

# Présentation Projet: Pykuten

## Classification multimodale

Participants:

Sana Lamiri

Ewen Bail

Olivier Douangvichith





# Plan de la Présentation

1. Contexte
2. Analyse exploratoire de données
3. Pré-traitement de données
4. Modèle d'identification d'images
5. Modèle d'identification de texte
6. Modèle final
7. Conclusion

# Contexte

## Cadre de projet:

- Concours organisé par Rakuten Institute of Technology
- Formation DataScientest
- Intérêt pédagogique du challenge
  - Quantité importante de données disponible
  - Confrontation à des cas réels d'exploitation
- Problématique introduite par le commerçant:
  - Catalogue en constante évolution
  - Disparité des produits présentés (images et descriptions)

## Objectif:

- Etablir un modèle de prédiction à partir des informations textuelles et de l'image associée à chaque article du catalogue.

Données  
images



Données  
textes

Designation	Description
Matelas de piscine California Dream de la marque Intex ? Matière : Vinyle ?	
Matelas de piscine	Dimensions (L x l) : 178 x 84 cm ?
Hawai Fleurs -	Nombre de places : 1 personne ? 3
Intex	couloirs aux choix

**Classe Produit:**  
**Accessoires piscine et spa**

**Dataset:** 84 916

**Fichiers d'entrée**

Fichiers de sortie

- ### Fichier de sortie:

- Distribution des classes => Déséquilibrée

Catégorie	Nombre d'articles
Accessoires pique et spa	10000
médiateur d'intérieur	5000
Méditerranée	5000
Éclairage et décoration	5000
Peinture	5000
Jouets	4800
Magazines et revues	4800
Archives journaux et presse	4800
Logis de maison	4500
Cartes à échanger	4000
Lignes 1	3200
Lignes 2	2800
Régimes	2800
Décoration extérieure et accessoires	2800
Utensiles de nettoyage, aspirateur	2800
Éclairage extérieur	2800
Lignes d'intérieur	2800
Lignes extérieur	2000
Accessoires de jeu vidéo	1800
Console de jeu et jeu vidéo	1500
Jeux en téléchargement	1000
Console de jeu vintage	1000
Armature	1000
Peintures, linéaire	1000
Association	1000
Jeux de rôle et jeux de figurines	1000

# Pré-traitement de données

Texte

Suppression des NaNs

Désignation + Description  
=> Text

HTML encoding => UTF8

Suppression balises  
HTML et caractères  
spéciaux

Suppression des  
StopWords

Dataset dont  
on connaît les  
classes

Entrainement  
72%

Validation  
18%

Test  
10%

Image

Chargement à partir  
de chemin d'accès

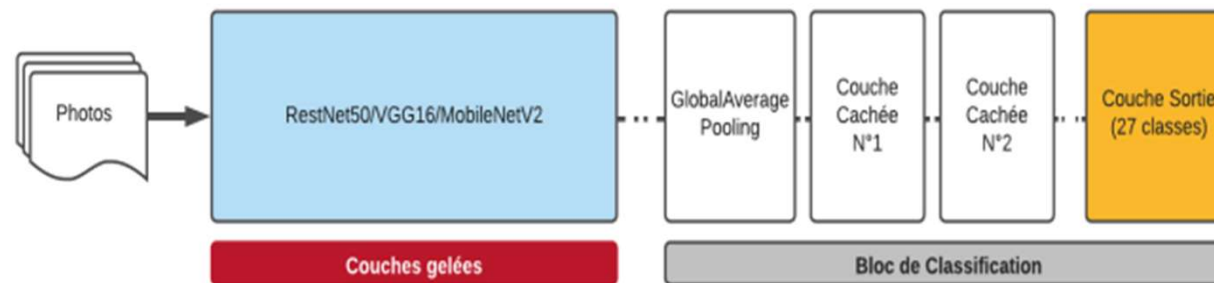
Générateur d'image

Redimensionnement,  
rotation, translation

Caractéristiques  
généralisables

# Modèle d'identification d'image

Vue d'ensemble du modèle



## Hyperparamètres:

- Loss: sparse\_categorical\_crossentropy
- Optimizer: Adam
- Learning Rate: 0,01 / 0,001
- Batch: 64
- Epoch: 10

## Entraînement:

3 callbacks:

- taux d'apprentissage
- stagnation de la fonction de loss
- checkpoints

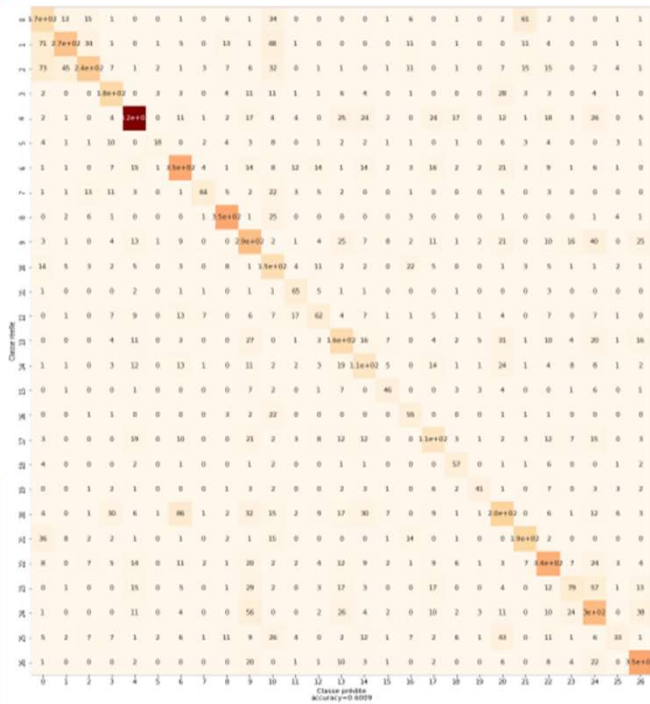
# Modèle d'identification d'image

## Résultats et évaluation des performances

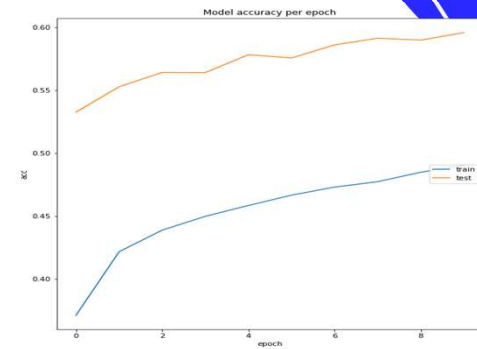
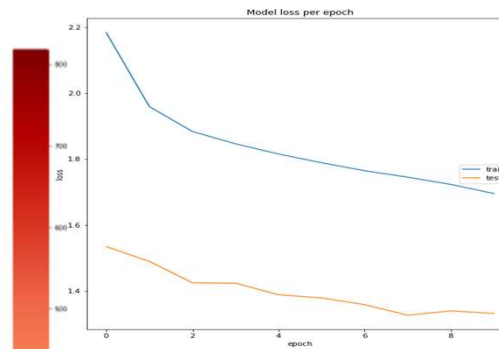
Modèle (base)	Learning rate	f1_score (Test)	Accuracy (Train)	Accuracy (Val)	Accuracy (Test)
ResNet50	0.01	0.59	0.45	0.55	0.59
<b>ResNet50</b>	<b>0.001</b>	<b>0.59</b>	<b>0.49</b>	<b>0.59</b>	<b>0.60</b>
MobilNetV2	0.01	0.38	0.44	0.52	0.42
MobilNetV2	0.001	0.42	0.45	0.50	0.44
VGG16	0.01	0.54	0.57	0.55	0.56

# Modèle d'identification d'image

CNN, base: ResNet50, LR: 0,001



Matrice de confusion



Historique d'apprentissage

	precision	recall	f1-score	support
0	0.41	0.53	0.47	312
1	0.76	0.57	0.65	476
2	0.73	0.51	0.60	478
3	0.62	0.68	0.65	267
4	0.85	0.80	0.83	1021
5	0.62	0.24	0.34	76
6	0.65	0.69	0.67	505
7	0.73	0.45	0.56	142
8	0.83	0.88	0.86	395
9	0.49	0.59	0.54	499
10	0.34	0.60	0.43	251
11	0.53	0.78	0.63	83
12	0.45	0.37	0.41	168
13	0.45	0.49	0.47	324
14	0.42	0.45	0.43	249
15	0.52	0.55	0.53	83
16	0.40	0.63	0.49	87
17	0.46	0.46	0.46	250
18	0.52	0.71	0.60	80
19	0.66	0.51	0.57	81
20	0.46	0.42	0.44	487
21	0.62	0.69	0.66	276
22	0.66	0.67	0.67	499
23	0.50	0.31	0.38	259
24	0.54	0.60	0.57	507
25	0.49	0.16	0.24	207
26	0.74	0.81	0.78	430
accuracy			0.60	8492
macro avg	0.57	0.56	0.55	8492
weighted avg	0.61	0.60	0.60	8492

Classification report



# Modèle d'identification de texte

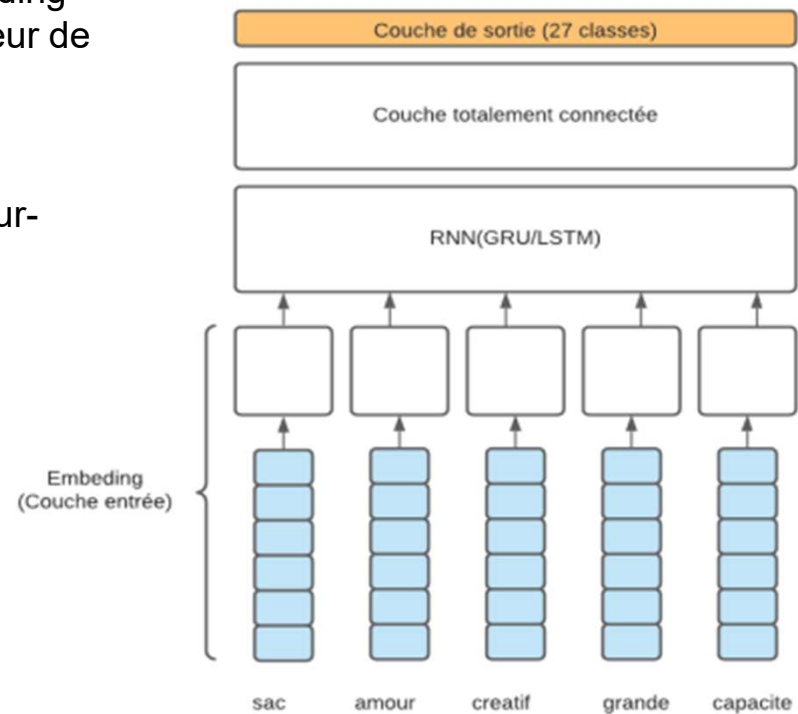
## Hyperparamètres:

- Couche Embedding: mot => vecteur d'embedding
- Couche RNN à base de GRU ou LSTM: vecteur de mots => suite de séquences
- Couche cachée : calcul de classification
- Couche de sortie: classification finale
- Couches de Dropout et de pooling: éviter le sur-apprentissage

## Entraînement:

- Taille du dictionnaire des mot: 10000
- Colonnes de la matrice d'embedding: 300
- Cellules GRU et LSTM : 256
- Optimizer Adam avec 2 LR: 0.01 et 0.001
- Epoch : 10

## Vue d'ensemble du modèle



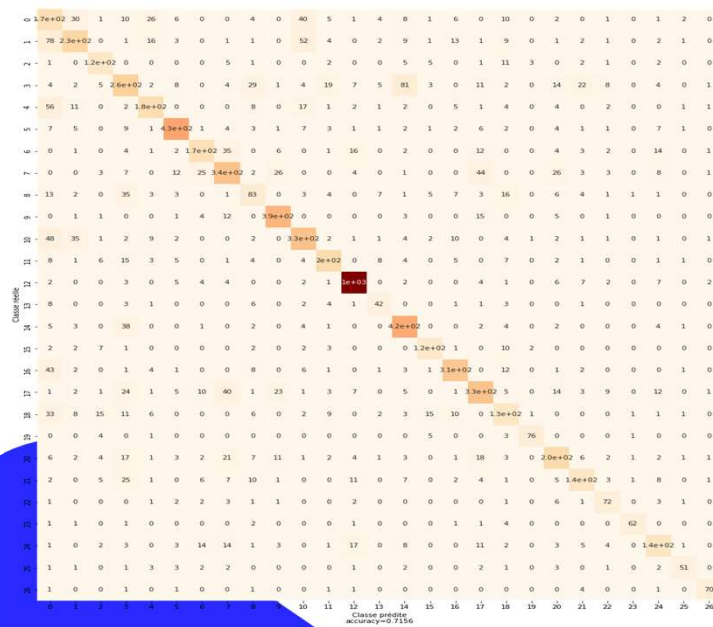
# Modèle d'identification de texte

Résultats et évaluation des performances

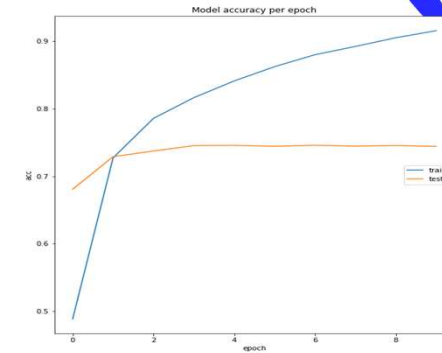
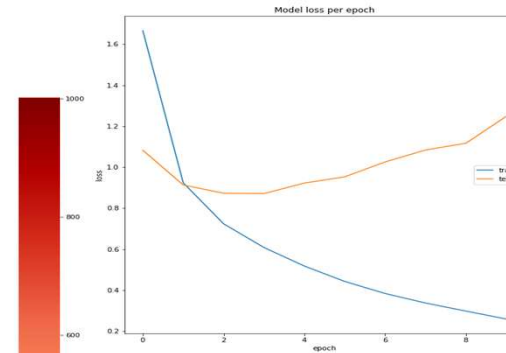
Modèle	f1_score (Test)	Accuracy (Train)	Accuracy (Val)	Accuracy (Test)	Learning rate
GRU	0.34	0.45	0.48	0.34	0.01
GRU	0.40	0.55	0.50	0.39	0.001
LSTM	0.69	0.90	0.70	0.69	0.01
<b>LSTM</b>	<b>0.72</b>	<b>0.95</b>	<b>0.74</b>	<b>0.71</b>	<b>0.001</b>

# Modèle d'identification de texte

base: RNN - LSTM, LR: 0,001



Matrice de confusion



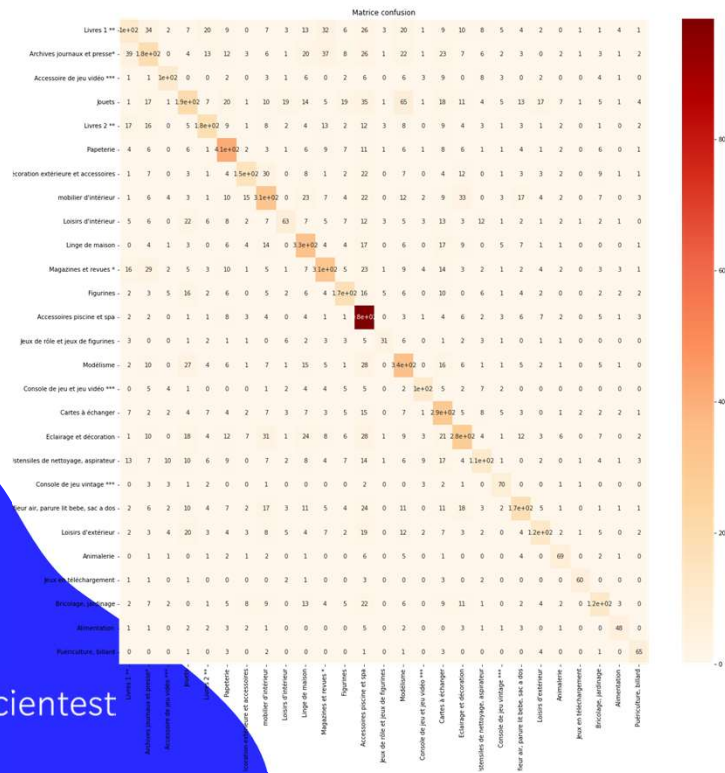
Historique d'apprentissage

	precision	recall	f1-score	support
0	0.35	0.52	0.42	332
1	0.68	0.54	0.60	428
2	0.69	0.76	0.72	161
3	0.55	0.52	0.54	496
4	0.70	0.61	0.65	302
5	0.87	0.86	0.87	503
6	0.71	0.62	0.66	274
7	0.69	0.67	0.68	500
8	0.45	0.42	0.43	199
9	0.84	0.90	0.87	431
10	0.69	0.72	0.71	463
11	0.75	0.72	0.73	273
12	0.93	0.95	0.94	1054
13	0.56	0.58	0.57	73
14	0.73	0.86	0.79	482
15	0.75	0.79	0.77	149
16	0.83	0.78	0.80	399
17	0.71	0.66	0.68	500
18	0.54	0.52	0.53	257
19	0.92	0.84	0.88	90
20	0.65	0.63	0.64	325
21	0.68	0.59	0.63	242
22	0.63	0.74	0.68	97
23	0.93	0.84	0.88	74
24	0.63	0.60	0.61	231
25	0.81	0.67	0.73	76
26	0.88	0.86	0.87	81
accuracy			0.72	8492
macro avg	0.71	0.70	0.70	8492
weighted avg	0.72	0.72	0.72	8492

Classification report

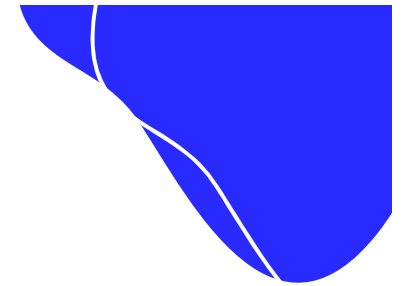
# Concaténation des deux modèles

- Confrontation simple des prédictions des deux modèles pour chaque classe:



$$P_{model\_final}(classe) = \frac{P_{model\_img}(classe) + P_{model\_txt}(classe)}{2}$$

	precision	recall	f1-score	support
0	0.45	0.31	0.37	332
1	0.50	0.44	0.47	428
2	0.49	0.66	0.56	161
3	0.50	0.38	0.43	496
4	0.32	0.60	0.42	302
5	0.89	0.81	0.85	503
6	0.46	0.57	0.51	274
7	0.73	0.61	0.66	500
8	0.20	0.34	0.25	199
9	0.74	0.75	0.74	431
10	0.74	0.65	0.70	463
11	0.65	0.58	0.62	273
12	0.94	0.93	0.93	1054
13	0.38	0.42	0.40	73
14	0.73	0.70	0.71	482
15	0.79	0.67	0.72	149
16	0.89	0.72	0.79	399
17	0.70	0.56	0.62	500
18	0.57	0.42	0.48	257
19	0.72	0.78	0.75	90
20	0.59	0.54	0.56	325
21	0.54	0.48	0.51	242
22	0.34	0.72	0.46	97
23	0.67	0.81	0.74	74
24	0.43	0.51	0.47	231
25	0.83	0.63	0.72	76
26	0.27	0.81	0.41	81
accuracy			0.63	8492
macro avg	0.60	0.61	0.59	8492
weighted avg	0.66	0.63	0.64	8492



# DEMO Streamlit





# Perspectives et conclusion



## Problématiques rencontrés :

- Volume de données et capacité de calculs
- Acquisition et maîtrise des concepts fondamentaux du deep learning.

## Perspectives d'amélioration :

- Etablir un modèle final plus robuste (en fonction des classes)
- Améliorer l'entraînement

## Conclusion :

- On peut considérer l'objectif atteint, autant d'un point de vue des performances, que pédagogique.



# Questions?

