Rapport SD701 — Projet Élections

Jia LIAO, Philéas SAMIR, Gwladys SANCHEZ, Pooran SHAHDI
MS BGD 21-22
Télécom Paris

***Résumé*—Nous avons travaillé sur les données ouvertes relatives aux élections entre 2012 et 2021. Nous avons nettoyé et exploré ces données, clusterisé des entités géographico-politiques par proximité électorale et tenté d'entraîner des modèles supervisés de classification à l'aide de ces données. Nos modèles pourraient être utilisés afin de faire des prédictions quant aux élections présidentielles à venir en mai 2022.**

I. Introduction

Nous avons récupéré les données ouvertes de tous les premiers tours scrutins de 2012 à 2021 sur data.gouv.fr. Les élections qui ont été traitées sont les suivantes :

- Présidentielles : 2012 et 2017,
- Législatives : 2012 et 2017,
- Européennes : 2014 et 2019,
- Municipales : 2014 et 2020,
- Départementales : 2015 et 2021,
- Régionales : 2015 et 2021.

Nous avons nettoyé et pré-traité ces données afin d'entraîner différents modèles. Nous avons agrégé les données par département pour chaque élection. Nous avons agrégé les votes blancs, nuls, et les abstentions, et nous préciserons dans quels cas nous prenons cette donnée en compte.

D'une part, nous avons utilisé des modèles de clustering afin d'identifier des départements qui ont des résultats électoraux similaires. Ces clusterings nous ont également permis d'effectuer des analyses sur l'influence du temps, ou du mode et des enjeux du scrutin, sur les comportements électoraux.

D'autre part, nous avons entraîné des modèles de classification qui nous permettent de prédire la nuance politique qui se retrouve en première place au premier tour d'une élection. Les données utilisées pour ces modèles de classification ont été enrichies par des données démographiques diverses.

Le code est accessible sur GitHub.

II. Collection et nettoyage

L'objectif du pré-traitement a été le suivant : représenter les données de chaque élection dans un format similaire afin de pouvoir entraîner nos différents modèles de la même façon sur chaque élection. Nous avons choisi de garder pour chaque élection les résultats agrégés par département, par nuance politique, en comptabilisant (mais en agrégeant) les abstentions et les votes blancs et nuls.

Nous avons choisi de ne garder que les résultats du premier tour de chaque élection, afin de simplifier le traitement. La plupart des élections a ses "codes nuances", qui permettent de regrouper ensemble des listes de la même obédience (parti, mouvement, divers, etc). Nous avons regroupé ensemble les codes nuances qui nous paraissaient similaires afin de ne garder que six catégories : extrême droite (ED), droite (D), centre (C), gauche (G), extrême gauche (EG), autres (A) ; voir Table I pour la répartition des différents codes nuances dans ces grandes catégories. Notre labélisation est discutable et liée à nos représentations ; le positionnement des partis, mouvements et personnages politiques sur le spectre gauche-droite est un champ d'étude à part entière.

Nuance	Partis/Codes
EG	COM, EXG, FG, LO, NPA, EG
G	SOC, DVG, PG, UG, RDG, VEC, LFI, FI, RDG, UGE, ECO, GJ, G
C	MMD, CMD, MDM, UC, UDI, LREM, UCD, UCG, REM, DVC, CEN, NCE, ALLI, C
D	DVD, UD, UMP, LR, DSV, PRV, D
ED	EXD, FN, RN, DLF, UXD, ED
A	REG, AUT, DIV, A

Table I – Correspondance entre toutes les nuances présentes dans les données et les nuances choisies, notre interprétation. Les code-nuances (i.e. les acronymes présents dans la table) sont attribués par le ministère de l'Intérieur pour certaines élection, voir par exemple [1].

Les données se présentent dans des fichiers de différents formats (csv, txt, xlsx). Il y a des similarités dans la façon dont les données des élections sont présentées d'une élection à la suivante, mais il y a aussi des différences, qui peuvent rendre le pré-traitement difficile. Chaque fichier a nécessité une préparation spécifique.

Deux élections municipales ont eu lieu entre 2012 et 2020 : l'une en 2014, et l'autre en 2020, au début de la pandémie. Les fichiers sont séparés entre les communes de moins de 1000 habitants, et les communes de plus de 1000 habitants. Cette division n'est pas arbitraire : le mode de scrutin change en fonction de la population. En effet, dans les communes de moins de 1000 habitants, on peut se présenter individuellement ou présenter une liste (dans les communes de plus de 1000 habitants, seules les listes sont autorisées), et les électeurs peuvent rayer des noms, voir [5]. Les suffrages sont comptabilisés individuellement. Pour les communes de moins de 1000 habitants, la majorité des listes sont aussi sans étiquette. Ces deux facteurs combinés nous ont poussé à écarter ces données : elles apportent peu d'informations sur les opinions des citoyens, et présentent un challenge particulier pour le comptage des voix (chaque citoyen vote individuellement pour plusieurs candidats, raye des noms, ce qui rend impossible la comptabilisation par liste).

De même, deux élections départementales se sont tenues dans la période visée par notre étude, la première en 2015, la seconde en 2020. Dans les deux cas, les résultats sont présentés de manière similaire, avec une ligne par commune (les données étant déjà agrégées par bureau de vote), et sans la difficulté précédente relative aux listes de candidats. En effet, lors des élections départementales, les citoyens de chaque ville doivent se prononcer sur le choix d'un binôme de conseillers départementaux pour le canton, et chaque ligne contient donc les informations relatives aux panneaux électoraux du canton, notamment le nombre de votes décomptés pour chaque binôme. Bien que ces binômes ne soient pas uniques au sein du département, mais différents entre chaque canton, leur nuance politique est explicitée sur le tableau des résultats, et les binômes sont indivisibles, ce qui en facilite le traitement.

Les élections européennes sont organisées autour de listes, et cette organisation se retrouve dans les fichiers. Les données n'étaient pas classées par Code nuance comme c'est habituellement le cas mais par nom de liste. Cette classification spéciale a requis que nous nous mettions dans le rôle des agents du Ministère de l'Intérieur chargés de classifier les listes. Nous présentons cette classification dans II.

Enfin, les élections présidentielles sont organisées sous la cinquième République autour de candidats individuels, et non pas autour de partis. Ce sont ces candidats qui organisent leurs campagnes, récoltent leurs signatures, etc. Le Ministère de l'Intérieur ne classifie pas ces candidats sur le spectre gauche-droite. Nous avons donc dû classifier nous-mêmes les candidats : voir Table III et IV.

Dans tous les cas, l'approche choisie pour obtenir des données uniformes a consisté à repasser en colonne les données réparties sur une même ligne, en obtenant alors pour chaque ville, autant de lignes que de scrutins. Ensuite, nous avons procédé à la "traduction" des nuances politiques d'après nos critères simplifiés de classification, définis ci-dessus, puis avons agrégé les données par département et type de parti, en sommant les nombres de voix par nuance. Enfin, nous avons repassé ces informations en ligne, de sorte à obtenir un tableau donnant pour chaque département (en ligne), les scores pour nos six courants politiques (en colonne).

Nous avons également ajouté la somme par département des abstentions, votes blancs, et nuls comme une seule valeur permettant d'illustrer la proportion de votes "non réalisés" d'une certaine manière. Le raisonnement a consisté à conserver certaines données "au cas où", en gardant à l'esprit qu'il sera plus facile de les retirer de nos modèles que de les y ajouter *a posteriori*.

Enfin, pour garantir l'uniformité de type des noms de départements, nous avons modifié tous les codes au format lettre relatifs aux départements et régions d'outre-mer (ZA, ...), afin de les remplacer par leur code de département à trois chiffres (971, ...).

III. Clustering

Nous avons entraîné un modèle de KMeans pour chaque élection, afin de regrouper (clusteriser) des départements aux sensibilités politiques similaires, et de voir si ces groupes sont résilients aux différents modes de scrutin. Nous avons systématiquement pris en compte l'abstention et les votes blancs et nuls dans nos axes de clustering.

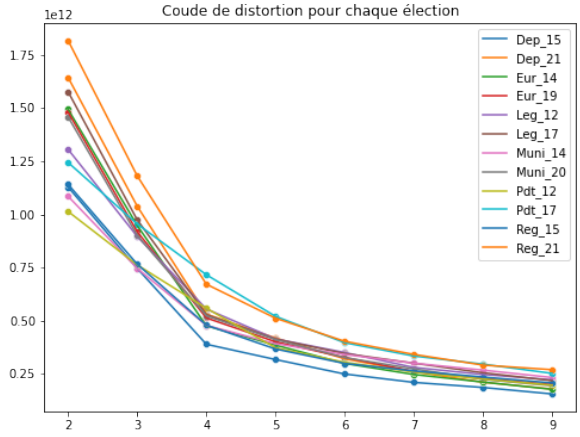


FIGURE 1 – Coude de distortion pour chaque élection. Le tracé motive le choix de $k = 4$.

Afin de décider du nombre de clusters, nous avons tracé des coudes, voir Figure 1. Nous avons choisi de garder le même nombre de clusters pour toutes les élections afin de pouvoir mesurer la résilience de ces clusters, comme expliqué plus haut. Le nombre de clusters a été fixé à $k = 4$.

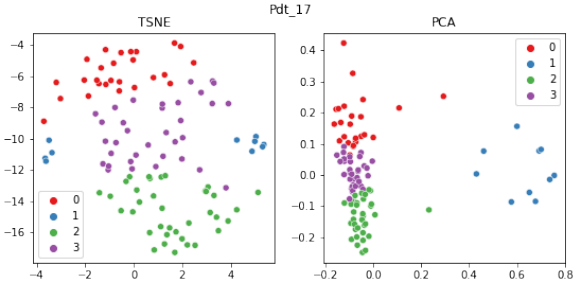


FIGURE 2 – Projection des résultats par département des élections présidentielles de 2017 dans un espace de dimension 2, à l'aide des méthodes TSNE et PCA respectivement. Le cluster bleu correspond empiriquement aux DROM-COM.

Nous avons aussi projeté les clusters prédits dans des espaces de dimension 2 à l'aide de méthodes de réduction de dimensions (PCA et T-SNE), afin de visualiser la proximité des clusters dans un espace restreint et interprétable. Ces graphiques nous montrent que certains clusters sont très distincts du reste des départements, voir par exemple pour les élections présidentielles de 2017 (voir Figure 2). Empiriquement, les clusters éloignés correspondent souvent aux départements, régions et collectivités d'Outre-mer (DROM-COM). Des figures similaires ont été générées pour chaque élection et sont accessibles sur Github.

Nuance	Listes
EG	"DÉMOCRATIE REPRÉSENTATIVE", "DÉCROISSANCE 2019", "LUTTE OUVRIÈRE", "POUR L'EUROPE DES GENS", "RÉVOLUTIONNAIRE"
G	"LA FRANCE INSOUMISE", "URGENCE ÉCOLOGIE", "ENVIE D'EUROPE", "LISTE CITOYENNE", "ESPERANTO", "EUROPE ÉCOLOGIE"
C	"RENAISSANCE", "LES EUROPÉENS"
D	"DEBOUT LA FRANCE", "ENSEMBLE POUR LE FREXIT", "UNION DROITE-CENTRE", "UDLEF"
ED	"UNE FRANCE ROYALE", "LA LIGNE CLAIRE", "ENSEMBLE PATRIOTES", "LISTE DE LA RECONQUÊTE", "PRENEZ LE POUVOIR"
A	"PARTI PIRATE", "PACE", "PARTI FED. EUROPÉEN", "INITIATIVE CITOYENNE", "ALLONS ENFANTS", "À VOIX ÉGALES", "NEUTRE ET ACTIF", "ÉVOLUTION CITOYENNE", "ALLIANCE JAUNE", "PARTI ANIMALISTE", "LES OUBLIES DE L'EUROPE", "EUROPE AU SERVICE PEUPLES"

TABLE II – Correspondance entre les listes présentes aux élections européennes de 2019 et les nuances choisies, notre interprétation.

Nuance	Candidats
EG	MÉLENCHON, POUTOU, ARTHAUD
G	JOLY, HOLLANDE
C	BAYROU
D	SARKOZY, DUPONT-AIGNAN
ED	LE PEN
A	CHEMINADE

TABLE III – Correspondance entre les candidats aux élections présidentielles de 2012 et les nuances choisies, notre interprétation.

Nuance	Candidats
EG	POUTOU, ARTHAUD
G	HAMON, MÉLENCHON
C	MACRON
D	FILLON, ASSELINEAU
ED	LE PEN, DUPONT-AIGNAN
A	CHEMINADE, LASSALLE

TABLE IV – Correspondance entre les candidats aux élections présidentielles de 2017 et les nuances choisies, notre interprétation.

Enfin, nous avons projeté les prédictions sur une carte de la France. Pour les élections européennes, nous avons par exemple la Figure 3. Pour les élections présidentielles, nous nous comparons à une visualisation du Ministère de l'Intérieur [2], ce qui nous donne la Figure 4. Certaines métriques permettent de comparer la similarité de deux clusterings, comme par exemple la *v-measure* :

$$v = \frac{2 \times \text{homogeneity} \times \text{completeness}}{(\text{homogeneity} + \text{completeness})}$$

Avec :

$$\text{homogeneity} = 1 - \frac{H(C|K)}{H(C)}$$

$$\text{completeness} = 1 - \frac{H(K|C)}{H(K)}$$

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n} \cdot \log\left(\frac{n_{c,k}}{n_k}\right)$$

K	ABN	A	C	D	ED	EG	G
0	0.48	0.03	0.48	0.35	0.34	0.03	0.52
1	**0.93**	0.01	0.14	0.16	0.15	0.02	0.18
2	0.51	0.02	0.34	0.35	0.58	0.03	0.39
3	0.49	0.03	0.42	0.38	0.48	0.03	0.45

TABLE V – Centroïdes des clusters de l'algorithme K-Means pour les élections présidentielles de 2017. Le cluster 1 correspond au cluster bleu de la figure 2. Il est marqué par une très forte abstention.

$$H(C) = -\sum_{c=1}^{|C|}\frac{n_c}{n} \cdot \log\left(\frac{n_c}{n}\right)$$

L'homogénéité et la complétude supposent des labels "vrais", on a donc : n le nombre d'observations, n_c le nombre d'observations appartenant à la classe c, n_k le nombre d'observations assignées au cluster k, $n_{c,k}$ le nombre d'observations de la classe véritable c assignées au cluster k. Contrairement à la complétude et à l'homogénéité, la v-mesure est symétrique. Une v-mesure de 1 signifie une similarité parfaite, une v-mesure de 0 signifie une dissimilarité parfaite. Nous nous sommes en grande partie basés sur [6].

En comparant la similarité des clusterings pour chaque élection, on obtient une matrice de v-mesures de dimension $n_{\text{élections}} \times n_{\text{élections}}$, et qui montre par ailleurs la symétrie de la v-mesure, voir Figure 10.

Afin d'extraire de l'information de ces v-mesures, nous avons voulu savoir si les comportements électoraux capturés par les clusters K-means sont plus similaires lors d'élections proches dans le temps ou lors d'élections du même type. Nous avons donc comparé pour chaque élection la v-mesure avec l'élection du même type, et avec l'élection la plus proche dans le temps. Par exemple, pour les élections départementales de 2015, les élections les plus proches sont les élections régionales de 2015 ($v_{\text{closest}} = 0.23$), et les élections du même type sont les élections départementales de 2021 ($v_{\text{same}} = 0.32$). Nous avons calculé la moyenne des v-similarités pour les élections les plus proches et les élections du même type, et nous obtenons que $\bar{v}_{\text{closest}} = 0.21$ et $\bar{v}_{\text{same}} = 0.29$. Nous trouvons également que pour 66% des élections, l'élection la

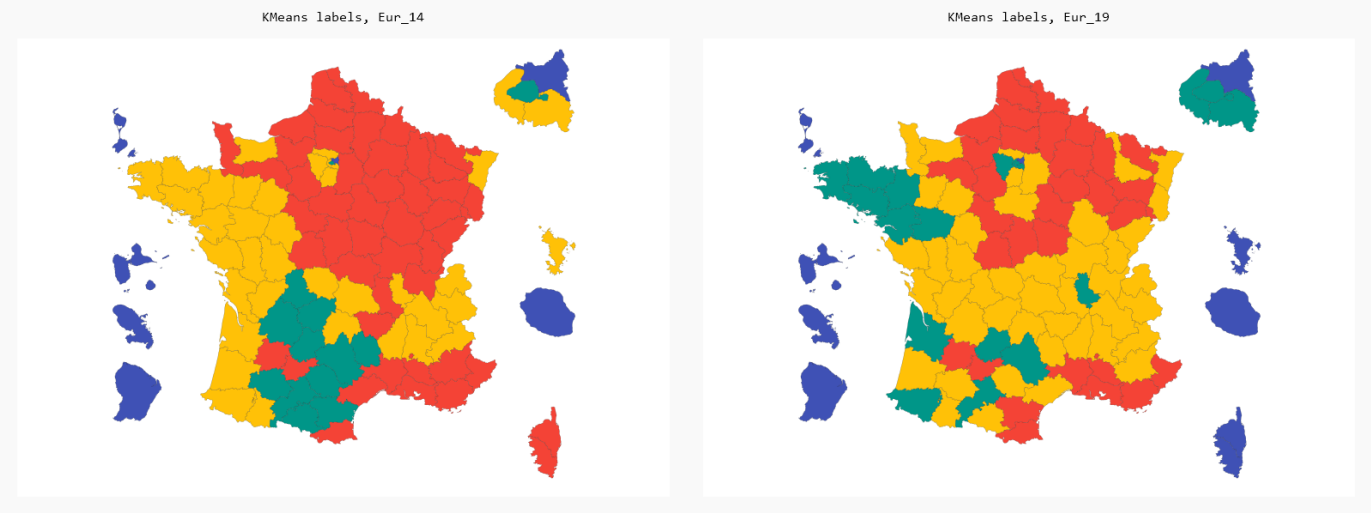


FIGURE 3 – Clustering des comportements électoraux des départements aux élections européennes de 2014 et 2019.

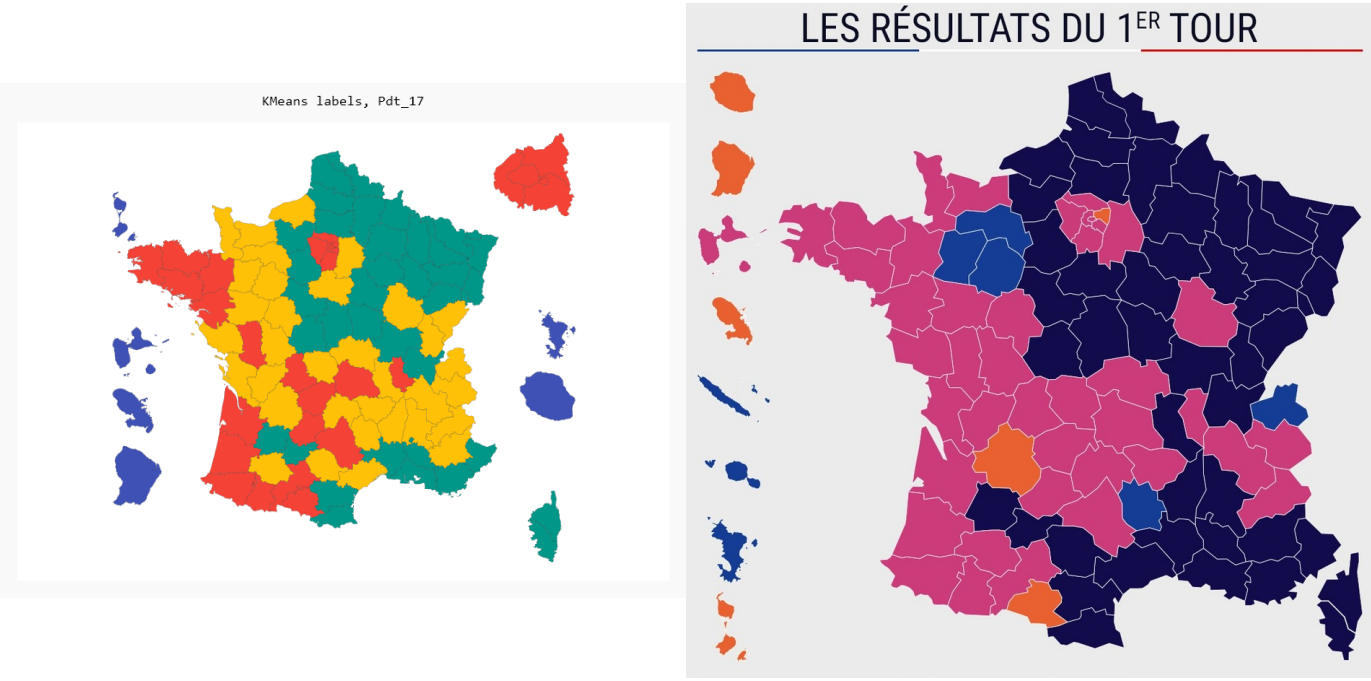


FIGURE 4 – Clustering des comportements électoraux des départements aux élections présidentielles de 2017 (gauche), carte des candidats en tête au premier tour par le Ministère de l'Intérieur (droite). La carte du Ministère s'interprète comme un clustering par argmax, nous agrégeons avec K-Means.

plus similaire par v-mesure est l'élection du même type, et non l'élection la plus proche dans le temps. On conclut donc qu'en moyenne, les comportements électoraux sont plus proches en fonction de l'élection en jeu qu'en fonction de la proximité temporelle des élections.

Notre clustering nous a permis de regrouper des départements ayant des comportements électoraux similaires à chaque élection sur plusieurs dimensions (le pourcentage de votes pour chaque nuance). Cette première agrégation nous a par la suite permis de comparer les comportements électoraux dans le temps sur la base des regroupements par élection, à l'aide de métriques de clustering. Nous montrons l'intérêt double du clustering : d'une part, les algorithmes peuvent être utilisés comme une fin, *i.e.* comme une façon de regrouper ensemble des observations similaires ; d'autre part ils peuvent être utilisés d'une façon un peu similaire à des algorithmes de réduction de dimension pour réduire la dimensionnalité et permettre d'autres opérations (on peut par exemple penser à la compression d'images). Ici, nous passons de vecteurs $x(6,)$ à un seul scalaire pour décrire chaque département à chaque

élection, ce qui facilite grandement les comparaisons par la suite.

Ces explorations nous montrent par exemple que l'abstention est extrêmement forte dans les DROM-COM, ce qui peut être considéré comme inquiétant ; d'autre part, les comportements électoraux ne doivent pas être analysés qu'à l'aune de saisonnalités ou de tendances dans le temps, mais aussi au regard du mode et aux enjeux du scrutin, qui a lui aussi une influence non-négligeable. Ce que nous montrons ici s'est par exemple illustré lors des dernières élections régionales : les scores des partis "traditionnels" (LR, PS) ont retrouvé des niveaux pré-2017, mais avec une abstention très forte et répartie inéquitablement dans la population (faible abstention des retraités, forte abstention des jeunes, des ouvriers, employés et professions intermédiaires) — voir [4] ou [3].

IV. Classification

Après ce travail de clustering, nous avons procédé à la classification de nos données, afin d'évaluer les résultats d'une approche supervisée sur les prédictions. Cette fois, l'objectif était d'entraîner différents algorithmes de classification sur toutes les données électorales entre l'année 2012 et l'année 2019 incluse, en prenant en considération d'autres *features* externes aux seules données politiques, et de les tester sur les élections de 2020 et 2021. Tous les scores obtenus ont ensuite été comparés à celui de l'approche simpliste qui consiste à prédire la nuance politique la plus représentée dans les suffrages, en l'occurrence G (la gauche).

A. Préparation des données pour la classification

1) Prise en compte de la démographie: Afin d'ajouter un critère de classification non-électoral à notre jeu de données, nous avons récupéré sur le site de l'INSEE les données démographiques annuelles par département, et avons ajouté à nos données électorales agrégées la mesure de l'évolution démographique annuelle pour chaque département (différence de la population de l'année A et de celle de l'année $A-1$).

Les données démographiques disponibles ne permettaient toutefois pas une couverture totale du pays, mais se limitaient à la métropole et aux *départements* d'outre-mer. A l'inverse, les *territoires* d'outre-mer ne sont caractérisés que par des recensements épisodiques et trop peu fréquents pour être significatifs dans cette étude. Nous avons donc choisi de restreindre la classification aux départements pour lesquels les données démographiques étaient disponibles.

2) Prise en compte du taux de chômage: De même, nous avons tenté d'élargir le nombre de *features* susceptibles d'influencer les scrutins en ajoutant les données sur les taux de chômage trimestriels départementaux. Dans ce cas, nous avons agrégé ces données pour obtenir une moyenne annuelle pour chaque département, et avons joint ces informations à nos données initiales. De la même manière, les données ne sont pas toujours exhaustives, mais le nombre de valeurs manquantes reste suffisamment faible pour se permettre de supprimer simplement les lignes correspondantes.

3) Normalisation: Ensuite, deux approches ont été envisagées : normaliser les données en divisant les informations démographiqueset de chômage par la population française totale sur l'année en question, de façon à travailler sur la proportion que représente chaque département du point de vue démographique/emploi, ou normaliser sur l'ensemble du *dataset*, de façon moins "intelligible" mais purement mathématique.

C'est finalement cette deuxième option que nous avons choisie, afin de garantir un minimum de transformations sur nos données. Nous avons alors centré et réduit les valeurs numériques du *dataset* grâce à la classe `StandardScaler` de la librairie `scikit-learn`.

B. Approche exploratoire : divers algorithmes aux paramètres par défaut

Nous avons commencé par une approche exploratoire, en évaluant plusieurs algorithmes de classification (paramètres par défaut de `sklearn`) sur chacun des jeux de données (avec ou sans chômage).

Modèle	Score	% baseline
Baseline	0.491525	100%
Randomforest	0.596610	**121.4%**
Logistic Regression	0.210169	42.8%
SVC	0.593220	**120.7%**
KNN	0.423729	86.2%
Tree	0.223729	45.5%
Naive Bayes	0.400000	81.4%

Table VI – Premiers scores obtenus par divers algorithmes de classification avec les paramètres par défaut, en utilisant les prédicteurs suivants : année, département (one-hot-vector), type d'élection (one-hot-vector), et la croissance démographique en un an du département.

Modèle	Score	% baseline
Baseline	0.494881	100%
Randomforest	0.569966	**115.2%**
Logistic Regression	0.426621	86.2%
SVC	0.593857	**120.0%**
KNN	0.423208	85.5%
Tree	0.259386	52.4%
Naive Bayes	0.409556	82.8%

Table VII – Premiers scores obtenus par divers algorithmes de classification avec les paramètres par défaut, en utilisant les prédicteurs suivants : année, département (one-hot-vector), type d'élection (one-hot-vector), la croissance démographique en un an du département, **et** le taux de chômage moyen annuel.

On constate au premier coup d'oeil que certains sont d'emblée plus performants que d'autres, et les meilleurs candidats semblent être le Support Vector Classifier, suivi de près par la Random Forest, et enfin la méthode des K plus proches voisins (KNN), déjà moins performante que la prédiction constante.

C. K Nearest Neighbors

L'algorithme choisit un paramètre $k = 5$ par défaut, ce qui peut sembler peu vu le nombre de dimensions de nos données. Nous avons alors cherché à définir le meilleur paramètre k pour ce classifieur, en traçant le score en fonction du nombre k de plus proches voisins à considérer, tous paramètres égaux par ailleurs (Figure 5 pour les données sans prise en compte du taux de chômage, 6 pour celles avec prise en compte du chômage). On choisit alors la valeur $k = 16$, ce qui permet

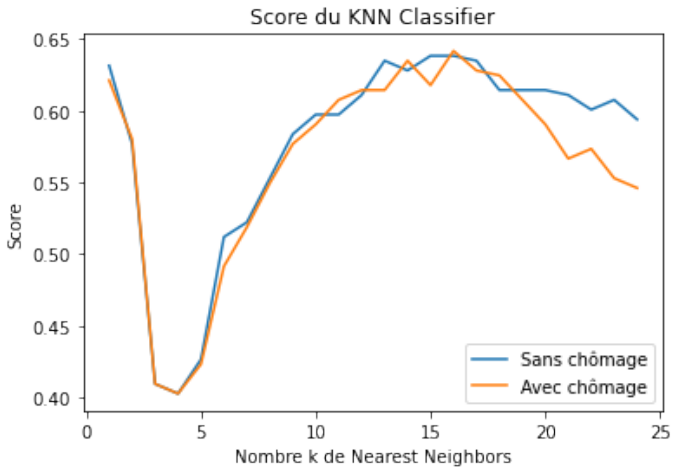


FIGURE 5 – Score de prédiction obtenu par l'algorithme KNN en faisant varier le nombre de proches voisins à considérer, pour chacun de nos jeux de données. Graphiquement comme par le calcul, on observe que le k idéal dans ces conditions est égal à 16.

d'obtenir un score de classification égal à $126,9\%$ et $129,7\%$ du score de base (sans puis avec chômage), et fait de cet algorithme le plus performant à ce stade.

Ces méthodes permettent d'obtenir les résultats similaires à la visualisation suivante : L'ensemble des résultats pour cette méthode et les différents paramètres est reproduit en annexe.

D. Decision Tree et Random Forest

On s'intéresse ensuite à l'algorithme de DecisionTree, en faisant varier le critère de split entre les deux valeurs possibles, "Gini" et "Entropy", ainsi que la profondeur maximale (Figure 7). Dans les deux cas, on observe que les performances sont systématiquement moins bonnes que celles atteintes pour une profondeur fixée à 1, ce qui correspond précisément à l'approche constante. L'algorithme de RandomForest, quant à lui, est nettement plus prometteur (Figure 8). Les performances maximales sont atteintes pour le second jeu de données (122% du score atteint en "baseline" avec le critère "gini" et 35 estimateurs), mais ces résultats dépendent fortement du `random_state` choisi, et sont donc à manipuler avec précautions. Il reste toutefois moins performant sur cet échantillon que l'algorithme KNN, et ce sur plusieurs itérations, en faisant varier le `random_state`.

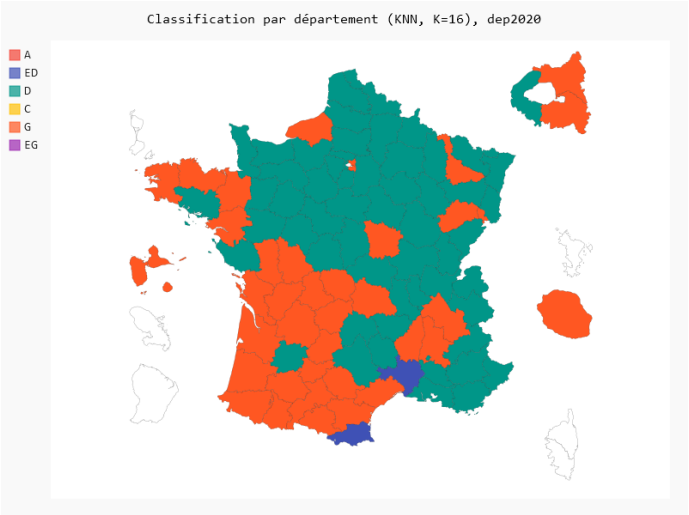


FIGURE 6 – Visualisation sur carte des résultats de classification obtenus sur les données avec chômage, par l'algorithme KNN (k=16) pour les départementales de 2020.

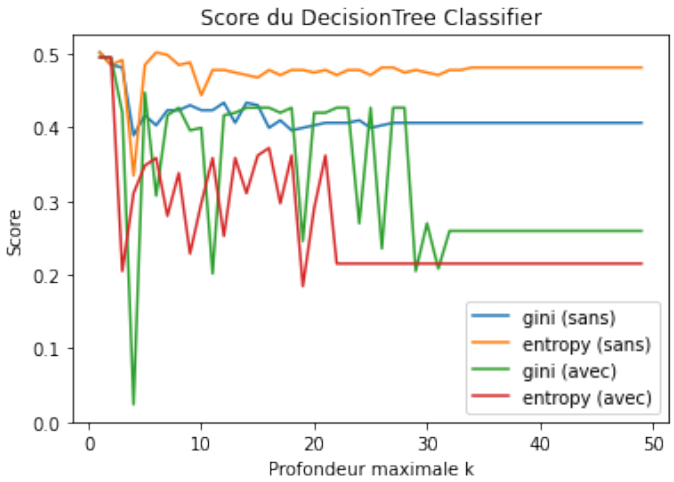


FIGURE 7 – Score de prédiction obtenu par l'algorithme DecisionTree en faisant varier le critère de split (*gini* ou *entropy*) et la profondeur maximale de l'arbre de décision (k), pour chacun de nos jeux de données (avec ou sans chômage). Graphiquement comme par le calcul, on observe que les performances sont toujours moins bonnes que la baseline, avec en outre une forte variabilité de l'allure des courbes d'une itération à l'autre (liée à la part d'aléatoire de la méthode)

Comme précédemment, nous visualisons les résultats fournis par cette méthode sur une carte, qui nous donne l'allure visible en Figure 9 dans le cas des régionales de 2021. L'ensemble des cartes obtenues, ainsi que la base de comparaison, est visible en annexe.

E. Support Vector Classifier

Pour finir, nous nous intéressons d'un peu plus près au Support Vector Classifier, en testant plusieurs noyaux. Nous

FIGURE 8 – Deux exemples de tracé du score de l'algorithme RandomForest en fonction du critère de split (*gini*, *entropy*), et du paramètre "nombre d'estimateurs", pour chacun de nos jeux de données, et pour deux itérations différentes (random_state différents). La variabilité observée ne permet pas de trancher sur les meilleurs paramètres à utiliser pour notre étude.

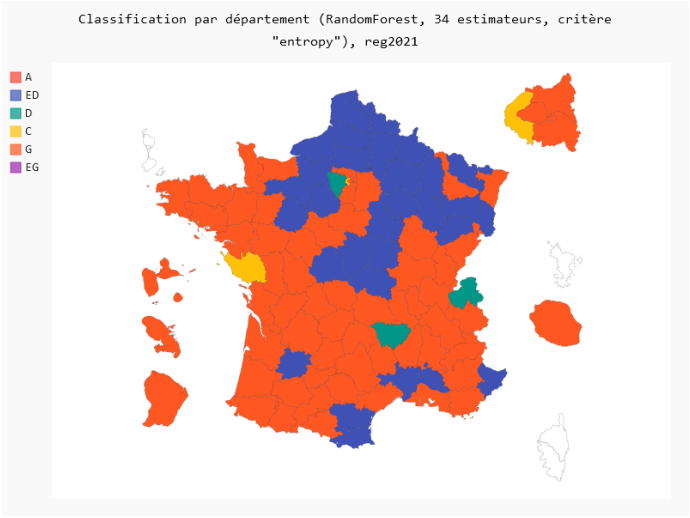


FIGURE 9 – Visualisation cartographique des résultats obtenus par l'algorithme RandomForest sur les données avec chômage, avec les paramètres criterion=entropy et n_estimator=34.

constatons que les algorithmes ont des performances très dissimilaires, et que les résultats sont légèrement moins bons en tenant compte des taux de chômage que lorsque cette *feature* n'est pas utilisée (mais ces écarts semblent peu significatifs). Dans tous les cas, ils restent moins bons que ceux obtenus par KNN (respectivement $122,2\%$ et $120,0\%$ de la baseline).

Dans ce cas, nous obtenons des visualisations similaires à celle obtenue précédemment à partir de notre jeu de résultats, et que nous proposons en annexe.

F. Visualisation

L'ensemble des cartes en annexe présente les résultats obtenus par les trois "meilleurs" algorithmes de classification sur nos différents jeux de données. Ces visualisations permettent d'évaluer plus intuitivement la performance des algorithmes, et de remettre en question la pertinence de nos modèles.

Kernel	Sans chômage	Avec chômage
linear	0.135593	0.153584
poly	0.600000	0.593857
rbf	0.593220	0.593857
sigmoid	0.505085	0.453925

TABLE VIII – Différents scores obtenus sur nos jeux de données en faisant varier la méthode à noyau utilisée par le Support Vector Classifier

Globalement, l'ensemble des classifieurs permet de distinguer une tendance globale qui se vérifie dans les données réelles : une séparation Nord-Ouest / Sud-Est relativement nette dans le comportement électoral des Français et bien connue en sciences sociales et politiques.

On observe toutefois que l'algorithme du KNN, bien que plus performant *a priori* (donnant le meilleur score), tend à lisser les résultats, et à creuser le déséquilibre des classes. C'est ainsi que les partis recueillant moins de voix (le centre, les extrêmes, et la catégorie "Autre") sont systématiquement sous-représentés dans les résultats KNN au profit de la gauche et de la droite.

Les deux autres algorithmes, la RandomForest et le SVC, sont plus intéressants à cet égard, et permettent plus de diversité de classe au sein des données, sans être exacts pour autant (sur-représentation fréquente de l'extrême droite).

Intuitivement, on ne perçoit pas de la même manière une erreur de classification entre la droite et l'extrême droite, qu'entre l'extrême droite et la gauche, tandis que pour la machine, cela revient au même. Une idée pour contrecarrer cet aspect pourrait consister à modéliser les classes de façon à leur donner une relation d'ordre, ce qui n'est pas le cas dans notre approche.

V. Conclusion

Nous proposons plusieurs méthodes pour extraire de l'information des résultats aux scrutins en France. La variété et le volume des données nécessite une bonne connaissance de celles-ci afin de pré-traiter et nettoyer correctement. Le pré-traitement et nos opérations de clustering mettent en évidence des éléments communs : chaque type de scrutin nécessite un nettoyage propre, et amène à des résultats propres. Aussi, les modèles de classification appliqués à nos données donnent des résultats limités. On peut en déduire ceci : le temps (l'année), le lieu (le département), le type d'élection et la démographie ne permettent pas d'expliquer complètement les variations des vainqueurs du premier tour. Ajouter le chômage comme feature ne permet pas d'améliorer les résultats non plus. Afin d'améliorer les modèles, on pourrait inclure ou créer de nouvelles features (données de sécurité, distribution des âges, revenus et capitaux moyens dans les territoires...), améliorer la granularité (niveau bureau de vote plutôt que département par exemple), ou même la granularité de nos labels : nous avons agrégé les partis et listes selon nos représentations, mais nous pouvons faire l'hypothèse que le spectre gauche-droite ne se vérifie pas dans les représentations des électeurs français. Nous pourrions également utiliser des modèles de séries temporelles, bien que dans ce cas nous disposerions de peu de points (en moyenne, il y a à peine un peu plus d'un scrutin par an, dont certains sont si rapprochés qu'ils ne peuvent pas capturer de variations dans l'opinion). Ce projet nous a permis de nous confronter à des données réelles et de faire du "big data mining" : bien que nos résultats ne soient pas spectaculaires, nous trouvons bien des corrélations, patterns et prédictions meilleures que n'en trouveraient des estimateurs basés sur une statistique (type classifieur dummy), ou des modèles déterministes.

Références

[1] Ministère de l'Intérieur. Circulaire relative à l'attribution des nuances politiques aux candidats aux élections municipales et communautaires des 15 et 22 mars 2020. https://www.legifrance.gouv.fr/circulaire/id/44929. Accessed : 15-12-2021.

[2] Ministère de l'Intérieur. Election présidentielle 2017 : résultats globaux du premier tour. https://mobile.interieur.gouv.fr/Archives/Archives-elections/Election-presidentielle-2017/Election-presidentielle-2017-resultats-globaux-du-premier-tour. Accessed : 15-12-2021.

[3] Ipsos. 2nd tour : qui sont les abstentionnistes ? https://www.ipsos.com/fr-fr/regionales-2021/2nd-tour-qui-sont-les-abstentionnistes. Accessed : 15-12-2021.

[4] L'Alsace. Âge, sexe, revenu, région... qui s'est le plus abstenu ce dimanche ? https://www.lalsace.fr/politique/2021/06/21/age-sexe-revenu-region-qui-s-est-le-plus-abstenu-ce-dimanche. Accessed : 15-12-2021.

[5] Vie Publique. Quel mode de scrutin dans les communes de moins de 1000 habitants ? http://shorturl.at/iowHP. Accessed : 15-12-2021.

[6] Sklearn. Clustering. https://scikit-learn.org/stable/modules/clustering.html. Accessed : 15-12-2021.

Annexes

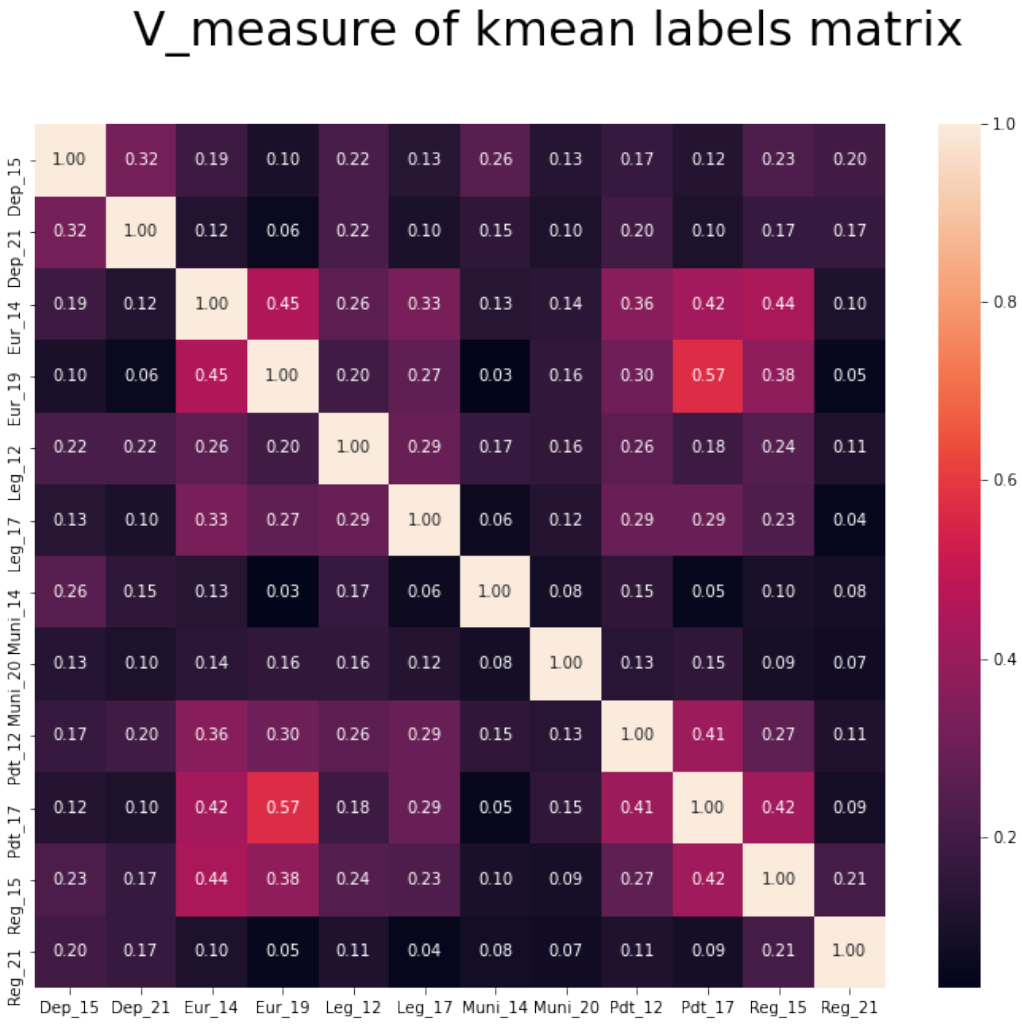
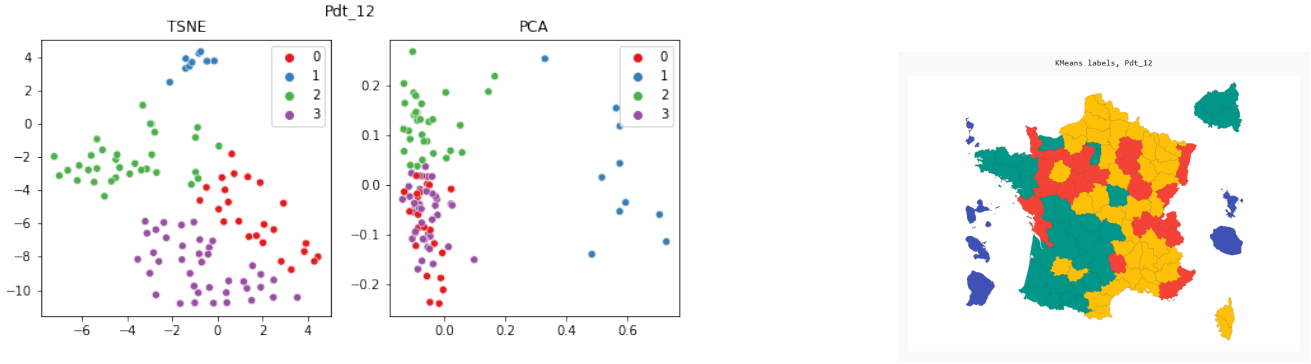
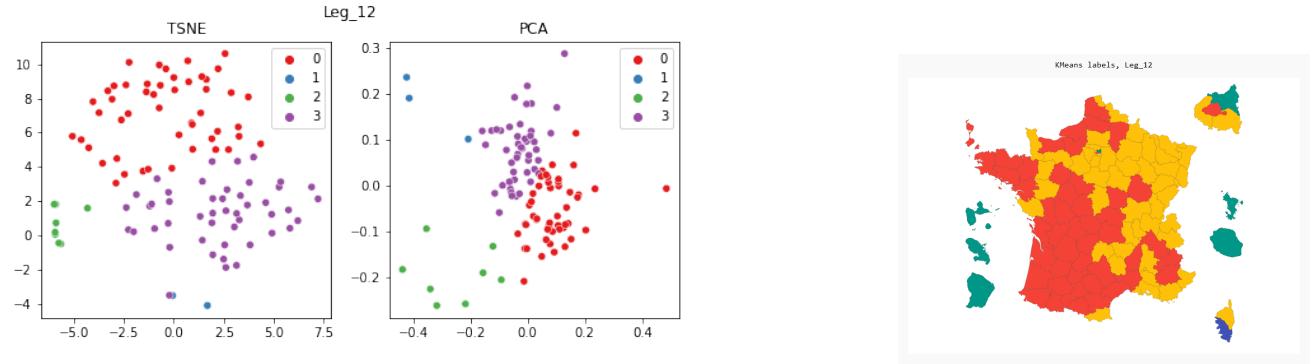


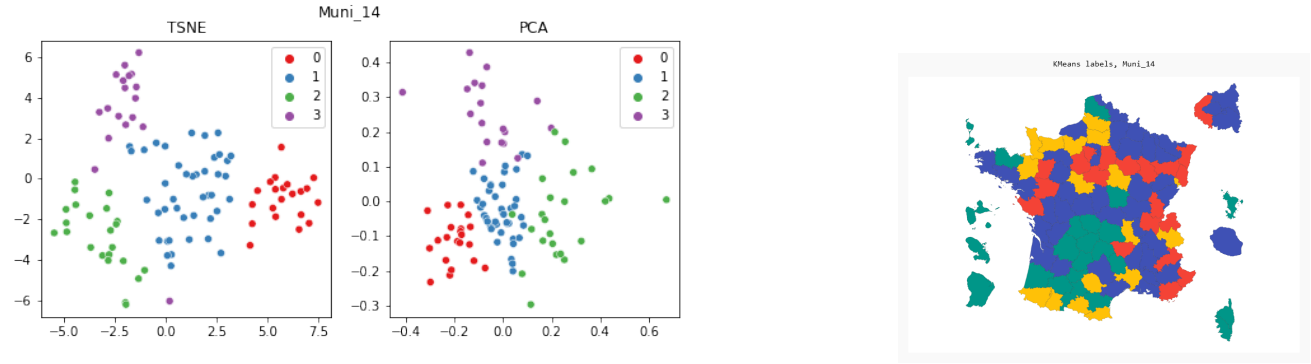
FIGURE 10 – V-mesure des clusterings K-means pour chaque couple d'élections. On peut l'interpréter de la façon suivante : les comportements électoraux aux élections municipales de 2014 sont très différents des comportements électoraux aux élections législatives de 2017 ; en revanche, les comportements électoraux aux élections présidentielles de 2017 sont très proches des comportements électoraux aux élections européennes de 2019.



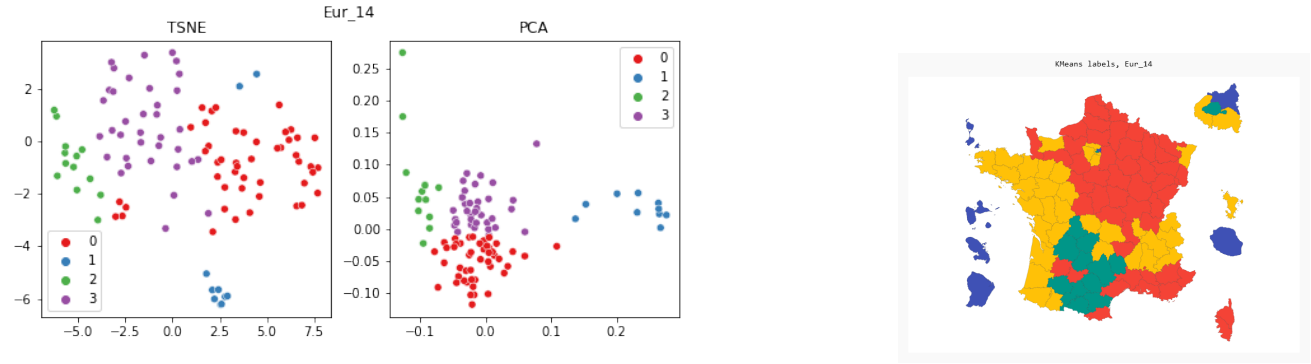
(a) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections présidentielles de 2012.



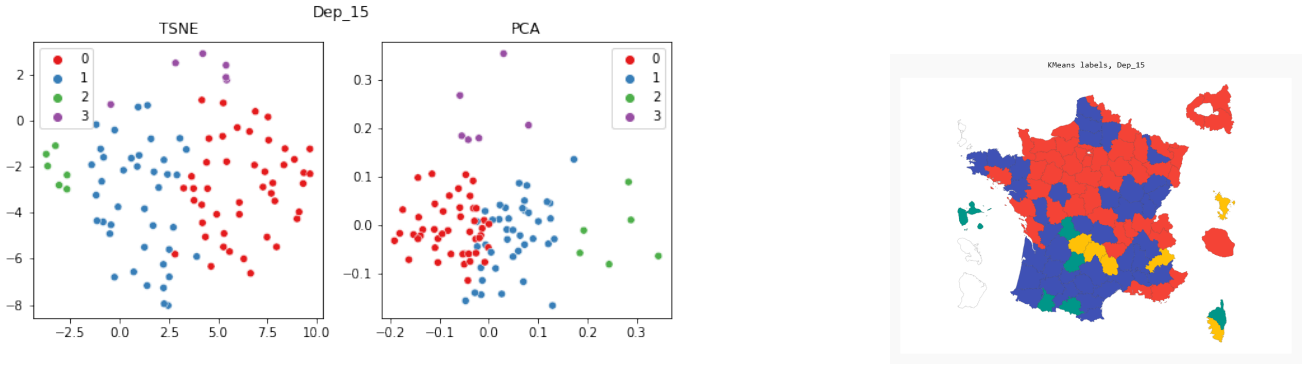
(b) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections législatives de 2012.



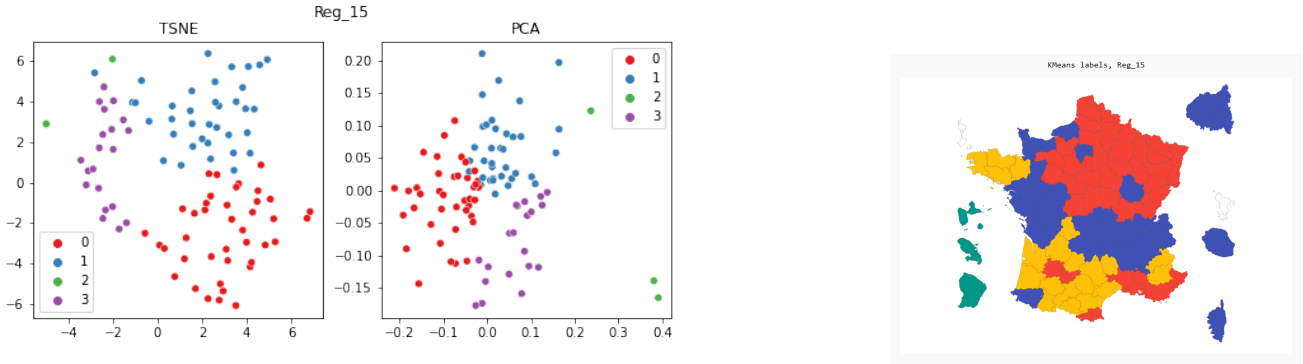
(c) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections municipales de 2014.



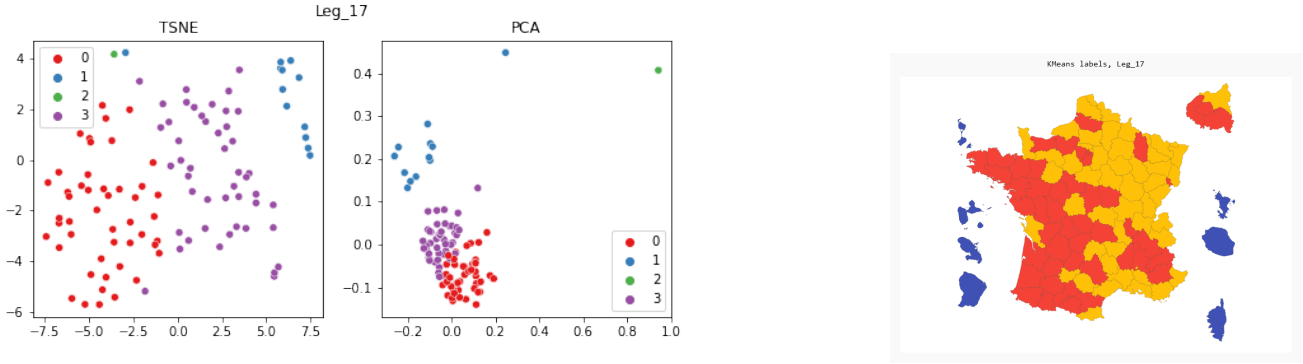
(d) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections européennes de 2014.



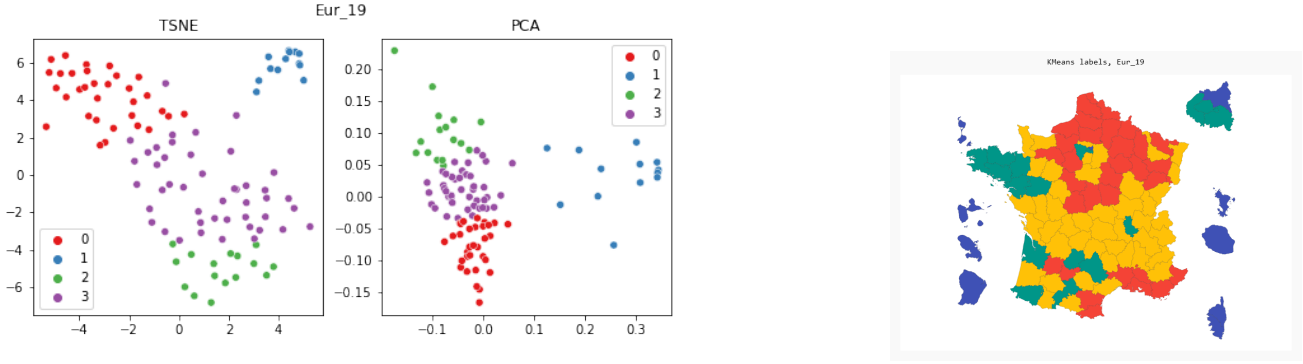
(e) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections départementales de 2015.



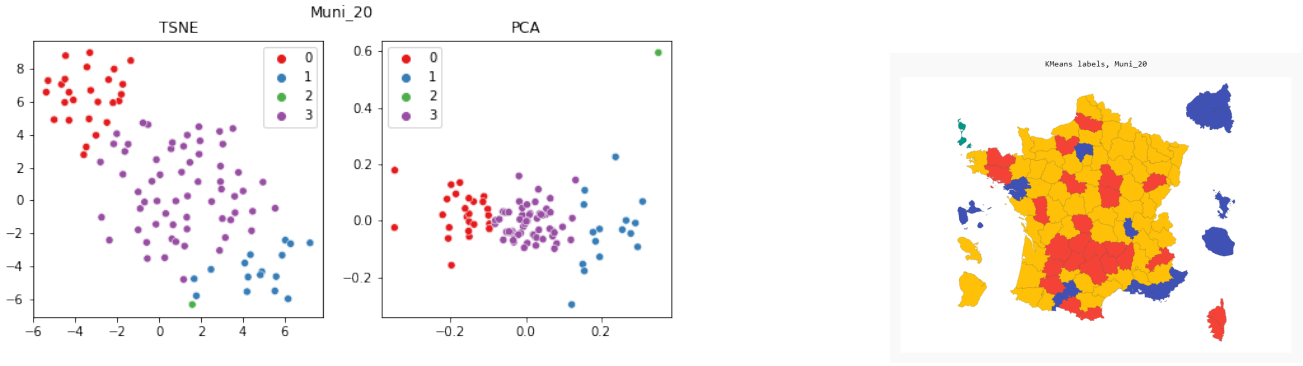
(f) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections régionales de 2015.



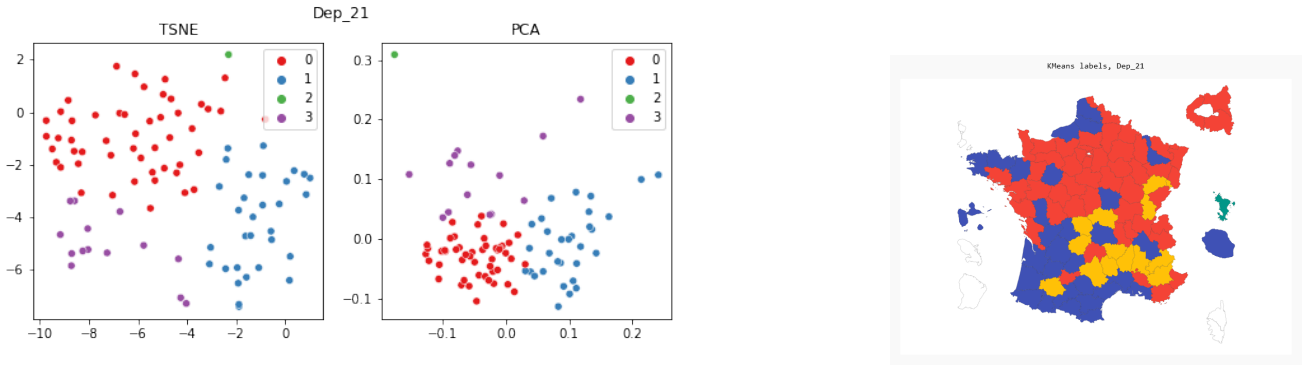
(g) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections législatives de 2017.



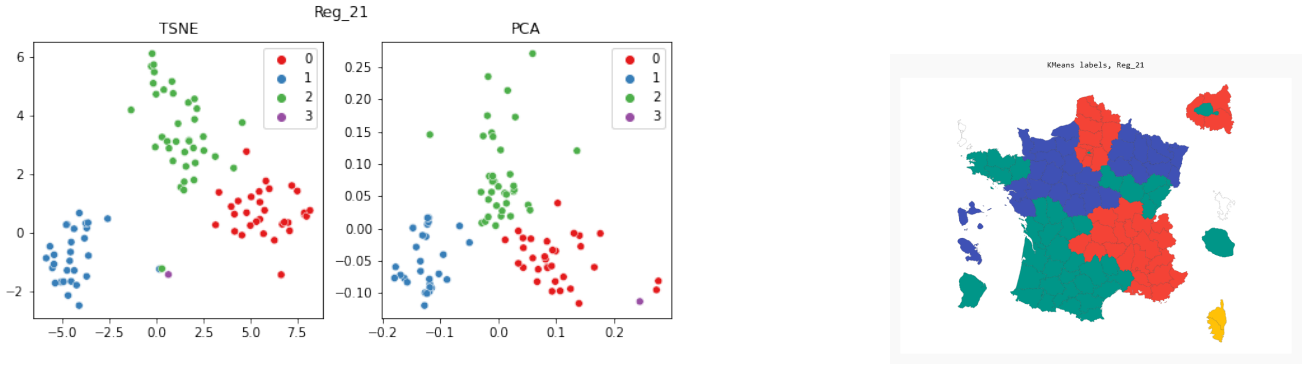
(h) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections européennes de 2019.



(i) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections municipales de 2020.

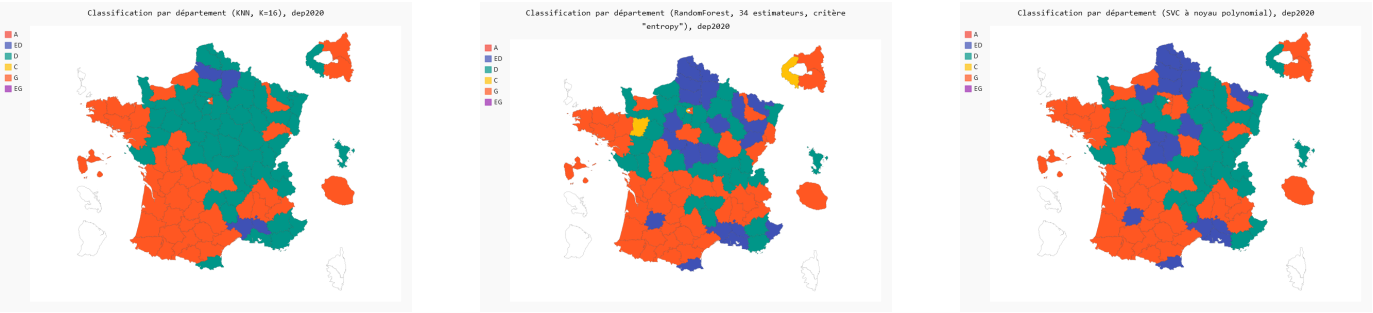


(j) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections départementales de 2021.

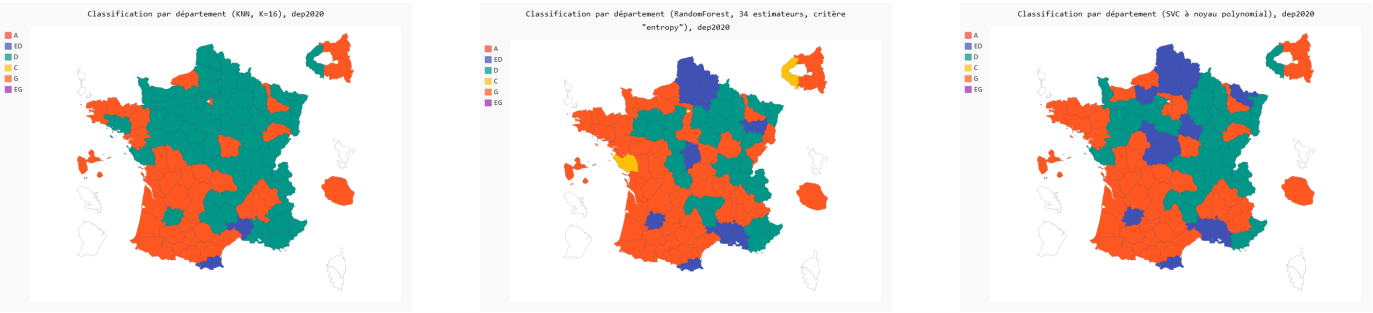


(k) Projection en deux dimensions (gauche) et sur une carte (droite) du clustering KMeans pour les élections régionales de 2021.

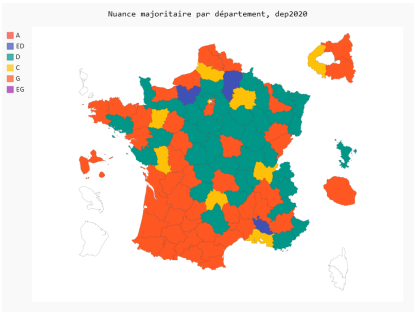
FIGURE 11 – Projection en deux dimensions et sur une carte des prédictions K-Means pour toutes les élections non présentées dans le corps du rapport.



(a) Résultats par KNN, RandomForest, et SVC pour les données sans le chômage

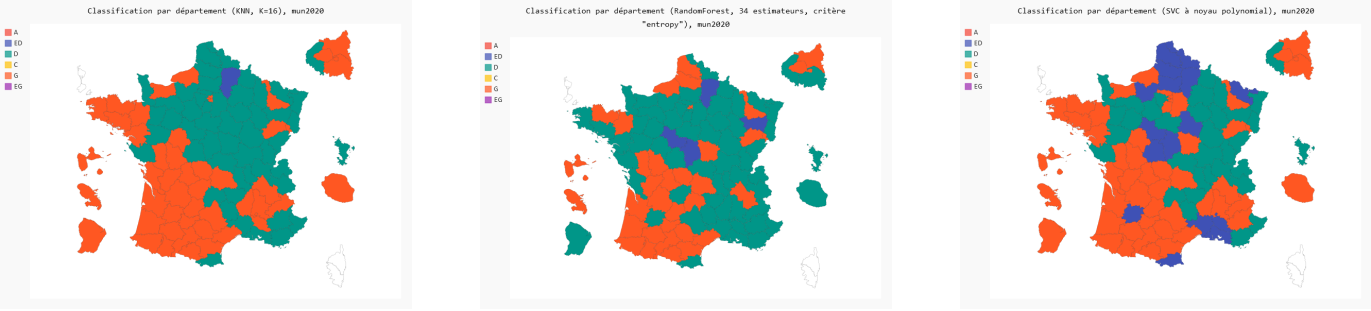


(b) Résultats par KNN, RandomForest, et SVC pour les données avec le chômage

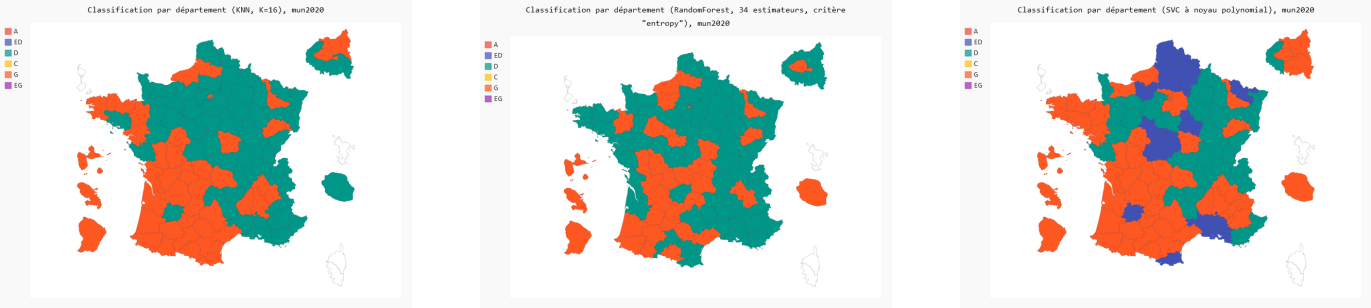


(c) Résultats réels

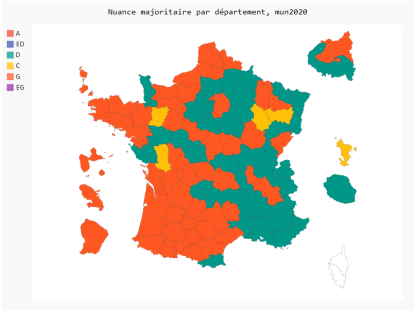
FIGURE 12 – Résultats obtenus pour les départementales 2020. On constate que l'algorithme des K plus proches voisins a tendance à "lisser" les résultats, en prédisant quasi exclusivement les deux partis majoritaires, sans retranscrire la variété des résultats. Cela n'est pas très étonnant vu les paramètres choisis (k "grand", égal à 16). A cet égard, l'algorithme de RandomForest semble visuellement le plus performant. Dans les trois cas, on retrouve plutôt bien la séparation "Nord-Ouest / Sud-Est" observable sur les données réelles.



(a) Résultats par KNN, RandomForest, et SVC pour les données sans le chômage

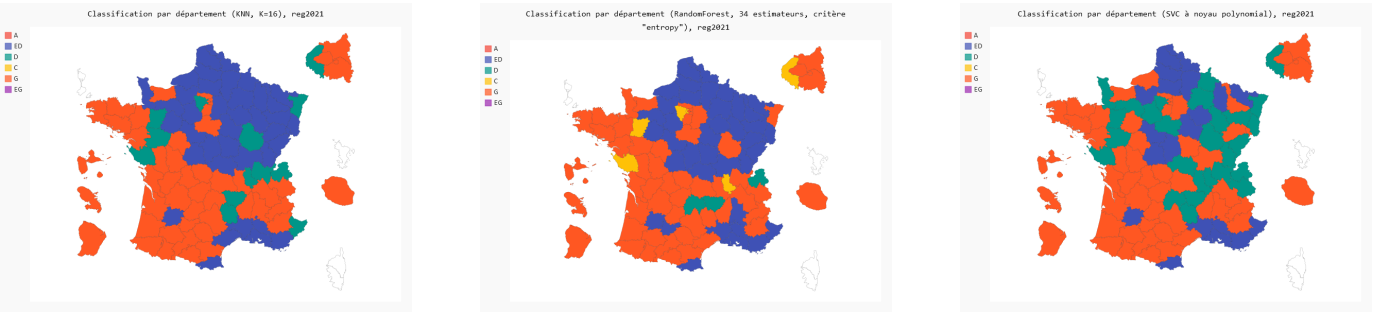


(b) Résultats par KNN, RandomForest, et SVC pour les données avec le chômage

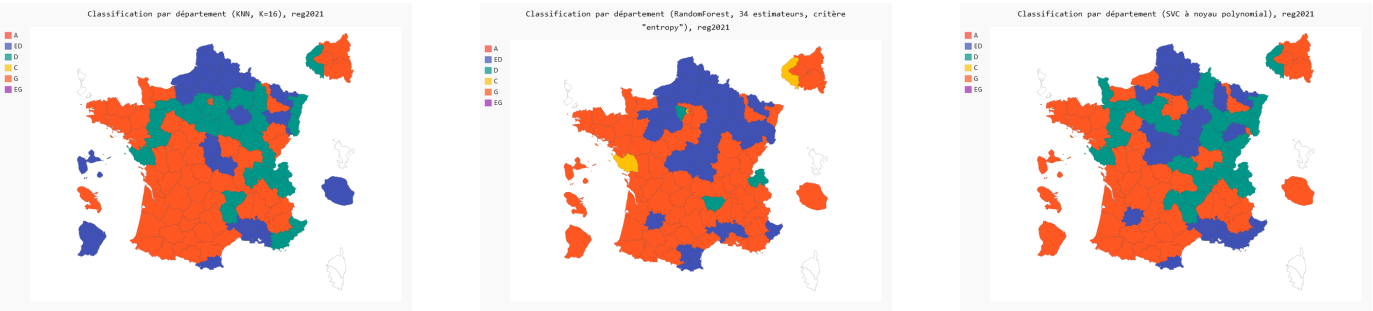


(c) Résultats réels

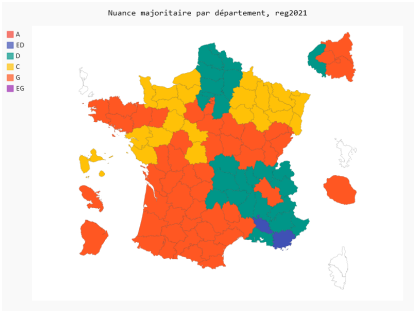
FIGURE 13 – Résultats obtenus pour les Municipales 2020. Comme précédemment, le KNN retransmet moins bien la variété des classes, ici déséquilibrées, mais le SVC et le RandomForest tendent à surreprésenter certaines nuances (en bleu, l'extrême droite), tandis que le KNN est visuellement plus proche des résultats réels.



(a) Résultats par KNN, RandomForest, et SVC pour les données sans le chômage



(b) Résultats par KNN, RandomForest, et SVC pour les données avec le chômage



(c) Résultats réels

FIGURE 14 – Résultats obtenus pour les Régionales 2021. Dans ce cas, où de nombreux départements ont voté pour un parti généralement moins représenté dans les données (en jaune, le centre), on constate qu'aucun algorithme n'est réellement efficace : si on retrouve toujours cette séparation Nord-Ouest / Sud-Est, tous les algorithmes tendant à sur-prédire l'extrême-droite (bleu) et sous-prédire le centre.