AVIGNON UNIVERSITÉ

Prospect d'une compagnie d'assurances

Groupe 08
Sylvain DE LANGEN
Audran BERT
Jérémie OPIGEZ

16 juillet 2023

Master 2 informatique
IA
UE Business intelligence & Systèmes décisionnels
ECUE Application Business Intelligence

Responsable
Vincent Labatut

UFR
SCIENCES
TECHNOLOGIES
SANTÉ

AVIGNON UNIVERSITÉ
CERI

CENTRE
D'**ENSEIGNEMENT**
ET DE **RECHERCHE**
EN **INFORMATIQUE**
ceri.univ-avignon.fr

Sommaire

Titre 1

Sommaire 2

1 Présentation 4
- 1.1 Contexte . . . 4
- 1.2 Organisation . . . 4

2 Données 5
- 2.1 Caractéristiques . . . 5
- 2.2 Nettoyage . . . 6
  - 2.2.1 Encodage . . . 6
  - 2.2.2 Entreprises figurant plusieurs fois dans le jeu de données . . . 6
  - 2.2.3 Valeurs manquantes . . . 6
  - 2.2.4 Valeurs aberrantes . . . 7
- 2.3 Traitement . . . 7
- 2.4 Analyse descriptive . . . 8
  - 2.4.1 Effectif . . . 8
  - 2.4.2 Chiffre d'affaires total . . . 8
  - 2.4.3 Chiffre d'affaires à l'exportation . . . 9
  - 2.4.4 Endettement . . . 9
  - 2.4.5 Ratio bénéfice . . . 10
  - 2.4.6 Évolution de l'effectif . . . 10
  - 2.4.7 Risque . . . 11
  - 2.4.8 Évolution du risque . . . 12
  - 2.4.9 Age . . . 13
  - 2.4.10 Changement de direction . . . 13
  - 2.4.11 Type com . . . 14
  - 2.4.12 Activité . . . 14
  - 2.4.13 Forme jur simpl . . . 16
  - 2.4.14 Actionnaire . . . 17
  - 2.4.15 Département . . . 17

3 Méthodes 19
- 3.1 Outils de fouille . . . 19
  - 3.1.1 Clustering . . . 19
  - 3.1.2 Classification supervisée . . . 20
- 3.2 Recodage & Rééquilibrage . . . 21
  - 3.2.1 Clustering . . . 21
  - 3.2.2 Classification supervisée . . . 21
- 3.3 Évaluation . . . 22
  - 3.3.1 Clustering . . . 22
  - 3.3.2 Classification supervisée . . . 22
- 3.4 Implémentation . . . 22
  - Analyse descriptive . . . 23
  - Kmeans . . . 23
  - Clustering mixte . . . 23
  - Prédiction d'attributs . . . 23

4 Résultats **24**
4.1 Typologie des entreprises 24
4.1.1 Kmeans avec 3 clusters 24
4.1.2 Clustering mixte avec 25 clusters 31
4.2 Prédiction d'attributs 36
4.2.1 Résultats divers 36
4.2.2 Arbres de décision 36

5 Conclusion **38**

1 Présentation

1.1 Contexte

Une compagnie d'assurance cherche à prospecter de nouveaux clients.

Pour ce faire, la compagnie a déjà compilé une base de données de 370000 entreprises, filtrée sur certains critères (financiers notamment) pour obtenir une liste de 108000 entreprises. Nous avons accès à cette dernière.

La compagnie a systématiquement proposé un rendez-vous aux entreprises de cette liste. Environ 12000 ont répondu favorablement. Nous avons également accès à cette information.

Notre objectif est, par l'étude détaillée de cette de base de données, d'identifier des *profils* d'entreprises, et d'étudier comment ces profils répondent à la demande de rendez-vous.

Ces informations permettraient, entre autres, de pouvoir filtrer et cibler plus efficacement des entreprises en amont, mais également de mieux comprendre le profil des entreprises qui acceptent les rendez-vous, notamment par rapport à leur santé financière.

1.2 Organisation

Nous sommes un groupe de 3. Il était donc important de bien distribuer les tâches pour maximiser notre productivité. Pour ce faire, nous avons dès le début mutualisé nos travaux sur un dépôt Git, où nous travaillons par branches.

Cette approche s'est avérée utile pour minimiser le travail dupliqué. Cela nous a permis de très rapidement mettre en commun des routines communes, notamment pour l'importation et le filtrage du jeu de données, tout au long du projet.

Pour l'exploration préliminaire des données, nous avons systématiquement travaillé sur des notebook Jupyter indépendants pour garder des traces et pouvoir effectuer des "brouillons" efficacement, mais auxquels chacun a un accès pratique. Cela simplifie notamment la mutualisation de ces analyses dans un document unique. Nous avons tâché de chercher et d'analyser le sens des colonnes, débattant sur celles-ci lorsque nécessaire, tout en évitant de dupliquer de la recherche sur les mêmes variables.

Certaines données étant très spécifiques au domaine du jeu de données, notamment celles financières, il nous a fallu tout au long du projet effectuer des recherches pour éclaircir le sens des termes, ou encore pour identifier l'importance de critères et savoir identifier quelles valeurs semblent incohérentes à des fins de nettoyage.

Pour ce qui est de l'aspect technique, nous avons réalisé la vaste majorité de notre travail d'analyse sous Python, qui bénéficie d'un écosystème complet et de bonnes capacités de prototypage. Nous avons utilisé les bibliothèques suivantes, qui ont été étudiées au sein des UCE de ce cours, et sont largement répandues dans l'industrie :

- Manipulation de données : Pandas, NumPy
- Génération de graphiques : Matplotlib
- Analyse de données & prédiction : Scikit-learn

Seaborn n'a pas été abordé en cours, mais nous en avons fait usage pour la génération de certains graphiques pour lesquels cette bibliothèque paraissait plus adaptée. Il s'agit d'une surcouche de Matplotlib qui présente des différences au niveau de l'interface et des graphiques générés, mais propose également davantage de fonctionnalités que Matplotlib n'expose pas directement. Seaborn est largement utilisé en analyse de données dans l'industrie et en recherche.

Bien que nous avons généralement tous collaboré et échangé sur chacune des parties du travail, en durée de travail absolue, on pourrait résumer le gros de la distribution du travail à :

- Audran : Clustering

- Jérémie : Analyse descriptive
- Sylvain : Prédiction d'attributs

2 Données

2.1 Caractéristiques

Le jeu de données nous est fourni au biais d'un fichier CSV `base_prospect.csv` avec un format d'encodage texte `ISO-8859-1`.

Le jeu de données contient 108 576 lignes.

Ce fichier comporte **18** colonnes, qui représentent :

- `code_cr` (chaîne de caractères, catégoriel, nominal) : Code régional de la caisse d'assurances de l'entreprise (par exemple, `GPVL` pour Groupama Pays Val de Loire). Ce code est systématiquement corrélé à certains départements.
- `dept` (entier, catégoriel, nominal) : Il s'agit du numéro de département de domiciliation de l'entreprise (par exemple, 84 pour le Vaucluse).
- `effectif` (entier, numérique) : Le nombre d'employés dans l'entreprise.
- `ca_total_FL` (entier, numérique) : Chiffre d'affaires total en k€.
- `ca_export_FK` (entier, numérique) : Part du chiffre d'affaires obtenu à l'exportation (vente en dehors de la France).
- `risque` (chaînes de caractères, catégoriel, ordinal) : Un score attribué à l'entreprise en fonction de sa santé financière. Un score plus faible signifie une santé inférieure. Dans le jeu de données, ce score est encodé sous forme d'intervalles : `1-6`, `7`, `8-10`, `11-13`, `14`.
- `endettement` (réel, numérique) : Ratio d'endettement = $\frac{capitaux\ propres}{total\ bilan}$. La formule correspond en fait au "Ratio d'autonomie financière" et non à la formule du ratio d'endettement. Plus la valeur est élevée, plus l'entreprise est en bonne santé.
- `evo_benefice` (réel, numérique) : Taux d'évolution du bénéfice. Une valeur positive indique que l'entreprise à amélioré ses bénéfices. Une valeur de 0 signifie que le bénéfice n'a pas changé, 1 signifie qu'il a doublé et -1 qu'il a été divisé par deux.
- `ratio_benef` (réel, numérique) : Ratio sur bénéfice $\frac{100 * benefices}{CA\ total}$. Une valeur proche de 0 signifie que l'entreprise ne tire que très peu de bénéfice par rapport à son chiffre d'affaire.
- `evo_effectif` (réel, numérique) : Taux d'évolution de l'effectif. Une valeur positive indique que l'entreprise a augmenté son effectif.
- `evo_risque` (entier, ordinal) : Décrit l'évolution du risque. Par exemple, une valeur de 2 dit que la valeur de risque a progressé de 2 catégories (-2 a reculé de 2). Cela va de -4 à 4 (il y a 5 catégories donc cela peut progresser maximum de 4 catégories). Il y a 936 valeurs manquantes.
- `age` (entier, numérique) : Âge de l'entreprise en années, entre 1 et 124.
- `type_com` (chaînes de caractères, catégoriel, nominal) : Décrit si l'entreprise est dans un pôle urbain, en périphérie d'un pôle urbain (commune monopolarisée/multipolarisée) ou en zone rural (espace à dominante rural). Il y a un peu plus de 1000 valeurs non renseignées.
- `activite` (chaînes de caractères, catégoriel) : Définit l'activité de l'entreprise, il y a 11 valeurs possibles.
- `actionnaire` (chaînes de caractères, catégoriel) : Le type d'actionnariat de l'entreprise (le cas échéant). Quatre types d'actionnariat sont distingués : "pas d'actionnaire", lorsque il n'y a pas d'actionnaire, "personne physique" et "famille" pour des uniques actionnaires le cas échéant, et "entreprise", qui couvre les autres cas.
- `forme_jur_simpl` : (chaînes de caractères, catégoriel) : Décrit la forme juridique de l'entreprise. Il y a 9 valeurs possibles, sachant que 3 types apparaissant moins de 100

fois. Il y a par exemple 48072 SARL et 22 Société Civile.
- `chgt_dir` : (booléen avec 1/3 de valeurs manquantes) : Est-ce que la direction de l'entreprise a changé récemment. True si l'entreprise a changé de direction.
- `rdv` (booléen) : Est-ce que l'entreprise concernée a répondu positivement (true) ou négativement (false) à la demande de rendez-vous après prospection. Dans cette liste, toutes les entreprises ont reçu une invitation à un rendez-vous.

2.2 Nettoyage

2.2.1 Encodage

Premièrement, le fichier de données étant encodé en `ISO-8859-1` plutôt qu'en UTF-8, nous nous sommes assurés que la conversion vers ce dernier ait lieu puisque Python 3 traite toutes les chaînes de caractères en encodage UTF-8. Omettre cette conversion aurait pour conséquence une mauvaise gestion de caractères spéciaux (tels que les accents), si ce n'est une corruption des données (insertion de "`,`" dans le CSV par exemple).

2.2.2 Entreprises figurant plusieurs fois dans le jeu de données

Les entreprises peuvent figurer plusieurs fois dans le jeu de données. Ainsi, on observe beaucoup d'instances avec des valeurs extrêmement similaires (âge, évolution de l'âge, des scores financiers, etc.). Quelques dizaines d'instances sont des dupliqués avec des valeurs strictement identiques. Étant donné que le fichier ne fournit pas d'identifiant unique pour chacune des entreprises qui auraient été sondées plusieurs fois, il nous est difficile d'estimer un nombre exact, mais nous estimons le nombre d'instances concernées à plusieurs milliers. Ainsi, détecter de manière fiable les instances dupliquées paraît peu viable, d'autant plus qu'il faudrait décider de quelles instances conserver dans le cadre de l'analyse.

Il convient de noter que parmi les entreprises manifestement dupliquées, la valeur de `rdv` est soit tout à 0 (rdv refusé), soit tout à 1 (rdv accepté), et moins couramment avec certaines à 0 et certaines à 1.

Nous supposons ainsi que la multiple présence de mêmes entreprises a été voulue lors de la construction du jeu de données.

Nous pensons ainsi que ce phénomène ne pose pas un problème majeur à l'analyse qui justifierait un traitement spécifique. En revanche, il convient de prendre en compte, par exemple, le risque accru de sur-apprentissage pour les méthodes de classement, ou encore lors de la présence de valeurs aberrantes notamment lors du clustering.

2.2.3 Valeurs manquantes

Dans les données, il y a des valeurs manquantes pour certains attributs comme `chgt_dir` ou `ca_export_FK` dans des proportions très variables d'un attribut à l'autre.

Nous parlerons de "traitement spécifique" lorsque nous effectuerons un traitement sur une variable pour palier de tels problèmes, ce que nous détaillerons dans 2.3.

La variable `chgt_dir` a par exemple un tiers de valeurs manquantes, ce qui selon nous est une information, car cela veut sûrement dire qu'il a été impossible de répondre à cette question. Cela peut être dû au type d'entreprise qui fait qu'il n'y a pas de direction, qu'il s'agit d'une information confidentielle. Nous avons donc créé une nouvelle valeur possible pour cette variable (traitement spécifique).

Dans les données, il y a aussi les attributs suivants avec des valeurs manquantes :

- `ca_export_FK` : 6477 valeurs manquantes. On peut supposer que les lignes avec des valeurs manquantes dans cet attribut correspondent à des entreprises sans CA à l'export. Il est donc envisageable de remplacer ces valeurs manquantes par des 0 (traitement spécifique).
- `risque` : 936 valeurs manquantes. Nous n'avons aucune hypothèses concertant les valeurs manquantes de cet attribut.
- `evo_risque` : 1538 valeurs manquantes. Nous n'avons aucune hypothèses concertant les valeurs manquantes de cet attribut.
- `type_com` : 1079 valeurs manquantes. Nous avons remplacé ces valeurs manquantes par la classe majoritaire.

Pour le traitement des valeurs manquantes pour les attributs n'ayant pas de traitement spécifique, se référait à la section 2.3.

2.2.4 Valeurs aberrantes

Certains attributs ont des valeurs aberrantes, telles que `endettement` ou `ratio_benef`. Cela signifie que certaines entreprises sont enregistrées dans le jeu de données avec des valeurs vraisemblablement incohérentes ou impossibles. Dans ces cas-là, plusieurs options de nettoyage se profilent, qui vont de la suppression de l'entreprise au remplacement des valeurs aberrantes.

La variable `ratio_benef` a plus de 100 valeurs au-dessus de 100 et plus de 8 000 en dessous de 0. Le `ratio_benef` étant un pourcentage des bénéfices sur le CA, il est impossible d'avoir un `ratio_benef` supérieur à 100, car cela voudrait dire avoir plus de bénéfices que de chiffre d'affaires. De même, il est improbable d'avoir ce pourcentage très en dessous de 0, car l'entreprise ferait rapidement faillite. On a donc remplacé les lignes en dessous de -100 (57 valeurs) et au-dessus de 100 (139 valeurs) par NaN.

Nous avons fait de même pour la variable `endettement`. Nous avons remplacé par NaN tout ce qui est en dessous de -1 (65 valeurs). Le ratio d'autonomie financière devrait se situer proche de 1. Cependant, nous supposons qu'ici cela doit se situer proche de 0 au vu du grand nombre de données entre -1 et 0.

Pour la variable `ca_export_FK`, nous avons remplacé par NaN tout ce qui est en dessous de 0. Il y a 205 valeurs en dessous de 0. Le CA ne peut pas être négatif en théorie (sauf pour une entreprise sur le point de fermer), il est donc aberrant que la part de CA à l'export soit négative. Le `ca_total_FL` a lui seulement 5 entreprises avec un CA inférieur à 0, ce qui paraît déjà plus cohérent sachant que toutes les entreprises ne font pas d'export.

2.3 Traitement

Nous avons ensuite regardé le nombre de NaN et de valeurs manquantes (NA) (seulement les valeurs manquantes n'appartenant pas à une variable ayant un traitement spécifique, c.-à-d. une valeur manquante dans `chgt_dir` ou `ca_export_FK` ne compte pas dans ce calcul) par ligne. Si le nombre cumulé de NaN et NA est égal ou supérieur à 2, alors nous avons retiré la ligne car cela veut dire que sur la ligne, il y a un problème au moins sur 2 variables. Cela a retiré 139 lignes.

Pour les variables numériques, nous avons remplacé les valeurs manquantes et NaN générés par les précédentes fonctions par la médiane. Pour les variables ordinales, nous les avons remplacés par la valeur la plus fréquente. Et enfin pour les catégorielles, nous avons remplacé par la classe majoritaire. Nous avons ensuite standardisé les variables numériques en utilisant le "StandardScaler" de sklearn. Ce dernier recentre les valeurs sur 0 (moyenne) avec un écart-type de 1.

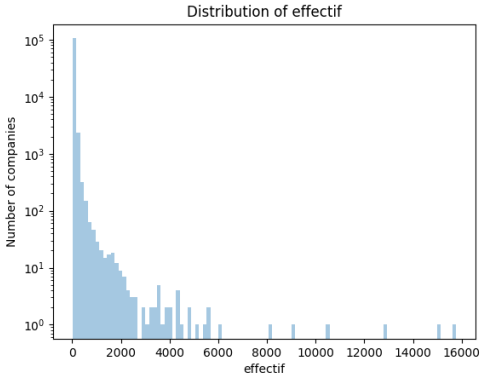
2.4 Analyse descriptive

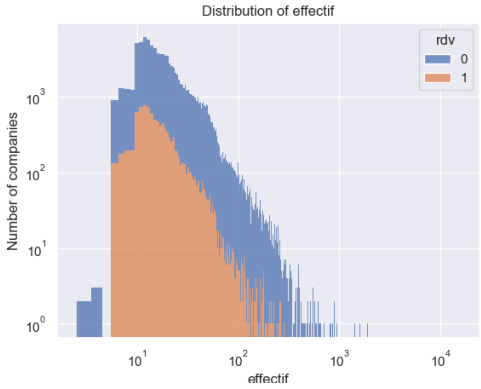
2.4.1 Effectif

- **min.** 3
- **max.** 15783
- **moy.** 39.03
- **med.** 21

La distribution des effectifs montre que le jeu de données prend aussi bien en compte des très petites entreprises (TPE) que des très grandes entreprises (TGE). La médiane se situe dans la catégorie des petites ou moyennes entreprises (PME) ce qui n'est pas anormal.

On observe pour le TPE un taux de rendez-vous proche de 15% et qui diminuera à 10% pour les PME, 5% pour les ETI et enfin aucun rendez-vous pour les TGE (qui sont au nombre de 11).





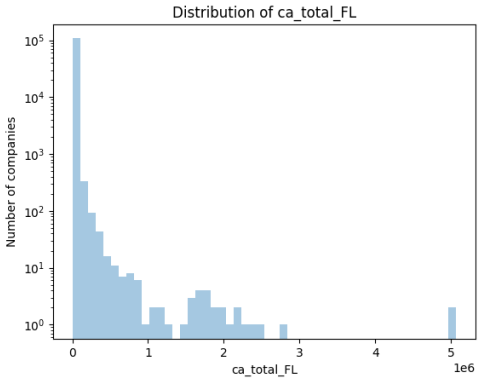
(a) Avec proportion de rendez-vous

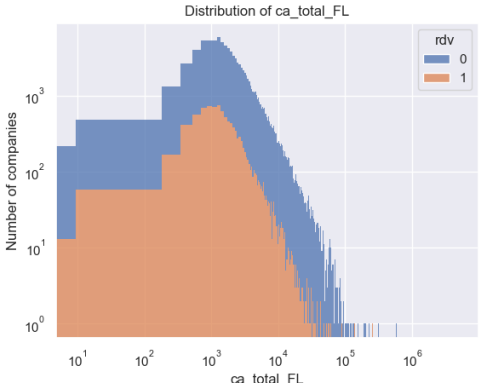
Figure 1. Distribution de l'effectif

2.4.2 Chiffre d'affaires total

- **min.** -162
- **max.** 5071221
- **moy.** 7072
- **med.** 2384

La distribution des chiffres d'affaires totaux montre une répartition assez similaire à l'effectif et donc les différentes tailles d'entreprises présentes dans le jeu de données.





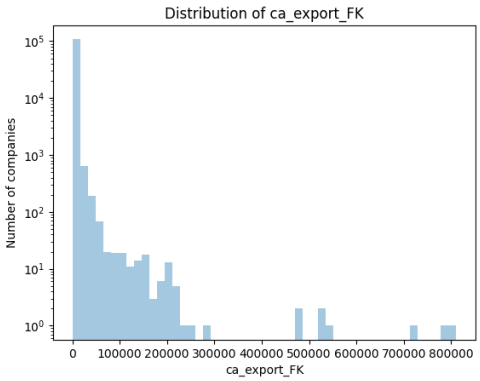
(a) Avec proportion de rendez-vous

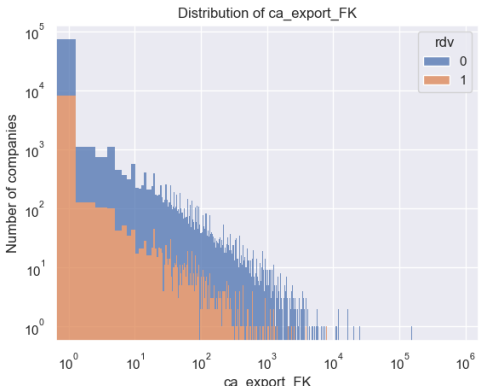
Figure 2. Distribution des chiffres d'affaires total

2.4.3 Chiffre d'affaires à l'exportation

- **min.** 0
- **max.** 811020
- **moy.** 815.14
- **med.** 1

Tout comme pour le chiffre d'affaire total on retrouve une distribution prenant une forme similaire à celui de l'effectif. A la différence que quasiment un quart des entreprises de semble pas faire d'exportation à l'étrange et ont donc une chiffre d'affaire à l'exportation de 0.





(a) Avec proportion de rendez-vous

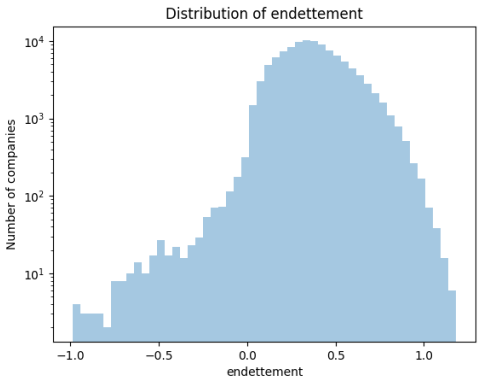
Figure 3. Distribution des chiffres d'affaires à l'exportation

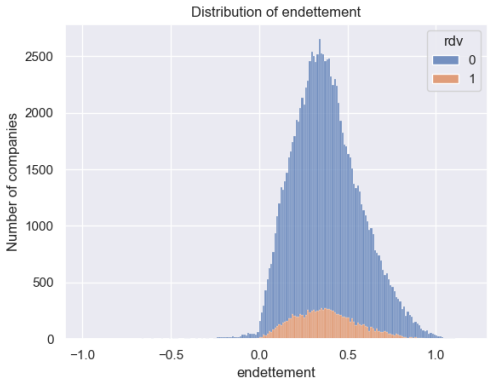
2.4.4 Endettement

- **min.** -0.9898
- **max.** 1.183
- **moy.** 0.3808
- **med.** 0.3655

L'endettement (qui pour rappel correspond au ratio d'autonomie financière) suit une

distribution de loi normal autour de la valeur de 0.38. La proportion de rendez-vous tourne autour des 10% quelque soit le taux.





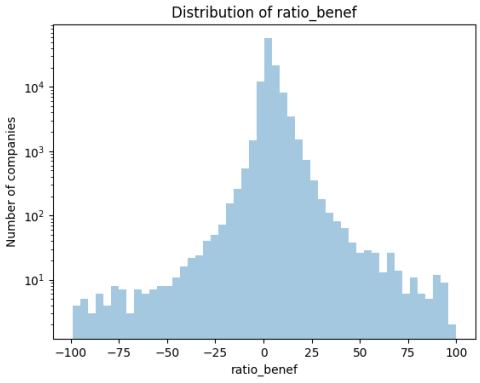
(a) Avec proportion de rendez-vous

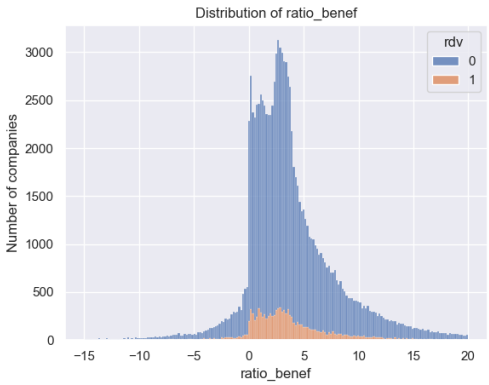
Figure 4. Distribution des taux d'endettement

2.4.5 Ratio bénéfice

- **min.** -99.15
- **max.** 100
- **moy.** 4.11
- **med.** 3.11

Une grande majorité des entreprises enregistre un ratio bénéfice entre 0 et 5.





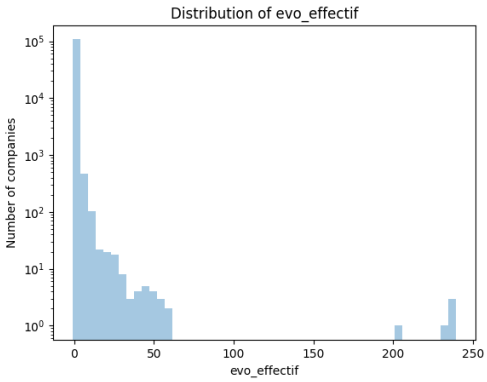
(a) Avec proportion de rendez-vous

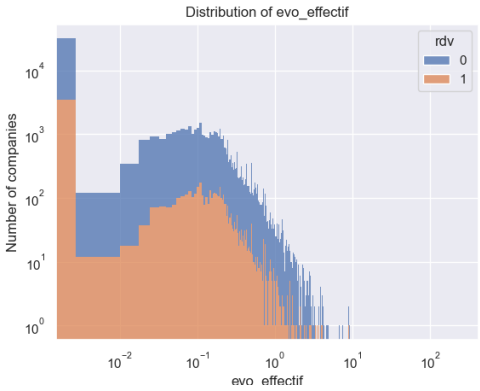
Figure 5. Distribution des ratios de bénéfice

2.4.6 Évolution de l'effectif

- **min.** -1.06
- **max.** 239.6
- **moy.** 0.18
- **med.** 0

L'évolution de l'effectif représente la croissance en terme de main d'oeuvre des entreprises. On observe quasiment un quart des entreprise avec une décroissance de leurs effectifs. 29% des entreprises n'ont eu aucune évolution de leurs effectifs et 30% des entreprises ont enregistré une évolution entre 0 et 25%. Le taux de rendez-vous est à environ 10% .





(a) Avec proportion de rendez-vous

Figure 6. Distribution des taux d'endettement

2.4.7 Risque

- **min.** 0
- **max.** 4
- **moy.** 3.11
- **med.** 3

Le risque représente le score de santé financière de l'entreprise. On peut observer que quasiment la moitié des entreprises possède le très bon score de 14. et qu'une petite minorité a un score de 7 ou moins.

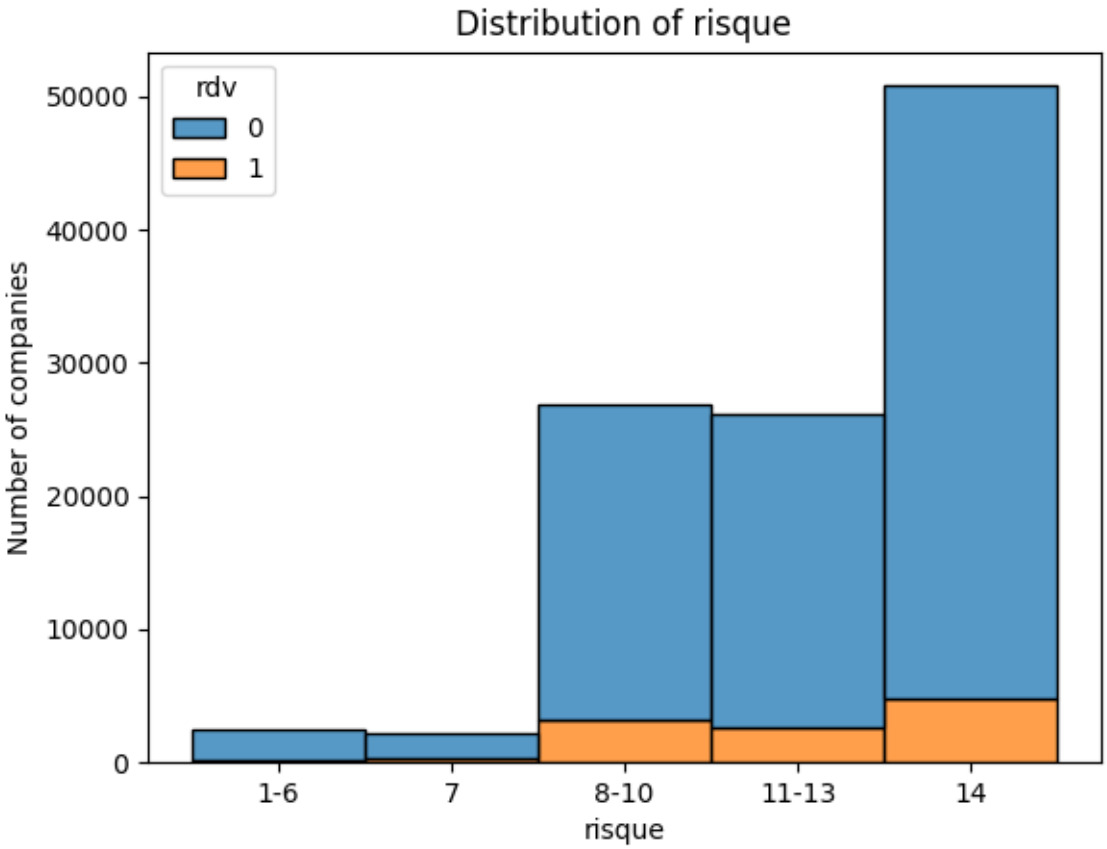
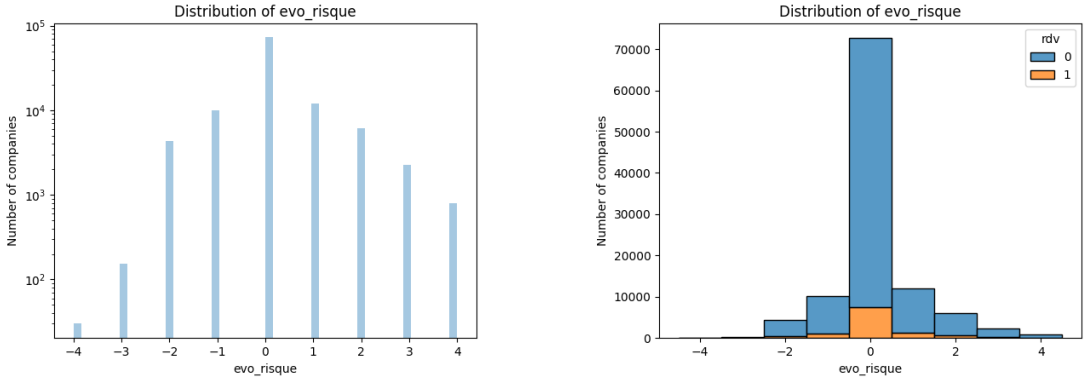


Figure 7. Distribution de l'évolution du risque

2.4.8 Évolution du risque

- **min.** -4
- **max.** 4
- **moy.** 0.14
- **med.** 0

L'évolution du risque représente une mesure de la variation du score de santé mentale au fil du temps. On observe que la grande majorité des entreprises ont une situation stable et que sur les entreprises restante on retrouve plus d'évolution positive que négative. La proportion de rendez-vous ne varie pas selon cette évolution est reste aux alentours de 10



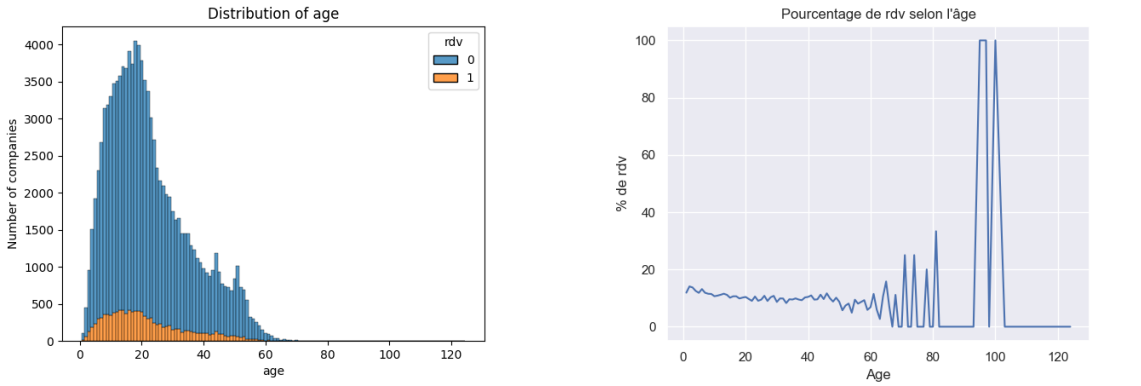
(a) Distribution avec mise à l'échelle (b) Avec proportion de rendez-vous

Figure 8. Distribution de l'évolution du risque

2.4.9 Age

- **min.** 1
- **max.** 124
- **moy.** 22.64
- **med.** 20

Nous observons des entreprises de tous les âges allant de 1 à 124 ans avec une majorités entre 10 et 20 ans. On peut observer que le taux de rendez-vous pour les entreprises très jeunes est d'environ 15% et qu'il se stabilise ensuite aux alentours de 10%.



(a) Distribution avec proportion de rendez-vous (b) Pourcentage de rendez-vous selon l'âge

Figure 9. Distribution de l'âge

2.4.10 Changement de direction

- **min.** 0
- **max.** 1
- **moy.** 0.176
- **med.** 0

2.4.11 Type com

Pour rappel, l'attribut type commune est un attribut catégoriel avec 4 valeurs possibles et 1079 valeurs manquantes. Les valeurs ne sont pas équitablement réparties comme on peut le voir sur 10. 14% des entreprises avec espace

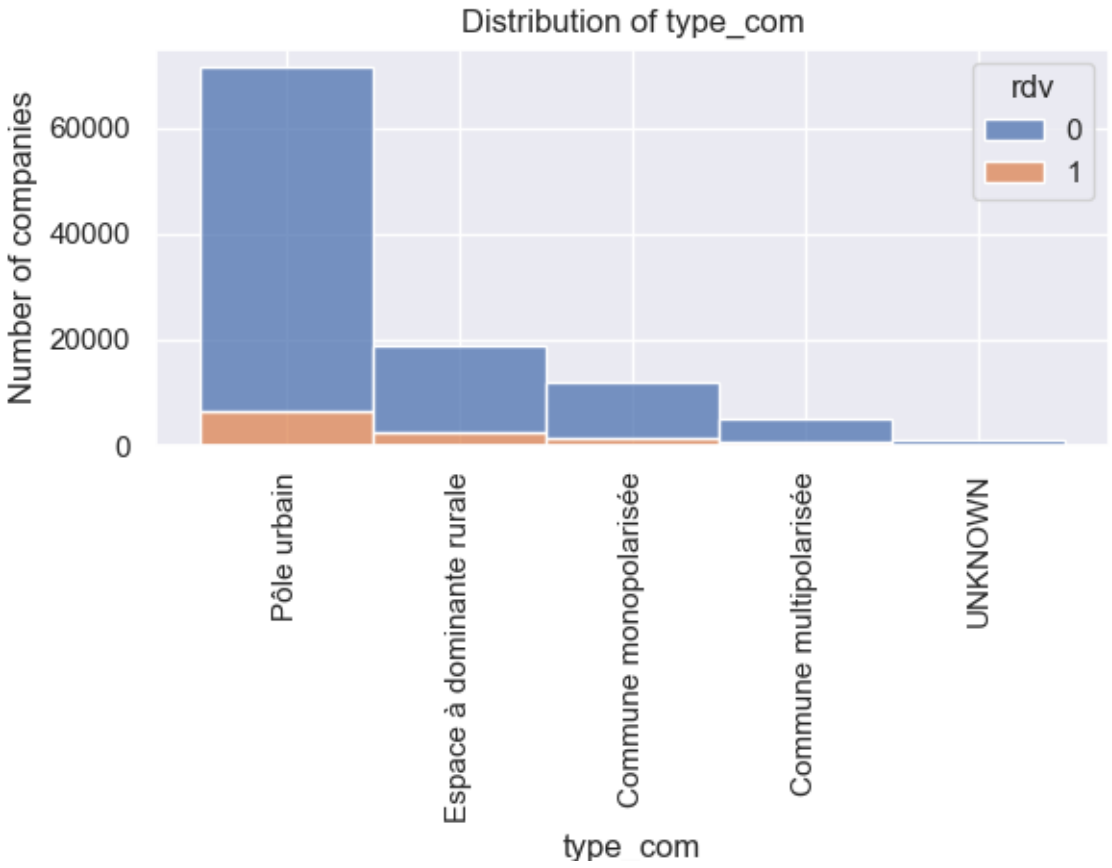


Figure 10. Distribution de l'attribut Type_com

2.4.12 Activité

Pour rappel, l'attribut activité est un attribut catégoriel avec 11 valeurs possibles et 0 valeurs manquantes. Les valeurs ne sont pas équitablement réparties comme on peut le voir sur 11. Au niveau des proportions de rendez-vous elles sont toutes à environ 10% sauf les deux activités dans le domaine de la construction : Travaux divers. Menuiserie. Miroiterie à 14% et Autres à 12%

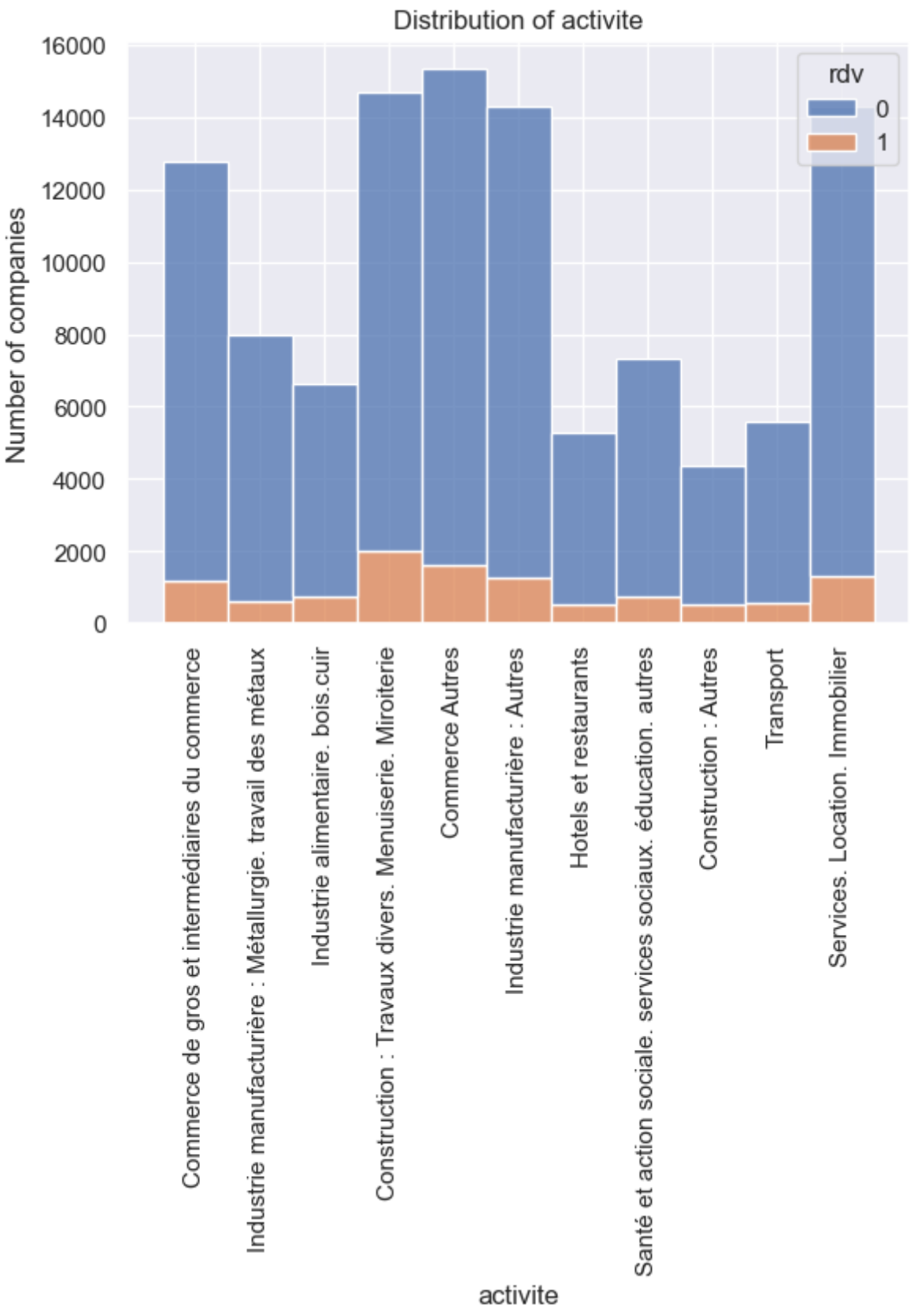


Figure 11. Distribution de l'attribut Activité

2.4.13 Forme jur simpl

Pour rappel, l'attribut forme juridique simplifié est un attribut catégoriel avec 6 valeurs possibles et 0 valeurs manquantes. Les valeurs ne sont pas équitablement réparties comme on peut le voir sur 13. La proportion de rendez-vous selon la forme juridique varie :

- Affaire personnelle : 15%
- Autres : 11%
- Coop. Ou union agricole : 28%
- Forme juridique agricole : 16%
- SARL : 12%
- Societe anonyme : 8%
- Société civile : 13%
- Société en nom collectif : 5%
- Société par actions simplifiée : 7%

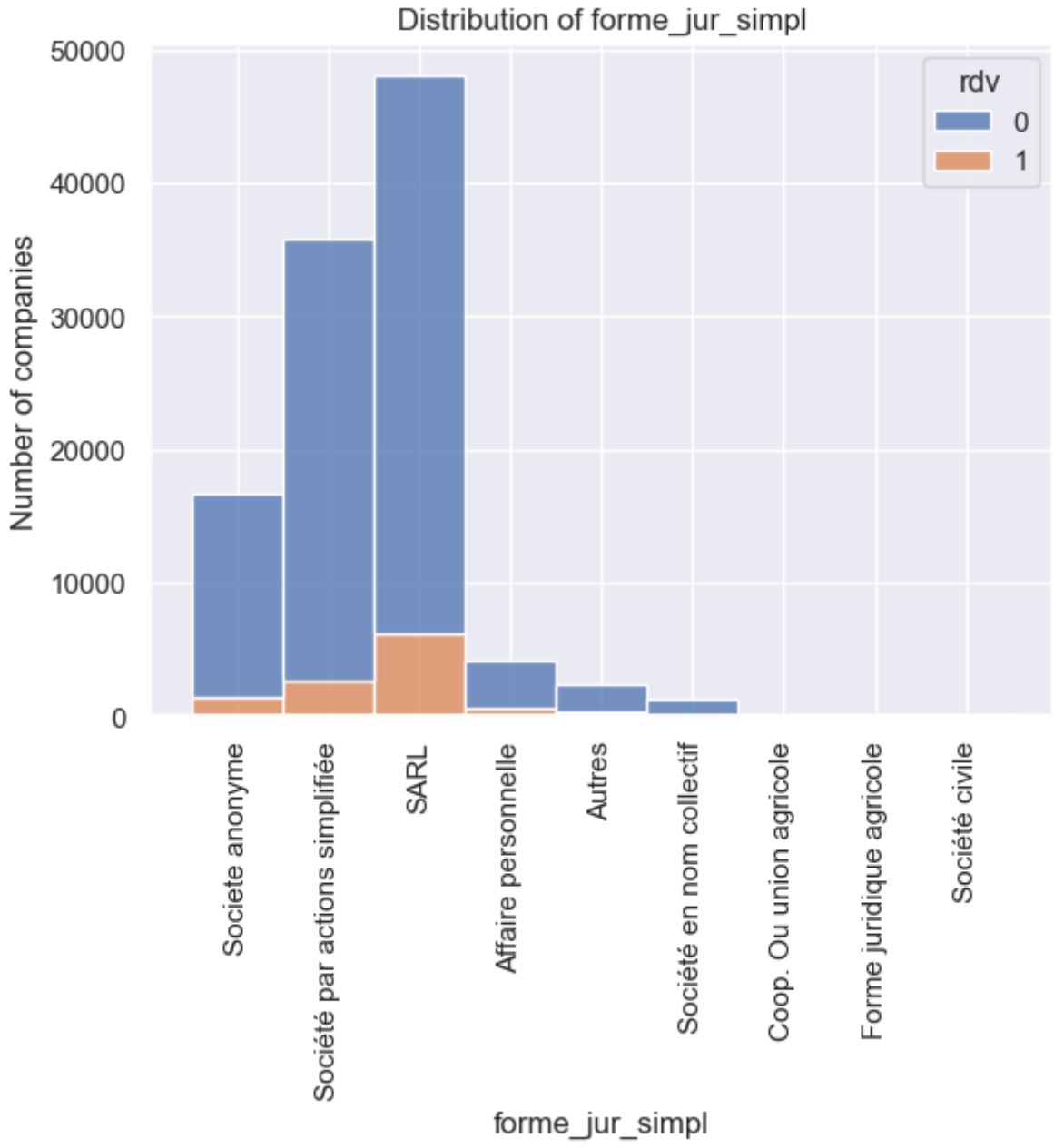


Figure 12. Distribution de l'attribut Forme_jur_simpl

2.4.14 Actionnaire

L'attribut actionnaire est un attribut catégoriel avec 4 valeurs possibles et 0 valeurs manquantes. Les valeurs ne sont pas équitablement réparties comme on peut le voir sur 12. La proportion de rendez-vous selon l'actionnaire varie :

- Personne physique et pas d'actionnaire : 12%
- Famille : 10%
- Entreprise : 7%

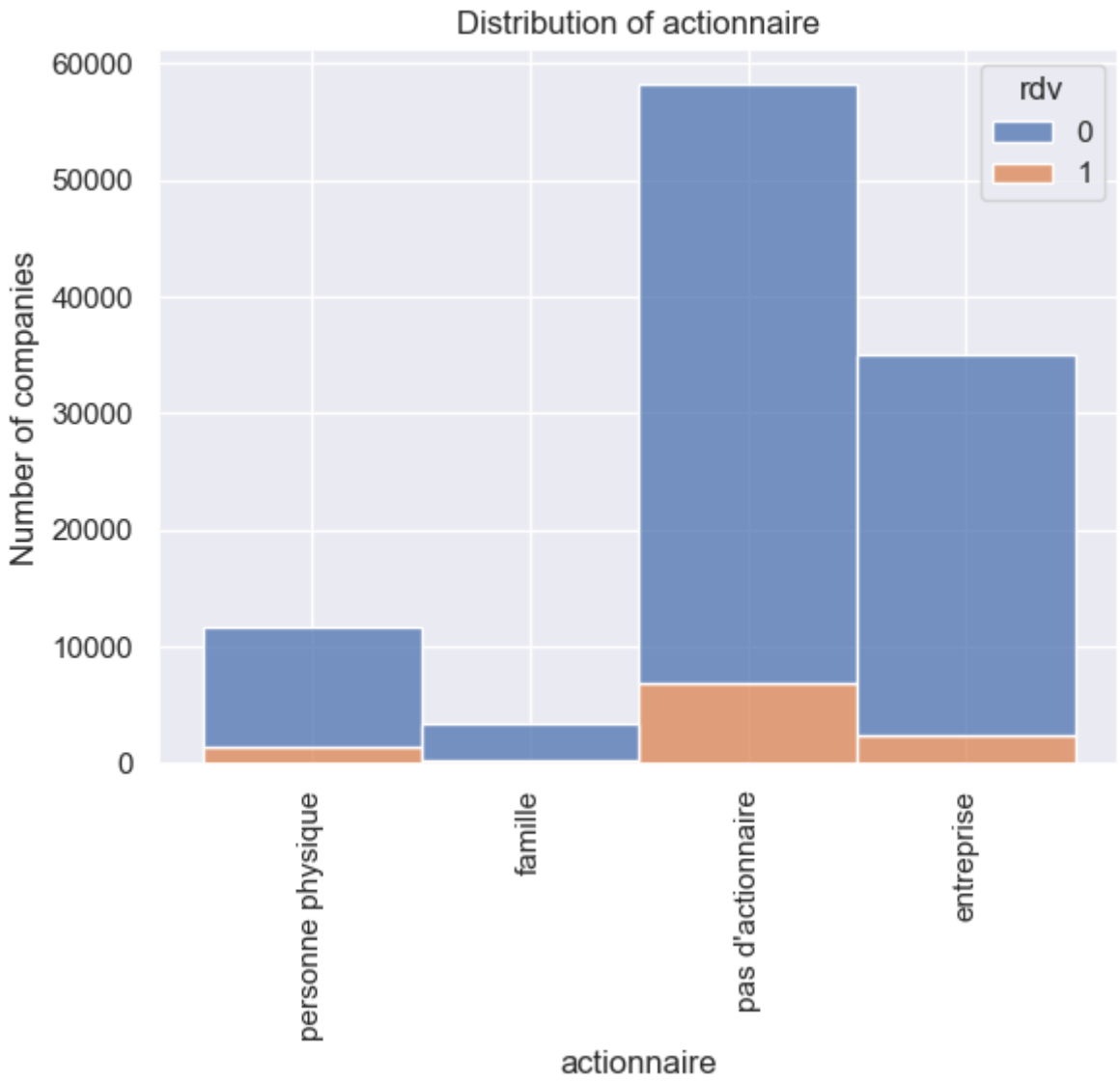


Figure 13. Distribution de l'attribut Actionnaire

2.4.15 Département

L'attribut département est un attribut catégoriel avec 88 valeurs possibles et 0 valeurs manquantes. Les valeurs ne sont pas équitablement réparties comme on peut le voir sur 14.

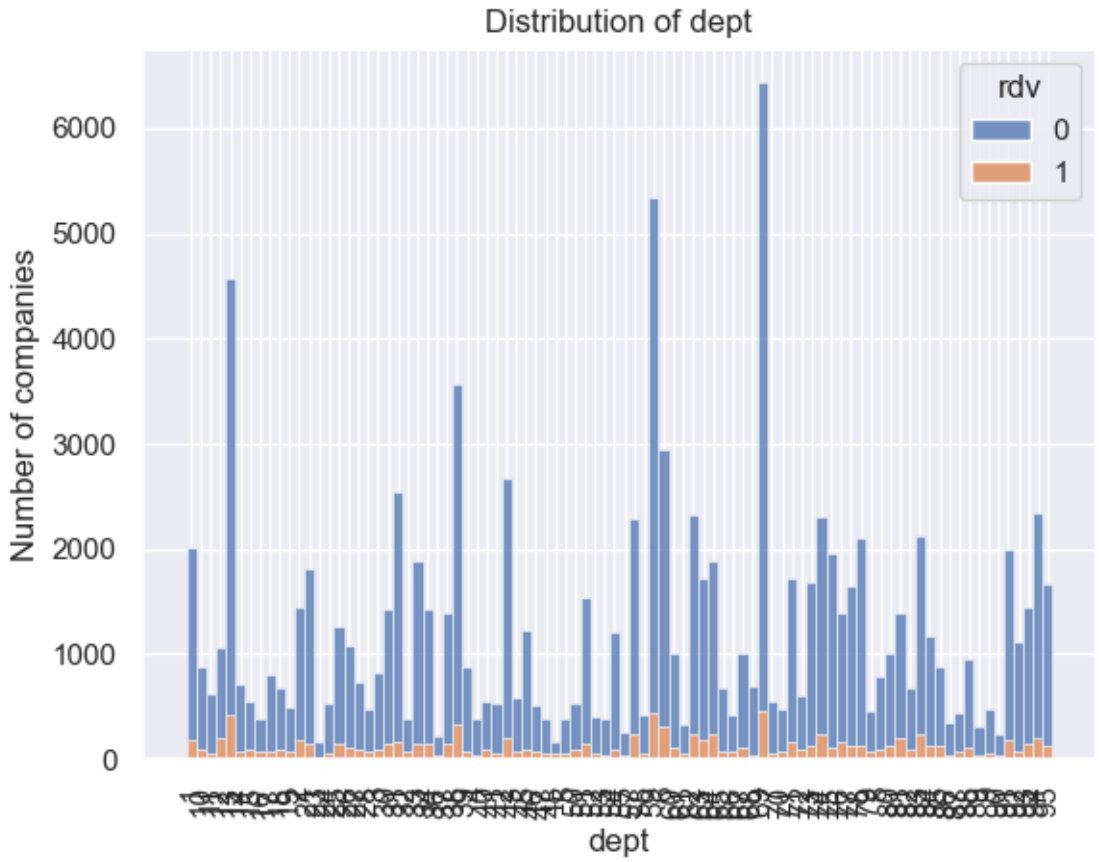


Figure 14. Distribution de l'attribut Département

3 Méthodes

3.1 Outils de fouille

3.1.1 Clustering

Nous avons décidé d'appliquer trois algorithmes de clustering : Kmeans, le clustering mixte. Nous avons décidé de ne pas utiliser la classification ascendante hiérarchique (abrégé en C.A.H) directement dû au trop grand nombre de données. Néanmoins, le clustering mixte utilise le C.A.H. En effet, le custering mixte consiste en l'application de kmeans dans le but de réduire le nombre de données sur lequel la C.A.H est effectué. Le Kmeans est une méthode gloutonne et n'a donc pas de problème à gérer la quantité de données présente ici. En ce qui concerne les règles d'associations, elles pourraient permettre de trouver des combinaisons de variables fréquentes et ayant un résultat intéressant.

Kmeans

Nous avons utilisé `sklearn.cluster.KMeans`. Cette classe a 9 arguments :

- `n_clusters` : int (default=8). Définit le nombre de clusters qui va être utilisé
- `init` : 'k-means++', 'random', callable or array-like of shape (n_clusters, n_features), default='k-means++'). Définit le type d'initialisation (k-means++ ou random) ou un array contenant les centroids ou une fonction permettant de définir les centroids au départ.
- `n_init` : 'auto' or int (default=10). Définit le nombre de fois que l'algorithme de k-means sera exécuté, c'est utile car kmeans est un algorithme glouton, les résultats varient donc d'une exécution à une autre. 'Auto' correspond à 10 si `init='random'`, 1 si `init='k-means++'`. Le résultat final sera le meilleur résultat obtenu, le meilleur étant celui où les centroids ont le moins bougé.
- `max_iter` : int (default=300). Définit le nombre maximum d'itérations de l'algorithme de k-means pour une seule exécution. Permet de réduire le temps d'exécutions.
- `tol` : float (default=1e-4). Définit le seuil à partir duquel l'algorithme s'arrête. Si la différence entre deux itérations est inférieure à `tol`, l'algorithme s'arrête.
- `verbose` : int (default=0). Définit le niveau de verbosité de l'algorithme, c.-à-d. à quel point il va afficher des informations à l'écran.
- `random_state` : int (default=None). Définit une seed pour contrôler la génération de nombres aléatoires pour l'initialisation des centres de cluster. Cela permet de rendre l'algorithme déterministe.
- `copy_x` : bool (default=True). Définit si l'algorithme modifiera les données d'entrées. Si vrai, une copie des données `X` sera effectuée.
- `algorithm` : "lloyd", "elkan", "auto", "full" (default="lloyd"). Définit l'algorithme de k-means à utiliser, "lloyd" correspond à l'algorithme de base.

Afin d'essayer d'obtenir les meilleurs clusters possibles, nous avons fait varier le nombre de clusters k. Nous avons également fait varier le nombre d'initialisations `n_init` car une valeur élevée permet d'avoir des résultats consistants, mais cela se fait au détriment du temps d'exécution. Nous avons également testé les différentes méthodes d'initialisation `init` pour voir laquelle donnait de meilleurs résultats. Enfin, nous avons fixé le `random_state` pour avoir des résultats déterministes et donc comparables d'une exécution à une autre.

Clustering mixte

Pour le clustering mixte, nous avons utilisé k-means que nous avons vu juste au-dessus et le clustering hiérarchique ascendant.

Pour réaliser le C.A.H, nous avons utilisé `scipy.cluster.hierarchy.dendrogram` pour le rendu du dendrogramme et `scipy.cluster.hierarchy.linkage` pour calculer le clustering

hiérarchique ascendant. Les paramètres de ce dernier nous intéressent et sont :

- `y`. Soit une matrice en 2 dimensions correspondant aux données, soit un une matrice des distances.
- `method` : str. La méthode de liaison (ou de regroupement) utilisée pour calculer les distances entre les groupes d'observations. Les valeurs possibles sont "single", "complete", "average", "weighted", "centroid", "median" et "ward".
- `metric` : str or function. La mesure de distance utilisée pour calculer les distances entre les paires d'observations. Les valeurs possibles sont "euclidean", "cityblock" (distance de Manhattan), "cosine", "correlation" et bien d'autres (voir la documentation pour la liste complète).
- `optimal_ordering` : bool (default=False). Si vrai, l'ordre optimal des feuilles de l'arbre sera déterminé pour améliorer la lisibilité de la visualisation de l'arbre. Par défaut à False car cela ralentit l'algorithme.

Nous avons fait varier la `method` utilisée. Les résultats changent beaucoup d'une méthode à une autre : En effet, le mode de calcul des distances influe considérablement sur le résultat.

3.1.2 Classification supervisée

Arbres de décision

Les arbres de décision fonctionnent par un partitionnement de l'espace des variables qui est réalisée de manière itérative. Les arbres de décision présentent certains avantages considérables dans notre cas.
D'une part, ils sont très directement interprétables, car les conditions sont triviales : à chaque étape, il ne s'agit que d'une comparaison pour une unique dimension. Au final, il est possible d'obtenir un arbre relativement concis dans une tâche de prédiction, qui explicite la distribution des données dans chaque branche de l'arbre.
D'autre part, ils sont particulièrement paramétrables. Il est ainsi assez facile d'obtenir un arbre relativement petit lorsqu'on veut privilégier l'interprétablité à la précision du modèle.

Certaines méthodes similaires auraient pu être appliquées, tel que la méthode des forêts aléatoires. Cependant, celle-ci étant nettement moins interprétable et ayant plus vocation à améliorer la précision du modèle, nous n'en avons pas fait usage.

Pour ce qui est de l'implémentation, nous avons utilisé `sklearn.tree.DecisionTreeClassifier`. Le constructeur a 12 arguments :

- `criterion` : parmi `gini`/`entropy`/`log_loss`. Il s'agit du critère que l'algorithme de construction de l'arbre va chercher à maximiser. Par défaut, et c'est ce que l'on va utiliser, il s'agit du gini, qui est le critère de pureté.
- `splitter` : parmi `random`/`best`. En mode `best`, lors d'une division de noeud, l'algorithme va sélectionner la variable avec la plus grande importance (au vu du critère) pour réaliser la comparaison. En mode `random`, l'algorithme va sélectionner une variable aléatoirement, mais avec une distribution qui suit l'importance du critère. Par défaut, `best` est utilisé.
- `max_depth` : int. Il s'agit de la profondeur maximale de l'arbre, c'est-à-dire du nombre maximum de noeuds que l'on peut parcourir avant d'atteindre une feuille. Par défaut, il n'y en a pas.
- `min_samples_split` : int. Empêche la division d'un noeud de l'arbre lorsque le nombre d'instances couvertes par le noeud en question est inférieur à cette valeur.
- `min_samples_leaf` : int. Empêche la création de feuilles dans l'arbre lorsque le nombre d'instances qu'elles couvreraient serait inférieur à cette valeur.
- `min_weight_fraction_leaf` : float. Empêche la création de feuilles dans l'arbre lorsque

le nombre d'instances qu'elles couvreraient serait inférieur à cette valeur, en termes de pourcentage des instances, et en prenant en compte la pondération des classes le cas échéant.

- `max_features` : Permet de définir un nombre maximum de variables à prendre en compte lors de la division d'un noeud, qui seront choisies aléatoirement. Utile lorsque le nombre de variables est très élevé pour réduire le nombre de variables à évaluer par itération. Par défaut, toutes les variables sont toujours prises en compte.
- `random_state` : int. Correspond à la graine initiale du générateur de nombres pseudo-aléatoires utilisé par l'algorithme de construction de l'arbre.
- `max_leaf_nodes` : int. Empêche la création de feuilles supplémentaires lorsque ce nombre de feuilles est atteint.
- `min_impurity_decrease` : float. Empêche la création d'un noeud si l'amélioration de la pureté est inférieure à ce seuil. Par défaut, ce critère est ignoré (0.0).
- `class_weight` : dictionnaire classe-poids, "balanced" ou aucune pondération (poids de 1) : Permet d'assigner des poids aux différentes classes à prédire. Pondérer des classes minoritaires peut forcer l'arbre construit à prédire davantage celle-ci.
- `ccp_alpha` : float. Durant la construction du graphe, plus ce seuil est élevé, plus l'algorithme de construction va agressivement élaguer des branches qui ont beaucoup de noeuds mais qui contribuent peu à l'amélioration de la pureté. L'utilisation de cette valeur peut être utile pour générer un arbre moins complexe sans trop en sacrifier la précision de classification.

Nous avons tenté d'appliquer cette méthode à la prédiction de différentes variables, afin de détecter la présence de tendances, mais également sur une tâche plus complexe qui est la prise de rendez-vous.

Dans l'espoir d'obtenir un arbre relativement simple à interpréter, mais dans l'espoir de ne pas trop en sacrifier les performances, nous avons tenté de jouer avec les paramètres `min_samples_leaf`, `max_leaf_nodes` et `ccp_alpha`.

Pour `ccp_alpha`, nous avons remarqué qu'il y a besoin d'un réglage très fin, car il dépend beaucoup de notre objectif mais également de la variable à prédire. Cela s'explique certainement par la difficulté de la tâche de prédiction, puisque cet élagage prend en compte un critère de pureté.

3.2 Recodage & Rééquilibrage

3.2.1 Clustering

Kmeans et clustering mixte

L'algorithme du Kmeans a besoin de valeurs numériques uniquement, il est donc nécessaire de transformer les variables catégorielles en variables numériques. Pour cela, il existe plusieurs possibilités, nous avons choisi d'utiliser des vecteurs de type "one hot". Cela signifie que cela, pour chaque valeur possible de chaque variable catégorielle, une nouvelle variable numérique sera créée. Cela peut noyer les variables numériques, car il y aura donc beaucoup plus de variables catégorielles que de variables numériques dans les données transformées. Aucun équilibrage n'a été effectué.

3.2.2 Classification supervisée

Arbres de décision

Les arbres de décision peuvent trivialement opérer sur des variables booléennes (dans la représentation graphique, il va s'agir des comparaisons $<= 0.5$. Par simplicité, nous avons transformé les variables catégorielles en variables numériques avec des vecteurs one-hot comme pour le clustering.

Pour ce qui est de l'équilibrage des données, nous avons remarqué que les méthodes de prédiction atteignaient très rarement des résultats convaincants pour la prédiction de l'attribut `rdv`. Nous associons ceci au fait que la tâche semble significativement complexe et que, sans rééquilibrage des classes, la plupart des méthodes se limitent à prédire la classe majoritaire (à savoir pas de rendez-vous).

Pour les arbres de décision, nous avons ainsi pondéré les classes afin que `rdv` et `non-rdv` soient égales au niveau du classifieur.

3.3 Évaluation

3.3.1 Clustering

Kmeans

Nous avons utilisé la métrique du $\mathtt{r}^2$ et le `silhouette score` sur un sous ensemble correspondant à 10% du dataset de départ pour déterminer le nombre de cluster à utiliser dans le `kmeans`. Nous avons utilisé 10% des données, car le $\mathtt{r}^2$ a une complexité de $O(n^2)$ où n est le nombre de données. Ces métriques permettent d'avoir un premier avis sur la quantité de cluster à avoir. Cependant, l'évaluation se fait majoritairement à la main en analysant les résultats pour voir si des tendances intéressantes s'en dégagent.

Clustering mixte

Pour le clustering mixte, on peut regarder la distance entre les différents merge dans le dendogram. Si la distance est très élevée, cela veut dire que les deux clusters qui sont regroupés sont très éloignés. À l'inverse, si elle est petite, les deux clusters sont proches (similaires donc).

3.3.2 Classification supervisée

Arbres de décision

Étant donné que nous avons équilibré les classes, pour l'évaluation, nous nous sommes contentés de mesurer la précision lors de la prédiction sur une validation croisée 5-fold. Cependant, notre objectif dans l'absolu n'est pas de développer une manière fiable de prédire la prise de rendez-vous, mais de dégager des tendances qui permettraient de mieux cibler les entreprises qui acceptent les rendez-vous, et, inversement, d'élaguer les entreprises qui ont une très faible chance d'accepter le rendez-vous.

De plus, du fait de la présence des mêmes entreprises plusieurs fois dans le jeu de données, nous ne pouvons réellement nous fier à malgré la validation croisée, puisqu'il existe une forte dépendance entre les instances que nous ne pourrions éviter au vu du jeu de données actuel, et qui risquent de grandement accroître le risque de sur-apprentissage.

Nous avons ainsi préféré utiliser les arbres de décision comme un outil pour tirer de grandes tendances et dont nous limitions volontairement la taille pour conserver l'aspect interprétable.

3.4 Implémentation

Nous avons concentré la logique d'importation du jeu de données et de son prétraitement dans le fichier `dataloader.py`.

Différentes fonctions sont exposées et sa fonctionnalité est scindée en trois étapes :

- L'importation du dataset
- Le nettoyage des données
- La standardisation des données

Ce fichier énumère par ailleurs les différentes colonnes, les catégorisant par type de variable (nominale, ordinale, numérique).

Pour ces étapes d'importation, nous avons utilisé les fonctionnalités fournies par Pandas, NumPy et scikit-learn (pour l'imputage de valeurs et la standardisation notamment).

Analyse descriptive

Le notebook `nb-descriptive_graphs.ipynb` contient la génération des graphes utilisés dans la partie Analyse descriptive du rapport. Ces graphes s'appuient sur la librairie Seaborn. La méthode "describe" proposait par pandas sur le dataframe permet aussi d'obtenir des informations sur les statistiques des données (minimum, maximum, moyenne, médiane).

Kmeans

Le script `kmeans.py` contient tout le traitement et la génération de plots lié à Kmeans. Ce script utilise un autre fichier `kmeans_utilities.py` qui contient les fonctions liées à Kmeans.

- Chargement des données via le dataloader
- Génération des courbes silhouette et r^2
- Calcul de kmeans avec 3 clusters
- Affichage du résultat via une ACP
- Affichage du plot sur les rendez-vous
- Affichage des plots sur les régions
- Affichage des plots sur les départements
- Affichage de la position du centroid du premier cluster
- Affichage des variables caractérisant les clusters

Clustering mixte

Le script `clustering_mixte.py` contient tout le traitement et la génération de plots lié au clustering mixte (kmeans+clustering hiérarchique sur les clusters générés).

- Chargement des données via le dataloader
- Calcul de kmeans avec 25 clusters
- Affichage du dendogramme
- Unification des clusters (manuel, basé sur le dendogramme)
- Affichage du résultat via une ACP
- Affichage du plot sur les rendez-vous
- Affichage des plots sur les départements

Prédiction d'attributs

Le notebook `prediction-attribut.ipynb` regroupe les travaux qui ont été effectués pour la prédiction d'attributs, dont la génération de l'arbre de décision pour la prédiction de `rdv`. D'autres différentes expérimentations, certaines n'ayant pas été mentionnées dans le rapport, y figurent.

Quelques fonctions sont définies pour simplifier l'expérimentation sur des tâches de prédiction, permettant de tester des méthodes de classification en cross-validation pour la prédiction de classe avec n'importe quelle combinaison de colonnes souhaitées.

- Chargement des données via le dataloader
- Prédiction de la colonne `rdv` avec toutes les colonnes et différentes méthodes de classification, sans équilibrage de classes
- Comparaison de différents facteurs d'élagage pour les arbres de décision
- (hors rapport) Test de prédiction de la colonne "risque" en fonction de certains critères
- (hors rapport) Test de prédiction de la colonne "activite" en fonction des colonnes catégorielles
- Prédiction de `rdv` avec des arbres de décision, avec équilibrage des classes
- Présentation de l'arbre de décision

4 Résultats

4.1 Typologie des entreprises

4.1.1 Kmeans avec 3 clusters

Données utilisées

Kmeans a été appliqué sur toutes les variables numériques et ordinales. Nous avons essayé d'appliquer le kmeans sur toutes les variables en utilisant la transformation évoquée plus haut. Les résultats sont les mêmes à quelques instances près sur kmeans avec 3 clusters. Nous avons donc décidé d'utiliser seulement les données numériques et ordinales (pour le kmeans avec 3 clusters) pour le gain de temps d'exécution.

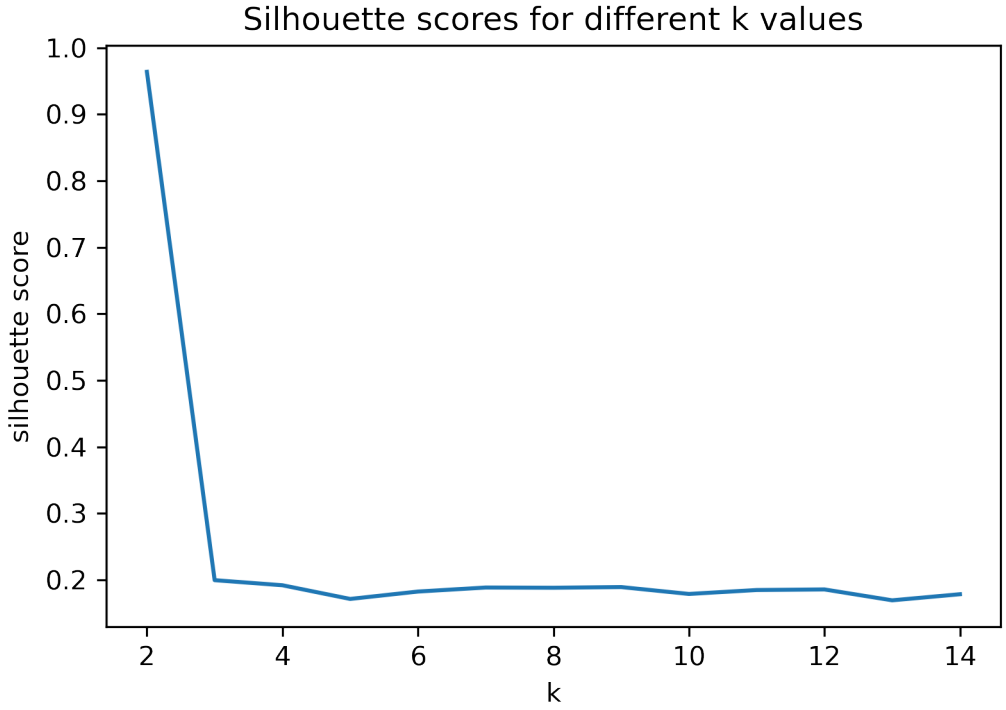


Figure 15. Courbe du score silhouette

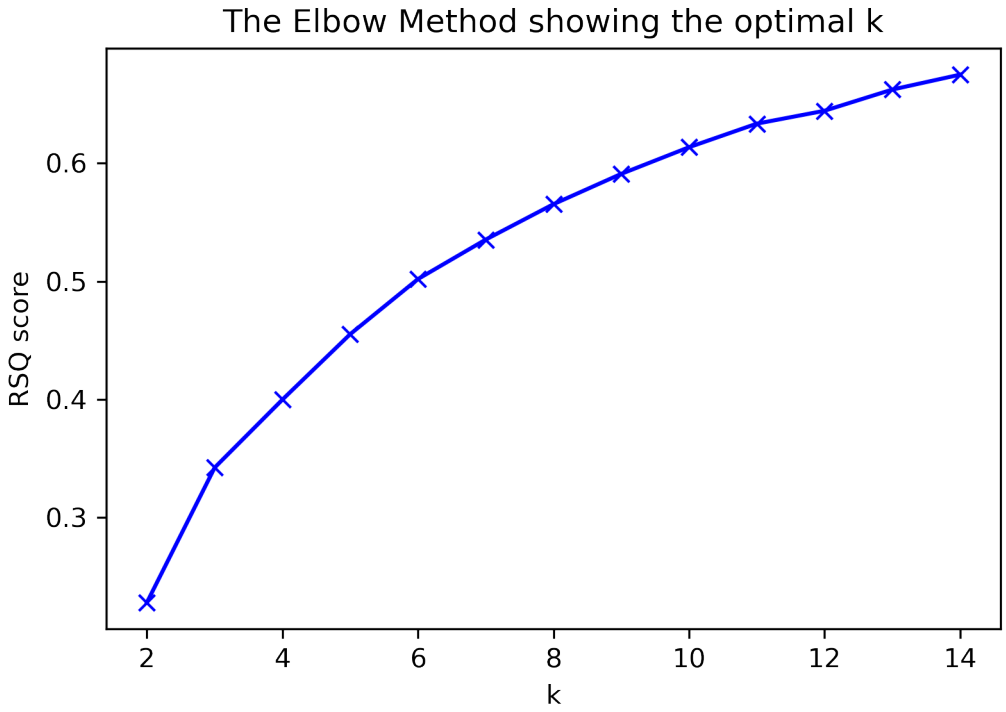


Figure 16. Courbe du R square

Métriques d'évaluation de la qualité

On peut voir sur 15 que selon ce score, le nombre idéal de cluster est 3 (grosse chute du score à k=3). Sur le 16 rien ne ressort vraiment, ici, on cherche le moment où la courbe s'aplatit. Selon le silhouette, il faudrait donc seulement 3 clusters.

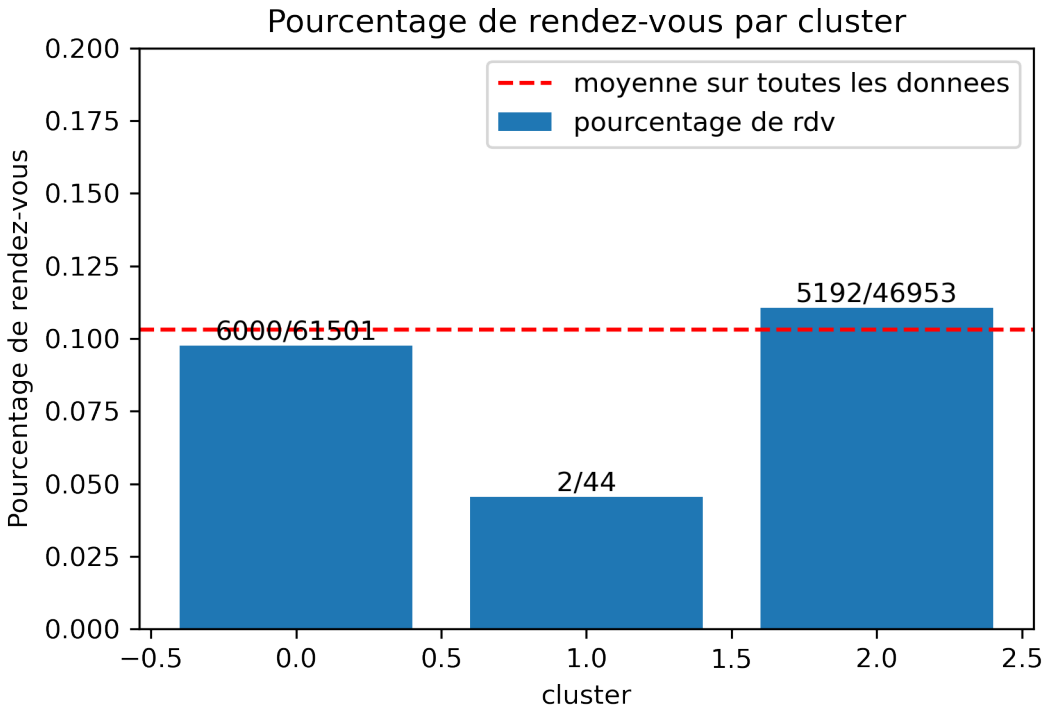


Figure 17. Pourcentage de rdv par cluster

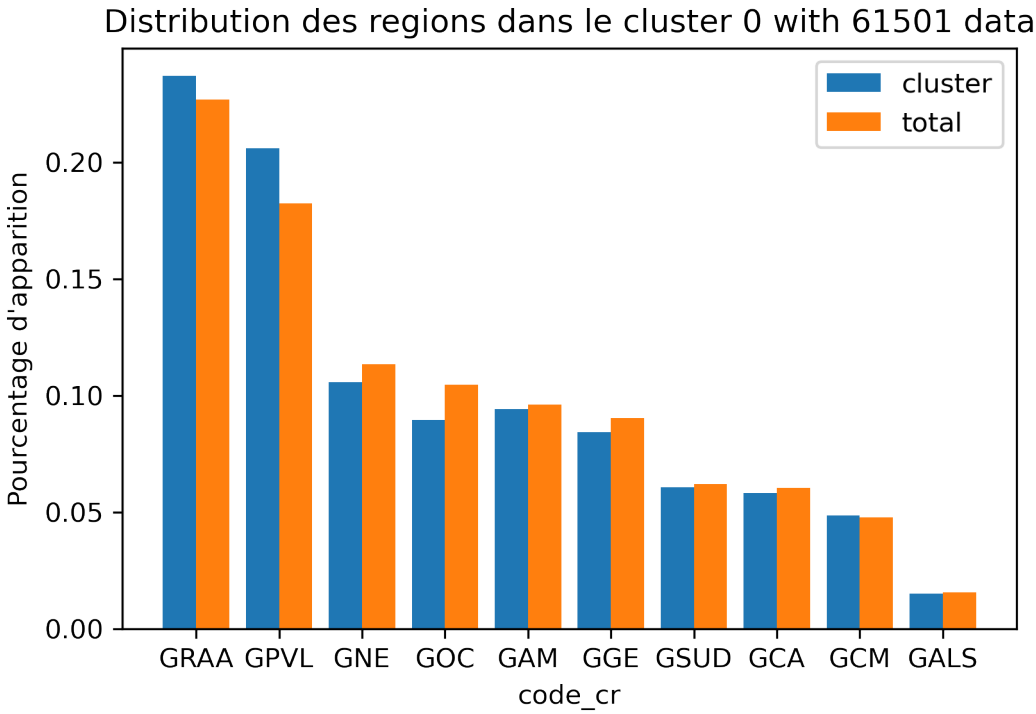


Figure 18. Proportion des régions dans le cluster 0

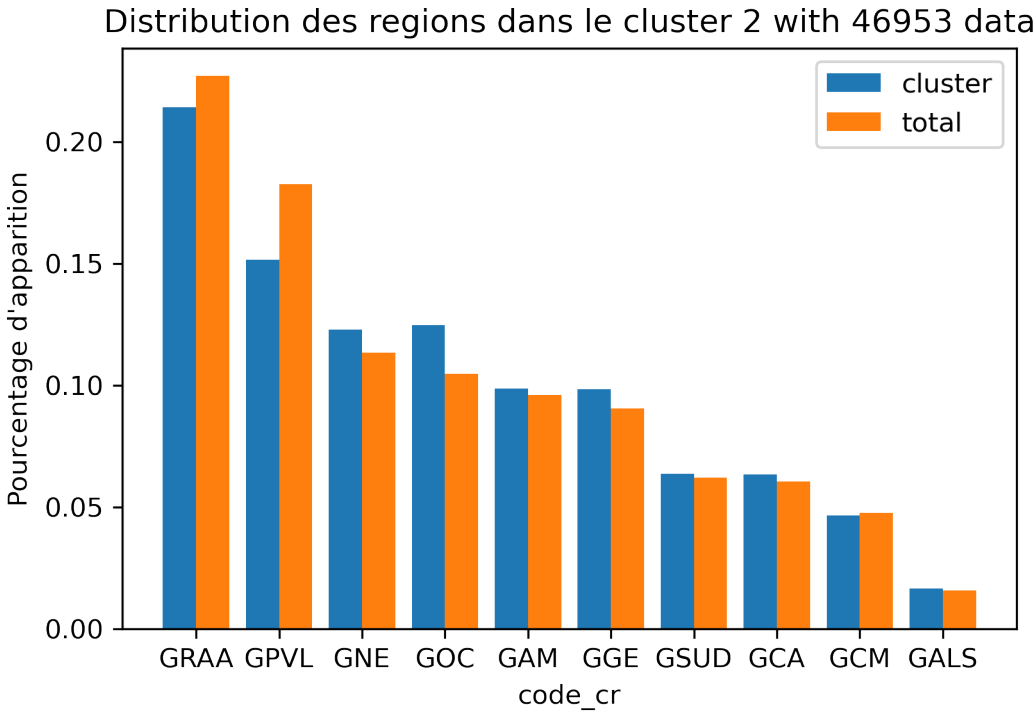


Figure 19. Proportion des régions dans le cluster 1

Analyse des rendez-vous

Avec seulement 3 clusters, les résultats ne sont pas très intéressants, il y a très peu de tendance.

On peut par exemple voir sur 17 que l'on est assez proche de la moyenne de rendez-vous. On remarque une légère différence entre le cluster 0 et 2 (2% de rdv en plus dans 2). Le cluster 1 avec 44 valeurs lui a une très faible proportion de rdv, mais 44 données correspond à une infime partie du dataset.

Analyse des régions

Sur les régions, il y a quelques régions surreprésentées dans les clusters. Par exemple, GPVL est surreprésenté de 3% dans le cluster 0 18 et est sous représenté de 3% dans le cluster 2 19, ce qui donne un écart de 6% d'apparition de la région entre les deux clusters.

Pour le dernier cluster (44 instances), plus de 50% des entreprises sont dans la région GNE.

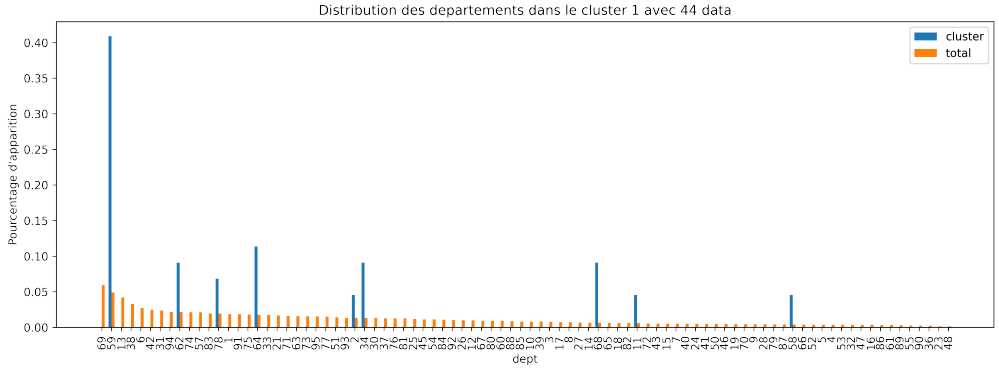


Figure 20. Proportion des départements dans le cluster 1

Analyse des départements

Si on analyse les départements, rien de bien intéressant ressort, si une région est surreprésentée, les départements de la région le sont aussi. On constate cependant que dans le cluster 1, la surreprésentation de la région GNE est surtout dû au département 59 ("Nord") qui représente 40% des données du cluster.

Caractérisation des clusters

Enfin, pour finir avec cette présentation des résultats du Kmeans avec 3 clusters, nous allons regarder ce par quoi sont caractérisés nos 3 clusters :

```
The 5 most important features for each cluster
Cluster 0:
1: risque: 3.80
2: endettement: 0.37
3: evo_risque: 0.33
4: ratio_benef: 0.32
5: age: 0.11
Cluster 1:
1: ca_total_FL: 37.12
2: effectif: 25.42
3: ca_export_FK: 14.34
4: evo_effectif: 13.87
5: risque: 3.11
Cluster 2:
```

```
1: risque: 2.21
2: endettement: 0.48
3: ratio_benef: 0.42
4: age: 0.15
5: evo_risque: 0.12
```

Le petit cluster (1) correspond en fait à de grosses entreprises avec de l'export : gros effectif, gros CA, gros CA à l'export, haute valeur de risque. Ces grosses entreprises se trouvent majoritairement dans le département du "Nord" mais représentent un très petit nombre d'entreprises dans les données. Les clusters 0 et 2 sont plus difficiles à séparer et preuve en est qu'ils sont très proches sur 21. Leurs différences sont que le cluster 0 a une valeur de risque beaucoup plus élevé, un moins bon ratio de bénéfices, un moins bon score d'endettement. Les entreprises du cluster 2 sont légèrement plus vieilles et ont une moins bonne croissance de leur valeur de risque.

Lorsque l'on utilise toutes les variables, la variable la plus importante reste le risque. Les autres variables les plus importantes sont différentes, mais sont les mêmes entre les clusters comme au-dessus où cluster 0 et 2 ont les mêmes variables. C'est surement dû aux déséquilibres dans les variables catégorielles.

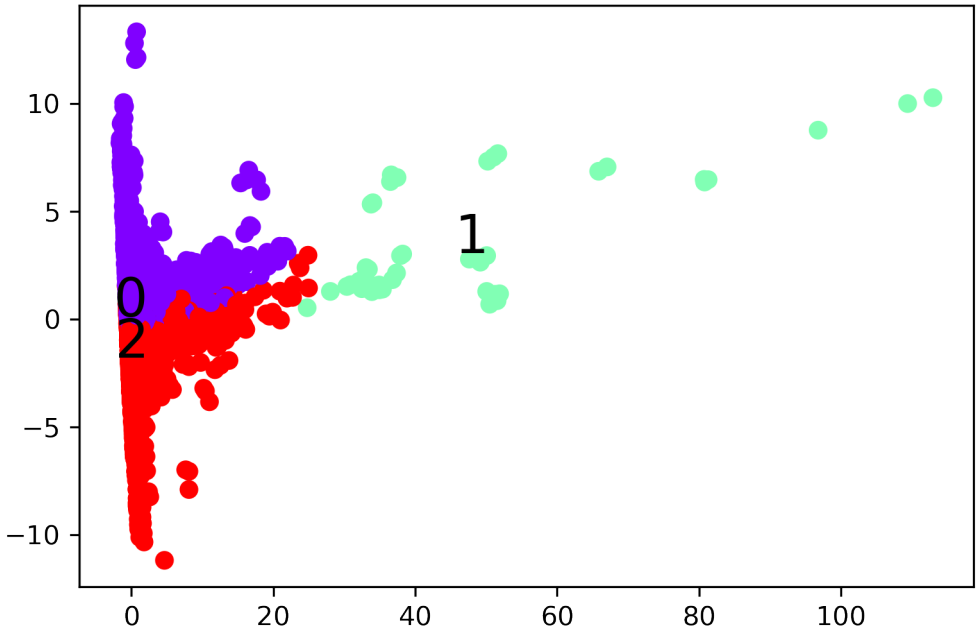


Figure 21. ACP des instances et clusters

Analyse de l'ACP

Si on utilise une ACP pour projeter les clusters et leurs instances dans 2 dimensions, on obtient 21. On peut voir que le cluster 1 représente sur l'ACP toutes les instances qui sont projetées au-delà de 25 sur l'axe X. Les clusters 0 et 2 étant séparés sur l'axe Y. Comme on peut le voir sur 22 l'axe X correspond à un mix entre effectif, CA (les deux sont très corrélés comme vu dans la partie descriptive), le chiffre à l'export et l'évolution de l'effectif. En somme, l'axe X correspond aux variables les plus importantes dans le cluster 1. L'axe Y correspond lui à un mix entre endettement, ratio bénéfice et risque. Et c'est bien cela qui différencie les

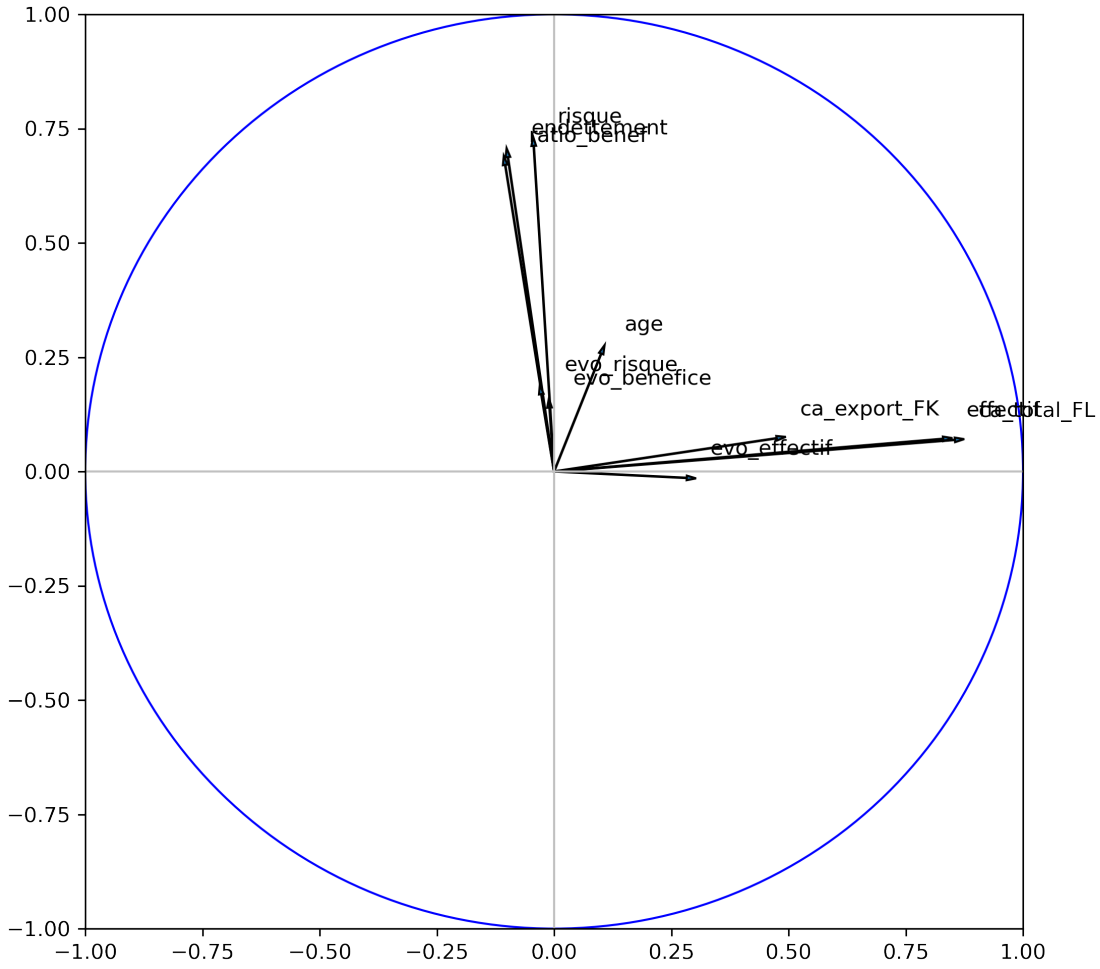


Figure 22. Projection des variables dans l'ACP

autres clusters.

Conlusion de l'analyse

Pour conclure cette analyse de Kmeans avec 3 clusters, les très grosses entreprises faisant de l'export et peu risqué sont rares dans le dataset et elles ont refusé de prendre un rendez-vous. Les entreprises avec la valeur de risque la plus faible prennent plus souvent rendez-vous que celles avec un risque élevé. Pour rappel, un risque élevé signifie qu'au contraire l'entreprise est peu risqué. Ces entreprises qui sont plus risquées sont étonnamment moins endettées et ont un meilleur ratio de bénéfices, que celles avec une valeur de risque élevé.

4.1.2 Clustering mixte avec 25 clusters

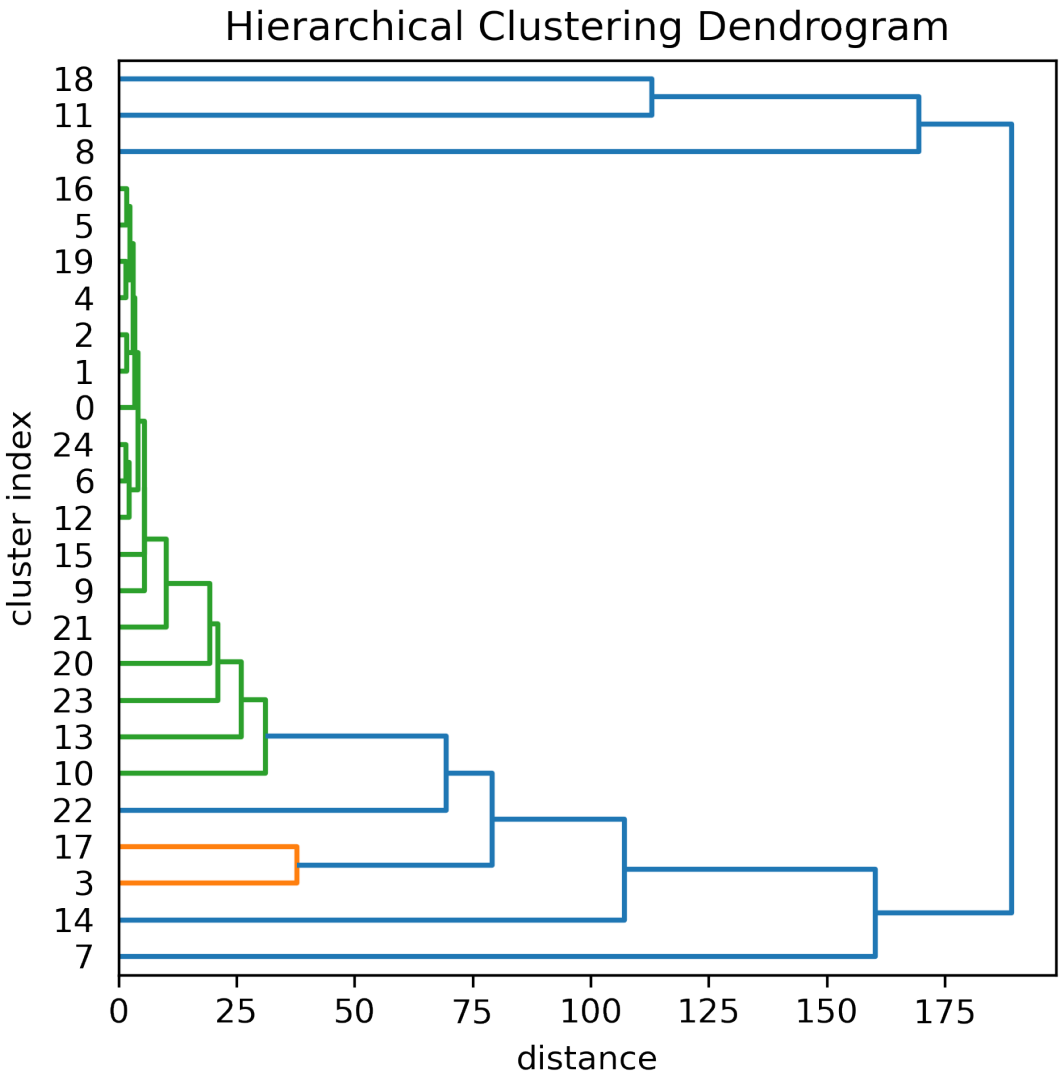


Figure 23. Clustering mixte avec toutes les variables

Données et évaluation

Les plots 23 et 24 montrent les dendrogrammes obtenus en utilisant toutes les variables (y compris les catégorielles) dans le premier et sans les données catégorielles dans le second. On peut voir que contrairement au Kmeans, les résultats sont relativement similaires. Dans les deux cas, on voit un groupe de 3 clusters en haut, un groupe central où les clusters sont très proches et un groupe. La manière dont les regroupements dans la partie centrale sont faits est différente, dans celui avec toutes les variables le groupe central se regroupe à chaque fois entre le groupe et 1 élément. Tandis que dans celui sans les variables catégorielles, il y a plus de sous groupes.

Si on regarde le dendrogramme, le bon nombre de cluster serait plutôt entre 8 et 2 (en regardant les distances entre les regroupements). Ce qui change beaucoup par rapport aux 3 clusters donnés par le score silhouette. Cette fois-ci, on va donc regarder 8 clusters. Nous avons comparé le résultat avec les variables catégorielles et sans, avec, il y a un

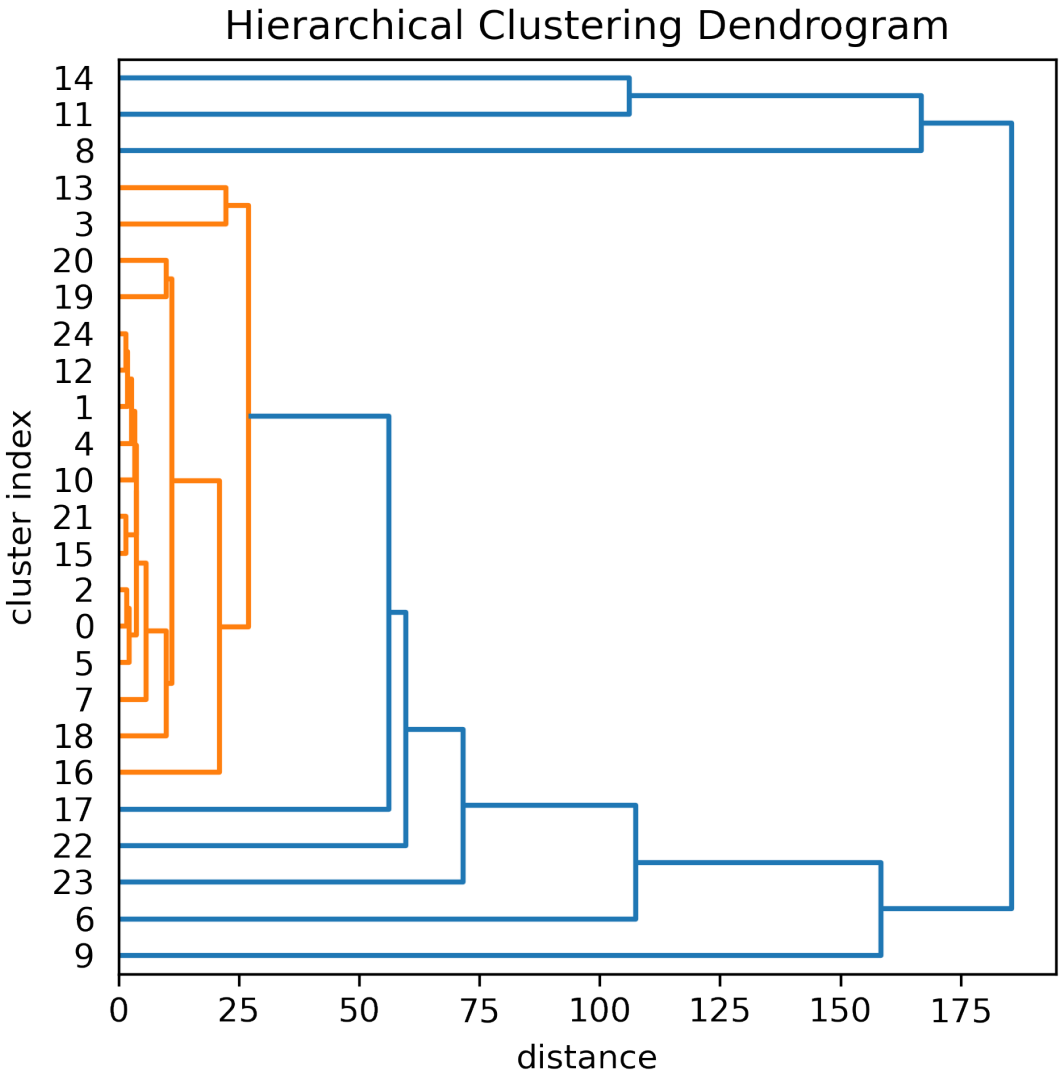


Figure 24. Clustering mixte sans les variables catégorielles

cluster qui regroupe plus de 99% des instances, sans, il y a quelques clusters avec des tailles significatives. Nous allons donc continuer à ne pas utiliser les variables catégorielles.

Analyse des rendez-vous

On peut voir sur 25 que sur les 8 clusters, 5 ont moins de 1000 instances, et sur les 3 restants, 2 ont moins de 10 000 instances. On peut déjà dire que le clustering mixte (et surtout la partie hiérarchique) génère facilement des clusters avec seulement quelques données qui sont très éloignés du reste. On peut voir que le cluster 6 est quelques % en dessous de la moyenne des rendez-vous. Tandis que la plupart des petits clusters à l'exception du cluster 5 ont un taux de rendez-vous très inférieur à la moyenne.

Analyse des régions

Sur les régions, il y a quelques régions surreprésentées dans les clusters. On peut voir sur 26 que la région GRAA est légèrement sous-représentée tandis que la région GPVL est légèrement sur-représentée. Au contraire, dans 27 on constate l'inverse, il y a aussi une légère sur-représentation de la région GGE. Les autres clusters ne portent aucune information

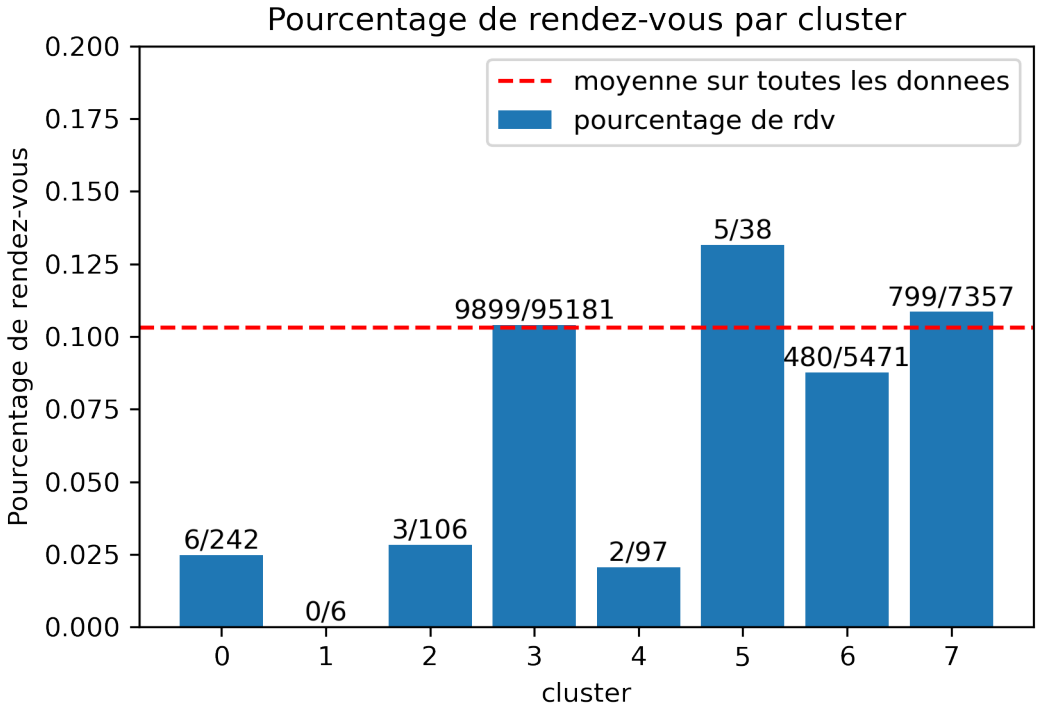


Figure 25. Pourcentage de rdv par cluster

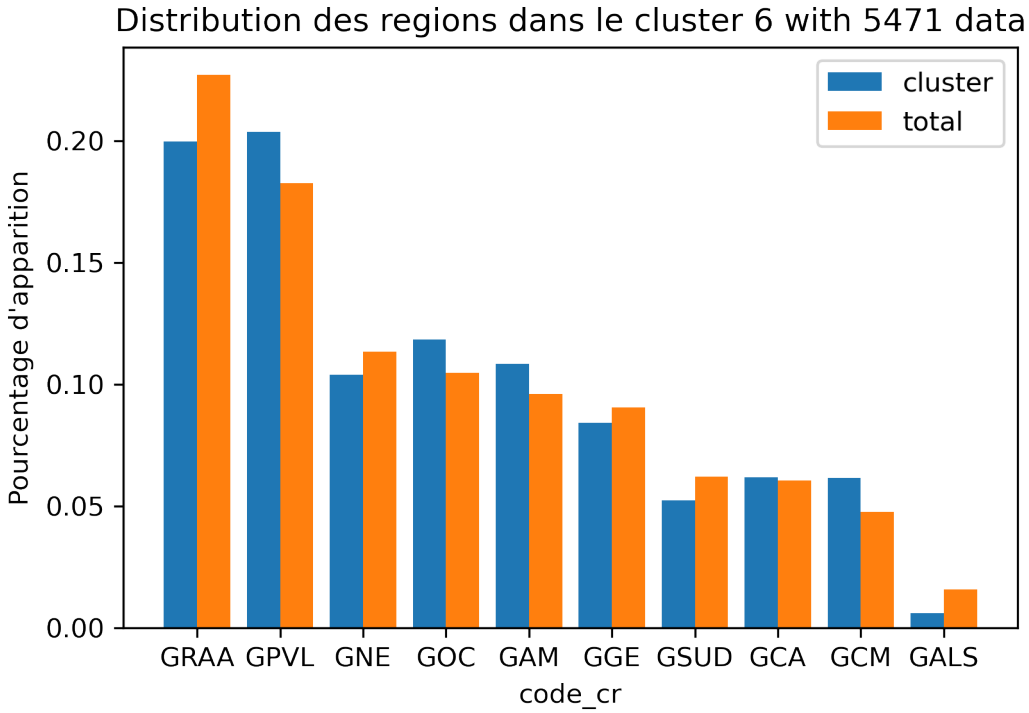


Figure 26. Proportion des régions dans le cluster 6

intéressante sur cet attribut.

Analyse des départements

Si on analyse les départements, rien de bien intéressant ressort, si une région est surre-

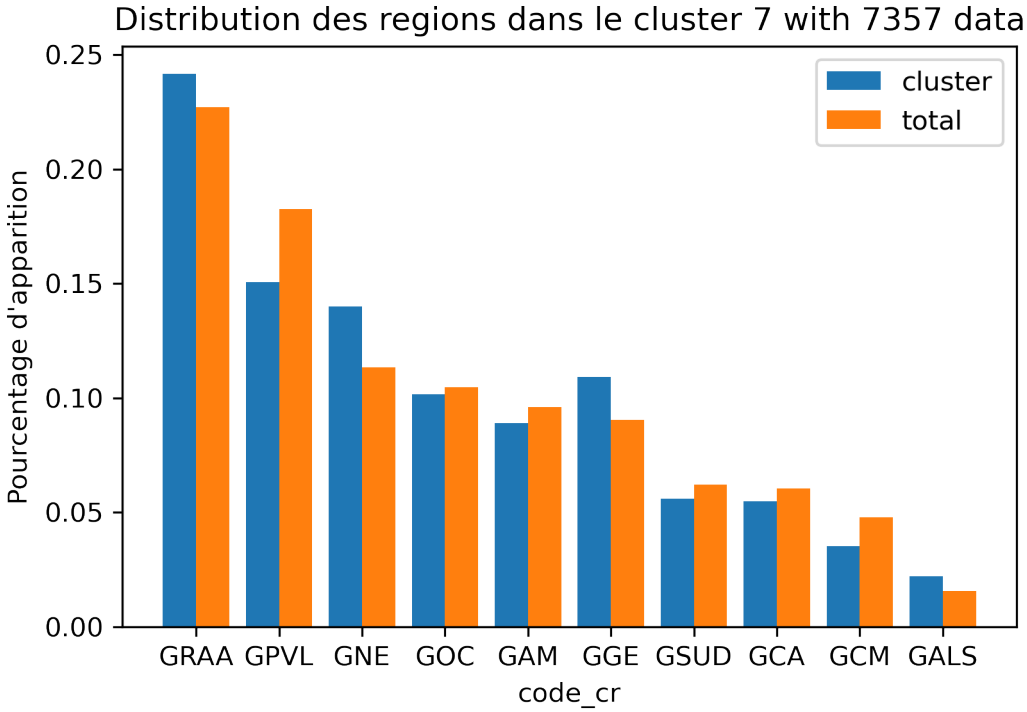


Figure 27. Proportion des régions dans le cluster 7

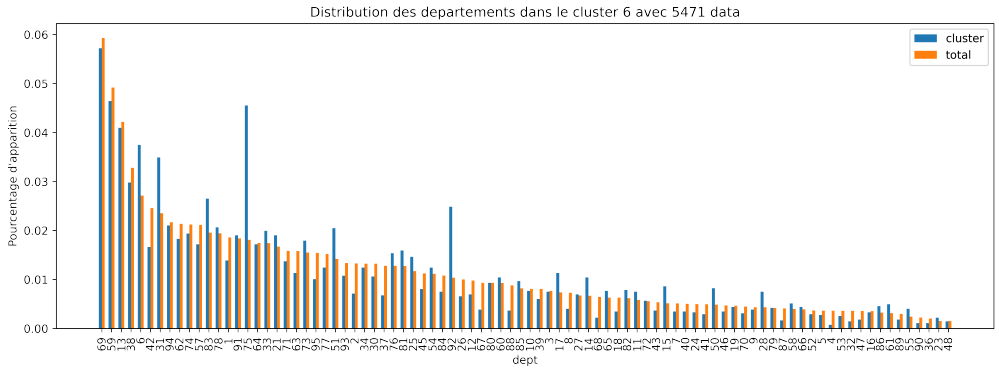


Figure 28. Proportion des départements dans le cluster 6

présentée, les départements de la région le sont aussi. On constate cependant que dans le cluster 6, la surreprésentation du département 75 ("Paris") et du 92 qui représente une partie anormalement élevée des données du cluster.

Analyse de l'ACP

Le cercle de corrélation reste le même donc voir 22. On peut voir sur 29 que le cluster violet, rouge, orange, bleu-vert beige se différencie sur l'axe des Y. Bleu-foncé, vert et bleu se différencie des autres par l'axe des X. Cependant, bleu et vert se superposent beaucoup dans cet affichage tandis que bleu-vert (le gros cluster avec 90k instances) se superposent avec beaucoup de cluster sur Y.

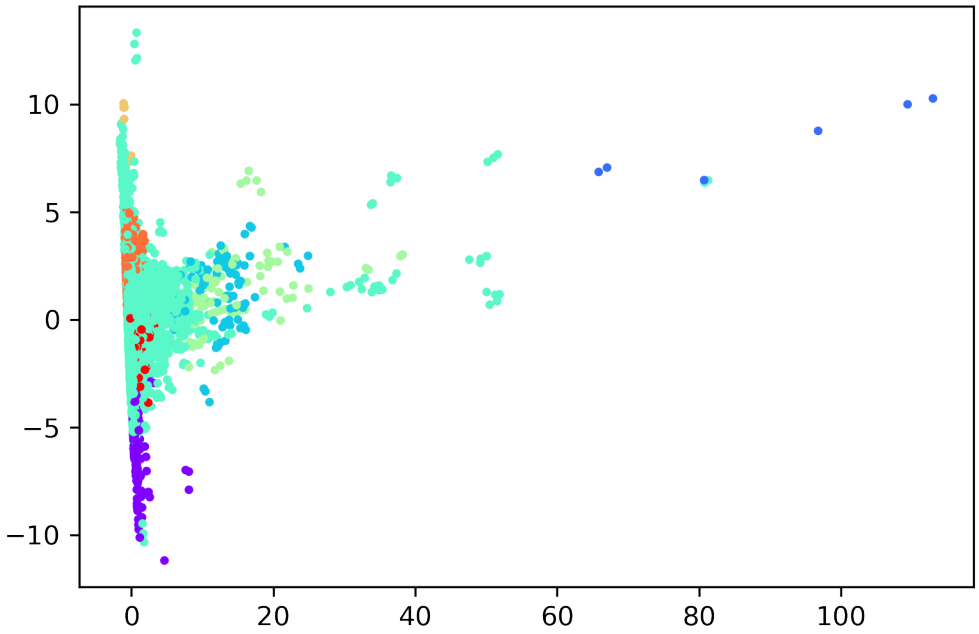


Figure 29. ACP des instances et clusters

Conclusion de l'analyse

Les résultats sont moins intéressants qu'avec le Kmeans, car ce type de clustering a tendance à isoler les données extrêmes dans des clusters et donc à créer quelques très gros clusters qui du coup ne portent pas d'informations très intéressantes. On peut noter tout de même quelques spécificités, notamment le cluster 6 qui prend peu de rendez-vous. Cette approche aurait été surement pratique pour isoler des données aberrantes ou extrêmes.

4.2 Prédiction d'attributs

4.2.1 Résultats divers

Nous avons testé plusieurs méthodes de classification pour la prédiction de la variable `rdv` en cross-validation afin d'obtenir de premiers résultats, présentés sans rééquilibrage des classes :

- DummyClassifier : 88.90% ± 0.00%
- GaussianNB : 49.67% ± 6.56%
- LogisticRegression : 88.90% ± 0.00%
- MLPClassifier : 92.52% ± 2.16%
- KNeighborsClassifier : 91.69% ± 1.16%
- RandomForestClassifier : 92.61% ± 3.52%

Ici, les valeurs par défaut ont été retenues dans l'optique de faire une courte comparaison. Certaines méthodes échouent complètement à prédire la prise de rendez-vous, et la plupart parviennent difficilement à atteindre de meilleurs scores que la prédiction de la classe majoritaire (DummyClassifier).

4.2.2 Arbres de décision

En procédant à un équilibrage des classes, nous avons testé la prédiction sur des arbres de décision avec un paramètre `ccp_alpha`. Ici, un score de 50% serait atteint par un modèle qui prédit toujours la même classe. Plus le paramètre `ccp_alpha` est élevé, plus aggressif l'élagage est, et plus simple l'arbre sera.

- $1e-7$: 83.29% ±13.83%
- $1e-6$: 83.34% ±13.81%
- $1e-5$: 83.08% ±13.96%
- $5e-5$: 76.91% ±12.17%
- $1e-4$: 68.90% ±10.97%
- $1e-3$: 51.81% ±22.60%

Nous remarquons ici un très grand impact de la complexité de l'arbre sur la qualité du résultat. Les arbres les plus complexes ont tendance à contenir des branches pour des cas qui couvrent assez peu d'instances, ce qui a grande tendance à favoriser le sur-apprentissage. Dans notre cas, et comme expliqué précédemment, il est probable que le sur-apprentissage est d'autant plus facilité que plusieurs profils très similaires sont présents dans le jeu de données, avec des valeurs de `rdv` égales, qui est causé par la présence multiple de mêmes entreprises.

Au-delà de chercher à prédire avec haute certitude si un rendez-vous va se produire, ce qui semble impossible, nous cherchons notamment à appliquer une méthode interprétable qui nous permettrait de mieux comprendre les données et les tendances qui se dégagent.

Pour rappel, à cette fin, nous avons fait en sorte de jouer sur les attributs pour limiter la taille du graphe, et élaguer au maximum les branches qui contribuent peu au modèle final. Les critères finaux ont été `ccp_alpha=0.00015`, `max_leaf_nodes=20` et `min_samples_leaf=200` afin de limiter au maximum le sur-apprentissage et rendre la plus facile l'interprétation.

Pour les figures suivantes, les classes sont équilibrées. C'est-à-dire qu'une distribution entre les classes de `[0.5, 0.5]` correspond à la distribution moyenne du dataset, alors que `[0.8, 0.2]` correspondrait à une grande sur-représentation de non-prise de rendez-vous.

La première ligne correspond au nom de variable (potentiellement tronqué si trop long) et au seuil considéré. La deuxième ligne correspond au Gini pour le noeud. Le troisième correspond au nombre d'instances couvertes par le noeud dans le jeu de données. La quatrième correspond à la distribution des classes pour le noeud (aussi représentée par la couleur du noeud). Enfin, la cinquième correspond à la classe majoritaire.

En somme, nous sommes particulièrement intéressés par deux cas de figures : Les tendances généralistes, qui couvrent un large nombre d'entreprises mais ont un effet relativement moindre sur la prise de rendez-vous, et les "exceptions", qui sont des cas particuliers soulevés par plusieurs centaines d'entreprises avec un écart fort avec le comportement usuel des entreprises.

La branche partant vers la gauche correspond au cas où la condition est vérifiée, celle partant vers la droite correspond au cas où elle ne l'est pas. Dans le cas des variables booléennes, une valeur de 0 correspondant à une valeur fausse et 1 à une valeur vraie, une comparaison $<= 0.5$ signifie que la branche de gauche est prise pour une valeur fausse et la branche de droite est prise pour une valeur vraie.

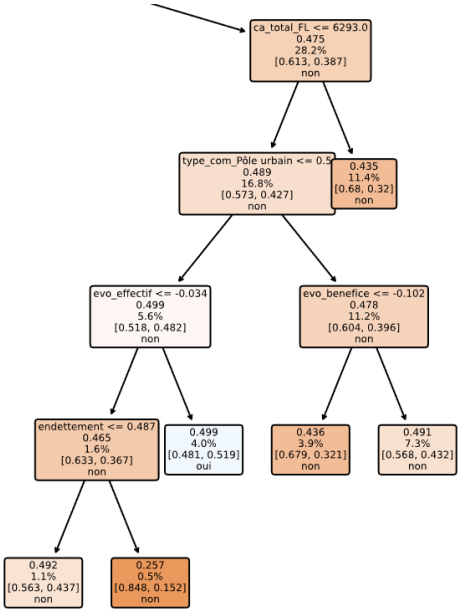


Figure 30. Branche "actionnaire entreprise"

La figure 30 correspond à la branche "actionnaire = entreprise". Nous constatons ainsi une tendance notable : Les entreprises avec un actionnariat "classique" tendent à généralement moins accepter les rendez-vous. On peut observer certains cas particuliers, notamment celui du bon taux d'endettement (compte tenu de certaines précondition) où l'on observe une forte sous-représentation de la prise de rendez-vous.

À l'inverse, la figure 31 correspond à la branche "actionnaire autre qu'entreprise", ce qui correspond aux entreprises sans actionnariat, familiales, individuelles, etc., qui est un cas de figure où les entreprises ont plus tendance à accepter les rendez-vous.

Notamment, on observe un bon taux d'acceptation de rendez-vous pour les petites entreprises (effectif <20), qui semblent plus faciles à cibler, notamment dans le domaine rural.

Dans le sens inverse, on retrouve encore une corrélation entre grande entreprise et faible taux de rendez-vous : la catégorie correspondant à un effectif $>= 25$ voit un généralement faible taux de prise de rendez-vous. Chose intéressante, cette tendance s'accroît encore

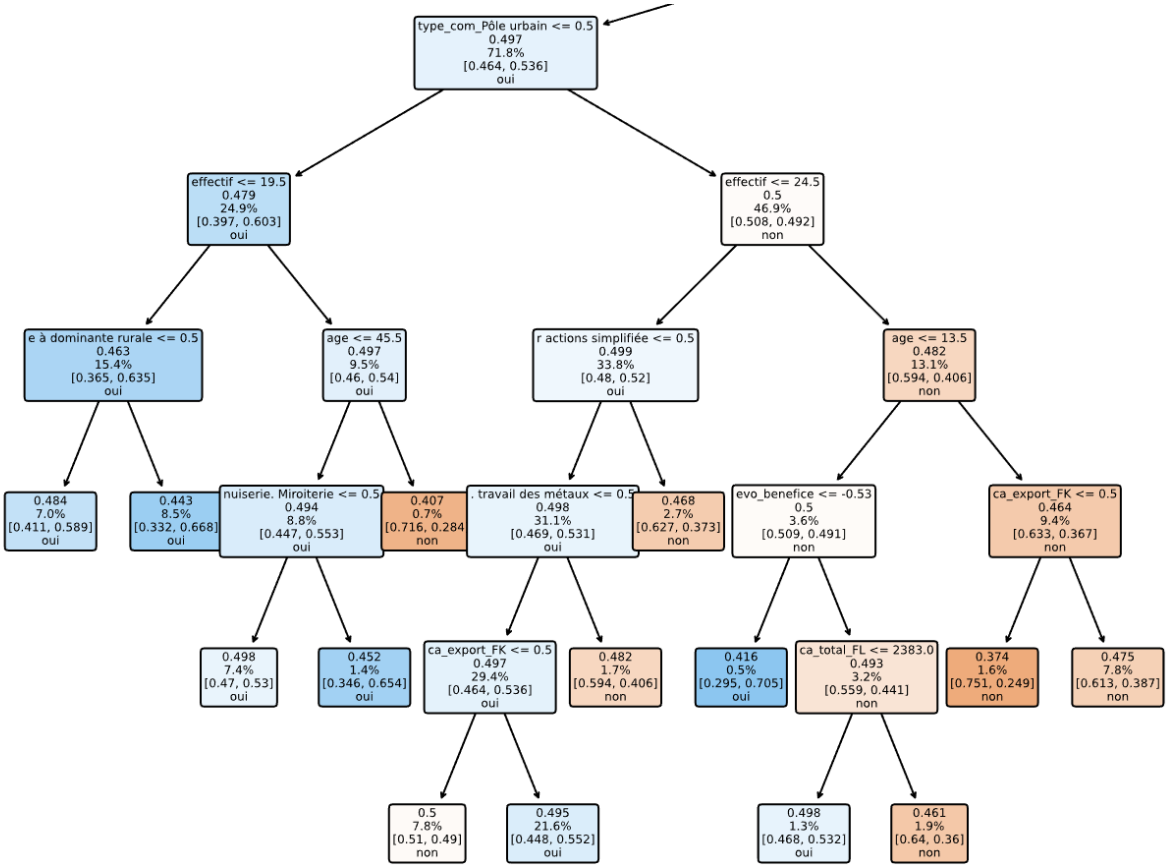


Figure 31. Branche non actionnaire entreprise

davantage pour les entreprises qui réalisent pas ou très peu de leur chiffre d'affaire à l'exportation.

Cette analyse tend à montrer que les grandes entreprises tendent à moins accepter de rendez-vous, alors que les plus petites entreprises tendent à davantage les accepter.

5 Conclusion

Dans le cadre de ce projet, dont l'objectif était la fouille de données, nous avons eu à nous renseigner en profondeur sur un domaine spécifique dans l'objectif de comprendre les données, de pré-traiter ces données, et d'effectuer différentes analyses dans le but de mieux comprendre le profil des entreprises et des tendances dans leurs comportements vis-à-vis de la prise de rendez-vous.

L'utilisation de données réelles est intéressante et introduit les enjeux d'une situation réelle, bien que parfois nous avons passé beaucoup de temps à chercher à inférer la signification de certaines variables et problématiques, nous posant des questions qu'il était difficile de répondre sans pouvoir interagir avec les personnes qui ont constitué le jeu de données.

Notre travail présente certaines limitations dans l'analyse. Par exemple, nous n'avons pas réellement pu effectuer de traitement spécifique des entreprises que nous avons identifiées comme étant présentes plusieurs fois dans le jeu de données, ni cerner avec exactitude leur nombre. Cela a eu des conséquences significatives dans notre prédiction d'attributs.

En revanche, nous pensons avoir opéré à une description et une analyse de la distribution

des variables raisonnables, et avons tâché d'identifier et de comprendre les limitations dans nos approches. Compte tenu du jeu de données, l'analyse que nous avons effectué dès le début du projet nous paraît relativement robuste.

Il est possible que pousser encore davantage l'analyse dans l'objectif d'identifier des corrélations pourrait éclaircir davantage l'analyse et l'identification de tendances dans les données, néanmoins, nous pensons avoir été relativement exhaustif à ce niveau.

Globalement, nous pensons qu'une description plus approfondie des données et une introduction à la terminologie employée plus poussée aurait pu aider. Certains termes étaient en effet confus, parfois ayant des définitions confuses ou subtilement différentes selon le contexte.