



DataScientest • com

Rapport Technique d'évaluation

Équipe : Rakupy

Classification de produits e-commerce

Rakuten France Multimodal Product Data Classification



Promotion Juin 2021/Formation continue

Participants :

Tooba SHAUKAT

Brahim CHLAGHME

Mourad MECHERI

Mentor:

Gaspard GRIMM

Sommaire

1	Contexte	4
2	Objectifs.....	5
3	Data	5
3.1	Description des données	5
3.2	Modèle de référence	6
3.3	Exploration et Visualisation des données	7
3.3.1	Analyse des données	7
3.3.2	DataViz.....	9
3.3.3	Observations.....	13
3.3.4	Identification des variables	14
3.4	Préprocessing des données textes	16
3.4.1	Nettoyage du texte.....	16
3.4.2	Représentation vectorielle	16
3.5	Préprocessing des Images	17
4	Projet	18
4.1	Modélisation partie Texte	18
4.1.1	Machine learning.....	18
4.1.2	Deep learning	19
4.1.3	Conclusion	21
4.2	Modélisation partie Images.....	21
4.2.1	Concepts et définitions.....	21
4.2.2	Stratégie	22
5	Classification bimodale.....	34
5.1	Choix des meilleurs modèles	34
5.2	Combinaisons utilisées - Texte et Images	35
5.2.1	Simple DNN, Conv1D et Xception.....	35
5.2.2	Simple DNN, Conv1D et InceptionV3	37
5.3	Conclusion	38
5.4	Problèmes rencontrés	38
5.5	Pistes d'amélioration.....	39
6	Bilan	40
7	Annexes	41
7.1	Annexe 3.3.1 - Répartition des « NaNs » par classe de produit.....	42
7.2	Annexe 4.1.2.2 - Analyse matrices de confusion -Texte	43
7.2.1	Annexe 4.1.2.2.1 - Simple DNN	43
7.2.2	Annexe 4.2.2.1.2 - Conv1D	45

7.3	Annexe 4.2.2.5.5 Performances des modèles CNN	47
7.4	Description des fichiers de code	48
8	Bibliographie.....	50

1 CONTEXTE

Ce challenge porte sur le thème de la classification des produits. L'objectif est de prédire le code type (prdtypecode) de chaque produit sachant des données textuelles (désignation et description des produits) ainsi que des données images (image du produit) tel qu'il est défini dans le catalogue de Rakuten France.

Le catalogage des produits selon des données différentes (textes et images) est particulièrement important dans le domaine de l'e-commerce, avec des applications allant de la recherche et des recommandations personnalisées à la compréhension des requêtes. Les approches de catégorisation manuelles et/ou basées sur des règles ne sont pas évolutives car les produits commerciaux sont organisés en de nombreuses et parfois en milliers de classes. Dans ce contexte, le déploiement d'approches de classification multimodales automatisées serait une technique utile pour les entreprises de commerce électronique car elles permettent :

- L'utilisation des données texte et image associées aux produit permet d'éviter les doublons, en particulier lorsqu'elles vendent à la fois des produits neufs et d'occasion provenant de marchands professionnels et non professionnels, comme le fait Rakuten.
- Classification de produit à grande échelle pour mieux gérer les offres du site
- Proposition automatique des catégories aux utilisateurs mettant en vente des produits sur le site et l'amélioration de l'expérience client
- La mise en place des systèmes de recommandation de produits qui auront un impact rapide sur les ventes et l'augmentation des revenus de l'entreprise. Les avancées dans ce domaine de recherche ont été limitées en raison du manque de données réelles provenant de catalogues commerciaux. Le défi présente plusieurs aspects de recherche intéressants en raison de la nature intrinsèquement bruyante des étiquettes et des images des produits, de la taille des catalogues de commerce électronique modernes et de la distribution déséquilibrée typique des données.

2 OBJECTIFS

L'objectif de ce challenge, à partir des données textes et images, est de prédire le type des produits et les classer dans leurs catégories correspondantes, avec le moins d'erreurs possible. Les différentes classes de produits sont fournies dans le catalogue Rakuten.

Le catalogue Rakuten comprend plus de 1000 classes, mais dans le cadre de ce projet, les données fournies concernent 27 d'entre elles.

Nous sommes 3 personnes avec différents profils :

Brahim: Un background industriel et supplychain.

Mourad: Un background dans le domaine de l'informatique et des réseaux de télécommunication.

Tooba: 14 ans après avoir obtenu sa licence en mathématique, a pris le courage de reprendre ces études.

Ce projet s'aligne avec les principaux thèmes de notre formation à savoir : le Preprocessing et la Visualisation des données, la manipulation de textes et d'images, l'utilisation de modèles de Deep Learning & Machine Learning et l'optimisation des résultats obtenus. Ce projet était sans aucun doute un levier puissant de montée en compétence pour nous trois.

3 DATA

3.1 DESCRIPTION DES DONNÉES

Pour ce challenge, Rakuten France met à disposition environ 99 000 listes de produits au format CSV, y compris l'ensemble d'entraînement (84 916) et l'ensemble de test (13 812).

Le jeu de données se compose de :

- Désignations de produits,
- Descriptions de produits,
- Images de produits,
- Code de produit correspondant.

Les données sont réparties suivant des critères , formant 4 datasets distincts :

- **X_train_update.csv**: il s'agit d'un tableau contenant des échantillons d'entraînement, avec la description textuelle ainsi que le référencement du fichier image associé
- **y_train_CVw08PX.csv** : ce fichier correspond à un tableau contenant les 84916 codes des produits (prdtypecode)
- **X_test_update.csv**: il s'agit d'un tableau contenant des échantillons de test
- **images.zip**: ce fichier contient toutes les images. Une fois le .zip décompressé , on obtient deux sous dossiers :
 - « image_train » contenant 84 916 images
 - « image_test » contenant 13 812 images

La première ligne des fichiers csv contient les colonnes suivantes :

- « Id »: un Id entier pour le produit. Cet Id est utilisé pour associer le produit à son code de produit
- « designation »: le titre du produit, un texte court résumant le produit
- «description»: un texte plus détaillé décrivant le produit. Tous les utilisateurs n'utilisent pas ce champ, donc le champ description peut contenir des valeurs manquantes
- « productid »: un identifiant unique pour le produit
- « imageid»: un identifiant unique pour l'image associée au produit

Les noms des fichiers images comprennent « imageid » et le « productid » de l'article correspondant, ce qui permet de les associer aux données textes (designation et description), au productid et au imageid fournis dans les fichiers csv.

Les noms des fichiers images sont sous le format : *image_imageid_product_productid.jpg*

3.2 MODÈLE DE RÉFÉRENCE

Un modèle de référence est indiqué dans [le site](#) du challenge Rakuten. La référence indique deux modèles distincts pour les images et le texte.

- Pour les données images, une version du modèle Residual Networks (ResNet) ([Référence](#)) est utilisée. Le ResNet5 pré-entraîné avec un jeu de données Imagenet. 27 couches différentes du haut sont dégelées, dont 8 couches de convolution pour l'entraînement. Le

réseau final contient 12144667 paramètres entraînables et 23643035 paramètres non entraînaibles.

- Pour les données textes, un classificateur RNN simplifié est utilisé. Seuls les champs de « designation » sont utilisés dans ce modèle de référence.

Il est demandé d'utiliser la métrique **weighted-F1-score** pour évaluer les performances de la classification.

Le modèle de référence décrit ci-dessus obtient les résultats suivants :

- RNN utilisant les données texte : 0.8113 weighted F1-score
- ResNet sur les données image : 0.5534 weighted F1-score

L'objectif du projet étant est de faire mieux que les résultats du modèle de référence.

3.3 EXPLORATION ET VISUALISATION DES DONNÉES

3.3.1 Analyse des données

Suite à l'exploration des données, nous avons observé que :

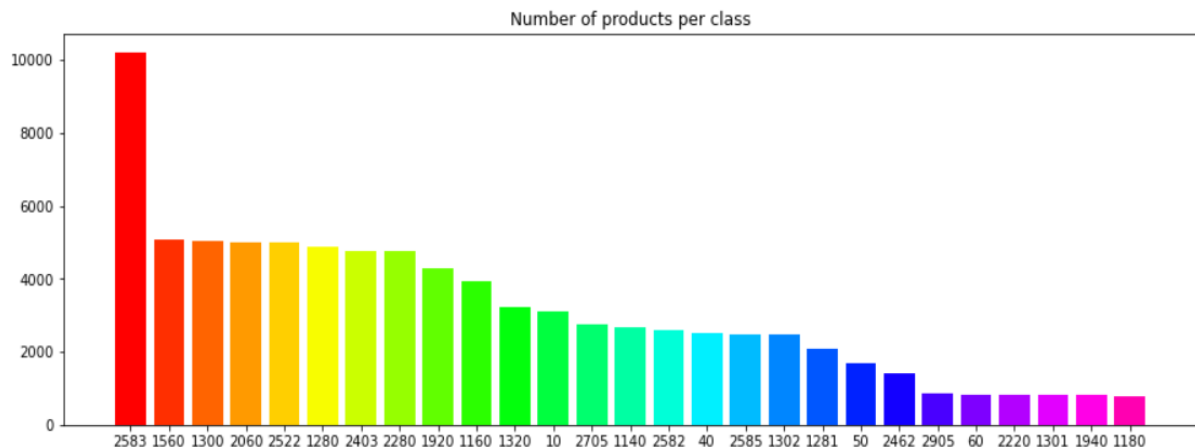
- Le jeu de données est complet
- Le jeu de données présentes des caractères spéciaux
- Le jeu de données ne présente aucun doublon
- Chaque article est identifié par un code «productid » unique
- Tous les articles ont un champ «designation » renseigné
- Présence de valeurs manquantes dans la colonne «description» comme montré dans la figure ci-dessous
 - Les valeurs manquantes représentent 35% de l'ensemble des colonnes « description »

```
np.round(xtrain.isna().sum()*100/len(xtrain),2)
```

designation	0.00
description	35.09
productid	0.00
imageid	0.00
text	0.00

- Les classes 2403, 2280, 1160 présentent le plus de valeurs manquantes (Voir Annexe 3.3.1 pour plus de détails)
- Les images sont en .JPG

- Les images sont en couleur et toutes de dimensions (500 x 500) pixels
- La représentation ci-dessous la répartition des différents produits par des classes :



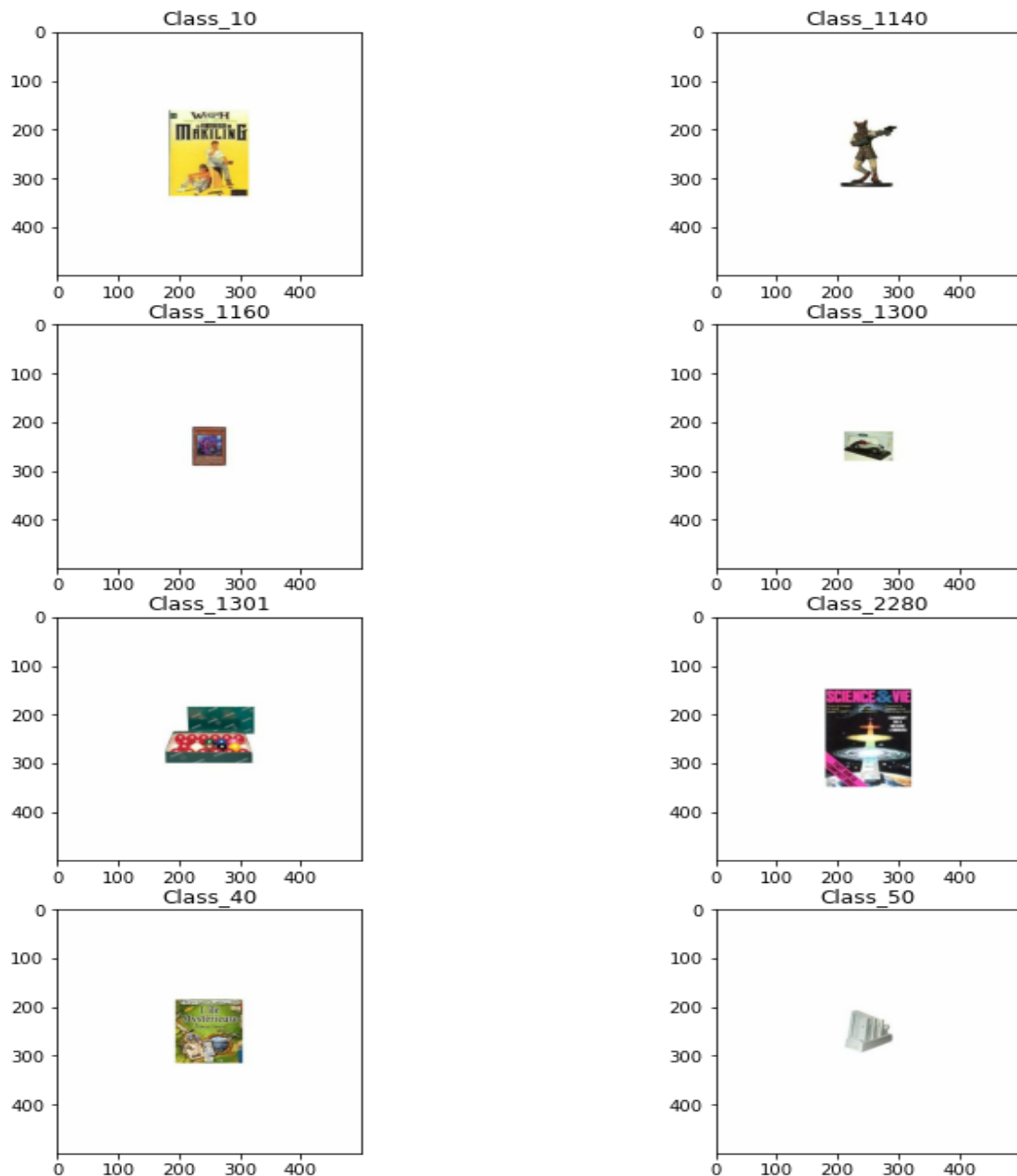
- La proportion des produits par catégorie est déséquilibrée. Les classes minoritaires « 1180, 1940 , 1301, 2220, 60» représentent entre 7 et 8 % du nombre d'échantillons dans la classe majoritaire « 2583 ». Les autres classes présentent des écarts moins importants. Dans l'ensemble, le jeu de données présente un déséquilibre plutôt modéré !

Classe (Product type codes)	Nombre d'observations	Proportion par rapport au dataset	% par rapport à la classe majoritaire « 2583 »
2583	10209	12.02%	-
1560	5073	5.97%	49,69 %
1300	5045	5.94%	49,42 %
2060	4993	5.88%	48,91 %
2522	4989	5.88%	48,87 %
1280	4870	5.74%	47,70 %
2403	4774	5.62%	46,76 %
2280	4760	5.61%	46,63 %
1920	4303	5.07%	42,15 %
1160	3953	4.66%	38,72 %
1320	3241	3.82%	31,75 %
10	3116	3.67%	30,52 %
2705	2761	3.25%	27,04 %
1140	2671	3.15%	26,16 %
2582	2589	3.05%	25,36 %
40	2508	2.95%	24,57 %
2585	2496	2.94%	24,45 %
1302	2491	2.93%	24,40 %
1281	2070	2.44%	20,28 %
50	1681	1.98%	16,47 %
2462	1421	1.67%	13,92 %
2905	872	1.03%	8,54 %
60	832	0.98%	8,15 %
2220	824	0.97%	8,07 %



3.3.3 Observations

- L'exploration visuelle et l'utilisation des WordCloud par classe montrent une forte ressemblance entre certaines classes (comme par exemple : 1140 et 1180 , 50 et 60, 1280 et 1302, 10 et 2280), et parfois deux classes différentes peuvent contenir le même type d'articles (comme par exemple les classes 1280 et 1302)
- Certaines images présentent un fond blanc important. Ce fond blanc ne semble pas être lié spécifiquement à la classe



- L'exploration visuelle et l'utilisation des WordCloud par classe nous ont permis d'identifier les catégories de produits suivantes :

Répartition des catégories de notre dataset

Product Code	Label	number_of_products
10	adult books	3116
2280	magazines	4760
50	video games accessories	1681
1280	toys for children	4870
2705	books	2761
2522	stationery	4989
2582	furniture kitchen and garden	2589
1560	interior furniture and bedding	5073
1281	board games	2070
1920	interior accessories	4303
2403	children books and magazines	4774
1140	figurines and Toy Pop	2671
2583	piscine spa	10209
1180	figurines, masks and role playing games	764
1300	remote controlled models	5045
2462	games	1421
1160	playing cards	3953
2060	decoration interior	4993
40	imported video games	2508
60	games and consoles	832
1320	early childhood	3241
1302	toys, outdoor playing, clothes	2491
2220	supplies for domestic animals	824
2905	online distribution of video games	872
2585	gardening and DIY	2496
1940	Food	803
1301	accessories children	807

3.3.4 Identification des variables

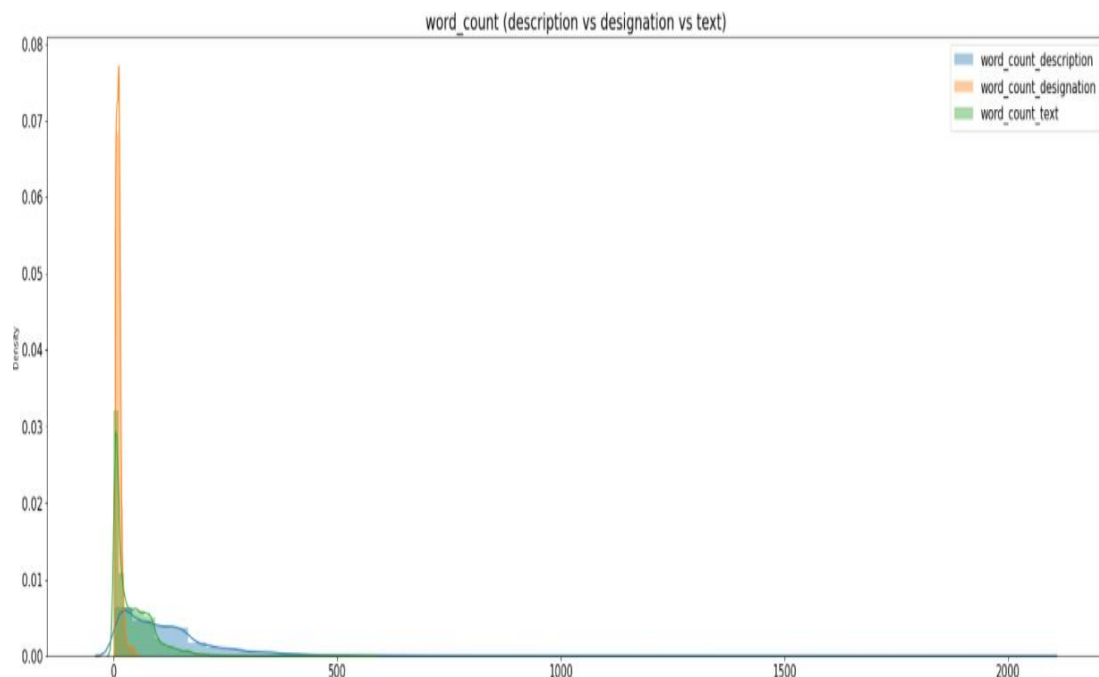
3.3.4.1 Variables explicatives (Features)

Dans un premier temps nous avons utilisé la colonne « designation » comme utilisé dans le modèle de référence du challenge Rakuten France. De plus, celle-ci ne contient pas de valeurs manquantes. Nous avons constaté par la suite que les performances sont limitées en se basant sur cette colonne uniquement.

Le champ «description» donne plus d’informations sur les produits, mais comporte un nombre important de valeurs manquantes (35 %).

Pour les données textuelles, nous avons donc opté pour une autre stratégie qui consiste à regrouper les deux variables «description» et «designation» en une variable unique appelée « **text** ». Ceci nous permet d’exploiter un maximum d'informations pour le problème de classification et pourrait donner de meilleures performances.

Les illustrations visuelles de la longueur des mots pour les variables « designation », « description » et « text » sont les suivantes :



3.3.4.2 Variable cible

Nous avons encodé nos 27 classes de code produit de 0 à 26. Ceci a été nécessaire pour l’entraînement de nos modèles. La table suivante donne la correspondance entre les deux champs.

Classe produit (prdtypecode)	Target
10	0
40	1
50	2
60	3
1140	4
1160	5
1180	6
1280	7

1281	8
1300	9
1301	10
1302	11
1320	12
1560	13
1920	14
1940	15
2060	16
2220	17
2280	18
2403	19
2462	20
2522	21
2582	22
2583	23
2585	24
2705	25
2905	26

3.4 PRÉPROCESSING DES DONNÉES TEXTES

3.4.1 Nettoyage du texte

Nous avons procédé aux opérations suivantes sur la colonne «text» :

- Mettre tous les mots en lettres minuscules
- Suppression des accents
- Suppression des balises HTML
- Instanciation des stopwords: Français, Anglais et Allemand
- Suppression des stopwords classiques, mais également des mots présents en grandes quantités dans toutes les classes trouver par les WordCloud.
- Suppression des mots ayant moins de deux lettres

3.4.2 Représentation vectorielle

L'objectif du projet est de classer les produits grâce à la désignation et la description donnée par le vendeur. Pour cet objectif nous avons appliqué différentes techniques - de text mining, tokenisation et la bibliothèque NLTK - qui ont permis de transformer les données texte en tokens.

Pour appliquer les différents modèles de la librairie scikit-learn, chaque observation du text est vectorisé à l'aide de la classe ***TfidfVectorizer*** en fixant les maximum features à 5000.

Pour appliquer les différents modèles du Deep Learning, nous avons séparé le texte en mots grâce à la classe **Tokenizer** de « *tf.keras.preprocessing.text* » avec un nombre maximum de mots 20 000. Nous avons ensuite défini la longueur maximum d'une séquence à 200.

3.5 PRÉPROCESSING DES IMAGES

Au début , nous avons redimensionnées les images en 256 x 256 pixels, d'après la littérature, la dimension optimale entre perte d'information et complexité est aux alentours de (256, 256). Le réduction de la taille des images permet une durée d'entraînement plus rapide.

Nous avons utilisé un générateur de données (*ImageDataGenerator*, du module *tensorflow.keras.preprocessing.image*). Dans ce cadre, nous avons appliqué des transformations géométriques sur nos images (rescale , zoom, rotation, redimensionnement, retournement horizontal et verticalement...). Le générateur de données permet également les avantages suivants :

- D'enrichir notre jeu de données , qui est particulièrement pertinent pour notre jeu de données qui présente un déséquilibre important entre les différentes classes
- Le réseau ne rencontre jamais deux fois la même image (car ces augmentations sont aléatoires). Cette méthode permet de s'assurer que le réseau voit des images variées à chaque epoch (cycle d'entraînement complet). Ceci permet au modèle de mieux se généraliser et de diminuer ainsi le risque de surapprentissage (*overfitting*)
- Les images sont augmentées en temps réel lors de l'entraînement du réseau. Elles ne sont donc jamais sauvegardées en mémoire. Cela permet de réduire les contraintes liées aux manques de ressources de calcul

4 PROJET

Notre stratégie de travail consiste à travailler sur la partie texte et image séparément ensuite de réaliser un modèle bimodal qui consiste à combiner les deux parties.

4.1 MODÉLISATION PARTIE TEXTE

4.1.1 Machine learning

4.1.1.1 Modèles et Paramètres utilisés

Modèle	Paramètres
LOGISTIC REGRESSION	multi_class='multinomial', class_weight= "balanced", max_iter=1000
SVM	gamma = 0.01, kernel = 'poly'
RANDOM FOREST CLASSIFIER	max_features= 'sqrt', min_samples_split= 4
VOTING CLASSIFIER	<ul style="list-style-type: none">• KNeighborsClassifier(n_neighbors=27)• RandomForestClassifier• LogisticRegression• Voting='hard'
XGBOOST	XGBClassifier(use_label_encoder=False)
LINEAR SVC	penalty="l2", dual=True, C=0.8, tol=1e-5, max_iter=4000

Ci-dessous un récapitulatif comparatif des résultats d'entraînement en choisissant dans un premier temps seulement la colonne designation et dans un second temps en utilisant les deux colonnes designation et description, combinées en une colonne «text» :

Modèle	Colonne designation	Colonnes designation & description (text)
LOGISTIC REGRESSION	0.75	0.77
SVM	0.12	0.22
RANDOM FOREST CLASSIFIER	0.76	0.77
VOTING CLASSIFIER	0.73	0.74
XGBOOST	0.75	0.77
LINEAR SVC	0.77	0.78

Sur la partie texte, nous avons utilisé au début plusieurs modèles de Machine Learning, mais ils ont montré leurs limites pour atteindre des performances satisfaisantes. Nous nous sommes donc orientés vers modèles de Deep Learning.

4.1.2 Deep learning

4.1.2.1 Choix des paramètres

- Batch size: 200
- epochs: 5 epochs
- Learning Rate: 0.001
- Optimizer: “adam”

4.1.2.2 Résultats

Modèle	Colonne designation	Colonnes designation & description (text)
GRUCELL	0.75	0.77
BILSTM	0.74	0.76
CONV1D	0.76	0.81
DNN	0.77	0.81

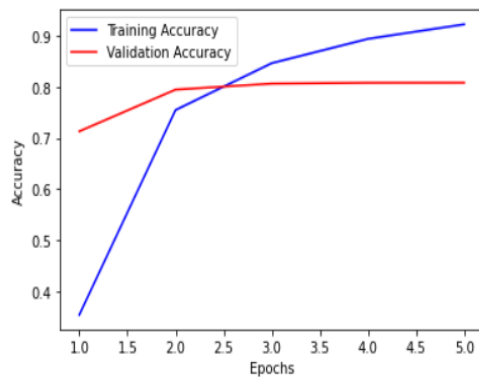


Figure DNN accuracy

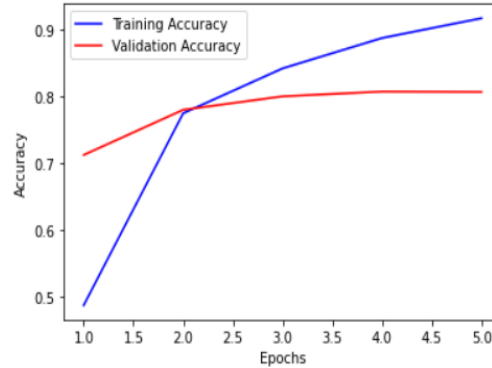


Figure Conv1D accuracy

Voici les résultats détaillés de Conv1D et Simple DNN (F1-score Weighted):

Target labels	Product code	F1-score	
		CONV1D	SIMPLE DNN
0	10	0,47	0,49
1	40	0,68	0,68
2	50	0,79	0,81
3	60	0,86	0,89
4	1140	0,77	0,79
5	1160	0,86	0,88
6	1180	0,64	0,58
7	1280	0,72	0,73
8	1281	0,56	0,60
9	1300	0,94	0,94
10	1301	0,94	0,92

11	1302	0,78	0,80
12	1320	0,79	0,79
13	1560	0,83	0,83
14	1920	0,92	0,90
15	1940	0,82	0,82
16	2060	0,81	0,80
17	2220	0,79	0,82
18	2280	0,68	0,72
19	2403	0,73	0,76
20	2462	0,79	0,80
21	2522	0,91	0,92
22	2582	0,75	0,75
23	2583	0,97	0,97
24	2585	0,78	0,79
25	2705	0,65	0,71
26	2905	0,99	1,00

F1-score		0,80	0,81
Weighted AVG			

Ci-après les classes les moins bien prédites : le F1-score est inférieur à celui du modèle de référence (0.80)

Texte - Conv1D	
Classe produit	Labels
10	adult books
1281	board games
1180	figurines, masks and role playing games
2705	books
40	imported video games
2280	magazines
1280	toys for children
2403	children books and magazines
2582	furniture kitchen and garden
1140	figurines and Toy Pop
1302	toys, outdoor playing, clothes
2585	gardening and DIY
50	video games accessories
1320	early childhood
2220	supplies for domestic animals
2462	games

Texte - Simple DNN	
Classe produit	Labels
10	adult books
1180	figurines, masks and role playing games
1281	board games
40	imported video games
2705	books
2280	magazines

1280	toys for children
2582	furniture kitchen and garden
2403	children books and magazines
1140	figurines and Toy Pop
2585	gardening and DIY
1320	early childhood

Les classes de produits les moins bien prédites sont globalement les mêmes pour les deux modèles Texte, sauf pour les classes de produits: 1302, 50, 2220 et 2462 qui sont mieux prédites par le modèle Simple DNN

Note : Voir Annexe 4.1.2.2.1 et Annexe 4.1.2.2.1 pour l'analyse des Matrices de Confusion des deux modèle Simple DNN et Conv1D.

4.1.3 Conclusion

Les modèles **Conv1D** et **DNN** (Deep Neural Network) donnent les meilleurs scores. Le modèle **LinearSVC** a également donné des résultats de scoring intéressants. Enfin, nous avons constaté que le modèle Conv1D présente la particularité d'être beaucoup plus rapide à l'entraînement.

4.2 MODÉLISATION PARTIE IMAGES

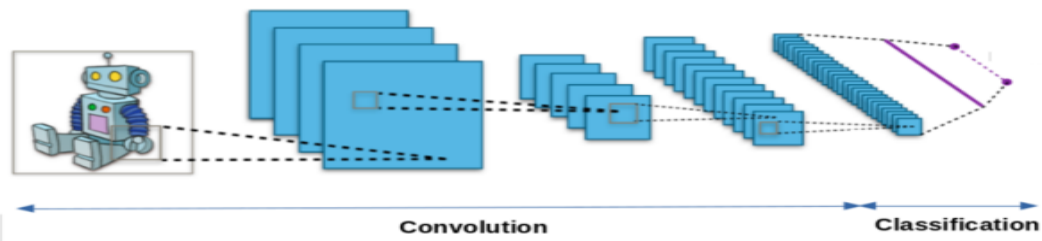
Sachant la complexité de notre jeu de données images et les ressemblances qui existent entre les différentes classes , nous avons rapidement opté pour l'utilisation des méthodes d'apprentissage par transfert(transfert learning) avec des modèles reposant les réseaux de neurones artificiels, capables d'apprentissage profond (Deep Learning). Nous avons ainsi utilisé les modèles les plus adaptés aux images : les réseau de neurones convolutifs (en anglais CNN ou ConvNet pour Convolutional Neural Networks).

4.2.1 Concepts et définitions

4.2.1.1 Les réseaux de neurones convolutifs

Les réseaux de neurones convolutifs sont directement inspirés du cortex visuel des vertébrés. Un réseau de neurones à convolution, appelé aussi convnet (pour «Convolutional Network»), ou encore CNN (pour «Convolutional Neural Network») .

On distingue deux parties, une première partie que l'on appelle la partie convolutive du modèle et la seconde partie, que l'on va appeler la partie classification du modèle([Réf](#))



La première partie de convolution fonctionne comme un extracteur de features en appliquant des opérations de filtrage par convolution. La seconde est une succession de dense layer qui a pour but de classifier.

4.2.1.2 *Le Transfert Learning*

Principe :

Le Transfer Learning, ou apprentissage par transfert en français, désigne l'ensemble des méthodes qui permettent de transférer les connaissances acquises à partir de la résolution de problèmes donnés pour traiter un autre problème.

En effet , construire son propre réseau CNN de A à Z peut s'avérer long et fastidieux . De plus, il nécessiterait des temps de calculs élevés et des ressources importantes. C'est pourquoi , l'idée d'utiliser les connaissances acquises par un modèle pré-entraîné semble la bonne stratégie pour résoudre efficacement des problèmes complexes de classification d'images.

4.2.2 Stratégie

4.2.2.1 *Choix des modèles pré-entraînés*

Parmi la listes des modèles présentés [ici](#), nous avons retenu les modèles qui ont le moins de paramètres afin de réduire les durées d'entraînements et limiter les contraintes liées au manque de ressources de calcul.

- Voici la Listes de modèles choisis :

Modèles	Nombre de paramètres
MobileNetV2	3538984
NASNetMobile	5326716
DenseNet121	8062504
Xception	22910480
InceptionV3	23851784
ResNet50(Modèle de référence)	25636712
InceptionResNetV2	55873736
EfficientNetB7	66658687
VGG16	138357544
VGG19	143667240

4.2.2.2 Séparation du jeu données

Nous avons utilisés l'ensemble des images, soit 84 916 images avec les proportions suivantes: 80 % pour le training set et 20 % pour le testing set, soit 67933 images pour l'entraînement et 16983 pour le test.

4.2.2.3 Choix de métriques d'évaluation

Etant donné le déséquilibre entre les classes et comme est préconisé dans le challenge Rakuten, nous avons opté pour l'utilisation de la métrique f1-score et notamment le f1-score weighted comme critère d'évaluations des résultats de nos modèles. Nous avons également fait le choix de relever l'accuracy car le déséquilibre entre les classes n'est pas si extrême.

- Le **F1-score** est une métrique pour évaluer la performance des modèles de classification à 2 classes ou plus. Il est particulièrement utilisé pour les problèmes utilisant des données déséquilibrées. Il peut être interprété comme une moyenne pondérée de la précision et du rappel

Précision : Elle permet de connaître le nombre de prédictions positifs bien effectuées

Rappel (recall en anglais) : correspond à proportion d'individus positifs effectivement bien détectés par le classifieur

Remarques :

- Le **F1-score weighted average**(Le F1-score moyen pondéré) est calculé en prenant la moyenne de tous les F1-scores par classe tout en tenant compte du nombre d'occurrences réelles de chaque classe dans le jeu de données. Cette mesure est préférable dans le cas où l'on souhaite attribuer une plus grande contribution aux classes avec le plus d'observations dans l'ensemble des données.

- Autrement , dans le cas de jeu de données déséquilibré où toutes les classes sont d'importance égale, l'utilisation de la mesure ***F1-score macro average*** serait un bon choix car elle traite toutes les classes de la même manière.

4.2.2.4 Paramètres & Étapes

4.2.2.4.1 Paramètres

- Augmentation de données appliquée aux images du jeu de données d'entraînement :

Mise à l'échelle des pixels entre 0 et 1	rescale = 1/255
Étirement de l'image selon un certain angle. Donne à l'image un aspect plus écrasé ou étiré en diagonale	shear_range=0.2
Plage de rotation à rotation aléatoire [-45, 45]	rotation_range = 45
Mouvement de l'image dans le sens horizontal par rapport à la largeur de l'image	width_shift_range = 0.1
Mouvement de l'image dans le sens vertical par rapport à la hauteur de l'image	height_shift_range = 0.1
Plage de zoom : zoom arrière et zoom avant [-20%, +20%]	zoom_range = 0.2
Retournement de l'image horizontalement	horizontal_flip = True
Retournement de l'image verticalement	vertical_flip = True

- Optimiseur : Adam est un optimiseur adaptatif éprouvé et très populaire
- Fonction de perte: la « *Categorical Cross-Entropy* » adaptée à notre cas de classification multi-classes
- Batch size : 32 et 64 ont été testés
- Learning rate (LR) : modèles entraînés avec à LR à 0.001 (valeur par défaut), ensuite un de très faible valeur à 0.00001. Et enfin avec les valeurs obtenues à l'issue de l'étape de l'optimisation des Learning Rate (Etape 4)

4.2.2.4.2 Étapes

La stratégie employée est la suivante :

Etape 1 : Selection des modèles

- Entraîner les modèles pré-entraînés sélectionnés avec des images d'entrées de différentes tailles, et ce sans augmentation de données ni fine-tuning
- Utiliser le Learning rate(LR) par défaut (0.001)
- Varier le batch size de 32 à 64

- Varier le nombre d'Epoch de 10, 20 à 40
- Noter les 5 meilleurs modèles selon la mesure F1-score weighted

Etape 2 : Augmentation des données avec les meilleurs modèles obtenus de l'Etape 1

- Entraîner les 5 meilleurs modèles de l'Etape 1 avec augmentation des données (sans fine-tuning et LR toujours par défaut à 0.001)

Etape 3 : Augmentation des données + fine tuning des modèles

- Fine tuning des meilleurs modèles obtenus à l'Etape 1
- Utilise un LR très faible (0.00001)
- Et augmentation du nombre d'Epochs de 40 , 50 à 60

Etape 4: Optimization LR (Sans augmentation des données ni fine-tuning)

- Nous avons suivi la méthode décrite dans [cet article](#):

En utilisant Le callback LearningRateScheduler :

- Commencez l'entraînement avec un très faible LR (0,00001) puis augmenter sa valeur de manière exponentielle au fur à mesure des Epochs jusqu'à atteindre $\approx 2,54$
- Etudier les courbes des résultats (perte et accuracy en fonction du LR et des Epochs)
- Identifier le LR qui réalise la plus petite perte (correspond en général à la plus grande accuracy), à condition que les valeurs qui l'entourent ne soient pas trop volatiles

Etape 5: Augmentation des données , fine-tuning avec les LR optimisés

- Ré-entraîner les 5 modèles retenus avec les LR optimaux identifiés à l'étape précédente

4.2.2.5 Résultats

4.2.2.5.1 Étape1 : Entraînement des modèles pré-sélectionnés

Model	Optimal model input_shape	Image size input	LR(Default)	Nbr Epoch	Batch size	Accuracy	F1 score weighted
InceptionResNetV2	299x299	500x500	0.001	40	64	0,62	0,61
DenseNet121	224x224	256x256	0.001	40	64	0,61	0,61
Xception	299x299	500x500	0.001	40	64	0,61	0,60
InceptionV3	299x299	500x500	0.001	40	64	0,60	0,59
MobileNetV2	224x224	256x256	0.001	40	64	0,59	0,59
MobileNetV2	224x224	256x256	0.001	20	32	0,59	0,58
VGG16	224x224	256x256	0.001	40	64	0,58	0,57
VGG16	224x224	256x256	0.001	20	64	0,58	0,57
InceptionResNetV2	224x224	256x256	0.001	40	64	0,58	0,57
NASNetMobile	224x224	256x256	0.001	40	64	0,58	0,56
Xception	224x224	256x256	0.001	40	64	0,58	0,56
InceptionV3	256x256	256x256	0.001	40	64	0,57	0,56
VGG19	224x224	256x256	0.001	40	64	0,57	0,56
VGG16	128x128	256x256	0.001	10	32	0,54	0,53
ResNet50	128x128	500x500	0.001	50	32	0,33	0,29
ResNet50	256x256	256x256	0.001	40	64	0,33	0,29
EfficientNetB7	224x224	256x256	0.001	40	64	0,12	0,03

Les 5 meilleurs modèles selon F1-score weighted sont :

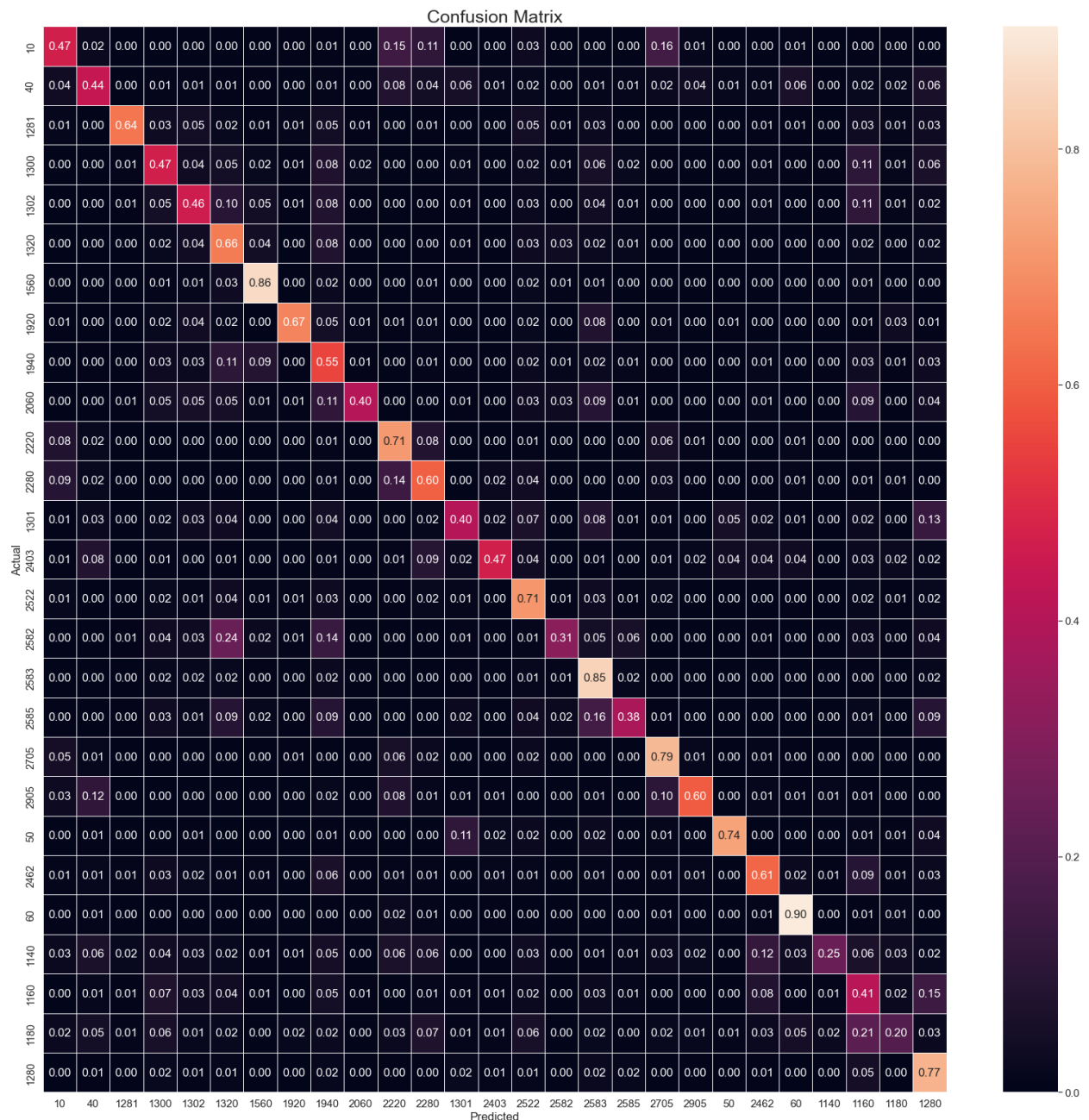
Model	Optimal input_shape model	Image size input	LR (Default)	Nbr Epoch	Batch size	Accuracy	F1 score weighted
InceptionResNetV2	299x299	500x500	0.001	40	64	0,62	0,61
DenseNet121	224x224	256x256	0.001	40	64	0,61	0,61
Xception	299x299	500x500	0.001	40	64	0,61	0,60
InceptionV3	299x299	500x500	0.001	40	64	0,60	0,59
MobileNetV2	224x224	256x256	0.001	40	64	0,59	0,59

(F1-score par classe, par catégorie et par modèle)

Classe produit	Catégorie	InceptionResNetV2	DenseNet121	Xception	InceptionV3	MobileNetV2
1180	figurines, masks and role playing games	0,27	0,29	0,25	0,23	0,23
1140	figurines and Toy Pop	0,33	0,39	0,37	0,32	0,34
2582	Furniture kitchen and garden	0,41	0,42	0,46	0,43	0,41
1160	Playing cards	0,43	0,42	0,45	0,41	0,41
1300	Remote controlled models	0,44	0,43	0,44	0,38	0,35
1301	Accessories children	0,45	0,36	0,39	0,34	0,30
2585	gardening and DIY	0,45	0,44	0,44	0,38	0,38
10	adult books	0,49	0,51	0,48	0,54	0,50
40	Imported video games	0,49	0,53	0,48	0,42	0,49
1302	toys, outdoor playing, clothes	0,49	0,47	0,48	0,44	0,46
2060	Decoration interior	0,49	0,41	0,40	0,37	0,34
1940	Food	0,52	0,49	0,47	0,47	0,47
2403	children books and magazines	0,55	0,51	0,52	0,45	0,50
1320	Early childhood	0,59	0,58	0,58	0,59	0,56
2905	online distribution of video games	0,59	0,64	0,52	0,59	0,57
2280	magazines	0,62	0,62	0,60	0,61	0,62
2462	games	0,62	0,63	0,63	0,55	0,59
1920	Interior accessories	0,66	0,62	0,67	0,66	0,60
1281	Board games	0,67	0,59	0,66	0,61	0,64
2522	stationery	0,67	0,66	0,68	0,68	0,66
2220	supplies for domestic animals	0,68	0,67	0,67	0,67	0,68
1280	toys for children	0,69	0,69	0,68	0,65	0,64
2705	books	0,69	0,73	0,68	0,70	0,66
50	Video games accessories	0,70	0,71	0,70	0,61	0,64
1560	Interior furniture and bedding	0,81	0,79	0,76	0,75	0,78
2583	piscine spa	0,83	0,83	0,82	0,81	0,81
60	games and consoles	0,88	0,89	0,87	0,87	0,89

- Les classes de produits les moins bien prédites (F1-score < 0.55) sont les mêmes pour tous les modèles. Il s'agit de classes et catégories suivantes :

- Analyse de la Matrice de Confusion - **InceptionResNetV2**



- Classe 10 (adult books)
 - ➔ confondue avec la classe 2705 (books) dans **16%** des cas
 - ➔ confondue avec la classe 2220(supplies for domestic animals) dans **15%** des cas
 - ➔ confondue avec la classe 2280 (magazines) dans **11%** des cas
- Classe 40 (imported video games)

Confondue pratiquement avec toutes les classes avec des petits pourcentages :

 - ➔ dans 6 à 8 % des cas avec chacune des classes: 10, 2280, 2905, 1301, 60, 1280 et 2220.
 - ➔ dans 1 à 2% avec chacune des classes: 1300, 1302, 1320, 1940, 2403, 2583, 2585, 50, 2462, 1160 et 1180
- Classe 1300 (remote controlled models)

- ➔ confondue avec la classe 1160 dans **11%** des cas
- ➔ dans 4 à 8% avec chacune des classes: 1940, 1302, 1320, 2583, 1280
- ➔ dans 1 à 2% avec chacune des classes: 1281, 1560, 1920, 2060, 1301, 2522, 2582, 2585, 2462, 1180
- Classe 1302 (toys, outdoor playing, clothes)
 - ➔ dans 11% des cas avec la classe 1160
 - ➔ dans 4 à 8% avec chacune des classes: 2583, 1300 et 1940
- Classe 1940 (Food)
 - ➔ dans 11% des cas avec la classe 1320
 - ➔ dans 9% des cas avec la classe 1560
 - ➔ dans 4 à 8% avec chacune des classes: 1300, 1302 et 1940
 - ➔ dans 1 à 3% des cas avec chacune des classes 2060, 2280, 2522, 2582, 2583, 2585, 2462, 1160, 1180 et 1280
- Classe 2060 (decoration interior)
 - ➔ dans 11% des cas avec la classe 1940
 - ➔ dans 9% des cas avec chacune des classes 2583 , 1160
 - ➔ dans 4 à 5% avec chacune des classes : 1280, 1320, 1302, 1300
- Classe 1301 (accessories children)
 - ➔ dans 13% des cas avec la classe 1280
 - ➔ dans 4 à 8% avec chacune des classes: 50, 2583, 2522, 1940, 1320
- Classe 2403 (children books and magazines)
 - ➔ dans 8 à 9% des cas chacune des classes 40 et 2280
 - ➔ dans 4 % avec chacune des classes: 60, 2462, 50, 2522
- Classe 2582 (furniture kitchen and garden)
 - ➔ dans **24%** des cas avec la classe 1320
 - ➔ dans **14 %** des cas avec la classe 1940
- Classe 2585 (gardening and DIY)
 - ➔ dans **16%** des cas avec la classe 2583
 - ➔ dans **9 %** des cas avec chacune des classes 1940 et 1320
- Classe 1140 (figurines and Toy Pop)
 - ➔ dans **12%** des cas avec la classe 2462
 - ➔ dans **3 à 6 %** des cas avec chacune des classes 40, 1300, 1302, 2220, 2280, 2522, 2705, 60, 1160, 1180
- Classe 1160 (playing cards)

- ➔ dans **15%** des cas avec la classe 1280
- ➔ dans **7 à 8 %** des cas avec chacune des classes 1300, 2462
- Classe 1180 (figurines, masks and role playing games)
 - ➔ dans **21%** des cas avec la classe 1160
 - ➔ dans **5 à 7 %** des cas avec chacune des classes 40, 1300, 2280, 2522, 60,

4.2.2.5.2 Étape 2 : Augmentation des données avec les meilleurs modèles obtenus Étape 1

Model	input_shape model	Images size input	LR (default)	Nbr Epoch	Batch size	Accuracy	F1 score weighted
InceptionResNetV2	299x299	500x500	0,001	40	64	0.59	0,58
Xception	299x299	500x500	0,001	40	64	0,59	0,58
DenseNet121	224x224	256x256	0,001	40	64	0,58	0,57
InceptionV3	299x299	500x500	0,001	40	64	0,58	0,56
MobileNetV2	224x224	256x256	0,001	40	64	0,56	0,55

- L'augmentation des données ne semble pas apporter une amélioration au niveau des performances, voire les dégrade un peu ! Cependant, ils pourraient mieux se généraliser sur des nouveaux jeu de données

4.2.2.5.3 Étape 3 : Augmentation des données + fine tuning des modèles

- Décongélation de certaines couches de nos modèles pré-entraînés

Model	Nbr unfrozen layers	Input shape model	Images size input	LR	Nbr Epoch	Batch size	Accuracy	F1 score weighted
Xception	60	299x299	500x500	0,00001	40	64	0,66	0,65
InceptionV3	100	299x299	500x500	0,00001	50	64	0,64	0,64
MobileNetV2	all base model	224x224	256x256	0,00001	50	64	0,63	0,63
InceptionResNetV2	8	299x299	500x500	0,00001	50	64	0,64	0,63
DenseNet121	4	224x224	256x256	0,00001	50	64	0,59	0,59

- Nous observons une amélioration au niveau de tous les modèles, sauf pour le MobileNetV2
- Xception passe en tête avec 0.65 en F1-score weighted avec une amélioration de 5% par rapport à son score obtenu à l'étape 1

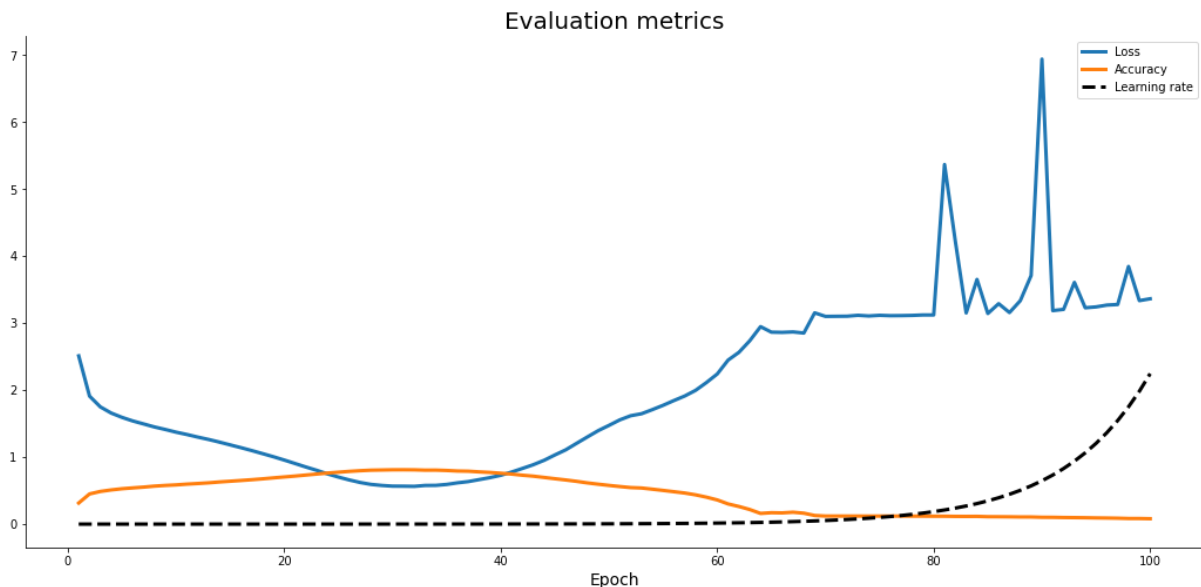
4.2.2.5.4 Étape 4 : Optimisation LR (Sans augmentation des données ni fine-tuning)

Dans cette partie nous avons suivi la méthode d'optimisation du LR [décrite ici](#) (avec l'utilisation du callback LearningRateScheduler), qui consiste à :

- Commencez l'entraînement avec un très petit LR : 0,00001 puis augmenter sa valeur de manière exponentielle au fur et à mesure des Epochs jusqu'à atteindre $\approx 2,54$

L'idée est de sélectionner un taux d'apprentissage qui réalise la perte la plus faible, à condition que les valeurs qui l'entourent ne soient pas trop volatiles.

Exemple : Xception :

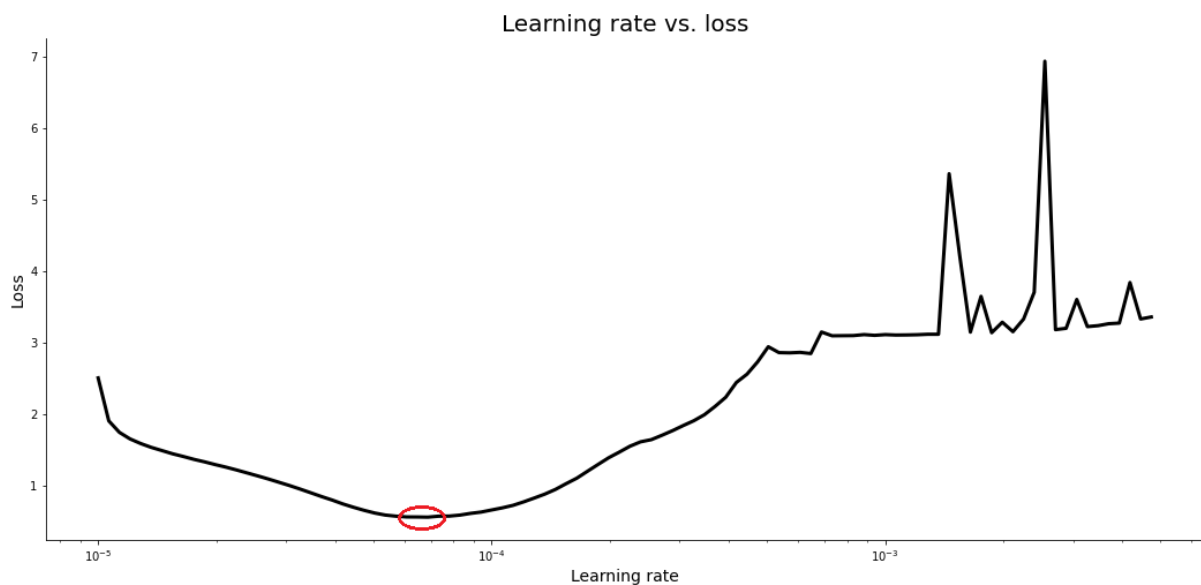


L'accuracy :

- L'accuracy augmente légèrement jusqu'à l'Epoch 30
- Commence ensuite à baisser jusqu'à l'Epoch 63
- Elle fluctue très légèrement entre l'Epoch 64 et 67
- S'aplatie ensuite à partir de l'Epoch 68 et ce jusqu'à la fin de l'entraînement

La perte :

- L'exact opposé est arrivé à la perte jusqu'à l'Epoch 63 (Ce qui est logique)
- Fluctue légèrement entre l'Epoch 64 et 69
- S'aplatie pratiquement entre l'Epoch 70 et 80
- Fluctue ensuite significativement à partir de l'Epoch 80 et jusqu'à la fin de l'entraînement



Le taux d'apprentissage optimal(avec la plus faible perte) est d'environ 0,0004.

Voici les LR obtenus pour nos différents modèles:

Model	LR Obtenu
Xception	0,0004
InceptionV3	0.0003
MobileNetV2	0.0003
InceptionResNetV2	0.0003
DenseNet121	0.0004

4.2.2.5.5 Étape 5 : Augmentation des données, fine-tuning avec les LR optimisés

Model	Nbr unfrozen	LR	input_shape model	images	Nbr Epoch	Batch size	Accuracy	F1 score weighted
	Layers			size folder train				
Xception	60	0,0004	299x299	500x500	100	64	0,66	0,66
InceptionV3	100	0.0003	299x299	500x500	100	64	0,65	0,65
InceptionResNetV2	8	0.0003	299x299	500x500	100	64	0,62	0,62
DenseNet121	10	0.0004	224x224	256x256	100	64	0,60	0.59
MobileNetV2	all base model	0.0003	224x224	256x256	100	64	0,58	0,57

- Pas d'amélioration significative observée (Seulement 1% de plus en au niveau du F1-score weighted pour le modèle Xception et 1% de plus en accuracy pour le modèle InceptionV3). Les deux modèles semblent avoir atteint leur limites avec ce jeu de données Images !
- Par contre, ils atteignent ces performances beaucoup plus rapidement avec les LR optimisés Ci-après les durées d'entraînement observées pour les modèles Xception, InceptionV3 avec LR à 0,00001 et à 0,0003 :

Modèle	Learning Rate	Durée de l'entraînement
Xception	0,00001	10h 41min 21s

InceptionV3	0,00001	7h 19min 16s
Xception	0,0004	4h 39min 26s
InceptionV3	0.0003	5h 11min 15s

- Ci-dessous les classes les moins bien prédites avec les deux modèles Xception et InceptionV3(F1-score inférieur à celui du modèle de référence (0.55) :

Images – Xception, LR = 0,0004		
Classe Produit	Labels	f1-score
1281	board games	0,34
2220	supplies for domestic animals	0,39
1180	figurines, masks and role playing games	0,40
2582	furniture kitchen and garden	0,42
1280	toys for children	0,47
50	video games accessories	0,47
2585	gardening and DIY	0,48
1302	toys, outdoor playing, clothes	0,49
1281	early childhood	0,52

Images - InceptionV3, LR = 0.0003		
Classe Produit	Labels	f1-score
1281	board games	0,32
50	video games accessories	0,41
1180	figurines, masks and role playing games	0,41
2220	supplies for domestic animals	0,41
1280	toys for children	0,46
2582	furniture kitchen and garden	0,46
1302	toys, outdoor playing, clothes	0,47
2585	gardening and DIY	0,47
1940	Food	0,50
1320	early childhood	0,51

(Voir Annexe 4.2.2.5.5 pour plus de détails)

- A l'exception de la classe 2060 , il s'agit des mêmes classes qui sont les moins bien prédites avec les deux modèles Images
- De manière générale, la définition relativement imprécise des 27 classes, présentant des objets avec de fortes similarités visuelles inter classes, complexifie et limite notablement les possibilités de la phase d'apprentissage

4.2.2.5.6 Conclusion

- Les 2 meilleurs résultats ont été obtenus avec :

➔ Xception : F1-score weighted à 0.66, LR à 0,0004

➔ Inception : F1-score weighted 0.64 , LR à 0,0003

Avec augmentation des données et fine-tuning

Ces résultats dépassent largement ceux annoncés dans le modèle de référence Rakuten(0.55)

5 CLASSIFICATION BIMODALE

Dans cette partie, nous abordons le défi principal du challenge Rakuten , à savoir, une classification multimodales(texte et image).

L'idée consiste à combiner de multiples modèles de base en un seul modèle, et d'obtenir de meilleures performances prédictives supérieures à celles de ces modèle pris individuellement. La combinaison de deux modèles d'apprentissage, autrement connu sous le nom d'apprentissage d'ensemble, comporte de nombreuses techniques. Parmi les plus citées dans la littérature, le Voting, le Stacking.

Dans le cadre de notre problème de classification, nous avons utilisé l'approche de Voting (Max Voting et Weighted Average Voting) comme suit :

- Max Voting : choisir la prédiction avec la probabilité la plus élevée parmi les modèles combinés
- Weighted Average : associer un poids à chaque modèle selon son f1-score

5.1 CHOIX DES MEILLEURS MODÈLES

Parmi l'ensemble des modèles Texte et Images testés, les meilleures performances ont été obtenues avec les modèles suivants:

- **Partie Texte :**

- Embedding + Conv1D : F1-score weighted et une accuracy de 0.80
- Simple DNN : F1-score weighted et une accuracy de 0.81

Partie Images

- Xception: F1-score weighted accuracy à 0.66
- InceptionV3 : F1-score weighted et accuracy à 0.64

5.2 COMBINAISONS UTILISÉES - TEXTE ET IMAGES

Nous avons utilisé deux combinaisons pour le système de vote :

- Un premier modèle qui combine les deux modèles prédictifs basés sur le texte : Simple DNN et Conv1D avec le modèle d'images Xception
- Et un deuxième en utilisant les mêmes modèles texte : Simple DNN, Conv1D avec le modèle d'image InceptionV3

5.2.1 Simple DNN, Conv1D et Xception

Les poids associés pour le vote pondéré (Weighted Average) aux différents modèles sont :

- Simple DNN: **0.81**
- Conv1D: **0.80**
- Xception: **0.65**

Voting : Xception, Conv1D et Simple DNN

Classe produit	Labels	Max Voting f1-score	Weighted Average Voting f1-score	Nombre d'observations
1281	board games	0,55	0,58	408
1180	figurines, masks and role playing games	0,58	0,59	144
10	adult books	0,69	0,69	643
1280	toys for children	0,72	0,74	1002
2582	furniture kitchen and garden	0,72	0,75	533
1302	toys, outdoor playing, clothes	0,76	0,78	514
40	imported video games	0,78	0,79	503
2220	supplies for domestic animals	0,79	0,79	190
2585	gardening and DIY	0,79	0,79	468
2060	decoration interior	0,78	0,80	970
1320	early childhood	0,78	0,81	664
2462	games	0,78	0,81	301
50	video games accessories	0,79	0,81	319
1140	figurines and Toy Pop	0,82	0,82	545
1560	interior furniture and bedding	0,82	0,83	993
2403	children books and magazines	0,82	0,83	961
2280	magazines	0,83	0,83	911
1940	Food	0,84	0,85	159
60	games and consoles	0,87	0,87	179
2705	books	0,89	0,89	557
1301	accessories children	0,89	0,90	162
1920	interior accessories	0,90	0,91	832
2522	stationery	0,91	0,92	1011
2905	online distribution of video games	0,92	0,93	146
1300	remote controlled models	0,93	0,93	993
2583	piscine spa	0,95	0,95	2083
1160	playing cards	0,96	0,96	793

accuracy	0,83	0,84	16984
macro avg	0,81	0,82	16984
weighted avg	0,83	0,84	16984

Résultats

- Le score F1-score weighted obtenu est de **0.84**
- Le Max Voting et le Weighted Average Voting donnent pratiquement les mêmes performances (avec 1% de plus en macro Avg pour Weighted Average Voting)
- Une amélioration est observée pour les classes de produits 2705, 40, 2280, 2403, 1140, 50, 1320, 2220, 2462, 2060 (F1-score est maintenant > 0.8)

- Les classes de produits 1281, 1180, 10, 1280, 2582, 1302, 2585 sont les moins bien prédites (F1-score < 0.8)

5.2.2 Simple DNN, Conv1D et InceptionV3

Les poids associés pour le vote pondéré(WeightedAverage) aux différents modèles sont:

- Simple DNN: **0.81**
- Conv1D: **0.80**
- InceptionV3: **0.64**

Résultats

Voting: Inception, Conv1D et Simple DNN				
Classe produit	Labels	Max Voting f1-score	Weighted Average Voting f1-score	Nombre d'observations
1281	board games	0,57	0,59	408
1180	figurines, masks and role playing games	0,59	0,60	144
10	adult books	0,66	0,66	643
1280	toys for children	0,73	0,73	1002
2582	furniture kitchen and garden	0,76	0,76	533
2220	supplies for domestic animals	0,77	0,79	190
1940	Food	0,75	0,79	159
2585	gardening and DIY	0,78	0,80	468
1320	early childhood	0,79	0,80	664
2060	decoration interior	0,79	0,80	970
40	imported video games	0,80	0,80	503
1140	figurines and Toy Pop	0,79	0,80	545
1302	toys, outdoor playing, clothes	0,80	0,81	514
50	video games accessories	0,79	0,82	319
2403	children books and magazines	0,82	0,82	961
2280	magazines	0,82	0,82	911
2462	games	0,82	0,83	301
1560	interior furniture and bedding	0,83	0,84	993
2705	books	0,88	0,88	557

1920	interior accessories	0,88	0,89	832
60	games and consoles	0,90	0,91	179
1301	accessories children	0,92	0,92	162
2522	stationery	0,91	0,92	1011
2905	online distribution of video games	0,91	0,93	146
1300	remote controlled models	0,93	0,93	993
2583	piscine spa	0,96	0,96	2083
1160	playing cards	0,96	0,59	793

accuracy	0,84	0,84	16984
macro avg	0,81	0,82	16984
weighted avg	0,83	0,84	16984

- Le score F1-score weighted obtenu est de **0.84**
- Le Max Voting et Weighted Average Voting donnent pratiquement les mêmes performances (avec 1% de plus en macro Avg pour le Weighted Average Voting)
- Une amélioration est observée pour les classes de produits 2705, 2280, 2403, 1140, 1302, 50, 1320, 2462, 2060 (F1-score est maintenant > 0.8)
- Les classes de produits 1281, 1180, 10, 1280, 2582, 1302, 2585 sont les moins bien prédites (F1-score < 0.8)

5.3 CONCLUSION

- Le meilleur résultat(de **0.84**) est observé avec la combinaison des trois modèles Xception/Conv1D/Simple DNN en réalisant un Voting weighted
- Pas de différence très significative entre les deux combinaisons Xception/Conv1D/Simple DNN et InceptionV3/Conv1D/Simple DNN

5.4 PROBLÈMES RENCONTRÉS

Parmi les contraintes et les difficultés rencontrées, nous citons:

- Overfitting des algorithmes de machines learning,
- Limitation de puissance de calcul des ordinateurs et notamment pour les modèles de Deep Learning. Nous avons utilisé « Google Colab » mais ce dernier ne nous a pas

permis d'avoir plus de puissance. De plus, les sessions de travail ne sont que de 90 minutes, ce qui interrompait nos exécutions,

- Certaines classes présentent des images avec des fortes ressemblances, ce qui rend l'apprentissage plus complexe,
- Gestion des variables cibles lorsqu'elles ne sont dichotomisées (`class_mode = 'sparse'`) sous le générateur de données (`ImageDataGenerator`) :
 - L'itérateur d'images change les numéros de classes et leur assigne des nouveaux labels de manière alphanumérique. Les prédictions/probas sont faites ensuite avec ces labels. Cela a posé un problème lors de la partie Bimodale/Voting pour faire correspondre les probabilités/classes prédites par les modèles CNN et les modèles Texte. Cela a nécessité un traitement supplémentaire pour palier à ce problème
 - En théorie, on devrait être capable de fixer ces labels avec les mêmes numéros de classe d'entrée. Le problème est connu (voir [ticket](#)) et serait corrigé sur les dernières versions Tensorflow + Keras 2.7. Cependant, nous n'avons pas pu le faire bien que nous soyons dans cette version 2.7

5.5 PISTES D'AMÉLIORATION

- Préprocessing des images qui présentent un fond blanc important, comme par exemple, remplacer les bords blancs par du noir, effectuer des zooms, etc
- Améliorer la partie augmentation des données images en appliquant d'autres transformations comme par exemple la luminosité, crop, modifier les couleurs etc
- Gérer le déséquilibre de notre jeu de données
- Implémenter d'autres approches d'ensembling pour la classification bimodale telle que le concatenate

6 BILAN

Ce projet a été un excellent moyen de mettre en pratique les différentes connaissances acquises en cours. Nous avons ainsi pu aborder les méthodes de Machine et Deep Learning, de la Computer Vision(CV), *du* Natural Language Processing(NLP) , de la DataViz et de la Classification Multiclass & Multimodale.

En terme de métier de Data Scientist, nous avons pu réaliser les étapes d'un projet Data :

1. Définition du problème ,
2. Acquisition des données,
3. Préparer et nettoyer les données,
4. Analyser, explorer et visualiser les données,
5. Choisir un modèle d'apprentissage,
6. Evaluer les résultats, et ajuster ou modifier le modèle d'apprentissage

Les challenges que nous avons rencontrés pendant le projet ont été les suivants :

- La taille des données Image nécessitait des équipements assez puissants en mémoire et en GPU. Google Colab n'a pas été d'une grande utilité(de très faibles performances, voire inférieures à celles de nos PCs + les sessions de travail qui s'interrompent après un délai d'inactivité de 90 minutes)
- La différence entre certaines classes n'était pas évidente, même à l'oeil humain.
- Les modèles de Deep Learning pour la classification d'Images ont des temps d'entraînement très important et souvent avec peu d'amélioration en terme de performance

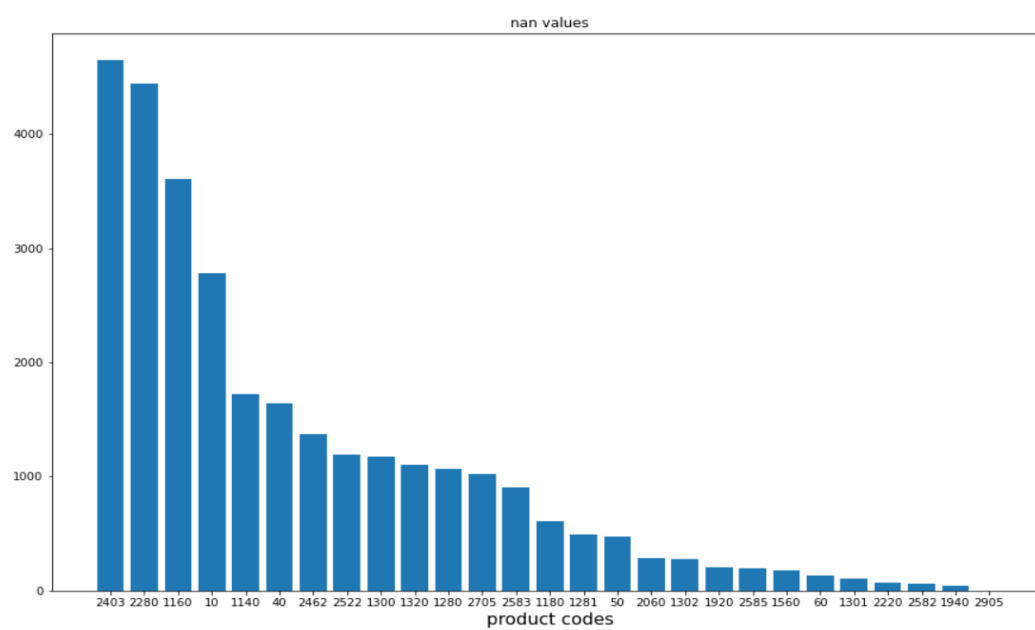
Sur le plan technique du projet, l'objectif a été atteint et les performances obtenues dépassent celles du modèles de référence, notamment celles obtenues avec la classification d'Image(+10 % par rapport au benchmark).

Nous comptons soumettre nos résultats et voir combien nous serons classés parmi les équipes ayant participé à ce challenge.

Si le temps le permet , nous pensons améliorer nos résultats d'ici la soutenance, prévue pour le 22 mars. Nous pourrions utiliser d'autres méthodes de Classification Bimodale comme par exemple le concatenate

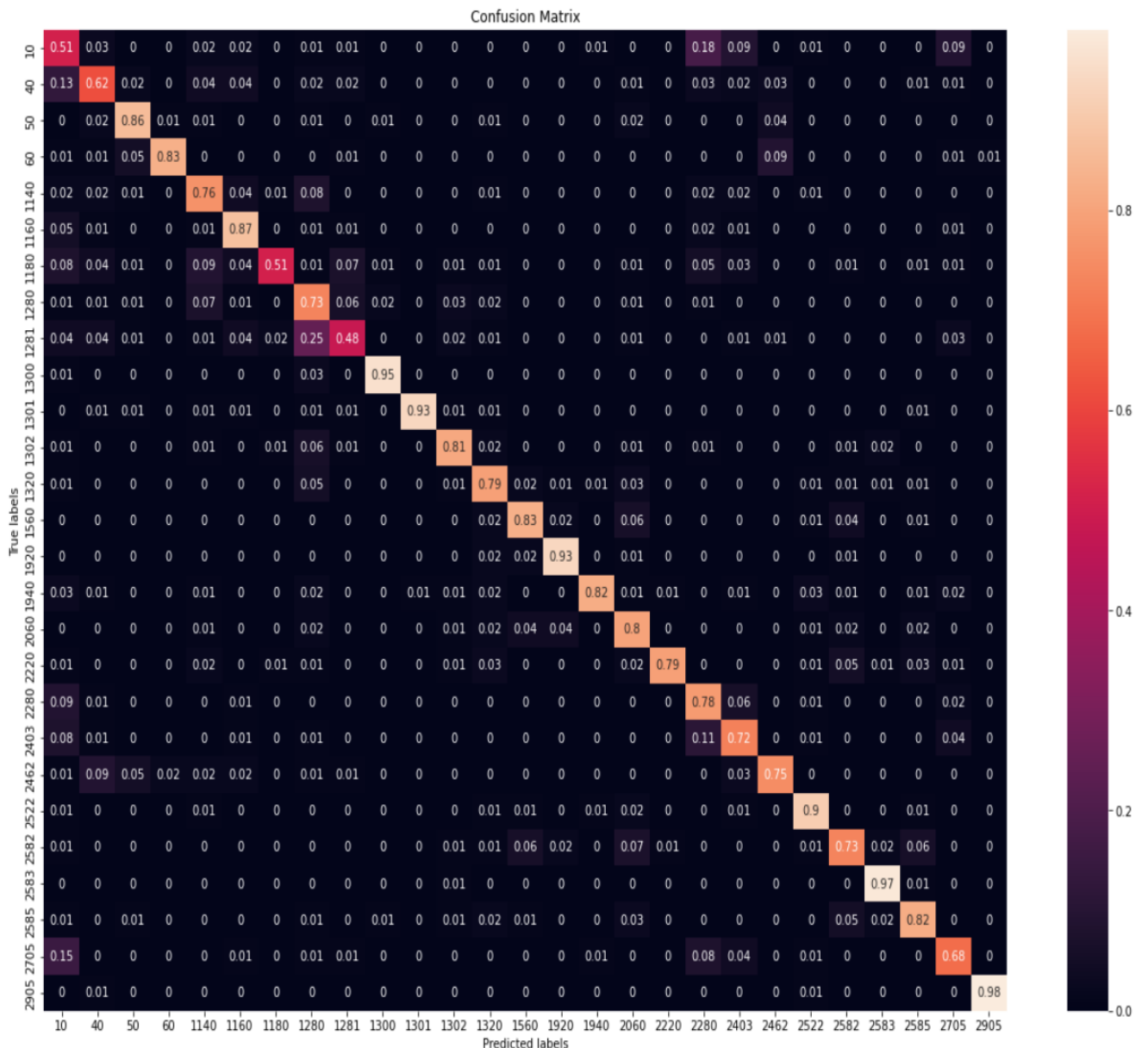
7 ANNEXES

7.1 ANNEXE 3.3.1 - RÉPARTITION DES « NANS » PAR CLASSE DE PRODUIT



7.2 ANNEXE 4.1.2.2 - ANALYSE MATRICES DE CONFUSION -TEXTE

7.2.1 Annexe 4.1.2.2.1 - Simple DNN



Classe 10 (adult books)

- ➔ confondue avec la classe 2705(books) dans **13%** des cas
- ➔ confondue avec la classe 40 (imported video games) dans **10%** des cas
- ➔ confondue avec la classe 2403 (children books and magazines) dans **12%** des cas
- ➔ confondue avec la classe 2280 (magazines) dans **14%** des cas

Classe 40 (imported video games)

- ➔ confondue avec la classe 2462 (games) dans **5%** des cas

Classe 1140 (figurine and toy pop)

- ➔ confondue avec la classe 1280 (toys for children) dans **11%** des cas
- ➔ confondue avec la classe 1180 (figurine, masks and role playing games) dans **2%** des cas

Classe 1280 (toys for children)

- ➔ confondue pratiquement avec toutes les classes avec des petits pourcentages :
- ➔ confondue avec la classe 1281 (board games) dans 10% des cas
- ➔ confondue avec la classe 1302 (toys, outdoor playing and clothes) dans 3% des cas
- ➔ confondue avec la classe 1140 (figurine and toy pop) dans 4% des cas

Classe 1281 (board games)

- ➔ confondue avec la classe 1280 (toys for children) dans **12%** des cas

Classe 1560 (interior furniture and bedding)

- ➔ confondue avec la classe 2582 (furniture, kitchen and garden) dans **2%** des cas

Classe 2060 (decoration interior)

- ➔ confondue avec la classe 2582 (furniture, kitchen and garden) dans **3%** des cas
- ➔ confondue avec la classe 1560 (interior furniture and bedding) dans **5%** des cas

Classe 2280 (magazines)

- ➔ confondue avec la classe 2705 (books) dans **4%** des cas
- ➔ confondue avec la classe 2403 (children books and magazines) dans **11%** des cas
- ➔ confondue avec la classe 10 (adult books) dans **11%** des cas

Classe 2403 (children books and magazines)

- ➔ confondue avec la classe 10 (adult books) dans **5%** des cas

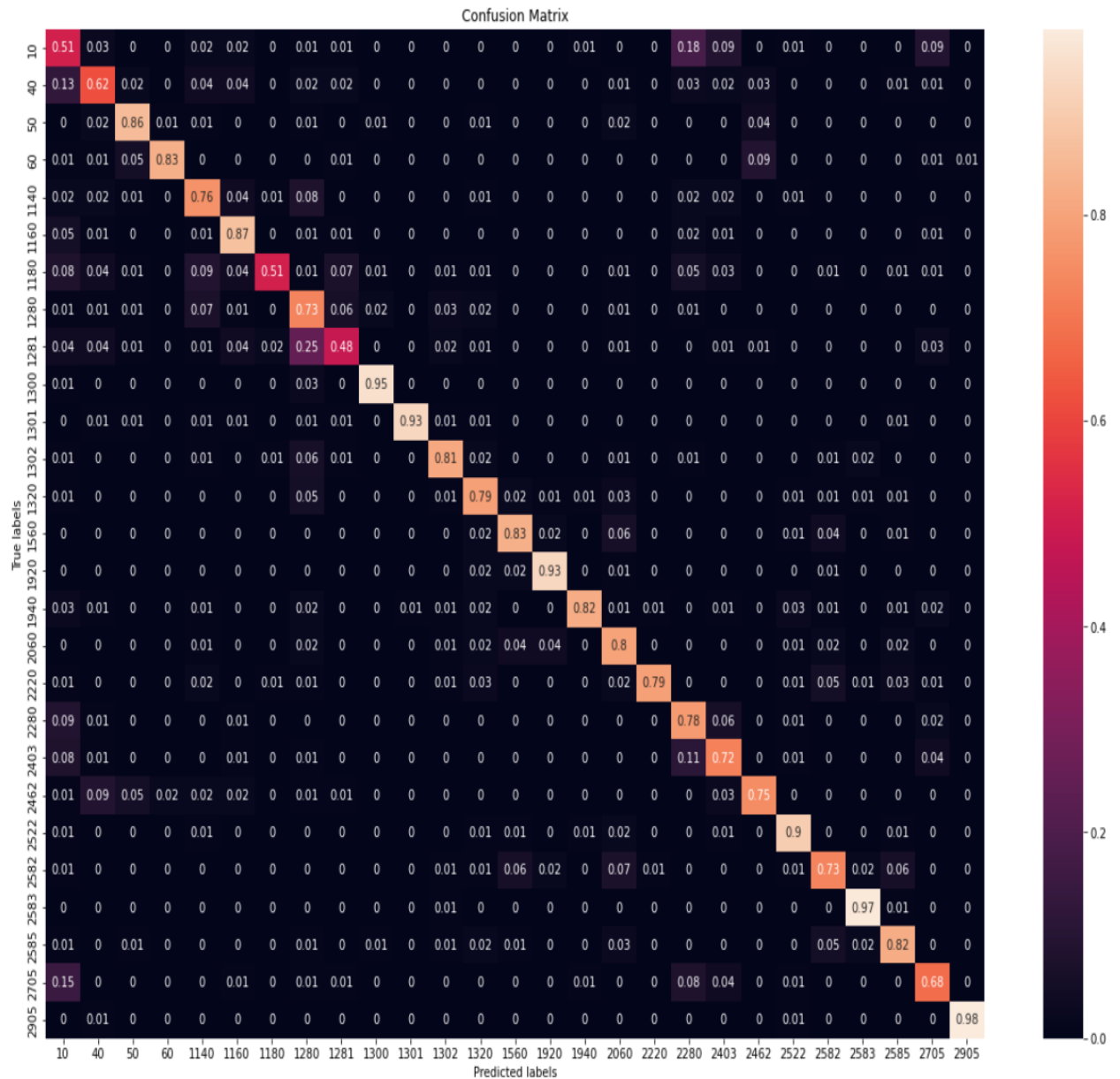
Classe 2462 (games)

- ➔ confondue avec la classe 60 (games and consoles) dans **5%** des cas

Classe 2585 (gardening and DIY)

- ➔ confondue avec la classe 2582 (furniture, kitchen and garden) dans **5%** des cas

7.2.2 Annexe 4.2.2.1.2 - Conv1D



Classe 10 (adult books)

- ➔ confondue avec la classe 2705 (books) dans **13%** des cas
- ➔ confondue avec la classe 40 (imported video games) dans **10%** des cas
- ➔ confondue avec la classe 2403 (children books and magazines) dans **12%** des cas
- ➔ confondue avec la classe 2280 (magazines) dans **14%** des cas

Classe 40 (imported video games)

- ➔ confondue avec la classe 2462 (games) dans **5%** des cas

Classe 1140 (figurine and toy pop)

- ➔ confondue avec la classe 1280 (toys for children) dans **11%** des cas

- ➔ confondue avec la classe 1180(figurine, masks and role-playing games) dans **2%** des cas

Classe 1280 (toys for children)

- ➔ confondue pratiquement avec toutes les classes avec des petits pourcentages :
- ➔ confondue avec la classe 1281 (board games) dans **10%** des cas
- ➔ confondue avec la classe 1140 (figurine and toy pop) dans **4%** des cas

Classe 1281 (board games)

- ➔ confondue avec la classe 1180 (figurine, masks and role-playing games) dans **2%** des cas
- ➔ confondue avec la classe 1280 (toys for children) dans **12%** des cas

Classe 1560 (interior furniture and bedding)

- ➔ confondue avec la classe 2582 (furniture, kitchen and garden) dans **2%** des cas

Classe 2060 (decoration interior)

- ➔ confondue avec la classe 2582 (furniture, kitchen and garden) dans **3%** des cas
- ➔ confondue avec la classe 1560 (interior furniture and bedding) dans **5%** des cas

Classe 2280 (magazines)

- ➔ confondue avec la classe 2403 (children books and magazines) dans **11%** des cas
- ➔ confondue avec la classe 10 (adult books) dans **11%** des cas

Classe 2403(children books and magazines)

- ➔ confondue avec la classe 2280 (magazines) dans **5%** des cas
- ➔ confondue avec la classe 10 (adult books) dans **5%** des cas

Classe 2462 (games)

- ➔ confondue avec la classe 60 (games and consoles) dans **5%** des cas

Classe 2585 (gardening and DIY)

- ➔ confondue avec la classe 2582(furniture, kitchen and garden) dans **5%** des cas

7.3 ANNEXE 4.2.2.5.5 PERFORMANCES DES MODÈLES CNN

Image - Xception					
Classe produit	Labels	precision	recall	f1-score	support
1281	board games	0,33	0,36	0,34	408
2220	supplies for domestic animals	0,69	0,27	0,39	190
1180	figurines, masks and role playing games	0,55	0,31	0,40	144
2582	furniture kitchen and garden	0,49	0,31	0,42	533
1280	toys for children	0,55	0,31	0,47	1002
50	video games accessories	0,46	0,31	0,47	319
2585	gardening and DIY	0,57	0,31	0,48	468
1302	toys, outdoor playing, clothes	0,41	0,31	0,49	514
1320	early childhood	0,53	0,31	0,52	664
2060	decoration interior	0,51	0,31	0,55	970
1560	interior furniture and bedding	0,53	0,31	0,58	993
1301	accessories children	0,67	0,31	0,61	162
10	adult books	0,58	0,31	0,62	643
40	imported video games	0,68	0,31	0,63	503
2462	games	0,72	0,31	0,65	301
1140	figurines and Toy Pop	0,71	0,31	0,68	545
2403	children books and magazines	0,77	0,31	0,69	961
1940	Food	0,77	0,31	0,70	159
2905	online distribution of video games	0,66	0,31	0,70	146
2522	stationery	0,74	0,31	0,72	1011
1300	remote controlled models	0,71	0,31	0,75	993
60	games and consoles	0,79	0,31	0,76	179
2280	magazines	0,78	0,31	0,77	911
2705	books	0,73	0,31	0,79	557
1920	interior accessories	0,81	0,31	0,83	832
2583	piscine spa	0,80	0,31	0,84	2083
1160	playing cards	0,92	0,31	0,92	793

accuracy
macro avg
weighted avg

		0,66	16984
0,65	0,61	0,62	16984
0,67	0,66	0,66	16984

Images - Inception					
Classe produit	Labels	precision	recall	f1-score	support
1281	board games	0,36	0,28	0,32	1281
1180	figurines, masks and role playing games	0,60	0,31	0,41	1180
2220	supplies for domestic animals	0,43	0,39	0,41	2220
50	video games accessories	0,52	0,34	0,41	50
1280	toys for children	0,52	0,41	0,46	1280
2582	furniture kitchen and garden	0,47	0,46	0,46	2582
2585	gardening and DIY	0,45	0,50	0,47	2585
1302	toys, outdoor playing, clothes	0,62	0,37	0,47	1302
1940	Food	0,38	0,73	0,50	1940
1320	early childhood	0,56	0,48	0,51	1320

2060	decoration interior	0,46	0,58	0,51	2060
1560	interior furniture and bedding	0,61	0,54	0,57	1560
10	adult books	0,59	0,61	0,60	10
2462	games	0,88	0,47	0,61	2462
1301	accessories children	0,80	0,51	0,62	1301
40	imported video games	0,69	0,60	0,65	40
1140	figurines and Toy Pop	0,58	0,74	0,65	1140
2905	online distribution of video games	0,63	0,69	0,66	2905
2403	children books and magazines	0,69	0,70	0,69	2403
2522	stationery	0,74	0,65	0,69	2522
1300	remote controlled models	0,65	0,81	0,72	1300
2280	magazines	0,74	0,76	0,75	2280
1920	interior accessories	0,66	0,88	0,76	1920
60	games and consoles	0,70	0,82	0,76	60
2705	books	0,73	0,83	0,78	2705
2583	piscine spa	0,83	0,86	0,84	2583
1160	playing cards	0,94	0,89	0,92	1160

accuracy
macro avg
weighted avg

	0,65	0,65	0,65
0,62	0,60	0,60	0,62
0,65	0,65	0,64	0,65

7.4 Description des fichiers de code

Les fichiers de code sous Github se trouvent sous : [ici](#)

Fichier	Description
Text_Cleaning.ipynb	Notebook de nettoyage des données Texte
Data_Viz_Text.ipynb	Notebook DataViz Texte et Images
Change_product_class_number_image_name.ipynb	Notebook pour le changement des numéros de classe et ajout des noms des images à partir des productid et imageid

Xception_All_Train_data_299x299_100Epochs_Accuracy_66.36%	Notebook de classification à partir des données Images avec Xception
InceptionV3_All_Train_data_299x299_100Epochs_Accuracy_65%	Notebook de classification à partir des données Images avec Inception
rakuten_text_Conv1D.ipynb	Notebook de classification à partir des données Texte avec Conv1D
rakuten_text_Simple_DNN.ipynb	Notebook de classification à partir des données Texte avec Simple DNN
Models_Voting_Simple_DNN_CON1D_Xception. ipynb	Notebook de classification multimodale à partir des images et textes (Xception , Simple DNN et Conv1D)
Models_Voting_Simple_DNN_CON1D_InceptionV3 ipynb	Notebook de classification multimodale à partir des images et textes (Inception , Simple DNN et Conv1D)
x_train_cleaned_V0.pkl	Jeu de données Train complet avec Texte nettoyé
train_cleaned&Labels_V1.pkl	Jeu de données Train complet avec Texte nettoyé + les catégories des produits
df_train_final.pkl	Jeu de données Train complet Final avec Texte nettoyé , les catégories des produits, les noms des fichiers images et modification des numéros des classes

8 BIBLIOGRAPHIE

- Supports de cours DataScientest
- Aurélien Géron - Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow (Concepts, Tools, and Techniques to Build Intelligent Systems)
- Francois Chollet - Deep Learning with Python
- <https://challengedata.ens.fr/participants/challenges/35/>
- <https://keras.io/api/applications/>
- https://keras.io/guides/transfer_learning/
- <https://towardsdatascience.com/how-to-optimize-learning-rate-with-tensorflow-its-easier-than-you-think-164f980a7c7b>
- <https://www.datasciencetoday.net/index.php/en-us/deep-learning/173-les-reseaux-de-neurones-convolutifs>
- <https://control.com/technical-articles/combining-two-deep-learning-models/#:~:text=The%20most%20common%20method%20to,techniques%20to%20build%20a%20model.>
- <https://medium.com/deeplab-ai/shopping-e-commerce-products-by-example-through-deep-learning-114e334d7aba>
- <https://towardsdatascience.com/retrieving-similar-e-commerce-images-using-deep-learning-6d43ed05f46b>
- <https://www.bigcommerce.com/blog/ecommerce-machine-learning/#use-cases-for-ecommerce-machine-learning>
- <https://www.telusinternational.com/articles/what-is-product-categorization>
- <https://stackoverflow.com/questions/60216418/keras-imagedatagenerator-flow-class-indices-for-sparse-and-categorical-classes>
- <https://medium.com/@vitalshchutski/french-nlp-entamez-le-camembert-avec-les-librairies-fast-bert-et-transformers-14e65f84c148>