DataScientest

French Industry

Analyse comparative des données sur les salaires en France selon les déterminants géographiques et socio-économiques

BENLALA Sarah, BEYE Charles et MARECHAL Louis

30/08/2024

Table des matières

Introduction 2

Présentation des sources et jeux de données : 2

I. Dataviz et interprétations 3

1. Salaire moyen par région 3
2. Population par région 3
3. Salaire moyen par département : zoom sur la région parisienne 4
4. Distribution du salaire par sexe 5
5. Distribution du salaire par catégorie d'emploi 6
6. Salaire moyen par catégorie d'âge et sexe 7
7. Salaire par sexe et catégorie d'emploi 7
8. Répartition de la population 8
9. Répartition des grandes entreprises en France (avec plus de 500 salariés) 11
10. Heatmap 13
11. Test statistique 14

II. Etapes de preprocessing des datasets en vue du ML 16

1. Merge des jeux de données : 16
2. Gestion des NaNs 16
3. Numérisation des colonnes : 17
4. Création de colonnes 17
5. Sélection et suppression des colonnes 17
6. Variables (features) conservées et jeu de données final 18

III. Machine Learning 19

1. Sélection de modèles de Machine Learning 19
2. Linear Regression 19
3. Lasso 24
4. Decision Tree regressor 26
5. Gradient Boosting 33
6. Random Forest Regressor 36
7. Ridge 39
8. Résumé des métriques et choix du modèle 42

Conclusion 43

Introduction

Agence publique chargée de collecter des données sur l'économie et la population française, l'Insee possède et diffuse une base de données précieuse pour les autorités, les sociologues, les journalistes ou encore, entre autres… les data analysts. En effet, ses études rendent possible une analyse de la société sous différents axes, et notamment par le prisme des inégalités, principalement salariales : c'est ce thème qui sera l'objet du présent projet.

Répartition de la population par ville ou par région, distribution et tailles des entreprises sur le territoire national, ou encore salaire moyen par catégorie, sont autant de variables à notre disposition dont nous tenterons d'extraire du sens. Notre objectif sera d'une part de tenter, sur base des jeux de données à notre disposition, de dresser un tableau des inégalités salariales en France, mais également d'identifier les variables exerçant une influence, positive ou négative, sur le salaire. Enfin, nous serons amenés à nous pencher sur la structuration des jeux de données mis à disposition par l'Insee : permettent-ils une bonne interprétabilité des tendances ? Jusqu'à quel niveau de détail permettent-ils d'analyser les mécanismes expliquant les inégalités ?

Présentation des sources et jeux de données :

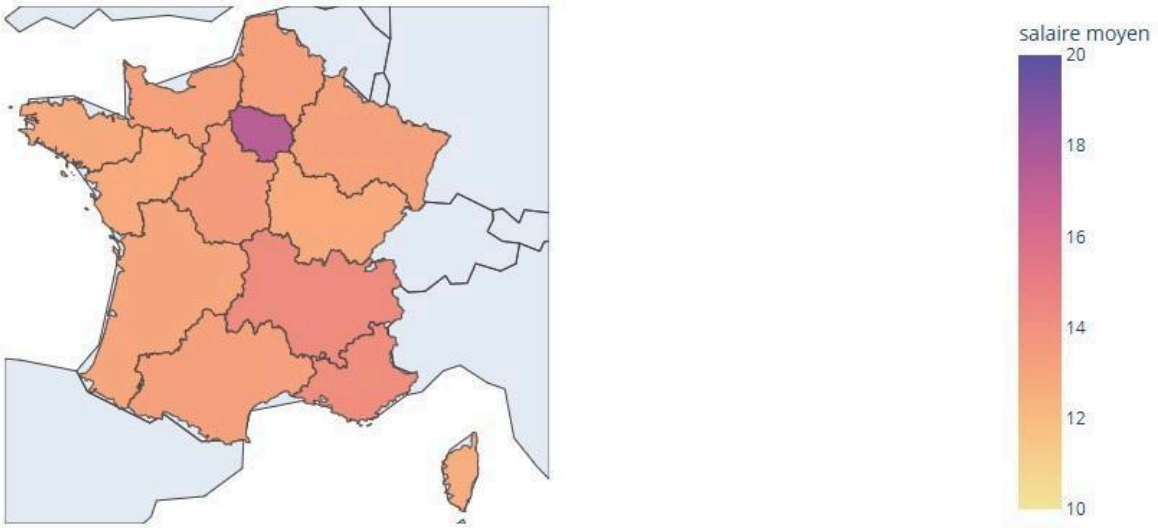
Afin de mener à bien le présent projet, nous avions à notre disposition 4 jeux de données différents issus de l'Insee :

- Un dataset *name_geographic_information.csv* (dimension : 36 840 entrées x 14 colonnes), reprenant pour chaque code géographique de la ville, des données géographiques comme la région d'appartenance, le numéro du département, la préfecture etc.
- Un dataset *base_etablissement_par_tranche_effectif.csv* (36 681 entrées x 14 colonnes), reprenant pour chaque code géographique le nombre d'entreprises implantées par ville, classées par tranches de salariés (entre 1 et 5 employés, entre 6 et 9 etc)...
- Un dataset *population.csv* (8 536 584 entrées x 7 colonnes), reprenant diverses informations administratives pour chaque code géographique sur la répartition de la population par sexe, catégorie d'âge et mode de cohabitation.
- Un dataset *salaire.csv* (5 136 entrée x 26 colonnes), contenant pour chaque code géographique, notre variable cible (SNHM14, soit le salaire net moyen par heure), mais également les salaires par catégorie (sexe, âge, catégorie socio-professionnelle).

I. Dataviz et interprétations

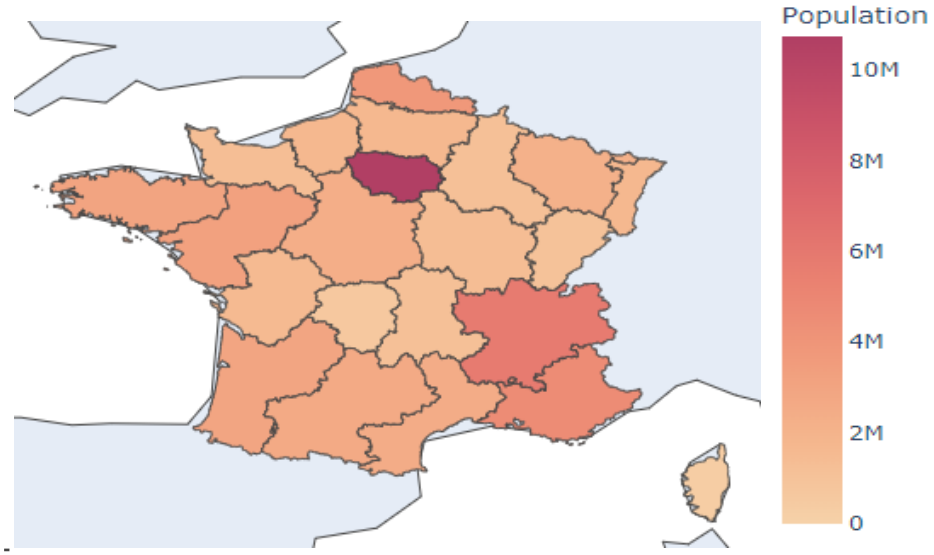
La visualisation de nos jeux de données nous permet de faire plusieurs observations quant à la répartition des inégalités salariales en France et des facteurs expliquant celle-ci.

1. Salaire moyen par région



Cette carte du salaire moyen par région en France nous montre une répartition relativement homogène du salaire dans le pays, à l'exception de la région Ile-de-France qui semble compter un salaire moyen extrêmement élevé par rapport au reste des régions de France métropolitaine. Loin après l'Ile de France, le salaire moyen par ville est légèrement plus élevé en Auvergne-Rhône-Alpes et en PACA qu'ailleurs.

2. Population par région

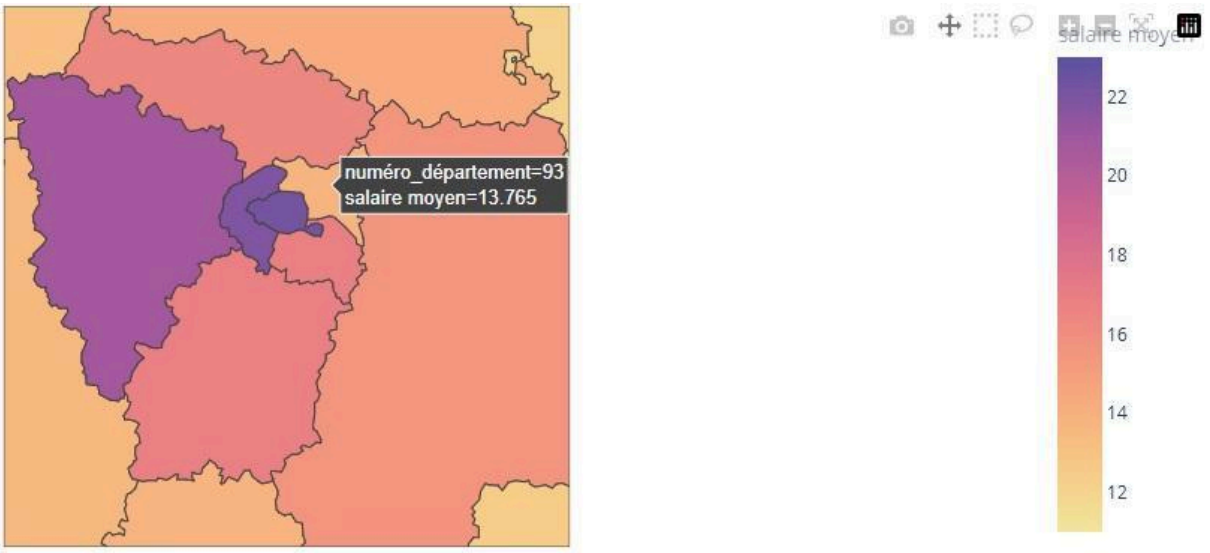


Cette carte confirme la grande différence de distribution de la population en France. L'Île de France est de très loin la région la plus peuplée avec plus de 10 millions d'habitants sur son territoire. Les régions Auvergne-Rhône Alpes et PACA arrivent en 2ème et 3ème position avec une population qui s'établit autour de 5 millions d'habitants.

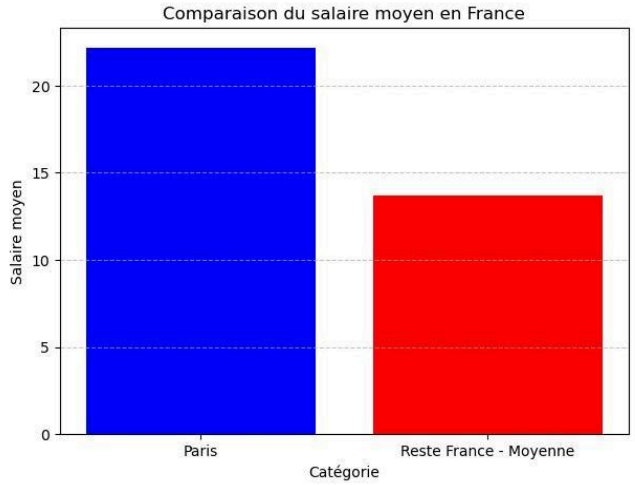
Les deux cartes montrent ainsi une possible corrélation entre le nombre d'habitants d'une région et le salaire moyen de celle-ci, qu'il conviendra de vérifier plus tard dans ce rapport à l'aide de modèles prédictifs de machine learning.

3. Salaire moyen par département : zoom sur la région parisienne

Pour en revenir au salaire moyen, un zoom sur la région parisienne, désormais découpée en départements (voir figure ci-dessous), nous permet d'apporter de la nuance à l'analyse. En effet, on peut constater que le salaire, particulièrement élevé dans la région, ne connaît pas une distribution homogène dans l'ensemble des départements d'Ile de France. Ainsi, la Seine-Saint-Denis (93) compte un salaire moyen par ville largement en-deça de Paris (75) et Hauts-de-Seine (92), les deux départements d'IDF où les salaires moyens sont les plus élevés. Viennent ensuite les Yvelines (78) alors que le reste des départements franciliens ont un salaire moyennement élevé.

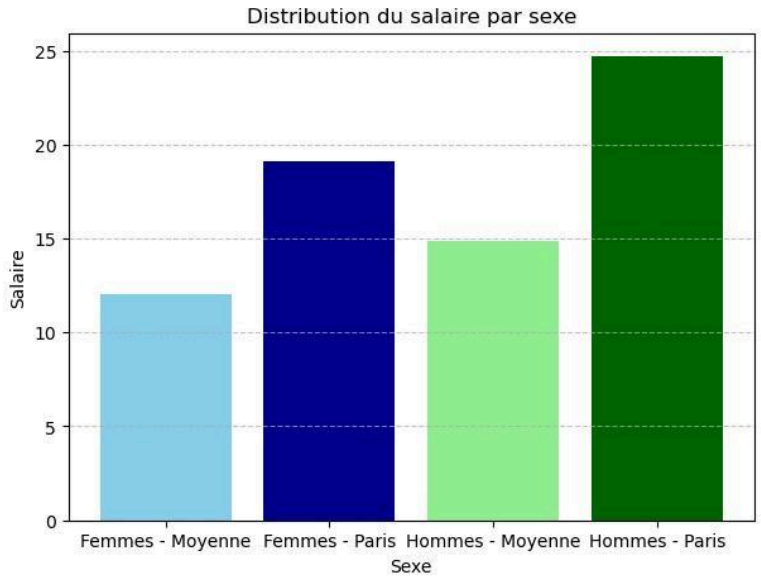


Sans surprise, on retrouve cette forte disparité de la variable “salaire net moyen par heure” lorsque l’on met d’un côté la ville de Paris, et de l’autre, le reste de la France, avec respectivement environ un salaire horaire moyen de 22 et de 14 euros.



4. Distribution du salaire par sexe

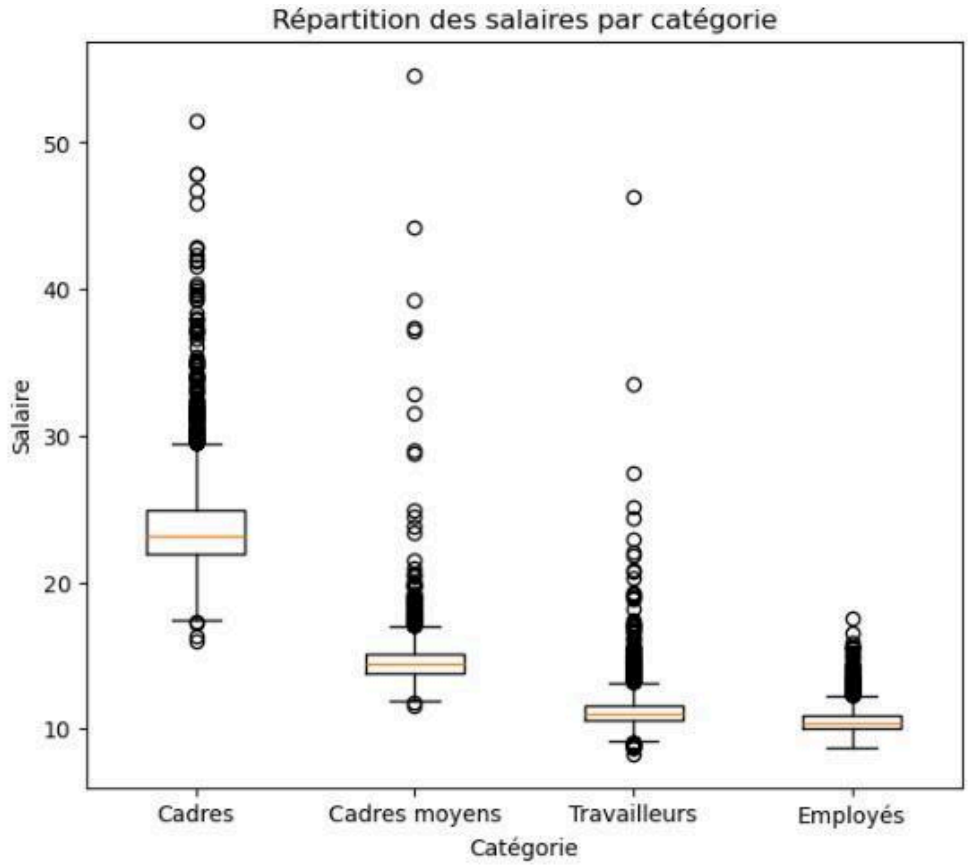
Si l’on introduit ensuite une variable “Sexe” (voir figure ci-dessous), on constate une différence entre les salaires nets moyens des hommes et des femmes, tant au niveau de la moyenne nationale, qu’au sein de Paris. L’écart au sein de la capitale entre les sexes est même plus important qu’au niveau de la France entière. Il faut ici nuancer notre interprétation en soulignant un biais de notre jeu de données, qui comporte peu de détails pourtant cruciaux pour interpréter plus en profondeur ces résultats (par exemple : niveau d’études, type de formation, secteur d’activité etc). Dès lors, nous ne sommes pas en mesure de tirer davantage de conclusions sur cet axe d’analyse que celle d’une distribution inégale du salaire selon le sexe.



5. Distribution du salaire par catégorie d'emploi

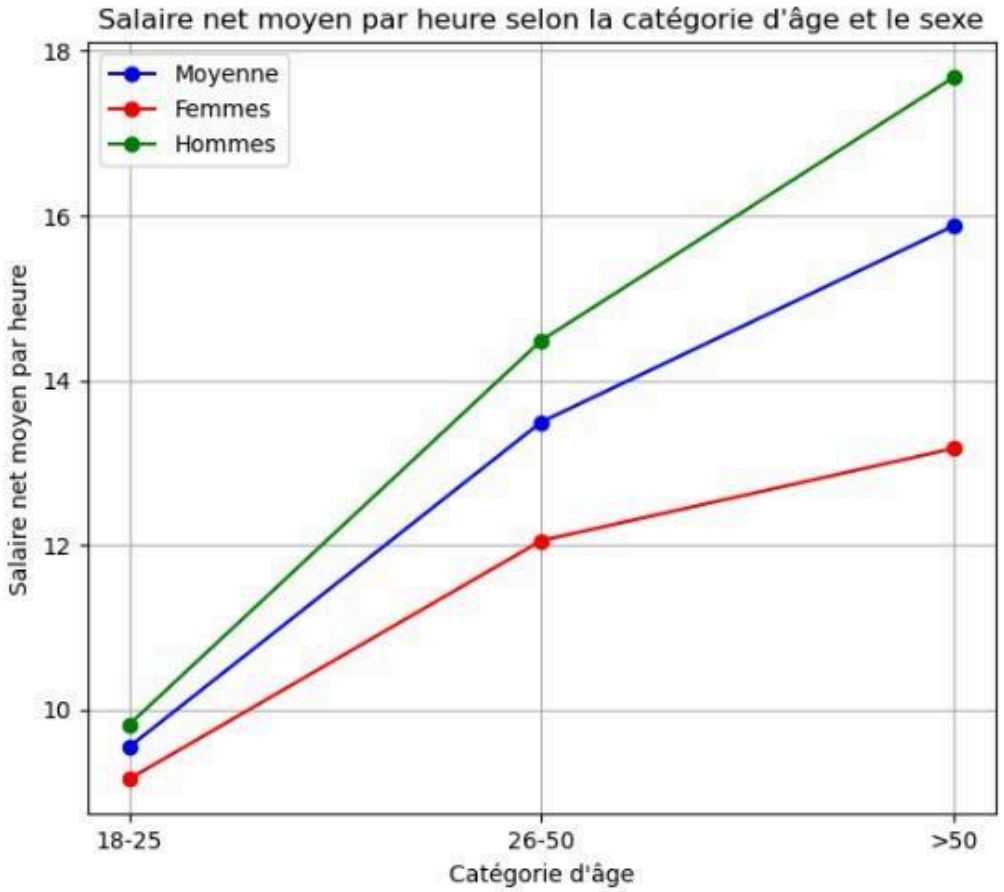
Si l'on regarde désormais (voir ci-dessous) la répartition des salaires moyens par ville, pour chaque catégorie d'emploi, on remarque que le fait d'appartenir ou non à la catégorie des cadres semble impacter de manière très positive la moyenne des salaires. La moyenne des trois autres catégories est en effet beaucoup plus basse. Par ailleurs, on peut remarquer que les valeurs extrêmes sont concentrées davantage au-dessus de la moyenne qu'en dessous. Celles-ci ne semblent pas être des valeurs aberrantes, mais de simples *outliers*.

A ce stade, on peut constater que la moyenne du salaire est pour chaque catégorie assez basse et que certaines villes offrent un salaire parfois beaucoup plus élevé que la moyenne de chaque catégorie. En revanche, peu de villes offrent un salaire nettement inférieur à la moyenne. Au niveau de la disparité, les valeurs extrêmes pour les employés restent assez proches de la moyenne, la distribution est relativement homogène, aucune ville n'offre un salaire moyen extrêmement élevé aux employés. L'inverse peut être observé pour la catégorie "travailleurs" qui malgré une moyenne basse, peuvent espérer de meilleurs salaires dans certaines villes comme on le voit avec les valeurs extrêmes qui montent plus haut. La disparité pour les cadres moyens est importante, avec des maximums qui montent assez haut même si la moyenne est plus basse que pour les cadres.



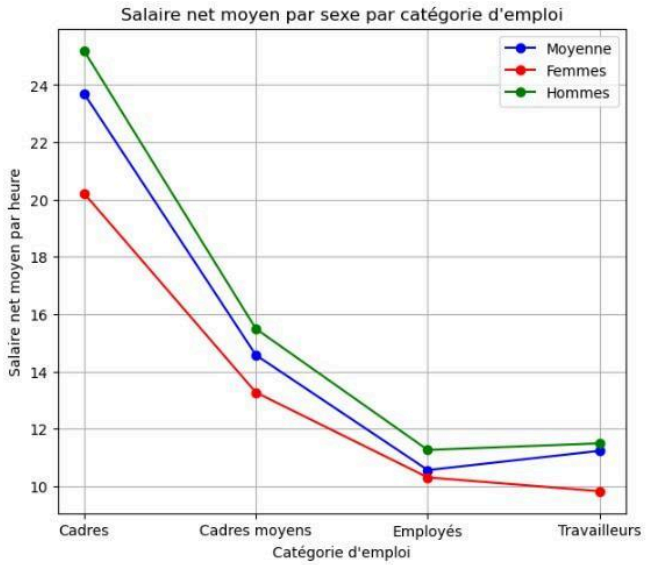
6. Salaire moyen par catégorie d'âge et sexe

Intéressons-nous désormais à l'âge et son influence sur le salaire. Sur la figure ci-dessous on constate deux choses. Tout d'abord le salaire a tendance à croître avec l'âge pour tout le monde (homme et femme). Il croit assez fortement entre la première et deuxième catégorie d'âge. Ensuite il croît de manière différenciée entre hommes (croissance importante, quasi linéaire) et les femmes (croissance beaucoup moins marquée). Par ailleurs, on retrouve pour chaque catégorie d'âge les disparités hommes / femmes notées précédemment avec les réserves expliquées plus haut quant à l'interprétabilité des résultats. Soulignons ici un autre biais de notre jeu de données : les catégories d'âge qui sont assez larges et rendent difficiles l'interprétation. Ainsi, par exemple, la catégorie 26-50 comporte à la fois les jeunes travailleurs fraîchement diplômés ainsi que les travailleurs confirmés. Cette structuration des données de l'Insee nous empêche de rentrer dans le détail et de vérifier par exemple une hypothèse selon laquelle le salaire moyen des 26-30 ans serait moins important que celui des 40-45 ans.

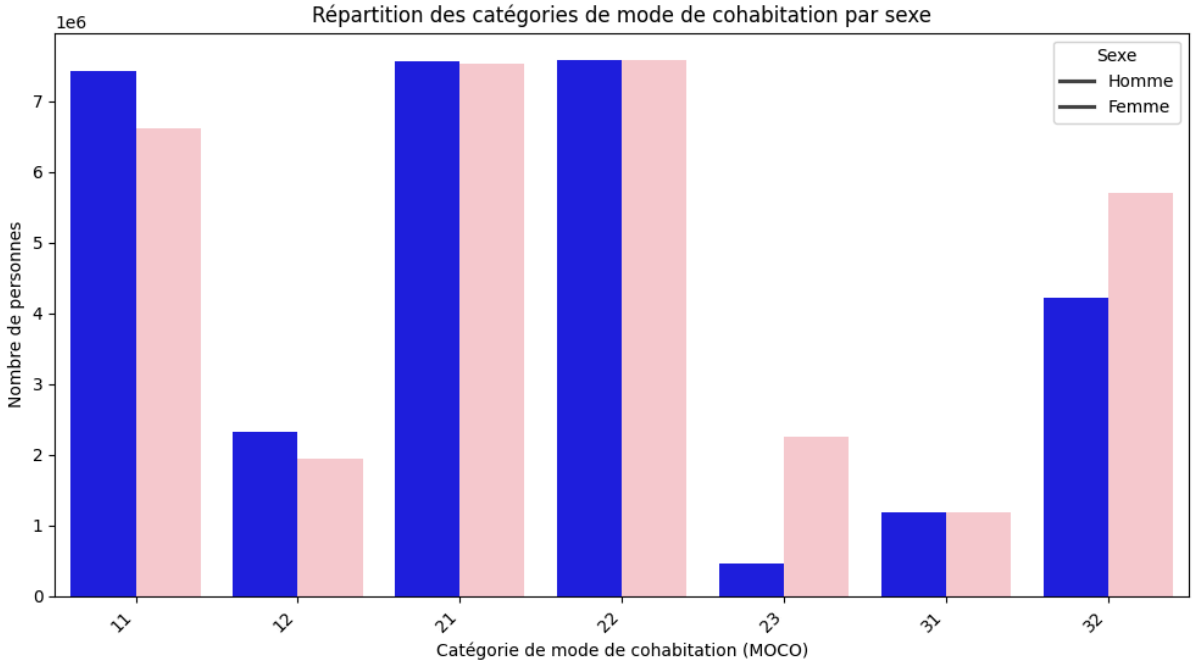


7. Salaire par sexe et catégorie d'emploi

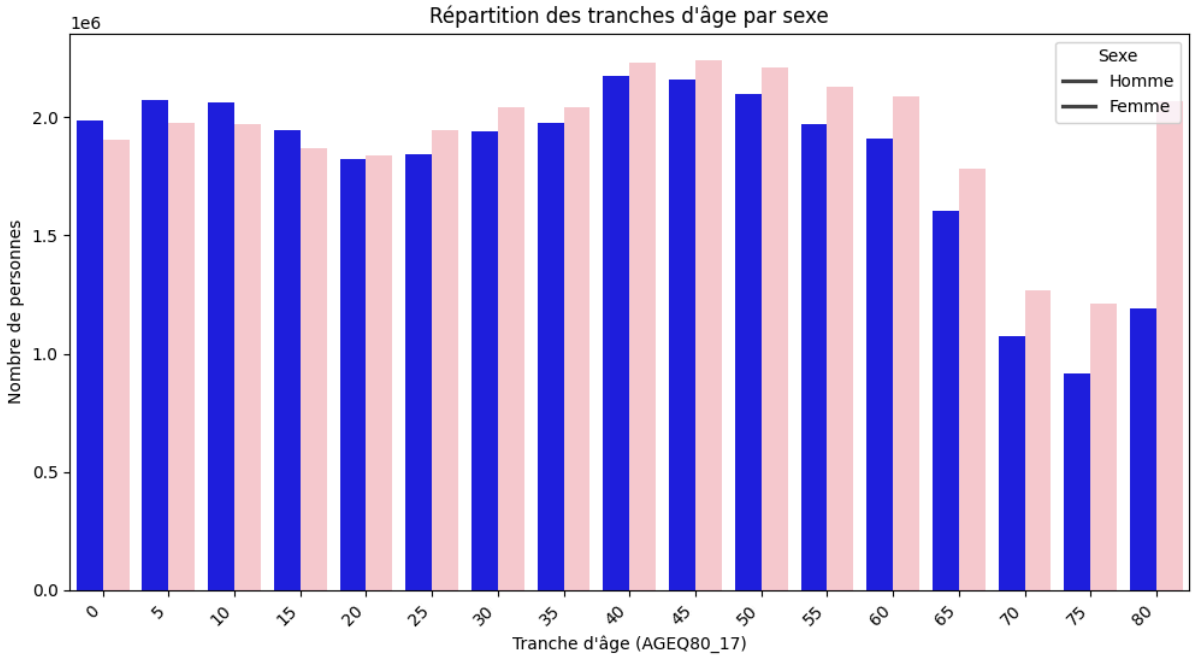
Enfin, le graphique ci-dessous, combinant les variables sexe et catégorie d'emplois, apporte une nuance sur l'écart entre employés et travailleurs. Ainsi, les travailleuses sont moins bien payées que les employées en moyenne, alors qu'en moyenne un travailleur est mieux payé qu'un employé. Pour le reste, la différence entre catégories d'emploi suit quasiment la même courbe qu'il s'agisse d'hommes ou de femmes.



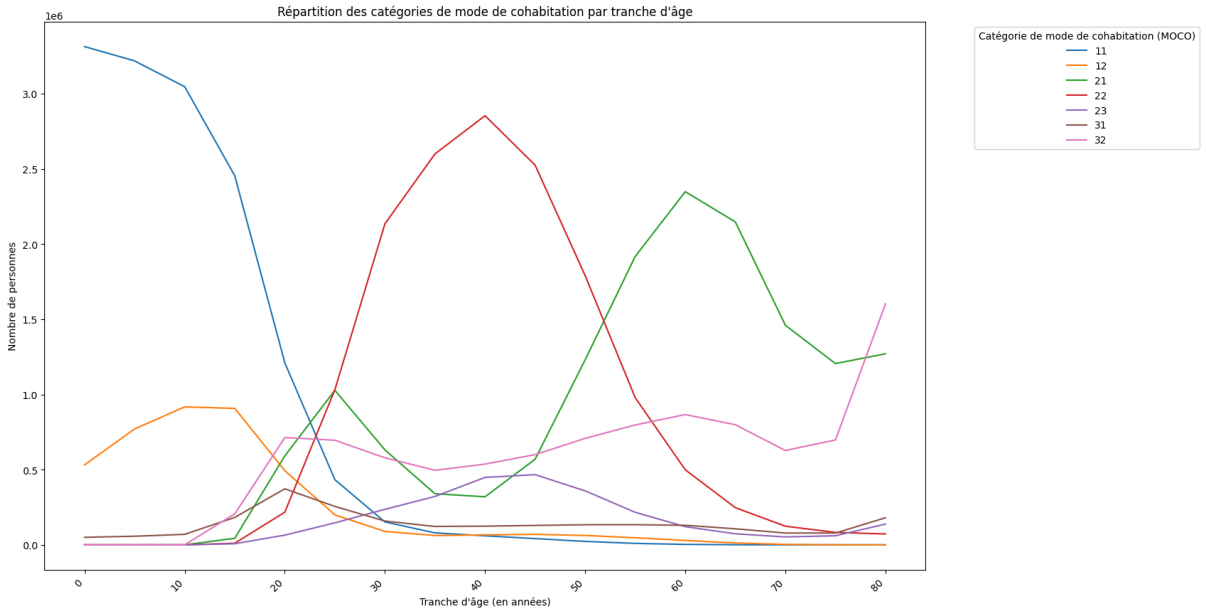
8. Répartition de la population :



Ce graphique présente la distribution des modes de cohabitation les plus habituels dans la population française, à savoir le couple, le couple avec enfants et les parents avec enfants. Les catégories de mode de cohabitation 21 (adultes vivant en couple sans enfant) et 22 (adultes vivant en couple avec enfants) présentent quasiment la même répartition entre l'homme et la femme : autour de 7,5 millions de personnes. Les catégories 12 (enfants vivant avec un seul parent), 23 (adultes vivant seuls avec enfants)et 31 (personnes étrangères à la famille vivant au foyer) sont les moins représentées dans ce dataframe. 2,5 millions de femmes vivent seules avec enfants tandis que les hommes sont environ 500 000 dans cette situation. La catégorie 11, qui indique les enfants vivant avec deux parents est aussi bien représentée avec 7,3 millions d'individus de type masculin et 6,5 millions d'individus de type féminin.

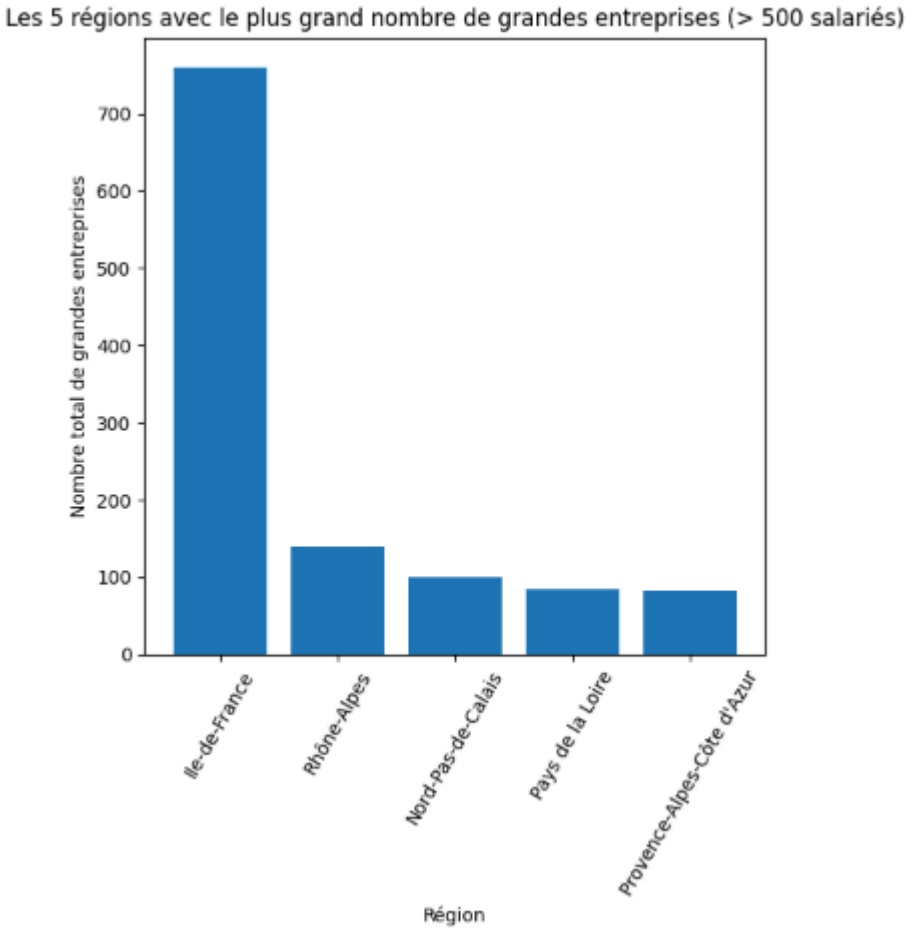


La figure ci-dessus indique la proportion d'hommes et de femmes par tranches d'âge. De 0 à 15 ans, on peut constater que les hommes sont légèrement plus nombreux que les femmes. Ensuite la tendance s'inverse et s'intensifie de 25 à 80 ans. Cette différence est flagrante pour la tranche d'âge 80-84 ans, dans laquelle les femmes représentent 2 millions de personnes et les hommes 1,2 millions. On peut penser que le taux de mortalité touche davantage les hommes à cet âge et que l'espérance de vie est plus importante chez les femmes. Des facteurs perturbateurs majeurs influençant la distribution des âges et des sexes sont probablement à l'œuvre, ce qu'il conviendrait de vérifier dans le cadre d'analyses plus poussées, que ne nous permettent pas pour l'heure les données à notre disposition.

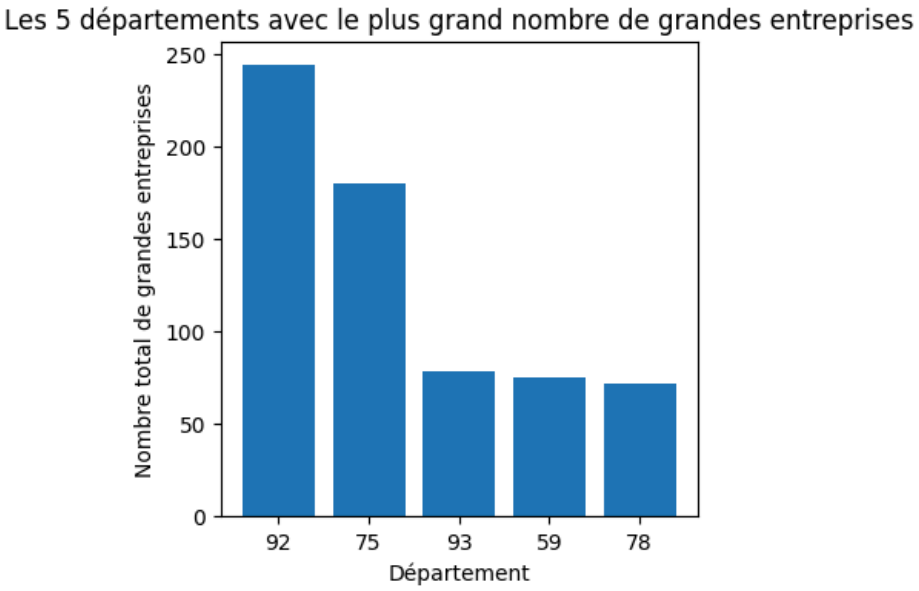


Comme le montre la figure ci-dessus, les différentes catégories de cohabitation évoluent avec l'âge. Par exemple, on peut constater que la catégorie 22 (adultes vivant en couple avec enfants) est la plus représentée et est la situation la plus fréquente entre 20 et 50 ans. Cela semble correspondre à un schéma typique de parents dont les enfants quittent le foyer familial dans cette tranche d'âge. La répartition de l'âge dans la catégorie 11 (enfants vivant avec deux parents) semble également logique. En effet, le graphique montre que ces enfants quittent le foyer à partir de 20 ans. La courbe de la catégorie 21 (adultes vivant en couple sans enfant) montre deux pics à 25 ans et 65 ans. La catégorie 32 (personnes vivant seules) stagne de 20 à 75 ans et après cet âge augmentent considérablement.

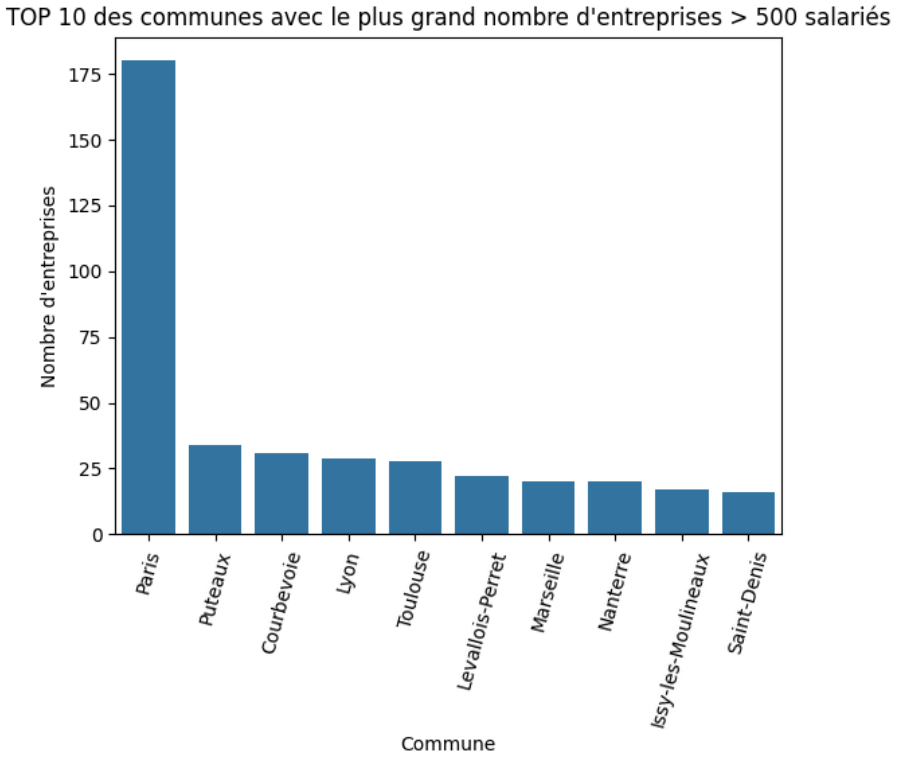
9. Répartition des grandes entreprises en France (avec plus de 500 salariés)



L'analyse de la répartition des entreprises par taille sur le territoire français laisse entrevoir différents *patterns*. Premièrement, on observe sans surprise que la région Ile de France est celle qui concentre le plus grand nombre de très grosses entreprises.

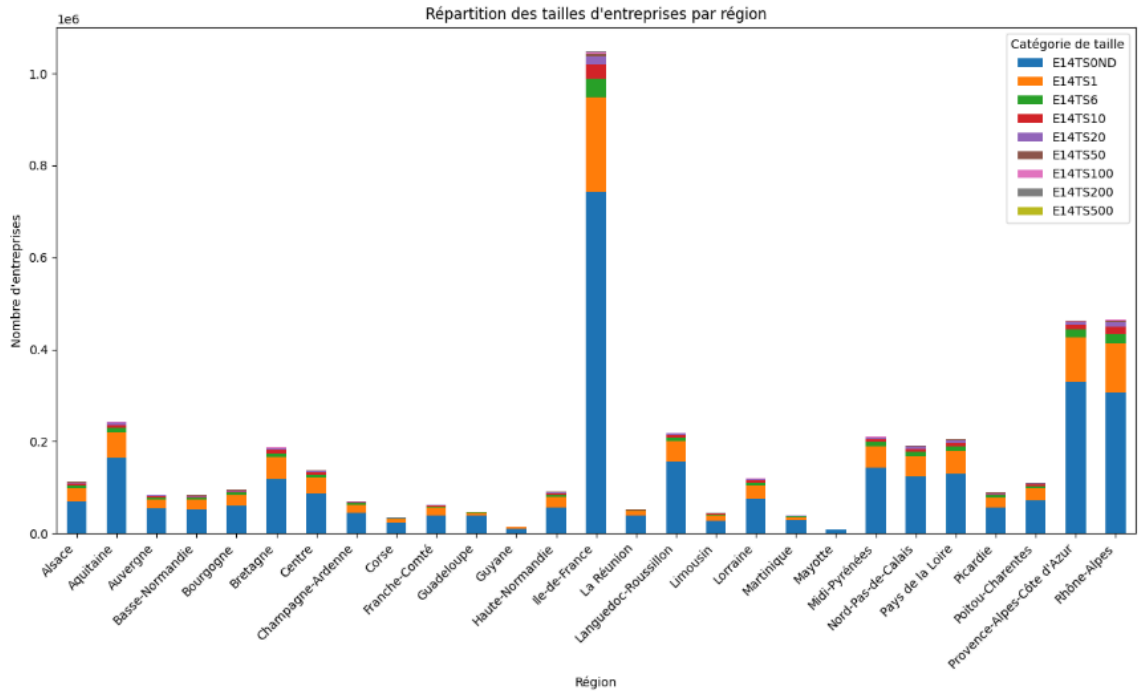


C'est donc assez logiquement que l'on retrouve majoritairement des départements d'Île-de-France dans le classement présenté ci-dessus. Néanmoins, le Nord, traditionnellement un bastion industriel, est également présent dans ce top 5.



Les Hauts-de-Seine (92), qui arrivent en tête des départements avec le plus de grandes entreprises, accueille le centre d'affaires de la Défense à Puteaux, Courbevoie et Nanterre. On retrouve donc logiquement ces communes, dans le top 10 des villes françaises avec le nombre le plus élevé de grandes entreprises. Dans le reste du classement, on note la présence des trois plus

grandes villes françaises, Paris, Lyon et Marseille, ainsi que de localités de la petite couronne parisienne.

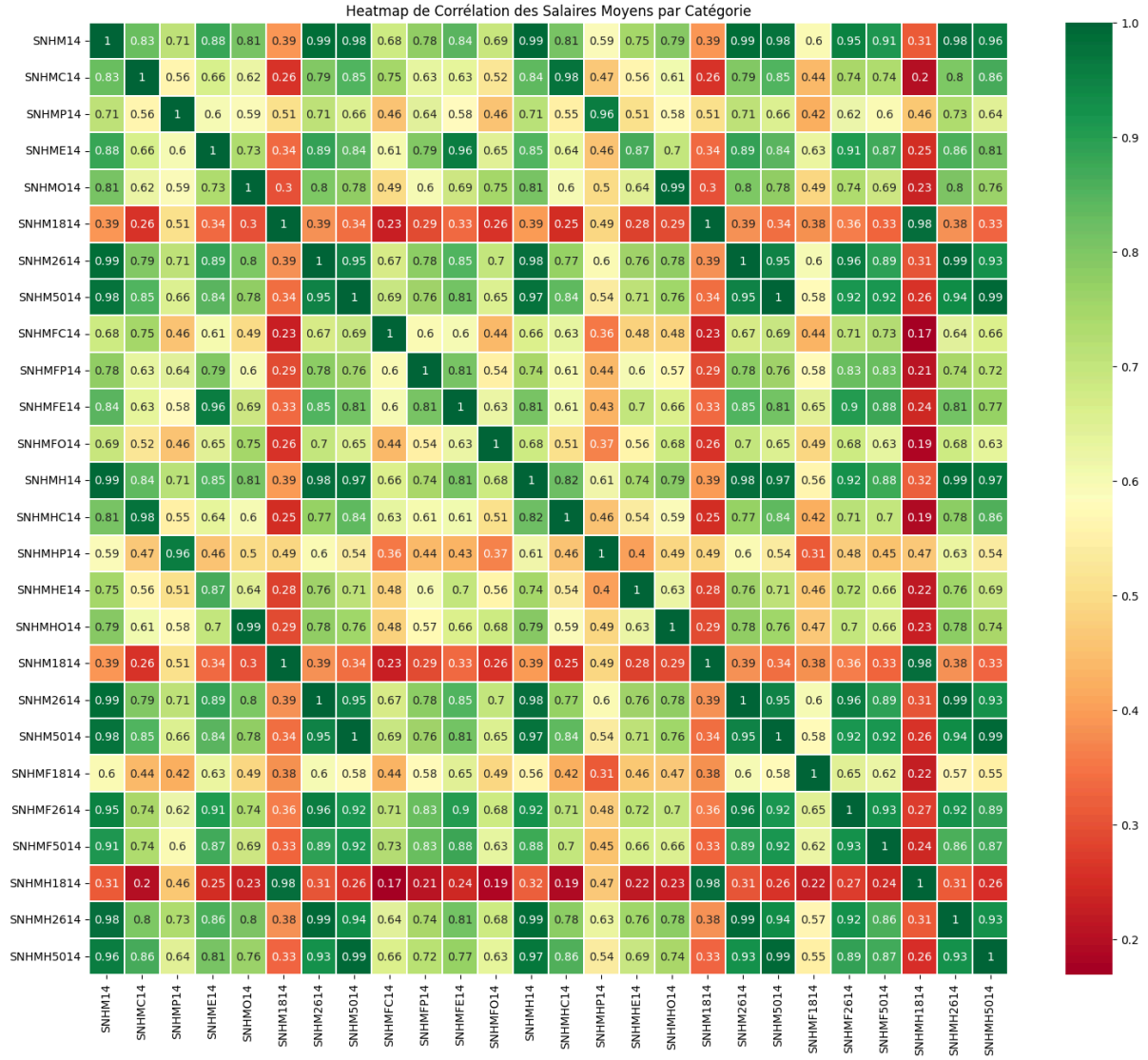


Enfin, lorsque l'on regarde la répartition des tailles d'entreprises par région, présentée ci-dessus, on remarque que la région Ile-de-France est de loin la région la plus peuplée en entreprise, suivie par la région Rhône-Alpes et la région PACA.

La structure de la répartition des entreprises semble a priori assez similaire dans toutes les régions. On obtient une pyramide avec à la base les très nombreuses petites entreprises et en haut les grosses entreprises, beaucoup plus rares.

Il convient également de noter que les entreprises de taille "inconnue ou nulle" selon la formulation de l'Insee (colorée en bleu dans le graphique), sont majoritaires. On peut émettre l'hypothèse que cette catégorie regroupe majoritairement des statuts particuliers de type auto entreprise sans salariés.

10. Heatmap



Corrélations Fortes (0.7 et plus)

- **SNHM14 (salaire net moyen) avec** :
    - **SNHM2614 (26-50 ans)** : Corrélation de 0.99, très forte. Indique que les salaires des 26-50 ans influencent fortement le salaire net moyen.
    - **SNHM5014 (>50 ans)** : Corrélation de 0.98, très forte. Indique que les salaires des plus de 50 ans influencent également fortement le salaire net moyen.
    - **SNHMC14 (cadres)** : Corrélation de 0.83, forte. Les salaires des cadres sont un composant majeur du salaire net moyen.
    - **SNHME14 (employés)** : Corrélation de 0.88, forte. Les salaires des employés sont également un composant majeur du salaire net moyen.

    - **SNHMO14 (travailleurs)** : Corrélation de 0.81, forte. Les salaires des travailleurs contribuent également significativement.

Corrélations Modérées (0.5 à 0.7)

- **SNHMP14 (cadres moyens) avec** :
    - **SNHM14** : Corrélation de 0.71, modérée. Les salaires des cadres moyens influencent le salaire net moyen mais moins que les cadres supérieurs.
    - **SNHMC14 (cadres)** : Corrélation de 0.56, modérée. Les salaires des cadres moyens sont modérément corrélés avec ceux des cadres supérieurs.
- **SNHM1814 (18-25 ans) avec** :
    - **SNHM14** : Corrélation de 0.39, faible. Les salaires des jeunes influencent moins le salaire net moyen.

Corrélations Faibles (moins de 0.5)

- **SNHM1814 (18-25 ans) avec** :
    - La plupart des autres catégories montrent des corrélations faibles, suggérant que les salaires des jeunes sont déterminés par des facteurs différents par rapport aux autres groupes d'âge et catégories.

Salaires fortement corrélés : Les salaires des tranches d'âge plus élevées (26-50 ans et >50 ans), des cadres, employés et travailleurs sont fortement corrélés avec le salaire net moyen, indiquant qu'ils contribuent de manière significative à la détermination du salaire global.

Disparités pour les jeunes (18-25 ans) : Les salaires des jeunes montrent des corrélations plus faibles avec les autres catégories, suggérant que leurs salaires sont influencés par des facteurs différents.

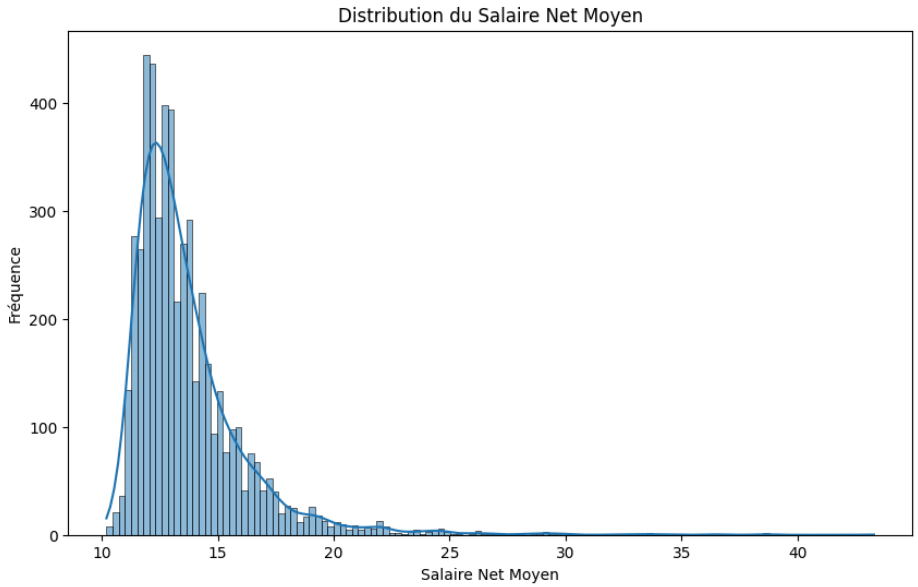
Importance des catégories de postes : Les cadres, employés et travailleurs montrent des corrélations fortes entre eux, ce qui indique une certaine cohérence dans la manière dont les salaires sont distribués parmi ces catégories.

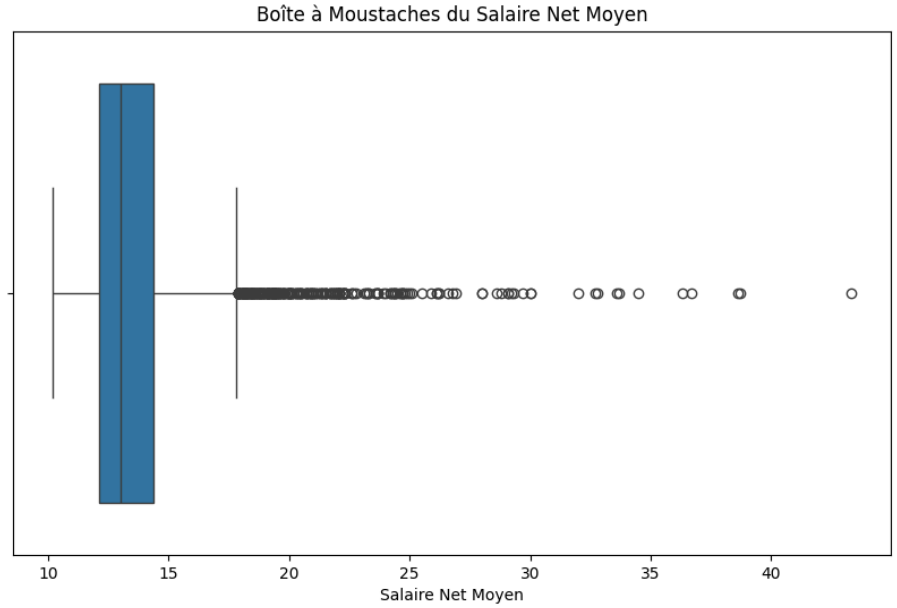
11. Test statistique

Test de normalité pour SNHM14 (salaire net moyen) :
Statistics=0.755, p=0.00000
L'échantillon ne semble pas gaussien (rejet de H0) avec Statistics=0.755, p=0.00000

Les résultats indiquent que l'échantillon de la variable SNHM14 (salaire net moyen) ne suit pas une distribution normale. La valeur p est extrêmement petite (0.00000), ce qui signifie qu'il y a une probabilité quasi nulle que l'échantillon suive une distribution normale. La statistique de test Shapiro-Wilk est de 0.755. Cette valeur est utilisée pour évaluer la normalité de l'échantillon. Des valeurs proches de 1 indiquent généralement une distribution normale.

Les figures suivantes (histogramme et boite à moustache) permettent de rendre plus lisibles les résultats du test statistique.





En effet, ces visualisations confirment les résultats du test de Shapiro-Wilk, qui indique que les données ne suivent pas une distribution normale.
L'asymétrie positive et les nombreux *outliers,* ou valeurs extrêmes, montrent une distribution non gaussienne.

II. Etapes de preprocessing des datasets en vue du ML

Pour préparer les données à une modélisation et en optimiser l'interprétabilité, nous avons suivi les principales étapes suivantes (Codes et détails consultables dans le Notebook 0*3_preProcessing*).

1. Merge des jeux de données :

Après une analyse de chacun des dataset, il est apparu que les 4 Datasets pouvaient être **mergés** ensemble grâce à une colonne commune contenant le code géographique de la ville, généralement appellée "CODGEO", mais aussi code_insee dans l'un des datasets.

Pour ce faire, nous avons **renommé** la colonne pré-citée et en avons **unifié** les valeurs. En effet, dans certains datasets, pour la même ville, les codes géographiques avaient été encodés avec un zéro initial mais pas dans d'autres.

Nous avons supprimé les lignes **doublons**, qui posaient problème pour le merge, ne conservant que la première occurrence. Il s'agissait généralement des mêmes informations géographiques pour une même ville.

Le dataset "salaire" étant celui qui contenait notre variable cible, nous avons choisi d'en conserver toutes les lignes (le jeu de donnée initial contenant une ligne par CODGEO) et d'y agréger avec la méthode "merge" les autres informations (taille des entreprises, informations géographiques, répartition de la population etc).

2. Gestion des NaNs

Le CODGEO 61483 (qui correspond à Bagnoles de l'Orne en Normandie) présent dans le dataset salaire n'existait pas dans le dataset name_geographic_information, et a donc créé une ligne de NaN. Nous avons supprimé cette ligne, estimant que, comme il ne s'agit pas d'une ville majeure, cela n'impacterait pas significativement notre analyse. Les lignes de name_geographic_information où la valeur de CODGEO n'avait pas d'équivalent dans salaire ont été supprimées car elles n'apportaient aucune information sur le salaire (notre variable cible). Au final, notre dataframe df contenait donc **5135 lignes et 42 colonnes** (contre 5136 lignes dans notre dataset initial salaire, qui comprenait notre variable cible).

Dans cinq colonnes (correspondant aux informations provenant du dataset population), des NaNs sont apparus pour 29 lignes de notre jeu de données final. Il s'agissait de lignes correspondant à un code géographique présent dans le dataset salaire, mais pas dans population. Ayant remarqué que la majorité de ces lignes provenaient d'une même région (CODGEO commençant par 49), nous avons décidé de conserver les NaNs à ce stade afin de ne pas déséquilibrer notre jeu de données, et de les traiter plus tard, dans le cadre de notre modélisation de Machine Learning.

3. Numérisation des colonnes :

Pour obtenir un jeu de données exploitable par un modèle de Machine Learning, nous avons **numérisé** l'ensemble des variables, dont certaines étaient en format "string". Cela a été fait à l'aide de la méthode "replace" et de dictionnaires (de type : 'Lyon': '1', etc), consultables dans les notebooks.

La méthode a été un peu plus spécifique pour la colonne CODGEO, qui contenait certains codes non-numériques, distinguant les villes de Corses du Nord et du Sud (commençant respectivement par 2A et 2B). Ne souhaitant pas conserver ce niveau de distinction, A et B ont été remplacés par "0" (2A001 est par exemple devenu 20001). Cette méthode de réencodage avait d'ailleurs été utilisée par l'Insee dans la colonne code_insee de l'un des datasets, ce qui facilitait la correspondance.

4. Création de colonnes

Nous avons procédé à la **création de plusieurs colonnes** en regroupant des informations qui étaient à l'origine trop segmentées pour être interprétables. Notamment, alors que le dataset population était organisé avec une colonne "NB" reprenant le nombre de personnes puis de nombreuses catégories (âge par groupe de 5 ans etc), nous l'avons réorganisé à l'aide d'un **groupby** pour le rendre plus lisible et cohérent avec les autres jeux de données.

Nous avons créé les colonnes "hommes", "femmes", "15_24ans", "24_49ans", et "50_plus_ans" dont la valeur correspond pour chaque code géographique au nombre de personnes de ces catégories. De même, pour optimiser l'interprétabilité de nos résultats, nous avons choisi de regrouper les entreprises (divisées en 8 tailles différentes dans le jeu de données initial) en 4 catégories : "micro_entreprises", "petites_entreprises", "moyennes_entreprises" et "grandes_entreprises".

Ces catégories ont été choisies en utilisant, dans la mesure du possible, les seuils officiels utilisés par l'administration publique française (source : https://entreprendre.service-public.fr/actualites/A17100).

Il faut ici noter que nous avons adapté ces seuils à nos données : ainsi, alors que le seuil officiel en France pour une grande entreprise est en réalité de 250 salariés, les données de l'Insee à notre disposition marquaient la distinction à 200 ou 499 salariés. Nous avons donc choisis de considérer comme "grande entreprise" les entreprises de 200 salariés ou plus. Cela a pour effet une légère surévaluation du nombre de grandes entreprises dans notre jeu de données.

Pour plus de lisibilité, la colonne comprenant le "nombre d'entreprises de taille inconnue ou nulle dans la ville" a été renommée "autres_entreprises". Ne disposant pas de davantage d'information sur le contenu de cette colonne malgré nos recherches, nous émettons l'hypothèse qu'elle pourrait représenter en majorité les auto-entreprises n'embauchant pas de salariés.

5. Sélection et suppression des colonnes

Les colonnes inutiles et/ou redondantes ont été **supprimées**, comme par exemple LIBGEO (nom de la ville), qui correspondait sous format "string" à la variable CODGEO, davantage adaptée au format numérique en vue d'une problématique de Machine Learning. Des colonnes comme la

latitude, longitude et l'éloignement, qui comprenaient environ 8% de valeurs manquantes, ont été supprimées.

Enfin, nous avons procédé après analyse à une **pré-sélection** des variables pertinentes pour notre projet.
Ainsi, les colonnes correspondant à la population en-dessous de 15 ans (borne la plus proche de l'âge légal de travail, fixé en France à 16 ans) n'ont pas été retenues puisque ne comprenant pas de population active. Il est dès lors à noter que, dans notre jeu de données final, la densité de population (somme des hommes et des femmes par exemple, ou somme des catégories d'âge) s'entend pour la population active au-dessus de 15 ans. D'autres colonnes comme le mode de cohabitation, non pertinentes par rapport à notre variable cible, ont été écartées.

6. Variables (features) conservées et jeu de données final

Nous avons enfin exporté notre jeu de données final, comprenant l'ensemble des variables explicatives et cible que nous souhaitions conserver à ce stade. Il a par la suite été ré-importé dans le cadre de notre modélisation et les éventuelles modifications ultérieures ont été réalisées après séparation du jeu d'entraînement et de test, afin d'éviter toute fuite des données.

```
RangeIndex: 5135 entries, 0 to 5134
Data columns (total 42 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CODGEO                  5135 non-null   int64
 1   SNHM14                  5135 non-null   float64
 2   net_cadre               5135 non-null   float64
 3   net_cadre_moyen         5135 non-null   float64
 4   net_employe             5135 non-null   float64
 5   net_travailleur         5135 non-null   float64
 6   net_femme               5135 non-null   float64
 7   net_cadre_femme         5135 non-null   float64
 8   net_cadre_moyen_femme   5135 non-null   float64
 9   net_employe_femme       5135 non-null   float64
 10  net_travailleur_femme   5135 non-null   float64
 11  net_homme               5135 non-null   float64
 12  net_cadre_homme         5135 non-null   float64
 13  net_cadre_moyen_homme   5135 non-null   float64
 14  net_employe_homme       5135 non-null   float64
 15  net_travailleur_homme   5135 non-null   float64
 16  net_18_25               5135 non-null   float64
 17  net_26_50               5135 non-null   float64
 18  net_50_plus             5135 non-null   float64
 19  net_18_25_femme         5135 non-null   float64
 20  net_26_50_femme         5135 non-null   float64
 21  net_50_plus_femme       5135 non-null   float64
 22  net_18_25_homme         5135 non-null   float64
 23  net_26_50_homme         5135 non-null   float64
 24  net_50_plus_homme       5135 non-null   float64
 25  EU_circo                5135 non-null   float64
 26  code_région             5135 non-null   float64
 27  chef.lieu_région        5135 non-null   float64
 28  numéro_département      5135 non-null   float64
 29  préfecture              5135 non-null   float64
 30  numéro_circonscription  5135 non-null   float64
 31  total_entreprises       5135 non-null   int64
 32  autres_entreprises      5135 non-null   int64
 33  micro_entreprises       5135 non-null   int64
 34  petites_entreprises     5135 non-null   int64
 35  moyennes_entreprises    5135 non-null   int64
 36  grandes_entreprises     5135 non-null   int64
 37  hommes                  5106 non-null   float64
 38  femmes                  5106 non-null   float64
 39  15_24ans                5106 non-null   float64
 40  24_49ans                5106 non-null   float64
 41  50_plus_ans             5106 non-null   float64
dtypes: float64(35), int64(7)
memory usage: 1.6 MB
```

III. Machine Learning

1. Sélection de modèles de Machine Learning

Notre objectif de modélisation est de prédire le salaire net moyen par heure selon différentes *features,* ou variables explicatives retenues à ce stade (nombre d'hommes et de femmes par ville, salaire par catégorie socio-professionnelle, densité de population, etc). En plus de la prédiction rendue possible par le Machine Learning, cela nous permet également de déterminer quelles variables ont le plus d'impact, positif ou négatif, sur le salaire moyen, afin de pouvoir comparer les observations retenues par les modèles avec nos premières visualisation des données, présentées plus tôt dans ce travail. Il s'agit donc dans ce cas d'une problématique de régression linéaire et nous choisissons les modèles de Machine Learning suivant, qui nous semblent les plus adaptés :

- LinearRegression
- Lasso
- DecisionTreeRegressor
- GradientBoostong
- RandomForestRegressor
- Ridge

Pour l'ensemble des modèles suivant, nous réalisons les étapes classiques de préparation au Machine Learning à savoir :

- Isoler la variable cible ("SNHM14", dans notre cas) de notre jeu de données
- Séparer en un jeu de données d'entraînement et un jeu de test (sur base de 20% et d'un random_state fixé à)
- Gestion des NaNs créés par la fusion des jeux de données : 21 lignes à traiter sur X_train et 8 sur X_test, sur 5 colonnes : "hommes", "femmes", 15_24ans, 24_49ans, 50_plus_ans
- Etape éventuelle, selon le modèle, de mise à l'échelle des colonnes pertinentes (donc toutes à l'exception des codes régions, numéro de région etc), via par exemple une normalisation ou Z-standardisation.
- Entrainement du modèle
- Affichage des prédictions et des "features importance"
- Evaluation des performances par l'accuracy (R2), et les métriques MAE, MSE, RMSE
- Ajustements et optimisation éventuelle du modèle

2. Linear Regression

Caractéristiques du modèle : Réalise une régression linéaire (établit une approximation linéaire entre la variable cible et les variables explicatives) qui cherche à minimiser la différence entre les prédictions de la cible et les vraies valeurs de celle-ci.

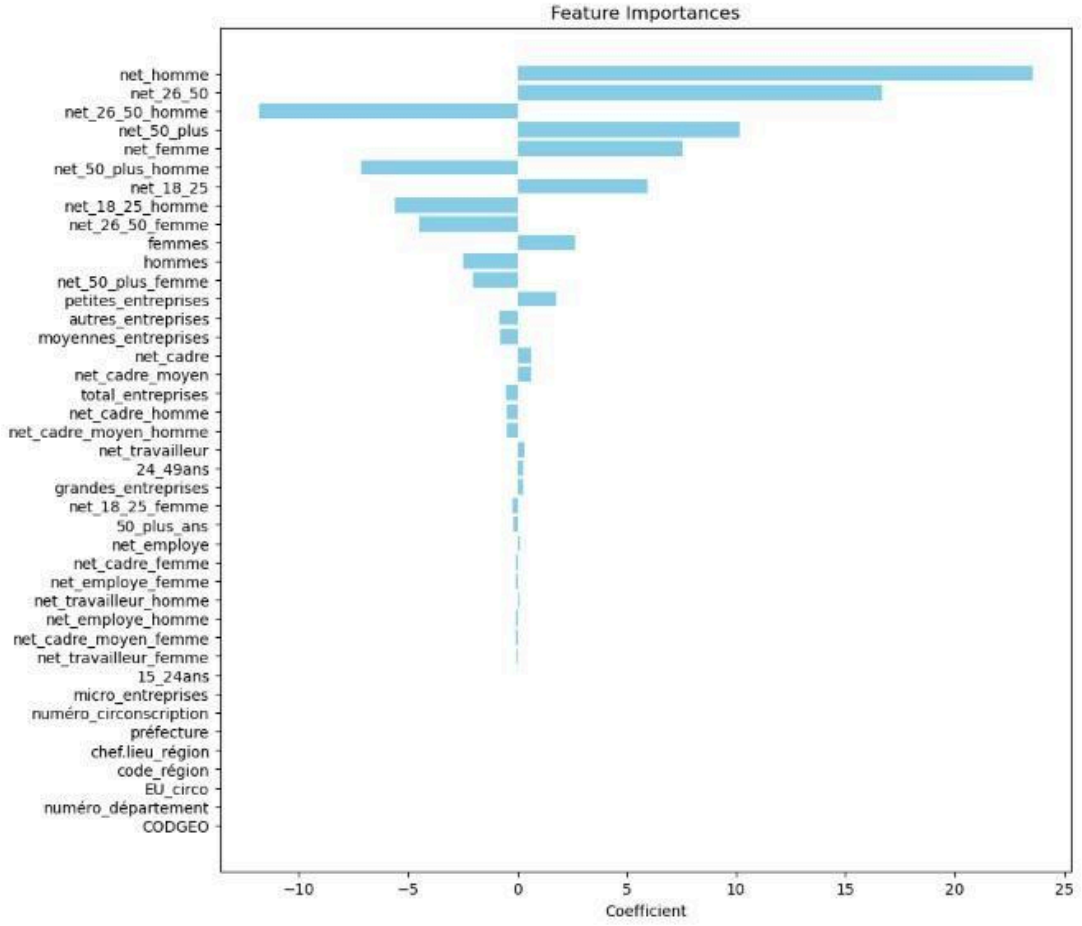
Notes sur le preprocessing :

- NaNs remplacés ligne par ligne par la médiane des valeurs pour une région donnée (tri par la colonne "code_région")

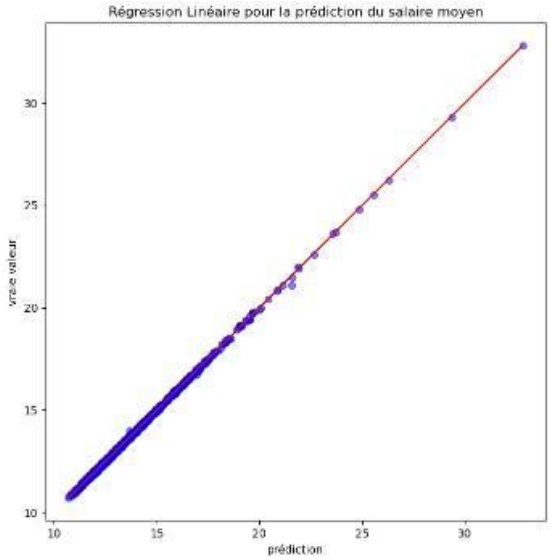
- Normalisation entre 0 et 1 avec un MinMaxScaler()

a. Premiers résultats

Les premiers résultats du modèle montrent, comme on le voit ci-dessous, un problème majeur de surapprentissage avec un score d'accuracy proche de 1 et un modèle qui prédit toujours juste. Cela est évidemment irréaliste et on peut estimer que le modèle ne parviendra pas à s'adapter à des données inconnues. Quand on regarde le Feature importances, on se rend compte que les 9 variables les plus utilisées par le modèle comprennent des informations sur notre variable cible (net_homme par exemple contient de l'information sur le salaire net moyen). Ces résultats sont liés à la structure de notre jeu de données qui comprend un important biais. Cela pose un problème car le modèle a accès à ces informations et prédit donc trop facilement.

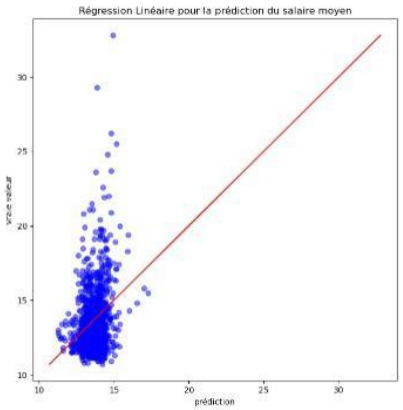


	Metric	Train Score	Test Score
0	R2	0.9996730600115881	0.9994713489808192
1	MAE	0.03779460716072061	0.0394264172399338
2	MSE	0.002271180939658528	0.002620229290562866
3	RMSE	0.04765690862465303	0.05118817530018887



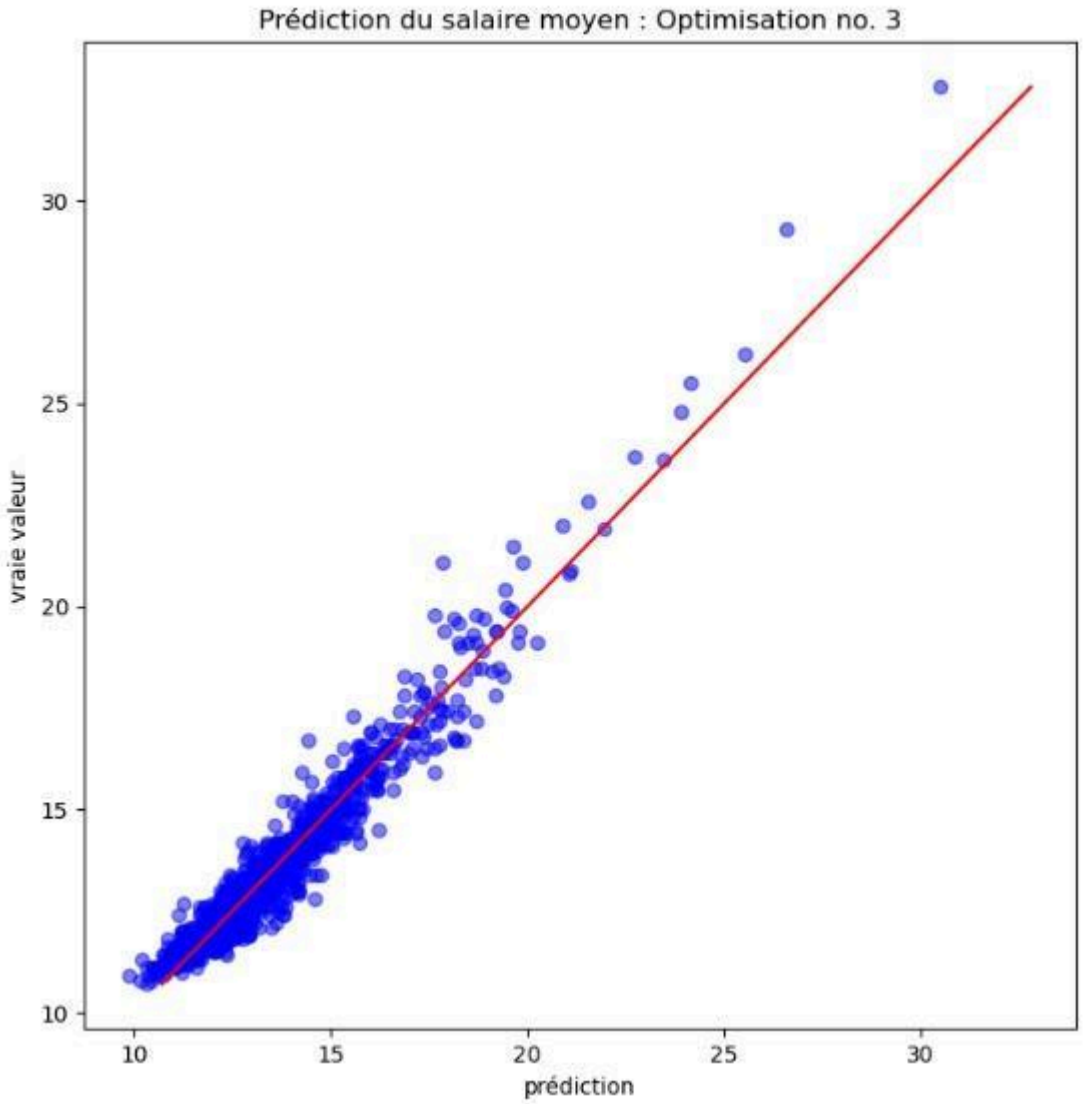
b. Résultats sans biais

Afin de tenter de supprimer tous biais, nous supprimons toutes les variables contenant le terme “net” (car il s’agit de toutes celles qui contiennent une partie de notre variable cible et pourraient fausser le modèle). Avec une accuracy de 0.05 sur le jeu de test, nous avons un modèle qui ne fonctionne pas. Sans les variables contenant de l’information directe sur le net, il ne reste pas assez de variables explicatives à notre modèle pour prédire correctement, comme on le voit ci-dessous. Cela résume le problème lié à notre jeu de données : les variables explicatives (par exemple la catégorie socio-professionnelle) sont imbriquées dans la variable cible, avec par exemple la colonne “net_cadre”.

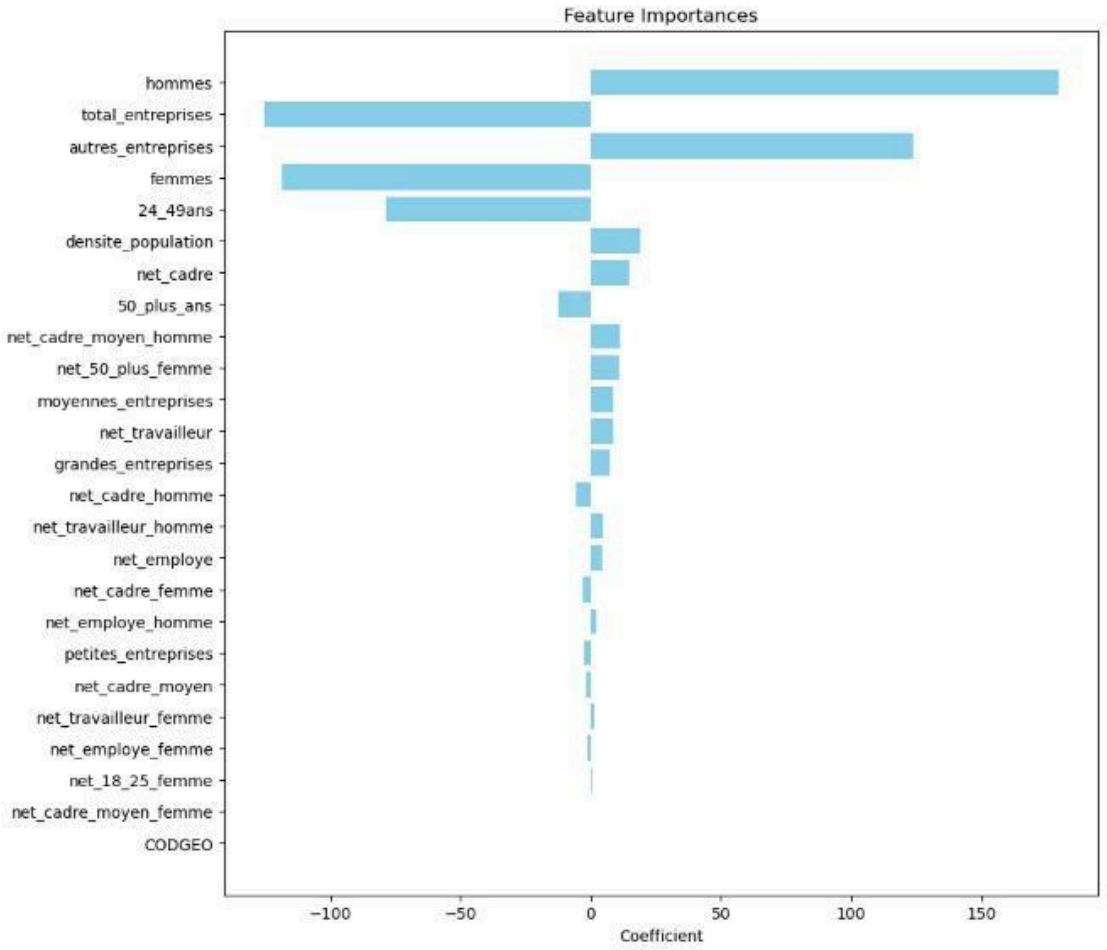


c. Optimisation du modèle LinearRegression

Pour optimiser notre modèle et le rendre performant tout en limitant le surapprentissage, nous décidons alors de supprimer d’une part les variables non-utilisées par le modèle afin de réduire le bruit (code_région etc), et celles qui sont trop fortement corrélées à notre variable cible. Afin de limiter la fuite d’information inhérente à la structure de notre jeu de données, nous choisissons de supprimer toutes les variables (9) qui sont davantage corrélées que la variable “femmes”. Celle-ci est en effet la variable ne contenant aucune information sur du salaire, à être considérée comme la plus importante par notre modèle.



Comme on le voit, la capacité de prédiction du modèle est plus réaliste, les résidus sont situés autour de la ligne ce qui indique une forte performance, mais sans non plus coller à la ligne. En outre, on peut noter que le modèle a tendance à légèrement sous-estimer les valeurs extrêmes au-delà de 23.



	Metric	Train Score	Test Score
0	R2	0.9512728152557106	0.9389261726395862
1	MAE	0.42553969765966926	0.41192144734865666
2	MSE	0.3384971467454773	0.30270901886184265
3	RMSE	0.5818050762458826	0.5501899843343594

Malgré de bonnes performances de prédiction, l'analyse des métriques indique toujours un problème de surapprentissage qui semble insoluble en raison de la nature de notre jeu de données. Nous arrivons soit à un modèle non-performant (point b), soit à un modèle qui surapprend (point c). La métrique MAE nous apprend que le modèle se trompe en moyenne de 0.41 (sur l'échelle du salaire) ce qui est une très bonne performance.

Concernant l'analyse des résultats, on voit que le nombre d'hommes est très fortement positivement corrélé au salaire, et le nombre de femmes négativement, ce qui corrobore nos premières observations. Le nombre total d'entreprises est également une variable explicative majeure du modèle, qui semble indiquer que plus il est élevé dans une ville, moins le salaire est élevé. On peut émettre l'hypothèse que cette situation met la population active en concurrence, ce qui pourrait tirer les salaires vers le bas par la loi de l'offre et la demande. Enfin, un nombre élevé de population entre 24 et 49 ans dans une ville a un impact négatif important sur le salaire. Là aussi, cela corrobore nos premières impressions car il s'agit de la catégorie dans laquelle on retrouve les jeunes travailleurs, or nous avons observé précédemment une

augmentation du salaire avec l'âge. En outre, la densité de population semble jouer un rôle, comme nous l'avions pressenti avec l'analyse cartographique plus tôt dans ce rapport, même si la corrélation établie par le modèle n'est pas aussi marquée que nous aurions pu l'attendre.

3. Lasso

Caractéristiques du modèle : Etablit une sélection automatique des variables explicatives les plus importantes, qu'elle favorise, et pénalise les moins importantes.

Notes sur le preprocessing :

- NaNs remplacés ligne par ligne par la médiane des valeurs pour une région donnée (tri par la colonne "code_région")
- Z-Standardisation avec un StandardScaler()

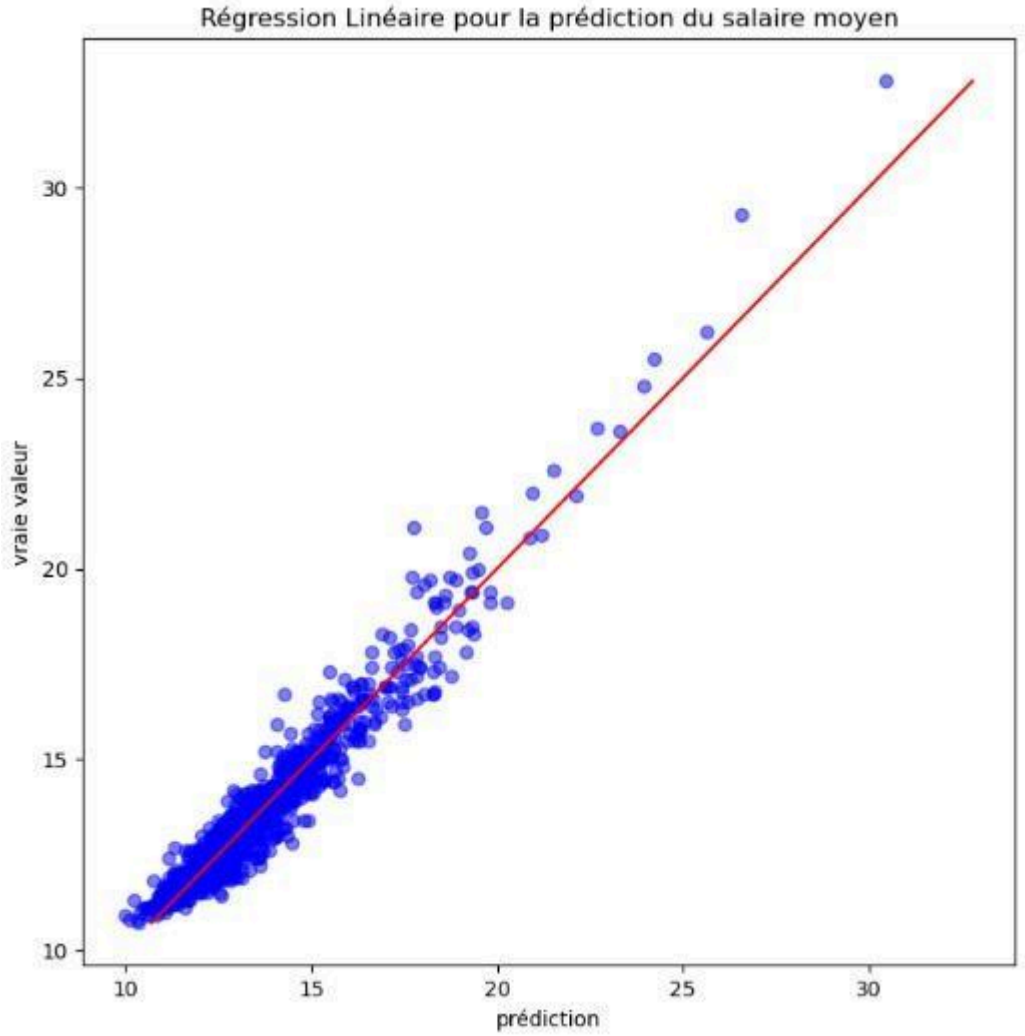
Le premier essai sur le modèle avec une normalisation des données s'avère un échec (scores R2 de 0.06 et de 0.02 respectivement sur les jeux de train et de test). Le modèle semble davantage adapté à des données standardisées, qui donnent des scores de 0.84 et 0.82 pour un paramétrage "standard".

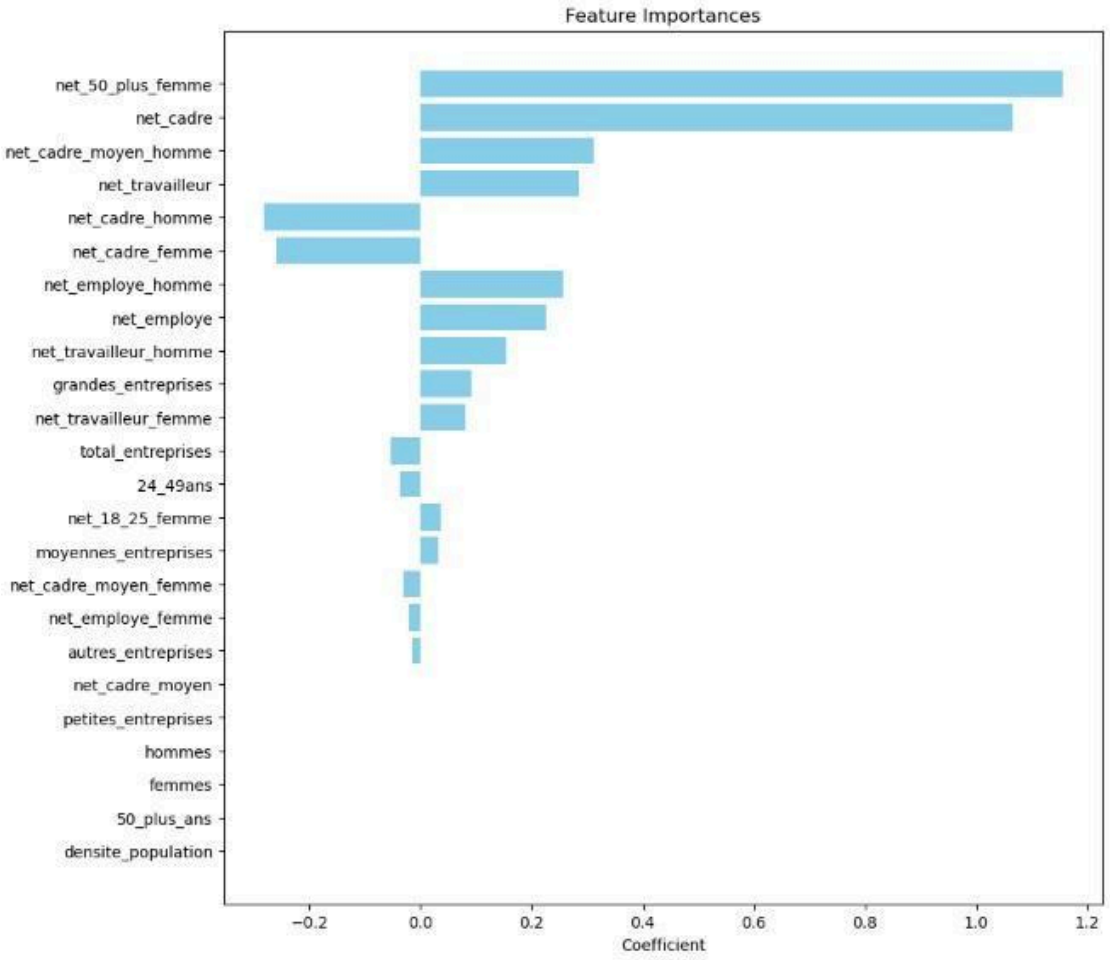
Pour tenter d'optimiser le modèle nous supprimons les 9 colonnes les plus utilisées par le modèle précédent (voir modèle 1 LinearRegression) ainsi que celles non utilisées par le modèle précédent, et utilisons un GridSearchCV pour déterminer le paramètre "alpha" (qui détermine le taux de péanlisation que le modèle attribue aux variables trop corrélées) optimal du modèle, qui est dans notre cas 0.001.

On obtient alors un modèle aux performances proches du modèle optimisé de LinearRegression, avec le même problème de surapprentissage. A l'inverse du modèle de LinearRegression, on constate par contre que le Lasso manque ici de fiabilité : en effet il utilise majoritairement (9 variables les plus importantes du modèle) des variables contenant du salaire, alors même qu'on avait déjà éliminé les variables trop corrélées.

Cela réduit l'utilité du modèle car on peut estimer qu'il s'adaptera difficilement à de nouvelles données si celles-ci ne contiennent aucune information sur du net. Par ailleurs, dans la pratique, le fait de savoir que le salaire moyen est lié au salaire moyen des hommes, des femmes, des employés etc, n'est pas réellement pertinent. En revanche, le modèle Lasso nous apprend ici que le nombre de grandes entreprises est corrélé positivement au salaire. Entre les deux modèles nous privilégierons le LinearRegression qui nous apprend davantage sur les corrélations, pour un niveau de performance similaire.

	Metric	Train Score	Test Score
0	R2	0.9492448570628822	0.9371755038582743
1	MAE	0.43154101819345647	0.41561065248261025
2	MSE	0.352584930917576	0.31138611103122094
3	RMSE	0.5937886247795389	0.5580198124002597





4. Decision Tree regressor

Il s'agit d'un modèle d'apprentissage automatique qui utilise une structure en arbre pour prédire des valeurs numériques. L'arbre de décision est construit de manière récursive, en sélectionnant à chaque étape la caractéristique qui divise le mieux les données en sous-ensembles homogènes par rapport à la variable cible, ici SNHM14 (salaire net moyen). La construction de l'arbre repose sur le choix de la caractéristique et du seuil qui minimisent le plus l'erreur de prédiction. Les feuilles de l'arbre représentent les valeurs numériques prédites pour les observations qui atteignent ces feuilles.

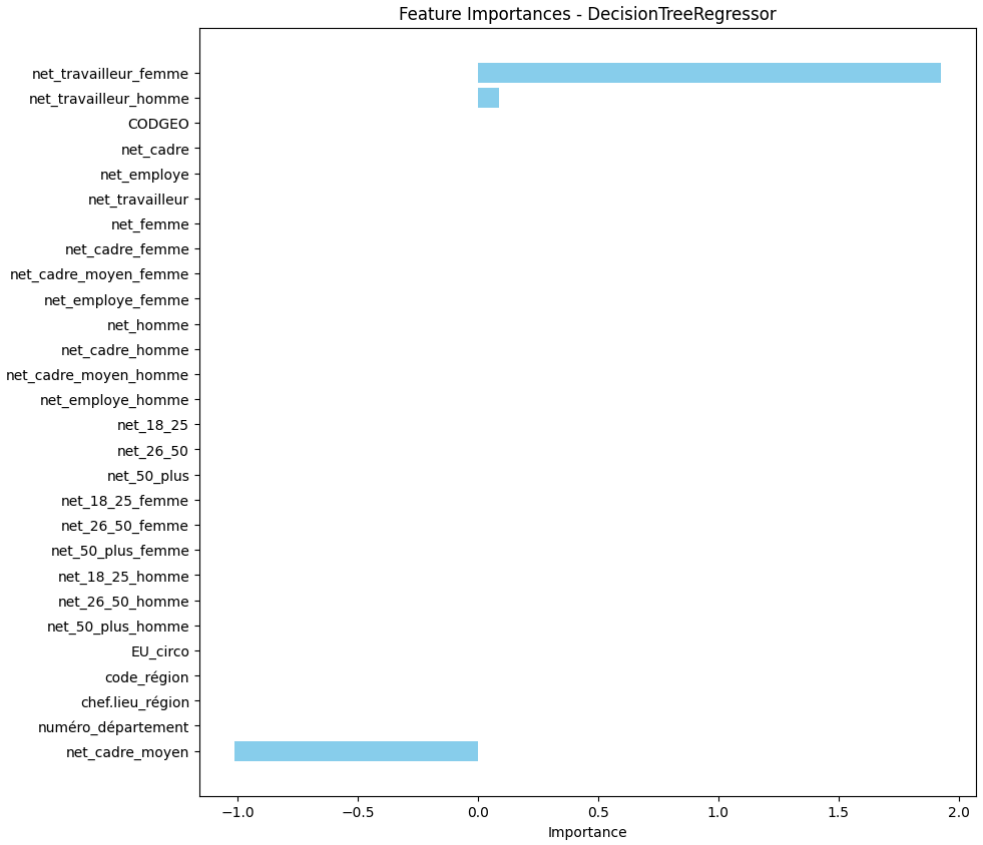
Modifications :

- Calcul de la médiane pour les régions : 25,52,53,82 qui s'applique aux colonnes : code région, hommes, femmes,15_24, 24_49, 50_plus.
- Remplacer les valeurs pour les régions indiquées ci-dessus dans X_train et remplacer les valeurs de X_test pour chaque ville (8)
- Convertir les colonnes de X_train et X_test au format (int).
- Ajouter la colonne densité (homme + femme).
- Normaliser les colonnes

Notes sur le preprocessing :

- NaNs remplacés ligne par ligne par la médiane des valeurs pour une région donnée (tri par la colonne “code_région”)
- Normalisation entre 0 et 1 avec un MinMaxScaler()

a. Premiers résultats

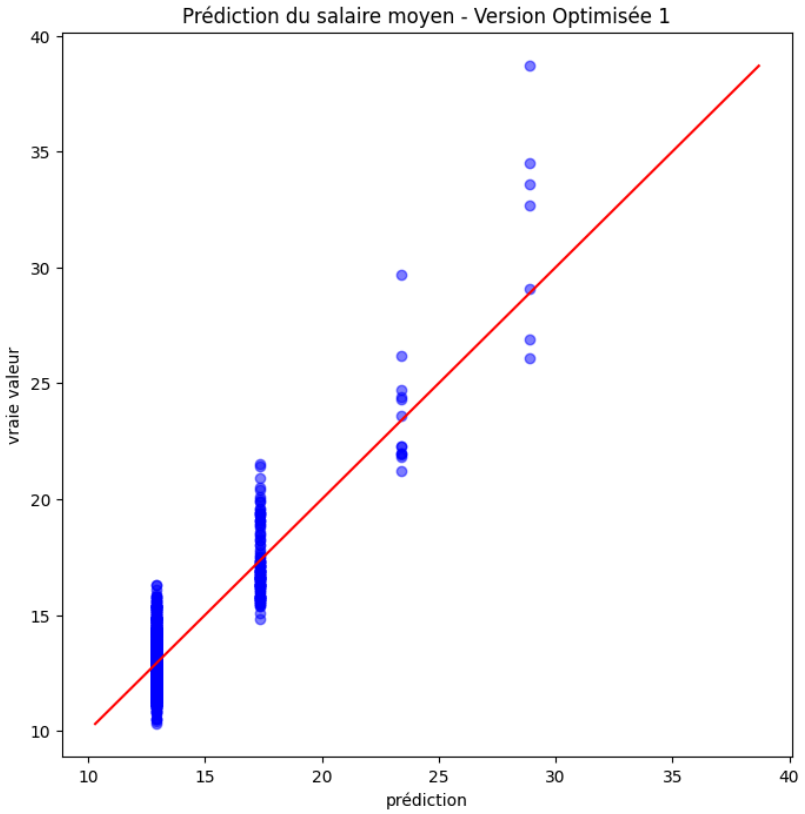


	Metric	Train Score	Test Score
0	R2	0.7171133429465026	0.7296434897553941
1	MAE	1.0427565827640601	1.0344577107375525
2	MSE	1.8151733910870207	1.9147282677359694
3	RMSE	1.3472837084619633	1.3837370659688095

- **R^2 (Coefficient de Détermination)** :
  - **Entraînement** : 0.7171 (Le modèle explique 71.71% de la variance des données d'entraînement).
  - **Test** : 0.7296 (Bonne généralisation avec une légère amélioration sur le test).

- **MAE (Erreur Absolue Moyenne)** :
    - **Entraînement** : 1.0428 (Prédictions en moyenne à 1.04 unités des valeurs réelles).
    - **Test** : 1.0345 (Prédictions légèrement plus précises sur le test).
- **MSE (Erreur Quadratique Moyenne)** :
    - **Entraînement** : 1.8152 (Indique des erreurs modérées, sensibles aux grandes erreurs).
    - **Test** : 1.9147 (MSE légèrement supérieur sur le test, mais proche de l'entraînement).
- **RMSE (Racine Carrée de l'Erreur Quadratique Moyenne)** :
    - **Entraînement** : 1.3473 (Erreur moyenne modérée).
    - **Test** : 1.3837 (RMSE légèrement supérieur sur le test).

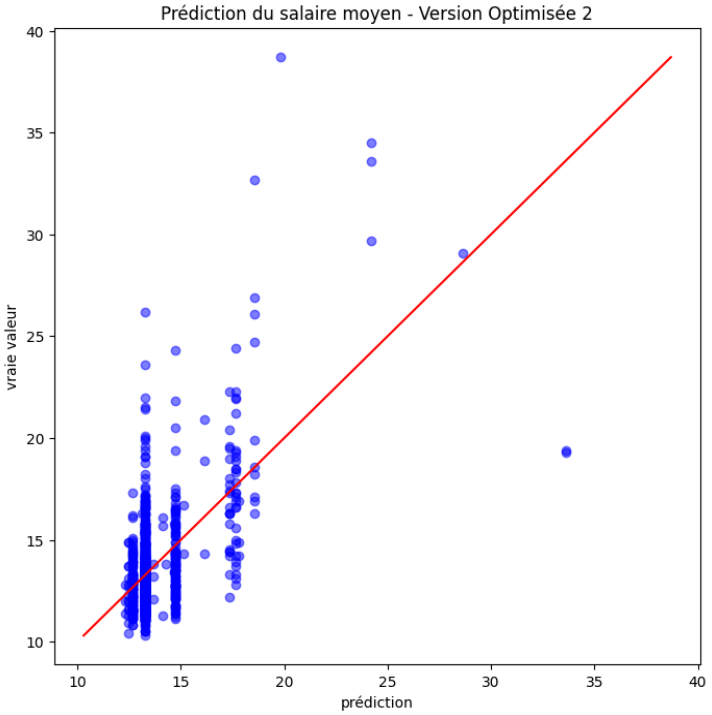
Le modèle généralise bien aux nouvelles données, avec des performances similaires entre l'entraînement et le test. Les erreurs sont faibles et cohérentes, indiquant des prédictions globalement précises. Ces résultats semblent indiquer une performance plutôt bonne du modèle. Mais une visualisation des prédictions permet d'aller plus loin dans l'analyse.



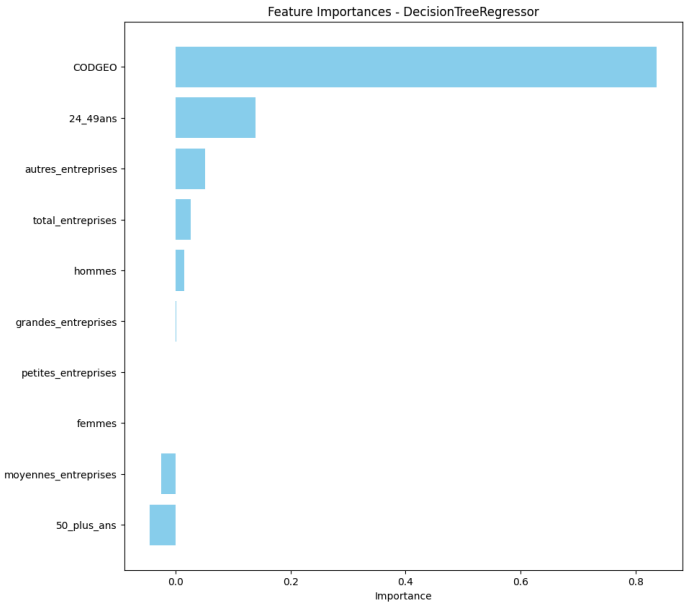
Ainsi, sur la figure ci-dessus, on observe des groupes de points à différentes plages de valeurs. Cela pourrait suggérer que le modèle capture bien certaines structures dans les données, mais il peut y avoir des limites dans sa capacité à prédire avec précision sur l'ensemble des plages de valeurs. Le scatterplot montre que le modèle fonctionne bien, avec la plupart des prédictions proches des valeurs réelles. On peut en outre constater que le modèle cherche à catégoriser, ou classifier les valeurs selon quatre groupes. Une optimisation des paramètres du modèle pour que celui-ci classifie en davantage de groupes permettrait probablement d'affiner la capacité de prédiction du modèle en la rendant plus précise.

b. Résultats sans biais

Afin d'éviter les biais, nous avons supprimé toutes les variables contenant "net", car elles sont directement liées à notre variable cible, risquant ainsi de fausser le modèle. Toutefois, la fiabilité des prédictions s'en ressent fortement et les scores R^2 montrent que le modèle explique environ 40,13 % de la variance des données d'entraînement et 33,47 % de la variance des données de test. Ces scores sont relativement faibles, ce qui suggère que le modèle ne parvient pas à bien capturer la structure des données.

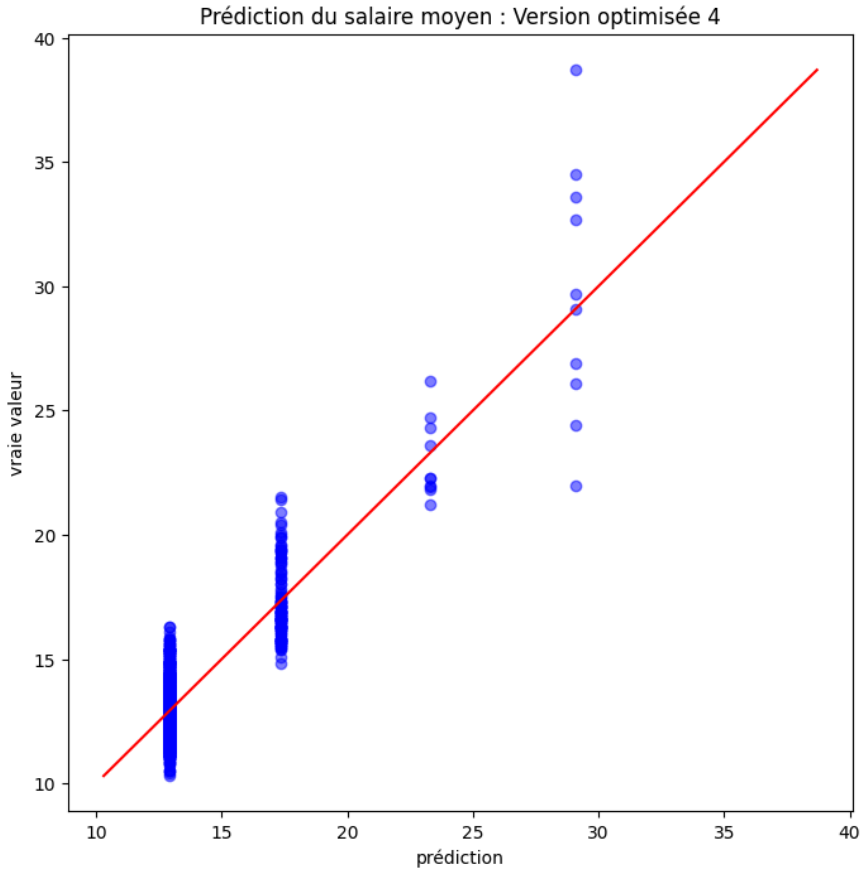


On observe que la majorité des points sont groupés dans la partie inférieure gauche du graphique, autour des valeurs de prédiction comprises entre 10 et 20. Cela suggère que le modèle fait beaucoup de prédictions dans cette plage de valeurs, ce qui pourrait indiquer une concentration de la majorité des données réelles dans cette plage.



c. Optimisation du modèle Decision Tree regressor

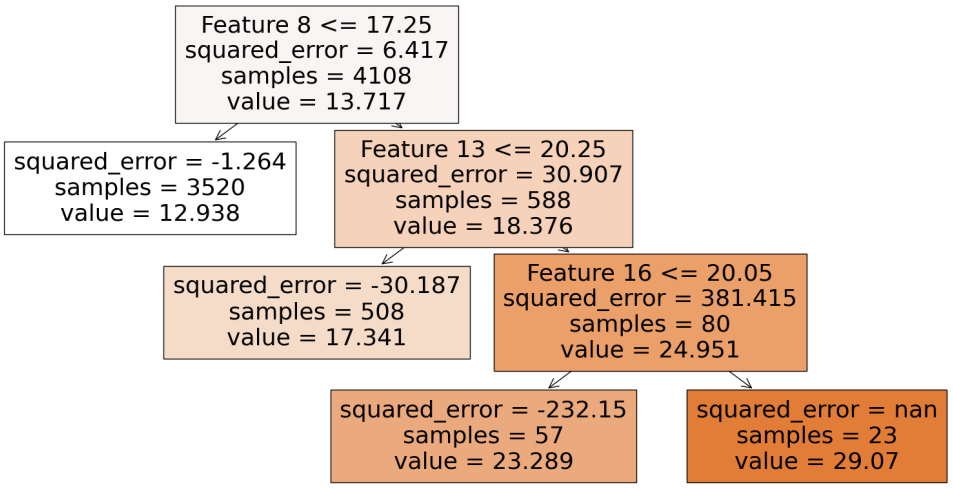
Pour améliorer notre modèle tout en évitant le surapprentissage, nous avons supprimé les variables non corrélées et non utilisées par le modèle, ainsi que celles qui étaient fortement corrélées à notre variable cible (comme "net employé/homme", "net50plus homme", "net travailleur femme"). Cette approche vise à limiter la fuite d'information due à la structure de notre jeu de données et à renforcer la pertinence des variables restantes.



Des groupes distincts de points apparaissent à différentes plages de valeurs, surtout entre 10 et 20, ainsi qu'entre 25 et 30. Cela suggère que le modèle capte certaines structures dans les données, mais il y a aussi une dispersion, notamment pour les valeurs plus élevées. Le modèle optimisé offre des performances proches de la première version du Decision Tree Regressor mise en œuvre et présentée plus haut dans ce rapport.

	Metric	Train Score	Test Score
0	R2	0.7378771212708835	0.7654628383850479
1	MAE	1.029253045174475	1.0131495156063939
2	MSE	1.68194032062192	1.6610472326795689
3	RMSE	1.296896418617123	1.2888162136936239

Le modèle est plutôt performant et bien équilibré entre l'entraînement et le test, avec des résultats très similaires sur toutes les métriques. Cela suggère que le modèle généralise bien sans signe de surajustement, offrant ainsi des prédictions fiables sur de nouvelles données.



Feature 8 : net_homme

Feature 13 : net_26_50

Feature 16 : net_26_50_femme

Comme on le constate sur la structure de l'arbre de décision, **net_homme** est la première caractéristique utilisée pour diviser les données, ce qui suggère qu'elle est très influente pour prédire la cible. **net_26_50** et **net_26_50_femme** sont également des caractéristiques importantes dans les sous-groupes, avec des seuils spécifiques utilisés pour affiner les prédictions.

Au cours de ce processus d'optimisation, nous avons réussi à améliorer de manière significative la performance du modèle de régression. En éliminant les variables non corrélées ainsi que celles trop fortement corrélées avec la variable cible, nous avons simplifié le modèle tout en préservant l'essentiel des informations nécessaires. Cette approche a aidé à éviter le surajustement et à améliorer la capacité du modèle à généraliser sur des données nouvelles.

Ensuite, nous avons affiné les hyperparamètres en utilisant des techniques comme GridSearchCV, ce qui a permis de maximiser la précision du modèle. Cette étape a été cruciale pour s'assurer que le modèle capturait correctement les tendances sous-jacentes des données.

En analysant les erreurs, nous avons pu identifier des écarts, en particulier pour les valeurs plus élevées, et effectuer des ajustements supplémentaires pour améliorer la performance du modèle dans ces cas. Nous avons également envisagé l'utilisation de méthodes d'ensemble comme le Gradient Boosting pour renforcer la robustesse et la stabilité des prédictions.

Finalement, ces optimisations ont permis de construire un modèle plutôt équilibré et performant, capable de faire des prédictions fiables avec des erreurs moyennes relativement

faibles. Il reste néanmoins encore des marges d'amélioration, notamment pour les prédictions des valeurs extrêmes.

5. Gradient Boosting

Le **Gradient Boosting** est une technique de machine learning utilisée principalement pour les tâches de régression et de classification. Elle combine plusieurs modèles faibles (généralement des arbres de décision) pour créer un modèle plus robuste et performant.

Modifications :

- Calcul de la médiane pour les régions : 25,52,53,82 qui s'applique aux colonnes : code région, hommes, femmes,15_24, 24_49, 50_plus.
- Remplacer les valeurs pour les régions indiquées ci-dessus dans X_train. Et remplacer les valeurs de X_test pour chaque ville (8)
- Convertir les colonnes de X_train et X_test au format (int).
- Ajouter la colonne densité (homme + femme).
- Normaliser les colonnes

Nous obtenons donc un jeu de données, prêt pour le machine learning, de 4108 entrées et 41 colonnes

a. Premiers résultats

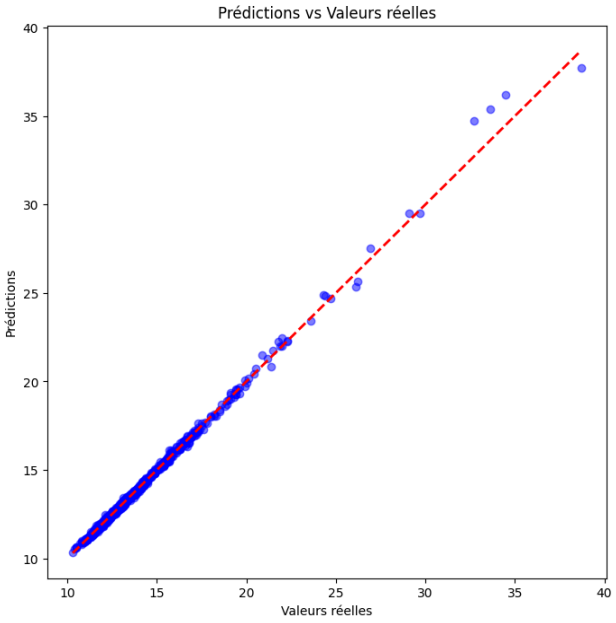
Ces scores indiquent que le modèle explique presque parfaitement la variance des données, avec un très faible écart d'erreur quadratique moyenne (RMSE) sur les ensembles d'entraînement et de test. Cependant, il convient de noter que de tels résultats, avec un R^2 aussi proche de 1 sur les données d'entraînement, sont le signe d'un important surapprentissage du modèle, qui pourrait vraisemblablement mal s'adapter à de nouvelles données.

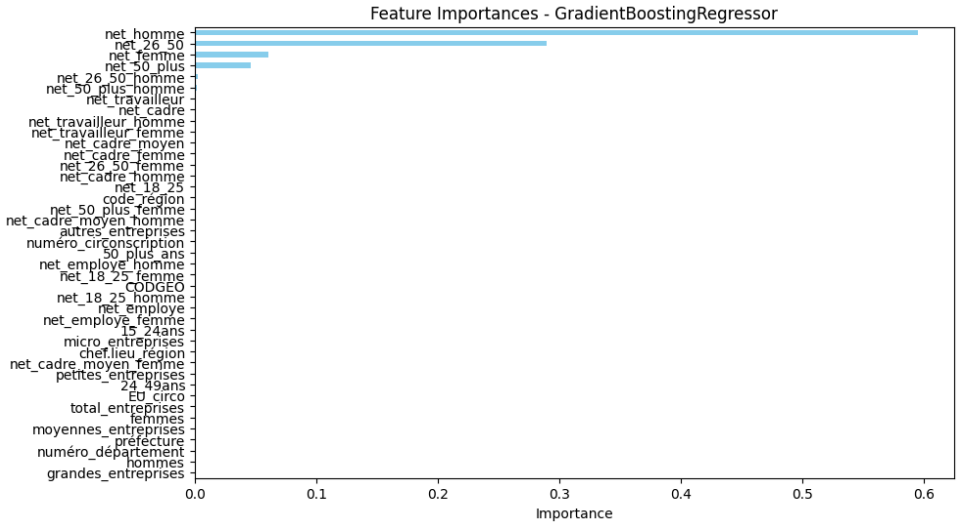
R^2 (Entraînement) : 0.9990629266401417

R^2 (Test) : 0.9969580154178247

RMSE (Entraînement) : 0.07754246848900301

RMSE (Test) : 0.1467789115362258





b. Résultats sans biais

Pour éviter les biais et tenter de limiter le surapprentissage nous supprimons les variables non corrélées et non pertinentes pour le modèle, car elles ont peu d'impact. De plus, nous retirons toutes les variables liées aux salaires, car elles pourraient introduire trop d'informations sur la variable cible, risquant ainsi de fausser le modèle.

R^2 (Entraînement) : 0.5325485976637674
R^2 (Test) : 0.43203890181725035
RMSE (Entraînement) : 1.731893012423306
RMSE (Test) : 2.005600612729252

Sur l'ensemble de test, le modèle explique environ 43,20 % de la variance des données, un score inférieur à celui de l'entraînement, suggérant des difficultés de généralisation sur de nouvelles données.

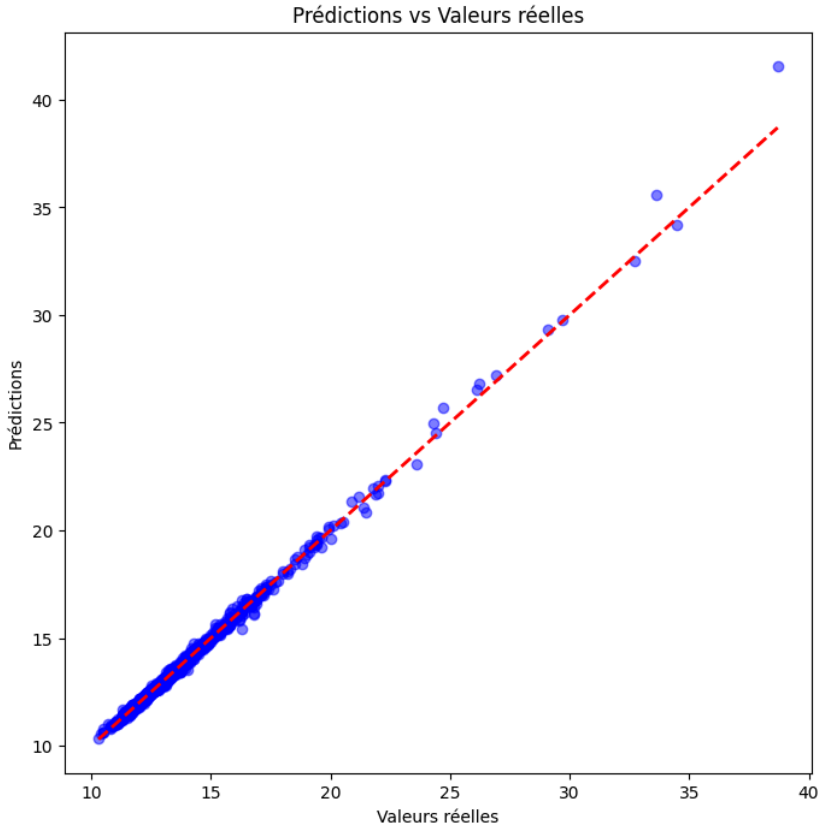
Le RMSE sur l'ensemble de test est d'environ 2,01 unités, ce qui indique des erreurs de prédiction légèrement plus élevées. Cela pourrait être un signe de surajustement ou d'une complexité insuffisante du modèle. Ce modèle, clairement beaucoup moins performant, ne semble pas bien capturer les relations entre les variables dans les données et produit des prédictions peu précises.

c. Optimisations du modèle Gradient Boosting

Pour améliorer la performance du modèle tout en limitant le surapprentissage, nous retirons les variables non corrélées qui ne sont pas pertinentes et ont peu d'impact sur le modèle. Nous allons toutefois réintégrer les variables corrélées, à l'exception de "net_homme", qui est la plus corrélée et pourrait introduire un biais important dans le modèle. Cette approche vise à conserver les variables les plus significatives tout en évitant la fuite d'information susceptible de fausser les résultats.

- **R^2 (Entraînement)** : 0.9978
- **R^2 (Test)** : 0.9957
- **RMSE (Entraînement)** : 0.1185
- **RMSE (Test)** : 0.1746

Ces scores, proches de la première mise en œuvre du modèle présentée plus haut, indiquent que le modèle paraît toujours souffrir de surapprentissage.



La cohérence des points qui collent à la ligne indique des prédictions quasi-parfaites du modèle (ce qui pose question quant à sa fiabilité dans le cas de nouvelles données), ce qui est cohérent avec les métriques R^2 et RMSE.

6. Random Forest Regressor

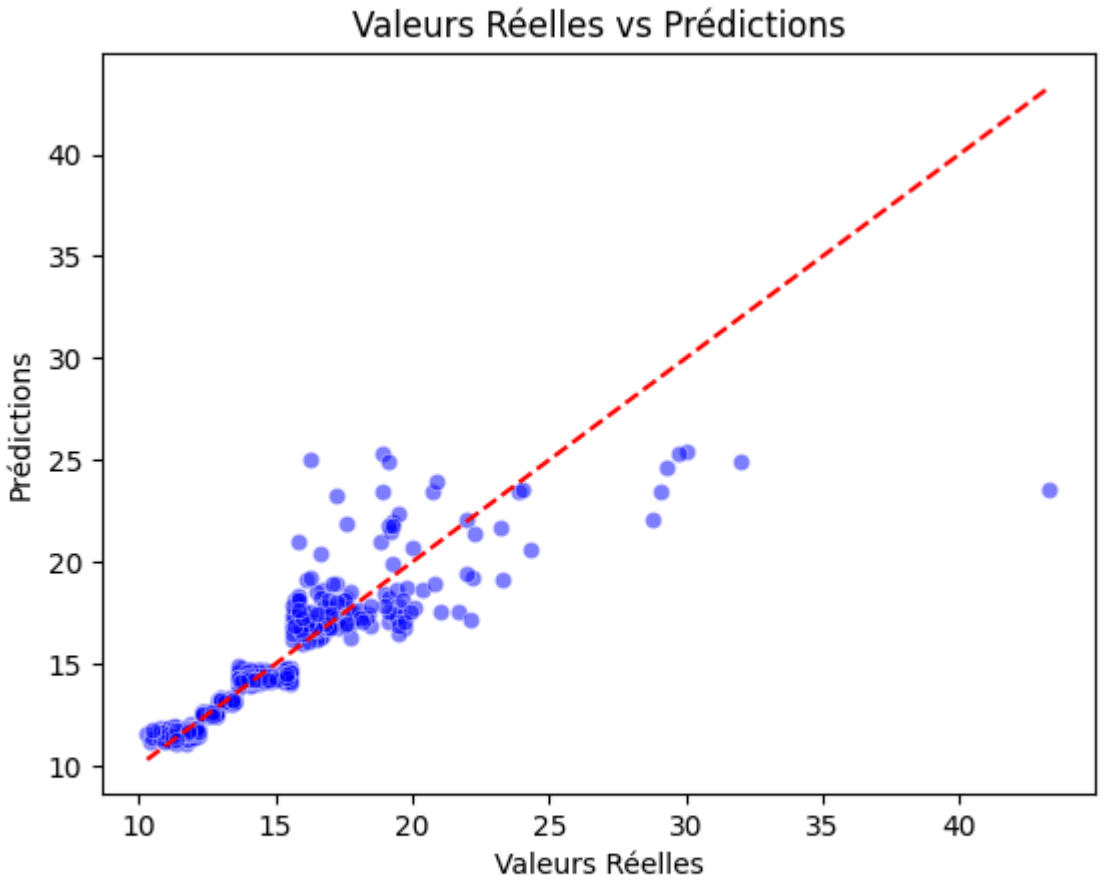
Le modèle Random Forest Regressor est un algorithme d'apprentissage automatique utilisé pour des tâches de régression. Il fonctionne en créant plusieurs arbres de décision sur des sous-ensembles de données et en combinant leurs prédictions pour obtenir une estimation plus précise. Cela permet de réduire les erreurs de prédiction en moyenne et de rendre le modèle plus robuste face aux variations des données.

Modifications

- Suppression des colonnes salaires
- Ajout d'une colonne catégorielle (de 1 à 5) pour la moyenne des salaires par département
- Ajout d'une colonne population totale
- Normalisation des colonnes numériques
- One Hot Encodage de la colonne code_région

a. Premiers résultats

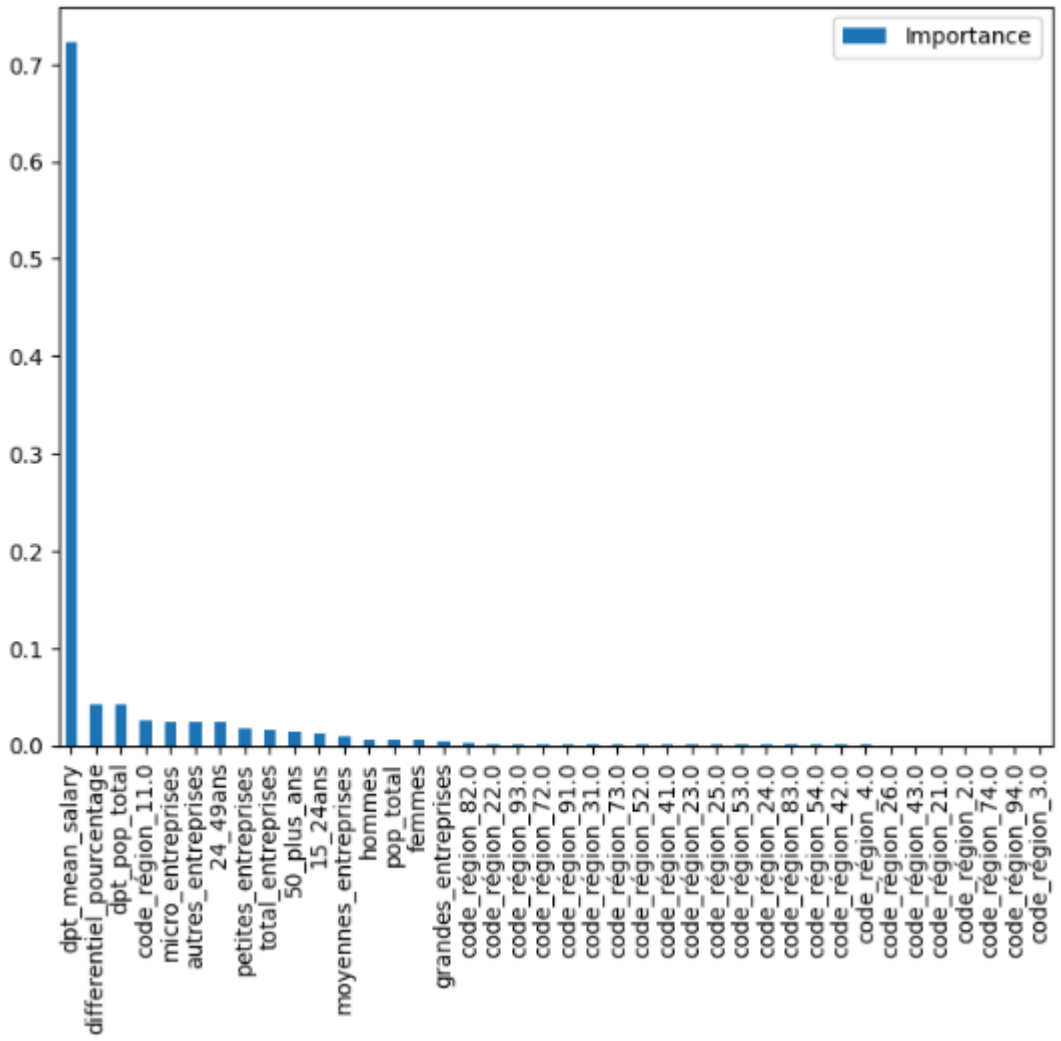
```
score sur X_train_encoded =  0.9730651269077443
score sur X_test_encoded =  0.8073687311188573
```




```
MAE train=  0.19578844270323137
MSE train=  0.17454942360430986
RMSE train= 0.41779112437234694
MAE test=  0.5348346379647748
MSE test=  1.3361592162426612
RMSE test= 1.1559235339081306
```

Le graphique illustre la performance du modèle Random Forest Regressor pour prédire le salaire net horaire moyen dans différentes villes françaises.

Globalement, les prédictions sont plutôt bonnes, car la majorité des points se situent près de la ligne rouge qui représente une prédiction parfaite. Le modèle obtient un excellent score de 0,973 sur les données d'entraînement, mais ce score diminue à 0,808 sur les données de test, ce qui suggère un surapprentissage (overfitting). En effet, le modèle semble bien s'adapter aux données qu'il connaît, mais montre des faiblesses sur de nouvelles données. Les erreurs moyennes restent faibles, ce qui est rassurant pour la précision des prédictions. Cependant, on observe que les prédictions deviennent moins précises pour les villes où le salaire horaire est plus élevé, avec quelques points assez éloignés de la ligne du milieu. En résumé, ce modèle est performant dans l'ensemble, bien qu'il montre certaines limites, notamment pour les valeurs les plus extrêmes, et pourrait bénéficier d'une optimisation pour mieux généraliser.

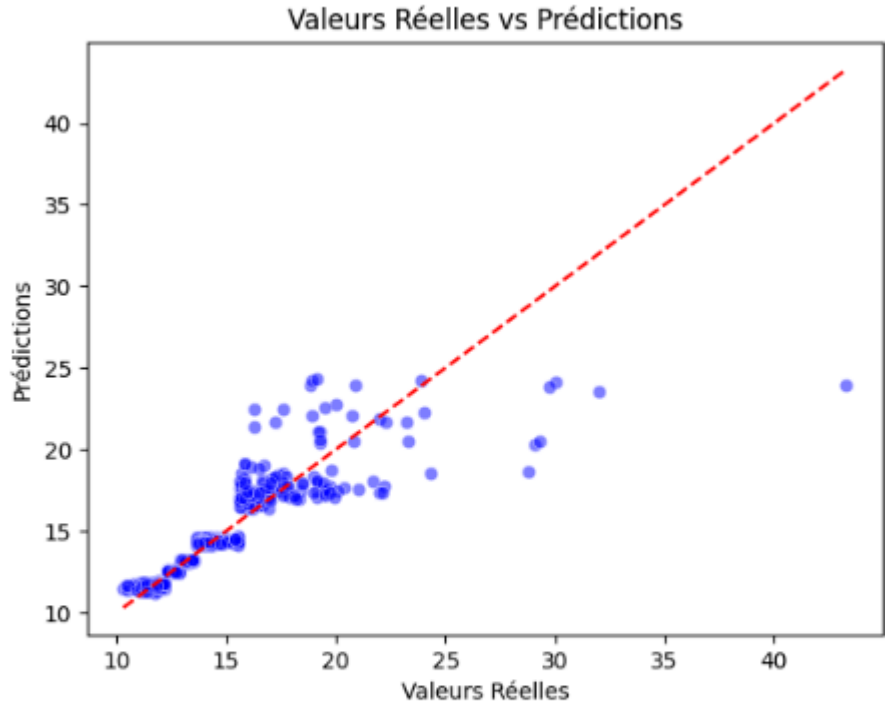


Dans cet histogramme qui présente l'importance des différentes features utilisées par le modèle Random Forest Regressor on observe que la variable "diff_mean_salary" est de loin en tête, avec une importance proche de 0,7, ce qui signifie qu'elle joue un rôle majeur dans les prédictions. Cette variable semble avoir une influence déterminante sur le modèle. Les autres variables, telles que "differential_de_population_2014", "part_agri", et "pop_totale", ont une importance beaucoup plus faible, toutes en dessous de 0,05, ce qui suggère qu'elles contribuent peu aux prédictions comparées à "diff_mean_salary".

Les variables restantes, notamment les codes de régions (à part code_région_11.0, représentant l'Île-de-France), montrent une importance presque négligeable, avec des valeurs proches de zéro. Cela peut indiquer que ces variables n'apportent que peu d'informations nouvelles ou significatives pour la prédiction du salaire horaire moyen.

b. Ajout de paramètres

```
0.8388306768950455
0.7854048760950905
```




```
MAE train=  0.46235749986095204
MSE train=  1.0444457025771203
RMSE train= 1.0219812633199887
MAE test=  0.5455025556859538
MSE test=  1.4885083518979505
RMSE test= 1.2200444057074114
```

Après avoir ajusté les paramètres du modèle Random Forest Regressor, avec une profondeur maximale de 10 et un nombre minimum d'échantillons par feuille de 15, la performance du modèle a changé.

Le score R^2 pour les données d'entraînement a chuté de 0,973 à 0,838, et celui pour les données de test est passé de 0,808 à 0,735. Les erreurs, mesurées par MAE, MSE et RMSE, ont également augmenté, indiquant que les prédictions sont moins précises qu'avant.

Ces modifications ont simplifié le modèle, réduisant le risque de surapprentissage (qui reste néanmoins présent) et améliorant sa généralisation, mais cela a entraîné une légère perte de précision.

7. Ridge

Le modèle Ridge est une méthode de régression qui améliore les prédictions en ajoutant une pénalité aux coefficients des variables. Cette pénalité aide à éviter que certaines variables ne dominent trop les résultats, surtout quand il y en a beaucoup ou qu'elles sont très similaires. En somme, Ridge rend le modèle plus stable et moins sensible aux erreurs en équilibrant

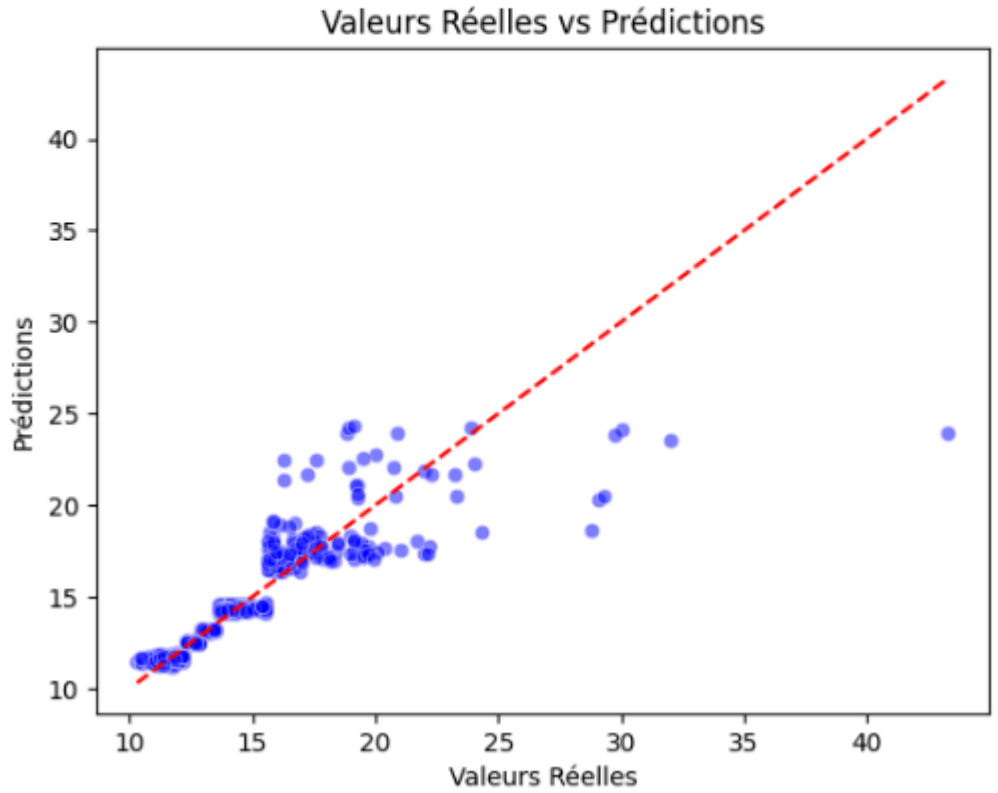
l'importance des différentes variables. Cela s'avère utile pour obtenir des prévisions plus fiables et éviter les problèmes liés à des données trop complexes.

Modifications

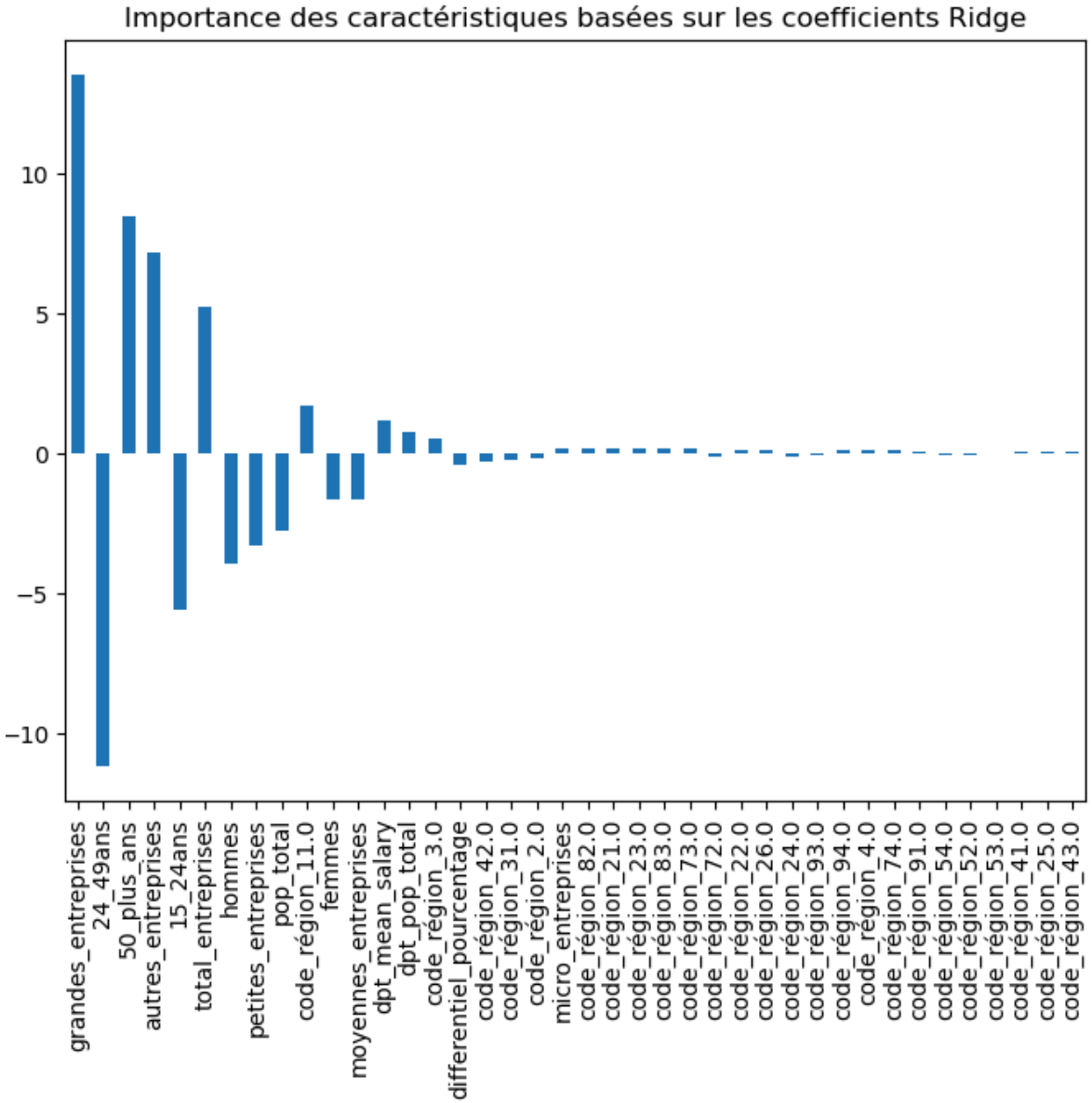
- Suppression des colonnes salaires
- Ajout d'une colonne catégorielle (de 1 à 5) pour la moyenne des salaires par département
- Ajout d'une colonne population totale
- Normalisation des colonnes numériques
- One Hot Encodage de la colonne code_région

Pour ce modèle, nous testons plusieurs paramètres alpha afin d'améliorer au mieux la capacité du modèle Ridge à prédire notre target.

```
best_alpha =  0.060000000000000005
score de X_train_encoded =  0.6598531211747154
score de X_test_encoded =  0.6278988597941655
MAE train=  0.46235749986095204
MSE train=  1.0444457025771203
RMSE train= 1.0219812633199887
MAE test=  0.5455025556859538
MSE test=  1.4885083518979505
RMSE test= 1.2200444057074114
```



On observe que le graphique est assez similaire à celui obtenu grâce au modèle du Random Tree Regressor. Cela est sans doute la conséquence d'un alpha en définitive assez faible (0.06) qui pénalise un peu les coefficients trop importants pour éviter qu'ils ne dominent la prédiction, tout en conservant une bonne flexibilité pour s'adapter aux données. Comme pour le Random Forest Regressor, les prédictions sont globalement proches des valeurs réelles, mais on observe cependant un manque de précision pour prédire les salaires les plus élevés.



Lorsqu'on se penche sur l'importance relative des features de notre modèle, on observe que certaines variables ont un impact significatif sur le salaire. Par exemple, la présence de grandes entreprises influence fortement le salaire, avec un coefficient positif élevé. En revanche, les petites entreprises semblent avoir un effet inverse. D'autres caractéristiques comme le nombre de salariés ou certaines catégories professionnelles (entreprises de 5 à 9 salariés, autres artisans, etc.) ont également un poids notable. En revanche, les codes régionaux et d'autres

variables présentent des coefficients proches de zéro, indiquant une influence marginale sur les prédictions. Ce résultat suggère que certaines caractéristiques géographiques sont moins pertinentes pour expliquer les variations de salaires que les aspects économiques et professionnels.

8. Résumé des métriques et choix du modèle

Modèle retenu : Suite aux analyses et comparaison des métriques ci-dessus, le modèle que nous choisissons comme le plus fiable est le LinearRegression no 2, optimisé avec le GridSearchCV et ses hyper-paramètres. En effet, c'est celui dont la prédiction semble la plus fiable (R^2) avec une marge d'erreur assez faible et un overfitting relativement mesuré. En outre, l'importance des variables est plus distribuée que pour les autres modèles ce qui lui permet de ne pas dépendre que d'une valeur pour faire la prédiction, ce qui est plus sûr.

Modèles de Machine Learning	R² (X_train)	R² (X_test)	MSE (y_train)	MSE (y_test)	RMSE (y_train)	RMSE (y_test)	MAE (y_train)	MAE (y_test)
Linear Regression n°1	0,9997	0,9995	0,0023	0,0026	0,0477	0,0512	0,0378	0,0394
LinearRegression n°2	0,9513	0,9389	0,3385	0,3027	0,5818	0,5502	0,4255	0,4119
Lasso	0,0658	0,0260	6,4897	4,8278	2,5475	2,1972	1,7064	1,5718
Lasso	0,9492	0,9372	0,4315	0,4156	0,3526	0,3114	0,5938	0,5580
Decision Tree regressor	0,7171	0,7296	1,8152	1,9147	1,3473	1,3837	1,0428	1,0345
Decision Tree regressor	0,7092	0,7224	1,8661	1,9662	1,3660	1,4022	1,0531	1,0480
Decision Tree regressor	0,7379	0,7655	1,6819	1,6610	1,2969	1,2888	1,0293	1,0131
Gradient Boosting	0,9991	0,9970			0,0775	0,1468		
RandomForestRegressor n°1	0,9731	0,8074	0,1745	1,3362	0,4178	1,1559	0,1958	0,5348
RandomForestRegressor n°2	0,8388	0,7854	1,0444	1,4885	1,0220	1,2200	0,4624	0,5455
Ridge	0,6597	0,6278	1,0444	2,5818	1,0220	1,6068	0,4624	0,8243

Conclusion

Notre rapport se fixait pour objectif de répondre à différentes questions comme “quel est l’état des lieux de l’inégalité salariale en France ?” ou encore “quelles variables expliquent cette situation ?”. Notre analyse a partiellement permis de répondre à ces interrogations. En effet, au vu des données analysées, l’inégalité semble d’une part territoriale entre la région parisienne (haut salaire) et le reste du pays. Cette répartition territoriale semble, comme nous l’avons vu, suivre dans une certaine mesure la densité de population : plus une zone est densément peuplée, plus le salaire tend à y être élevé. En outre, nous avons pu visualiser et confirmer par modélisation, des inégalités salariales importantes dans l’ensemble du pays par exemple entre :

- hommes et femmes
- cadres d’une part et autres catégories socio-professionnelles d’autre part
- jeunes salariés et salariés plus âgés

Le nombre d’entreprises implantées dans la ville semble également corrélé au salaire moyen qui y est pratiqué, ce qui pourrait - mais il ne s’agit que d’une hypothèse - s’expliquer par une mise en concurrence plus forte dans les zones à forte implantation entrepreneuriale.

Néanmoins, si nous sommes parvenus à avoir une bonne image globale de la situation, nous avons été confrontés à une limite inhérente à la structuration des jeux de données à disposition, qui ne nous permet pas d’accéder à un grand niveau de détails quant à l’analyse de ces variables. Ainsi d’une part, certaines variables importantes pour expliquer les inégalités salariales (niveau d’éducation, secteur d’emploi, etc) n’étaient pas disponibles dans le cadre de notre projet.

D’autre part notre variable cible était fortement dispersée dans de nombreuses variables (e.g. salaire net moyen homme cadre, salaire net moyen 26-50 ans), ce qui a rendu difficile la modélisation par Machine Learning. En effet, le fait de retirer toute variable contenant une partie de l’information sur notre variable cible rendait le modèle inopérant, car ne disposant plus d’assez d’informations. Dès lors, nos modèles ont souvent fait face à du surapprentissage, ce qui laisse à penser qu’ils auraient du mal à s’adapter à de nouvelles données. Le modèle retenu, le Linear Regression optimisé, comprend néanmoins un bon équilibre entre de bonnes performances de prédiction, une certaine fiabilité (liée au nombre élevé de *features* sur lesquelles se base le modèle pour fonctionner) et un surapprentissage existant mais relativement maîtrisé.

La prédiction salariale peut s’avérer très utile pour une multitude d’utilisations. Ainsi, par exemple, elle pourrait être utilisée par les autorités pour mieux identifier les mécanismes à l'œuvre sur l’inégalité salariale et ensuite orienter les politiques publiques dans la direction souhaitée. Une autre utilisation pourrait être, par exemple, celle d’entreprises souhaitant s’implanter et avoir un aperçu du marché salarial afin d’adapter leur positionnement.

Il serait dès lors intéressant de réaliser une nouvelle analyse avec des données supplémentaires et une structuration plus adaptée au Machine Learning (dans laquelle par exemple les variables seraient davantage compartimentées). Celle-ci pourrait alors éventuellement permettre une meilleure compréhension de la thématique des inégalités salariales et des mécanismes qui y sont à l'œuvre.