

DATASCIENCE

CLASSIFICATION DES PRODUITS E COMMERCE

RAKUTEN

RAPPORT FINAL

Saliou Diedhiou, Eren Ustundag, Luc Jamet, Leila Ruiz



Table des matières

Contexte.....	4
Objectifs.....	4
Cadre.....	5
Pertinence.....	6
Particularités et limitations.....	6
Preprocessing.....	7
Labellisation.....	7
Langues multiples.....	7
Quelques exemples.....	7
Visualisations et Statistiques.....	11
Word Cloud par Classe.....	12
Panel d'images.....	13
Distribution des classes.....	14
Distribution des classes et des valeurs manquantes.....	15
Distribution du nombre de caractères désignatifs selon la catégorie.....	16
Taux de produits par catégorie.....	18
Conclusion.....	19
Classification du problème.....	20
Choix du modèle et optimisation.....	21
Classification de texte : Le Text Mining.....	21
Pré-traitement des données.....	22
Les modèles retenus pour la classification de textes.....	23
Entraînement des modèles retenus.....	24
Performance des modèles à l'entraînement.....	24
Evaluation des modèles retenus.....	26
Classification d'images : Les modèles de Deep Learning.....	27
Pré-traitement des données.....	29
Les modèles retenus pour la classification d'images.....	29
Entraînement des modèles retenus.....	31
Performance des modèles à l'entraînement.....	32
Evaluation des modèles retenus.....	33
Techniques d'optimisation des paramètres :.....	33
Techniques d'interprétabilité :.....	34
Amélioration des performances :.....	34

Classification à plusieurs modèles.....	36
Système de vote.....	36
Conclusion.....	40
Difficultés rencontrées.....	40
Bilan.....	41
Suite du projet.....	42
Bibliographie.....	43

Contexte

La classification des produits par la catégorisation des textes et des images est un sujet fondamental pour tout marché du e-commerce, car il permet une mise en application dans la recherche et la recommandation personnalisée. Ce sujet a un intérêt tout particulier pour les entreprises de e-commerce comme Rakuten qui ont des difficultés à catégoriser les produits à partir des images et des textes fournis par différents fournisseurs à la fois professionnels et non professionnels et à éviter les doublons. La problématique présente plusieurs aspects de recherche intéressants en raison de la nature très aléatoire des textes et des images produits et de la distribution très déséquilibrée des données.

L'importance de cette classification réside dans sa capacité à améliorer l'expérience utilisateur en facilitant la navigation et la découverte de produits pertinents. En outre, une classification précise des produits permet aux entreprises de mieux comprendre les préférences des clients et d'adapter leur offre en conséquence.

Du point de vue économique, l'amélioration de la classification des produits peut entraîner une augmentation des ventes, une meilleure satisfaction des clients et une réduction des coûts liés aux retours de produits. En investissant dans le développement et l'optimisation des modèles de classification, les entreprises de e-commerce peuvent espérer un retour sur investissement significatif grâce à une meilleure compréhension des préférences des clients et à des recommandations personnalisées plus pertinentes.

Objectifs

L'objectif de ce projet est la classification multimodale à grande échelle (textes et images) des données produits en codes type de produit.

La multimodalité à grande échelle fait référence à l'intégration et à l'analyse de données provenant de sources et de modes multiples, tels que le texte, l'audio, les images et la vidéo, à grande échelle. Il s'agit de traiter et de comprendre des données provenant de modalités multiples afin d'en tirer des enseignements et de prendre des décisions.

L'analyse multimodale à grande échelle peut être appliquée à un large éventail de domaines, tels que la vision par ordinateur, le traitement du langage naturel, la reconnaissance vocale et la robotique. Elle peut être utilisée pour développer des systèmes intelligents capables de reconnaître le langage naturel et d'y répondre, d'interpréter des images et des vidéos, et même de contrôler des objets physiques dans le monde réel.

Dans le cadre du projet Rakuten, l'objectif est de modéliser un classificateur pour catégoriser chacun des produits en fonction des différentes données disponibles.

Cadre

Les données utilisées sont mises à disposition dans le cadre du challenge de **Rakuten France Multimodal Product Data Classification** organisé par Rakuten Institute of technology de Paris.

Les données sont constituées de trois fichiers csv contenant les désignations et descriptions de produits ainsi que d'une banque d'images les représentant. Chaque observation comporte une image, une désignation et optionnellement une description plus détaillée.

Les fichiers sont décomposés en jeu d'entraînement et de test. Les jeux de données disponibles pour l'entraînement portent sur 84916 produits (deux fichiers csv). Les jeux de données disponibles pour le test portent sur 13812 produits (un fichier). Toutes les images ont pour résolution 500x500 pixels et sont en couleur.

Les données sont divisées selon deux critères, formant quatre ensembles distincts : formation ou test, entrée ou sortie.

- X_train_update.csv : fichier d'entrée d'entraînement (variables explicatives, colonnes techniques pour faire le lien avec les images)
- Y_train_CVw08PX.csv : fichier de sortie d'apprentissage (variable cible)
- X_test_update.csv : fichier d'entrée de test (variables explicatives et colonnes techniques, pas de variable cible)

Un fichier images.zip contenant toutes les images est également fourni. En décompressant ce fichier, on obtient un dossier nommé images avec deux sous-dossiers nommés image_train et image_test, contenant respectivement les images d'entraînement et de test.

Pertinence

Les données textuelles comportent peu de variables, celles qui sont pertinentes étant la désignation et la description du produit ainsi que son image. Le jeu de données ne comporte aucun doublon, et tous les articles ont un code image identifiant faisant référence à l'image associée.

La variable cible est un code type produit faisant office de catégorie, regroupant un ensemble de produits répondant aux mêmes caractéristiques.

Particularités et limitations

Le code type produit est un identifiant purement technique et ne permet pas à première vue de nommer la catégorie correspondante.

Les désignations et descriptions comportent des langues multiples, des balises HTML et des caractères encodés. Les descriptions sont absentes d'environ 35% des fiches produits, contrairement aux désignations qui sont systématiquement disponibles.

Concernant les images, certaines comportent des cadres et marges qui pourraient rendre les prochaines étapes d'analyse difficiles.

Preprocessing

Après examen du dataframe, nous avons remarqué que la part des données manquantes sur les descriptions était importante. Nous avons décidé de conserver la variable « description », car elle est pertinente pour nos analyses, mais de la regrouper avec la variable « désignation ». Cette décision nous permettra de poursuivre nos analyses avec une meilleure compréhension des données.

Nous avons également détecté la présence de balises HTML que nous avons retirées et de caractères encodés que nous avons redressés.

Labellisation

Les codes types de produits n'étant pas très lisibles, nous avons retravaillé la labellisation des catégories produits.

Langues multiples

Nous avons pu constater la présence de différentes langues, nous avons prévu d'ajouter une colonne permettant la traduction des descriptions. Pour cela, nous avons gardé l'ensemble des données, quelle que soit la longueur de la description et la désignation associée. Dans certains cas, la description manquait. Nous avons tout de même traité l'intégralité des données du set d'entraînement.

Cette phase de traduction assure une meilleure cohérence des données d'entrée en assurant qu'un même mot soit présent sous la même forme à travers les observations.

Quelques exemples

Pour la partie “entraînement”, nous partons de deux fichiers, dont le premier, qui contient les variables explicatives, commence par ces quelques lignes que nous présentons ici sous forme tabulaire (la description est abrégée) :

	designation	description	productid	imageid
0	Olivia: Personalisiertes Notizbuch / 150 Seiten / Punktraster / Ca Din A5 / Rosen-Design		3804725264	1263597046
1	Journal Des Arts (Le) N° 133 Du 28/09/2001 - L'art Et Son Marche Salon D'art Asiatique A Paris - Jacques Barrere - Francois Perrier - La Reforme Des Ventes Aux Encheres Publiques - Le Sna Fete Ses Cent Ans.		436067568	1008141237
2	Grand Stylet Ergonomique Bleu Gamepad Nintendo Wii U - Speedlink Pilot Style	"PILOT STYLE Touch Pen de marque Speedlink est 1 stylet ergonomique pour GamePad Nintendo Wii U. Pour un confort optimal et une précision maximale sur le GamePad de la Wii U (...)	201115110	938777978

Dans ce premier fichier, la 1ère colonne est anonyme et contient des numéros qui s'incrémentent à partir de 0. Il s'agit donc d'une simple colonne d'index que nous chargeons comme telle. Les 4e et 5e colonnes sont purement techniques et permettent de faire le lien entre chaque produit et son image.

Le second fichier d'entraînement, lui, est porteur des mêmes index et d'un code représentant le type de produit, lequel représente la variable cible :

	prdtypecode
0	10
1	2280
2	50

La jonction entre les data frames issus de ces deux fichiers peut se faire sur la colonne des index.

Pour illustrer les étapes de preprocessing nécessaires, voici un échantillon de 4 lignes ; seuls les index, les variables explicatives et la variable cible sont montrées ici :

index	designation	description	prdtypecode
0	Olivia: Personalisiertes Notizbuch / 150 Seiten / Punktraster / Ca Din A5 / Rosen-Design		10
2	Grand Stylet Ergonomique Bleu Gamepad Nintendo Wii U - Speedlink Pilot Style	PILOT STYLE Touch Pen de marque Speedlink est 1 stylet ergonomique pour GamePad Nintendo Wii U. Pour un confort optimal et une précision maximale sur le GamePad de la Wii U: ce grand stylet hautement ergonomique est non seulement parfaitement adapté à votre main mais aussi très élégant. Il est livré avec un support qui se fixe sans adhésif à l'arrière du GamePad Caractéristiques: Modèle: Speedlink PILOT STYLE Touch Pen Couleur: Bleu Ref. Fabricant: SL-3468-BE Compatibilité: GamePad Nintendo Wii U Forme particulièrement ergonomique excellente tenue en main Pointe à revêtement longue durée conçue pour ne pas abîmer l'écran tactile En bonus : Support inclus pour GamePad 	50
4	La Guerre Des Tuques	Luc a des idées de grandeur. Il veut organiser un jeu de guerre de boules de neige et s'arranger pour en être le vainqueur incontestable. Mais Sophie s'en mêle et chambarde tous ses plans...	2705
6	Christof E: Bildungsprozesse n Auf Der Spur		10

8	Puzzle Scooby-Doo Avec Poster 2x35 Pieces		1280
----------	--	--	------

Après fusion des colonnes "designation" et "description" en une seule colonne "design_describe", retrait des balises HTML, redressement des caractères et nommage des types de produit dans une colonne "Labels" :

index	design_describe	Labels
0	Olivia: Personalisiertes Notizbuch / 150 Seiten / Punktraster / Ca Din A5 / Rosen-Design	Livres occasions
2	Grand Stylet Ergonomique Bleu Gamepad Nintendo Wii U - Speedlink Pilot Style PILOT STYLE Touch Pen de marque Speedlink est 1 stylet ergonomique pour GamePad Nintendo Wii U. Pour un confort optimal et une precision maximale sur le GamePad de la Wii U: ce grand stylet hautement ergonomique est non seulement parfaitement adapte a votre main mais aussi tres elegant. Il est livre avec un support qui se fixe sans adhesif a l'arriere du GamePad Caracteristiques: Modele: Speedlink PILOT STYLE Touch Pen Couleur: Bleu Ref. Fabricant: SL-3468-BE Compatibilite: GamePad Nintendo Wii U Forme particulierement ergonomique excellente tenue en main Pointe a revetement longue duree concue pour ne pas abimer l'écran tactile En bonus : Support inclu pour GamePad	Loisirs intérieur
4	La Guerre Des Tuques Luc a des idées de grandeur. Il veut organiser un jeu de guerre de boules de neige et s'arranger pour en être le vainqueur incontesté. Mais Sophie s'en mêle et chambarde tous ses plans...	Livres
6	Christof E: Bildungsprozessen Auf Der Spur	Livres occasions
8	Puzzle Scooby-Doo Avec Poster 2x35 Pieces	Jouets

Et enfin, après traduction des désignations et descriptions dans une nouvelle colonne "designe_decrit" :

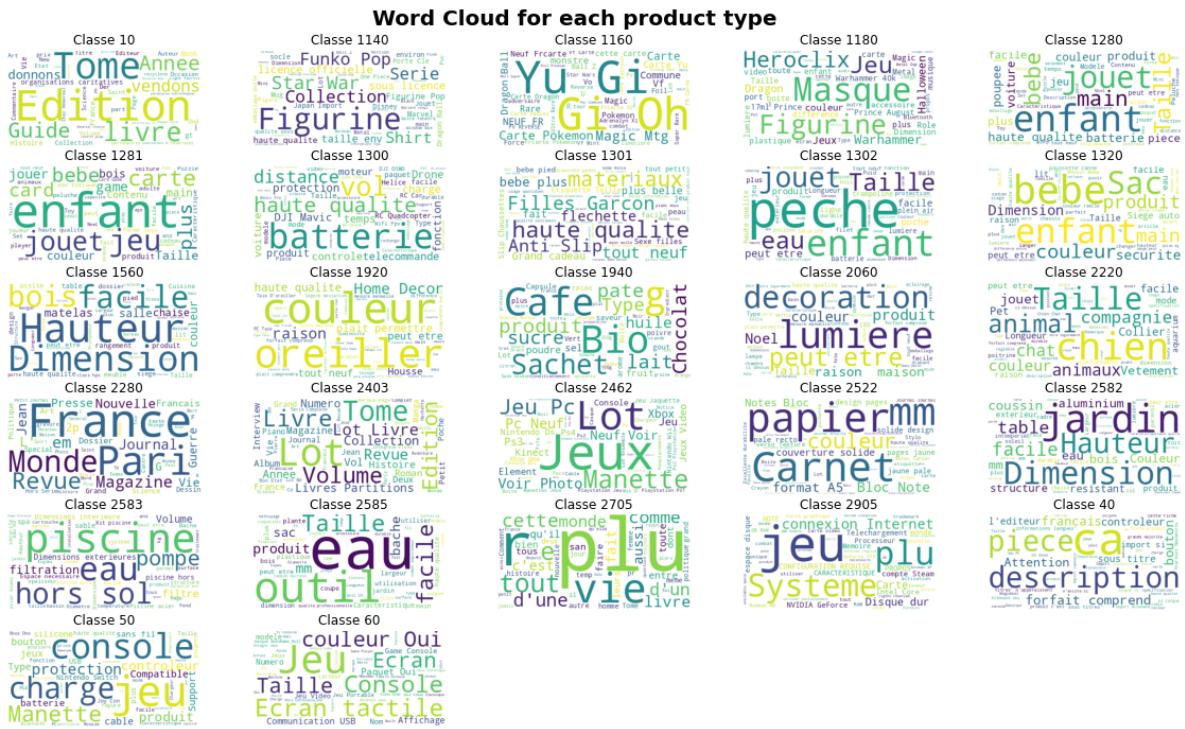
index	designe_decrit	Labels
0	Olivia : Carnet Personnalisé / 150 Pages / Dot Grid / Ca Din A5 / Rose Design	Livres occasions
2	Grand Stylet Ergonomique Bleu Gamepad Nintendo Wii U - Speedlink Pilot Style PILOT STYLE Touch Pen de marque Speedlink est 1 stylet ergonomique pour GamePad Nintendo Wii U. Pour un confort optimal et une precision maximale sur le GamePad de la Wii U: ce grand stylet hautement ergonomique est non seulement parfaitement adapte a votre main mais aussi tres elegant. Il est livre avec un support qui se fixe sans adhesif a l'arriere du GamePad Caracteristiques: Modele: Speedlink PILOT STYLE Touch Pen Couleur: Bleu Ref. Fabricant: SL-3468-BE Compatibilite: GamePad Nintendo Wii U Forme particulierement ergonomique excellente tenue en main Pointe a revetement longue duree concue pour ne pas abimer l'écran tactile En bonus : Support inclu pour GamePad	Loisirs intérieur
4	La Guerre Des Tuques Luc a des idées de grandeur. Il veut organiser un jeu de guerre de boules de neige et s'arranger pour en être le vainqueur incontesté. Mais Sophie s'en mêle et chambarde tous ses plans...	Livres
6	Christof E: Processus éducatifs sur le Spur	Livres occasions
8	Puzzle Scooby-Doo Avec Poster 2x35 Pieces	Jouets

Visualisations et Statistiques

N'ayant que très peu de variables dans notre dataframe, nous avons axé l'effort de nos analyses sur la qualité du contenu de chacun d'entre eux. sur la particularité de certaines valeurs de nos variables explicatives et nous avons analysé la distribution de nos données.

Word Cloud par Classe

Le nuage de mots-clés, ou nuage de tags (en anglais tag cloud, word cloud ou keyword cloud) est une représentation visuelle des mots-clés (tags) i.e. les plus représentés. Concrètement, plus un mot-clé est cité, plus il apparaît dans une grande taille dans le nuage de mots. Il demande cependant à retirer la présence de certains mots récurrents qui ne sont pas porteurs de sens (et, de, ce, cette, du...) nommés **“mots vides”**. Nous faisons appel à la bibliothèque stopword pour réaliser une exclusion de ces mots avant l'exécution du wordcloud.

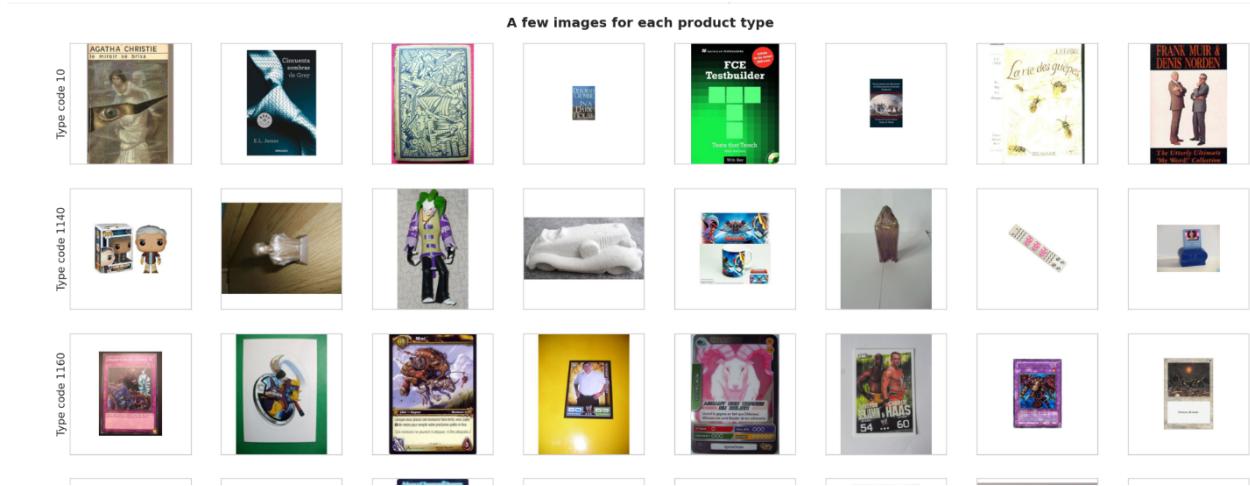


Interprétation :

Cette représentation par word cloud permet de mettre en évidence les mots les plus fréquents au sein de chaque catégorie, et donc de définir la correspondance des classes de produits, il a demandé pour certains cas une utilisation conjointe d'un panel d'images pour venir compléter l'analyse. Parmi les mots les plus récurrents, on en retrouve certains comme "eau", "piscine", "taille", etc., en lien avec la catégorie '2583' la plus représentée : "Accessoires piscine et spa".

Panel d'images

Ce graphique permet d'afficher un échantillonnage d'images associées à chacun des codes type de produit cible (visible ligne par ligne sur la gauche).

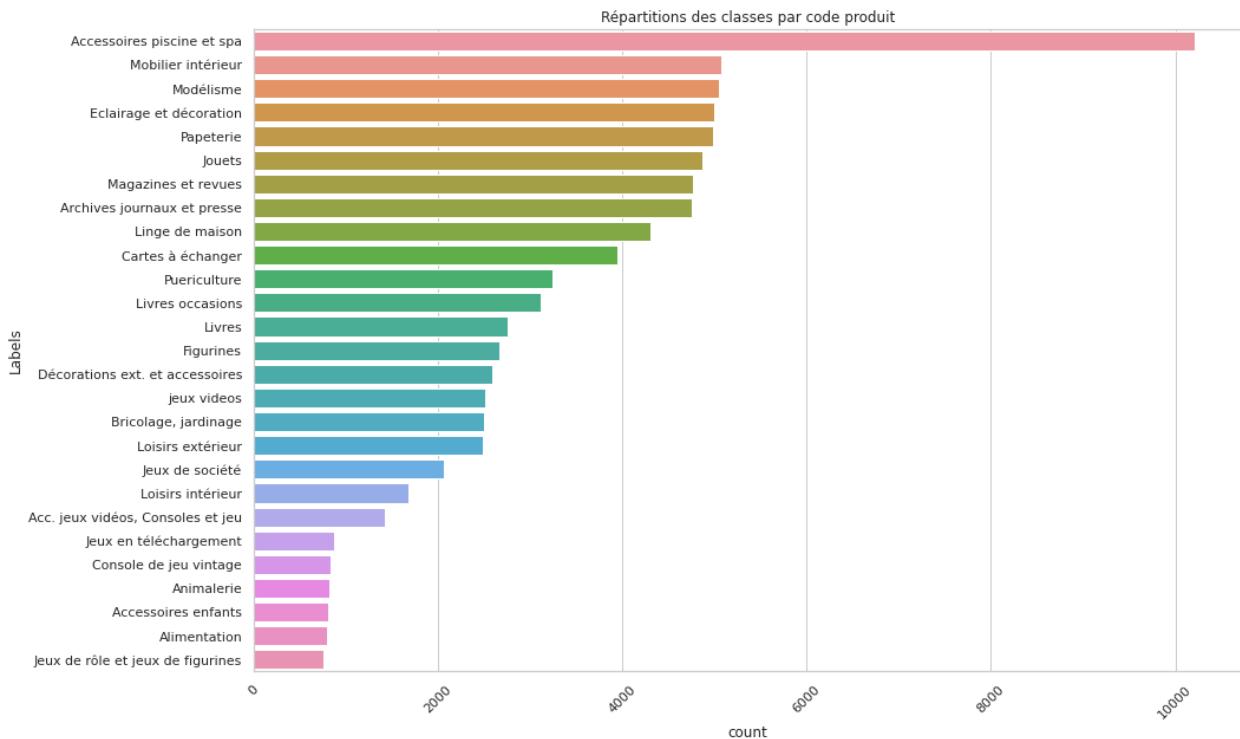


Interprétation :

Cette représentation visuelle permet de définir la correspondance des classes de produits conjointement avec les word cloud par classe. Certaines catégories comme la catégorie 10, Livre d'occasion et 1301, Accessoires enfants, sont difficilement classifiables. On peut supposer que la prédition de classification des images sera plus difficile à interpréter sur ces classes lors de la seconde phase de notre analyse.

Distribution des classes

Ce graphique met en évidence les catégories les plus représentées en volume de produits.

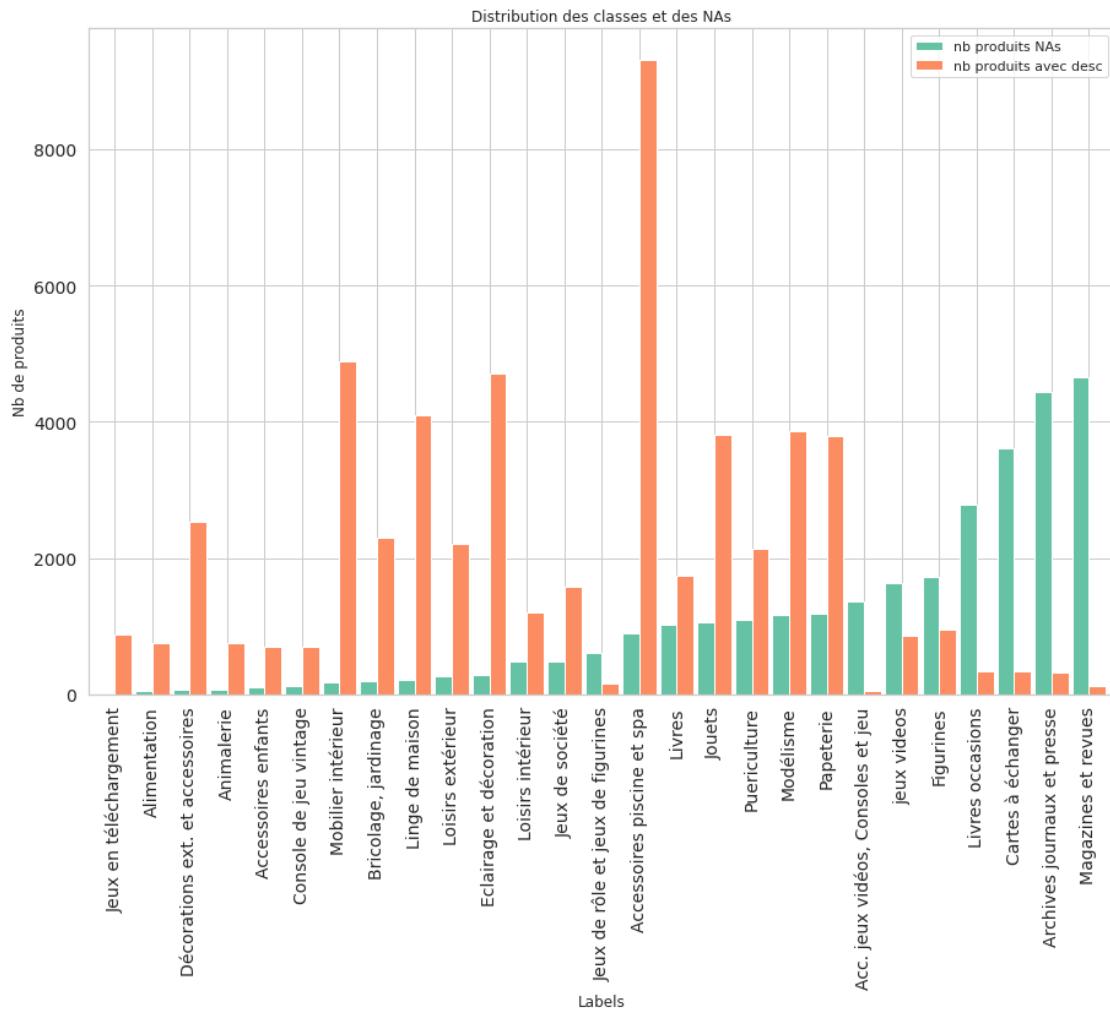


Interprétation :

On constate un déséquilibre prononcé dans la représentation des différents types de produit. Le type n°2853 - "Accessoires piscine et spa", le plus représenté, compte environ 13 fois plus de produits que le type le moins représenté (n°1180 - "Jeux de rôle et jeux de figurines").

Distribution des classes et des valeurs manquantes

Cette figure permet de visualiser la distribution du nombre de produits avec ou sans description.



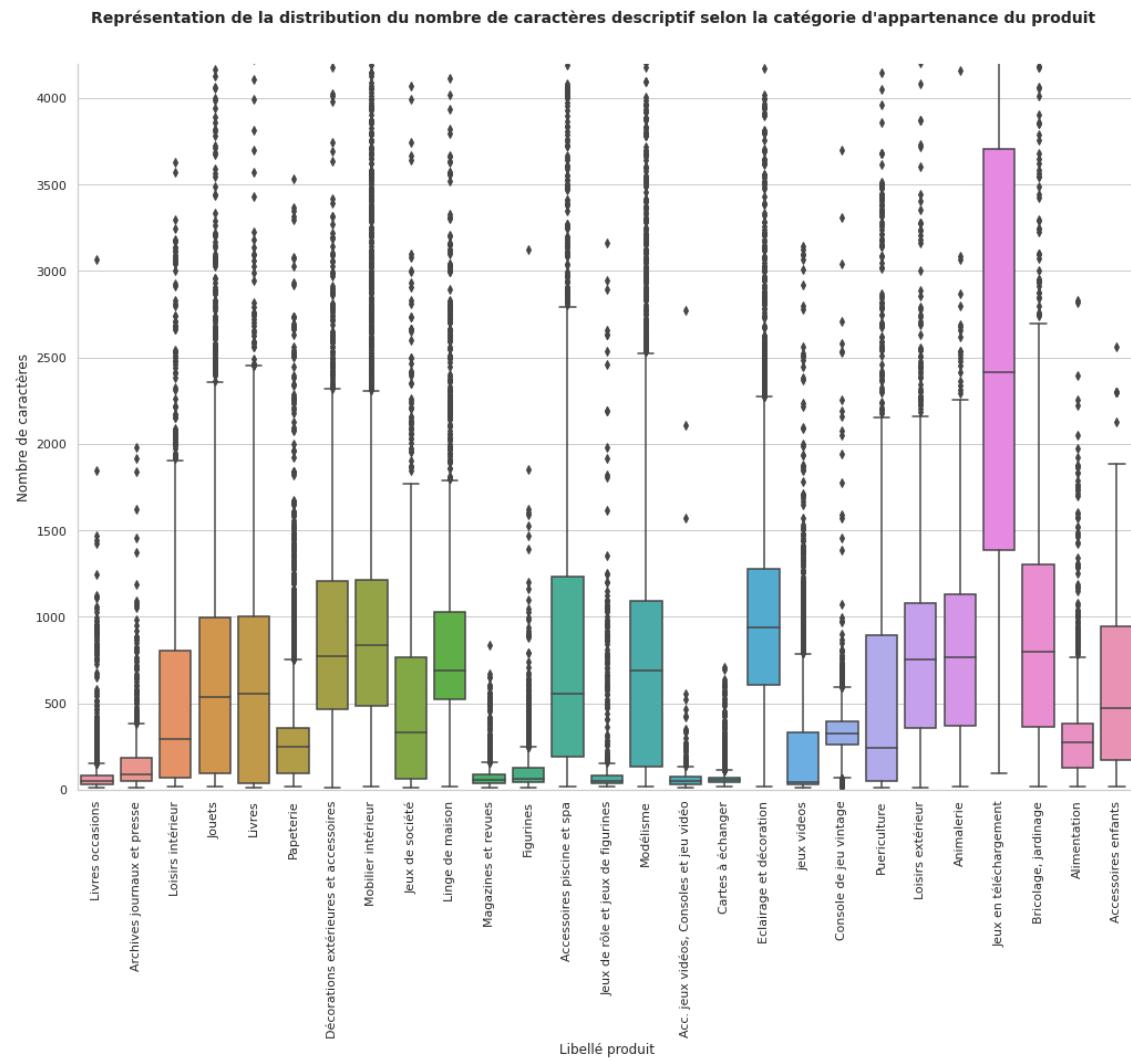
Interprétation :

Il est clair que certaines classes de produits sont plus touchées que d'autres par les descriptions manquantes.

Peut-être que certains produits sont davantage vendus par des vendeurs indépendants qui ne prennent pas le temps de saisir une description détaillée (exemples des catégories Livres d'occasion, Magazines et revues, Cartes à échanger). Souvent, l'essentiel des informations se trouve dans le champ 'désignation'.

Distribution du nombre de caractères désinatifs selon la catégorie

Cette figure permet de visualiser la distribution du nombre de caractères descriptifs selon la catégorie d'appartenance du produit.



Interprétation :

On constate que certaines classes ont moins de contenus descriptifs. A titre indicatif, les catégories de produits les moins décrites, sont :

- 10 - Des livres d'occasion
- 2403 - Des magazines et revues
- 1140 - Des figurines

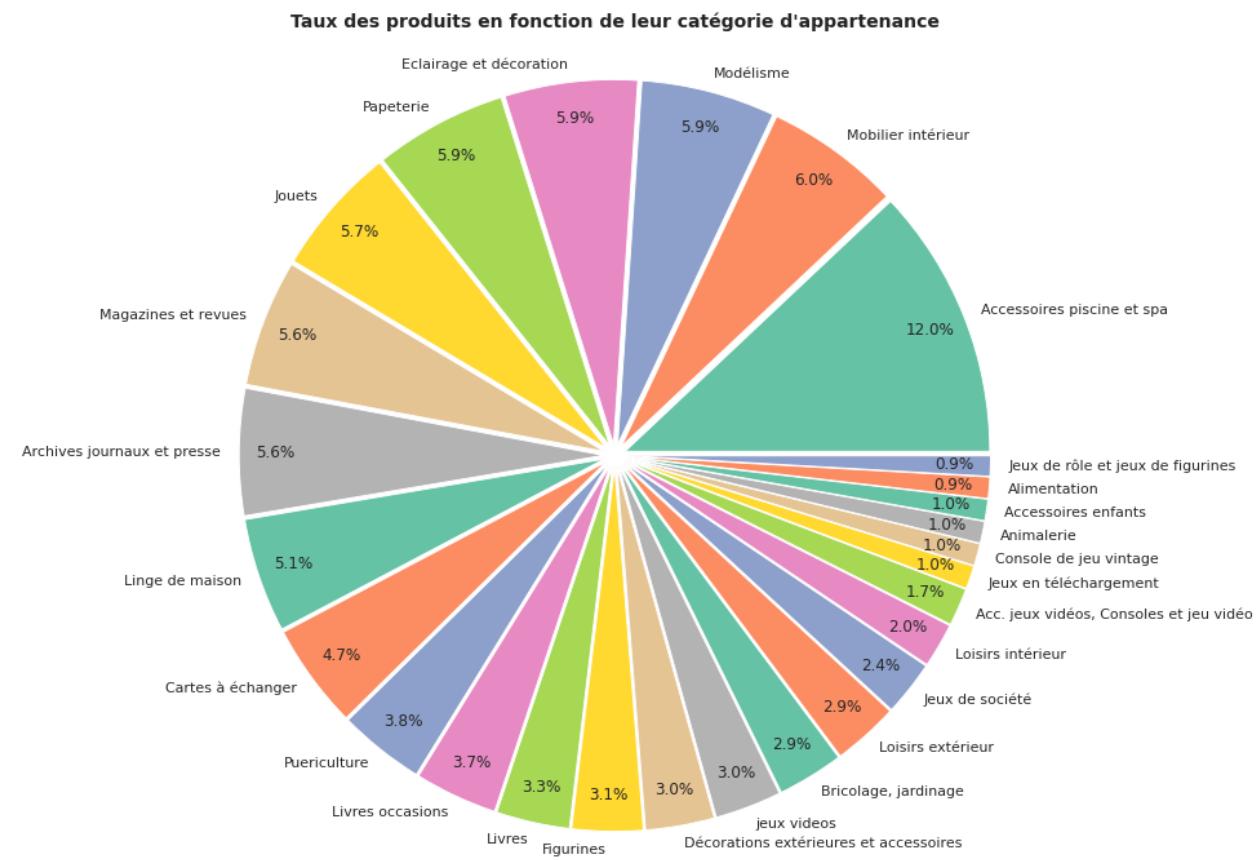
-
- 1180 - Jeux de rôle et jeux de figurines
 - 2462 - Acc. jeux vidéos, Consoles et jeu
 - 1160 - Cartes à échanger

Et la catégorie de produits la plus décrite est :

- 2905 - Des jeux en téléchargement

Taux de produits par catégorie

Cette figure permet de visualiser le taux des produits en fonction de leur catégorie d'appartenance.



Interprétation :

Ce graphique représente la distribution du nombre de produits par catégorie sous une forme différente de l'affichage en barres horizontales. Nous le montrons comme rappel, pour comparaison avec l'analyse du volume de données textuelles disponibles.

Nous pouvons remarquer que les produits de la catégorie ayant le plus de descriptions représentent 1% des produits de cette base de données.

D'ailleurs, la catégorie ayant le plus de produit dans son répertoire est la 2583 (piscines et accessoires) avec 12% des éléments de cette base de données. Cette dernière catégorie

semble présenter une description de longueur standard par rapport à l'ensemble des catégories observées.

Conclusion

Sur la base de ces différentes analyses graphiques, nous pouvons conclure que certaines catégories sont mieux représentées que d'autres, d'une part en nombre de produits, d'autre part en richesse de description. Il faudra porter une attention particulière aux biais d'entraînement. Un risque important est de classifier certains produits dans la mauvaise catégorie, tout particulièrement ceux dont la catégorie réelle est peu représentée et ceux décrits en peu de mots. Pour y remédier, il sera possible d'utiliser un F1-score pondéré par la population des différentes catégories.

Il est aussi ressorti que l'image de certains produits ne permettait pas d'identifier clairement leur catégorie. Il est donc possible que l'apprentissage à partir des images aboutisse lui aussi à des erreurs importantes de classification pour certains types de produit.

Enfin, il sera important de croiser les apprentissages sur les textes et sur les images pour obtenir un résultat d'ensemble cohérent.

Classification du problème

Nous sommes confrontés dans le cadre de ce projet à un problème de classification. Celui-ci consiste à utiliser des techniques d'apprentissage automatique pour classer des produits à l'aide des textes et des images qui les accompagnent.

La principale métrique de performance utilisée pour évaluer les modèles est l'accuracy (précision). Cette métrique mesure la proportion de prédictions correctes par rapport à l'ensemble des échantillons. L'accuracy est couramment utilisée pour évaluer les performances globales d'un modèle de classification.

En plus de l'accuracy, nous avons également utilisé deux autres métriques dans notre analyse :

La Balanced Accuracy (accuracy équilibrée) : Cette métrique permet d'évaluer la performance d'un modèle lorsque les classes sont déséquilibrées. Elle calcule la moyenne des rappels (recalls) par classe, en prenant en compte le déséquilibre de la distribution des classes. Ainsi, elle fournit une mesure plus robuste lorsque certaines classes sont moins représentées que d'autres.

Le F1-score : Le F1-score est une métrique qui combine à la fois la précision et le rappel pour évaluer la performance d'un modèle de classification. Cette métrique est particulièrement utile lorsque les classes sont déséquilibrées. Le F1-score est calculé comme la moyenne harmonique de la précision et du rappel.

L'utilisation de ces métriques complémentaires, en plus de l'accuracy, nous permet d'avoir une évaluation plus approfondie des performances des modèles. Elles prennent en compte le déséquilibre des classes et évaluent à la fois la précision et le rappel. Cela nous offre une vision plus complète de la capacité du modèle à prédire avec précision les différentes classes du problème de classification.

Choix du modèle et optimisation

Classification de texte : Le Text Mining

Plusieurs algorithmes de classification ont été explorés dans le cadre de ce projet, notamment :

- TF-IDF avec K-nearest neighbors (KNN)
- TF-IDF avec Random Forest (RF)
- TF-IDF avec Stochastic Gradient Descent (SGD)
- Word2Vec
- FastText
- CamemBERT

Le K-nearest neighbors, Random Forest et Stochastic Gradient Descent sont des modèles classiques de classification. Leur entraînement peut être modulé par des hyperparamètres dont nous avons pris soin d'explorer les effets sur les performances et le surapprentissage à l'aide de grilles de recherche à validation croisée.

En revanche, FastText et Word2Vec sont des approches plus récentes basées sur la représentation vectorielle des mots.

FastText est un modèle d'apprentissage de représentations de mots et de phrases qui repose sur la méthode des sacs de mots (bag-of-words) et des n-grammes de caractères. Développé par Facebook AI Research, FastText est particulièrement adapté à la classification de textes de grande taille. Il utilise une représentation vectorielle des mots appelée "embedding", où chaque mot est représenté par un vecteur dense de nombres réels. Les vecteurs FastText peuvent capturer les similarités sémantiques entre les mots, ce qui les rend utiles comme fonctionnalités d'entrée pour des tâches de classification ou de recherche d'informations. De plus, FastText propose une implémentation optimisée qui permet un entraînement rapide, même sur de grands corpus de textes.

Word2Vec, développé par Google, est un autre modèle populaire d'apprentissage de représentations de mots. Il utilise des réseaux de neurones pour apprendre des

représentations continues de mots à partir de grands corpus de textes. Les vecteurs Word2Vec sont obtenus en optimisant un objectif de prédiction de contexte, où le modèle tente de prédire les mots environnants d'un mot donné. Ces vecteurs sont utilisés pour mesurer la similarité sémantique entre les mots, effectuer des opérations vectorielles sur les mots et améliorer les performances des modèles de classification de texte.

CamemBERT, développé par l'équipe de recherche de Hugging Face, est un modèle de traitement du langage naturel spécifiquement adapté à la langue française. Basé sur l'architecture du transformer, CamemBERT est pré-entraîné sur des corpus de textes français en utilisant des tâches d'apprentissage automatique telles que la prédiction de mots masqués et la prédiction de la phrase suivante. Il est capable de capturer les relations sémantiques et syntaxiques entre les mots et les phrases grâce à des mécanismes d'attention et une représentation contextuelle des mots. Le tokenizer CamemBERT segmente les phrases en sous-unités linguistiques (tokens) pour une représentation plus fine des textes français, en tenant compte des spécificités telles que les liaisons entre les mots.

Dans le cadre de ce projet, le modèle TextCNN, qui est un modèle de classification de texte basé sur les réseaux de convolution, a été utilisé en combinaison avec CamemBERT. TextCNN utilise des couches de convolutions unidimensionnelles pour extraire des caractéristiques locales des séquences de mots, puis applique des opérations de pooling pour agréger les informations les plus importantes et réduire la dimensionnalité des représentations. Cette combinaison permet de bénéficier à la fois de l'apprentissage de représentations contextuelles fines de CamemBERT et des capacités de classification du modèle TextCNN.

Pré-traitement des données

Avant d'appliquer les algorithmes de classification, un processus de prétraitement des données a été effectué pour garantir la qualité et la cohérence des informations utilisées. Ce prétraitement comprend plusieurs étapes essentielles. Tout d'abord, une étape de nettoyage a été réalisée en amont pour éliminer les données indésirables telles que les caractères spéciaux, la ponctuation et les symboles. En outre, certains textes étant rédigés en langue étrangère, nous les avons traduits automatiquement en français. Ensuite, les données ont été normalisées en convertissant tous les textes en minuscules afin d'éviter

les variations de casse qui pourraient affecter la performance des modèles de classification. Enfin, les stop words, tels que "le", "la", "et", ont été supprimés car ils n'apportent pas de valeur informative significative à certains modèles de classification. Ce processus de prétraitement des données garantit que les informations utilisées par les algorithmes de classification sont cohérentes, normalisées et débarrassées de bruit ou d'éléments redondants.

Cependant, les stop words en français pouvant contenir des informations pertinentes pour la compréhension et la représentation du texte, leur suppression n'a pas été effectuée pour la modélisation avec CamemBERT afin de préserver l'intégrité des données et d'exploiter au mieux les informations linguistiques contenues dans le texte.

Les modèles retenus pour la classification de textes

Après avoir exploré plusieurs modèles classiques et optimisé leurs hyperparamètres à l'aide d'une recherche par grille (Grid Search), nous avons retenu les modèles Random Forest et Stochastic Gradient Descent en raison de leurs performances comparatives. Le modèle K-Nearest-Neighbors (KNN), lui, a donné des résultats sensiblement inférieurs pour des temps de calcul beaucoup plus grands.

Après utilisation d'une vectorisation TF-IDF et calage des hyperparamètres, nous avons obtenu les准确acies pondérées suivantes :

- Random Forest : 73% sur les données d'entraînement et 70% sur les données de test
- Stochastic Gradient Descent : 78% sur les données d'entraînement et 73% sur les données de test

Nous avons ensuite combiné Word2Vec avec Random Forest et Stochastic Gradient Descent, ce qui a permis d'obtenir des performances comparables aux modèles classiques. Ces combinaisons ont atteint respectivement une précision de 70% et 67%, avec une disparité similaire entre les classes prédites. Nous avons également effectué une modélisation Word2Vec en utilisant un réseau dense, ce qui a entraîné une accuracy supérieure de 72%, mais avec une disparité plus importante entre les classes.

FastText a obtenu une performance supérieure sur les données de test, malgré le risque de surapprentissage avec une accuracy de 78%.

Enfin, nous avons choisi d'utiliser CamemBERT en combinaison avec le modèle TextCNN en raison de son F1-score élevé, atteignant près de 84%. Cette combinaison a également permis de réduire de manière significative les erreurs de prédiction des classes

En résumé, nous avons sélectionné les modèles FastText, et CamemBERT avec le modèle TextCNN en fonction de leurs performances respectives en termes de précision, de F1-score et de capacité à réduire les erreurs de prédiction des classes.

Entraînement des modèles retenus

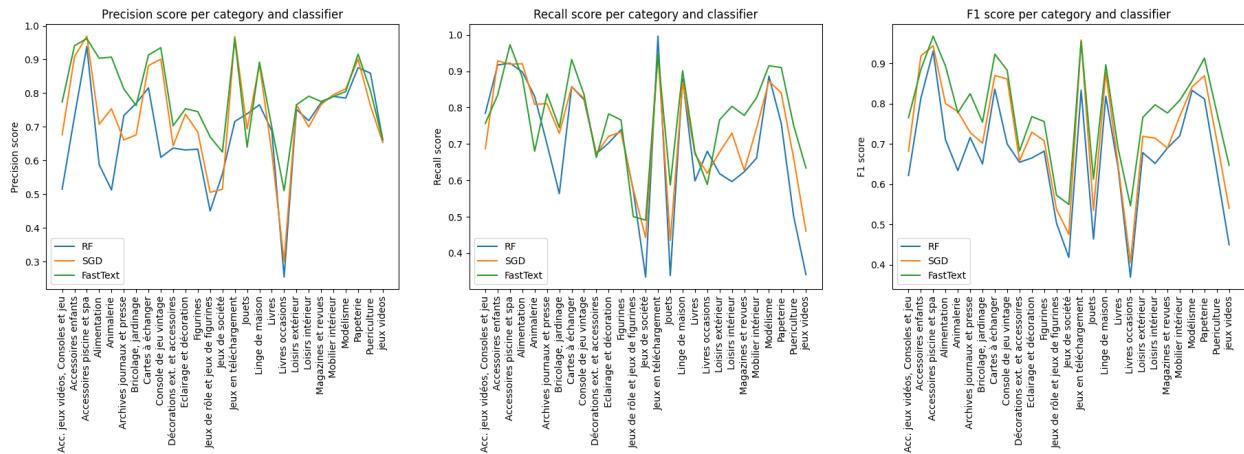
Pour l'entraînement des modèles Random Forest et Stochastic Gradient Descent, nous avons suivi une procédure standard :

1. exploration des hyperparamètres à l'aide d'une grille de recherche à validation croisée (3 "folds" stratifiés pour conserver la représentativité des valeurs-cibles)
2. sélection des hyperparamètres en cherchant le meilleur compromis entre performance et limitation du surapprentissage
3. découpage des données d'entrée en un jeu d'entraînement (67% du volume) et un jeu de test (33% du volume) en mode stratifié
4. ré-entraînement avec les hyper paramètres choisis
5. évaluation des scores sur les données d'entraînement et de test

La procédure a été différente pour l'entraînement et le calage du modèle FastText : nous avons travaillé avec une grille de recherche d'hyper paramètres mais sans validation croisée, l'entraînement et la validation s'effectuant sur les mêmes données que celles du point 3 du travail sur les modèles Random Forest et Stochastic Gradient Descent. Ajoutons qu'il est possible de fournir un jeu de vecteurs pré-entraînés à ce modèle, y compris en langue française. Ceci nous a permis d'améliorer les performances de FastText en ajoutant environ 1 point à l'accuracy sur les données de test.

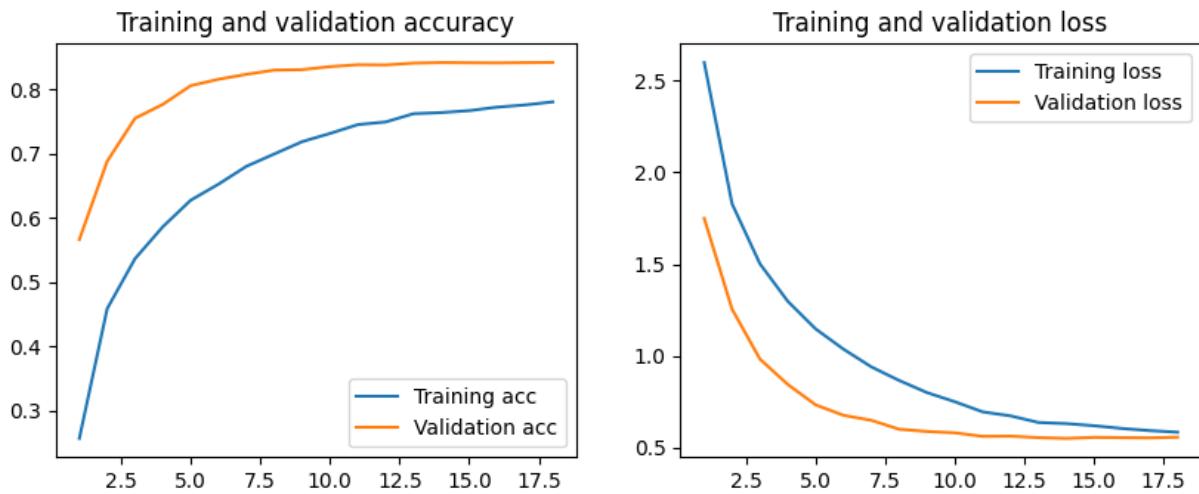
Performance des modèles à l'entraînement

- Random Forest, Stochastic Gradient Descent et FastText : scores (précision, rappel et F1) sur les données de test, par valeur de classe-cible



FastText montre une précision bien supérieure sur certaines classes mais également de très bon résultats de recall, ce qui montre clairement que ce modèle détecte mieux un certain nombre de classes comparé à Random Forest et Stochastic Gradient Descent.

- CamemBERT avec Text CNN : accuracy et perte par nombre d'époches



Nous constatons une diminution régulière de la loss pour les données d'entraînement comme pour les données de validation ce qui indique que notre modèle apprend à mieux ajuster les données et à minimiser les erreurs, signe d'une capacité de généralisation de notre modèle.

Evaluation des modèles retenus

FastText a obtenu une bonne performance avec une accuracy et une F1-score de 77%.

Par contre nous avons une disparité plus importante des classes mal prédites.

▷ Résultats des classes les plus mal prédites avec FastText :
La classe Livres occasions a souvent été prise pour la classe Livres
La classe Livres a souvent été prise pour la classe Livres occasions
La classe Décorations ext. et accessoires a souvent été prise pour la classe Bricolage, jardinage
La classe Jeux de société a souvent été prise pour la classe Jouets
La classe Jeux de rôle et jeux de figurines a souvent été prise pour la classe Jeux de société
La classe Jeux en téléchargement a souvent été prise pour la classe Livres

Le modèle Word2Vec qui a été entraîné en utilisant un réseau dense, a permis d'obtenir une accuracy de 72%, mais avec une disparité toute aussi importante entre les classes.

Résultats des classes les plus mal prédites avec le réseau dense :
La classe Jeux de rôle et jeux de figurines a souvent été prise pour la classe Acc. jeux vidéos, Consoles et jeu
La classe Eclairage et décoration a souvent été prise pour la classe jeux vidéos
La classe jeux vidéos a souvent été prise pour la classe Papeterie
La classe Animalerie a souvent été prise pour la classe Magazines et revues
La classe Jeux en téléchargement a souvent été prise pour la classe Acc. jeux vidéos, Consoles et jeu

Enfin, CamemBERT en combinaison avec le modèle TextCNN a permis d'obtenir une F1-score de près de 84%. Cette combinaison a également permis de réduire de manière significative les erreurs de prédiction des classes.

Résultats des classes les plus mal prédites avec CamemBERT :
La classe Accessoires piscine et spa a souvent été prise pour la classe Acc. jeux vidéos, Consoles et jeu
La classe Jeux de rôle et jeux de figurines a souvent été prise pour la classe Acc. jeux vidéos, Consoles et jeu
La classe Eclairage et décoration a souvent été prise pour la classe jeux vidéos
La classe jeux vidéos a souvent été prise pour la classe Loisirs extérieur

Les classes les moins bien prédites sont principalement les classes que nous avions identifiées lors de la phase préparatoire. Nous avions en effet identifié que certaines classes étaient les moins décrites en texte (10 - Des livres d'occasion, 2403 - Des magazines et revues, 1180 - Jeux de rôle et jeux de figurines, 2462 - Acc. jeux vidéos, Consoles et jeu, 1160 - Cartes à échanger).

Le modèle CamemBERT avec TextCNN a donc une bonne capacité à réduire les erreurs de prédiction des classes.

Classification d'images : Les modèles de Deep Learning

Les études ont montré que lorsque nous souhaitons analyser des images, ce sont les réseaux de neurones à convolution qui s'en sortent le mieux. Ils sont conçus grâce à des expériences effectuées sur le cortex et le système de vision des animaux, ces réseaux sont beaucoup plus légers que leurs confrères composés de couches de neurones entièrement connectés les uns aux autres.

Sur la base de nos images, nous avons entraîné différents modèles de deep learning :

- CNN (Convolutional Neural Networks) classique
- VGG16
- VGG19
- Resnet152
- Xception
- InceptionV3

Les Réseaux Neuronaux Convolutifs (CNN) classiques sont des modèles fondamentaux de la classification d'images, ayant prouvé leur efficacité dans une variété de tâches de vision par ordinateur.

VGG16, VGG19 et ResNet152, tout comme les CNN classiques, reposent sur l'architecture de convolution, mais avec des structures de réseau plus profondes et des techniques d'optimisation plus avancées. Ils représentent une évolution des approches classiques et sont connus pour avoir une performance supérieure sur de nombreuses tâches de classification d'images.

InceptionV3 et Xception sont des modèles plus récents qui introduisent des architectures de réseau innovantes.

CNN classique : Les réseaux de neurones convolutifs (CNN) sont les modèles de base de la classification d'images. Ils sont caractérisés par :

- L'utilisation de couches de convolution pour extraire automatiquement les caractéristiques des images.

-
- La possibilité de traiter des images en entrée de différentes tailles.
 - Leur efficacité pour diverses tâches de classification d'images grâce à leur capacité à apprendre des caractéristiques hiérarchiques à partir des données d'entrée.

VGG16 et VGG19 : Les modèles VGG16 et VGG19 sont des réseaux de neurones convolutifs développés par l'Université d'Oxford. Ils sont caractérisés par :

- Une architecture profonde avec respectivement 16 et 19 couches de poids.
- L'utilisation de petites convolutions 3x3, qui leur permet d'apprendre une hiérarchie de caractéristiques complexes.
- Ces modèles sont connus pour leur simplicité et leur capacité à traiter des images de taille standard (224x224 pixels), ce qui les rend adaptés à de nombreuses tâches de classification d'images.

ResNet152 : ResNet152 est un modèle de classification d'images basé sur les réseaux de neurones convolutifs, développé par Microsoft. Voici ses caractéristiques principales :

- ResNet152 utilise des "connexions résiduelles" pour permettre le flux d'information à travers les couches du réseau, ce qui aide à éviter le problème de disparition du gradient dans les réseaux profonds.
- Il a une architecture profonde de 152 couches qui lui permet d'apprendre des représentations de caractéristiques de haut niveau.
- ResNet152 est souvent utilisé pour des tâches de classification d'images de grande taille en raison de sa capacité à capturer des informations de haut niveau.

Xception : Xception est un modèle de classification d'images basé sur les réseaux de neurones convolutifs, développé par François Chollet, le créateur de Keras. Voici les éléments qui le caractérisent :

- Il améliore le modèle Inception en remplaçant les opérations de convolution standard par des opérations de convolution séparables en profondeur.
- Cette approche permet de gérer les cartes d'entités spatiales et de canal de manière distincte, ce qui améliore l'efficacité et la performance du modèle.
- Xception est souvent utilisé pour des tâches de classification d'images de grande taille en raison de sa capacité à capturer des informations de haut niveau grâce à sa profondeur et à ses connexions résiduelles.

InceptionV3 : InceptionV3 est un autre modèle de classification d'images basé sur les réseaux de neurones convolutifs, développé par Google. Il est caractérisé par :

- L'utilisation de modules qui permettent au réseau de choisir les échelles de caractéristiques les plus pertinentes pour l'apprentissage.
- Une architecture plus profonde que les modèles CNN traditionnels, avec moins de paramètres, ce qui rend le modèle plus efficace.
- InceptionV3 est capable de capturer les détails complexes dans les images grâce à sa profondeur et ses connexions résiduelles.

Pré-traitement des données

À partir du dataset original que nous nommons *df*, contenant à la fois les images, les labels, les codes des produits et ceux des images, nous avons créé des échantillons d'entraînement (*X_train*, *y_train*) et de test (*X_test*, *y_test*). Il faudra respecter un certain ratio entre ces jeux de données. Ainsi, nous avons décidé de récupérer 80% des données pour le dataset d'entraînement et donc 20% pour le dataset de validation.

Ces différentes datasets nous ont permis par la suite de générer trois autres datasets (*train_dataset*, *validation_dataset* et *test_dataset*) à partir du générateur d'images *ImageDataGenerator*. Le générateur d'images permet de générer de nouvelles images à partir de celles déjà disponibles. Nous avons appliqué un redimensionnement de l'image au format 299,299 au lieu de 500,500 (format original des images), ainsi qu'un *zoom_range* de 0,2 pour supprimer les bruits pour les images centrées disposant d'un cadre blanc.

Le premier dataset (*train_dataset*) va permettre à notre réseau d'apprendre et d'extraire des caractéristiques distinctes de chacune de nos images. Le second (*validation_dataset*) quant à lui va servir à valider le modèle en fin de chaque itération au cours de l'entraînement. En effet, en montrant de nouvelles images à notre réseau, il va lui permettre de se recalibrer pour éviter de sur-apprendre les images du jeu de données d'entraînement. Cette calibration va lui permettre une bien meilleure généralisation de données.

Les modèles retenus pour la classification d'images

Trois modèles ont été retenus pour mettre en place notre classification : Xception, ResNet152 et InceptionV3. Les performances de ces modèles sur les données de

formation et de validation ont montré leur robustesse et leur capacité à extraire des caractéristiques pertinentes des images pour une prédiction précise.

InceptionV3 : Avec une accuracy d'entraînement de 68,12% et une accuracy de validation de 60,69%, InceptionV3 a démontré sa capacité à apprendre efficacement à partir des données d'entraînement tout en conservant une certaine capacité de généralisation sur les données de validation. Bien qu'il y ait un certain niveau de surapprentissage, il est un candidat solide pour la tâche.

Xception : Le modèle Xception, bien qu'il ait affiché des performances légèrement inférieures à InceptionV3, a prouvé sa supériorité en termes de généralisation, comme en témoigne l'écart plus faible entre les précisions d'entraînement et de validation. L'analyse de la matrice de confusion a également révélé une diagonale plus définie que les deux autres modèles retenus, indiquant ainsi que les prédictions du modèle correspondent de manière cohérente aux classes réelles des images.

ResNet152 : Le modèle ResNet152 a démontré une performance tout à fait respectable. Au fil des epochs, la précision s'est améliorée de manière constante, tandis que la perte diminuait, indiquant une amélioration progressive de la performance du modèle. Après 30 epochs, ResNet152 atteignait une précision de 52,96% sur l'ensemble d'entraînement avec une perte de 1,71. Sur l'ensemble de validation, l'accuracy était de 52,21% avec une perte de 1,72, tandis que sur l'ensemble de test, l'accuracy était de 52,08% avec une perte de 1,67. Bien que ces valeurs soient légèrement inférieures à celles des autres modèles, la performance constante de ResNet152 sur les trois ensembles de données montre sa robustesse et sa capacité à généraliser.

En comparaison, les modèles tels que VGG16, VGG19 et des CNN simples ont été jugés moins performants. Bien que les modèles VGG soient largement utilisés dans la classification d'images, ils peuvent être limités dans leur efficacité et leur capacité à traiter des tâches plus complexes. De même, les réseaux convolutifs simples peuvent manquer de profondeur ou de complexité pour capturer toutes les caractéristiques pertinentes des images.

En conclusion, l'adoption des architectures InceptionV3, Xception et ResNet152 a été motivée par la recherche de performances robustes et de solides capacités de généralisation pour cette tâche de classification d'images. Ces modèles constituent un bon point de départ pour l'exploration et la classification multimodales ultérieures.

Entraînement des modèles retenus

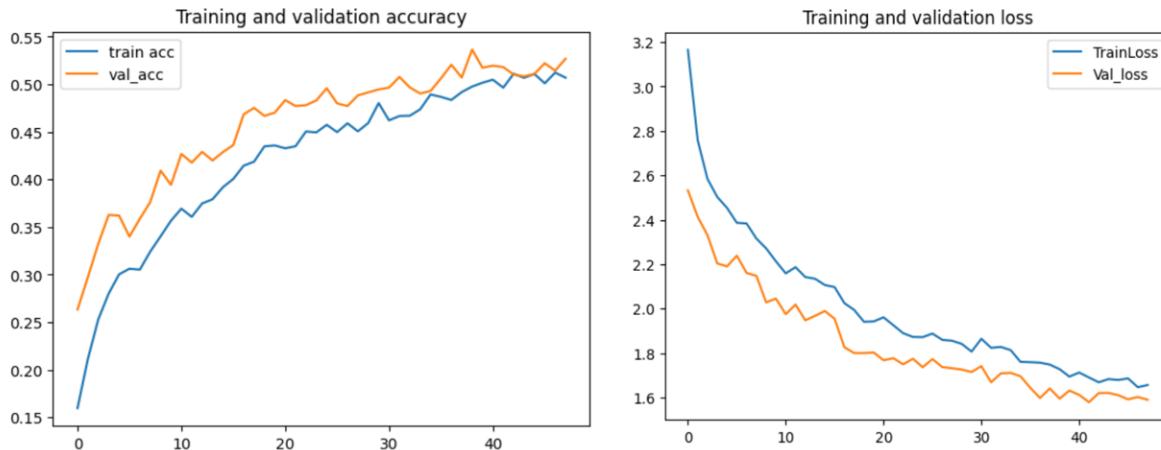
L'entraînement a été réalisé à partir des *DataframeIterator* d'entraînement et de validation, qui envoient les images et les labels associés vers la méthode *fit()* (de la classe *Model* de *tensorflow.keras.models*), en configurant pour la fin de chaque epoch l'appel de fonctions dites callbacks (héritant de la classe *Callback* de *tensorflow.keras.callbacks*).

Les modèles ont été entraînés avec une taille de batch de 64 et un learning rate $lr=0.0001$, afin de garder un temps d'entraînement raisonnable sur le jeu de données complet et un optimizer Adam (*opt = tf.keras.optimizers.Adam(lr = 0.0001)*) et pour 48 epoch pour le premier modèle, 30 pour le second et 20 epoch pour le troisième.

Une fois les modèles entraînés, nous avons observé comment ils se sont comportés. Cette observation à partir de la fonction de perte ou de précision, va nous permettre d'avoir de réels informations et indices sur le comportement de notre réseau, et ce sur le jeu de données d'entraînement et de validation.

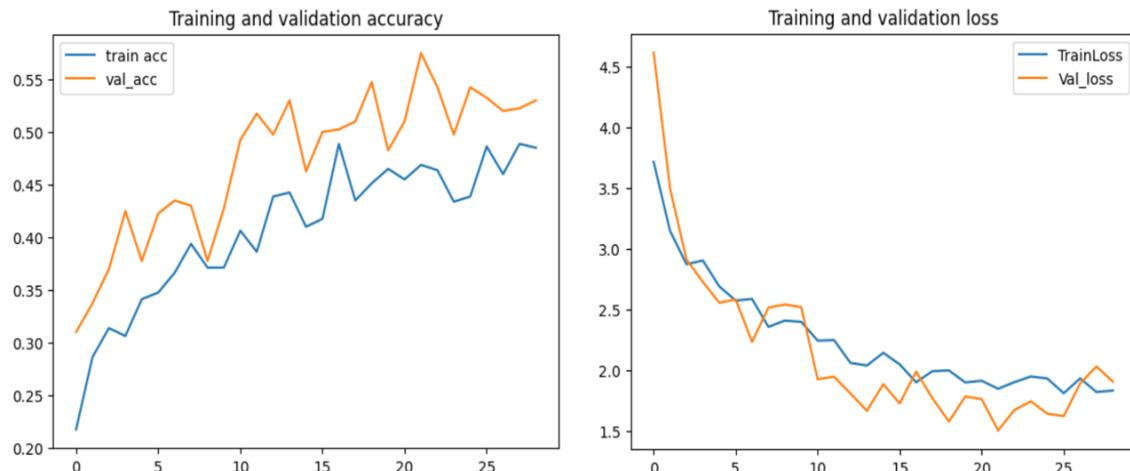
Performance des modèles à l'entraînement

- Modèles Xception

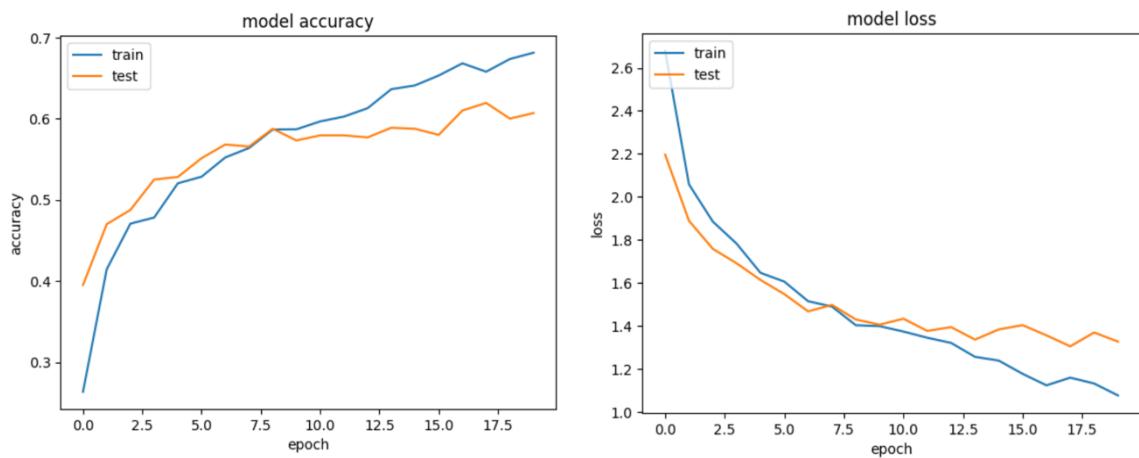


On observe à partir de ce graphique que le modèle présente un meilleur score sur le jeu de données de validation que sur celui d'entraînement. Cela peut résulter du fait d'avoir redimensionné les images, ce qui a par la suite perturbé les observations d'entraînement, mais pas celles de validation.

- Modèle ResNet152



- Modèle InceptionV3



On peut constater que le modèle n'a pas fini d'apprendre, en effet la courbe concernant le jeu de données de validation connaît une stagnation.

Evaluation des modèles retenus

Xception a atteint une accuracy de test de 52.46% avec une perte (loss) de 1.58. ResNet152 a atteint une accuracy de test identique de 52.08% avec une perte de 1.67. InceptionV3 a été retenu pour sa meilleure performance avec une accuracy de test de 61.69% et une perte de 1.29.

Techniques d'optimisation des paramètres

Tous les modèles classiques ont été optimisés par Grille (GridSearch) et tous les modèles de Deep Learning ont été entraînés avec un dropout de 50% pour réduire le surapprentissage.

Pour le traitement d'images, nous avons ajouté un callback *ReduceLROnPlateau* pour ajuster le taux d'apprentissage en fonction de la performance du modèle.

Pour Xception et ResNet152, la technique de *min_lr* a été appliquée pour définir la valeur minimale du taux d'apprentissage.

Pour InceptionV3, le callback *ModelCheckpoint* a été utilisé pour sauvegarder le meilleur modèle basé sur la perte de validation, une data augmentation pourrait permettre aussi une meilleure généralisation du modèle.

L'ajout de la classification d'images complémentaire à la classification de texte peut apporter une dimension supplémentaire à la tâche de classification.

Techniques d'interprétabilité

L'analyse du graphe des évolutions des métriques nous permet d'évaluer la performance et la capacité de généralisation de notre modèle. Une diminution régulière indique que notre modèle apprend à mieux ajuster les données et à minimiser les erreurs. Si la perte diminue pour les données d'entraînement mais augmente pour les données de validation, cela peut également être un signe de surapprentissage. Cela nous aide à identifier les problèmes potentiels et à prendre des décisions pour améliorer les performances du modèle, que ce soit par des ajustements d'hyperparamètres, des techniques de régularisation ou d'autres méthodes appropriées.

Nous avons aussi utilisé des outils classiques tels que le classification report et la matrice de confusion pour évaluer les performances de notre modèle et identifier les classes les plus mal prédites. Ces techniques nous ont permis d'obtenir une vision globale des performances du modèle et de cibler les classes qui nécessitent une attention particulière.

Amélioration des performances

Texte :

L'utilisation de modèles plus récents comme FastText et CamemBERT a conduit à une amélioration significative des performances par rapport aux modèles classiques. L'utilisation de vecteurs pré-entraînés a également contribué à une légère amélioration des résultats.

La réduction des classes mal prédites a été observée avec l'utilisation du tokenizer CamemBERT et TextCNN.

Images :

L'exploration d'autres architectures de modèles, telles que DenseNet ou EfficientNet, pourrait être envisagée pour améliorer les performances.

La réalisation d'une augmentation de données sur les images permettrait d'augmenter la taille du dataset et d'améliorer la capacité de généralisation des modèles.

L'ajustement des hyper paramètres, tels que le taux d'apprentissage ou la taille du dropout, a également été exploré pour optimiser les performances des modèles.

Proposition d'optimisation :

Une piste d'optimisation envisagée est la mise en place d'un **système de vote ou de prédiction multimodale**. Cette approche consisterait à combiner les prédictions des modèles entraînés sur les données textuelles et des modèles entraînés sur les données d'images en utilisant un système de vote majoritaire ou pondéré. En combinant l'information textuelle et visuelle, cela permettrait d'obtenir une prédiction finale plus robuste.

L'implémentation d'un système de vote entre les modèles textuels et les modèles d'images pourrait contribuer à augmenter les performances globales du modèle. En exploitant les forces de chaque modalité (texte et image), cette approche pourrait améliorer la capacité de généralisation du modèle.

En conclusion, bien que les performances des modèles ne soient pas très élevées dans l'état actuel, l'ajout de la classification d'images en complément de la classification de texte offre une approche plus holistique pour résoudre le problème de classification. Des améliorations potentielles, telles que l'utilisation de modèles plus avancés et l'augmentation de données, pourraient être explorées pour obtenir de meilleures performances à l'avenir. L'exploitation d'un système de vote ou de prédiction multimodale constitue une piste prometteuse pour améliorer la précision et la capacité de généralisation de notre modèle.

Classification à plusieurs modèles

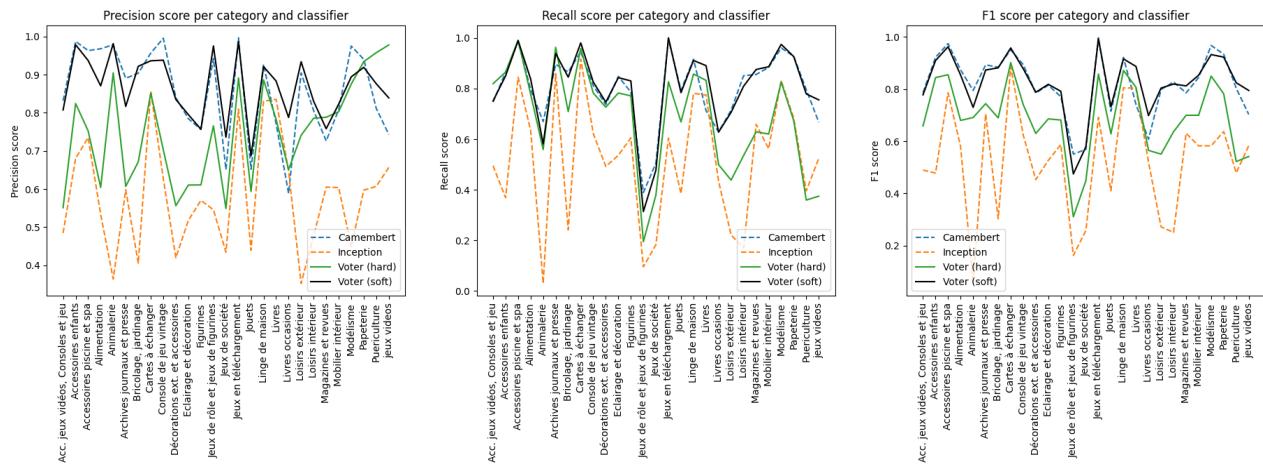
Les données d'entrée étant de nature multiple (texte et images), il était intéressant de combiner plusieurs modèles pour les prédictions. Ceci a été fait avec les modèles CamemBERT du text mining et InceptionV3 pour la computer vision.

Système de vote

La combinaison de modèles la plus simple est celle du "voteur" : on évalue les prédictions de modèles individuels sur un même jeu de données d'entrée et on compare les résultats pour aboutir à une prédiction "commune". Deux modes de vote existent :

- le mode “dur” (“hard”), qui consiste, pour chaque observation, à choisir la classe prédite par le plus grand nombre de modèles individuels ;
 - le modèle “doux” (“soft”), où l’on s’en remet aux probabilités de prédition classe par classe des différents modèles. Une moyenne éventuellement pondérée des résultats des modèles est calculée, et c’est la classe portant la probabilité moyenne la plus haute qui est retenue.

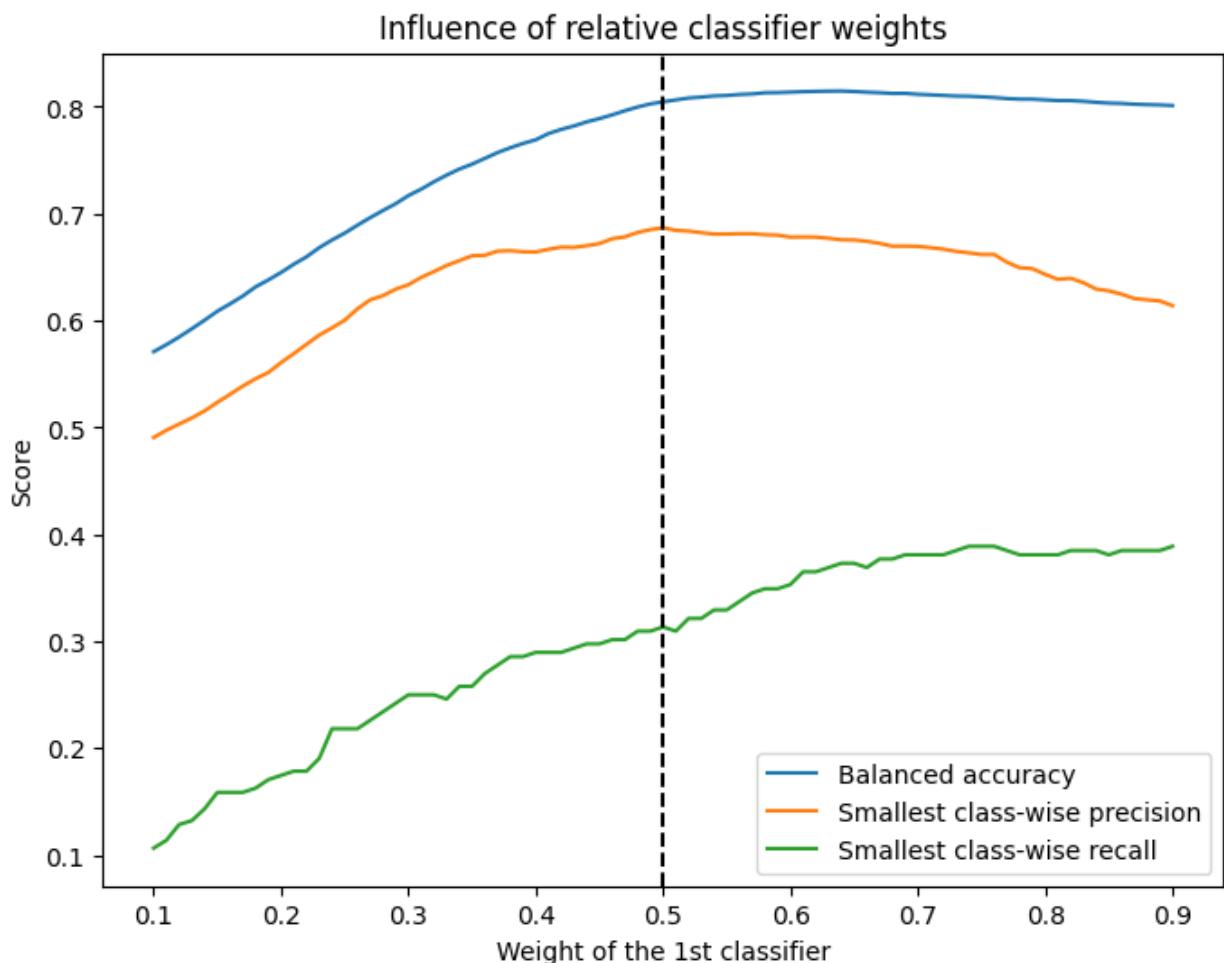
L'application de ce principe aux deux modèles pris en compte a donné des résultats modérés. Voici un comparatif entre ces deux modèles, le vote en mode "dur" et le vote en mode "doux" :



Il en ressort plusieurs choses :

- Le mode “dur” donne des résultats décevants, presque toujours intermédiaires entre CamemBERT et InceptionV3. Dans la mesure où il cherche une majorité entre seulement deux “voix”, cela pouvait être anticipé ;
- Le mode “doux” apporte un très léger gain dans l’accuracy équilibrée par rapport au modèle CamemBERT seul, environ 0.6 point. A l’échelle des classes individuelles, on peut constater tantôt une amélioration, tantôt une détérioration dans les prédictions. En somme, le résultat apparaît mitigé.

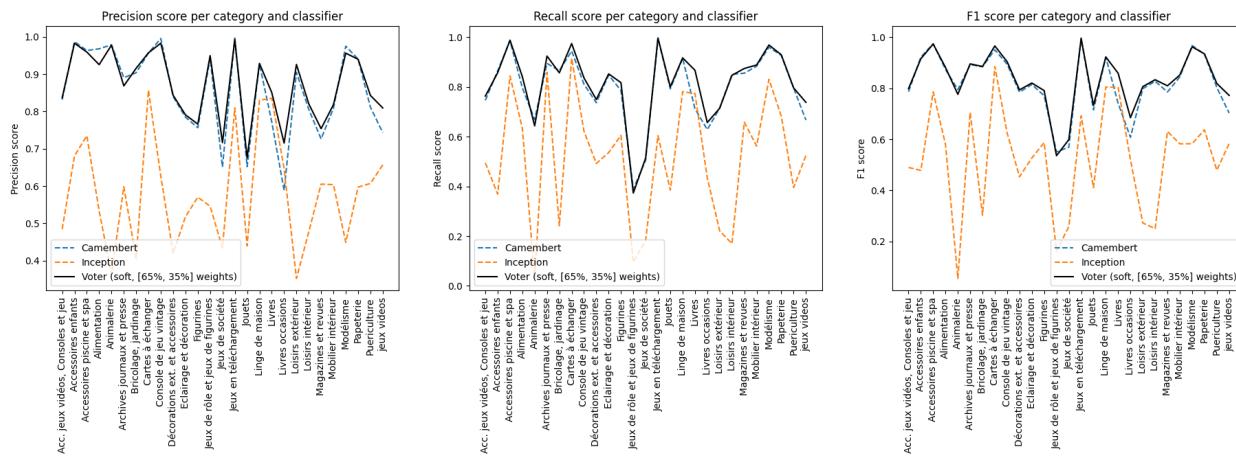
Une optimisation a toutefois été omise à ce stade. Il est en réalité possible de pondérer les classificateurs mis en jeu. Nous avons ainsi mesuré l’effet du poids attribué au modèle CamemBERT (et, symétriquement, au modèle InceptionV3) sur l’accuracy équilibrée ainsi que sur les rappels et précisions pour les classes les moins bien prédites :



Comme nous pouvons le constater, il est indispensable d'assigner un poids d'au moins 50% au modèle CamemBERT, ce qui semble cohérent avec sa meilleure capacité de prédiction que celle d'InceptionV3. Un peu plus en détail :

- choisir des poids identiques pour les deux modèles maximise la précision pour la classe la plus difficile à prédire, même si donner un poids plus important à CamemBERT (jusqu'à 75%) reste raisonnable sur cet aspect
- le poids optimal à donner à CamemBERT pour maximiser le pire des rappels est de 74%
- quant à l'accuracy équilibrée, son maximum est atteint avec un poids de 64% pour CamemBERT et 36% InceptionV3

Dans la suite, nous avons retenu des poids de 65% et 35% pour CamemBERT et InceptionV3 respectivement. Cela se traduit ainsi sur les scores classe par classe :



Le bénéfice d'une attribution de poids correctement choisis apparaît clairement : s'il est vrai que certaines classes sont moins bien prédites en comparaison avec CamemBERT (p.ex., une précision inférieure pour la catégorie "Alimentation"), on peut surtout constater que le système de vote est moins enclin aux erreurs pour les classes les plus problématiques, telles "Jeux de société" et "Livres occasions". Le système de vote est ainsi plus robuste que chacun des deux modèles (text mining et computer vision) pris séparément.

L'accuracy équilibrée atteinte avec le voteur pondéré atteint 81% (F1-score de 86%), soit un point de plus que CamemBERT.

Conclusion

Difficultés rencontrées

Lors de la réalisation de ce projet, nous avons fait face à plusieurs difficultés qui ont pu ralentir la mise en place de notre solution :

Verrou scientifique : Le principal verrou scientifique auquel nous avons été confrontés était la combinaison des données textuelles et des données d'images pour la classification. L'intégration de ces deux modalités d'information a nécessité une compréhension approfondie des techniques de traitement du langage naturel et de computer vision, ainsi que la recherche de modèles appropriés pour chaque modalité. La complexité de cette tâche exige des efforts supplémentaires pour garantir une intégration fluide et une performance optimale des modèles.

Jeux de données : L'acquisition et le traitement des jeux de données ont également représenté un défi. La volumétrie des données textuelles et d'images était importante, ce qui a nécessité des ressources de stockage suffisantes et des techniques de prétraitement efficaces pour gérer ces volumes de données. L'agrégation des données et leur mise en forme cohérente ont également demandé un travail minutieux.

Compétences : L'acquisition des compétences nécessaires pour maîtriser les techniques avancées de traitement du langage naturel et de computer vision a pris plus de temps que prévu. L'apprentissage des modèles de deep learning et des architectures spécifiques a exigé des recherches supplémentaires et des formations complémentaires. Certaines compétences spécifiques, non proposées dans notre formation initiale, ont nécessité une acquisition autonome.

Pertinence : L'approche adoptée pour combiner les informations textuelles et visuelles a été un point critique. La sélection des modèles appropriés pour chaque modalité et leur intégration pour obtenir une prédiction globale plus précise ont nécessité une réflexion

approfondie. La pertinence de l'approche a été évaluée en comparant les résultats obtenus avec le benchmark rakuten et en analysant les performances des modèles.

Infrastructure informatique : La puissance de stockage et de calcul était un élément à prendre en compte pour manipuler les jeux de données volumineux et entraîner les modèles de deep learning. L'optimisation des ressources informatiques et l'utilisation de plateformes adaptées ont été nécessaires pour réaliser les expérimentations de manière efficace.

Bilan

Dans le cadre de ce projet, notre contribution principale a été de développer une approche holistique en combinant les informations textuelles et visuelles pour la classification. Nous avons réalisé une sélection rigoureuse des modèles appropriés pour chaque modalité et optimiser leurs performances de façon progressive. De plus, nous avons exploité des techniques de prétraitement des données avancées pour garantir la qualité des informations utilisées.

Depuis la dernière itération, nous avons exploré différents modèles de prédiction multimodale pour tenter d'optimiser les performances mais avons été confrontés à des problématiques de capacité système.

Les résultats obtenus comparés au benchmark ont tout de même montré des performances encourageantes : Le benchmark des meilleures performances rakuten affiche un F1-score de 81% sur le texte (avec CNN classifier) et 55% sur les images (avec ResNet50) contre 84% sur le texte (avec CamemBERT) et 62% sur les images (avec InceptionV3) pour notre projet.

En conclusion, malgré les difficultés rencontrées, notre projet a abouti à une approche prometteuse pour la classification. L'intégration de l'information textuelle et visuelle va permettre d'améliorer les performances et d'ouvrir de nouvelles perspectives pour résoudre des problèmes de classification complexes.

Suite du projet

Dans le cadre de notre projet, des pistes d'amélioration peuvent encore être envisagées pour augmenter les performances de notre modèle comme la combinaison multimodale.

Il existe différentes solutions pour effectuer la classification multimodale comme :

Les Modèles d'apprentissage en fusion : Cette approche consiste à entraîner des modèles d'apprentissage séparés pour chaque modalité, puis à fusionner leurs sorties pour obtenir la prédiction finale.

Les Réseaux de neurones multimodaux (Dual Encoder) : Ces réseaux fusionnent les informations des différentes modalités dès les premières couches pour permettre une représentation conjointe des données.

Les Réseaux de neurones récursifs (LSTM Multimodal) : Ces modèles utilisent des réseaux récurrents ou des mécanismes de traitement de séquences pour traiter des données multimodales séquentielles.

L'exploitation de prédiction multimodale pourrait tout à fait contribuer à améliorer les performances globales du modèle. En exploitant les forces de chaque modalité (texte et image), cette approche pourrait améliorer la précision et la capacité de généralisation du modèle.

Notre projet a contribué à l'avancement des connaissances scientifiques en explorant la classification multimodale, en proposant une approche holistique pour la résolution du problème et en utilisant des techniques de prétraitement avancées. Les résultats obtenus ont ouvert de nouvelles perspectives de recherche et ont montré le potentiel de ces approches pour des applications réelles.

Bibliographie

- Text mining : [Text Mining : Classification Automatique de textes · HeadMind Partners](#)
- Word2Vec: [Word2Vec For Text Classification \[How To In Python & CNN\]](#)
- CamemBERT: [Analyse de sentiments avec CamemBERT | Le Data Scientist](#)
- FastText : [fastText](#)
- Keras Documentation officielle : [Keras API reference](#)
- Prédiction Multimodale :
<https://paperswithcode.com/paper/image-and-text-fusion-for-upmc-food-101-using>