Projet de NLP

Master 2 Modélisations, Statistiques, Économiques et Financière (MoSEF)

Challenge Rakuten : Prédiction Multimodales des Codes de Type de Produit

Réalisé par :
Salma Benmoussa
Charlotte Cegarra

Année Universitaire : 2024-2025

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Abstract

Le challenge du RIT porte sur la classification multimodale de références produits à large échelle, à partir d'images et de textes (titres et descriptions). Nous avons d'abord nettoyé l'ensemble visuel en supprimant les doublons et en redimensionnant les images, puis extrait des caractéristiques profondes via un CNN (EfficientNet-B0) complété par un modèle de type Transformer (Swin-Tiny). Côté texte, les libellés produits ont été encodés avec BERT, dont les représentations contextuelles ont été fusionnées à un traitement par CNN 1D pour capter à la fois la sémantique fine et les structures locales. Les deux modalités sont ensuite combinées par concaténation dans un perceptron multicouche, puis affinées via un bloc d'attention cross-modal pour modéliser l'interaction image–texte. L'évaluation, fondée sur le F1 pondéré et l'analyse de la matrice de confusion, confirme le gain apporté par la fusion multimodale par rapport aux approches unimodales et met en lumière des catégories encore difficiles à distinguer, ouvrant la voie à des techniques d'oversampling ciblé ou de rééquilibrage.

I - Introduction

Le commerce électronique connaît une forte expansion, porté par des géants comme Rakuten, qui gère un immense volume de produits et de données. Ce projet se concentre sur la classification multimodale des produits en combinant données textuelles (titres, descriptions) et visuelles (images) afin d'améliorer la précision des recherches et recommandations. La tâche est complexe en raison de la diversité des produits, du bruit dans les données et du déséquilibre des classes.
Pour répondre à cette problématique, ce rapport aborde tout d'abord l'exploration des données textuelles et des images, puis décrit le prétraitement appliqué à ces données, avant de présenter nos trois approches de modélisation : basée sur le texte, sur l'image et multimodale.

II - Compréhension et Exploration des Données (EDA)

Dans le cadre de ce challenge, nous avons été fournis avec trois ensembles de données : un ensemble d'entraînement comprenant 86 % des observations (soit 84 916 échantillons), un ensemble de test représentant 14 % des observations (soit 13 812 échantillons), ainsi que les images associées à ces produits. Les ensembles d'entraînement et de test contiennent quatre variables principales : **"designation"** (Le titre du produit), **"description"** (la description du produit), **"productid"** (l'identifiant unique du produit, utilisé pour associer chaque article à son code de type dans le catalogue) et enfin **"imageid"** (l'identifiant unique de l'image associée à chaque produit).
Dans cette section, nous procéderons à une exploration approfondie de ces données afin de mieux comprendre leur structure et d'identifier les éventuels problèmes avant de les soumettre aux modèles d'apprentissage automatique.

a) Analyse des datasets d'entraînement et de test (X_train et X_test)

Analyse des variables textuelles

L'analyse des variables textuelles révèle plusieurs particularités intéressantes : la variable « description » de l'ensemble d'entraînement comporte 29 800 valeurs manquantes (soit environ un tiers des données), tandis que l'ensemble de test en compte 4 886 (17 % des observations). Ces absences pouvant nuire à l'entraînement, nous les avons imputées par des chaînes vides, marquant explicitement l'absence de description pour certains produits. Par ailleurs, l'étude des longueurs des titres et des descriptions met en évidence une grande hétérogénéité : certaines catégories (notamment 2705 et 2965) présentent des textes particulièrement longs, alors que d'autres (10 et 50) sont beaucoup plus courts. L'examen des boxplots (cf. figure X) révèle par ailleurs des valeurs aberrantes dans plusieurs catégories, dont l'influence potentielle sur la performance des modèles nécessitera un traitement adapté.

Analyse de la cible d'entraînement

La variable cible, prdtypecode, regroupe 27 catégories distinctes, mais sa distribution est fortement déséquilibrée (cf. figure X) : le code 10209 y apparaît 10 209 fois, alors que le code 1180 n'est présent que 764 fois. Ce déséquilibre peut induire un biais de prédiction, favorisant les classes majoritaires au détriment des moins fréquentes. Pour y remédier, nous avons envisagé des stratégies de rééquilibrage des classes, telles que la sur-échantillonnage ou la sous-échantillonnage, afin d'améliorer la robustesse et la

justesse des modèles.

b) Analyse des images

Les **images** associées aux produits ont une taille uniforme de 500 pixels et sont toutes au format **PNG**. Une analyse approfondie révèle que les **doublons** sont beaucoup plus fréquents dans l'ensemble d'entraînement, avec plusieurs empreintes MD5 apparaissant plus de dix fois, alors qu'ils restent relativement rares dans l'ensemble de test (cf. figure d). Cette redondance risque de biaiser l'apprentissage en fournissant au modèle des exemples « faciles » de manière disproportionnée : il est donc essentiel de **gérer ces doublons** — soit en les supprimant, soit en les traitant spécifiquement lors du prétraitement — afin d'éviter qu'ils ne faussent les résultats de l'entraînement.
Cette exploration des données met en lumière l'importance de traiter les valeurs manquantes, les valeurs aberrantes, les doublons et le déséquilibre des classes afin d'assurer des modèles fiables et sans biais. Ces éléments guideront la suite du processus.

III- Prétraitement des Données

a) Prétraitement des variables textuelles

Avant de soumettre les données textuelles à un modèle d'apprentissage automatique, plusieurs étapes de nettoyage et de transformation ont été appliquées pour normaliser et simplifier les informations. Voici les étapes du prétraitement que nous avons suivies : **suppression des balises HTML** (telles que `<p>`, `<b>`, etc.), **conversion des entités HTML** (comme `&#39;` pour l'apostrophe), **conversion en minuscules** (uniformisation du texte), **suppression des accents**, **suppression des caractères spéciaux**, **tokenisation** (découpage du texte en mots individuels), **suppression des mots vides (stopwords)**, **application du stemming** (réduction des mots à leur racine) et enfin **suppression des mots trop courts et trop fréquents**.
Grâce à ce prétraitement, nous pouvons observer, à travers l'analyse des nuages de mots de l'ensemble d'entraînement (cf. figure e), que les mots les plus fréquents pour les titres des produits sont : kit, piscine, figurine, taie d'oreiller, décoration, console de jeu, etc. Et pour les descriptions (cf. figure f), les termes les plus fréquents incluent : taie d'oreiller, léger, manuel, extérieur, piscine, etc.
Une fois les données prétraitées, nous avons effectué la **vectorisation**. Nous avons testé deux approches : une approche de base avec `TF-IDF` et une approche plus avancée avec `Word2Vec`. Chacune des méthodes a été réutilisé lors de la mdoélisation. Nous avons utilisé un `CNN 1D` combiné avec le TF-IDF pour la modélisation basé sur le texte et nous avons utilisé Word2Vec pour la fusion multimodale (EfficientNet-B0) .

b) Prétraitement des images

Avant toute modélisation, nous avons veillé à garantir la qualité et la diversité de notre jeu de données visuel. Dans un premier temps, chaque image a été soumise à un processus de déduplication basé sur le calcul de son empreinte MD5 : toutes les occurrences d'un même contenu visuel ont été détectées et un seul exemplaire a été conservé par groupe de doublons. Cette opération a permis d'éliminer les biais induits par des répétitions accidentelles, assurant ainsi que l'apprentissage ne soit pas faussé par des redondances.
Ensuite, afin de rendre les images compatibles avec les réseaux de neurones pré-entraînés,

nous avons standardisé leur taille à 224×224 pixels (interpolation bilinéaire) et appliqué une normalisation conforme aux statistiques d'ImageNet (moyenne : [0.485, 0.456, 0.406], écart-type : [0.229, 0.224, 0.225]). Cette mise à l'échelle et cette normalisation garantissent une convergence plus rapide et une meilleure stabilité d'entraînement sur les architectures de vision utilisées par la suite.

IV - Modélisation

Dans cette section, trois types de modélisation ont été explorés : basée sur le **texte**, basée sur les **images**, et une approche de **fusion multimodale** combinant les données textuelles et visuelles pour améliorer la classification des produits.

a) Modélisation basée sur le texte

Dans le cadre de la modélisation des données textuelles, deux modèles ont été déployés : un modèle de CNN 1D utilisé comme référence de base (baseline) et un modèle BERT pour tester une approche plus avancée.
Le CNN 1D (avec TF-IDF) a été choisi comme modèle de référence en raison de sa simplicité et de son efficacité pour traiter des séquences de texte, en exploitant les relations locales entre les mots et en extrayant des caractéristiques pertinentes pour la classification.
Le modèle BERT, quant à lui, est un modèle de pointe pour le traitement du langage naturel, reposant sur des mécanismes d'attention qui permettent de capturer des relations globales et complexes entre les mots, en prenant en compte le contexte complet des phrases.
Les résultats obtenus pour les deux modèles sont les suivants :

- F1 score du CNN 1D : 0,8441
- F1 score de BERT : 0,7512

Comparativement au benchmark du challenge (0,8113), le modèle CNN 1D a surpassé BERT en termes de F1 score, montrant ainsi qu'une approche plus simple, mais bien adaptée à la tâche, peut donner de bons résultats dans ce cas précis.

b) Modélisation basée sur les images

Pour évaluer la seule modalité visuelle, nous avons tout d'abord extrait des descripteurs à partir de deux architectures : EfficientNet-B0 (via Torchvision) et Swin-Tiny (via timm). Ces backbones ont été gelés et séparés en deux pipelines : d'une part, les features d'EfficientNet-B0 ont été moyennées spatialement pour obtenir un vecteur de dimension 1280, et d'autre part, les représentations de Swin-Tiny ont été réduites à un vecteur global en moyennant ses cartes de caractéristiques. Chaque vecteur a ensuite été injecté dans un perceptron multicouche (MLP) de deux couches cachées. Après cinq epochs d'entraînement, l'EfficientNet-MLP a atteint un F1-pondéré de 0.6106 (accuracy 62.1 %), illustrant la pertinence de ses embeddings visuels : il a su distinguer un grand nombre de catégories malgré leur imbrication. En revanche, le MLP sur les features Swin-Tiny s'est effondré sur une prédiction majoritaire, aboutissant à un F1-pondéré de seulement 0.0187 (accuracy 10.1 %). Ce résultat souligne que la configuration de Swin-Tiny, sans ajustement plus fin ou régularisation spécifique, n'a pas capturé la diversité des classes du jeu de données.

Le benchmark du challenge, quant à lui, est de 0.5534, ce qui met en perspective les résultats obtenus avec l'approche visuelle seule. Ces résultats montrent que **EfficientNet-B0** a bien performé, avec un F1-pondéré de 0.6106, surpassant ainsi le benchmark du challenge, et prouve l'efficacité de cette architecture pour capturer les caractéristiques visuelles des produits.

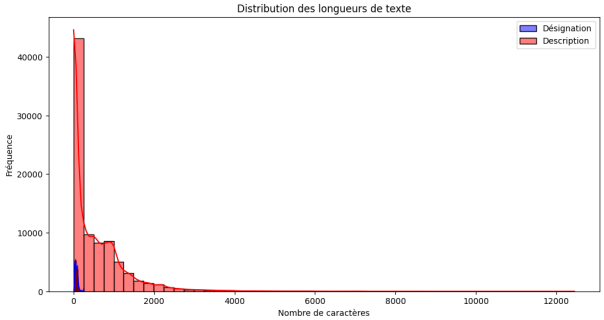
c) Fusion multimodale

Afin de tirer parti à la fois des informations visuelles et textuelles, nous avons exploré trois stratégies de fusion. Dans un premier temps, nous avons simplement concaténé le vecteur 1280-D issu d'EfficientNet-B0 avec le vecteur 300-D issu d'un Word2Vec entraîné sur les descriptions produits, puis fait passer l'ensemble dans un MLP à trois couches. En dix époques, cette approche « brute » a vu son F1-pondéré progresser de 0.7361 à 0.8521 (accuracy 85.2 %), démontrant la complémentarité puissante entre image et texte. La matrice de confusion associée (cf. annexe figure (e)) confirme une forte discrimination globale (diagonale majoritaire), tout en soulignant des confusions systématiques entre certaines catégories voisines : les classes 6 et 8 présentent un rappel particulièrement bas, ce qui oriente vers un sur-échantillonnage ou un rééquilibrage ciblé.
Nous avons ensuite introduit un module d'attention cross-modal : image et texte sont d'abord projetés en deux « tokens » de même dimension, traités par un bloc `MultiHeadAttention`, puis agrégés avant la classification. Cette légère sophistication a permis un léger gain, avec un F1-pondéré culminant à 0.8593, suggérant que l'attention modélise mieux les interactions fines entre modalités.
Enfin, nous avons testé la **Focal Loss** ($\gamma = 2$) pour accentuer l'apprentissage sur les classes rares : malgré une amélioration marginale pour ces dernières, le F1-pondéré global s'est stabilisé autour de 0.8341, inférieur au MLP de base. Cette contre-performance indique que le déséquilibre des classes n'était pas le principal frein, et que la focalisation sur les exemples minoritaires s'est faite au détriment de la précision des classes majoritaires.
La fusion multimodale par simple concaténation et MLP constitue un socle robuste (F1 0.8544), et l'ajout d'un mécanisme d'attention apporte un léger surcroît de performance (+0.0024). En revanche, la Focal Loss, bien qu'intéressante pour les classes rares, ne s'est pas traduite en gain global ici.

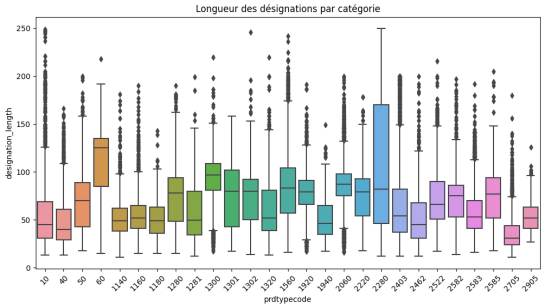
V - Conclusion

Notre étude met en évidence que seules, les informations textuelles surpassent nettement les seules images, ces dernières souffrant de performances limitées même avec des architectures lourdes. En revanche, la fusion multimodale, qui aligne puis combine les représentations visuelles et textuelles dans un perceptron enrichi par un mécanisme d'attention cross-modal, se révèle la plus performante, illustrant la force de la complémentarité des deux sources. L'analyse de la matrice de confusion souligne néanmoins que certaines catégories demeurent difficiles à discerner, ce qui invite à tester des approches d'oversampling ciblé ou de rééquilibrage, ainsi qu'à envisager l'intégration de modèles multimodaux plus avancés (ViLBERT, MMBT) pour renforcer la synergie image–texte.

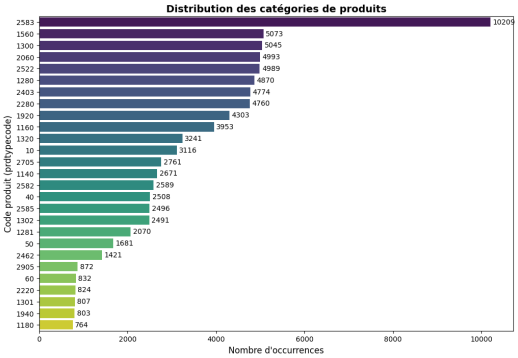
VI - Annexe



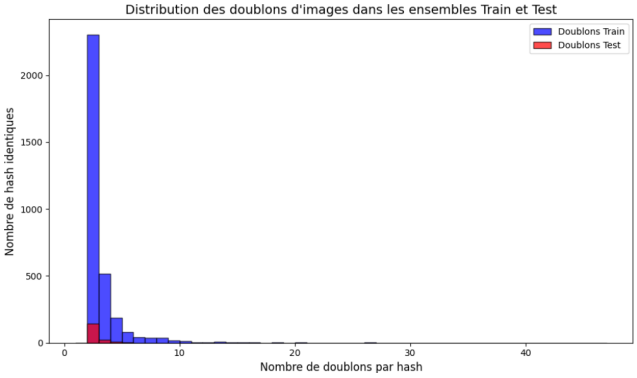
(a) Distribution des longueurs de texte (X_train)



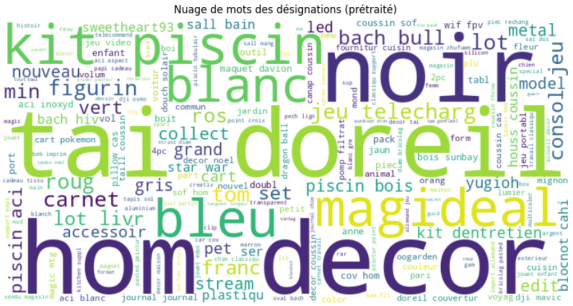
(b) Longueur des designations par catégorie (X_train)



(a) Distribution des codes types de produits (y_train)

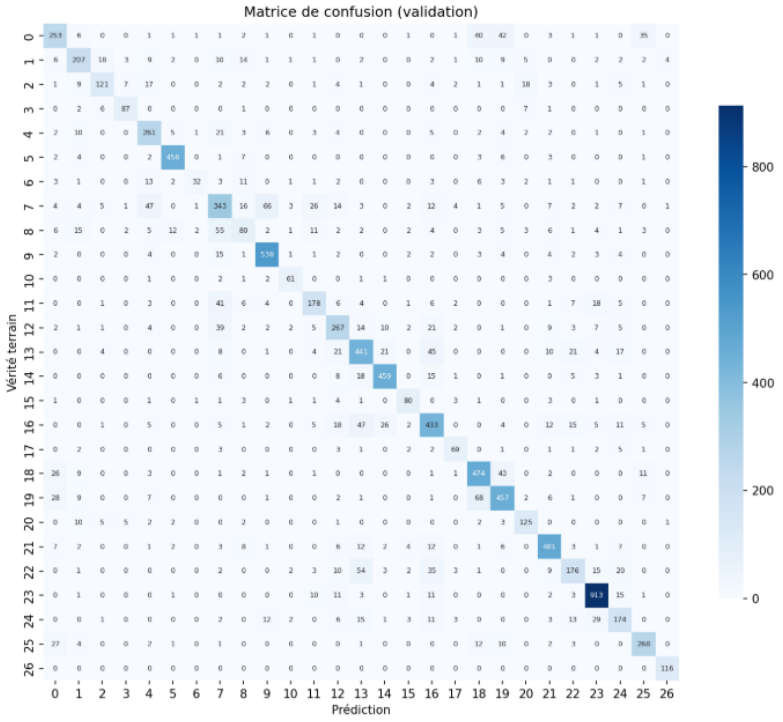


(b) Nombre de doublon par hash

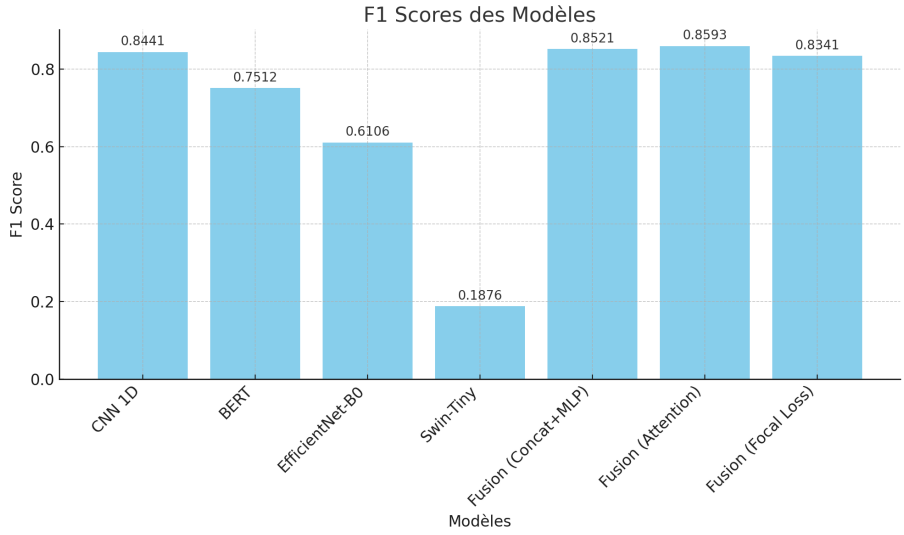


(c) Nuage de mots des designations

(d) Nuage de mots des descriptions



(e) Matrice de confusion



(f) Résumé des F1 Scores de nos modèles