Test technique Quantmetry

1. **Statistiques descriptives**

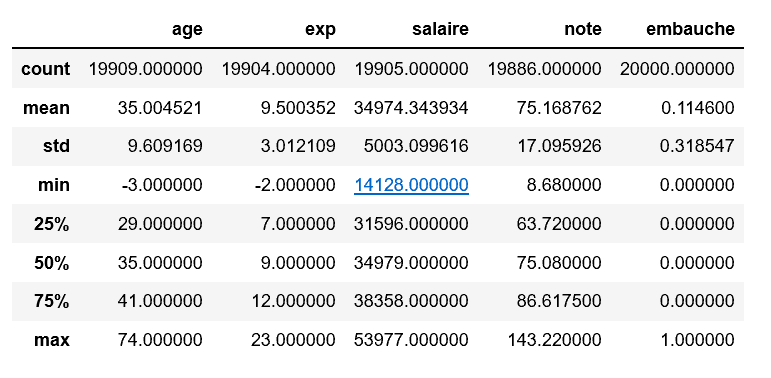
1.1

Après import du jeu de données, nous constatons que celui-ci comporte n=**20.000 individus** et p=**11 variables**, qui correspondent bien à celles décrites dans l’énoncé.

Premier constat, nous sommes face à un **échantillon de grande taille**, ce qui ne sera **pas limitant pour les algorithmes utilisés**. De plus, p est petit et nous n’aurons pas à recourir à des algorithmes de réduction de dimension pour palier au fléau de la dimension.

Le jeu de données comprend :

* **4 variables quantitatives** (age, exp, salaire, note)
* **6 variables qualitatives** (date, cheveux, sexe, diplome, specialite, dispo) dont une ordinale (date) et les autres nominales.

La méthode describe permet de décrire statistiquement nos variables quantitatives.

Nous constatons des valeurs aberrantes :

* Des **valeurs négatives pour age et exp** (ou age < exp) : comme il n’y a que 106 valeurs dans ce cas, **je supprime ces valeurs de l’analyse**
* Des **valeurs > 100 pour la variable note** : il y a 1452 valeurs concernées. Malgré ce nombre important, je fais le choix de **les exclure de l’analyse**, ne sachant pas d’où vient l’erreur (barème sur plus de 100, erreur de virgule…)

De plus, nous constatons que la variable à prédire, embauche, possède des **classes très déséquilibrées** (~10% d’embauche). Ceci risque d’**impacter le choix de l’algorithme et de sa fonction d’évaluation**.

L’affichage du tableau de valeurs manquantes donne les résultats suivants :

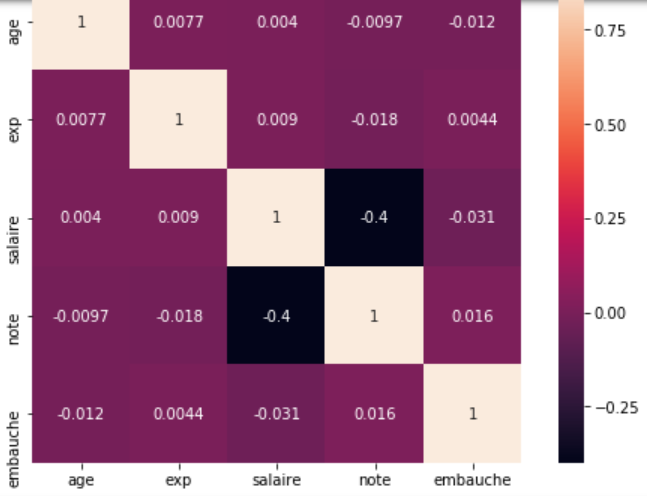


Je fais le choix de remplacer les valeurs manquantes par :

* La **moyenne pour les variables quantitatives**
* La **modalité la plus fréquente pour les variables qualitatives**

Afin de faire parler plus facilement la colonne date, j’effectue un split en 3 colonnes : annee, mois, jour.

Analyses des corrélations linéaires :



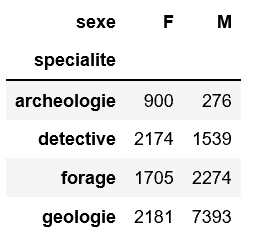
Cette matrice de corrélation montre que les variables quantitatives semblent indépendantes linéairement (attention, embauche a été détectée comme une variable quantitative mais elle est qualitative)

Au vue des analyses bivariées réalisées entre les variables et “embauche”, il semble que les **variables les plus influentes pour la prédiction sont : annee, diplome, specialite, cheveux** (racisme anti-roux??), **exp, note** (voir commentaires directement dans le code).

Comme les algorithmes de Machine Learning aiment les données normalisées, **j’effectue un StandardScaler() sur les données quantitatives**. De plus, les **variables qualitatives doivent être encodées**, ce que je réalise en les dummifiant à l’aide de get\_dummies().

1.2

On affiche la **table de contingence** entre les 2 variables.

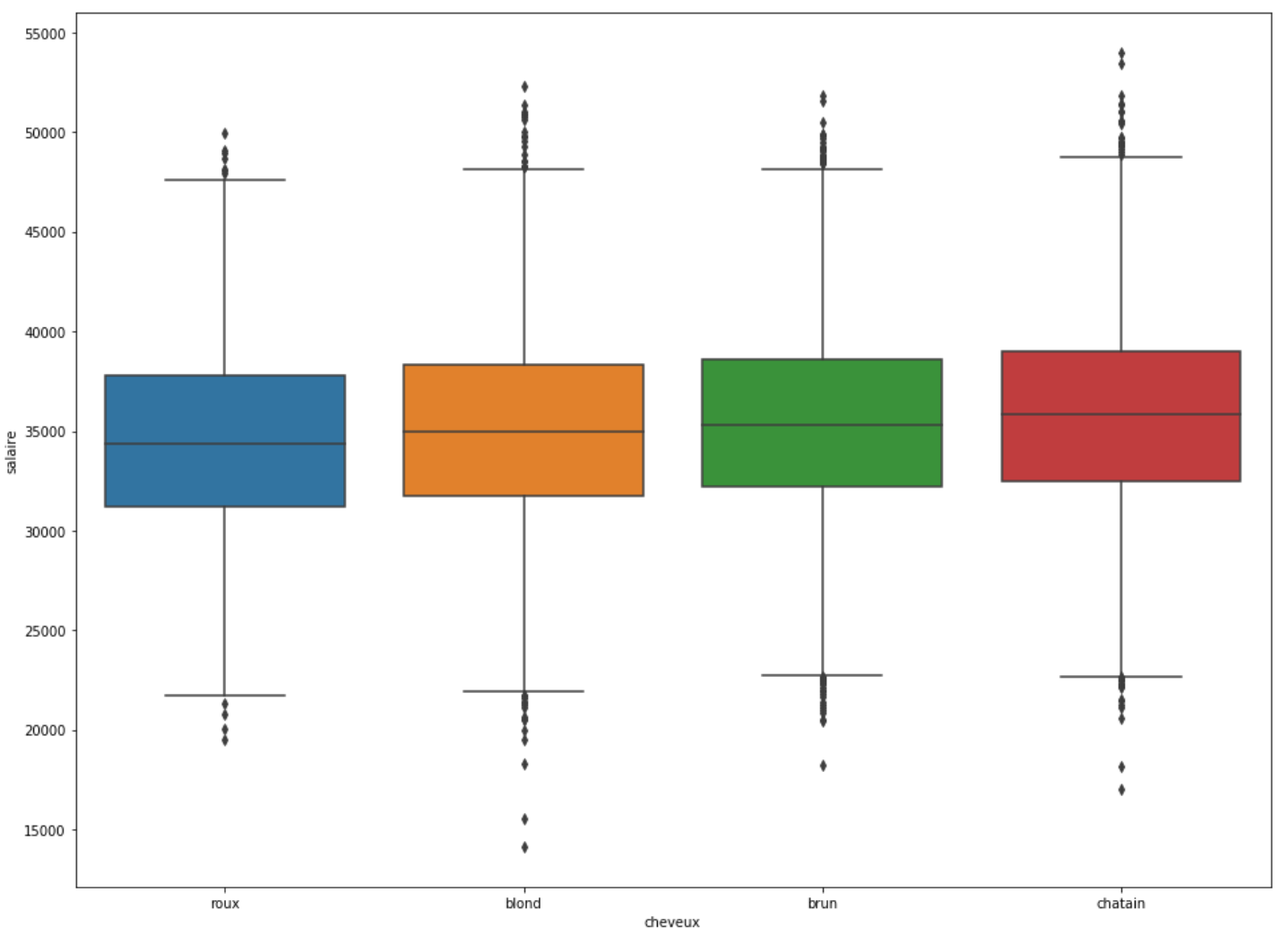


Comme on a affaire à 2 variables qualitatives, on effectue un **test statistique du chi2**.

On obtient une p\_value de 0.

Quelque soit l’erreur alpha que l’on choisit, l’hypothèse nulle est rejetée, **on ne peut pas conclure à l’indépendance des 2 variables**.

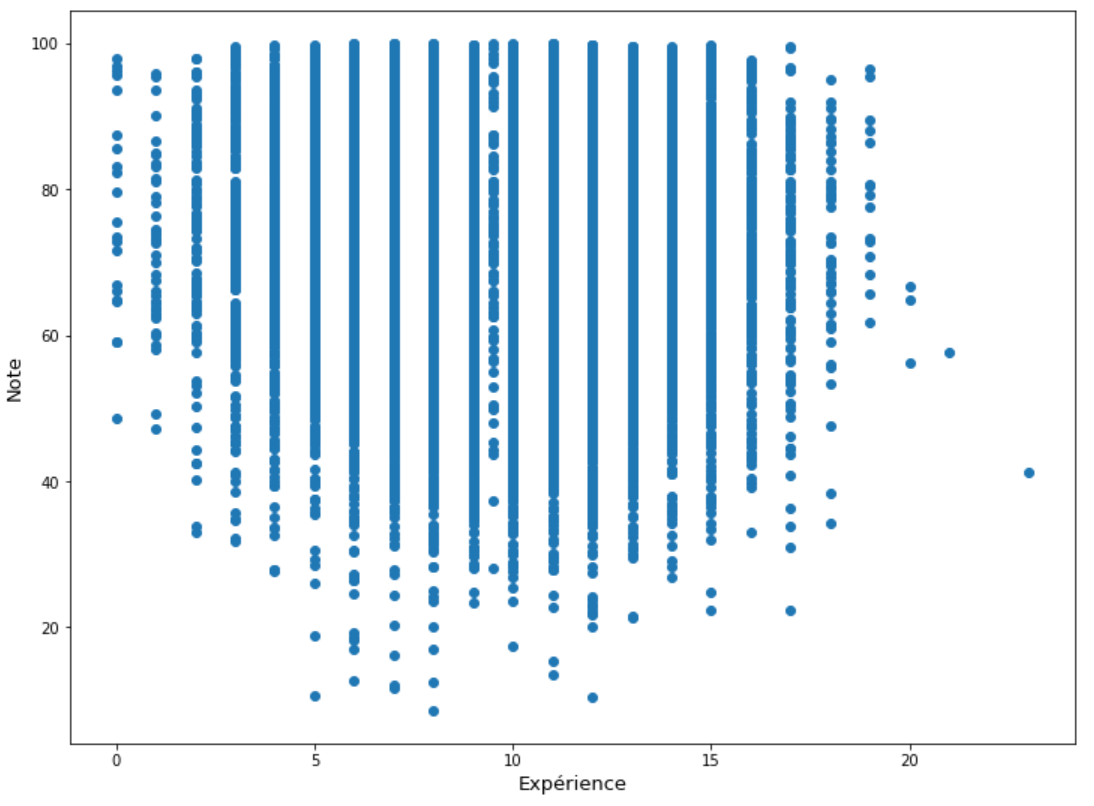
On applique une **méthode ANOVA**.



Les 2 variables ne semblent pas corrélées.

Le calcul du **rapport de corrélation η2Y/X=Vinterclasses/Vtotale donne 0.075**, ce qui confirme que les variables **ne sont pas corrélées**.

On affiche le **diagramme de dispersion** entre les 2 variables.



Il ne semble pas y avoir quelconque corrélation linéaire entre les variables. Vérifions-le par des tests statistiques.

Comme on est face à **2 variables quantitatives**, on effectue un **test de pearson** pour tester la corrélation linéaire.

Comme l’échantillon est grand, la loi des grands nombres nous permet d’être dans les hypothèses du test (normalité des distribution).

On obtient une p\_value de 0.016.

En prenant une erreur alpha = 1%, on a alpha = 0.01 < 0.016 = p, l’**hypothèse nulle est validée**, les 2 variables **ne sont pas corrélées linéairement**.

On fait un **test de spearman** pour voir si à défaut d’une corrélation linéaire, il n’y aurait pas une corrélation monotone entre les variables.

On obtient une p\_value de 0.015.

En prenant une erreur alpha = 1%, on a alpha = 0.01 < 0.016 = p, l’hypothèse nulle est validée, les 2 variables **ne sont pas corrélées de façon monotone**.

Cependant, même si des tests évidents ne peuvent pas le prouver, il semble y avoir un phénomène non linéaire (type quadratique), les personnes avec peu ou beaucoup d’expérience obtenant de meilleurs résultats que les autres. Ceci pourrait être vérifier en découpant l’âge en classe, ce qui ramènerait au même type de problème que la question précédente (une variable qualitative et une variable quantitative).

1. **Machine Learning**

2.1

Comme nous l’avons vu dans l’exploration des données, nous avons affaire à un problème de classification avec des classes assez déséquilibrées.

Afin de valider le modèle et choisir les hyperparamètres, j’effectue une **validation croisée en calculant le critère de performance pour différentes valeurs des hyperparamètres**.

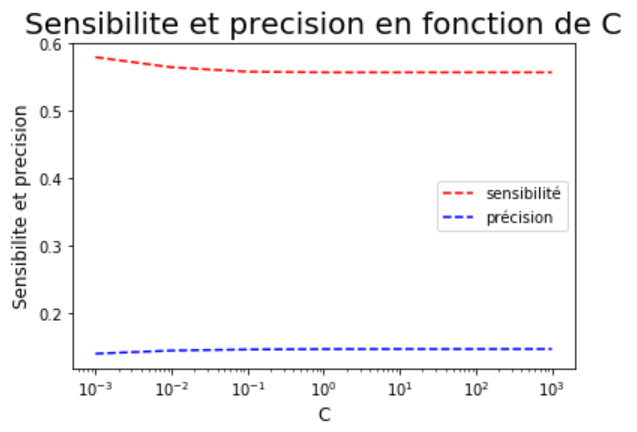
1. **Regression logistique**

Le premier test que j’effectue est sur une régression logistique classique. Même si les résultats ne sont pas bons, elle me servira de **baseline**.

Afin de prendre en compte le déséquilibre des classes, je teste le paramètre class\_weight à ‘balanced’.

J’optimise le paramètre C qui est le paramètre de régularisation de la régression.

Voici les résultats obtenus :

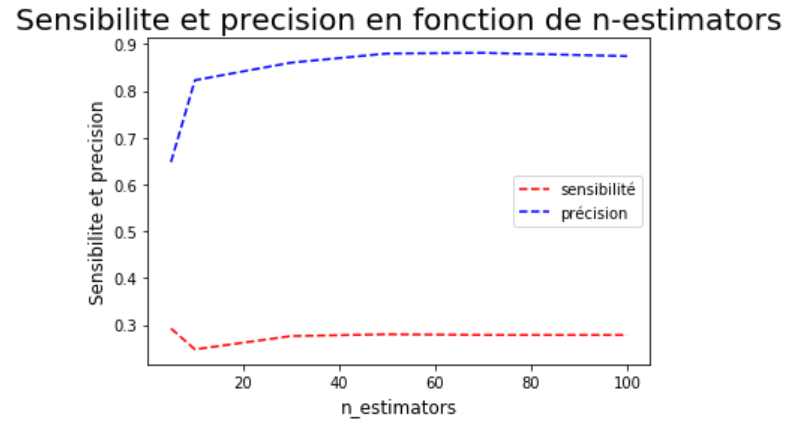


Les résultats ne sont pas bons : la sensibilité est certes assez bonne mais la précision elle est très mauvaise (on atteint presque la précision du cas hypothétique où tous les candidats sont prédits embauchés)

1. **Random Forest**

Le random forest est intéressant, le bagging permet de palier au problème d’overfitting. Enfin, il permet de donner une estimation de l’importance des features, ce qui nous permettra de répondre à la question suivante. Nous allons ici optimiser le nombre d’arbres de décisions qui sont créés (n\_estimators).

Voici les résultats obtenus :



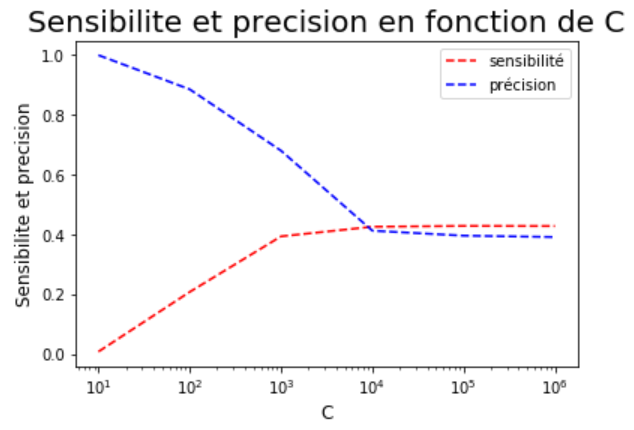
Les résultats sont légèrement meilleurs mais toujours mauvais.

1. **SVM à noyau**

Etant donné que lors de la question 2., nous avons constaté qu’il y avait des phénomènes quadratiques se produisant, j’ai l’idée d’utiliser un SVM à noyau polynomial.

J’optimise le paramètre C qui est le paramètre de régularisation de la régression dans l’espace de redescription.

Voici les résultats obtenus :



On obtient enfin des résultats plus convenables. Le meilleur hyperparamètre semble être C=1000.

La sensibilité est quasiment à sa valeur asymptotique et la précision est encore élevée.

2.2

Etant donné que la classe majoritaire à environ 90% est la classe 0 (non embauché), le risque est que le modèle prédise systématiquement des nouvelles valeurs dans la classe majoritaire, avec une accuracy de 90%, qui donnerait l’impression qu’on a des bons résultats alors qu’en réalité non.

Il est donc plus intéressant pour nous d’améliorer le taux de Faux Négatifs (FN) que le taux de Faux Positifs(FP) de la matrice de confusion.

Le critère de performance le plus utile semble donc la **Sensibilité = TP/(TP+FN)**. TP = True Positive

De plus, ce critère semble plus pertinent d’un point de vue métier. En effet, si on imagine que l’objectif du projet est de présélectionner les candidats “sérieux” pour le poste, il est toujours préférable de prendre trop de candidats potentiels (FP élevé), quitte à les éliminer par la suite, plutôt que d’éliminer des bons candidats (FN élevé).

Cependant en ne regardant que la sensibilité, on risque de choisir un modèle qui ne prédit que des embauches, et on ne sera pas plus avancé…

Il est donc important de jeter un oeil du côté des False Positives (FP) et de la **Précision = TP/(TP+FP)** pour vérifier que ça n’explose pas, ce qui implique une perte de temps pour l’entreprise à rencontrer de “mauvais candidats”.

Regardons maintenant les performances de notre modèle :

* **Sensibilité : 0.39**
* **Précision : 0.68**

Même si les résultats ne paraissent pas exceptionnels, en y réfléchissant de plus près, l’outil aide pas mal les RH : sans outil on avait 12% de chance que le candidat rencontré soit un bon candidat retenu, avec l’outil on passe à 40% ce qui n’est pas si mal! De plus, les résultats sont grandement améliorés par rapport à la baseline : la régression logistique.

2.3

Amélioration 1 : **Approfondir le feature engineering**

* **Mieux comprendre d’où viennent les données aberrantes**. Pour les notes > 100, est-ce que le barème était sur plus de 100? Dans ces cas-là, on peut ramener les notes aberrantes à 100.
* **Mieux exploiter le champs date** : y a-t-il des effets de saisonnalité (en regroupant les mois en saisons), une tendance d’embauche à la hausse. Selon le cas, éventuellement ajouter une feature qui explique mieux le phénomène (type, ‘saison’). On pourrait par exemple penser que le bon découpage est en trimestre ou semestre : c’est souvent à cette échelle que sont décidées/revues les stratégies de recrutement.

Vérifier avec le métier si les potentiels phénomènes que l’on a détecté doivent être captés par le modèle ou sont considérés comme du bruit (par ex, difficultés financières lié à un facteur extérieur en 2013 qui a impacté le recrutement).

Amélioration 2 : **Compenser les classes déséquilibrées**

* En Utilisant une méthode de ‘**cost-sensitive learning**’. Par exemple, utiliser le package **COSTCLA** qui permet d’entrainer des modèles ‘cost-sensitive’ de régression logistique et arbres de décision
* En jouant sur le dataset en faisant de l’**over-sampling** (augmenter le nombre d’éléments de la classe minoritaire dans le jeu d'entraînement) ou **under-sampling** (diminuer le nombre d’éléments de la classe majoritaire dans le jeu d'entraînement). La librairie **imblearn** propose un certain nombre de méthodes pour traiter ce problème

Amélioration 3 : Selon les besoins de l’entreprise, faire un **arbitrage entre critère de performance du modèle et temps de calcul**, en jouant sur les hyperparamètres et le choix de la méthode

Amélioration 4 : Si les performances des algorithmes testés ne sont pas bons, **c’est peut-être que les données expliquent mal le problème?** Il faudrait alors collecter de nouvelles données semblant mieux expliquer le phénomène étudié.