MINISTÈRE
DE L'ECONOMIE,
DES FINANCES
ET DE LA SOUVERAINETÉ
INDUSTRIELLE ET NUMÉRIQUE

Liberté
Égalité
Fraternité

CentraleSupélec

Prévision de la masse salariale du Conseil d'État

CentraleSupélec

Rapport final de projet

Mohamed AZAHRIOU
Hugo BEAUJARD
Hassan FAYAD
Youcef NAIT AMER

Pôle projet :
Modélisation mathématique

Encadrant : Gurvan HERMANGE

Client : Christophe GAIE – CISIRH

CentraleSupélec

Prévision de la masse salariale du Conseil d'État

Rapport final de projet

Mohamed AZAHRIOU
Hugo BEAUJARD
Hassan FAYAD
Youcef NAIT AMER

Pôle projet : Modélisation mathématique

Encadrant : Gurvan HERMANGE
Client : Christophe GAIE – CISIRH

Mai 2023

Contents

1 Abstract **2**

2 Mise en place du problème **2**
2.1 Mise en contexte . . . 2
2.2 objectif . . . 2
2.3 Fonctionnement du conseil d'État . . . 2
2.3.1 Découpage du conseil d'État . . . 2
2.4 Point d'indice et calcul de la rémunération . . . 3
2.5 La rémunération "exceptionnelle" . . . 4
2.6 Première approche . . . 4

3 Détermination du point d'indice **5**
3.1 Premier choix de modèle : un processus temporel . . . 5
3.2 Deuxième choix de modèle : modélisation déterministe . . . 6

4 Détermination de la rémunération des employés **7**
4.1 Introduction et formalisme . . . 7
4.1.1 Paramètres d'intérêt . . . 7
4.1.2 Répartition initiale . . . 8
4.2 Évolution du système . . . 9
4.3 Calcul du salaire . . . 10
4.3.1 Rémunération brute . . . 10
4.3.2 primes . . . 10

5 Modélisation algorithmique du conseil d'état **12**
5.1 Génération d'une population de fonctionnaires . . . 12
5.1.1 Méthodologie . . . 12
5.1.2 Résultats et discussion . . . 13
5.2 Associé chaque fonctionnaire à son métier . . . 13
5.2.1 Méthodologie . . . 13
5.2.2 Résultats et discussion . . . 14
5.3 Algorithme d'évolution de la masse salariale . . . 14
5.4 Résulat de la modélisation . . . 15

6 Prédiction de la masse salariale grâce au Machine Learning **17**
6.1 Time-Series Cross Validation . . . 17
6.1.1 Première approche . . . 17
6.2 Random Forest . . . 18
6.2.1 Random Forest - Régression Linéaire . . . 18
6.2.2 arbres décisionnels . . . 19
6.2.3 Construction d'arbres décisionnels . . . 19
6.2.4 Le principe de l'algorithme . . . 20
6.2.5 Algorithme Random Forest . . . 20
6.2.6 Algorithme de notre code . . . 21
6.3 XGBoost . . . 22
6.3.1 Construction d'arbre . . . 22
6.3.2 Gradient Boost . . . 22
6.3.3 Notre implémentation à la régression linéaire . . . 23
6.3.4 Analyse d'une Matrice de Confusion . . . 24

7 Interface graphique **25**

8 Conclusion **27**

9 Bibliographie **28**

1 Abstract

Ce mémoire a pour sujet la prévision de la masse salariale du Conseil d'État. Notre objectif est, ici, de présenter des outils mathématiques permettant de modéliser les flux d'argent au sein du conseil d'État afin de pouvoir les implémenter dans un algorithme. Nous poserons d'abord le contexte dans lequel s'inscrit le problème puis nous montrerons comment, notemment grâce à une modélisation ensembliste, en récupérant des données sur le Conseil d'État et par l'utilisation de methodes probabilistes telles que les chaines de Markov, nous avons pu réaliser une telle prédiction.

2 Mise en place du problème

2.1 Mise en contexte

Sous chaque quinquennat, les réformes visant à modifier la masse salariale des fonctionnaires sont nombreuses et perturbantes pour les prévisions budgétaires des différents corps de la fonction publique. Afin de faire face à ces évolutions et celles à venir telles que la modification de l'âge de la retraite ou encore la modification des grilles salariales notamment par l'ajout d'échelons supplémentaires, il est indispensable pour le CISIRH (Centre Interministériel de Service Informatiques relatifs aux Ressources Humaines) d'avoir un logiciel de prévision de cette masse salariale. Par l'étendue du nombre de paramètres à considérer et par le changement perpétuel de la composition des membres du Conseil d'Etat, l'Etat et le service de Monsieur Christophe GAIE recherche un logiciel paramétrable permettant de simplifier la prévision de la masse salariale du programme 165 propre aux agents du Conseil d'Etat. Afin de répondre à cette attente, nous proposons de concevoir un logiciel paramétrable permettant de prévoir et de modéliser avec la meilleur précision la masse salariale nécessaire au bon fonctionnement du Conseil d'Etat.

2.2 objectif

Pour cela nous envisageons la réalisation d'un programme informatique codé sous Python et d'une API permettant de renseigner la masse salariale d'une population d'agents du Conseil d'Etat.

2.3 Fonctionnement du conseil d'État

Le conseil d'État est ici notre objet d'étude. En particulier, pour connaître sa masse salariale il est important de s'interesser à la rémunération de ses fonctionnaires.

2.3.1 Découpage du conseil d'État

Le fonctionnement du conseil d'État se découpe en catégories (A,B et C), elles mêmes découpées en grades qui eux même se découpent en échelons. Dans le cadre de ce rapport, notre discussion sera centrée sur le salaire brut, qui représente le montant total avant déductions.

Les rémunérations au sein de la fonction publique sont soumises à une réglementation stricte. Elles sont définies par la loi et les règles qui les gouvernent sont beaucoup moins flexibles que dans le secteur privé. La rémunération d'un agent de la fonction publique est caractérisée par plusieurs paramètres. Tout d'abord, son salaire peut être divisé en deux parties distinctes : une partie fixe, appelée rémunération indiciaire, qui est déterminée par des grilles salariales spécifiques, et une partie variable, appelée rémunération "exceptionnelle", qui regroupe diverses primes et indemnités exceptionnelles telles que les bonifications indiciaires, le régime indemnitaire tenant compte des fonctions, des sujétions, de l'expertise et de l'engagement professionnel (RIFSEEP), l'indemnité de résidence, ainsi que les indemnités liées aux enfants à charge, entre autres.

Ces différentes composantes de la rémunération contribuent à déterminer le salaire global d'un agent de la fonction publique. La rémunération indiciaire, basée sur les grilles de salaires, représente la part fixe et prévisible du salaire, tandis que la rémunération "exceptionnelle" comprend des éléments variables et dépendants de certaines conditions ou performances.

Il est important de comprendre que ces paramètres de rémunération sont régis par des textes législatifs et réglementaires spécifiques, qui établissent les règles et les critères permettant de déterminer le niveau de rémunération d'un agent de la fonction publique. Cette réglementation vise à garantir l'équité et la transparence dans la rémunération des agents, ainsi qu'à assurer la cohérence et la gestion efficace des ressources publiques. Dans la suite de ce rapport,

nous examinerons de plus près chacun de ces paramètres de rémunération, en mettant l'accent sur leur impact sur la masse salariale globale et sur les enjeux associés à leur gestion dans le contexte de la fonction publique.

A **Attaché d'administration de l'Etat -corps interministériel - CIGEM** Vérifié le 06/01/2023 — Attaché d'administration hors classe

Echelon	Indice Brut	Indice majoré	Durée	Salaire brut
1	797	655	2 ans	3 176,77 €
2	850	695	2 ans	3 370,77 €
3	896	730	2 ans	3 540,52 €
4	946	768	2 ans 6 mois	3 724,82 €
5	995	806	3 ans	3 909,12 €
6	1027	830	-	4 025,52 €
Echelon spécial	HEA	-	1 an	4 316,53 €
	HEA2	-	1 an	4 486,28 €
	HEA3	-	-	4 714,23 €

Figure 1: Exemple d'une grille de rémunération d'un grade de catégorie A (en Janvier 2023)

2.4 Point d'indice et calcul de la rémunération

La grille de rémunération en figure 1 met en évidence un paramètre important dans le calcul du salaire d'un fonctionnaire au conseil d'État : l'indice majoré. La rémunération R d'un fonctionnaire du conseil d'état se calcule par la formule suivante :

$$R = IM \times p \tag{1}$$

Où p désigne le point d'indice. Le point d'indice (mensuel ou annuel) est un indice universel en France évoluant au cours du temps permettant de calculer le traitement brut, mensuel ou annuel, des fonctionnaires, magistrats et de certains agents contractuels.

L'enjeu est alors de connaître le point d'indice de la periode étudiée ainsi que la répartition au cours du temps dans chaque catégorie, grade et échelon des membres du conseil d'État afin de calculer la rémunération de chaque membre et donc la masse salariale.

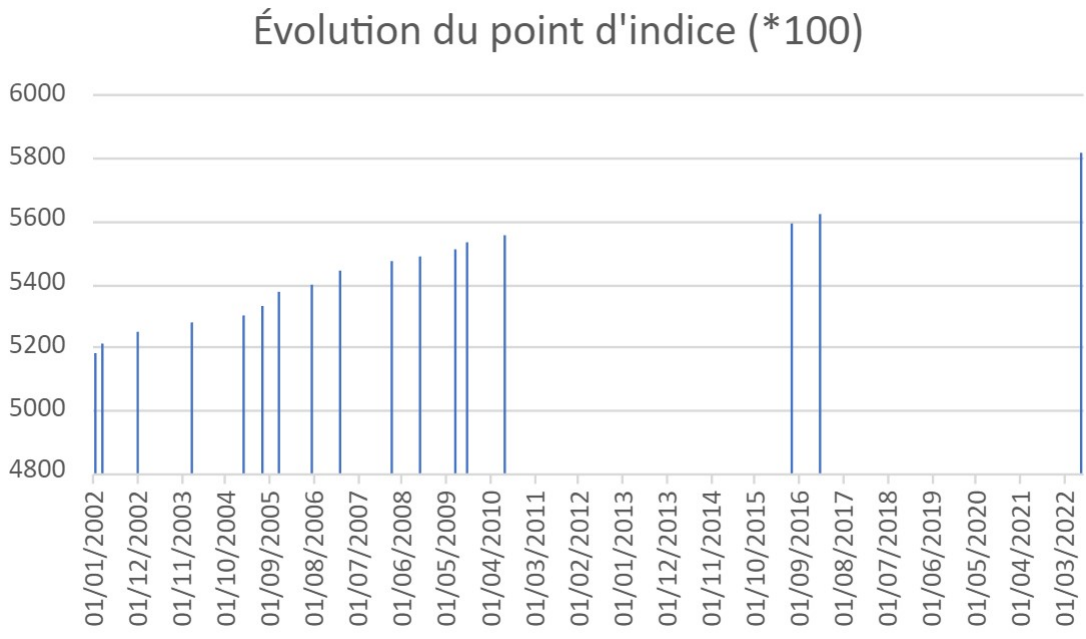

Figure 2: Évolution du point d'indice annuel($\times$100)

2.5 La rémunération "exceptionnelle"

La rémunération "exceptionnelle" s'assimile aux primes, qui occupent une part importante d'un agent public (allant jusqu'à 50

$$\text{Primes} = \sigma_{\text{prime}} S = S(\sigma_{\text{stat}} + \sigma_{\text{excep}})$$

Au 29/05/2022, la valeur du point d'indice est $p \approx 4.686025$ €/pt/mois.

2.6 Première approche

Où σ_{stat} correspond au taux de prime "statutaires" acquis grâce au niveau de responsabilité et à la difficulté des missions de l'agent, et σ_{excep} correspond à une part variable, correspondant à l'assiduité, le sens du service public...

Dans un premier temps, nous partons d'une modélisation simple. On considère que le salaire d'un agent est une variable aléatoire S qui peut se décomposer de la manière suivante :

$$S = P + \text{IM} \times p$$

$$P = \sigma S$$

Où IM, P et S sont des variables aléatoires réelles liées entre elles et $\sigma \in]0, 1[$ est également une variable aléatoire. P désigne le montant brut de prime perçu par un agent pour un mois donné. En modélisant le salaire perçu par un agent, il est alors possible de généraliser à un groupe de plusieurs agents, puis à la fonction publique entière. L'objectif est de saisir ici le comportement "macroscopique" de la masse salariale globale de l'État. Le système ci-dessus traduit simplement une décomposition du salaire en une part fixe et une part variable. En simplifiant ce système, on obtient alors :

$$S = \frac{\text{IM} \times p}{1 - \sigma}$$

Ainsi, estimer un salaire revient à estimer l'indice majoré IM et un certain taux de prime σ.

Pour simuler l'évolution du salaire au fil des années, on modélise de la même manière un processus stochastique $(S_t)_{1 \leq t \leq N}$ tel que :

$$S_t = \frac{\text{IM}_t \times p}{1 - \sigma_t}$$

Il suffit de se donner des lois de probabilité pour IM_t et pour σ_t pour en déduire une certaine loi de S_t.

Hypothèses retenues

Loi de IM_t

On modélise l'évolution de IM par revalorisation annuelle. Cela est justifié par l'adoption d'un système de grille et d'échelon indexé par l'ancienneté de l'agent : à chaque transition d'un échelon à un autre (toujours d'un échelon inférieur vers un échelon supérieur), IM est revalorisé d'un certain taux r. On a ainsi :

$$\forall t \geq 1, \quad \text{IM}_t = (1 + r_t)\text{IM}_{t-1}$$

Où r_t est (encore) un taux compris entre 0 et 1 et dont sa valeur en pourcentage $r_t\%$ suit une loi géométrique de raison $\alpha_r(t)$:

$$\forall t \geq 1, \quad r_t/100 \sim \text{G}(\alpha_r(a(t)))$$

3 Détermination du point d'indice

3.1 Premier choix de modèle : un processus temporel

Après une première discussion avec notre client, il semblait que l'évolution du point d'indice, bien que strictement croissante, était due au hasard et aux décisions du gouvernement à un instant donné. Nous avons choisi de modéliser le point d'indice comme un processus aléatoire autorégressif $p_{n+1} = p_n + \epsilon_n$, où ϵ_n sont des variables aléatoires indépendantes et identiquement distribuées (iid) suivant une loi normale $\mathcal{N}(\mu, \sigma^2)$. Ainsi, pour chaque année d'observation n, nous avons $\epsilon_n = p_{n+1} - p_n$. Nous pouvons estimer les valeurs de μ et σ^2 en utilisant la loi des grands nombres pour obtenir les moyennes et variances empiriques des ϵ_i. Pour K années d'observations du point d'indice, nous avons :

$$\hat{\mu} = \frac{1}{K-1} \sum_{i=0}^{K-1} (p_{i+1} - p_i)$$

$$\hat{\sigma}^2 = \frac{1}{K-1} \sum_{i=0}^{K-1} (p_{i+1} - p_i - \hat{\mu})^2$$

Pratiquement, ce qu'on peut faire aussi, et ce qu'on à implementer sur Python, c'est d'essayer des trouver les paramètres qui minimise le MSE (minimum squarred error). Ainsi on tourne le système sur beaucoup de paramètres et on voit pour quel jeu le MSE est minimal. On obtient ainsi les paramètres: $\mu = 34$ et $\sigma = 23$.

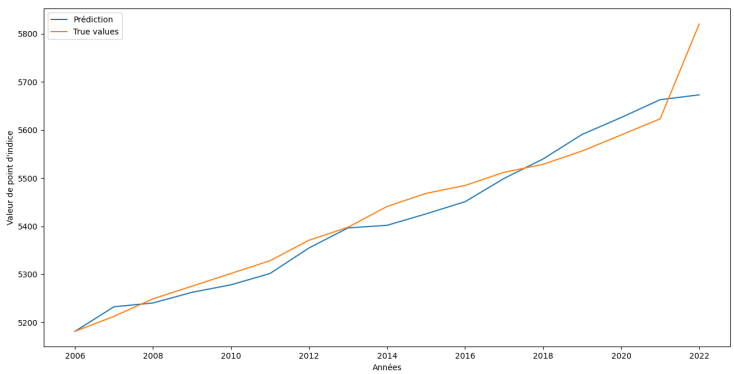

Figure 3: Évolution du point d'indice annuel ($\times 100$) et prédiction

Cependant, ce choix de modélisation présente un problème évident : la possibilité pour ϵ_i d'avoir des valeurs négatives. Bien que le fitting permette d'avoir un écart type rendant les valeurs négatives négligeables, il reste une probabilité non-nulle que cela se produise, ce qui contredit la croissance du point d'indice. Une piste d'amélioration serait de choisir une autre loi pour les ϵ_i. De plus, on observe des variations de pentes d'une année à l'autre bien trop chaotiques: la variation brusque entre 2021 et 2022 n'est pas modélisable par ce modèle.

3.2 Deuxième choix de modèle : modélisation déterministe

Notre deuxième approche consiste à modéliser le point d'indice de manière déterministe. Nous approchons le point d'indice en utilisant une régression polynomiale avec la fonction `polyfit` du module `numpy`. Cette fonction prend en entrée la liste des temps observés, les valeurs mesurées du point d'indice et le degré du polynôme par lequel on souhaite approximer la fonction. Elle cherche les coefficients du polynôme qui minimisent l'écart quadratique moyen. Nous posons $(t_i)1 \leq i \leq N$ comme la liste des N temps observés, $(p_i)1 \leq i \leq N$ comme les valeurs mesurées du point d'indice aux temps t_i, d comme le degré du polynôme et $a_0, a_1, \ldots, a_d$ comme les coefficients du polynôme à déterminer. Nous cherchons donc à minimiser

$$RMS(a_0, a_1, \ldots, a_d) = \sum_{i=1}^{N} (p_i - \sum_{k=0}^{d} a_k t_i^k)^2 \tag{2}$$

En posant :

$$T = \begin{bmatrix} 1 & t_1 & \ldots & t_1^d \\ 1 & t_2 & \ldots & t_2^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_N & \ldots & t_N^d \end{bmatrix}, \quad A = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{bmatrix}, \quad P = \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_N \end{bmatrix},$$

nous pouvons réécrire

$$RMS(A) = \|P - TA\|^2 \tag{3}$$

et ainsi ramener le problème au critère des moindres carrés:

$$\nabla \text{RMS}(A) = -2P^T(P - TA) + 2T^T(TA) \tag{4}$$

et nous cherchons un minimum, donc un extrémum de RMS, ce qui correspond à A tel que

$$\nabla \text{RMS}(A) = 0 \tag{5}$$

Nous avons alors tenté une régression pour d=1 et d=2. On remarque que la régression de degré 2 n'est pas n'écessairement plus efficace que celle de degré 1. On obtient pour la première le coefficient directeurs de ≈ 32.3 et une valeure à l'origine de ≈ -59632.3. Pour la deuxième, on obtient un coefficient devant le terme de degré 2 qui vaut ≈ 0.6, celui devant le terme de degré 1 qui vaut ≈ -2560.4 et celui à l'origine vaut ≈ 2551224.9.

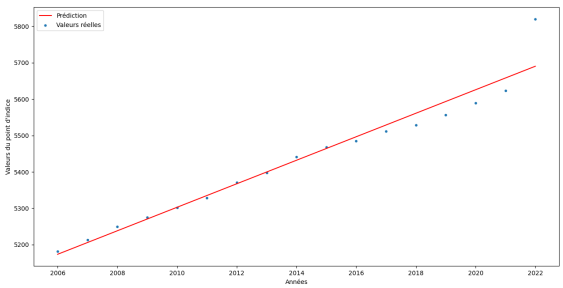

Figure 4: Régression linéaire de degré 1

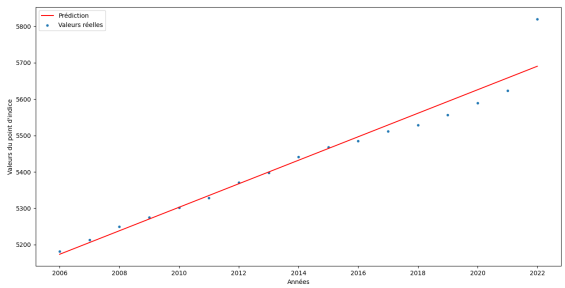

Figure 5: Régression linéaire de degré 2

Par souci de temps nous avons choisi alors de garder le deuxième choix de modélisation, parce qu'il est le plus simple et il nécessite le moins de temps de calcul.

4 Détermination de la rémunération des employés

Déterminer la répartition de chaque grade afin de connaître la rémunération est la partie la plus dure du processus. Nous avons fait le choix de représenter le conseil d'État par un ensemble fini M. Nous avons d'abord supposé que le nombre de fonctionnaires au conseil d'État était une entrée du programme. Par manque de données sur le conseil d'État et ses membres (celles-ci étant confidentielles), nous avons choisi de les inférer statistiquement via des données relatives à la population française en général (par exemple : le nombre d'enfants). On fait le choix de représenter de manière ensembliste le conseil d'État. On s'intéresse par la suite individuellement à chaque membre du conseil d'État.

4.1 Introduction et formalisme

Soient :

- M l'ensemble des membres du Conseil d'État, où $|M| = 200$.
- C l'ensemble des catégories possibles pour chaque membre du Conseil, où $|C| = 3$, représenté par $\{A, B, C\}$.
- G l'ensemble des grades possibles pour chaque catégorie pour chaque membre du Conseil, où $|G| = 3$, représenté par $\{g1, g2, g3\}$.
- E l'ensemble des échelons possibles pour chaque grade pour chaque membre du Conseil, où $|E| = 10$, représenté par $\{e1, e2, ..., e10\}$.

Pour commencer, nous devons initialiser les valeurs des variables du vecteur d'état X_0. Ces valeurs peuvent être basées sur les données de départ ou choisies de manière aléatoire, selon les informations disponibles.

4.1.1 Paramètres d'intérêt

On se place ici dans le cas où les données de départ sont inconnues, mis à part le nombre de fonctionnaires N_{fonc} qu'on suppose être une entrée du système. On suppose que chaque paramètre au temps 0 est une variable aléatoire de loi discrète à valeur dans un ensemble fini du même type afin de simplifier le programme.

Pour chaque paramètre $P(m)$, quel que soit $m \in M$, on suppose parfaitement connu l'ensemble d'arrivée de P noté E_p qu'on choisit tel que $Card(E_p) < \infty$. À cette variable aléatoire, on définit également pour tout $e \in E_p$ un poids w_e. On note par ailleurs W_p l'ensemble des poids pour p tel que

$$\forall p \in E_p, P(P(m) = p) = \frac{w_p p}{\sum_{e \in E_p} w_e} \tag{6}$$

Les poids sont choisis en fonction de données accessibles sur le conseil d'État ou la fonction publique. On s'intéresse aux paramètres suivants pour chaque fonctionnaire :

- la classe d'âge Ca(m) dont l'ensemble d'arrivée est

  $$E_{Ca} = \{I_1 = [\![15, 30]\!], I_2 = [\![30, 50]\!], I_3 = [\![50, 60]\!]\}$$

  et l'ensemble des poids $W_{Ca} = [14.3, 50.8, 34.9]$. L'âge de départ A(m,0) suit une loi uniforme discrète sur Ca(m)
- le type de fonction publique $F_p(m)$

  $$E_{Fp} = [Etat, territoriale, hospitaliere] \quad et \quad W_{Fp} = [44, 35, 21]$$

- le genre (que l'on simplifie ici par Homme ou Femme uniquement)

  $$E_g = [homme, femme] \quad avec \quad W_g = \{57, 43\}$$

- le nombre d'enfants $N_e(m)$, On suppose qu'un individu peut avoir de 0 à 4 enfants. On suppose par ailleurs que N_e est
- la zone géographique z E_z={1,2,3}
- le nombre de journée d'arrêt (arrêt maladie) $J_a(m)$ qui dépend de la classe d'âge Ca(m). on a :

$$E_{Ja} = [0, 4, 11.5, 22.5, 30]$$

quelque soit la valeur de Ca. Les poids diffèrent cependant selon la classe d'âge :

$$W_{Ja|I_1} = \{70, 19.2, 4.5, 1.2, 5.4\} \quad W_{Ja|I_2} = [62, 20.9, 6.08, 3.04, 6.3] \quad W_{Ja|I_3} = \{64, 15.48, 7.92, 2.16, 10.44\}$$

- La NBI (Nouvelle Bonification Indicière) correspond à une attribution de points d'indice majoré supplémentaires dépendant du métier exercé et d'autres facteurs. Ici, on la suppose aléatoire mais dépendant de la catégorie $C(t, m)$. On a :

$$E_{NBI|A} = [\![15, 121]\!] \quad E_{NBI|B} = [\![10, 31]\!] \quad E_{NBI|C} = [\![10, 21]\!]$$

et

$$W_{NBI|A} = \{e^{-\frac{i-15}{50}} - e^{-\frac{i+1-15}{50}}, i \in E_{NBI|A}\}$$
$$W_{NBI|B} = \{e^{-\frac{i-10}{15}} - e^{-\frac{i+1-150}{15}}, i \in E_{NBI|B}\}$$
$$W_{NBI|C} = \{e^{-\frac{i-10}{13}} - e^{-\frac{i+1-10}{13}}, i \in E_{NBI|C}\}.$$

4.1.2 Répartition initiale

Pour attribuer un grade et un échelon initial à chaque membre du conseil d'État $m \in M$, on commence par attribuer un indice majoré provisoire $I\tilde{M}(m)$. L'indice majoré est intrinsèquement lié à la responsabilité et l'importance du poste. Naturellement, on peut supposer que plus un individu est âgé plus il a de chance d'occuper de tels postes.

Pour simplifier la modélisation, on suppose que $I\tilde{M}(m)$ est un processus aléatoire de la forme :

$$I\tilde{M}(m) = \lfloor a \times A(m, 0) + b + \epsilon_m \rfloor$$

où ϵ_m est un bruit blanc $\epsilon_m \sim \mathcal{N}(0, \sigma)$. avec a,b,$\sigma$ des constantes que l'on fixe. On a choisi a=300/52, b=300-16a et $\sigma = 80$

Par la suite, on attribue un métier aléatoire selon ceux disponible pour la catégorie attribuée à l'individu. Pour chaque métier on dispose de la grille de rémunération (échellons et grades).

Le choix de G(m,0)-1 suit une loi binomiale de paramètres (Nombre de grade(C),p) où

$$p = \frac{Nombre\_de\_grade - 1 - \lfloor (A(m) - 21)/16) \rfloor}{Nombre\_de\_grade + 1}$$

avec A(m) l'âge du membre m

L'échelon choisi est alors celui dont l'indice majoré associé est le plus proche de $I\tilde{M}$

4.2 Évolution du système

L'évolution dans les échelons, grades et catégories des membres du conseil d'État est conditionnée par plusieurs facteurs. On sait qu'un individu à un instant donné dans un échelon donné, un grade donné, et une catégorie donnée à plusieurs choix d'évolutions pour l'année suivante. Il peut rester à la même situation ou monter d'un échelon (s'il a assez d'ancienneté dans le poste actuel). Il y a aussi la possibilité pour chaque fonctionnaire de monter d'un grade ou d'une catégorie par le biais d'un concours. Lorsque celui-ci réussi le concours pour monter de grade, il passe au grade supérieur et au premier échelon de celui-ci. De même lorsque le concours pour monter d'une catégorie est réussi, il monte à la catégorie superieure (de C à B ou de B à A) et retourne au premier échellon du premier grade de cette catégorie.

Après discution avec le client et pour facilité la modélisation, nous avons fait les choix d'hypothèses suivants :

- Le passage d'un échelon à un autre est déterministe, celui ci dépendra du temps passé par le fonctionnaire en cellui-ci à partir de données relatives à la durée moyenne passée par un membre du Conseil d'État dans cet échelon.
- Dans ce document, nous allons modéliser l'évolution des salaires des fonctionnaires en utilisant une chaîne de Markov. Le modèle est basé sur le code fourni, qui simule l'évolution des salaires des fonctionnaires sur une période donnée.

  Le modèle de la chaîne de Markov repose sur les paramètres suivants :

  - $N_{\text{états}}$: le nombre total d'états possibles pour le vecteur d'état X_t.
  - $P(t)$: la matrice de transition à l'instant t, où $P(t)_{l,l'}$ représente la probabilité de transition de l'état l à l'état l'.
  - X_t : le vecteur d'état à l'instant t, qui regroupe les variables suivantes :
    - $G_t \in G$: le grade à l'instant t.
    - $C_t \in C$: la catégorie à l'instant t.
- C_t et G_t sont supposés indépendants de l'échelon E, quelque soit l'échelon auquel se trouve le fonctionnaire il aura la même probabilité de réussir le concours. (cependant G_t et C_t ne sont pas indépendants)

La matrice de transition $P(t)$ représente les probabilités de transition entre les états à l'instant t. Pour construire cette matrice, nous devons définir les probabilités de transition $p_{l,l'}(t)$ pour chaque paire d'états l et l'. Ces probabilités dépendent des règles de passage entre les différents niveaux du salaire, des promotions de grade et de corps, ainsi que de l'évolution naturelle de l'indice majoré.

Les probabilités de transition peuvent être calculées en utilisant des fonctions qui associent les états actuels et les états suivants. Par exemple, la probabilité de transition $p_{l,l'}(t)$ peut être définie comme :

$$p_{l,l'}(t) = P((G_t, C_t) = (C_i, G_i) | (G_{t-1}, C_{t-1}) = (G'_i, C'_i))$$

où ici G_i,$G'_i \in G$, C_i,$C'_i \in C$ représentent des grades et échelons possibles.

Dans le code, ces probabilités de transition peuvent être calculées en fonction des règles de passage et des distributions définies.

Une fois que nous avons la matrice de transition $P(t)$, nous pouvons utiliser la chaîne de Markov pour évoluer l'état du vecteur X_t au fil du temps. À chaque instant t, nous tirons un nouvel état l' à partir de la loi donnée par la probabilité de transition $p_{l,l'}(t)$. Ensuite, nous mettons à jour les valeurs des variables du vecteur X_t en fonction de l'état l'.

Par exemple, si l'état l' correspond à une promotion de grade, nous mettons à jour le grade, l'indice majoré et l'échelon de l'agent en fonction des règles de promotion.

4.3 Calcul du salaire

Une fois que le vecteur d'état X_t a été mis à jour, nous pouvons utiliser les valeurs des variables pour calculer le salaire de l'agent à l'instant t. Cela peut être fait en utilisant les formules mathématiques appropriées, en tenant compte des attributs tels que l'indice majoré, les taux de prime et d'autres facteurs pertinents.

4.3.1 Rémunération brute

À chaque nouvelle itération t du programme, la rémunération brute d'un fonctionnaire du conseil d'État est calculée par la formule

$$R_0(m,t) = IM(m,t) \times p(t)$$

Cependant d'autres éléments induisant des fluctuations sont à considérer afin d'avoir la rémunération R réelle notemment les primes.

Lorsque la rémunération R réelle est déterminée, la masse salariale $M_s(t)$ réelle se calcule par la formule suivante :

$$M_s(t) = \sum_{m \in M} R(m,t)$$

4.3.2 primes

Il existe plusieurs primes et indemnités qui revalorisent le salaire d'un fonctionnaire au conseil d'État. Celles ci sont essentielles à prendre en compte dans notre modèle.

Ces augmentations sont sous la règlementation du Régime Indémnitaire tenant compte des Fonctions, des Sujétions, de l'Éxpertise et de l'Engagement Professionel (RIFSEEP), l'outil indemnitaire de l'État qui remplace la plupart des primes et indemnités existantes dans la fonction publique. Celui-ci se découpe en deux parties : Le CIA (Complément Indemnitaire Annuel) et l'IFSE (Indemnité de Fonctions, des Sujétions, et d'Expertise).

L'IFSE constitue l'indemnité principale du RIFSEEP. Cette revalorisation a pour objectif de valoriser l'exercice des fonctions. Les catégories sont rangées par ordre de résponsabilité demandé aux fonctionnaires. En effet un fonctionnaire en catégorie A aura plus de responsabilité qu'un fonctionnaire en catégorie B qui lui même en aura plus qu'un fonctionnaire en catégorie C. Il en va alors de même pour l'IFSE dont on supposera une valeur fixe par catégorie (en effet, on peut diviser d'autant plus le niveau de responsabilité mais on ne tiendra compte que de la catégorie ici afin de simplifier la modélisation.)

En nous basant sur le rapport du groupe de l'année dernière qui a utilisé les données moyennées publiques du Centre De Gestion de la Fonction Publique Territoriale des Landes (Nouvelle-Aquitaine) et qui les a adapté à notre problème, on a alors obtenu les valeurs suivantes :

$$IFSE(A) = 38328 euros/an$$

$$IFSE(B) = 15036 euros/an$$

$$IFSE(C) = 11736 euros/an$$

Le CIA est lui plus facultatif. Il s'agit d'un pourcentage de l'IFSE versé afin de recompenser un fonctionnaire pour sa contribution au collectif du travail, son investissement personnel, etc... Il y a pour chaque catégorie un pourcentage maximal que peut atteindre le CIA : 15% pour la catégorie A, 12% pour la catégorie B et 10% pour la catégorie C ce qui donne la valeur du CIA maximal pour chaque catégorie :

$$CIA_{max}(A) = 5749.2 euros/an$$

$$CIA_{max}(B) = 1804.32 euros/an$$

$$CIA_{max}(C) = 1173.6 euros/an$$

On fait le choix de définir le CIA de sorte à prendre en compte le nombre d'arrêts maladie et du nombre d'années d'experiences. en temps normal l'IFSE est censé prendre cela en compte mais pour simplifier le programme nous avons choisi de procéder ainsi.

On choisit de modéliser la distribution du CIA par une loi exponentielle de paramètre 1 dont la variable x dépendrait du nombre d'arrêt maladie et du nombre d'années d'expériences. Moins d'arrêt maladie et plus d'expérience impliquent un meilleur bonus. De plus pour une telle loi on a $f(x) \approx 0$ lorsque x>5.

On choisit alors notre paramètre tel que x>5 pour un nombre élevé d'absence (on choisit pour référence le nombre moyen d'arrêt maladie des fonctionnaires) et pour un nombre d'année d'activité élevé (le maximum étant atteint à la retraite). D'où le choix suivant :

$$x = \frac{5}{2}(y+z)$$
$$y = \frac{\text{retraite} - \text{nombre d'années d'activite}}{\text{retraite}}$$
$$z = \frac{\text{nombre d'arrêt maladie}}{\text{nombre moyen d'arrêt maladie}}$$

Ainsi la prime d'un membre du conseil d'État $m \in M$ se calcule par la formule suivante :

$$Prime(m,t) = IFSE(C(m,t)) + CIA_{max}(C(m,t))e^{-\frac{5}{2}(y(m,t)+z(m,t))}$$

Et la rémunération brute totale R d'un membre du conseil d'État est donnée par

$$R(m,t) = R_0(m,t) + Prime(m,t)$$

5 Modélisation algorithmique du conseil d'état

5.1 Génération d'une population de fonctionnaires

Dans le cadre d'une étude scientifique visant à analyser différents scénarios de composition du Conseil d'État, il est nécessaire de générer une population de fonctionnaires représentant cette formation. Dans cet article, nous présentons une approche de génération aléatoire de fonctionnaires basée sur des paramètres clés tels que la catégorie, l'âge, le sexe, le nombre d'enfants, la zone géographique et la Nouvelle Bonification Indiciaire (NBI). Ce besoin d'avoir recours à une génération de fonctionnaires découle de la confidentialité des données sur cette division de la fonction publique. Nous n'avons pas une idée plus précise du nombre de personnes dans chaque échelon et encore moins des données plus précises telles que l'âge des fonctionnaires ou le nombre moyen d'enfants. Cependant, cette algorithme pourrait permettre à une personne en possession de ces informations de calculer de manière plus efficace la masse salariale.

5.1.1 Méthodologie

Pour générer aléatoirement une population de fonctionnaires, nous avons mis en place la fonction `generer_pop(N)`. Cette fonction retourne une liste de N vecteurs, chaque vecteur représentant un fonctionnaire et contenant les informations suivantes :

- **Catégorie** : Chaque fonctionnaire est attribué à l'une des trois catégories possibles du Conseil d'État (A, B ou C). La répartition des fonctionnaires dans chaque catégorie est réalisée en utilisant des poids correspondant aux proportions observées au sein du Conseil d'État.
- **Indice brut majoré** : Cet indice est public, nous avons pu le trouver sur le site de la fonction publique.
- **Maladie** : Le nombre de jours de maladie que chaque fonctionnaire est susceptible de prendre durant une année civile. Des poids sont utilisés pour déterminer la répartition des jours de maladie dans chaque catégorie d'âge. Ces poids sont traduits à partir de données pour un échantillon de fonctionnaires. Une modélisation plus précise permettrait d'obtenir des résultats plus précis, et l'utilisation de techniques de machine learning pour ajuster les poids constitue une piste intéressante à explorer.
- **Âge** : L'âge de chaque fonctionnaire est déterminé en utilisant des poids correspondant aux proportions d'âges observées au sein du Conseil d'État. La répartition des âges est réalisée de manière cohérente avec les différentes catégories d'âge.
- **Nombre d'enfants à charge** : Le nombre d'enfants à charge de chaque fonctionnaire est calculé en fonction de son sexe. Des poids sont utilisés pour déterminer la répartition des nombres d'enfants dans chaque catégorie de sexe.
- **Zone géographique** : Chaque fonctionnaire est affecté à l'une des trois zones géographiques possibles (1, 2 ou 3) en utilisant des poids correspondant aux proportions observées. Ces zones géographiques exercent une influence sur le salaire de chaque fonctionnaire. Il est donc important de prendre en compte ce paramètre.
- **Nouvelle Bonification Indiciaire (NBI)** : La NBI attribuée à chaque fonctionnaire est calculée en fonction de sa catégorie. Des poids sont utilisés pour déterminer la répartition des bonifications dans chaque catégorie.

5.1.2 Résultats et discussion

La génération aléatoire de fonctionnaires selon la méthodologie décrite ci-dessus permet d'obtenir une population représentative de la formation du Conseil d'État. Les différentes caractéristiques des fonctionnaires, telles que la répartition par catégorie, l'âge, le sexe, le nombre d'enfants, la zone géographique et la NBI, sont cohérentes avec les données observées.

	Cat	IM	arret_maladie	age	enfants	zone_geo	NBI
0	1.0	257.0	0.0	39.0	0.0	1.0	30.0
1	1.0	400.0	4.0	37.0	2.0	3.0	110.0
2	1.0	358.0	0.0	34.0	2.0	1.0	39.0
3	3.0	598.0	30.0	59.0	0.0	2.0	16.0
4	1.0	458.0	0.0	28.0	1.0	1.0	29.0
...	...	...	...	...	...	...	...
4066	1.0	424.0	30.0	48.0	2.0	3.0	75.0
4067	1.0	362.0	30.0	34.0	0.0	2.0	115.0
4068	2.0	288.0	0.0	31.0	0.0	3.0	14.0
4069	1.0	325.0	0.0	25.0	0.0	1.0	82.0
4070	1.0	563.0	0.0	66.0	2.0	1.0	41.0

4071 rows × 7 columns

Figure 6: Exemple d'une génération des membres du Conseil d'Etat au nombre de 4071 pour l'année 2022

Il convient de noter que la génération aléatoire des fonctionnaires permet de créer une population diversifiée, reflétant les différentes caractéristiques présentes au sein du Conseil d'État.

Vis-à-vis de la génération de ces fonctionnaires, une piste d'amélioration serait une meilleure modélisation des poids en proposant une répartition selon un modèle mathématique.

5.2 Associé chaque fonctionnaire à son métier

L'objectif est de générer des informations telles que le métier, la durée d'exercice, la catégorie, l'indice majoré (IM), le grade, l'échelon et l'ancienneté dans le grade pour chaque fonctionnaire simulé. Ces données simulées sont basées sur des paramètres tels que la catégorie, l'indice majoré actuel, l'âge et d'autres facteurs liés à la fonction publique. Afin de classer les fonctionnaires au sein du Conseil d'Etat, nous avons récupéré l'ensemble des données et des classes relatives au conseil d'état, ce qui permet une répartition selon les métiers.

5.2.1 Méthodologie

Le code commence par initialiser des listes vides pour stocker les informations des fonctionnaires simulés. Ensuite, il itère sur le nombre de fonctionnaires souhaité (Nbr_fonct) et effectue les étapes suivantes pour chaque fonctionnaire :

1. Récupération de la catégorie du fonctionnaire à partir d'un DataFrame contenant des données sur les fonctionnaires existants. Ensuite, la catégorie est redéfinie en utilisant des conditions pour correspondre à la notation "A", "B" ou "C".

2. Sélection d'un métier possible pour le fonctionnaire en utilisant un DataFrame (`df`) contenant des informations sur les métiers disponibles pour chaque catégorie. Un métier est choisi de manière aléatoire parmi les métiers disponibles pour la catégorie spécifique.

3. Sélection du grade, de l'échelon et de l'indice majoré pour le fonctionnaire. Le code vérifie d'abord si l'indice majoré actuel du fonctionnaire est inférieur au minimum de l'indice majoré possible pour le métier choisi. Si c'est le cas, l'indice majoré, le grade et l'échelon sont déterminés en fonction du minimum de l'indice majoré possible. Sinon, si l'âge du fonctionnaire est inférieur ou égal à 27, le grade maximum pour le métier est attribué. Sinon, la formule (voir 4.1.2) est utilisée pour .calculer le nombre possible de grades en fonction de l'âge et une probabilité associée. Ensuite, un grade est choisi aléatoirement en fonction de cette probabilité. Enfin, l'indice majoré et l'échelon sont déterminés en fonction de la grille salariale correspondant au grade et au métier.

4. Attribution d'une ancienneté dans le grade aléatoire pour chaque fonctionnaire.

5. Conversion des valeurs de grade, indice majoré et échelon en entiers et stockage dans les listes correspondantes.

6. Ajout de la durée d'exercice pour chaque fonctionnaire en fonction du métier simulé. La durée est récupérée à partir d'un DataFrame (`df`) contenant des informations sur la durée d'exercice pour chaque métier.

5.2.2 Résultats et discussion

Le code permet de générer des données simulées pour une population de fonctionnaires, en prenant en compte différents paramètres tels que la catégorie, l'indice majoré actuel, l'âge et les métiers disponibles. Les données simulées incluent des informations telles que le métier, la durée d'exercice, la catégorie, l'indice majoré, le grade, l'échelon et l'ancienneté dans le grade pour chaque fonctionnaire.

Il convient de noter que ce code repose sur des hypothèses et des modélisations spécifiques pour la génération des données simulées. Des améliorations pourraient être apportées en utilisant des modèles mathématiques plus sophistiqués ou des techniques de machine learning pour ajuster les paramètres de manière plus précise. De plus, des validations et des comparaisons avec des données réelles pourraient être effectuées pour évaluer la pertinence des données simulées générées par ce code.

5.3 Algorithme d'évolution de la masse salariale

Le code `evolve` est une fonction qui permet de simuler l'évolution des fonctionnaires dans leur carrière au sein de la fonction publique. La fonction prend en argument un DataFrame contenant les informations sur les fonctionnaires, telles que leur ancienneté, leur fonction, leur échelon, leur grade et leur indice majoré. Elle prend également en compte le nombre d'années d'évolution souhaité, ainsi que des taux de promotion représentant la proportion maximale d'agents pouvant augmenter leur grade ou leur corps sur une année. Ces taux de promotion sont basés sur ce que notre client a pu nous fournir à ce sujet.

La fonction commence par initialiser une variable de contrôle appelée `changement` pour chaque fonctionnaire, qui représente le nombre de changements dans la situation salariale de l'agent. Ensuite, elle itère sur le nombre d'années d'évolution souhaité et pour chaque année et chaque fonctionnaire, elle effectue les étapes suivantes :

1. Elle récupère les informations spécifiques de l'agent à partir du DataFrame, telles que son ancienneté, sa fonction, sa catégorie, son échelon, son grade, son indice majoré et sa durée d'exercice.

2. Elle charge un DataFrame appelé `grille` correspondant à la grille salariale de l'agent pour l'année précédente, en se basant sur sa fonction.

3. Si la durée d'exercice de l'agent est supérieure ou égale à 10 ans, sa durée d'exercice est réinitialisée à 0. Ensuite, si la durée d'exercice correspond à une évolution d'échelon possible pour l'agent, son échelon est augmenté et son indice majoré est mis à jour en fonction de la grille salariale correspondante. Le nombre de changements pour cet agent est incrémenté. Les fonctions de passages ne sont pas exactement celles du Conseil d'Etat mais vis à vis d'une première approche, cela corresspond à une approche plus intuitive et plus compréhensible.

4. Si un nombre aléatoire inférieur au taux de promotion de grade est inférieur, ou si l'agent a atteint le grade maximum (grade 7), l'agent est éligible pour une promotion de grade. Dans ce cas, son grade est diminué d'une unité, son ancienneté est réinitialisée à 0, et un nouvel indice majoré et échelon sont déterminés en fonction de la grille salariale du nouveau grade. Le nombre de changements pour cet agent est incrémenté. Encore une fois, cette description n'est pas exactement représentative de la réalité.

5. Si un nombre aléatoire inférieur au taux de promotion de corps est inférieur, ou si l'agent a atteint le grade maximum (grade 10), l'agent est éligible pour une promotion de corps. Dans ce cas, la catégorie de l'agent est mise à jour (par exemple, de 'C' à 'B' ou de 'B' à 'A'), son ancienneté est réinitialisée à 0, et un nouveau métier est choisi de manière aléatoire parmi ceux disponibles pour la nouvelle catégorie. Ensuite, un nouvel indice majoré, échelon et grade sont déterminés en fonction de la grille salariale du nouveau métier. Le nombre de changements pour cet agent est incrémenté.

6. Enfin, la fonction calcule une prime pour chaque agent en fonction de certaines variables, telles que l'âge, les arrêts maladie, la catégorie, et les constantes spécifiées. La nouvelle prime est arrondie à deux décimales et mise à jour dans le DataFrame.

Ce processus d'évolution est répété pour chaque année et chaque agent dans le DataFrame, permettant de simuler l'évolution des carrières des fonctionnaires sur plusieurs années.

5.4 Résulat de la modélisation

Grâce à cette modélisation de l'évolution du Conseil d'état, en rentant différent paramètres tels que le point d'indice, le nombre de fonctionnaires, nous obtenons une prévision de l'évolution de la masse salariale allouée au Conseil d'Etat.

Nous obtenons différents résultats tels que l'évolution salariale pour chaque fonctionnaire :

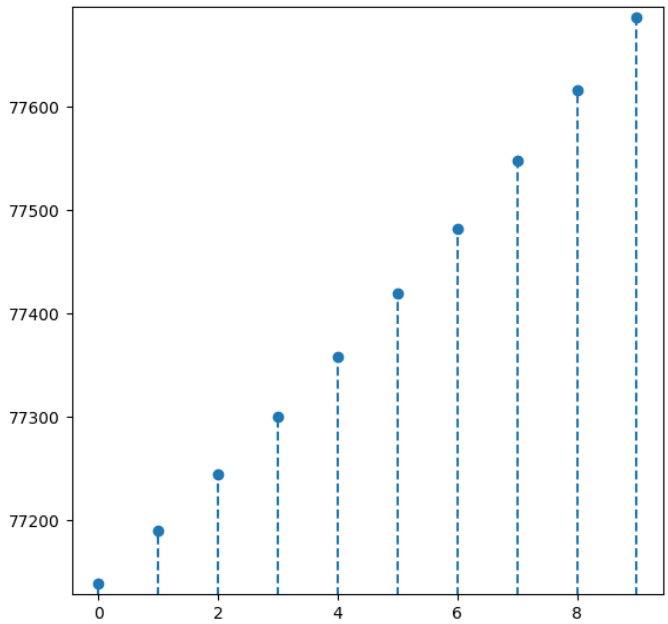

Figure 7: Evolution du salaire pour un agent du conseil d'Etat

Cette figure montre une évolution plutôt linéaire avec une prévision à 10 ans de la masse salariale d'un certain agent. On observe notamment que cet agent n'a pas changé de catégorie, ni n'a eu de grandes variations de ses primes.

Valeurs de la masse salariale pour les années à venir: Avec les données de l'année 2022, en gardant le point d'indice constant, l'évolution du conseil d'état indique une Il convient de noter que ce code repose sur certaines hypothèses

Table 1: Évolution de la masse salariale

Année	Masse salariale
2022	245,691,071.289
2023	247,596,158.046
2024	249,500,665.258
2025	251,203,735.563
2026	252,970,109.851
2027	254,755,862.438
2028	256,548,375.842
2029	258,435,551.132
2030	260,419,663.477

et modélisations spécifiques pour générer les évolutions de carrière. Des améliorations pourraient être apportées en utilisant des modèles plus sophistiqués ou des techniques de machine learning pour ajuster les paramètres de manière plus précise. De plus, des validations et des comparaisons avec des données réelles pourraient être effectuées pour évaluer la pertinence des données simulées générées par ce code.

De même que nous obtenons l'évolution de la masse salariale globale pour le Conseil d'Etat :

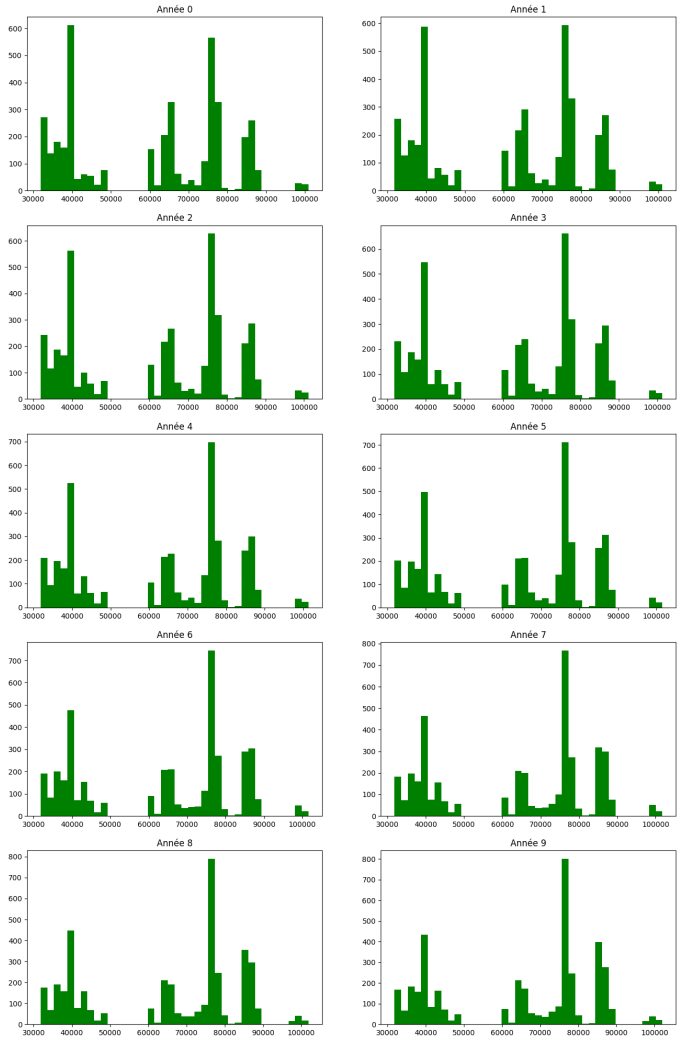

Figure 8: Evolution de la masse salariale du Conseil d'Etat

Cette évolution montre que la masse salariale évolue de manière non négligeable et la répartition connait des grosses variations. Dans le cadre de cette modélisation en particulier avec les paramètres de l'année 2022, nous obtenons une prévision à 10 ans de cette masse salariale.

6 Prédiction de la masse salariale grâce au Machine Learning

Cette partie comporte de nouvelles méthodes imaginées dans le cadre de la continuité du projet au semestre 7. Nous avons décidé afin d'améliorer les performances de notre logiciel d'avoir recours à des methodes d'apprentissage afin de déterminer les paramètres du système.

6.1 Time-Series Cross Validation

6.1.1 Première approche

Une première approche fut de faire une régréssion linéaire sur l'évolution de la masse salariale totale du conseil d'État de 2005 à 2022. Cependant, afin d'obtenir la fonction affine qui correspond le mieux, nous avons utilisé la methode de cross-validation :

On note notre fenêtre d'observation temporelle de l'évolution de la masse salariale $\mathcal{A} = [\![2005, 2022]\!]$. On choisit de façon aléatoire $n \in N$ sous-ensembles stricts distincts et ordonnés $A_k \subseteq \mathcal{A}$, $k \in [\![1, n]\!]$. On effectue une régression linéaire sur chaque A_k qu'on teste sur l'ensemble $\mathcal{A} \backslash A_k$ en calculant l'erreur quadratique moyenne. Cette méthode permet de trouver la régression linéaire la plus représentative de l'évolution de la masse salariale du conseil d'état. Deux méthodes ont alors été mises en place:

- Trouver la régression linéaire avec une RMSE (erreur quadratique) minimale.
- Pondérer chacune des régression linéaire inversement par rapport à leur RMSE pour obtenir une.

La méthode la plus efficace aura été celle utilisant uniquement la régression linéaire avec le plus petit RMSE. Dans un premier temps nous avons considéré uniquement les données que nous avions pour les années: $\mathcal{A} = [\![2005, 2022]\!]$. En obtenant un "split" comme étant celui conduisant au plus petit RMSE, on arrivait au résultat suivant:

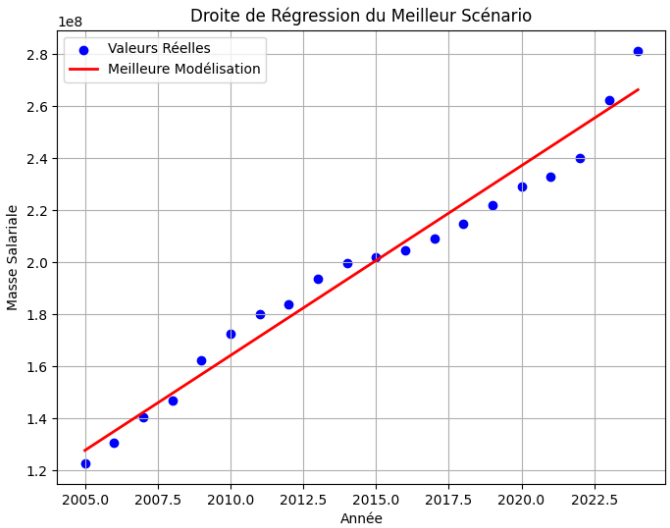

Figure 9: Régression linéaire sur la masse salariale minimisant le RMSE entre 2005 et 2022

Sur cette figure nous observons que la valeur de la masse salariale attendue en 2023 en prenant en compte uniquement les années précédentes est proche de la réalité. Cependant en visualisant l'évolution de la masse salariale au cours des dernières années, on visualise bien l'absence d'augmentation du point d'indice pendant 6 années entre 2010 et 2016 puis également entre 2017 et 2022 malgré un effectif en hausse.

Les années 2023 et 2024 marquent une rupture avec les années précédentes:

- Augmentation significative du point d'indice, passage de 5 820 à 5 907.
- Augmentation significative de l'effectif du conseil d'état, 200 membres de plus sont embauchés par rapport aux départs.

Cette méthode de Time-series cross validation nous permet d'obtenir une moyenne sur l'ensemble des entrainements de la moyenne quadratique entre les prédictions et les valeurs réelles: RMSE moyen sur l'ensemble de validation : 6439366.6170

Pour cela dans un second temps nous avons voulu évaluer une seconde régression linéaire prenant en compte également les nouvelles données 2023 et 2024 que nous n'avions pas considéré jusqu'alors. Cette approche a pour objectif de visualiser malgré les changements brusques signalés précédement, quelle serait la feuille de route ui devrait être suivie dans les années à venir.

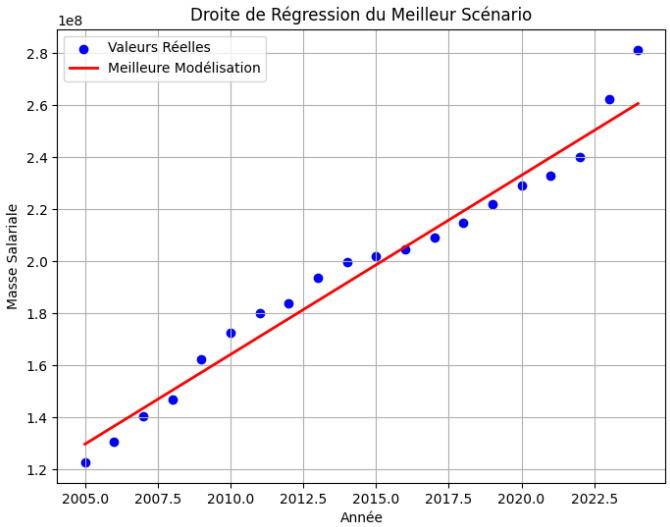

Figure 10: Régression linéaire sur la masse salariale minimisant le RMSE entre 2005 et 2024

Etant donné l'évolution de la masse salariale, nous avons voulu tester d'autres algorithmes d'intelligence artificielle.

6.2 Random Forest

6.2.1 Random Forest - Régression Linéaire

Dans notre quête de mettre en place des algorithme d'intelligence artificielle adaptés à notre modèle, nous avons dans un premier temps essayé par le même principe que précédement d'implémenter random forest sur de la régression linéaire. Comme cela pouvait être prévisible nous avons fait face à un problème récurrent en algorithmique qui n'est autre que l'overfitting: l'utilisation de méthode trop complexes pour un jeu de données qui n'en demande pas autant. Comme nous pouvons le voir sur le graphique ci-dessus, notre absence de jeu de données trop, important nous conduit à un overfitting qui ne permet pas de prédire avec précision les valeurs de la masse salariale dans les années à venir.

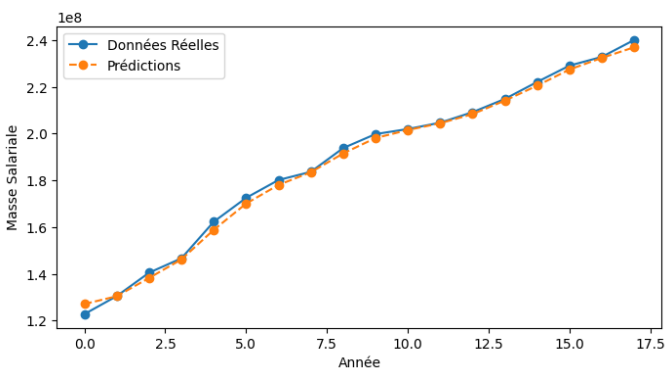

Figure 11: Overfitting du Random Forest sur la régression linéaire

Suite à cette étude, nous avons décidé d'implémenter l'algorithme Random Forest sur notre modélisation de l'évolution de la masse salariale et pas uniquement sur l'évolution graphique de cette dernière. Pour cela nous avons repensé notre algorithme d'évolution pour faire passer en entrée d'autres paramètres (qui peuvent être changés) permettant de visualiser l'importance de certains paramètres sur l'évolution de la masse salariale du Conseil d'Etat.

6.2.2 arbres décisionnels

Un arbre de décision est un modèle d'apprentissage automatique qui est utilisé pour prendre des décisions en formulant un ensemble de règles sous forme d'une structure arborescente. Ces arbres sont largement utilisés dans les domaines de la classification et de la régression. Voici les éléments de base d'un arbre de décision :

- Le Nœud racine : C'est le premier nœud de l'arbre, qui divise l'ensemble de données en sous-ensembles basés sur une caractéristique particulière. Celle-ci consiste en une condition booléenne.
- Les Nœuds internes : Ce sont les nœuds qui suivent le nœud racine et qui continuent à diviser l'ensemble de données en fonction de différentes caractéristiques. Chaque nœud interne représente une règle de décision basée sur une caractéristique particulière.
- Feuilles (nœuds terminaux) : Ce sont les nœuds finaux de l'arbre où aucune division supplémentaire n'est effectuée. Chaque feuille représente une classe (dans le cas de la classification) ou une valeur (dans le cas de la régression) attribuée à l'observation. Dans notre cas les feuilles représentent le salaire.
- Branches : Les branches relient les nœuds et représentent les résultats des tests effectués sur les caractéristiques.

L'interêt de la construction d'un arbre décisionnel à partir d'un jeu de donné est de pouvoir classifier toute nouvelle donnée et l'attribuer à une feuille. (Ici à partir d'un fonctionnaire, on veut pouvoir déduire son salaire.) Ou bien alors, à partir de certaines de données, pouvoir retrouver les données manquantes. (Par exemple, trouver le grade le plus probable d'un fonctionnaire à partir de son IM, son métier etc...). L'algorithme random forest se base sur l'utilisation de ces arbres décisionnels.

6.2.3 Construction d'arbres décisionnels

On note T=(x,y) l'ensemble de nos observations, un tableau avec $x = (x_1, ..., x_n)^T$ où les x_i sont des échantillons de taille p avec chaque indice $j \in [\![1, p]\!]$ correspondant à une caractéristique (par exemple l'age, le nombre d'individu dans une catégorie, ...) et $y = (y_1, ..., y_n)^T$ est un n-uplet correspondant aux valeurs ou classe qu'on cherche à prédire (Ici on cherche la masse salariale) (c'est ce qui correspondra aux feuille de l'arbre de décision qu'on veut construire).

L'algorithme de construction de notre arbre de décision est le suivant :

Algorithm 1 Arbre de Décision

```
function ArbreDecision(T)
    if condition d'arrêt then
        return feuille(T)
    else
        Choisir le "meilleur" attribut i entre 1 et p
        for chaque valeur v de l'attribut i do
            T[v] ← {(x, y) ∈ T | x_i = v}
            t[v] ← ArbreDecision(T[v])
        end for
        return noeud(i, {v → t[v]})
    end if
end function
```

6.2.4 Le principe de l'algorithme

Le principe de l'algorithme est de construire récurcivement l'arbre en commencant par choisir tout d'abord le noeud racine qui correspond au "meilleur" attribu pour la régression ou classification. C'est à dire le plus correllé avec la valeur finale y (Cette mesure de la corrélation sera détaillée plus tard). Après avoir choisi cet attribut j, on parcours toute les valeurs prises pour i par les x_{ij} et pour chacune d'entre-elles, on crée un sous-ensemble de nos observations $T_{ij} = (x_k, y_k), x_{kl} = x_{ij}$ et on applique à nouveau l'algorithme à ce sous-ensemble de donnée afin de trouver la meilleure variable d'interêt à nouveau etc.. Jusqu'à la condition d'arrêt.

Dans notre cas la condition d'arrêt pour un arbre est soit lorsque toutes les conditions sur les attribus apparassaient déjà chaque chemin de l'arbre, soit lorsque la classification de nos données est déjà parfaite (c'est à dire qu'à une certaine iteration de notre algorithme il existe un attribu parfaitement discriminant) ou quasi-parfaite (On fixe un seuil sur le nombre de x_i de chaque subdivision de notre data set $\tilde{T}$ afin de diminuer le temps de calcul)

Le choix du meilleur attribu se fait différement dans le cas d'une classification ou d'une régression, à chaque fois l'objectif est de trouver l'attribut minimisant la dispersion des échantillons. Dans notre cas, (une régression) l'ensemble des échantillons $T_i \in T$ peuvent être vus comme des variables aléatoires iid. ainsi pour tout sous-ensemble d'échantillon $\tilde{T}$ et tout choix d'attribut j, on produit une partition $\tilde{T} = \cup_{v_i} T_{v_i}$ où les T_{ij} représentent les sous-échantillons pour la valeur $v_i = x_{ij}$. La variance attendue après un branchement sur l'attribut i (pour une instance (x, y) tirée uniformément au hasard dans T) est alors :

$$V_i = \sum_{v_i} \frac{|T_{v_i}|}{T} V(T_{v_i})$$

On va alors chercher le i minimisant cette variance pour en faire le noeud racine.

6.2.5 Algorithme Random Forest

l'algorithme Random Forest se base sur la construction d'arbres décisionnels pour aider à la régression ou bien la classification. On commence par recueuillir des données (x,y) où x correspond à l'ensemble des données d'interêt pour la régression ou classification (valeurs, booléens,...) et y la valeur ou le caractère qu'on cherche à prédire (ici, la masse salariale).

Étape 1 : création d'un ensemble de données pour la régression et pour le test
On découpe nos données en un dataset de régression et un dataset de test afin de pouvoir réellement tester si notre régression est conforme et on test plusieurs fois l'algorithme sur plusieurs datasets différents (cf. la partie sur la cross-validation).

Étape 2 : Scrapping d'un nombre aléatoire de données et création des arbres de décision
Après avoir récuperé notre $x_regression$, on va pouvoir lui appliquer l'algorithme. x est un n-uplet dont chaque x_i correspond à échantillon dont la ligne j correspond à l'attribut j. On va selectionner un nombre aléatoire d'attributs j et créer un arbre de décision par la methode décrite précedement en utilisant nos données test et uniquement les attributs selectionnés.

Etape 3: Répétition du processus et résultat
On va créer plusieurs arbres sur le dataset de régréssion choisi et tester en implémentant nos données test sur chacun des arbres T_i le résultat de la prédiction est la médiane de chacun des résultats des arbres.

6.2.6 Algorithme de notre code

Dans notres cas, on a utilisé le code random forest de la bibliothèque sklearn sur notre dataset dont les attributs sont : le nombre de membre du conseil d'État, le point d'indice, le nombre d'individu de catégorie A, de catégorie B et de catégorie C et on cherche à prédire la masse salariale à partir de ces attributs. Par ailleurs, on pourrait modifier ces paramètres simplement pour analyser l'importance et complexifier l'algorithme. En implémentant cela sur notre prévision de la masse salariale nous pouvons analyser les résultats et évaluer la pertinence d'un tel algorithme. Par exemple, nous obtenons une RMSE de l'ordre de Mean Squared Error: 282198120726495.9 Nous remarquons alors qu'une régression linéaire est beaucoup plus efficace pour prévoir la masse salariale du Conseil d'Etat. Nous faisons encore face à de l'overfitting ici.

Cependant cette méthode nous permet d'effectuer une série d'observations sur l'importance que peuvent avoir certains paramètres sur l'écart entre la valeur réelle et les valeurs évaluées par algorithme.

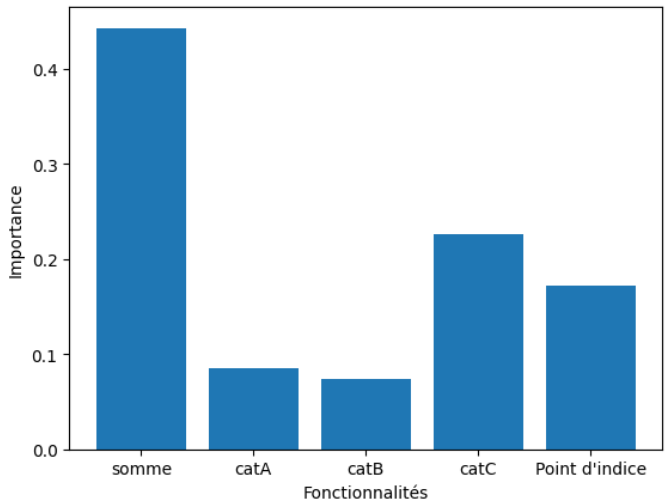

Figure 12: Importance des paramètres quant aux prédictions

Nous avons pu également visualiser le fonctionnement des arbres décisionnels adaptés à nos données pour comprendre leur fonctionnement et les choix effectués sur les modélisations:

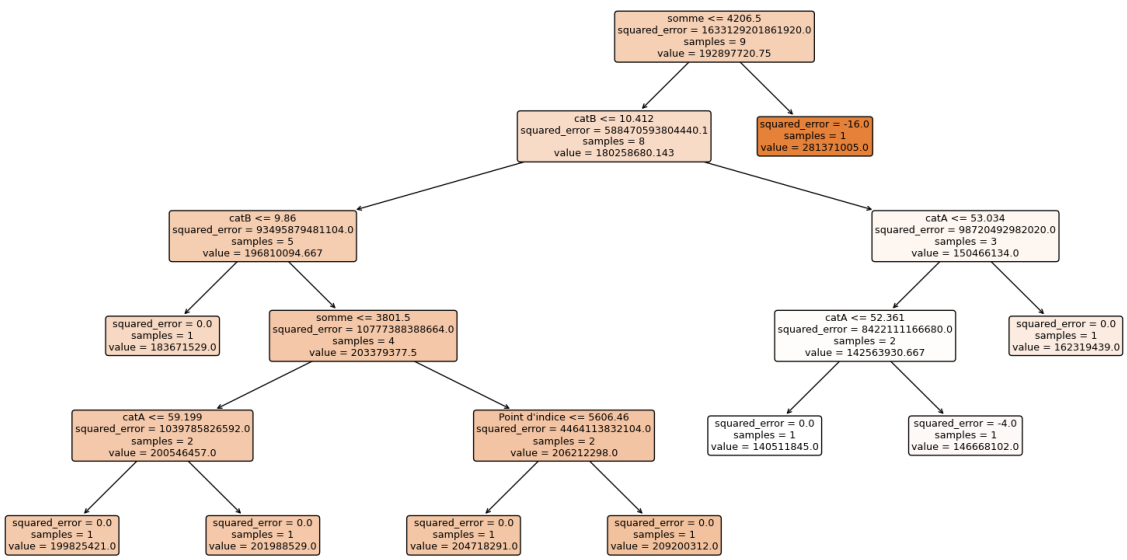

Figure 13: Arbre décisionnel

6.3 XGBoost

6.3.1 Construction d'arbre

La construction d'arbre pour l'algorithme XGBoost suit le même algorithme que précédement, cependant, la fonction permettant de trouver le i optimal est différente.

Cette fois ci on suppose que nos variables et attributs sont tous numériques (on fait une régression). On construit un noeud grâce à une frontière numérique sur l'attributs ($x_i < f$ par exemple).

On commence par se fixer une prédiction arbitraire pour tout x, p^0. On s'intéresse par la suite aux $r_i^0 = y_i - p^0$ les résidus par rapport à cette première prédiction. L'objectif de l'algorithme XGBoost dans la construction d'arbre est de créer des frontières de décision sur X pour le noeud considéré. Pour se faire, on calcule le gain de chaque frontière de décision pour chaque attribut possible

$$G = Left_{similarity} + Right_{similarity} - Root_{similarity} \tag{7}$$

où la similarité d'un noeud ou d'une feuille se calcule par l'expression :

$$\frac{(\sum_{j \in J} r_j)^2}{|J| + \lambda} \tag{8}$$

avec J les indices des différents échantillons passant par le noeud ou la feuille considerée avec λ un paramètre de régularisation. L'attribut choisi pour le noeud est alors celui qui maximise le gain avec la meilleure frontière possible.

6.3.2 Gradient Boost

L'algorithme Gradient Boost est un algorithme très proche de la descente de gradient à la différence que celui-ci se base sur des arbres de décision pour la regression ou la classification. On commence par introduire comme pour la descente de gradient un pas ρ et une fonction de perte $L(.,.)$ (On choisira l'erreur quadratique).

Étape 1 :

On effectue une première prédiction constante:

$$f^0(x) = argmin_\gamma \sum_i^p L(y_i, \gamma) \tag{9}$$

(Pour L l'erreur quadratique)

Étape 2 : Pour tout m allant de 1 à M (M étant l'étape finale fixée) :

On suppose f^{m-1} construite (il s'agit de fonction de prédiction à l'itération m-1), on pose

$$\forall i \in [\![1, p]\!] r_{im} = -[\frac{\partial L(y_i, f^{m-1}(x_i)}{\partial f^{m-1}(x_i)}] \tag{10}$$

valeur correspondant au résidu par rapport à la prédiction de f^{m-1} sur l'échantillon i. On effectue un arbre sur le résidut r_{im} et on numérote les noeuds terminaux R_{jm} pour chaque arbre de cette m-ième itération. On calcule alors:

$$\gamma_{jm} = argmin_\gamma \sum_{x_j} \in R_{jm} L(y_i, f^{m-1}(x_i) + \gamma \tag{11}$$

c'est à dire la valeur de gamma sur R_{jm} minimisant l'erreur. Et on actualise la fonction de prédiction au temps m :

$$f^m(x) = f^{m-1}(x) + \rho \sum_j \gamma_{jm} I(x \in R_{jm}) \tag{12}$$

On réitère l'algorithme ainsi plusieurs fois.

6.3.3 Notre implémentation à la régression linéaire

Nous avons dans un premier temps cherché à l'implémenter seulement aux données de la masse salariale pour effectuer l'algorithme de xgboost sur la régression linéaire. Notre motivation pour effectuer ce travail en plus de celui de Random Forest était la possibilité d'introduire plus facilement une variable exogène pour améliorer la précision. Nous avons décidé d'utiliser, après étude de la corrélation, le nombre de fonctionnaires dans le conseil d'état comme étant notre variable exogène. Cela nous a conduit à obtenir de nouveau un phénomène d'overfitting:

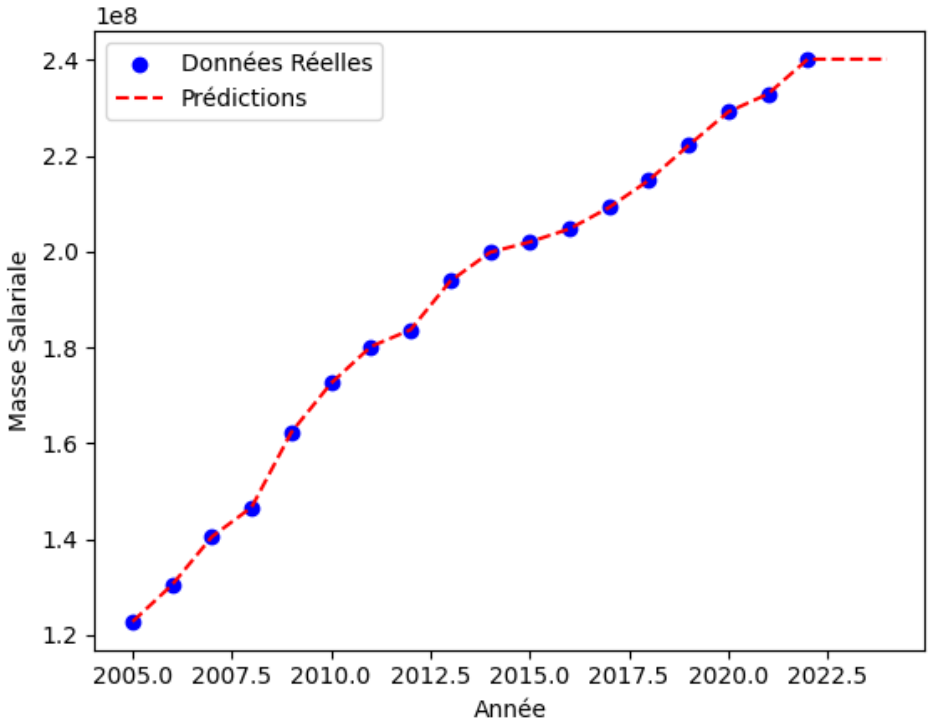

Figure 14: Overfitting sur XGBoost

C'est pourquoi dans un second temps nous avons décidé d'implémenter à l'instar de la méthode précédente, l'algorithme XGBoost directement sur notre algorithme de prédiction de la masse salariale. L'implémentation est proche de celle de Random Forest, cependant l'algorithme XGBoost permet d'accorder une attention particulière aux exemples mal classés en modifiant les poids à chaque itération. Cela améliore progressivement la performance du modèle en se concentrant sur les erreurs, étant donné que notre modèle peut fournir des erreurs systématiques, cette particularité est un atout pour notre code. Après avoir implémenté notre algorithme XGBoost sur notre système, nous avons pu obtenir des résultats quant aux prédictions de l'algorithme, on obtient toujours un RMSE qui est trop grand par rapport aux modèles basés sur la régression linéaire. Ce qui se confirme par la matrice de confusion de la prédiction des valeurs par rapport aux valeurs réelles. Une majorité de 0 indique de mauvaises prédictions.

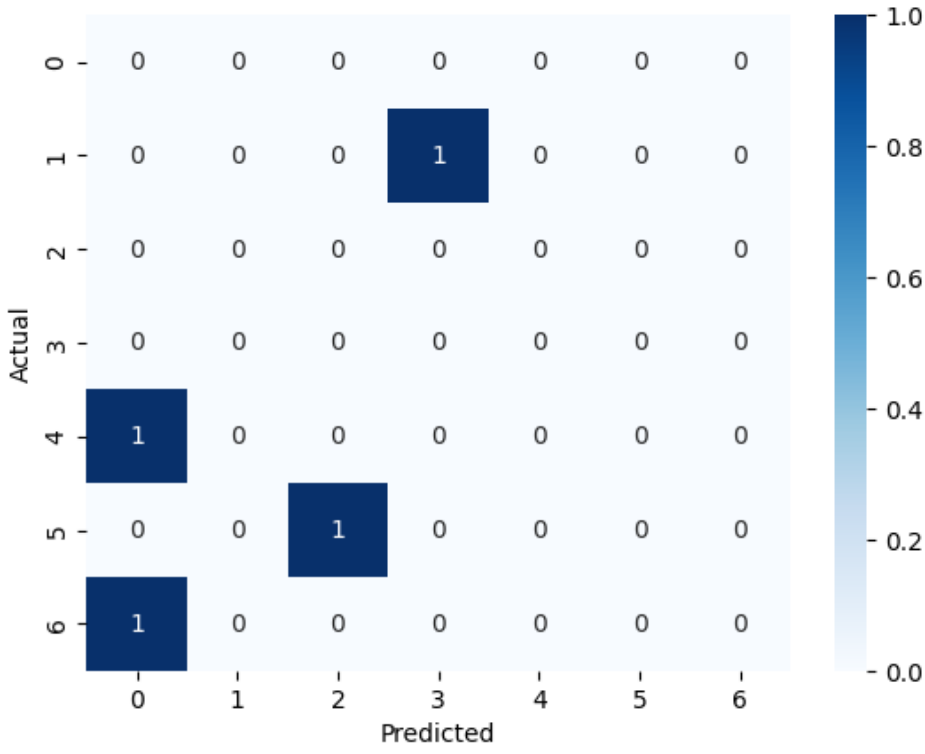

Figure 15: Matrice de confusion

6.3.4 Analyse d'une Matrice de Confusion

La matrice de confusion est un outil essentiel pour évaluer la performance d'un modèle de classification. Elle résume les résultats des prédictions en comparant les prédictions du modèle avec les classes réelles. La matrice est souvent présentée sous la forme suivante :

	Prédit Positif	Prédit Négatif
Réel Positif	TP	FN
Réel Négatif	FP	TN

où TP (True Positive), FN (False Negative), FP (False Positive) et TN (True Negative) sont les éléments de la matrice.

Une matrice qui contient un grand nombre de 0 0 et très peu de fois la valuer 1 indique une matrice déséquilibrée. Ceci montre que le modèle pourrait être être biaisé vers la classe majoritaire. Ainsi le modèle a une performance élecée pour la classe négative mais il a du mal à détecter la classe positive (faible nombre de TP). Pour expliquer cela, il pourrait être intéressant dans une étude complémentaire d'explorer des métriques plus spécifiques, telles que le rappel, pour évaluer la capacité du modèle à détecter la classe minoritaire.

7 Interface graphique

Dans le cadre du projet au semestre 7 nous avons décidé de mettre en place une interface homme-machine grâce à tkinter afin de facilité l'utilisation du programme. Lorsque le document *interface_finale.py* est lancé une fenêtre apparait pour nous proposer de soit laisser le programme choisir des paramètres de lui-même (voir partie 4) soit de donner nous-même les valeurs des paramètres. Si on choisit de laisser le programme faire de lui-même, on doit cependant entrer un nombre de fonctionnaires et une valeur du point d'indice.

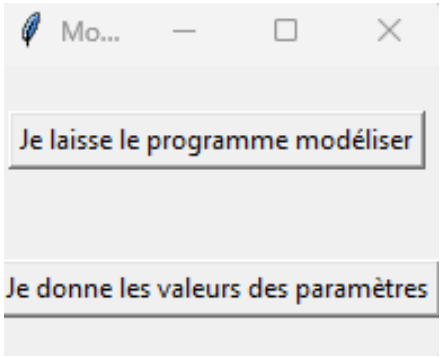

Figure 16: Interface graphique au lancement du programme

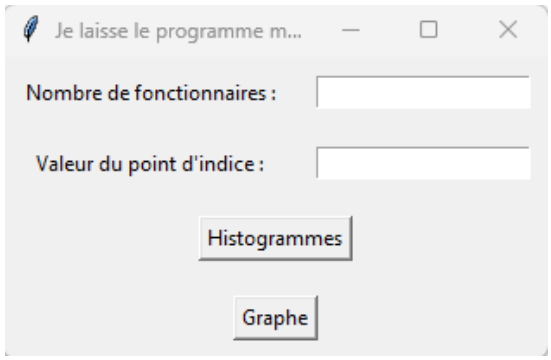

Figure 17: Interface graphique lorsque l'on choisit de laisser le programme modéliser

Lorsque l'on choisit d'entrer nos propres paramètres, une fenêtre s'ouvre qui nous demande d'entrer le nombre de fonctionnaire, la valeur du point d'indice, la proportion d'employés en catégorie A, B et C, la proportion agés entre 20 et 30 ans, 30 et 50 ans et 50 et 68 ans, la proportion par zone, la proportion par sexe ainsi que les valeurs du taux de passage entre les corps ou les grades, pour relancer l'algorithme décrit partie 4 avec ces valeurs. Le programme nous propose ensuite d'afficher l'évolution de la masse salariale soit par un graphe soit par un histogramme.

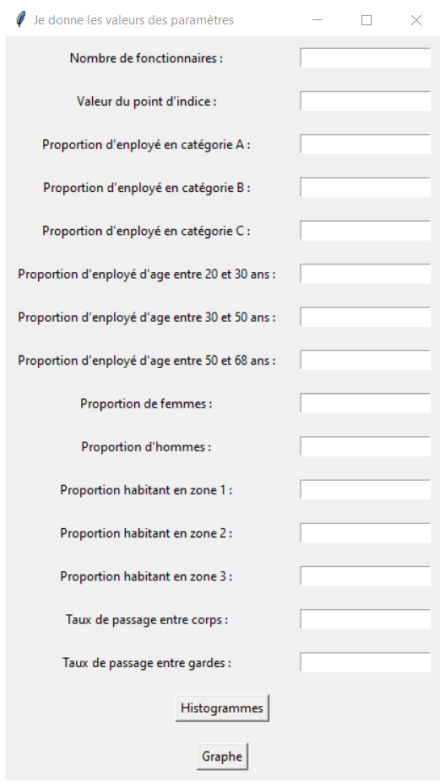

Figure 18: Interface graphique lorsque l'on choisit de remplir les paramètres

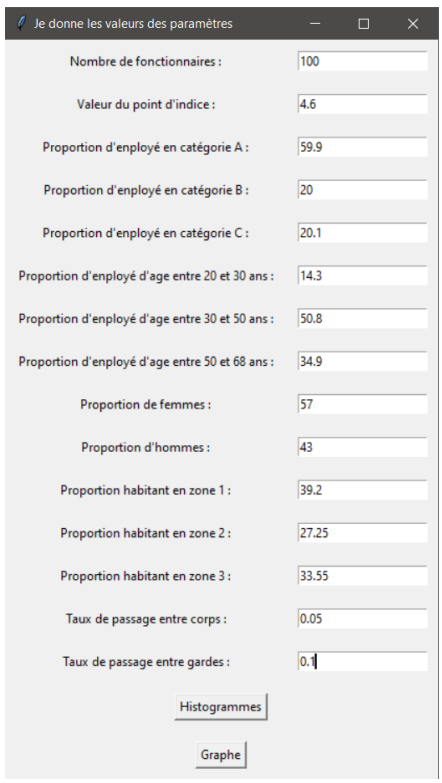

Figure 19: Interface graphique remplie avec le jeu de paramètres

On appuie sur le boutton "Histogrammes" et on obtient la figure suivante:

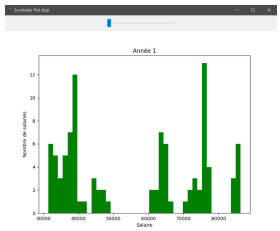

Figure 20: Année 1

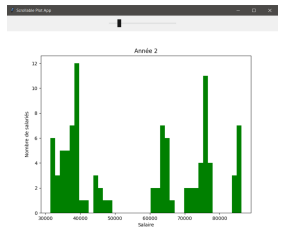

Figure 21: Année 2

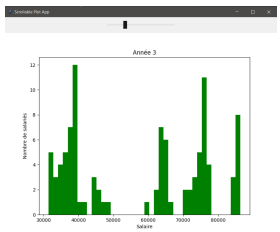

Figure 22: Année 3

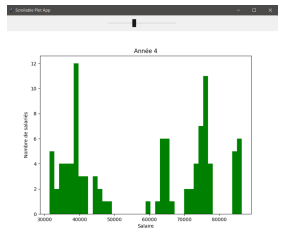

Figure 23: Année 4

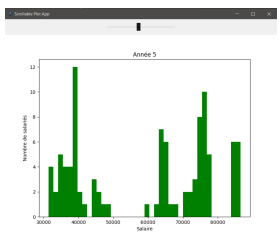

Figure 24: Année 5

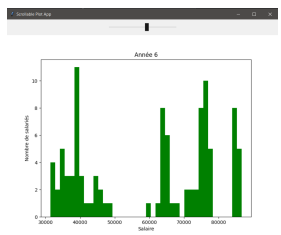

Figure 25: Année 6

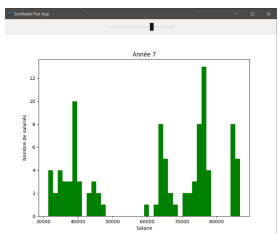

Figure 26: Année 7

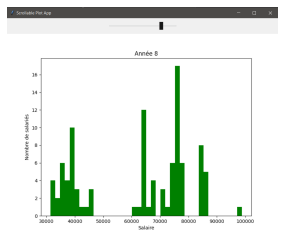

Figure 27: Année 8

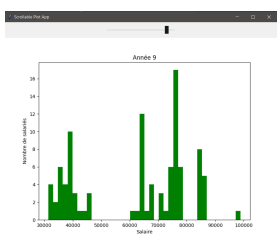

Figure 28: Année 9

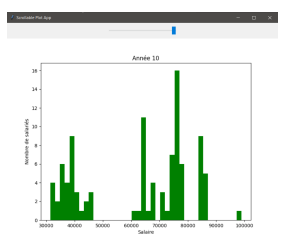

Figure 29: Année 10

Figure 30: Histogrammes d'évolution des salaires individuels sur 10 ans

Le basculement entre les différentes années se fait grace au curseur glissant se situant en haut de la figure.
On appuie ensuite sur le boutton "Graphe" et on obtient la figure suivante:

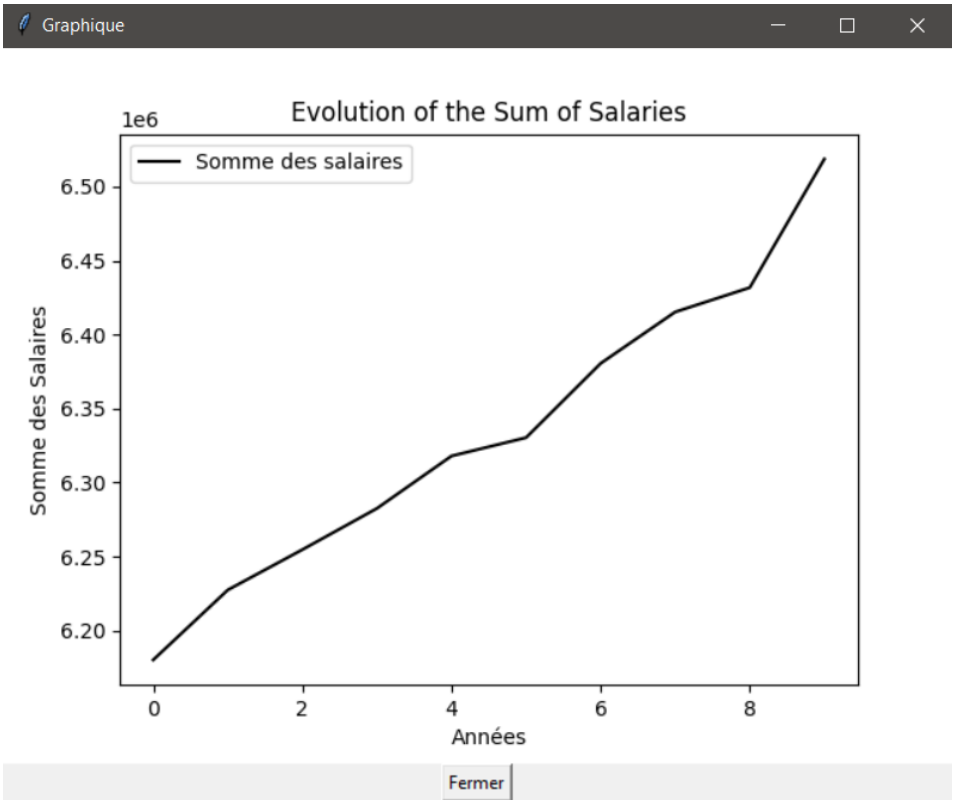

Figure 31: Interface graphique lorsque l'on choisit les paramètres

8 Conclusion

Le modèle choisi permet de capturer l'évolution stochastique des salaires des fonctionnaires en prenant en compte les règles de passage, les promotions de grade et de corps, ainsi que l'évolution naturelle des indices majorés. En utilisant ce modèle, nous pouvons simuler différents scénarios et étudier les effets de différents paramètres sur les salaires des fonctionnaires. Nous avons par ailleurs montré que celui-ci s'implémente également parfaitement sous la forme d'un algorithme.

Nous avons ici défini de nous même les paramètres du modèle mais pour améliorer la modélisation des indices majorés, nous pouvons également utiliser des techniques de machine learning, telles que la méthode de Random Forest. Random Forest est un algorithme d'apprentissage supervisé qui utilise un ensemble d'arbres de décision pour effectuer des prédictions. Dans notre cas, nous pouvons entraîner un modèle Random Forest en utilisant les données historiques des indices majorés et leurs années correspondantes comme entrée, et le point d'indice comme sortie. Cela nous permettra de prédire l'évolution future du point d'indice en fonction des années.

Le formalisme adapté à la méthode de Random Forest est le suivant :

Soit X une matrice de taille $N \times M$ contenant les caractéristiques des indices majorés (par exemple, les années) et Y un vecteur de taille N contenant les valeurs du point d'indice correspondantes. Nous pouvons entraîner le modèle Random Forest en utilisant la fonction $Y = f(X) + \epsilon$, où f est la fonction de prédiction et ϵ est le terme d'erreur. Le modèle tente d'apprendre la fonction f en minimisant l'erreur quadratique moyenne entre les prédictions et les valeurs réelles.

9 Bibliographie

References

[1] www.service-public.fr. Fiche de paie dans la fonction publique : quelles sont les règles ?. https://www.service-public.fr/particuliers/vosdroits/F34231.

[2] www.lafinancepourtous.com. La rémunération des fonctionnaires. https://www.lafinancepourtous.com/pratique/vie-pro/fonctionnaires/la-remuneration-des-fonctionnaires/.

[3] budget.gouv.fr. Conseil et Contrôle de l'État. https://www.budget.gouv.fr/documentation/documents-budgetaires/exercice-2022/projet-de-loi-de-finances/budget-general/conseil-et-controle-de-letat.

[4] Pierre-Loïc Méliot. Cours sur les chaînes de Markov de Pierre-Loïc Meliot (Université Paris-Saclay) :. https://www.imo.universite-paris-saclay.fr/~pierre-loic.meliot/agreg/markov.pdf.

[5] Laurent SAINT-MARTIN. RAPPORT FAIT AU NOM DE LA COMMISSION DES FINANCES, DE L'ÉCONOMIE GÉNÉRALE ET DU CONTRÔLE BUDGÉTAIRE SUR LE PROJET DE loi de finances pour 2022 (n° 4482). https://www.assemblee-nationale.fr/dyn/opendata/RAPPANR5L15B4524-tIII-a9.html.

[6] www.emploi-collectivites.fr. Salaire attaché d'administration de l'etat -corps interministériel - cigem . https://www.emploi-collectivites.fr/grille-indiciaire-etat-attache-administration-corps-interministeriel-cigem/1/5051.htm.

[7] Légifrance. Décret n° 2011-1317 du 17 octobre 2011 portant statut particulier du corps interministériel des attachés d'administration de l'Etat. https://www.legifrance.gouv.fr/loda/id/LEGIARTI000028020152/2013-10-02/.

[8] Centre Départemental de Gestion de la Fonction Publique Territoriale des Hautes-Pyrénées. Régime indemnitaire tenant compte des fonctions, des sujétions, de l'expertise et de l'engagement professionnel (RIFSEEP). https://www.cdg65.fr/grh_remuneration_acc_trait_rifseep.php.

[9] Stéphane Caron. Une introduction aux arbres de décision. https://scaron.info/doc/intro-arbres-decision/#rf.

[10] Xgboost developers. Tree Methods– XGboost https://xgboost.readthedocs.io/en/stable/treemethod.html

[11] Josh Starmer. StatQuest: Random Forests Part 1 - Building, Using and Evaluating https://youtu.be/J4WdyOWc_xQ?si=bcW2Fb6c8UN7KdB_

[12] Josh Starmer. Gradient Boost Part 2 (of 4): Regression Details https://www.youtube.com/watch?v=2xudPOBz-vs