Rapport du Projet Fil Rouge - AFZ

Analyse de données pour la prédiction des valeurs nutritionnelles de fourrages pour ruminants par apprentissage automatique et Grand Modèle de Langue (LLM)

Aristide Lauront, Raphaël Genin, Raphaël Rubrice, Matéo Petitet

AgroParisTech

université PARIS-SACLAY

En collaboration avec

ASSOCIATION FRANÇAISE DE ZOOTECHNIE
French association for animal production

Enseignants encadrants : Antoine Cornuejols, Christine Martin

Commanditaires : Gilles Tran, Valérie Heuzé

Table des matières

1 Introduction **2**

2 Matériels et méthodes **2**
2.1 Matériels et outils utilisés . . . 2
2.2 Description des données . . . 2
2.3 Approches d'apprentissage statistique non neuronales . . . 4
2.3.1 Vue globale des modèles . . . 4
2.3.2 Réduction de dimension . . . 5
2.3.3 Gestion des valeurs manquantes . . . 5
2.4 Approches d'apprentissage statistique neuronales . . . 5
2.4.1 Utilisation de Grands Modèles de Langue (Large Language Models) . . . 5
2.4.2 Choix du modèle de langue . . . 6
2.4.3 Raffinage (ou Fine-Tuning) des modèles choisis . . . 6
2.4.4 Entraînement et sélection des modèles d'apprentissage profonds . . . 7

3 Résultats **8**
3.1 Campagne d'expérimentation ML . . . 8
3.1.1 Remplacement des valeurs numériques manquantes et normalisation . . . 8
3.1.2 Détermination des meilleurs pré-traitements . . . 9
3.1.3 Résultats par des méthodes ensemblistes de boosting . . . 9
3.1.4 Effet des informations numériques et textuelles sur la qualité des prédictions . . . 10
3.2 Modèles neuronaux . . . 11
3.2.1 Qualité de prédiction d'un régresseur faible . . . 11
3.2.2 Effet de la tête de régression optimiséd et du raffinement . . . 11
3.2.3 Effet des informations textuelles . . . 13
3.3 Comparaison entre le meilleur modèle ML et les modèles DL . . . 13

4 Discussion **14**
4.1 Influence de l'optimisation de la tête de régression . . . 14
4.2 Choix de la fonction coût pour l'entraînement des réseaux de neurones . . . 15
4.3 Limites des méthodes d'apprentissages statistiques non neuronales . . . 15
4.4 Effet des libellés dans les modèles neuronaux . . . 16
4.5 Effet du raffinage sur les performances . . . 17
4.6 Choix du modèle de langage masqué . . . 17
4.7 Perspectives d'application des modèles proposés . . . 17

5 Conclusion **17**

Références **20**

1 Introduction

En production animale, la formulation des rations constitue un véritable défi. Celle-ci doit en effet répondre à plusieurs attendus que ce soit en terme de performance économique, environnementale ou aux attendus sociaux. Pour ce faire, l'INRAE développe depuis plus de 40 ans (Inra 1978 ; Jarrige 1988 ; Agabriel et (France) 2007) un système permettant de caractériser la valeur nutritionnelle d'aliments ainsi que la réponse des animaux aux apports nutritionnels. La ration est formulée par la recherche d'un optimum de la fonction de production.

Ce système est fondé sur des données expérimentales. Il établit des équations permettant de calculer des valeurs intermédiaires, comme les valeurs de digestibilité de la matière organique, évitant de reproduire ces expériences coûteuses. Néanmoins, en dehors des valeurs de référence présentes dans les tables, il reste complexe de calculer de nouvelles valeurs nutritionnelles car cela demande une connaissance experte des équations et des choix pour chaque typologie de fourrage. En somme, il n'existe pas de solution globale, de cadre commun pour la caractérisation d'un nouveau fourrage.

Nous nous sommes intéressés à la prédiction de ces valeurs, sans utiliser les équations mais en s'appuyant sur les valeurs de références présentes dans les tables. À l'aide de quelques données simples à recueillir sur les fourrages (MS, MM, MAT, CB, NDF, ADF, EE) et d'une description de notre fourrage nous avons cherché à prédire leurs unités fourragères énergétiques (UFL, UFV) ainsi que protéiques (BPR, PDI, PDIA). Nous avons voulu savoir si des modèles d'apprentissage automatique permettaient de retrouver les valeurs des tables en simplifiant le processus.

Ces travaux facilitent la qualification des fourrages et la gestion des prairies pour les éleveurs car ils peuvent être combinés avec des mesures non-invasives de valeurs chimiques sur les fourrages comme proposé par Bastianelli et al. 2019.

Nous avons tout d'abord abordé le problème sous l'angle de la régression par des modèles linéaires, non linéaires et ensemblistes. Puis, nous avons ensuite cherché à mieux intégrer les données textuelles de description des fourrages à l'aide de différents prétraitements mais aussi grâce aux grands modèles de langue.

Ces deux approches offrent des performances satisfaisantes avec une part de variance expliquée de l'ordre de 90%. Néanmoins, la prise en compte des valeurs textuelles apporte peu d'informations. On note également une différence de performance dans la prédiction des unités énergétiques par rapport aux unités protéiques.

2 Matériels et méthodes

2.1 Matériels et outils utilisés

Ce projet est réalisé à l'aide du langage de programmation Python en raison de son expressivité et de la multitude de librairies de référence disponibles dans le domaine de l'apprentissage statistique. En particulier, les librairies `scikit-learn` et `pytorch` ont été utilisé pour le développement des modèles d'Apprentissage Automatique respectivement non neuronaux et neuronaux. Afin de collaborer, nous avons utilisé le logiciel de forge `git` et hébergé le projet sur `Github`. Le code est disponible à l'adresse suivante : https ://github.com/lauronta/projet_fil_rouge_afz. Enfin, pour l'entraînement des réseaux de neurones, nous avons personnellement acheté des crédits de calcul sur la plateforme `Google Colab` afin d'avoir accès à des GPUs ce qui a permis de faire un ensemble d'expériences, autrement impossible avec des CPUs compte tenu de la limite de temps du projet.

2.2 Description des données

Les données de référence ont été fournies par l'AFZ, sous forme de tableaux Excel. Ces deux tableaux, un pour les fourrages et un pour les concentrés, correspondent aux tables de valeurs telles que publiées par l'INRAE (INRA 2018). Nous n'utiliserons finalement que la table des fourrages, celle-ci étant plus fournie en données et donc plus adapté à la tâche de prédiction formulée (973 lignes contre 172).
Chaque ligne du tableau correspond à un aliment. Chaque aliment est ainsi caractérisé par :

- un numéro de ligne
- son identifiant INRAE unique
- 5 libellés de plus en plus précis (du libellé 0, systématique, au libellé 4, facultatif) ; ces libellés décrivent qualitativement le fourrage
- 92 valeurs chimiques mesurées en laboratoire

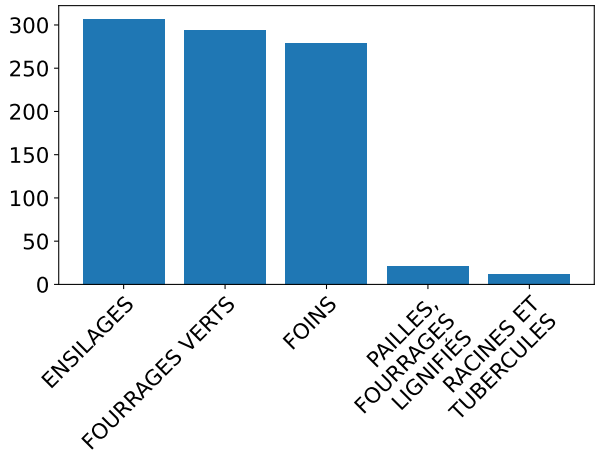


Figure 1 – Répartition des aliments en fonction du type de fourrage

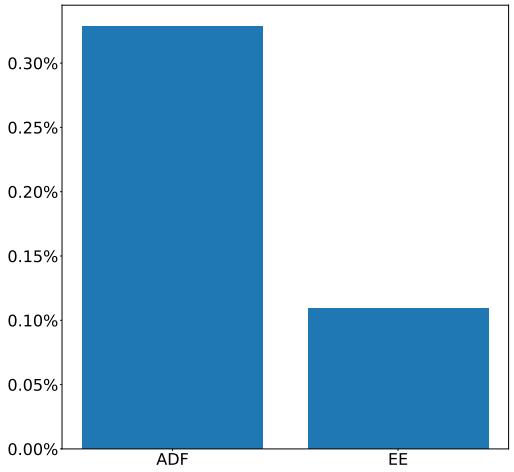


Figure 2 – Proportion de valeurs manquantes dans les caractéristiques intéressantes
Les caractéristiques sans valeurs manquantes ne sont pas affichées

Le tableau de fourrage décrit donc 973 aliments, répartis de manière assez inégale. La Figure 1 détaille la répartition des aliments décrits dans ce tableau. Nous pouvons constater que deux catégories peuvent être distinguées : les ensilages, fourrages verts et foins sont très bien représentés (plus de 250 aliments), tandis que les pailles et fourrages lignifiés et les racines et tubercules sont sous-représentés (moins de 25 aliments). La qualité attendue des prédictions pour ces typologies d'aliments seront donc probablement moins élevées sur ces aliments.
Toutes les valeurs chimiques présentes dans le tableau sont complexes et coûteuses à mesurer, elles ne peuvent donc pas l'être systématiquement pour chaque nouvel arrivage de fourrages. Nous cherchons à mesurer cinq de ces valeurs intéressantes en utilisant la description textuelle associée seulement à 7 valeurs chimiques mesurables aisément et à faible coût par infrarouge.
Les valeurs recherchées (cibles) sont des valeurs standards en zootechnie bovine. Ces cinq caractéristiques peuvent être subdivisées en deux catégories :

— les valeurs énergétiques, incluant :
  - l'**Unité Fourragère Lait** (UFL), quantité d'énergie nette absorbable pendant la lactation ou l'entretien du ruminant (1 UFL = 1700 kcal)
  - l'**Unité Fourragère Viande** (UFV), quantité d'énergie nette absorbable lors de l'engraissement d'un ruminant (1 UFV = 1820 kcal)

— les valeurs protéiques, incluant :
  - les **Protéines Digestibles dans l'Intestin** (PDI), valeurs nutritives en azote (protéines métabolisables) chez les ruminants
  - les **PDI d'origine Alimentaire** (PDIA), non dégradées dans le rumen
  - le **Bilan Protéique du Rumen** (BPR), différence entre les protéines ingérées et celles passant au duodénum

Les valeurs utilisées (source), quant à elles, sont sept caractéristiques pouvant être facilement mesurées par infrarouge à l'aide d'outils à faible coût. Il s'agit de :

- la **Matière Sèche** (MS), le restant après retrait de toute l'eau du produit
- la **Matière Minérale** (MM), portion non-organique du produit
- la **Matière Azotée Totale** (MAT), protéines brutes
- la **Cellulose Brute** (CB)
- les **"Neutral Detergent Fiber"** (NDF), fibres totales présentes dans l'aliment
- les **"Acid Detergent Fibers"** (ADF), fibres totales présentes dans l'aliment sauf hémicellulose
- le **"Ether Extract"** (EE), lipides brutes

Ces valeurs cibles et sources sont très bien décrites par le tableau de données. Parmi ces caractéristiques intéressantes, une très faible partie est manquante. Comme nous pouvons le constater dans la Figure 2,

seuls l'ADF et le EE présentent quelques valeurs manquantes, sans jamais dépasser 0,3% du total. Les données sont donc de bonne qualité.

Les libellés sont subdivisés selon 5 niveaux de précision, allant de L0 à L4. Chaque niveau apporte des précisions différentes, et tous ne sont pas systématiquement présents. Chaque niveau présente un nombre de modalités différent ; dans le détail, nous avons :

- **Niveau L0** → catégorie générale de l'aliment considéré (ex : FOURRAGES VERT) ; 5 modalités
- **Niveau L1** → sous-catégorie de l'aliment (ex : PRAIRIES PERMANENTES, PLAINE (NORMANDIE)) ; 53 modalités
- **Niveau L2** → précisions sur les conditions de culture et/ou récolte de l'aliment (ex : 1er cycle) ; 42 modalités, présence non-systématique
- **Niveau L3** → précisions supplémentaires sur les conditions de culture et/ou récolte de l'aliment (ex : 15-25 avril, déprimage, ST = 172°C) ; 113 modalités, présence non-systématique, informations parfois de même nature que le niveau 2
- **Niveau L4** → informations complémentaires sur l'aliment (ex : Épiaison du dactyle) ; 55 modalités, présence non-systématique

Les libellés sont standards du niveau 0 à 2, et deviennent ensuite plus imprécis et inconsistant.
Ces données textuelles sont importantes mais pas suffisantes pour raffiner un grand modèle de langue. Pour cette tâche, nous avons utilisé la base de données Feedipedia (© INRAE CIRAD AFZ FAO) intégrale en français et anglais, fournissant un corpus vocabulaire relatif à l'alimentation animale très détaillé. Feedipedia est la base de donnée en ligne de référence pour tout ce qui concerne l'alimentation animale. Ce corpus se présente sous forme d'un fichier excel contenant des noms et descriptions d'aliments en français et en anglais. Au total, 16 186 lignes de données y sont présentées.

2.3 Approches d'apprentissage statistique non neuronales

2.3.1 Vue globale des modèles

Dans un premier temps, nous nous sommes intéressés à des modèles d'apprentissage automatique non-neuronaux. Nous avons cherché à optimiser différents régresseurs sur nos données, de trois types différents :

— modèles linéaires
  - une simple **régression linéaire**
  - une régression de **Tikhonov**, ou régression arête, ou ridge (Tikhonov 1943)
  - une régression **Lasso** (Tibshirani 1996)
  - une régression **ElasticNet** (Zou et Hastie 2005)
  - une régression par **Séparateur à Vaste Marge linéaire** (Drucker et al. 1996)

— modèles non linéaires
  - une régression par **Séparateur à Vaste Marge avec noyau RBF** (Chang et al. 2010)
  - une régression utilisant la méthode des **k plus proches voisins** (Cover et Hart 1967)

— arbres de décision
  - une régression par **arbre de décision** (Breiman et al. 1984)
  - une régression par **forêts d'arbres décisionnels** (Breiman 2001)
  - une régression par **gradient boosting** (Friedman 2002)
  - une régression par **XGBoost** (T. Chen et Guestrin 2016)

La plupart de ces méthodes ne supportant pas la régression à plusieurs sorties, nous avons entraîné un régresseur par variable cible.

Les variables quantitatives ont été directement fournies en entrée dans les modèles, après éventuelle imputation des valeurs manquantes. Les variables textuelles (ordinales) ont quant à elles été vectorisées en utilisant un encodage disjonctif complet, dénommé encodage "one-hot" par la suite. Au vu des nombreuses modalités pour chaque libellé, cet encodage génère un nombre important de dimensions, ce qui a amené à explorer différentes stratégies de réduction de dimension telles que l'Analyse des Correspondances Multiples (ACM) ou l'Analyse Factorielle de Données Mixtes (AFDM), présentées par la suite.

La validation des modèles a été faite par validation croisée sur 5 plis, les variables d'entrée et de sortie ayant été normalisées pour obtenir les meilleures performances. Nous avons exploré différentes stratégies de pré-traitement comme énoncées précédemment, et utilisé un ou plusieurs libellé.

2.3.2 Réduction de dimension

Comme précisé précédemment, la dimensionnalité augmente très fortement avec l'encodage one-hot effectué sur les données catégorielle, passant de 12 à 44 dimensions en prenant seulement en compte les libellés de niveau 1 et 2. Il est donc apparu essentiel d'explorer la réduction de dimension.

Nous avons tout d'abord utilisé l'**Analyse des Correspondances Multiples** (ACM, Escofier et Jérôme Pagès 2008) afin de réduire la dimension des données qualitatives. Il s'agit d'une méthode d'analyse factorielle dédiée aux données qualitatives, ou catégorielles. Elle ne permet pas de considérer des variables quantitatives, à moins de les traiter comme des catégories.

Afin de considérer à la fois les variables qualitatives et quantitatives dans la réduction de dimension, nous nous sommes également tournés vers l'**Analyse Factorielle de Données Mixtes** (AFDM, J. Pagès 2004). Celle-ci permet l'analyse simultanée d'éléments mixtes, soit des éléments numériques et catégoriels. En ce sens, elle combine une Analyse en Composantes Principales (ACP) pour les premières et une ACM pour les secondes.

2.3.3 Gestion des valeurs manquantes

En plus de la réduction de dimension, nous avons traiter la question de la gestion des valeurs manquantes. Celles-ci sont très peu nombreuses, nous avons donc choisis de les imputer, l'imputation consistant en le remplacement des données manquantes par des valeurs substituées. Nous avons exploré plusieurs stratégies d'imputations présentes dans scikit-learn :

- une imputation **par la moyenne**, remplaçant les valeurs manquantes par la moyenne globale des valeurs présentes
- une imputation **par la médiane**, remplaçant les valeurs manquantes par la médiane des valeurs présentes
- une imputation **par la plus fréquente**, remplaçant les valeurs manquantes par la valeur la plus fréquente des valeurs présentes
- une imputation utilisant la méthode des **k plus proches voisins** (KNN Imputer)
- une imputation **itérative** (Iterative Imputer)
  , remplacement les valeurs manquantes d'après les corrélations statistiques entre les différentes variables

Les imputations par la moyenne, la médiane ou la plus fréquente sont des imputations simples. L'imputation par les k plus proches voisins utilise la méthode éponyme (Cover et Hart 1967)afin de remplacer la valeur manquante. Cette méthode est pertinente car dans notre contexte, nous pouvons supposer que les valeurs manquantes seront proches de celles d'aliments de la même catégorie, qui seront ici ses voisins utilisés pour la prédiction. L'imputation itérative est implémentée dans scikit-learn (*IterativeImputer* 2024) en utilisant un algorithme de type Round-robin pour réaliser la prédiction des valeurs.

2.4 Approches d'apprentissage statistique neuronales

2.4.1 Utilisation de Grands Modèles de Langue (Large Language Models)

Comme indiqué en introduction, les commanditaires avait une demande particulière : utiliser des Large Language Models dans le projet. Cette demande provient de l'espoir de pouvoir traiter des données sous forme de langage naturel fournis par l'utilisateur en utilisant l'état de l'art de ce domaine. Ces modèles sont constitués de deux parties : un tokenizer et un réseau de neurones basé sur l'architecture Transformer (Vaswani et al. 2023).

Le rôle du tokenizer est de transformer l'entrée textuelle en séquence de token ou jeton. Les jetons reconnus par un tokenizer dépendent de la façon dont il a été entraîné (essentiellement le corpus d'entrainement) ainsi que de l'algorithme utilisé. Selon ce dernier les tokens peuvent correspondre à des mots, des sous-mots voire des caractères. Une fois les tokens obtenus, chacun est associé à une représentation vectorielle appelée **embedding**. Lorsqu'une nouvelle entrée textuelle est saisie, le tokenizer transforme la séquence fournie en séquence de tokens, chacun d'entre eux étant identifié par leur index. Chaque index donne accès à l'embedding du token correspondant ; c'est cette séquence d'embedding qui est ensuite passée au réseau de neurones.

Le rôle de ce dernier est alors d'apprendre les liens complexes du langage comme la sémantique et la structure grammaticale à travers diverses tâches comme la prédiction de mots masqués ou la prédiction de la prochaine phrase.

De nombreux grands modèles de langues (LLM) sont accessibles de nos jours. Leur variété vient de la nature des documents utilisés pour la phase de pré-entraînement, de leurs spécificités architecturales et de leur raffinement (ou fine-tuning) sur un certains nombre de tâches. Ils se sont très rapidement répandus à travers les applications conversationnelles mais montrent depuis les deux dernières années leur capacité dans bien d'autres domaines comme synthèse de documents, la rédaction de code, etc. Une des véritables forces de ces modèles pour notre projet correspond à leur capacité à pouvoir représenter l'entrée textuelle en un vecteur. En effet, l'essentiel des LLMs ont été entraînés avec un token particulier appelé le token `[CLS]`. Ce token, ou équivalent, est toujours ajouté en début de séquence durant la phase d'entraînement. En raison du mécanisme de self-attention couplée à des masques "causaux" (ils n'ont rien de causaux, simplement chaque représentation de token n'évolue qu'en fonction des tokens suivant) utilisés dans les architectures de type Transformer, le jeton `[CLS]` évolue en fonction de l'ensemble de la séquence et est donc considéré comme la représentation vectorielle de l'ensemble de l'entrée textuelle. Ainsi, en se basant sur cette idée, nous avons proposé une approche consistant en l'extraction de l'embedding représentant la description textuelle du fourrage et le couplage de cet embedding aux 7 valeurs chimiques, puis l'utilisation d'une tête de régression afin de prédire les 5 valeurs d'intérêt (Figure 3). L'apprentissage se fera en optimisant une fonction de coût "SmoothL1" qui correspond à la définition suivante :

$$\text{SmoothL1}(x, y) = \begin{cases} \frac{1}{2}(x-y)^2, & \text{si } |x-y| < 1 \\ |x-y| - \frac{1}{2}, & \text{sinon} \end{cases} \tag{1}$$

2.4.2 Choix du modèle de langue

Le choix du LLM utilisé est en général guidé par 2 facteurs : sa taille sur disque et ses performances. N'ayant pas accès à des infrastructures de calculs puissantes, nous nous sommes concentrés sur la taille des modèles. En ce qui concerne les modèles de langue de petites tailles (de l'ordre de la centaine de millions de paramètres), il existe une architecture ayant montré des performances satisfaisantes : BERT (Devlin et al. 2019). Ce modèle fut l'un des premiers basés sur le Transformer à largement améliorer les performances en traitement de la langue. Cependant la question de la langue d'entraînement se posait. Le domaine agricole étant un langage assez particulier, nous nous sommes demandés s'il ne valait pas mieux utiliser un modèle entraîné spécifiquement sur du Français contrairement à BERT, issu d'un entraînement majoritairement en anglais. De tels projets existent comme FlauBERT (Le et al. 2020), CamemBERT (Martin et al. 2020) ou encore CamemBERTaV2 (Antoun et al. 2024). Le dernier, étant le plus récent, correspond à une déclinaison plus moderne de l'architecture RoBERTa appelée DeBERTa (He, Gao et W. Chen 2021). Cependant, il est commun d'utiliser des LLMs entraînés en anglais sur d'autres langues utilisant l'alphabet romain. Cela est a priori moins efficace (Ali et al. 2024) car la tokenisation qui en résulte est moins bonne, mais les langues utilisant l'alphabet romain ayant des structures et champs sémantiques assez similaires, nous décidons tout de même d'explorer cette voie.

Nous avons donc choisi de prendre le modèle le plus moderne en Français, CamemBERTaV2 et de comparer les performances obtenues avec l'architecture BERT pour observer si des différences notables se révélaient entre les deux modèles. Ainsi, nous utiliserons ces deux modèles afin d'extraire des informations textuelles une représentation vectorielle du fourrage.

2.4.3 Raffinage (ou Fine-Tuning) des modèles choisis

Au-delà de l'extraction d'embedding, la seconde force des LLMs est leur transférabilité et surtout la possibilité de les spécialiser une fois la phase de pré-traitement achevés. En effet, nous sommes passés depuis les 10 dernières années à l'ère des modèles de fondation, ces modèles pré-entraînés ensuite raffinés sur des tâches différentes du pré-entraînement ou sur un domaine sémantique particulier. Cela permet de renforcer les capacités du modèles sur des domaines souvent peu communs ou présentant un jargon spécifique comme la médecine (Devlin et al. 2019), où le vocabulaire et le sens des mots est sensiblement différent du langage courant. Cette approche usuelle dans l'utilisation de grands modèles de langues consiste donc à adapter la représentation contextuelle initiale, les représentations vectorielles des tokens, afin que cette dernière soit plus adaptée au domaine sur lequel est appliqué le grand modèle de langue.

Le raffinage de la représentation contextuelle des jetons au domaine de l'agriculture, et plus précisément à celui des fourrages, nous a semblé être une approche intéressante dans le contexte de notre problème. On observe en effet parmi les libellés de descriptions des mots dont le sens est inhérents au domaine agricole (par ex. : fructification, laiteux) et dont le sens pourra être mieux saisi, ou encore des mots qui pourraient ne pas être présent dans le vocabulaire initial des tokeniseurs, et donc être décomposé en jetons de plus

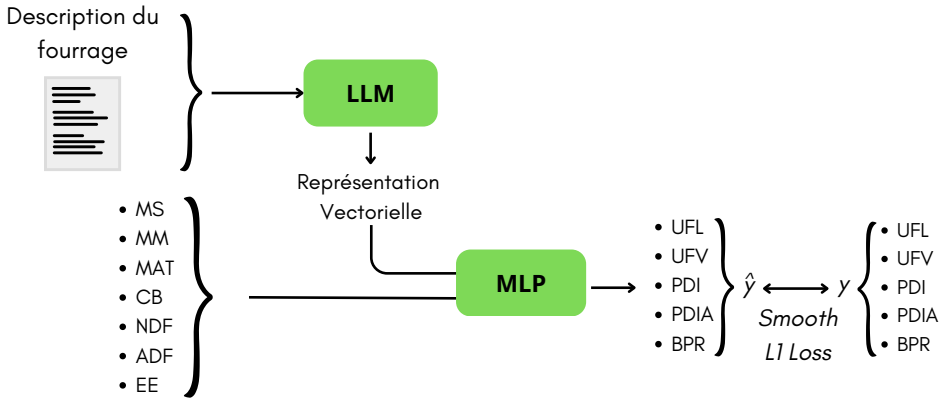


Figure 3 – Schéma de l'approche d'apprentissage neuronale proposée

petites tailles dont le sens devra être affinés (par ex. : ray-grass, dactyle). Ainsi, un raffinage nous a semblé être une approche pertinente pour améliorer les performances au sein de notre approche textuelles.
Afin de raffiner nos modèles, nous avons donc utilisé une version adaptée des modèles de langages masqués, introduit par le modèle BERT, proposés pour les modèles BERT et CamemBERTav2 et disponible sur la plateforme HuggingFace. Les modèles de langages masqués correspondent en effet au type de modèle qui permet de modifier la représentation vectorielle, et donc le sens, associé à chaque tokens. Nous avons utilisé ces modèles comme il suit. La liste des jetons a été copié, et une proportion de 0.15 des jetons, conformément à la proportion choisie lors de l'introduction du modèle, est remplacée par un jeton particulier : `[MASK]`, exception faite des jetons spéciaux : `[PAD]`, `[CLS]` et `[SEP]`, respectivement jetons de remplissage, jeton de classification, et jeton de séparation. Durant l'entraînement, le modèle agrège l'information de la phrase de manière bi-directionnelle sur le jeton `[MASK]`. La représentation vectorielle agrégée de `[MASK]` est ensuite projetée dans l'espace de dimension du vocabulaire du tokenizer, donnant un vecteur de taille 30522. Une normalisation est effectuée par la fonction softmax, le vecteur contient donc pour chaque mot du vocabulaire la probabilité que ce mot corresponde au mot masqué. Le mot ayant la plus haute probabilité est sélectionné et la perte est calculée par entropie croisée :

$$\mathcal{L}_{\text{MLM}} = -\sum_{i} p_i \log(\hat{p}_i) \tag{2}$$

avec
- p_i la probabilité réelle de la classe i,
- $\hat{p}_i$ la probabilité prédite que le token masqué appartienne à la classe i.

L'erreur est ensuite rétro-propagée au travers du réseau, et la représentation vectorielle des tokens (embedding) est raffinée sur les données utilisées pour ce modèle.

Nous avons utilisé les données Feedipedia ainsi qu'un jeu de données issu des données INRAE pour le raffinage de notre modèle, comprenant respectivement des descriptions de fourrages en langue française et en langue anglaise, et des associations de fourrages logiques. Le jeu de données total pèse 2794 MB, ce qui est relativement limité en comparaison des raffinages usuels (Mehrafarin, Rajaee et Pilehvar 2022). Le caractère limité du nombre d'exemples présents dans notre jeu de données est un autre facteur nous ayant poussé au choix de LLMs de très petites tailles car plus l'architecture grossit, plus l'étape de raffinement nécessite plus de données.

2.4.4 Entraînement et sélection des modèles d'apprentissage profonds

L'entraînement et la sélection des modèles se fait de la manière suivante. La campagne de paramétrisation des modèles est réalisée sur 15 epochs. Après chaque epoch, le modèle est évalué sur un ensemble de validation par une fonction de perte Hubert. Le modèle ayant le meilleur score sur l'ensemble de validation, sur les 15 epochs, est sélectionné. Le raffinement des modèles est réalisé sur 15 epochs par un modèle de langage masqué, comme détaillé dans la section précédente. A chaque epoch, le modèle est évalué sur un ensemble de test. Le modèle ayant les meilleures performances sur l'ensemble de test, sur les 15 epochs vues, est séléctionné. Une campagne d'hyper-paramétrisation a également été réalisée pour optimiser la

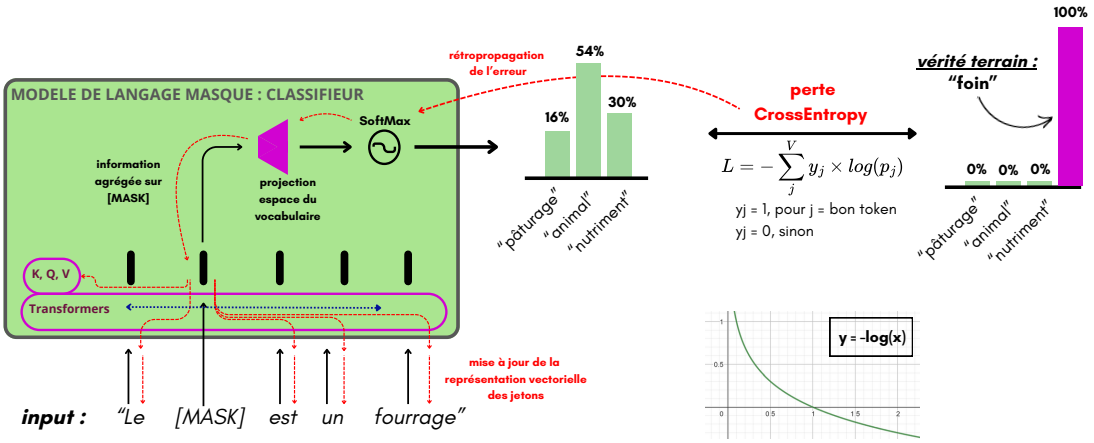


Figure 4 – Modèle de langage masqué pour le raffinage des grands modèles de langues

tête de régression. Cette campagne a été réalisé seulement sur le modèle CamemBERTaV2 non raffiné. La campagne est définie par une étude `Optuna` (Akiba et al. 2019), effectuant 100 études. L'évaluation est réalisée sur l'ensemble de validation après une epoch unique. Le meilleure modèle issu de cette campagne d'hyper-paramétrisation a été sélectionné.
Ainsi, le nombre d'epoch sur lesquelles les modèles sont séléctionnés pour le raffinement et la campagne de paramétrisation varie selon les modèles. Ces détails sont présentés dans la figure ci-dessous (Table 1) :

Modèle	**Raffinement**	**Meilleur epoch**	
		basique	**optimisé**
BERT	/	13	7
BERT	4	8	10
CamemBERTaV2	/	13	7
CamemBERTaV2	5	10	7

Table 1 – Meilleures epochs lors de l'entrainement (sur 20 epochs pour le Raffinement des Grands Modèles de Langues et sur 15 epochs pour la tâche de régression)

3 Résultats

Dans cette section, nous présentons les résultats de notre démarche en commençant d'abord par la campagne d'expérimentation basée sur les modèles non neuronaux (ML), suivie des résultats obtenus avec les modèles neuronaux (DL). Nous concluons par une comparaison entre les meilleurs modèles issus des deux approches.

3.1 Campagne d'expérimentation ML

L'établissement du niveau de performance atteignable par des méthodes d'apprentissage statistique classique afin d'appréhender les niveaux de performances possibles ainsi que de déterminer une méthode permettant d'obtenir un modèle satisfaisant s'est déroulé en deux étapes : la détermination des meilleurs pré-traitements suivie de l'identification du meilleur algorithme.

3.1.1 Remplacement des valeurs numériques manquantes et normalisation

Afin de déterminer le mode d'imputation des valeurs numériques, nous avons entraîné l'ensemble des modèles considérés dans la campagne avec différentes méthodes de remplacement des valeurs manquantes. Les moyennes des performances calculées sur l'ensemble des modèles, pour chaque méthode d'imputation

sont présentés en Figure 5. Chaque barre représente la moyenne des performances normalisées calculées pour chaque modèle pour chaque valeurs à prédire. Les résultats montrent qu'en moyenne, l'utilisation des K plus proches voisins pour estimer la valeur manquante donne de meilleures résultats.

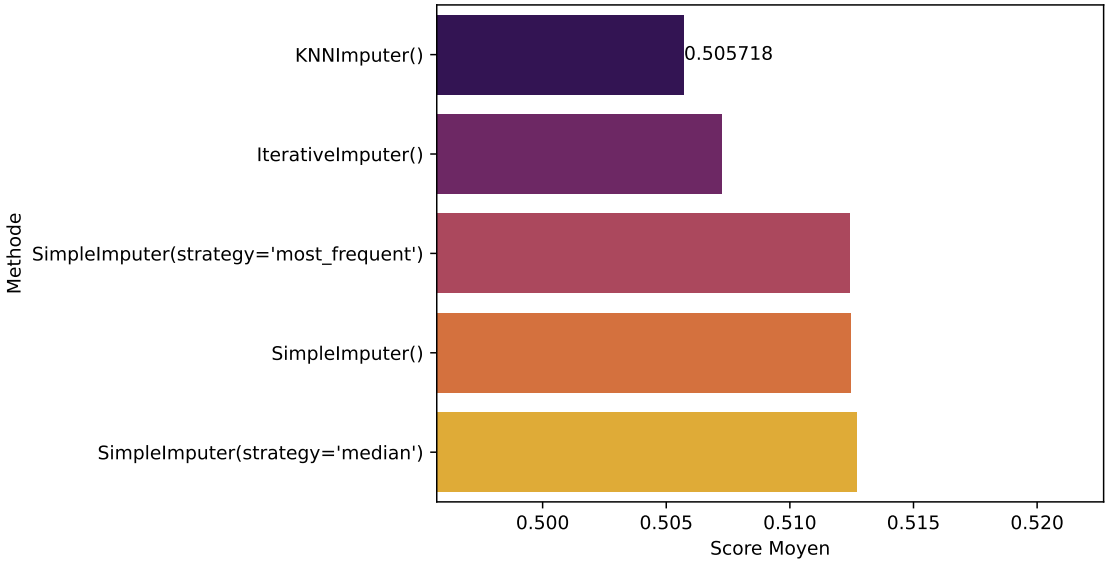


Figure 5 – Moyenne des performances RMSE en utilisant divers méthodes d'imputation des valeurs manquantes. De haut en bas : K plus proches voisins, Régresseur Bayésien de type Ridge, valeur la plus fréquente, moyenne, médiane. Ces valeurs sont calculées sur l'ensemble des valeurs à prédire et des modèles testés

Il nous a ensuite fallu déterminer la meilleure méthode de normalisation. Comme précédemment, les résultats sont présentés en Annexe 1. Nous utiliserons le `MinMaxScaler` par la suite.

3.1.2 Détermination des meilleurs pré-traitements

Comme indiqué plus tôt, nous avons explorer un ensemble de méthode afin de tirer partie des libéllés. L'ensemble de la campagne est résumé dans la Figure 6. Pour obtenir ces résultats, les différents modèles d'apprentissage statistique ont été testés sur chaque type de pré-traitement. La figure présente la moyenne (calculée sur les différentes valeurs à prédire) des meilleures scores RMSE obtenus pour chaque méthode de pré-traitement. Les meilleures méthodes correspondent à l'utilisation des deux premiers niveaux de libellés sous forme d'encodage disjoint complet accompagné des valeurs numériques ou à l'utilisation de l'analyse à composantes multiples appliquées à l'ensemble des libellés et de les accompagner des valeurs numériques.

Les meilleurs modèles ont ensuite été déterminés par validation croisée en utilisant le prétraitement L0 + L1 + VC. La Figure 7 montre les 3 meilleures algorithmes. Parmi le séparateur à vaste marge et le boosting de gradient, les implémentations de boosting de gradient se démarquent. Nous sélectionnons le `XGBoostRegressor` en raison de ses performances similaires sur le valeurs protéiques avec le `GradientBoostingRegressor` de `scikit-learn` tandis qu'il est meilleur sur les valeurs énergétiques, à savoir les plus complexes à prédire.

3.1.3 Résultats par des méthodes ensemblistes de boosting

Avec le modèle XGBoost sur les données numériques et textuelles, nous observons des niveaux de performances hétérogènes relativement au groupe de variable à prédire. Le modèle semble mieux capturer les informations relatives aux valeurs énergétiques UFL et UFV, avec un niveau de qualité de prédiction homogène au sein de ce sous-groupe cf. Table 2. Ces premiers résultats sont encourageants du fait de la proximité biologique de ces deux valeurs nutritionnelles ; trouver une grande différence entre les deux nous aurait alerté. Au sein du groupe de valeurs protéiques, une hétérogénéité est plus marquée dans les niveaux de prédictions entre BPR, et le groupe PDI et PDIA. Nous pouvons également noter ici que ces résultats sont en phase avec la très grande similarité entre PDI et PDIA. Néanmoins, on note que la valeur de BPR est moins bien prédite.

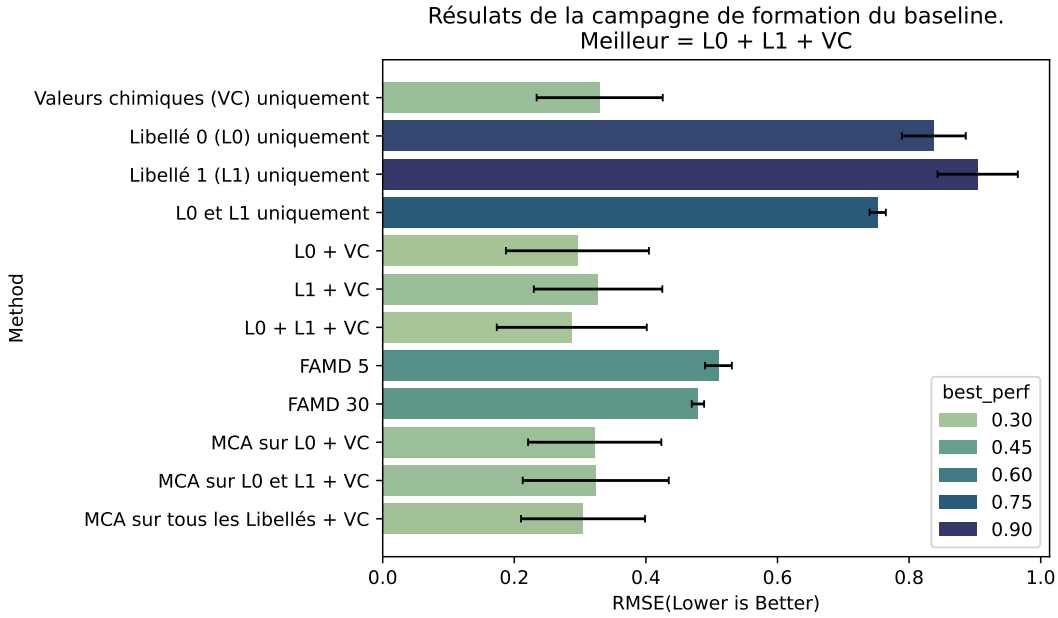


Figure 6 – Moyenne des performances RMSE en utilisant divers méthodes de prétraitements des Libéllés. Les barres d'erreurs correspondent à l'ecart type.

3.1.4 Effet des informations numériques et textuelles sur la qualité des prédictions

Les résultats montrent que la prise en compte des valeurs numériques et textuelles apportent de meilleurs performances que la prise en compte des informations numériques seules cf. Table 2. Les résultats montre une baisse du risque quadratique d'environ 0.02 sur les valeurs énergétiques de manière homogène, et de manière plus hétérogène, une baisse du risque quadratique de 2 pour les valeurs BPR et PDI, et une diminution moins marqué de 0.8 pour PDIA du modèle prenant en compte les informations textuelles par rapport à un modèle prenant en compte uniquement les valeurs numériques d'entrées. Cela conforte la capacité du modèle à extraire de l'information des données textuelles.
Nous avons également évalué les informations apportées par les données textuelles. Nous avons comparé le modèle XGBoost prenant en compte l'ensemble des données par rapport à un modèle prenant en compte seulement les valeurs textuelles. On observe également des performances homogène pour les valeurs énergétiques, avec une hausse du risque quadratique d'environ 0.2 (hausse de risque identique au modèle prenant seulement en compte les valeurs numériques). La perte d'information est plus marquée pour les valeurs protéiques, le risque quadratique augmente d'environ six (multiplication par trois) pour le BPR, et une hausse d'environ deux pour le PDI et le PDIA (multiplication par deux).
Ces résultats semblent donc indiquer que de l'information est contenue dans les textes et dans les valeurs numériques. Le modèle semble être en capacité d'extraire les deux types d'informations de manière indépendante. La combinaison de ces deux informations améliore la qualité des prédictions, encourageant une complémentarité des informations apportées par les deux modèles.

Modèle	**RMSE**				
	UFL	**UFV**	**BPR**	**PDI**	**PDIA**
XGBoostRegressor	**0.03**	**0.04**	**3.27**	**2.02**	**2.06**
XGBoostRegressor, Valeurs numériques	0.05	0.06	5.55	4.14	2.82
XGBoostRegressor, Libellés uniquement	0.05	0.06	11.51	4.46	3.78

Table 2 – Comparaison des résultats pour le modèle XGBoost

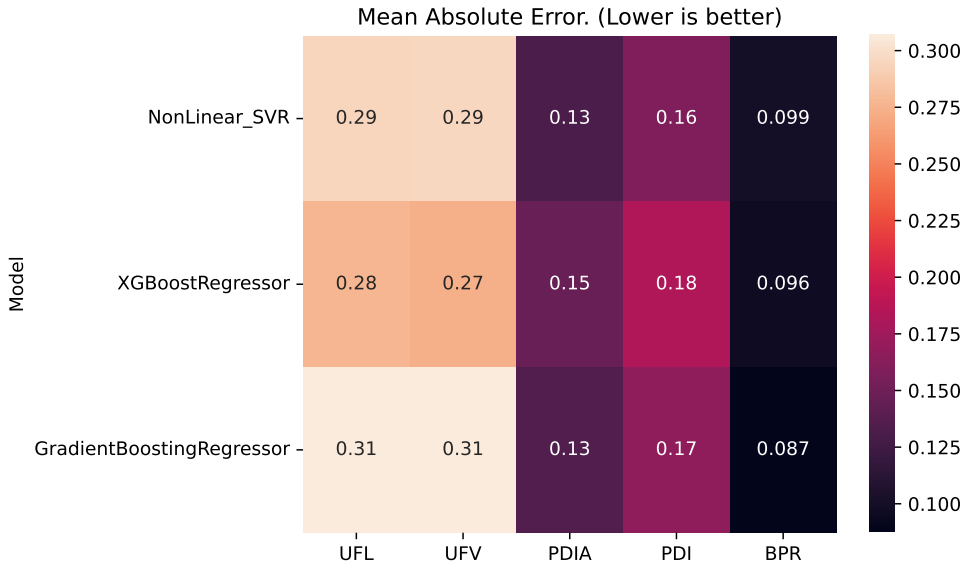


Figure 7 – Performance RMSE de validation croisée des 3 meilleurs algorithmes sur les valeurs à prédire.

3.2 Modèles neuronaux

Les architecture récentes de réseaux de neurones permettent de mieux capturer les informations textuelles que les approches traditionnelles de type "sac-de-mots". Cette meilleure intégration des données de textes permettrait d'abord de tirer partie des libellés, et nous espérons que ces informations puissent améliorer les performances des modèles. Cependant, et au delà des performances des modèles sur l'échantillon test, nous pouvons également espérer que les techniques de "zero-shots", capacité que possède les grands modèles de langage à généraliser des tâches à partir des grands volumes de données vus à l'entraînement et leurs spécialisation à capturer les contextes, puissent être performants dans des contextes qui n'auraient pas été vu durant l'entraînement.

3.2.1 Qualité de prédiction d'un régresseur faible

Nous avons tout d'abord souhaité comparer nos modèles avec un régresseur faible. Ce modèle prédit une valeur constante, à savoir la moyenne de la distribution $P(Y|X)$. La précision apportée par ce modèle est très faible : le coefficient de Pearson est très proche de 0 indiquant qu'aucune corrélation n'est capturée par le modèle Table 3.

Modèle	**RMSE**				
	UFL	**UFV**	**BPR**	**PDI**	**PDIA**
Régresseur faible	0.1302	0.1568	35.0161	12.0237	10.5821

Table 3 – Comparaison des résultats du modèle dummy

3.2.2 Effet de la tête de régression optimiséd et du raffinement

Nos modèles neuronaux profonds présentent des performances hétérogènes.
Tout d'abord, nous nous intéressons aux modèles ayant une tête de régression optimisée. Plus précisément, on observe d'abord les modèles dont l'encodeur, que ce soit BERT ou CamemBERTav2, ont été raffiné sur le corpus de texte d'information fourragère. On remarque tout d'abord que le modèle ayant comme encodeur CamemBERTaV2 semble avoir des performances plus importantes sur l'ensemble des valeurs protéiques, ainsi que sur UFL Table 4. Néanmoins, on n'observe pas de différence pour UFV. Le type du modèle semble donc avoir un effet sur les performances des modèles. Par ailleurs, pour comparer les performances apportées par le raffinage, on compare les modèles raffinés et non raffinés. Concernant CamemBERTaV2 premièrement, les valeurs d'erreur quadratique moyenne sur les valeurs énergétiques ne sont pas modifié. Sur les valeurs protéiques, les résultats sont plus contrastés : les valeurs associés à PDI

et PDIA sont très similaires. Néanmoins, on constate une hausse du risque quadratique pour la valeur de BPR : le raffinage du modèle de CamemBERTaV2 semble diminuer les performances du modèle pour cette valeur. Comme pour CamemBERTaV2, la valeur du risque quadratique est identique pour la valeur d'UFL et PDIA. Néanmoins, le raffinage du modèle semble à nouveau restreindre les capacités d'apprentissages des modèles pour les valeurs d'UFV, de BPR, et de PDI. Ces résultats d'effet du raffinage permettent de mettre en perspective la transposition de la relative liaison biologique entre PDI et PDIA. Ces résultats souligne également que l'effet du raffinage peut avoir un effet sur les valeurs protéiques comme sur les valeurs énergétiques, contrairement à ce qui apparaissait dans les résultats de comparaison entre BERT et BERT raffiné.

Afin d'évaluer les performances apportées par l'optimisation de la tête de régression, on s'intéresse également aux modèles raffinés et non raffinés. Tout d'abord, on observe une homogénéité des performances sur les valeurs énergétiques Table 4. Une tête de régression basique semble diminuer la variance entre les modèles de la précision des prédictions. Ceci encourage l'hypothèse que l'effet des encodeurs (BERT et CamemBERTav2) et leur possible raffinage a un effet moindre sur la qualité des prédictions, tandis que la tête de régression a un effet important sur la qualité des données. Dans une moindre mesure, on observe également quelques variations entre les modèles pour une cible donnée. Néanmoins, on observe une bien plus grande homogénéité qu'en utilisant une tête de régression optimisée. Ces observations renforcent l'hypothèse que les encodeurs, par leur nature ou leur raffinage, extraient les mêmes informations, et que la tête de régression constitue le levier de sensibilité pour obtenir de bonnes performances.
En comparant à nouveau les modèles BERT et CamemBERTaV2 raffinés cette fois-ci avec des têtes de régression "basic", on observe que le modèle BERT raffiné est meilleur pour les valeurs énergétiques UFV et UFL Table 5. Le modèle BERT raffiné est également meilleure sur la valeur de BPR. Néanmoins, nous n'observons pas de différence entre les qualités de prédictions de PDI et PDIA. Par ailleurs, les résultats de régression via l'utilisation d'une tête de régression "basic" permettent de mettre en évidence que le raffinage de BERT n'a pas d'effet : les risques quadratiques sont identiques en comparant BERT et BERT raffiné pour l'ensemble des valeurs cibles. Cependant, on observe en comparant les résultats de CamemBERTaV2 et CamemBERTaV2 raffinés que le raffinage semble diminuer la performances des prédictions pour ce modèle, confirmant la tendance observée sur les têtes de régression optimisée. CamemBERTav2 non raffiné a une plus grande qualité de prédictions sur les valeurs énergétiques, ainsi que sur la valeur BPR. Il apparaît tout de même que CamemBERTaV2 raffiné a de meilleurs qualité de prédiction pour PDI et PDIA. Ainsi, ces résultats soulignent à nouveau la tendance de BERT à mieux capturer les dépendances informationnelles dans les jeux de données. Néanmoins, l'effet du raffinage des modèles de langage, qu' il d'agisse de BERT ou CamemBERTaV2, semble anormalement diminuer les performances des modèles.

Ces premières observations nous apportent plusieurs informations. Les modèles neuronaux semblent mieux capturer l'information relative aux valeurs énergétiques que protéiques, quels que soient les modèles de langues, leur raffinage, ou la tête de régression. On observe tout de même des niveaux de performances très intéressants et les modèles semble pertinents. De plus, les modèles utilisant une tête de régression basique semblent plus performants sur l'ensemble des valeurs cibles que les modèles équipés de têtes de régression optimisée. Plus précisément, la différence entre les performances de prédictions d'UFV, entre les modèles basiques et optimisés, souligne que l'information est moins bien capturé lorsque pour par les têtes de régression optimisés, et souligne ainsi le rôle du décodeur dans la faible qualité des prédictions. De manière plus surprenante, BERT ayant une tête de régression optimisée semble très mal capturer l'information relative aux UFL.

Cible	BERT	BERT Raffiné	CamemBERTaV2	CamemBERTaV2 Raffiné
UFL	0.08	0.08	**0.05**	**0.05**
UFV	**0.06**	0.09	0.09	0.09
BPR	3.93	**3.73**	4.02	4.28
PDI	3.22	**3.05**	3.44	3.35
PDIA	2.36	**2.3**	2.68	2.62

Table 4 – Comparaison des RMSE pour des modèles avec une **tête de régression optimisée**, raffinés ou non.

Cible	BERT	BERT Raffiné	CamemBERTaV2	CamemBERTaV2 Raffiné
UFL	**0.04**	**0.04**	**0.04**	**0.04**
UFV	**0.05**	**0.05**	**0.05**	**0.05**
BPR	3.31	3.35	**3.19**	3.64
PDI	2.43	2.53	**2.3**	2.43
PDIA	1.8	1.83	**1.71**	1.75

Table 5 – Comparaison des RMSE pour des modèles avec une **tête de régression basique**, raffinés ou non.

3.2.3 Effet des informations textuelles

Afin de s'intéresser aux informations apportées indépendamment par les textes et les valeurs numériques, on s'intéresse à 3 modèles : un réseau de neurones seul en entrée, seulement les valeurs numériques d'entrée, et deux encodeurs, BERT et CamemBERTaV2, non raffiné, qui ne prennent en entrée que les entrées textuelles de descriptions des fourrages.
Tout d'abord, on observe que les modèles ne prenant en compte que les valeurs textuelles ont de très mauvaises performances, indépendamment du modèle utilisé Table 6. Le risque quadratique pour les deux modèles - BERT et CamemBERTaV2 - est en moyenne multiplié par deux pour les valeurs énergétiques UFV et UFL. Concernant les valeurs protéiques, l'erreur quadratique moyenne est multipliée par 10. Pour les deux groupes de valeurs nutritionnelles, on observe des coefficients de Pearson très proche de 0, n'indiquant qu'aucune information n'a pu être extraite des textes. Ces résultats semblent donc montrer que l'information textuelle comporte soit très peu d'information, soit qu'elle est très mal capturée par les modèles de langages.
Les performances du modèle n'utilisant que les valeurs numériques sont meilleures que les modèles incluant des informations textuelles Table 6. Le risque quadratique est plus faible pour l'ensemble des valeurs énergétiques, ainsi que pour PDI et PDIA. Seul BPR a un niveau de performance similaire aux meilleurs modèles neuronaux.
Les informations textuelles ne semblent pas contenir d'informations d'après les résultats que nous obtenons sur les modèles prenant en compte les informations textuelles seulement. La meilleure qualité des prédictions prenant en compte seulement les valeurs numériques semblent indiquer que les informations textuelles impactent négativement la qualité des prédictions.

Cible	BERT, Texte uniquement	CamemBERTaV2, Texte uniquement	Perceptron Multi Couche, Numériques uniquement
UFL	0.11	0.11	**0.03**
UFV	0.13	0.13	**0.04**
BPR	34.23	34.24	**3.31**
PDI	11.4	11.39	**2.15**
PDIA	10.19	10.19	**1.51**

Table 6 – Comparaison des RMSE pour des modèles non raffinés en mode basique utilisant uniquement des informations textuelles comparé à l'utilisation uniquement des valeurs numériques.

3.3 Comparaison entre le meilleur modèle ML et les modèles DL

Dans cette partie, nous confrontons les performances du meilleur modèle issu de la campagne ML avec celles obtenues par les approches DL. Cette comparaison permet de dégager les points forts et les limites de chaque approche.

Les résultats soulignent que XGBoost a de très bonnes performances relativement aux autres modèles, informés par les données numériques seulement, ou augmenté des données textuelles.
Concernant les données énergétiques, XGBoost surpasse en performances les modèles d'apprentissage profond entraîné sur les données numériques et textuelles. La RMSE des prédictions par XGBoost est de 0.03 pour l'UFL et de 0.04 pour l'UFV, alors que les meilleurs modèles (modèle de langues raffinés et non raffinés avec une tête de régression basique) obtenaient respectivement des scores de 0.04 et de 0.05 pour les valeurs d'UFL et d'UFV Table 5.
En comparaison avec les niveaux de performances de l'apprentissage profond entraîné sur les données numériques seules, les niveaux de performances sont équivalents aux modèles d'apprentissage profond, entraîné seulement sur les données numériques (Table 6). Il est également intéressant de noter que l'écart

en proportion entre les valeurs d'UFV et d'UFL pour les modèles comportant une tête de régression basique est conservé dans les prédictions du modèle XGBoost.

Concernant les données protéiques, la qualité des prédictions est plus hétérogène.
En effet, pour les modèles prenant en compte les données textuelles et numériques, PDI est mieux prédit par les modèles XGBoost, avec une valeur de RMSE égale 2.02, contre 2.3 pour le modèle CamemBERTaV2 non raffiné avec une tête de régression basique (Table 2). Néanmoins, les valeurs de BPR et PDIA sont moins bien prédites par le modèle XGBoost. Pour BPR spécifiquement, la valeur de RMSE est de 3.27 (Table 5), contre 3.19 pour le modèle CamemBERTaV2. De manière similaire, PDIA a une valeur de RMSE de 2.06, contre 1.71 pour le modèle CamemBERTaV2 non raffiné (Table 5).
En comparaison avec le modèle d'apprentissage profond informé seulement des données numériques, les performances sont très similaires pour BPR : 3.31 contre 3.27 respectivement, meilleur pour la valeur de PDI : 2.02 contre 2.15, et enfin moins performante pour la valeur de PDIA : 2.06 contre 1.51.

Ainsi, la comparaison des performances des modèles XGBoost, et des modèles d'apprentissages profonds sur les données textes uniquement et ajoutés des données numériques montre que les performances apporté par XGBoost sont soit meilleures, soit très proches des valeurs de référence de prédictions établis par les modèles profond. Ces modèles pourront donc être retenus en perspective d'une application dans des contextes connus.

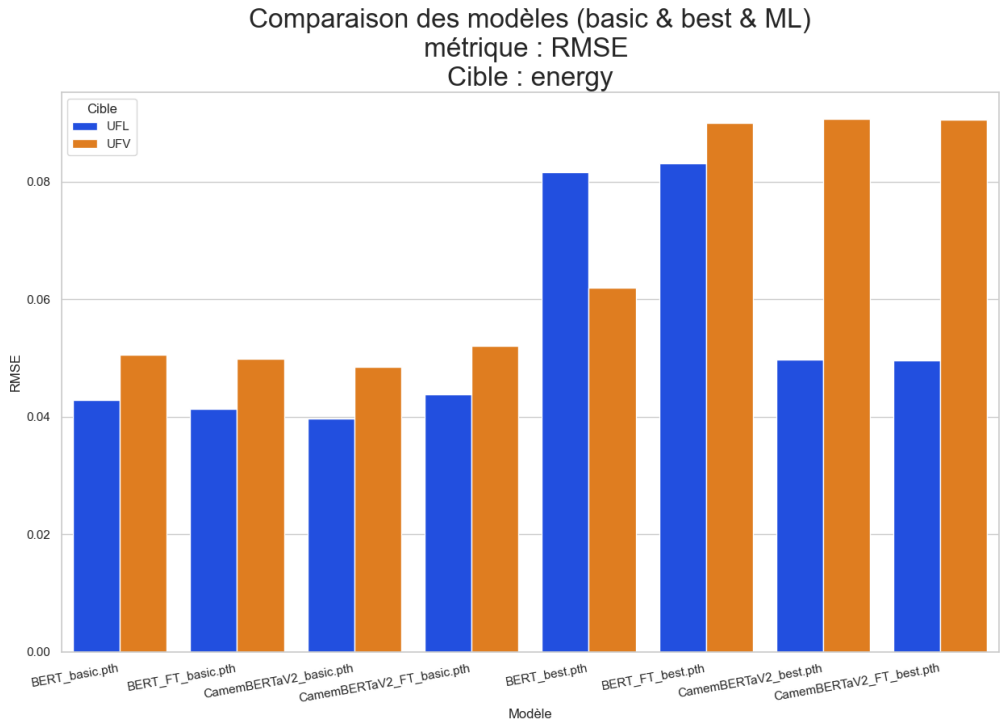


Figure 8 – Effet de la tête de régression "optimisée" pour les valeurs énergétiques (UFL et UFV). Chaque barre correspond à la moyenne du score RMSE sur chaque lot (batch) de test. On observe de moins bonnes performances.

4 Discussion

4.1 Influence de l'optimisation de la tête de régression

L'effet des têtes de régression est assez paradoxale. D'un côté, les têtes de régression optimisées dites "best" présentent des performances plus faibles que les têtes dites "basic". Or les têtes optimisées l'ont été grâce à une campagne d'optimisation de l'architecture : nombre de couches, et taille de chaque couche, fonctions d'activations (une trentaine de paramètres au total) sur le modèle CamemBERTaV2 non raffiné. Une explication possible de cette sous-performance des modèles comportant une tête best réside peut-être d'abord dans le fait que la tête de régression a été optimisée sur une architecture unique : CamemBERTaV2. Par ailleurs, la tête a été optimisé sur une seule epoch pour des soucis de ressources computationnelles.

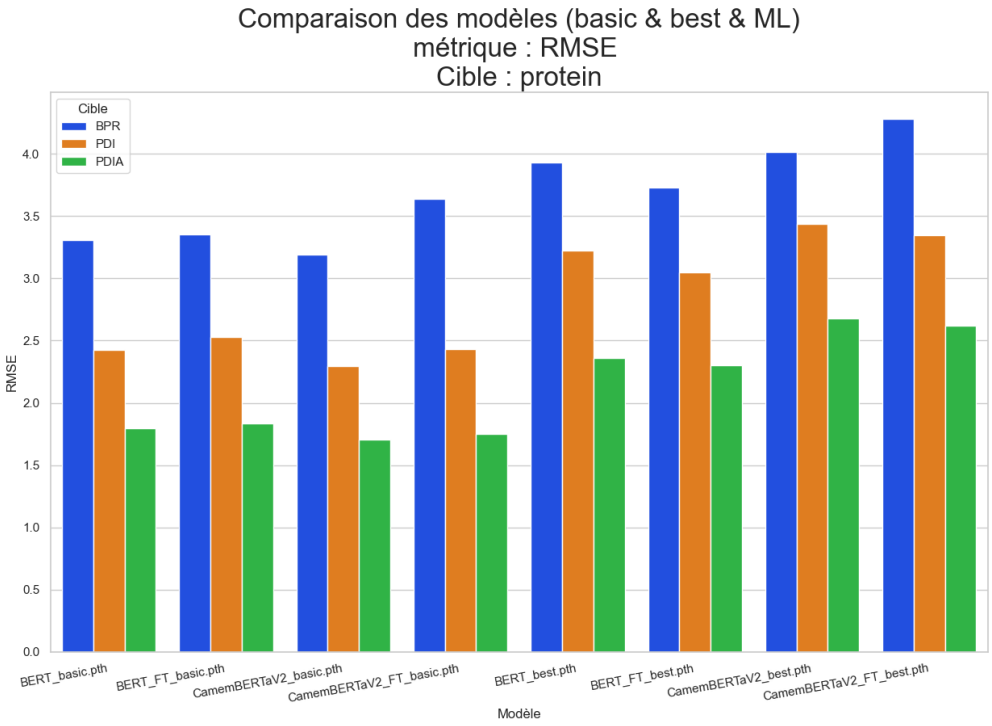


Figure 9 – Effet de la tête de régression "optimisée" pour les valeurs protéiques (BPR, PDI, PDIA). Chaque barre correspond à la moyenne du score RMSE sur chaque lot (batch) de test. On observe de moins bonnes performances.

Cela a conduit à une exploration relativement faible des architectures possibles car le nombre de tête envisageables augmente de manière exponentielle par rapport aux paramètres à tester. Par ailleurs, il est largement possible que la meilleure architecture au bout d'une époque converge vers un minimum local qui est moins bon que les minima locaux atteint plus tard au bout de 15 epochs lors de la validation.
Ainsi, il serait nécessaire de réaliser à nouveau la campagne d'optimisation des hyper-paramètres sur un nombre d'epoch égal à celui utilisé pour la validation, soit 15 epochs. Par ailleurs, il serait également souhaitable d'effectuer cette recherche de la meilleure architecture de tête de régression pour chaque grand modèle de langue, ici, pour les quatre configurations possible : BERT et CamemBERTaV2, raffiné et non raffiné. En effet, bien que ces modèles retournent des vecteurs de la même dimension, il est tout à fait possible que l'espace latent ne soit pas structuré de la même manière et qu'il soit nécessaire d'avoir des architecture différentes pour capturer ces différentes organisations.

4.2 Choix de la fonction coût pour l'entraînement des réseaux de neurones

Nous avons utilisé la fonction coût SmoothL1 qui s'apparente à une fonction telle que l'Erreur Absolue Moyenne (EAM) mais dont les pentes sont différentes et la différentiabilité en 0 assurée. Il est important de noter que le choix d'une fonction coût qui ne serait pas similaire à la métrique que nous utilisons pour évaluer et comparer nos modèles peut mener à des solutions suboptimales vis à vis de la métrique choisie. Dans l'ensemble du rapport nous avons décidé de présenter les scores RMSE ce qui fait du choix de la SmoothL1 Loss un choix discutable puisque la très classique Mean Squared Error Loss (critère des moindres carrés) serait ici la plus adaptée. Ceci s'explique par le fait que nous avons démarré notre projet en utilisant l'EAM puis changé de métrique pour la RMSE. Celle-ci permettait en effet de présenter des scores comparables entre valeurs à prédire (qui, nous le rappelons, évoluent dans des intervalles très différents).

4.3 Limites des méthodes d'apprentissages statistiques non neuronales

On a observé durant l'analyse des résultats, que le modèle XGBoost possédait d'excellentes qualités de prédictions, avec des valeurs de RMSE minimales ou équivalentes pour certaines valeurs cibles.
Par ailleurs, il est intéressant de noter que ces qualités de prédictions sont excellentes à la vue des

ressources computationnelles exigées par les modèles XGBoost. Néanmoins, la principale limite des modèles d'apprentissage statistique non neuronaux est l'absence de capacités dites "zero-shots", c'est à dire de réalisation de prédictions sur des données d'entrées qui n'aurait pas été vu durant l'entraînement du modèle. Or cette capacité de "zero-shots" est un attendu du projet. Elle est en effet cruciale dans le domaine des rations fourragères : la difficulté de récolter des données sur l'ensemble des territoires agricoles afin de paramétrer des modèles mécaniste est un véritable verrou à l'analyse fourragère dans certaines régions. Cette technique de "zero-shots", qui est permise par les grands modèles de langue offre une capacité à s'adapter au contexte et ce qui n'est pas le cas des modèles XGBoost. En ce sens, bien que les performances des modèles XGBoost soient excellentes, d'autant plus relativement à les ressources computationnelles que ces modèles demandent, leurs incapacités à généraliser dans des contextes différents limitent leur intérêt dans le cadre du projet.

4.4 Effet des libellés dans les modèles neuronaux

Comme nous l'avons vu précédemment, l'ajout des libellés dans les modèles neuronaux affecte négativement les performances. En comparaison, le modèle profond utilisant uniquement les valeurs numériques est meilleur que ceux utilisant également le texte. Et les modèles ne prenant en compte que les données textuelles ne capture aucune information. Pourtant, dans le cadre des modèles non-neuronaux, l'ajout des libellé apporte des informations en ce qu'il augmente les performances des modèles. Il semblerait donc que l'ajout des données textuelles augmente le bruit dans les données ce qui dilue l'information présente et fait diminuer les performances.
Nous avons tenté d'explorer cette hypothèse en comparant la distribution des poids des neurones de la première couche de la tête de régression (celle connecté directement à la concaténation représentation vectorielle textuelle-valeurs chimiques). Ces poids représentent l'importance des neurones de la vectorielle textuelle et ceux prenant les valeurs chimiques. On compare la distribution des poids des neurones du modèle BERT, non raffiné, et ayant une tête basique Figure 10 :

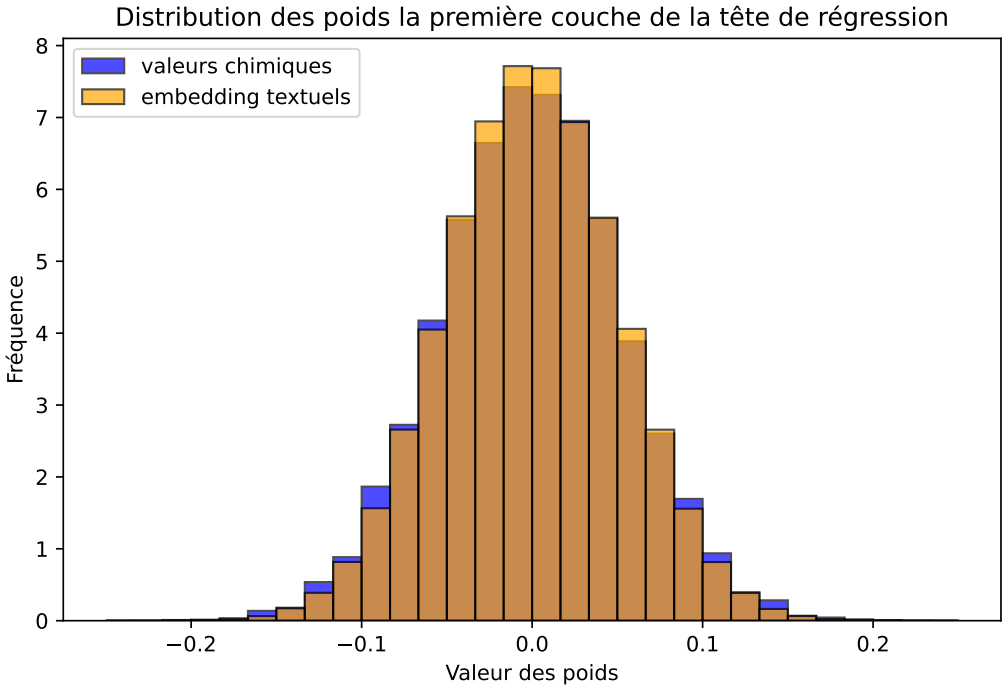


Figure 10 – Distribution des poids de la première couche de la tête de régression. On observe des distributions quasiment identiques.

On observe que la distribution des poids est identique, normale, ayant une moyenne très proche de zéros. Cela semble indiquer que les poids attribués aux valeurs chimiques ne sont pas plus élevés que ceux attribués aux valeurs chimiques. Ceux-ci est étonnant. Les résultats précédents ont montré que bien

plus d'informations étaient contenues dans les valeurs chimiques que textuelles. Cette distribution semble indiquer le contraire. Il semblerait que la disproportion dans la taille de la représentation vectorielle des textes (768) par rapport aux valeurs chimique (7) a un effet de "dilution" de l'information.

Il serait intéressant de comparer les distributions, et plus précisément leurs moyennes en assumant une distribution normale par un test de Student. Plus particulièrement, on pourrait comparer avec les modèles optimisés qui réduisent la taille de la représentation vectorielle des textes à un espace à 16 dimensions pour évaluer si les poids ont augmenté pour les valeurs chimiques.

4.5 Effet du raffinage sur les performances

Le raffinage des modèles de langues ne semblent pas apporter beaucoup d'informations. Ceci peut en partie être expliqué par le fait que, comme vu précédemment, les données textuelles ne contiennent pas beaucoup d'information pour nos modèles neuronaux. De plus, le raffinage diminue même les performances ; nous pouvons en déduire que les modèles brutes sont déjà très bons et suffisamment spécialisés sur notre tâche pour ne pas nécessiter d'entraînement supplémentaire.

4.6 Choix du modèle de langage masqué

Afin de raffiner nos modèles de langues, nous avons utilisé un modèle de langage masqué. Ces modèles ont été introduit par Devlin et al. 2019. Nous avons choisi cette approche car elle est généralisée, bien documentée et relativement facile à prendre en main. Néanmoins il existe d'autres techniques pour raffiner les modèles. Des modèles plus récents, tel qu'ELECTRA proposé par He, Gao et W. Chen 2021, se basant sur la détection de jetons incorrects a montré de bonnes performances dans cette tâche

4.7 Perspectives d'application des modèles proposés

Il existe plusieurs méthodes pour sauvegarder les modèles entraînés. L'inférence étant beaucoup moins coûteuse que l'entraînement, particulièrement dans le cadre de petits modèles comme ceux discutés ici, il est possible d'envisager exécuter le modèle retenu sur un smartphone sous forme d'une application, ou couplé à un analyseur de fourrage infrarouge, projet que l'Association française de Zootechnie. Dans cette optique, une démo web écrite en Python, avec la bibliothèque Flask, a été réalisée pour montrer une preuve de concept. Les modèles doivent encore être explorés afin de valider leurs performances finales, mais en conservant les modèles légers, un déploiement sur le terrain en local parait envisageable. Une fois le modèle éprouvé sur le terrain en France métropolitaine, nous pouvons imaginer des généralisations du modèle à d'autres cas d'applications (ex : élevage de non-ruminants) ou zones géographiques (ex : DROM, péninsule ibérique). Ces possibilités semblent bien raisonner avec les enjeux de l'analyse fourragère, ainsi qu'avec les attentes de l'AFZ.

5 Conclusion

L'objectif du projet était donc de développer des modèles statistiques complémentaires aux modèles mécanistes, afin de palier au manque d'exhaustivité des données pour paramétrer ces derniers modèles. Dans nos travaux, nous avons donc comparé des approches d'apprentissage neuronale et non neuronale. La qualité des prédictions sur l'ensemble des cibles nutritionnelles, ainsi que la frugalité des approches par boosting - tel qu'XGBoost implémenté par la librairie homonyme - démontre la pertinence des estimateurs non neuronaux pour cette tâche de régression fourragère. En opposition, les modèles neuronaux paraissent plus difficiles à stabiliser, et la variété des approches neuronales - raffinement des modèles de langue, choix de l'architecture de régression - ont permis d'identifier des modèles avec d'excellents niveaux de prédictions, similaires ou excédant ceux obtenus par les meilleures approches non neuronales. Un point à noter est qu'il semblerait que les meilleures approches non neuronales par boosting capturent correctement l'information contenue dans les libellés textuels associés au fourrage, ce qui n'est pas le cas des modèles profonds. Cette incapacité des modèles neuronaux à extraire l'information textuelle est majeure, dans le sens où il s'agit de l'objectif même du projet afin de généraliser les données fourragères. De manière plus générale, malgré des moyens matériels limités et une quantité de données limitée, nous avons réussi à établir un modèle de prédiction de bonne qualité. Avec davantage de temps et de moyens, il nous parait sûrement possible d'expérimenter d'autres architectures permettant de mieux capturer les données textuelles et d'améliorer les performances des modèles. Après discussion avec des experts, les résultats semblent prometteurs mais

il faudrait toutefois pouvoir tester les capacités de généralisation de nos modèles sur de nouvelles données avant d'envisager leur adaptation pour des applications en conditions réelles.

Remerciements

Nous souhaitons remercier tout d'abord Valérie Heuzé, et Gilles Tran de l'Association Française de Zootechnie, pour leur ambition, leur confiance et leur enthousiasme pour nos travaux. Nous souhaitons également remercier leurs collègues de l'Association française de Zootechnie, pour l'intérêt et toutes les questions qu'ils ont posées lors de notre présentation. De plus, nous souhaitons également remercier Madame Martin et Monsieur Cornuejols, nos professeurs, pour leur aide et leur soutien. Nous souhaitons également remercier Vincent Guigue, également professeur à AgroParisTech, pour le temps qu'il nous a accordé, ses conseils précieux, et son aide pratique.

Références

Agabriel, J. et Inra (France) (2007). *Alimentation des bovins, ovins et caprins : besoins des animaux, valeurs des aliments : tables Inra 2007*. Guide pratique. Quae. ISBN : 9782759200207. URL : https://books.google.fr/books?id=GEeKowZdIF4C.

Akiba, Takuya et al. (2019). "Optuna : A Next-generation Hyperparameter Optimization Framework". In : *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Ali, Mehdi et al. (17 mars 2024). *Tokenizer Choice For LLM Training : Negligible or Crucial ?* DOI : 10.48550/arXiv.2310.08754. arXiv : 2310.08754. URL : http://arxiv.org/abs/2310.08754 (visité le 27/02/2025).

Antoun, Wissam et al. (13 nov. 2024). *CamemBERT 2.0 : A Smarter French Language Model Aged to Perfection*. DOI : 10.48550/arXiv.2411.08868. arXiv : 2411.08868. URL : http://arxiv.org/abs/2411.08868 (visité le 10/01/2025).

Bastianelli, Denis et al. (18 jan. 2019). "La spectrométrie dans le proche infrarouge pour la caractérisation des ressources alimentaires". In : *INRA Productions Animales* 31.3, p. 237-254. ISSN : 2273-7766, 2273-774X. DOI : 10.20870/productions-animales.2018.31.2.2330. URL : https://productions-animales.org/article/view/2330 (visité le 24/02/2025).

Breiman, Leo (1er oct. 2001). "Random Forests". In : *Machine Learning* 45.1, p. 5-32. ISSN : 1573-0565. DOI : 10.1023/A:1010933404324. URL : https://doi.org/10.1023/A:1010933404324 (visité le 25/02/2025).

Breiman, Leo et al. (1984). *Classification And Regression Trees*. 1re éd. Routledge. ISBN : 9781315139470. DOI : 10.1201/9781315139470. URL : https://www.taylorfrancis.com/books/9781351460491 (visité le 25/02/2025).

Chang, Yin-Wen et al. (2010). "Training and Testing Low-degree Polynomial Data Mappings via Linear SVM". In : *Journal of Machine Learning Research* 11.48, p. 1471-1490. URL : http://jmlr.org/papers/v11/chang10a.html.

Chen, Tianqi et Carlos Guestrin (13 août 2016). "XGBoost : A Scalable Tree Boosting System". In : *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA : Association for Computing Machinery, p. 785-794. ISBN : 9781450342322. DOI : 10.1145/2939672.2939785. URL : https://dl.acm.org/doi/10.1145/2939672.2939785 (visité le 25/02/2025).

Cover, T. et P. Hart (jan. 1967). "Nearest neighbor pattern classification". In : *IEEE Transactions on Information Theory* 13.1, p. 21-27. ISSN : 0018-9448, 1557-9654. DOI : 10.1109/TIT.1967.1053964. URL : http://ieeexplore.ieee.org/document/1053964/ (visité le 25/11/2024).

Devlin, Jacob et al. (24 mai 2019). *BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding*. DOI : 10.48550/arXiv.1810.04805. arXiv : 1810.04805. URL : http://arxiv.org/abs/1810.04805 (visité le 27/02/2025).

Drucker, Harris et al. (1996). "Support Vector Regression Machines". In : *Advances in Neural Information Processing Systems*. T. 9. MIT Press. URL : https://papers.nips.cc/paper_files/paper/1996/hash/d38901788c533e8286cb6400b40b386d-Abstract.html (visité le 25/02/2025).

Escofier, Brigitte et Jérôme Pagès (2008). *Analyses factorielles simples et multiples : objectifs, méthodes et interprétation*. 4e éd. Sciences sup. Paris : Dunod. ISBN : 9782100519323.

Friedman, Jerome H. (fév. 2002). "Stochastic gradient boosting". In : *Computational Statistics & Data Analysis* 38.4, p. 367-378. ISSN : 01679473. DOI : 10.1016/S0167-9473(01)00065-2. URL : https://linkinghub.elsevier.com/retrieve/pii/S0167947301000652 (visité le 25/02/2025).

He, Pengcheng, Jianfeng Gao et Weizhu Chen (2021). *DeBERTaV3 : Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing*. DOI : 10.48550/ARXIV.2111.09543. URL : https://arxiv.org/abs/2111.09543 (visité le 27/02/2025).

INRA (2018). "Alimentation des ruminants : Apports nutritionnels-Besoins et réponses des animaux-Rationnement-Tables des valeurs des aliments". In : *Éditions Quae. 728p. ISBN : 978-2-7592-2867-6.*

Inra (1978). "Alimentation des ruminants". In : *INRA Publ., Versailles.*

IterativeImputer (2024). scikit-learn. URL : https://scikit-learn/stable/modules/generated/sklearn.impute.IterativeImputer.html (visité le 25/11/2024).

Jarrige, Robert (1988). "Alimentation des bovins, ovins et caprins". In.

Le, Hang et al. (mai 2020). "FlauBERT : Unsupervised Language Model Pre-training for French". In : *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France : European

Language Resources Association, p. 2479-2490. URL : https://www.aclweb.org/anthology/2020.lrec-1.302.

Martin, Louis et al. (2020). "CamemBERT : a Tasty French Language Model". In : *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7203-7219. DOI : 10.18653/v1/2020.acl-main.645. arXiv : 1911.03894[cs]. URL : http://arxiv.org/abs/1911.03894 (visité le 06/01/2025).

Mehrafarin, Houman, Sara Rajaee et Mohammad Taher Pilehvar (17 mars 2022). *On the Importance of Data Size in Probing Fine-tuned Models*. DOI : 10.48550/arXiv.2203.09627. arXiv : 2203.09627. URL : http://arxiv.org/abs/2203.09627 (visité le 29/01/2025).

Pagès, J. (2004). "Analyse factorielle de données mixtes". In : *Revue de Statistique Appliquée* 52.4, p. 93-111. ISSN : 2400-4812. URL : http://www.numdam.org/item/?id=RSA_2004__52_4_93_0 (visité le 11/01/2025).

Tibshirani, Robert (1er jan. 1996). "Regression Shrinkage and Selection Via the Lasso". In : *Journal of the Royal Statistical Society Series B : Statistical Methodology* 58.1, p. 267-288. ISSN : 1369-7412, 1467-9868. DOI : 10.1111/j.2517-6161.1996.tb02080.x. URL : https://academic.oup.com/jrsssb/article/58/1/267/7027929 (visité le 24/11/2024).

Tikhonov, A. N. (1943). "On the stability of inverse problems". In : *Proceedings of the USSR Academy of Sciences* 39, p. 195-198. URL : https://api.semanticscholar.org/CorpusID:202866372.

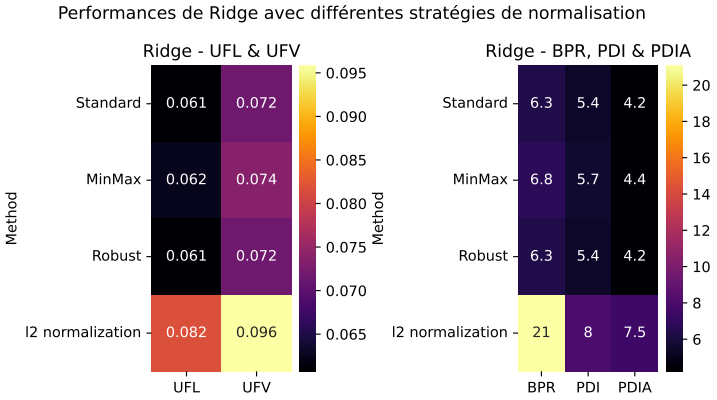
Vaswani, Ashish et al. (2 août 2023). *Attention Is All You Need*. DOI : 10.48550/arXiv.1706.03762. arXiv : 1706.03762. URL : http://arxiv.org/abs/1706.03762 (visité le 27/01/2025).

Zou, Hui et Trevor Hastie (1er avr. 2005). "Regularization and Variable Selection Via the Elastic Net". In : *Journal of the Royal Statistical Society Series B : Statistical Methodology* 67.2, p. 301-320. ISSN : 1369-7412, 1467-9868. DOI : 10.1111/j.1467-9868.2005.00503.x. URL : https://academic.oup.com/jrsssb/article/67/2/301/7109482 (visité le 24/11/2024).

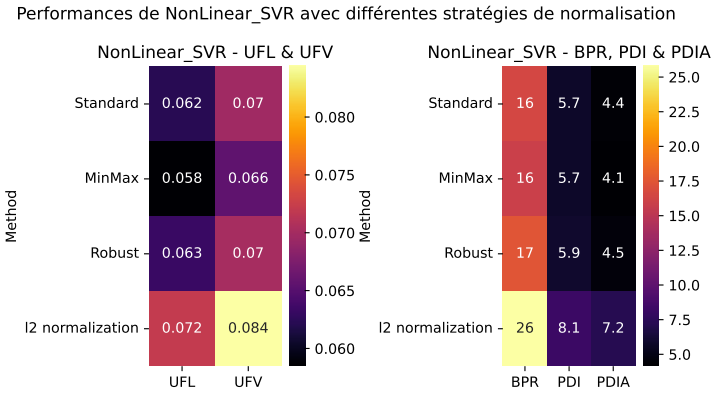
Annexes

Résultats de l'étude de la meilleure normalisation pour les modèles non neuronaux

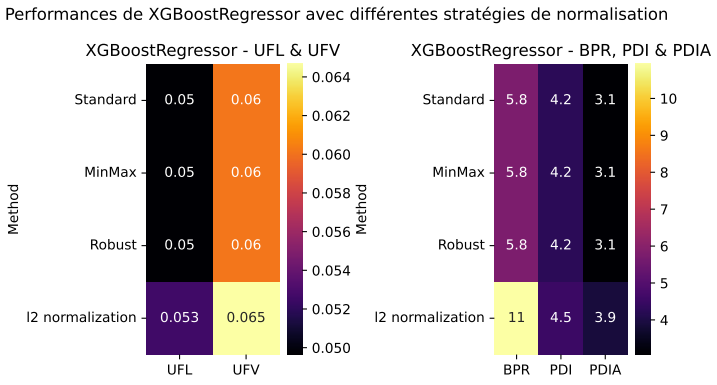
Les résultats suivant correspondent aux performances obtenues en normalisant selon différentes méthodes : Standardisation, Min-Max, Médiane/Intervalle Inter Quantile et par normalisation L2. Seules les performances d'un type de modèle (régression linéaire, Séprateur à vaste marge et approche boosting d'arbre) sont présentées.



(a) Régression Ridge



(b) SVR non linéaire



(c) XGBoost Regressor

Figure 11 – Comparaison des performances selon la normalisation pour différents modèles non neuronaux.