur A était le plus performant pour les textes français, surpassant les résultats mentionnées dans les actes de 2009. L'archive comprenant les données et les codes sont disponibles sur le site[1] github, avec des branches correspondantes de chaque classifieur.

1 Introduction

Notre étude s'inscrit dans la cinquième édition du défi fouille de texte (DEFT), orientée vers la fouille d'opinion. Au sein de cette édition, trois tâches ont été proposées, s'appuyant sur deux corpus multilingues (français, anglais et italien). La troisième tâche, qui constitue l'axe principal de notre recherche, est à déterminer l'affiliation politique des parlementaires à partir de leurs interventions. Notre focus se porte sur cette tâche en exploitant spécifiquement les interventions en français issues du corpus de débats parlementaires européens.

Pour cette tâche précise, notre exploration commence par les classificateurs linéaires tels que la régression logistique et le NaïveBayes. Aussi, nous avons essayé les méthodes ensemblistes : y compris le Random Forest et des algorithmes de Boosting tels que LightGBM.

Basés sur les résultats obtenus par ces différents classificateurs, ainsi que la discussion sur leur pertinence et efficacité dans le contexte spécifique des débats parlementaires européens, nous examinerons également l'impact de diverses techniques pendant tous les étapes, telles que le rééquilibrage des classes, la sélection des caractéristiques, et l'ajustement des paramètres des modèles sur la performance de la classification.

[1] https://github.com/XINHAO-ZHANG/5eme_DEFT_Fouille_Opinion

2 Traitement du corpus

Pour mener à bien notre tâche de classification, nous avons besoin de chaque texte d'orateur et son parti. Par conséquent, pour les fichiers XML d'origine, nous avons d'abord extrait les balises `texte` sous chaque balise `doc` dans l'ensemble d'entraînement, ainsi que leur catégorie politique correspondante. Nous avons ensuite transformé ces données en objets `DataFrame` en utilisant une liste de dictionnaires, puis les avons exportées au format CSV. Cela a contribué à rendre la structure plus claire et à faciliter les étapes suivantes de prétraitement du texte. Pour l'ensemble de test, nous avons suivi un processus similaire en combinant le fichier XML avec le fichier texte pour obtenir le CSV de même structure.

Dans la phase de prétraitement du texte, nous avons utilisé NLTK pour la normalisation de la casse et pour supprimer les mots vides, les caractères spéciaux et les chiffres, car ces éléments n'apportent aucune valeur à la tâche de classification. Ensuite, nous avons encapsulé le texte traité dans un nouveau fichier CSV.

Dans la phase de sélection des caractéristiques, nous avons utilisé TF-IDF pour la vectorisation du texte. Lors de la configuration des paramètres de cette fonctionnalité, nous avons choisi max_df=0.6 pour filtrer davantage les mots qui apparaissent fréquemment.

3 Méthodologies et évaluation

Dans notre étude, nous avons adopté une approche méthodologique diversifiée, en exploitant plusieurs modèles de classification afin d'assurer une analyse à la fois précise et robuste. Cette section détaillera chacun des modèles que nous avons utilisés, conformément à ce qui a été présenté dans

l'introduction. Nous mettrons l'accent sur les spécificités de chaque modèle, évaluerons leurs performances et résultats, et expliquerons comment ils ont été appliqués spécifiquement dans le cadre de notre recherche.

3.1 Classificateurs linéaires

Nous avons considéré l'approche des classificateurs linéaires, qui sont souvent utilisés comme point de départ pour la classification de textes en raison de leur simplicité et de leur efficacité dans de nombreux contextes. Ces modèles sont particulièrement utiles lorsqu'il s'agit de traites des ensembles de données de grande taille car ils tendent à être moins gourmands en ressources de calcul tout en fournissant des résultats robustes.

3.1.1 Régression Logistique

Nous avons premièrement exploré le modèle de la régression logistique. Ce modèle spécifique démontre une performance insatifaisante en l'absence du ré-équilibre préalable de distribution des données. Cette impuissance se manifeste par une macro F-mesure moyenne de 0,62. Néanmoins, il est à noter que l'efficacité du modèle s'améliore considérablement avec l'application de divers traitements des données, tels que le rééchantillonnage, où l'on observe une amélioration de la performance à hauteur de 77%. Malgré ces améliorations, les résultats globaux obtenus par la régression logistique demeurent relativement médiocres. C'est aussi pourquoi ce modèle constitura notre modèle baseline avec lequel nous comparerons nos prochains essais.

	P	**R**	**F1**	**Sup**
ELDR	0.94	0.63	0.75	1339
GUE-NGL	0.87	0.77	0.82	1792
PPE-DE	0.68	0.90	0.77	4571
PSE	0.78	0.72	0.75	3626
Verts-ALE	0.93	0.64	0.75	1585
Accuracy			0.7691	
Kappa-Score			0.6827	

Table 1: Résultats de Classification par Regression Logistique avec les données ré-équilibrées

3.1.2 Multinomial Naive Bayes

Le modèle Naïve Bayésien est relativement facile à implémenter et il offre une efficacité élevée en matière de classification. Pour cette tâche qui consiste à classer des textes en plusieurs catégories, avec des caractéristiques discrètes, nous avons choisi le modèle de classification MultinomialNB dans sklearn.

En ce qui concerne l'ajustement des paramètres, étant donné les quantités différentes des catégories, un taux d'apprentissage trop élevé peut entraîner des problèmes de probabilités nulles. Ainsi, nous avons fixé le taux d'apprentissage à `0,001` pour obtenir les résultats suivants :

	P	**R**	**F1**	**Sup**
ELDR	0.73	0.66	0.69	1399
GUE-NGL	0.81	0.73	0.77	1792
PPE-DE	0.74	0.77	0.76	4571
PSE	0.68	0.73	0.71	3626
Verts-Ale	0.73	0.67	0.70	1585
Accuracy			0.7318	
Kappa-Score			0.6398	

Table 2: Résultats de Classification par Multinomial NB avec alpha = 0.001

Les résultats sont moins satisfaisants par rapport au baseline. Aussi,nous observons qu'il n'y a pas de corrélation évidente entre la taille du support et la performance. Par exemple, la classe PPE-DE a le plus grand nombre d'échantillons (4571), mais n'a pas nécessairement la meilleure performance en termes de précision ou de score F1.

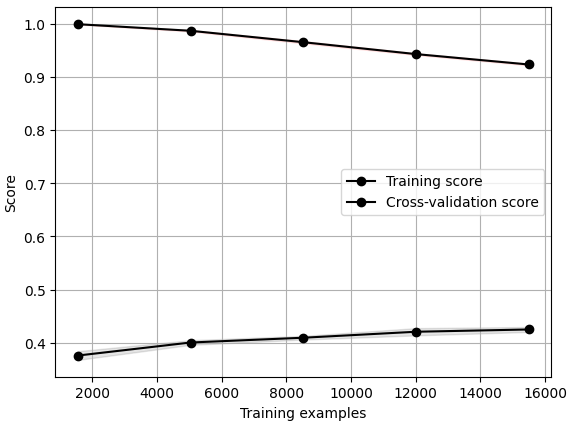

Figure 1: Learning courbe du modèle NB, présentant des signes de surajustement

[h]

De plus, nous avons constaté que, même après une nouvelle sélection de caractéristiques et un ajustement des paramètres, le modèle Naïve Bayes présente toujours des signes de surajustement. Cela pourrait être dû au fait que le modèle n'est pas assez complexe pour capturer toutes les relations com-

plexes dans les données. La simplicité du modèle limite sa capacité à atteindre un score plus élevé sur les données d'entraînement.

3.2 Méthodes ensemblistes

3.2.1 Random Forest

Dans le cadre de notre étude sur les problématiques de classification, nous avons initialement adopté l'arbre de décision comme modèle de base, attirés par sa simplicité et son intuitivité. Toutefois, lors des premiers tests, nous avons constaté que l'arbre de décision ne fournissait pas des résultats satisfaisants, en particulier en raison de sa tendance au surajustement face à des ensembles de données complexes. Cette observation nous a conduits à réévaluer notre approche et à chercher des alternatives plus performantes. Nous avons donc décidé de nous concentrer sur l'évaluation de la forêt aléatoire, un modèle composé de multiples arbres de décision. Cette méthode offre une solution efficace au problème du surajustement rencontré avec l'arbre de décision isolé, en améliorant à la fois la fiabilité et la généralisation du modèle.

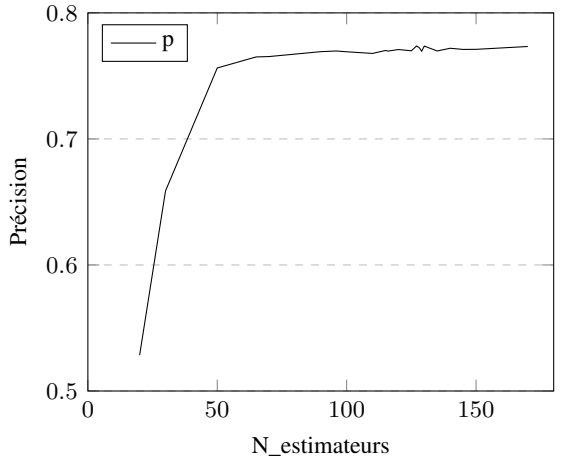

Figure 2: Exemple de performance(précision) du modèle avec le nombre d'estimateurs

Cependant, nous avons également rencontré certains problèmes lors de la construction de ce classificateur : tout d'abord, la configuration des paramètres de l'algorithme de forêt aléatoire est assez complexe, avec plusieurs paramètres à ajuster. Comme notre corpus est un vecteur de haute dimension et de grande taille, cela a entraîné une consommation de temps considérable dans le processus de résolution des hyperparamètres. Au cours de ce processus, bien que nous ayons ajusté le `random_state` pour stabiliser les résultats, l'exécution de la validation croisée et de la recherche en grille (Grid Search) pour trouver les valeurs optimales pour des paramètres tels que `n_estimators` et `max_depth` nécessite toujours beaucoup de temps et de puissance de calcul. En outre, en raison de la nature aléatoire, chaque fois que la forêt aléatoire est entraînée, même si les autres paramètres restent les mêmes, le modèle final peut être légèrement différent, ce qui conduit à une moindre interprétabilité.

Après l'ajustement des paramètres, les meilleures résultats de ce classificateur sont indiqués dans le tableau 2 :

	P	**R**	**F1**	**Sup**
ELDR	0.99	0.62	0.76	1399
GUE-NGL	0.95	0.74	0.83	1792
PPE-DE	0.67	0.94	0.78	4571
PSE	0.80	0.72	0.75	3626
Verts-Ale	1.00	0.61	0.76	1585
Accuracy			0.7760	
Kappa-Score			0.6896	

Table 3: Résultats de Classification par Random Forest

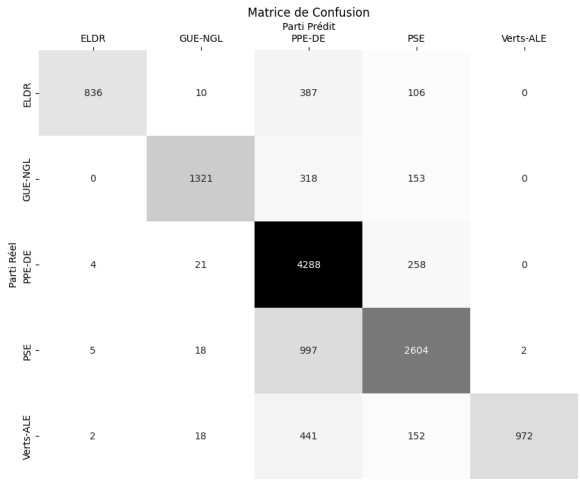

Figure 3: Matrice de confusion de la classification RandomForest

D'après nos résultats, le classifieur RandomForest a une précision globale de plus de 77, ce qui est un chiffre relativement idéal par rapport au baseline. Les taux de précision et de rappel varient selon les catégories. Par exemple, la classe ELDR a un taux de précision élevé (0.99) mais un taux de rappel plus bas (0.62). Pour la catégorie PPE-DE, qui a un plus grand nombre d'échantillons, le taux de rappel est élevé (0.94) mais le taux de précision

est plus bas (0.67), ce qui indique que le classificateur a tendance à classer incorrectement les échantillons dans les catégories plus importantes. Par conséquent, nous avons tenté d'utiliser le SMOTE pour rééchantillonner, mais cela n'a pas mené à une amélioration significative.

Class	Original	Resampled
0	2008	6858
1	2687	6858
2	6858	6858
3	5436	6858
4	2376	6858

Table 4: Classe Distributions dont les labels 0-5 correspondant aux 5 partis

3.2.2 Le gradient Boosting Decision Tree - LightGBM (Light Gradient Boosting Machine)

La méthode du Gradient Boosting Decision Tree, introduite par Friedman en 1999 et 2002, représente le boosting comme un processus de descente de gradient dans un espace fonctionnel, orienté vers un problème d'optimisation dont l'objectif est de minimiser l'erreur attendue. Cette technique construit progressivement un modèle prédictif à travers des itérations successives, formant une suite d'arbres de décision initialement simples qui sont ensuite continuellement affinés et améliorés à chaque étape. Nous avons initialement envisagé l'utilisation de XGBoost (eXtreme Gradient Boosting) en raison de ses performances reconnues dans le développement de modèles prédictifs. Cependant, nous avons rencontré des difficultés significatives avec XGBoost, principalement dues à la grande taille de notre jeu de données et à la complexité du modèle, qui entraînaient une consommation excessive de mémoire et provoquaient le crash du noyau de calcul. Malgré plusieurs tentatives d'optimisation, l'utilisation de XGBoost s'est avérée infructueuse dans notre contexte.

En conséquence, nous avons orienté notre choix vers LightGBM (Light Gradient Boosting Machine), un autre framework de Gradient Boosting. LightGBM est particulièrement adapté pour gérer de grandes quantités de données tout en minimisant la consommation de mémoire. Grâce à ses techniques avancées comme les algorithmes d'arbres de décision basés sur des histogrammes et l'échantillonnage unilatéral basé sur le gradient, LightGBM offre une combinaison efficace de rapidité et d'efficacité. Nous avons choisi d'utiliser les paramètres par défaut de LightGBM pour notre modèle, afin de maintenir un équilibre optimal entre efficacité et performance, évitant ainsi les problèmes rencontrés avec XGBoost. Les résultats obtenus avec LightGBM sont présentés ci-dessous:

	P	R	F1	Sup
ELDR	0.89	0.51	0.65	1399
GUE-NGL	0.83	0.72	0.77	1792
PPE-DE	0.67	0.87	0.75	4571
PSE	0.69	0.69	0.69	3626
Verts-Ale	0.83	0.55	0.66	1585
Accuracy			0.7203	
Kappa-Score			0.6147	

Table 5: Résultats de Classification par LightGBM

On peut voir dans les résultats que le modèle affiche une performance globale satisfaisante avec une précision de 0.7203 et un score Kappa de 0.6147, indiquant une amélioration significative par rapport à une prédiction aléatoire, bien qu'il reste un potentiel d'amélioration en termes de cohérence ; pour les classifications individuelles, le PPE-DE se distingue par un taux de rappel élevé mais une précision moindre, tandis que GUE-NGL montre un équilibre entre précision et rappel, et ELDR ainsi que Verts-Ale présentent une haute précision mais un faible taux de rappel, révélant ainsi des opportunités d'optimisation, en particulier dans l'équilibrage de la précision et du rappel pour certaines catégories.

4 Conclusion

Notre étude a révélé des perspectives intéressantes dans la classification des affiliations politiques à partir de discours parlementaires en français. Nous avons examiné divers classificateurs, y compris des méthodes linéaires et ensemblistes, et évalué leurs performances dans le cadre de cette tâche spécifique.

La Régression Logistique, bien qu'elle serve de baseline, a montré des limites en termes de précision et de rappel, malgré une amélioration notable avec le rééquilibrage des données. Le modèle Multinomial Naive Bayes, quant à lui, s'est avéré simple à implémenter et performant, mais a rencontré des problèmes de surajustement, suggérant une complexité insuffisante pour notre jeu de données.

LightGBM, bien qu'il offre une amélioration

notable par rapport au baseline, a révélé des opportunités d'optimisation, en particulier dans l'équilibrage de la précision et du rappel pour certaines catégories. Sa performance globale satisfaisante, combinée à une consommation de mémoire moindre, en fait une alternative viable, en particulier pour les grands ensembles de données.

Les méthodes ensemblistes, représentées par Random Forest et LightGBM, ont offert des résultats plus prometteurs. Le Random Forest, malgré sa complexité et ses exigences en termes de puissance de calcul, a démontré une précision globale supérieure avec un meilleur équilibre entre précision et rappel par catégorie. Cependant, il a également montré une tendance à classer incorrectement dans des catégories plus grandes, malgré les tentatives de rééquilibrage des données.

Malgré certains succès, il existe des marges d'amélioration significatives. La sélection des caractéristiques, notamment par le biais du TF-IDF, pourrait bénéficier d'une réévaluation pour améliorer la précision des modèles.

D'autre part, l'adoption d'une stratégie d'ensemble basée sur un sous-ensemble de l'échantillon plutôt que sur l'ensemble complet du jeu de données pourrait augmenter l'efficacité du calcul sans compromettre significativement la performance.

De plus, un ajustement plus raffiné des paramètres des modèles, notamment à travers la validation croisée et la recherche en grille, est essentiel pour optimiser les performances. En résumé, bien que nos résultats offrent des perspectives intéressantes pour l'analyse de texte politique, ils soulignent également la nécessité d'une exploration plus approfondie des techniques de traitement des données et d'amélioration des algorithmes de classification.

En somme, bien que les résultats actuels soient prometteurs, l'analyse de texte politique reste un domaine vaste et complexe, où l'innovation continue dans les méthodes de traitement des données et l'amélioration des algorithmes est cruciale.