Projet ML

Parkinson Telemonitoring

Table of Contents

[1 Introduction 2](#_Toc189859468)

[2 Présentation du dataset 2](#_Toc189859469)

[3 EDA – Exploration des données 3](#_Toc189859470)

[3.1 Ensuite, il peut être intéressant de détecter des *outliers*, des valeurs aberrantes. La *Figure 3* permet de mettre en avant ces *outliers*. En revanche, tous ceux identifiés sur cette figure ne sont pas pertinent au regard des histogrammes en *Figure 2*. De ce fait, une seule valeur aberrante sera supprimée du dataset, il s’agit de l’ensemble des données associées à l’individu le plus jeune. D’un point de vue médical, ceci peut avoir du sens car cette maladie est généralement développée à un stade plus avancé de la vie. 4](#_Toc189859471)

[4 Modèle de prédiction 4](#_Toc189859472)

[4.1 Méthodes linéaires 4](#_Toc189859473)

[4.1.1 Régression linéaire / polynomiale 5](#_Toc189859474)

[4.1.2 Régression Lasso 5](#_Toc189859475)

[4.1.3 Régression Ridge 5](#_Toc189859476)

[4.2 Méthode non-paramétrique 5](#_Toc189859477)

[4.2.1 K-plus-proches-voisins (KNN) 5](#_Toc189859478)

[4.3 Support Vector Machines (SVM) 6](#_Toc189859479)

[4.4 Méthodes ensemblistes 6](#_Toc189859480)

[4.4.1 RandomForest 6](#_Toc189859481)

[4.4.2 Agrégation simple (moyenne arithmétique) 7](#_Toc189859482)

[4.4.3 Agrégation par stacking 7](#_Toc189859483)

[4.4.4 AdaBoost 7](#_Toc189859484)

[4.4.5 Gradient Boosting 7](#_Toc189859485)

[5 Meilleur modèle optimisé 7](#_Toc189859486)

[6 Conclusion 7](#_Toc189859487)

[Annexe 8](#_Toc189859488)

**Clément Peillon**

Github : [ML\_Parkinson\_Telemonitoring](https://github.com/DataPyPeil/ML_Parkinson_Telemonitoring)

SIGMA Clermont

Mastère Spécialisé Expert en Science des Données

# Introduction

La maladie de Parkinson touchait en 2019 plus de 8.5 millions de personnes dans le monde selon l’OMS. Cette maladie se manifeste progressivement chez les individus et peut représenter un réel handicap. Le suivi des symptômes est onéreux pour les hôpitaux, nécessite une mobilisation importante de personnel soignant spécialisé et exige la présence du patient pour réaliser des tests souvent chronophages. Pour faire face à ces problèmes, des chercheurs de l’université d’Oxford (Tsanas Athanasios, 2009) ont proposé une méthode de suivi non-invasive basée sur l’analyse de tests de langage et ont mis en ligne une partie de la base de données anonimisée qu’ils ont constituée. Dans ce rapport, on se propose d’étudier différents algorithmes de machine learning pour tenter de prédire l’évolution de la maladie chez les patients.

# Présentation du dataset

Accessible sur le site de [UC Irvine Machine Learning Repository](https://archive.ics.uci.edu/dataset/189/parkinsons+telemonitoring), ce dataset regroupe les données de 42 patients recrutés pour une campagne de test sur six mois afin de mesurer l’évolution de leur maladie de Parkinson. Ces tests présentent deux avantages majeurs. D’abord, ils sont non-invasifs donc le patient n’est pas affecté physiquement et sera moins enclin au stress. De plus, comme il s’agit d’enregistrement de la voix, ces tests peuvent être simplement réalisés à la maison sans avoir besoin de se déplacer dans un hôpital ou une clinique. Cela bénéficie à la fois au patient mais aussi au service hospitalier qui gagnent du temps pour traiter d’autres patients.

Les tests effectués environ tous les deux/trois jours sur la période ont permis de récolter 5875 données. Chaque test consiste en un enregistrement vocal duquel les chercheurs ont extrait 16 caractéristiques détaillées ci-dessous :

* **Jitter (%), Jitter (Abs), Jitter:RAP, Jitter:PPQ5, Jitter:DDP:** Mesures de la variabilité de la fréquence fondamentale (F0) de la voix, qui est souvent altérée chez les patients atteints de la maladie de Parkinson. Un *jitter* élevé indique une instabilité vocale.
* **Shimmer, Shimmer (dB), Shimmer:APQ3, Shimmer:APQ5, Shimmer:APQ11, Shimmer:DDA:** Mesures de la variabilité de l'amplitude de la voix. Un *shimmer* élevé est associé à une voix plus rauque ou instable, un symptôme courant chez les patients parkinsoniens.
* **NHR, HNR**: rapport harmonique-bruit. Mesure de la proportion d'énergie harmonique par rapport au bruit dans le signal vocal. Un HNR plus faible indique une voix plus bruyante et moins harmonieuse.
* **RPDE, DFA, PPE**: mesures non linéaires de la dynamique vocale.

En plus de ces caractéristiques extraites de la piste audio, ont été relevés **l’âge** et le **sexe** du patient ainsi qu’un indicateur de l’avancement dans l’étude **time\_test**.

Enfin, pour conclure avec la présentation du dataset, il convient de présenter quelles sont les variables à prédire. D’abord, **motor\_UPDRS** est une évaluation de la fonction motrice des patient atteints de Parkinson. L'échelle UPDRS (Unified Parkinson's Disease Rating Scale) est utilisée pour évaluer la progression de la maladie. La partie motrice de cette échelle évalue des aspects tels que la tremblote, la rigidité, la bradykinésie, etc… La variable **totale\_UPDRS** représentela somme des scores de toutes les sections de l'échelle UPDRS, comprenant les aspects moteurs, non moteurs et les complications liées au traitement. Ce score global reflète la sévérité globale de la maladie. On comprend alors qu’il sera sûrement intéressant de n’étudier qu’une des deux variables car celles-ci sont fortement. On vérifiera cette hypothèse en mesurant leur corrélation.

# EDA – Exploration des données

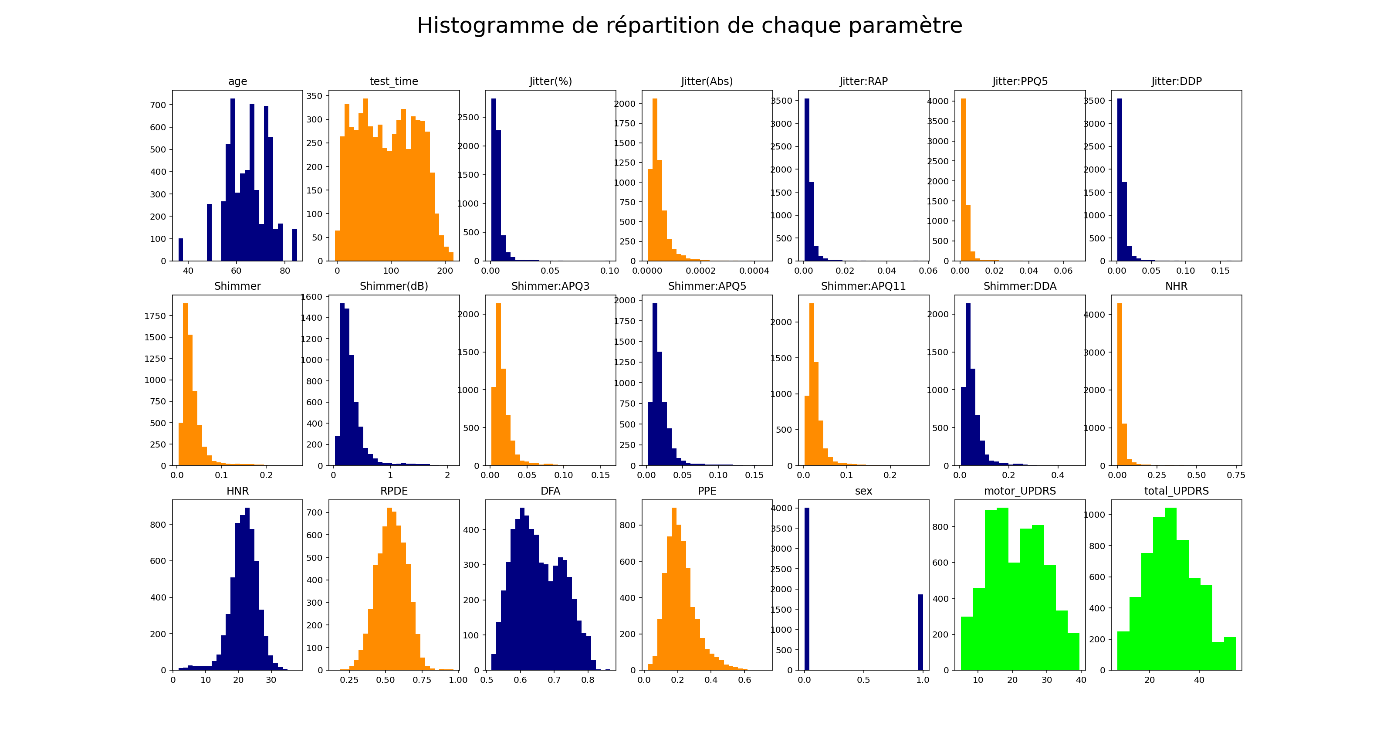


Figure 1: Distribution des variables

Tel que partagé par les chercheurs, le jeu de données nécessite quelques ajustements avant de pouvoir être utilisé pour l’ajustement d’un modèle de machine learning. En effet, les variables **Jitter** et **Shimmer** semblent être distribuées selon des lois exponentielles (*Figure 1*). Pour faciliter l’apprentissage, on ramènera ces variables à des distributions normales en utilisant la fonction logarithme népérien (*Figure 2*). Cette modification est nécessaire car certains modèles présupposent que les variables ou les résidus sont distribuées selon une loi gaussienne, c’est le cas pour la régression linéaire. Pour d’autres algorithmes, comme celui des k-plus proches voisins, avoir des variables normales permet d’utiliser la distance euclidienne qui a plus de sens dans un espace normalisé.

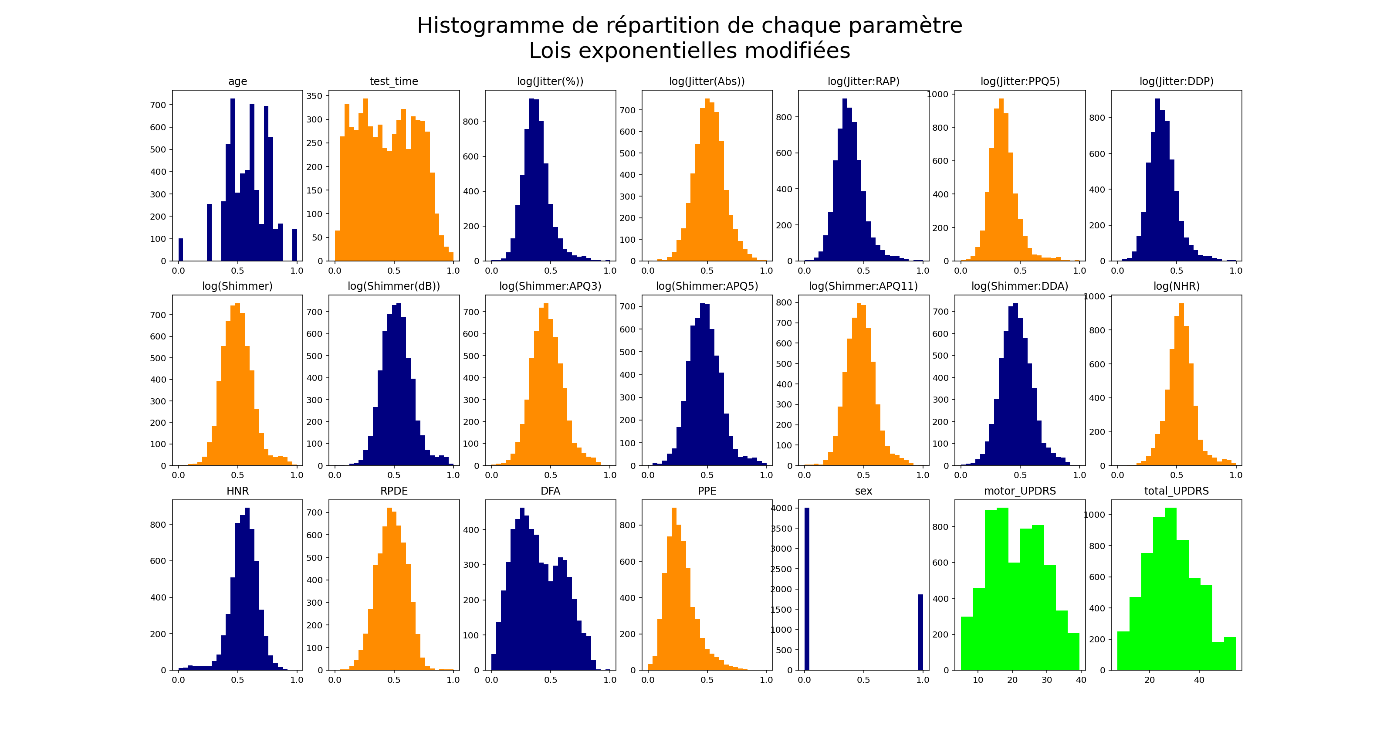


Figure 2: Distribution des variables après ajustement

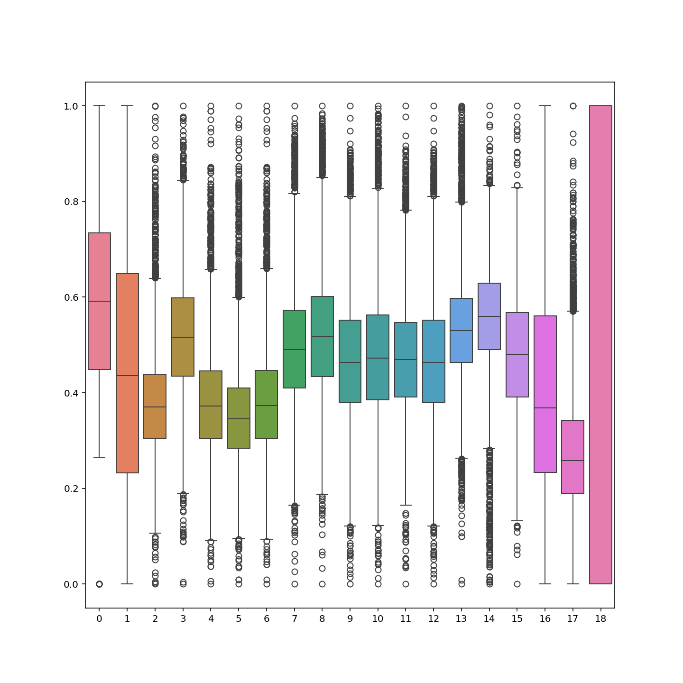


Figure 3: Boxplot pour détection des outliers

Ensuite, il peut être intéressant de détecter des *outliers*, des valeurs aberrantes. La *Figure 3* permet de mettre en avant ces *outliers*. En revanche, tous ceux identifiés sur cette figure ne sont pas pertinent au regard des histogrammes en *Figure 2*. De ce fait, une seule valeur aberrante sera supprimée du dataset, il s’agit de l’ensemble des données associées à l’individu le plus jeune. D’un point de vue médical, ceci peut avoir du sens car cette maladie est généralement développée à un stade plus avancé de la vie.

Enfin, pour conclure l’exploration du dataset, on peut s’intéresser à la corrélation entre les variables explicatives et celles à prédire (*Figure 4*). Cette figure nous permet de confirmer l’hypothèse sur la forte corrélation entre les deux variables à prédire. Ainsi, on fait le choix, arbitraire, de ne conserver que la variable **total\_UPDRS**. Prédire les deux n’aurait que peu d’intérêt vis-à-vis du surcoût computationnel que ça représenterait.

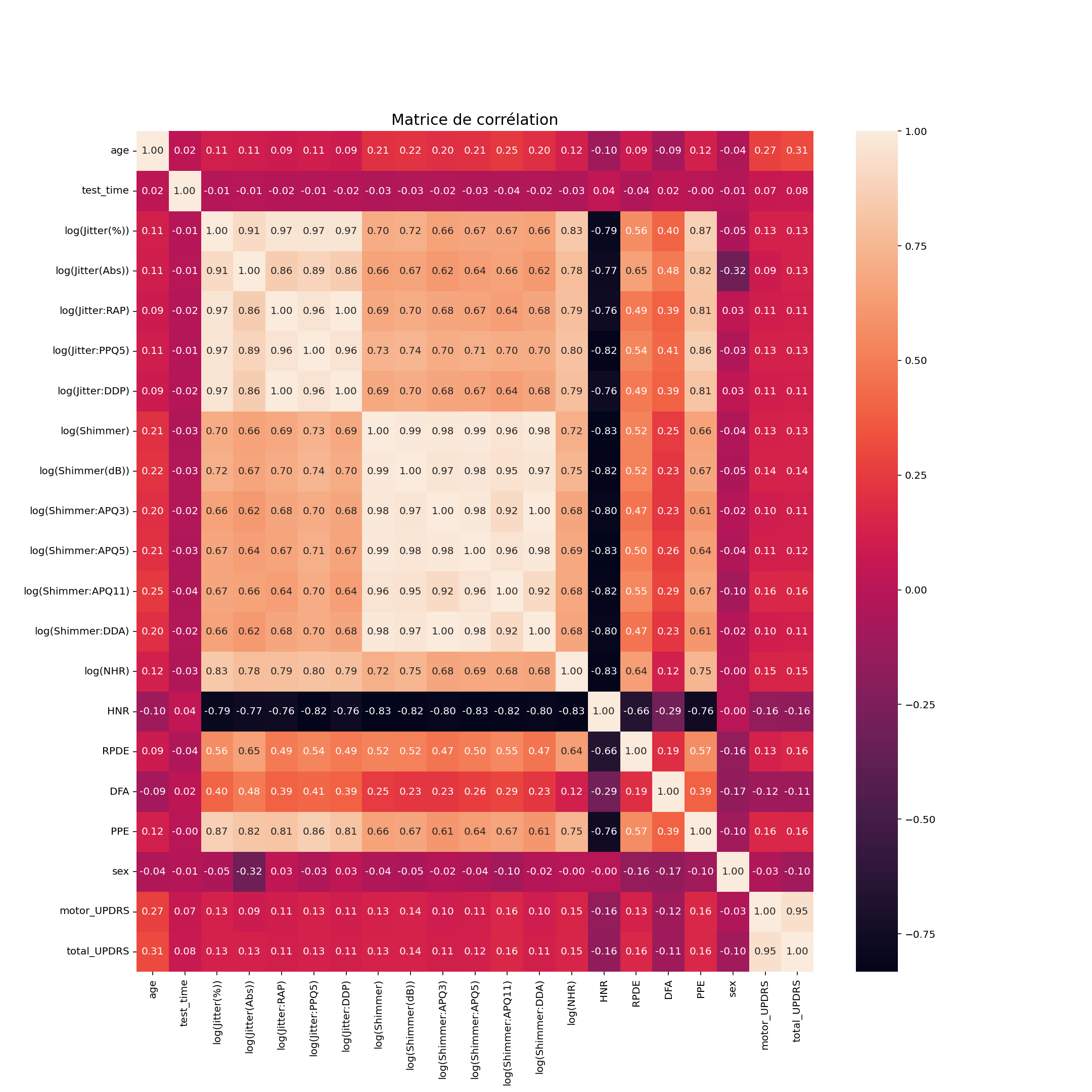


Figure 4: Corrélation des variables explicatives avec les variables à expliquer

# Modèle de prédiction

Afin de choisir le meilleur modèle pour cette tâche de régression, on va entrainer plusieurs modèles sur un jeu de données d’entrainement et les tester sur un autre jeu de données inconnues du point de vue du modèle.

## Méthodes linéaires

Dans cette partie, on cherche à trouver une fonction polynomiale en les variables explicatives pour prédire **total\_UPDRS**. La classe de fonctions recherchées couvre ici les polynômes de degré 1, 3 et 5.

### Régression linéaire / polynomiale

Dans cette partie, on cherche une relation purement linéaire. A chaque variable est associé un poids correspondant à l’influence de la variable sur la prédiction.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure 5: Actual vs Predicted - LinearRegression

Comme on pouvait le prédire en consultant la matrice de corrélation, les résultats observés ne sont pas probants pour le polynôme de degré 1. Plus surprenant, on voit que le modèle prédit parfaitement les résultats en test laissant penser que cette relation est la meilleure. En revanche, on peut émettre certaines réserves quand à la prédiction du résultat avec un tel polynôme pour des patients qui sortiraient légèrement du spectre des patients étudiés.

### Régression Lasso

En intégrant une pénalité à la fonction coût avec la norme L1 sur les coefficients du modèle, on entraine un modèle de régression LASSO. En théorie, ce type de modèle permet de faire de la sélection de variables en réduisant à zéro celles qui n’ont pas de lien avec la prédiction. En revanche, dans le cas où les variables d’entrées sont corrélées entres elles comme dans notre cas (Annexe 1), ce modèle éprouve des difficultés. Les résultats, non satisfaisant, sont présentés en annexe.

### Régression Ridge

La régression Ridge quant à elle impose une pénalité avec la norme L2 des poids du modèle. Cette méthode s’est révélée être moins efficace que la régression linéaire non-pénalisée avec les différentes pénalités testées.

## Méthode non-paramétrique

### K-plus-proches-voisins (KNN)

La méthode des k-plus-proches-voisins fait la moyenne des k-voisins les plus proches de la donnée étudiée pour faire la prédiction. Cette méthode testée avec plusieurs valeurs pour k n’est pas concluante.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Figure 6: Actual vs Predicted - KNN

## Support Vector Machines (SVM)

Le `SVR` (Support Vector Regressor) est un modèle de régression basé sur les SVM, utilisé pour prédire des valeurs continues. Il fonctionne en trouvant un hyperplan qui maximise la marge entre les points de données et la ligne de régression, tout en minimisant l'erreur de prédiction. Le SVR est efficace pour gérer les données non linéaires grâce à l'utilisation de fonctions noyaux, comme le noyau RBF, qui transforment les données dans un espace de plus haute dimension où une séparation linéaire est possible.

|  |  |
| --- | --- |
|  |  |

Figure 7: Actual vs Predicted – SVM

Comme présenté sur la figure 7, les SVM ne sont pas de bons modèles dans ce cas d’utilisation. L’erreur en généralisation n’est pas acceptable.

## Méthodes ensemblistes

### RandomForest

Le RandomForest combine plusieurs arbres de décision et du bagging pour entrainer chaque arbre sur un sous-ensemble aléatoire des données et limiter le sur-apprentissage. En fonction du nombre *d’estimator*, on peut capturer des relations plus ou moins complexes. On peut vérifier ce comportement en analysant l’erreur (*Figure 8*), Elle diminue à mesure que le nombre *d’estimator* augmente. De plus, la performance du modèle est idéale et l’erreur relativement faible pour 50 et 100 *estimator.*

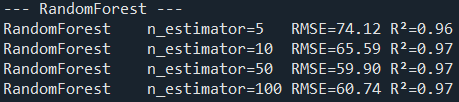


Figure 8: Score de prédiction - RandomForest

|  |  |
| --- | --- |
|  |  |

Figure 9: Actual vs Predicted – RandomForest

### Agrégation simple (moyenne arithmétique)

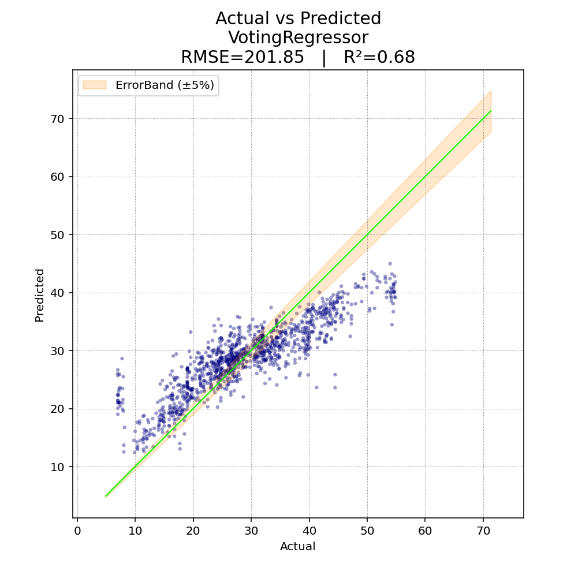


Figure 10: Actual vs Predicted

VotingRegressor

La méthode d’agrégation simple *VotingRegressor* est un modèle ensembliste. Il combine les prédictions de plusieurs modèles de régression pour améliorer la précision globale. Il agrège les prédictions en calculant leur moyenne, ce qui permet de réduire la variance et d'obtenir des prédictions plus robustes.

En choisissant les modèles linéaires, SVM, KNN et RandomForest précédant, on obtient un modèle moyen en prédiction. Ce score doit pouvoir être amélioré avec d’autres modèles de régression. En revanche, l’inconvénient de ce modèle est son temps d’entrainement car il additionne le temps nécessaire à chaque modèle pour s’entrainer. Pour une performance comme telle, ce modèle ne sera pas retenu pour la suite.

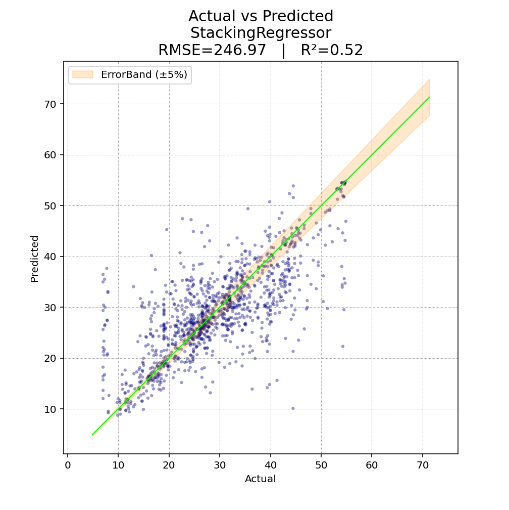


Figure 11: Actual vs Predicted

StackingRegressor

### Agrégation par stacking

L’agrégation par stacking repose sur les prédictions de plusieurs modèles pour alimenter un méta-modèle qui combine ces prédictions pour produire une estimation finale. En général, cette estimation est plus précise que chaque modèle individuellement.

Dans notre cas, ce n’est pas le cas mais ce modèle pourrait être améliorer en utilisant à nouveau d’autres modèles pour faire les premières prédictions.

### AdaBoost

AdaBoost est un algorithme de boosting qui entraîne séquentiellement des modèles faibles, en ajustant le poids des instances mal classifiées pour améliorer les performances globales.

Ce modèle est très performant avec très peu *d’estimator*. Plus ce nombre augmente, plus la précision augmente aussi mais la complexité augmente et le temps pour entrainer le modèle aussi.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure 12: Actual vs Predicted - AdaBoost

### GradientBoosting

Le GradientBoosting construit des modèles de manière itérative en ajustant chaque nouveau modèle aux erreurs résiduelles du modèle précédent, optimisant ainsi la précision des prédictions. Plus le nombre de modèles est élevé, plus la complexité augmente mais plus la précision augmente aussi.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure 13: Actual vs Predicted - GradientBoosting

# Meilleur modèle optimisé

# Conclusion

# Annexe

Annexe 1: Matrice de corrélation des variables