Classifiez automatiquement des biens de consommation

Etudier la faisabilité d'une classification de produits à partir de leur description et d'images - Procéder à une classification supervisée à partir des images

Date de la soutenance : 06/04/2024

Antoine Arragon

Objectifs

Etudier la faisabilité d'un moteur de classification d'articles en différentes catégories, à partir de descriptions et d'images postées par des vendeurs de la plateforme de e-commerce 'Place de Marché'. Il s'agit de **fluidifier l'expérience des utilisateurs**.

- ➢ Preprocessing et extraction de features : texte/image ;
- ➢ Réduction de dimension et étude de faisabilité de la classification ;
- ➢ Effectuer une classification supervisée à partir des images ;
- ➢ Objectifs :
  - Se prononcer sur la classification : visualisations et indicateur de similarité
  - Entraîner des modèles de classification supervisée à partir des images
  - Récolter et filtrer des informations via une API

Présentation rapide des données

- Dataset contenant des informations sur 1050 produits : nom du produit, identifiant, prix, description, catégorie...
- 7 catégories principales : Home Furnishing - Baby Care - Watches - Home Decor & Festive Needs - Kitchen & Dining - Beauty and Personal Care - Computers ;
- 150 produits par catégorie ;
- On a également des images correspondant aux produits :

Kitchen & Dining

Watches

Les textes et images fournis ne relèvent pas de la propriété intellectuelle.

Plan :

1. Preprocessing du texte et étude de faisabilité de la classification
    - 1.1. Prétraitements appliqués aux descriptions
    - 1.2. Faisabilité classification : Approche “Bag of Words”
    - 1.3. Faisabilité classification : Approche “Word/Sentence Embedding”

2. Preprocessing des images, étude de faisabilité de la classification
    - 2.1. Approche SIFT - extraction de features, analyse des résultats
    - 2.2. Approche CNN - extraction de features, analyse des résultats

3. Classification supervisée des images
    - 3.1. Méthodologie et entraînements des modèles
    - 3.2. Inclusion de data augmentation
    - 3.3. Synthèse comparative

4. Requêtes API

Conclusion

1. Preprocessing du texte et étude de faisabilité de la classification

1.1. Prétraitements appliqués au descriptions

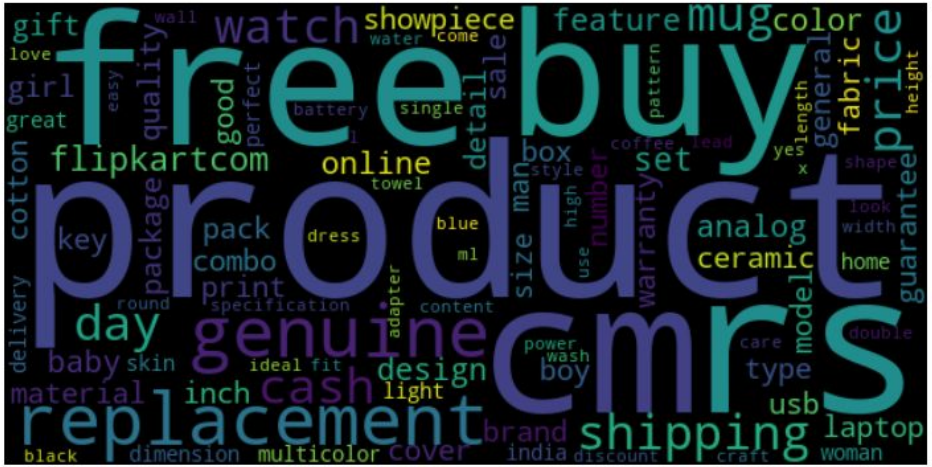
- Utilisation des librairies spaCy et nltk
- Passage en minuscules ➢ exclusion des ‘stopwords’ ➢ exclusion de mots fréquents spécifiques au domaine ➢ exclusion de mots composés d’une ou deux lettres ➢ lemmatisation ou stemming.
- Exemple :

```
Texte original : Buy Nkp Cotton Bath Towel at Rs. 549 at Flipkart.com. Only Genuine Products. Free Shipping. Cash On Delivery!
Nombre de caractères - texte original : 109

Texte preprocessing spaCy (lemmatisation) : buy nkp cotton bath towel rs flipkartcom genuine product free shipping cash delivery
Nombre de caractères - texte preprocessing spaCy : 84

Texte preprocessing nltk (stemming) : buy nkp cotton bath towel rs flipkart com genuin product free ship cash deliveri
Nombre de caractères - texte preprocessing spaCy : 80
```

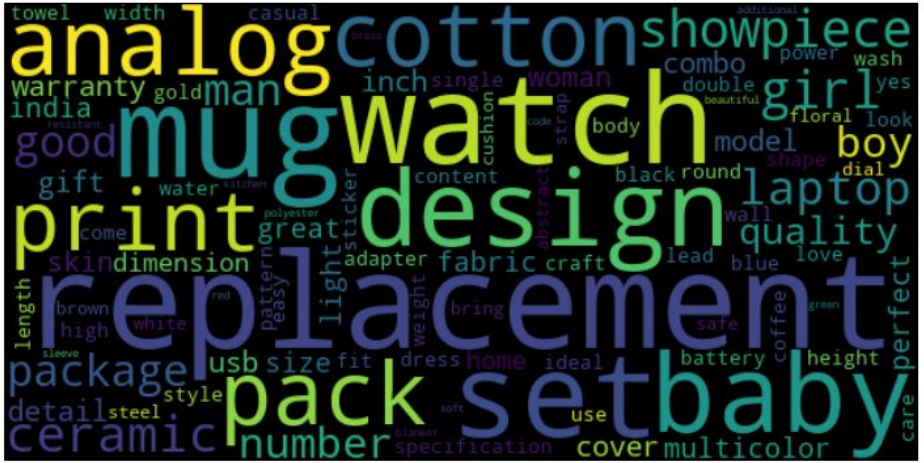
1. Preprocessing du texte et étude de faisabilité de la classification

1.1. Prétraitements appliqués au descriptions

Wordcloud “brut” :



Wordcloud post traitement :



1. Preprocessing du texte et étude de faisabilité de la classification

1.2. Faisabilité classification : Approche "Bag of Words"

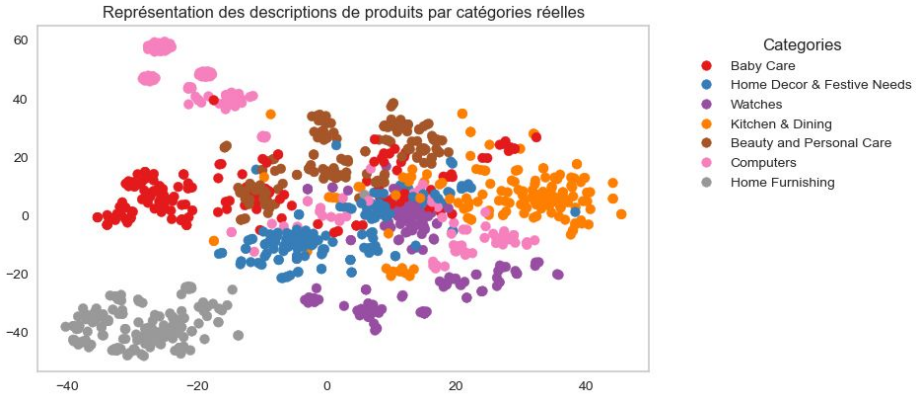
Idée : Transformer le texte prétraité en vecteurs de valeurs numériques assimilables par des modèles de machine learning

Approche **Bag of Words** : méthode simple et efficace qui se base sur le vocabulaire (mots uniques) du corpus et qui crée des vecteurs : soit par comptage "simple" de mots par document (CountVectorizer), soit par calcul d'une fréquence d'apparition de mots par document (TF-IDF ➢ donne un poids plus important aux mots assez peu présents dans l'ensemble du corpus).

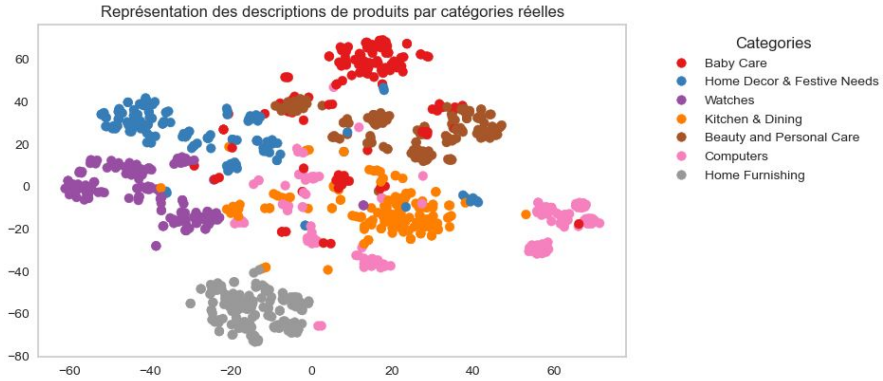
- ➢ Inconvénient : ordre des mots non pris en compte donc perte de sens, de contexte.
- ➢ Méthodologie :
  - Utilisation de CountVectorizer et TF-IDF sur les descriptions prétraitées : création de features
  - Réduction de dimension des features générées par t-SNE
  - Clustering (7 clusters)
  - Projection et visualisation en 2D / analyse graphique
  - Calcul de l'ARI - indice de similarité ➢ comparaison labels prédits vs catégories réelles

1. Preprocessing du texte et étude de faisabilité de la classification

1.2. Faisabilité classification : Approche “Bag of Words”



CountVectorizer :
ARI = 0.4824
t-SNE perplexity = 30



TF-IDF :
ARI = 0.6434
t-SNE perplexity = 20

1. Preprocessing du texte et étude de faisabilité de la classification

1.3. Faisabilité classification : Approche "Word/Sentence embedding"

Approche **Word/Sentence Embedding** : méthode consistant à utiliser des réseaux de neurones, plus ou moins profonds, qui vont capturer le sens de mots, de séquences ou de phrases complètes et transformer ces éléments en vecteurs. Ces algorithmes sont capables de saisir les relations sémantiques entre des mots ou groupes de mots. Des mots ou groupes de mots similaires auront des représentations vectorielles proches.

➢ Méthodologie :

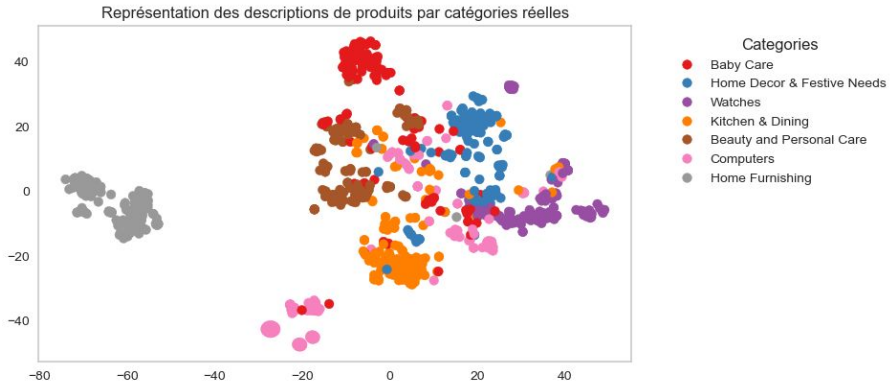
- Utilisation des modèles Word2Vec, BERT(HuggingFace / Tensorflow) et USE pour créer des features
- Réduction de dimension des features générées par PCA (n_components = 0.9) puis t-SNE
- Clustering (7 clusters)
- Projection et visualisation en 2D / analyse graphique
- Calcul de l'ARI - indice de similarité.

1. Preprocessing du texte et étude de faisabilité de la classification

1.3. Faisabilité classification : Approche “Word/Sentence embedding”

Principes de **Word2vec** :

- Utilisation d’un modèle pré-entraîné sur des articles google news : chaque mot du corpus, s’il est présent dans le corpus d'entraînement du modèle, aura un vecteur prédéfini associé ;
- Ce vecteur capture les relations sémantiques de ce mot, notamment grâce à un paramètre fixant la taille d’une fenêtre contextuelle permettant la prise en compte de mots voisins ;
- On se sert ici du modèle pour créer de nouvelles features, ces vecteurs, qui sont ensuite utilisées comme input d’un modèle de clustering ;
- On analyse les similarités entre les labels et les catégories réelles.



ARI = 0.5248
t-SNE perplexity = 30

1. Preprocessing du texte et étude de faisabilité de la classification

1.3. Faisabilité classification : Approche "Word/Sentence embedding"

Modèle **BERT** (Bidirectional Encoder Representations from Transformers) :

Spécificités :

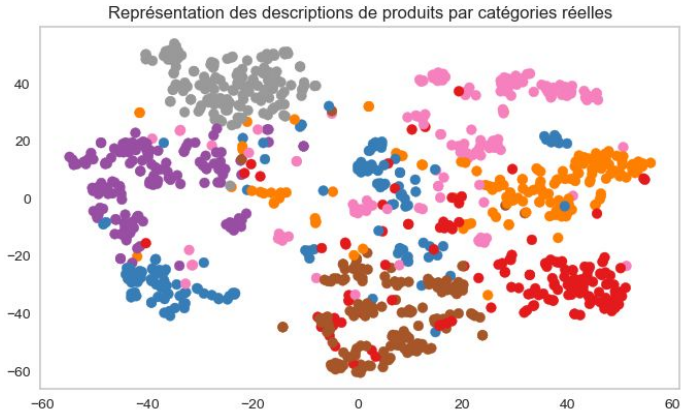
- ➢ Architecture Transformer qui permet de capter de manière plus performante que les modèles antérieurs les relations sémantiques entre les mots et donc le sens et le contexte de certaines phrases ;
- ➢ Bidirectionnel, à la différence de certains modèles précédent : prise en compte du contexte avant et après un mot ;
- ➢ Entraîné sur un très large corpus ;
- ➢ Entraîné à prédire des mots masqués dans une phrase et à repérer s'il y a des liens ou non entre des séquences.

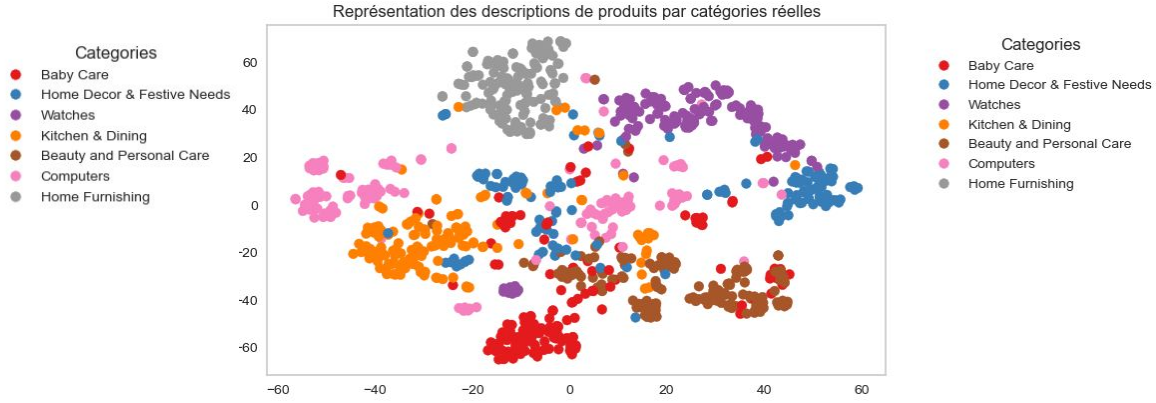
Utilisation de modèles pré-entraînés, via les bibliothèques Huggingface et Tensorflow, dans le cadre d'extraction de features.

Utilisation de ces features dans un modèle de clustering, visualisation et ARI comme précédemment.

1. Preprocessing du texte et étude de faisabilité de la classification

1.3. Faisabilité classification : Approche “Word/Sentence embedding”





BERT HuggingFace :
ARI = 0.5233
t-SNE perplexity = 20

BERT TensorFlow :
ARI = 0.4842
t-SNE perplexity = 20

1. Preprocessing du texte et étude de faisabilité de la classification

1.3. Faisabilité classification : Approche "Word/Sentence embedding"

Modèle **USE** (Universal Sentence Encoder) :

Spécificités :

- ➢ Architecture Transformer qui permet de capter de manière plus performante que les modèles antérieurs les relations sémantiques entre les mots et donc le sens et le contexte de certaines phrases ;
- ➢ Encodage de phrases complètes ;
- ➢ Entraîné sur un large corpus multilingue ;

Utilisation du modèle pré-entraîné, via la bibliothèque Tensorflow, dans le cadre d'extraction de features.

Utilisation de ces features dans un modèle de clustering, visualisation et ARI comme précédemment.

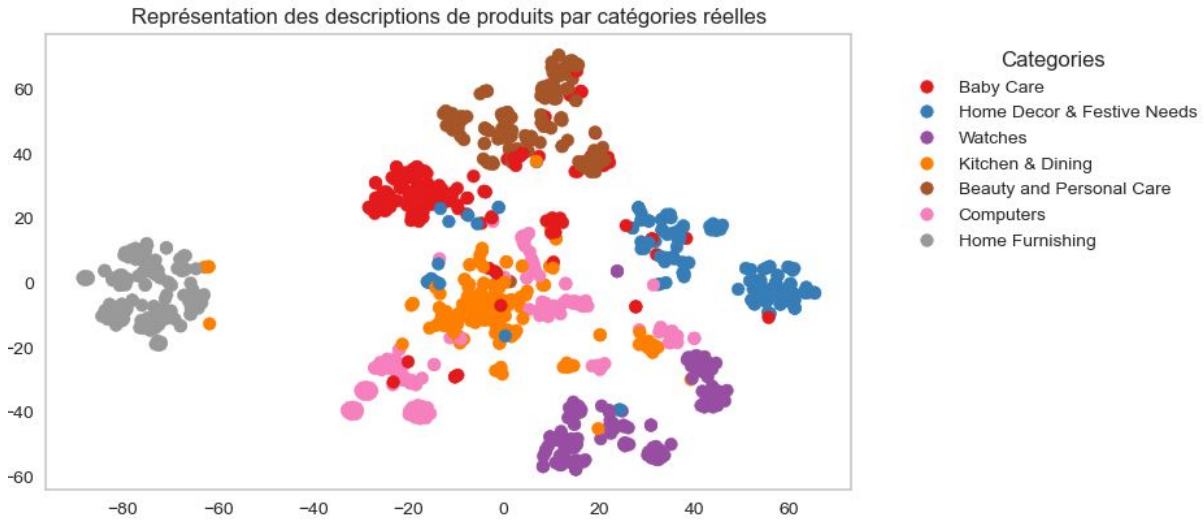
1. Preprocessing du texte et étude de faisabilité de la classification

1.3. Faisabilité classification : Approche “Word/Sentence embedding”

➢ Modèle avec lequel nous obtenons le meilleur ARI : 0.6738 ;

➢ Catégories de produits plutôt bien séparées sur le graphique ;

➢ **Classification à partir des descriptions semble possible.**



2. Preprocessing des images, étude de faisabilité de la classification

2.1. Approche SIFT - extraction de features - analyse des résultats

Exemple de différents traitements sur une image :

Image originale redimensionnée

Image en niveaux de gris

Equalized Image

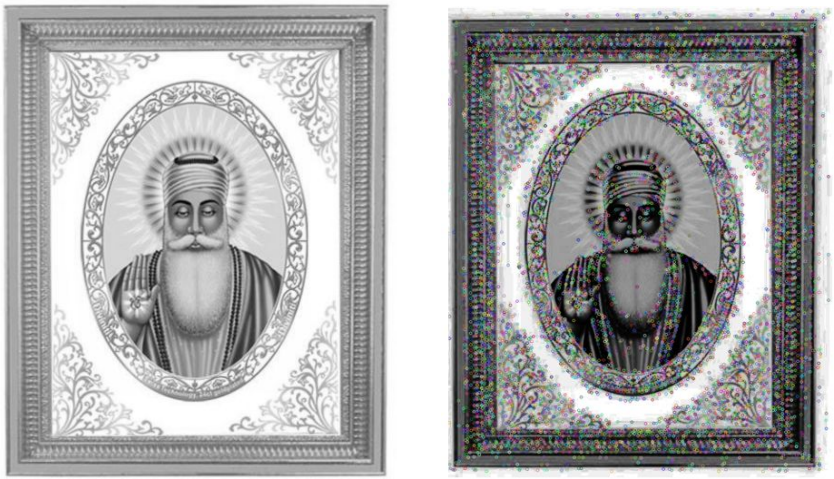
Image après filtrage gaussien

2. Preprocessing des images, étude de faisabilité de la classification

2.1. Approche SIFT - extraction de features - analyse des résultats

Principes de l'approche SIFT - Scale Invariant Feature Transform :

- Détecter des points d'intérêts dans une image : coins, bords, point d'intensité lumineuse
- Calculer des descripteurs : vecteurs qui décrivent les points d'intérêts
- Création de "visual words" par application d'un clustering aux descripteurs de features
- Création d'un histogramme par image où sont représentés les fréquences d'apparition de chaque "visual word"
- Création de bag of features d'une image : vecteur où sont représentées les fréquences de l'histogramme



Points d'intérêts sur une image

2. Preprocessing des images, étude de faisabilité de la classification

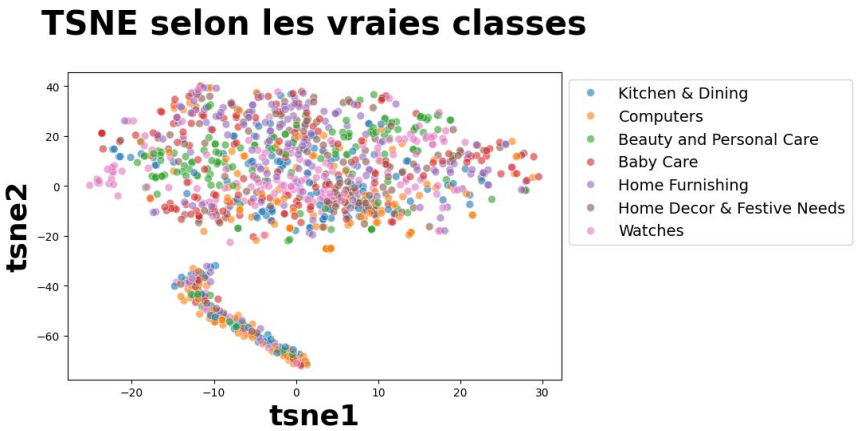
2.1. Approche SIFT - extraction de features - analyse des résultats

Réduction de dimension ➢ réduction du nombre de features par image :

- PCA (n_components = 0.99) : 727 ➢ 331
- t-SNE pour faciliter projection / visualisation en 2D

Clustering et calcul de l'ARI :

- ARI de 0.05
- Confirme analyse graphique : résultats non concluants via cette approche

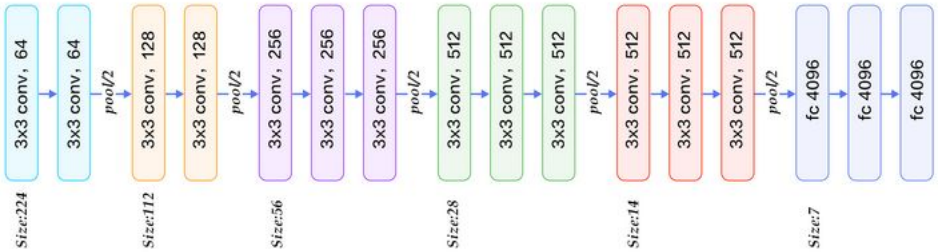


2. Preprocessing des images, étude de faisabilité de la classification

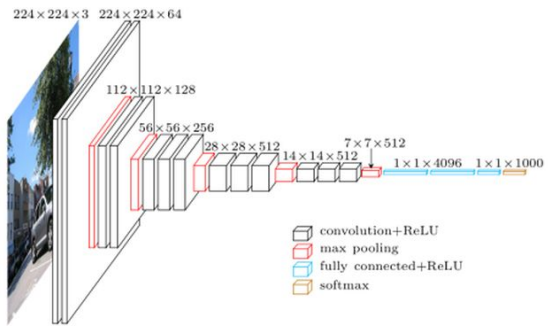
2.2. Approche CNN - extraction de features, analyse des résultats

Utilisation d'un modèle VGG16 :

- Réseau de neurones convolutifs à 16 couches
- Transfer learning : gel des poids du modèle pré-entraîné
- Suppression des dernières couches : but = extraction de features



Architecture de VGG-16



Représentation 3D de l'architecture de VGG-16

2. Preprocessing des images, étude de faisabilité de la classification

2.2. Approche CNN - extraction de features, analyse des résultats

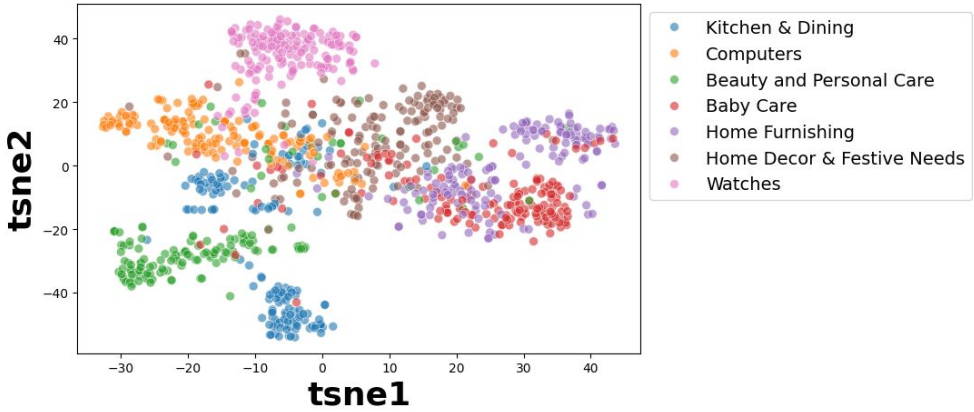
Réduction de dimension des features :

- PCA (n_components = 0.99) : 4096 ➢ 803
- t-SNE pour faciliter projection / visualisation en 2D

Clustering et calcul de l'ARI :

- ARI de 0.51
- Bien meilleure segmentation ici
- Séparation assez nette de certaines catégories sur le graphique

TSNE selon les vraies classes

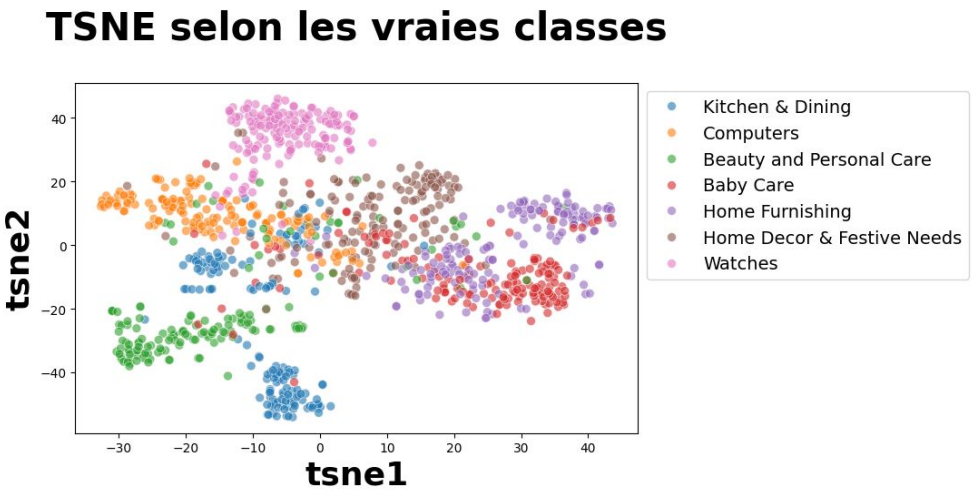


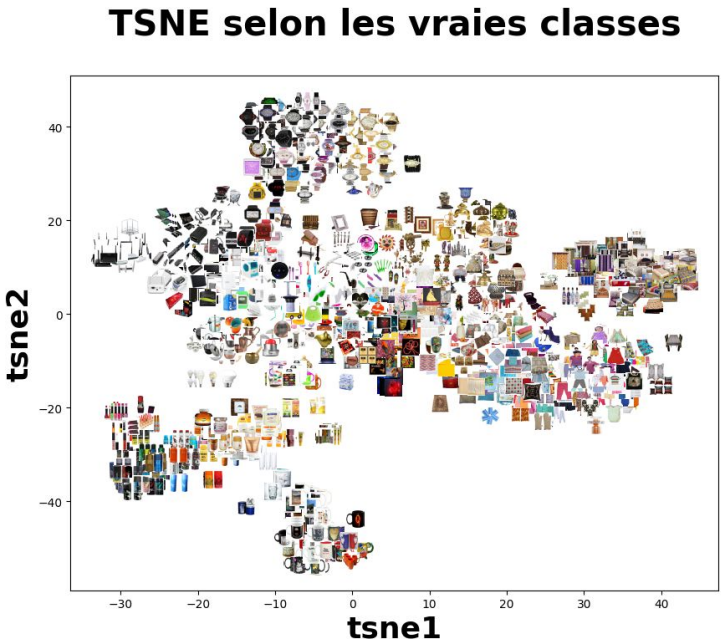
L'utilisation du modèle VGG16 nous permet de conclure à la faisabilité d'une classification supervisée à partir des images.

2. Preprocessing des images, étude de faisabilité de la classification

2.2. Approche CNN - extraction de features, analyse des résultats

Illustration segmentation avec images :





3. Classification supervisée des images

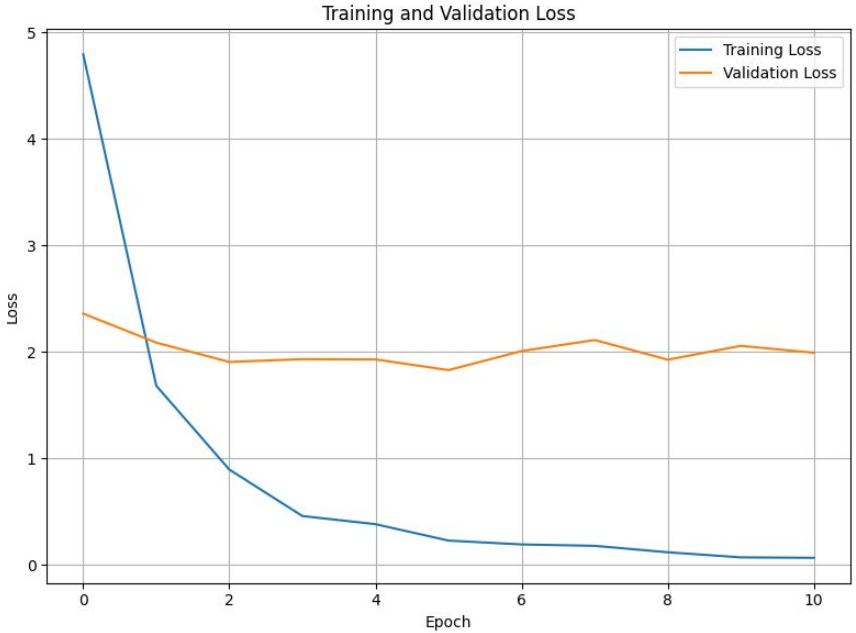
3.1. Méthodologie et entraînements des modèles

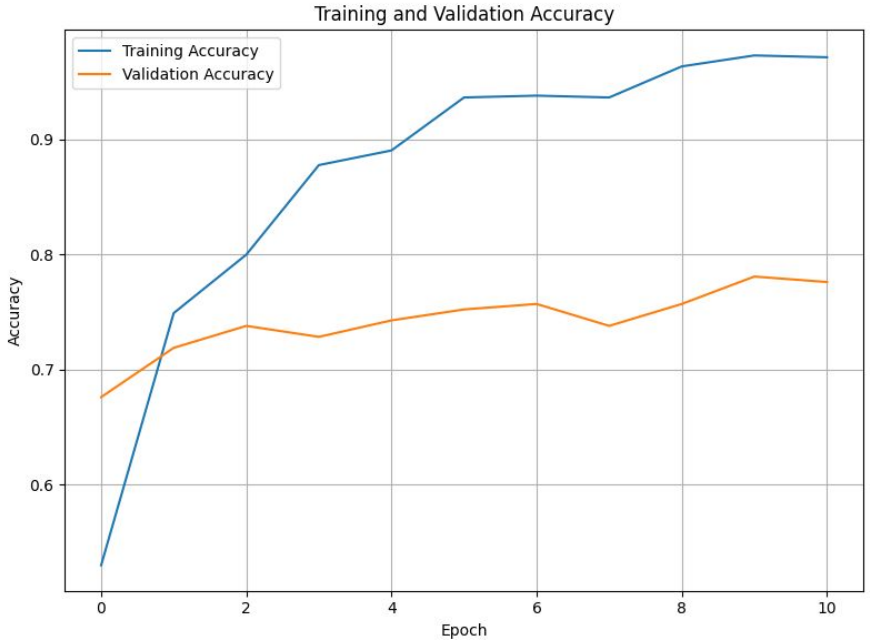
- Séparation du jeu de données en **jeu d'entraînement et de test** ;
- Création de sous-dossiers d'images par catégorie de produit ;
- Utilisation d'une fonction de la librairie keras (`image_dataset_from_directory`) afin de **créer des datasets d'entraînement, de validation et de tests, directement à partir de ces dossiers d'images** ;
- Utilisation de 4 modèles, réseaux de neurones convolutifs, d'abord sans data augmentation ;
- Adaptation de la couche de sortie de ces modèles à notre problème de classification : nombre de classes = 7 ;
- **Transfert learning** : gel des poids des modèles pré-entraînés (gain de temps et de moyens techniques) ;
- Analyse des résultats selon **différentes métriques**, en particulier **accuracy** et **temps d'entraînement** ;
- Modification de certains **hyperparamètres** : batch_size, Dropout, fonction d'activation, optimizer ;
- Ajout de data augmentation ➢ quel impact sur les résultats ?
- Synthèse comparative.

3. Classification supervisée des images

3.1. Méthodologie et entraînements des modèles

Exemple, modèle VGG19 :

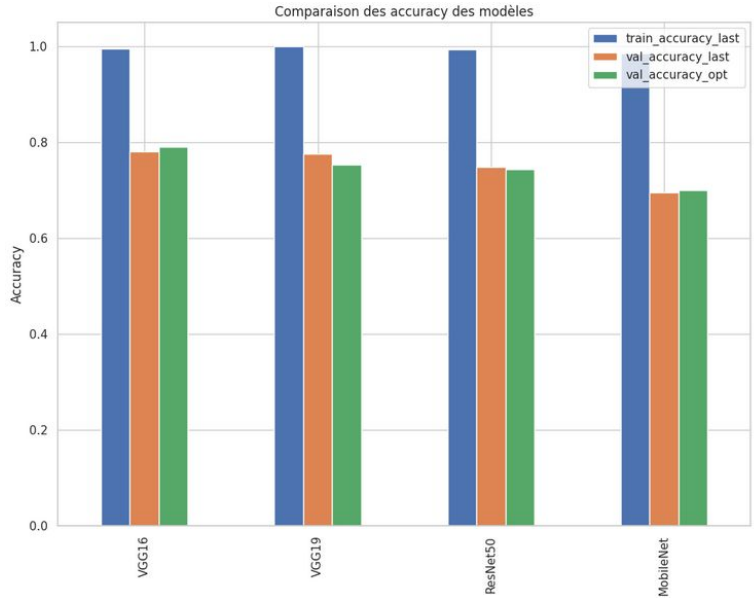


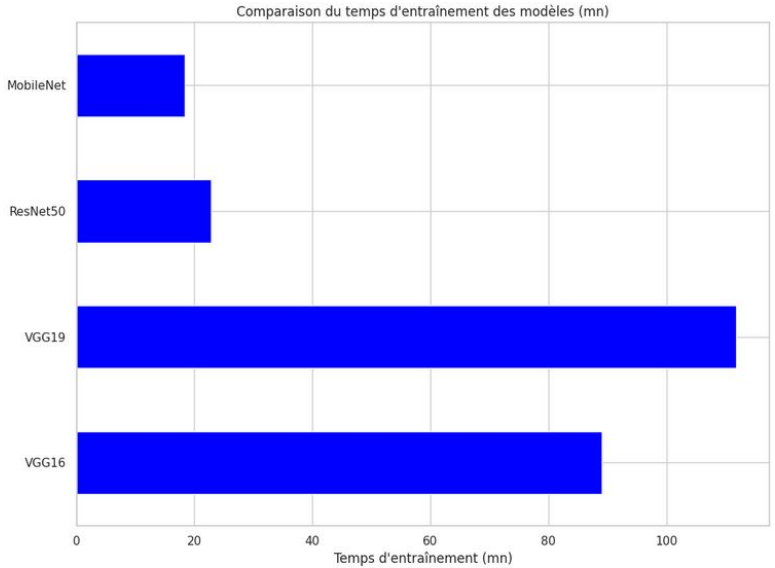


3. Classification supervisée des images

3.1. Méthodologie et entraînements des modèles

Résultats :





Constat : overfitting et nette différence de temps d'entraînement entre les modèles.

3. Classification supervisée des images

3.2. Inclusion de data augmentation

Principe :

- Augmenter la taille du jeu de données d'entraînement en appliquant des transformations diverses aux données existantes (ex : rotation, translation, changement d'échelle....)
- Peut améliorer la généralisation d'un modèle et atténuer l'overfitting.

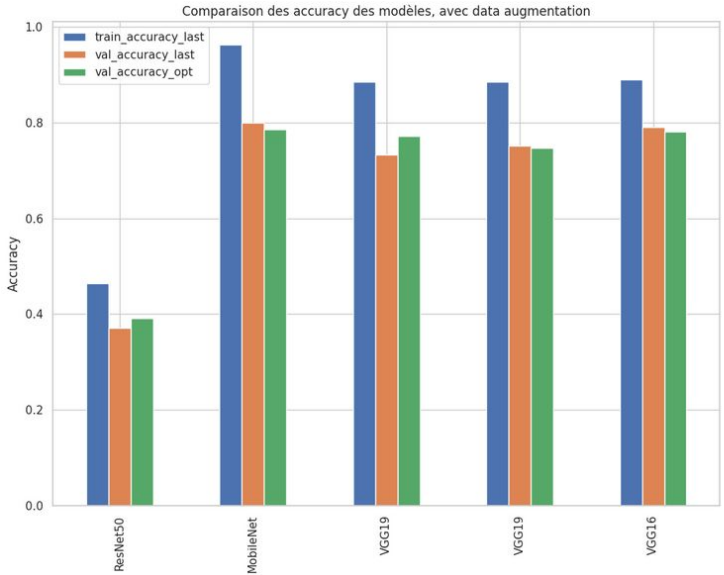
Image originale

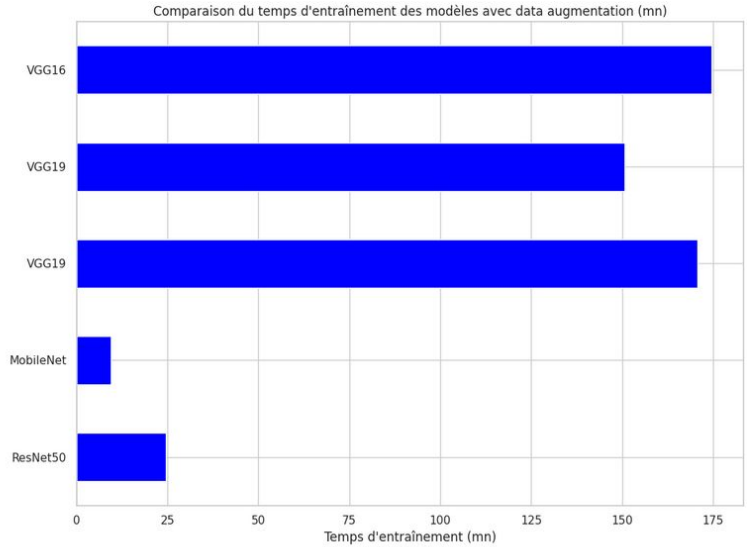
Image augmentée

3. Classification supervisée des images

3.2. Inclusion de data augmentation

Résultats :

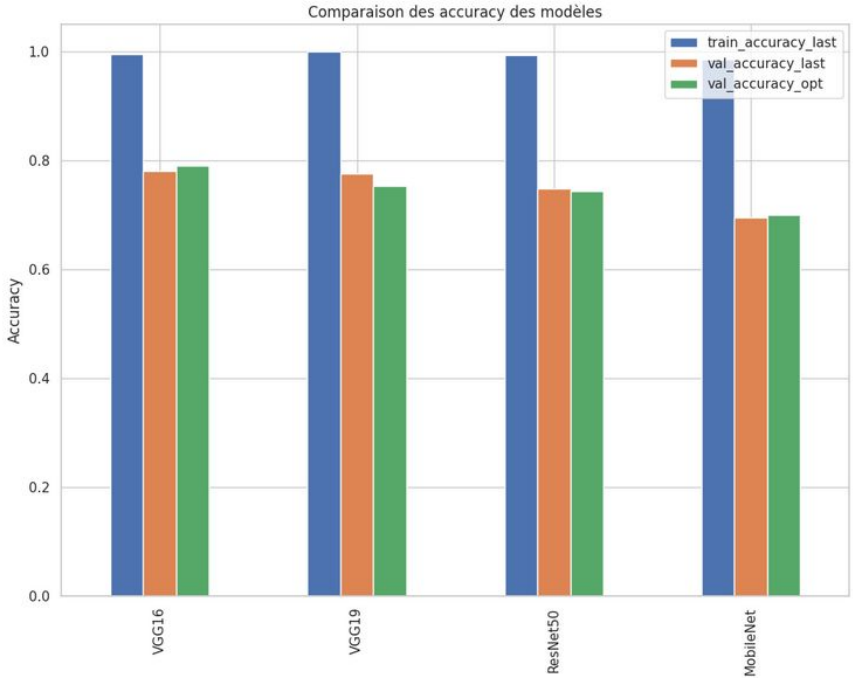


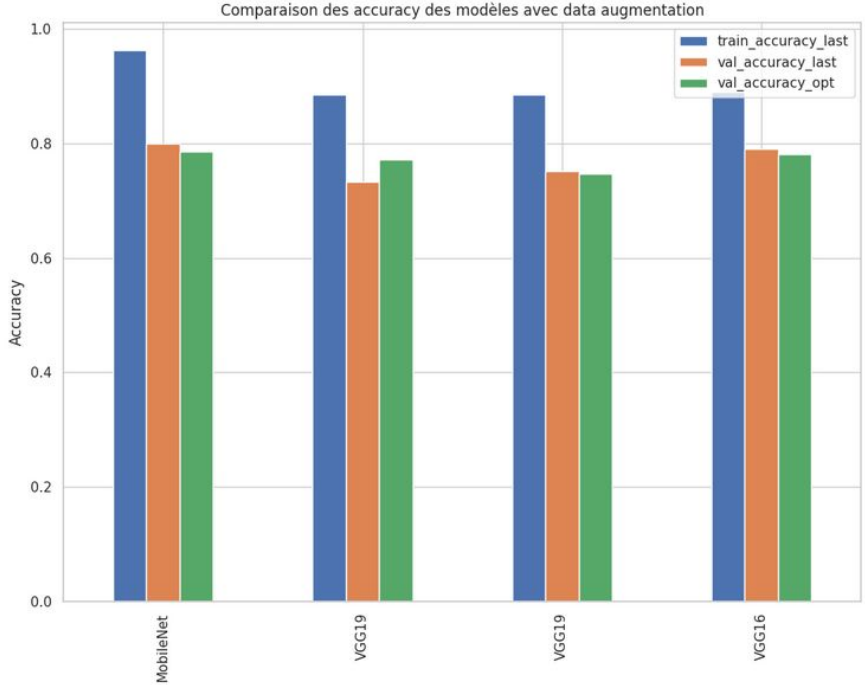


3. Classification supervisée des images

3.3. Synthèse comparative

Comparaison graphique :

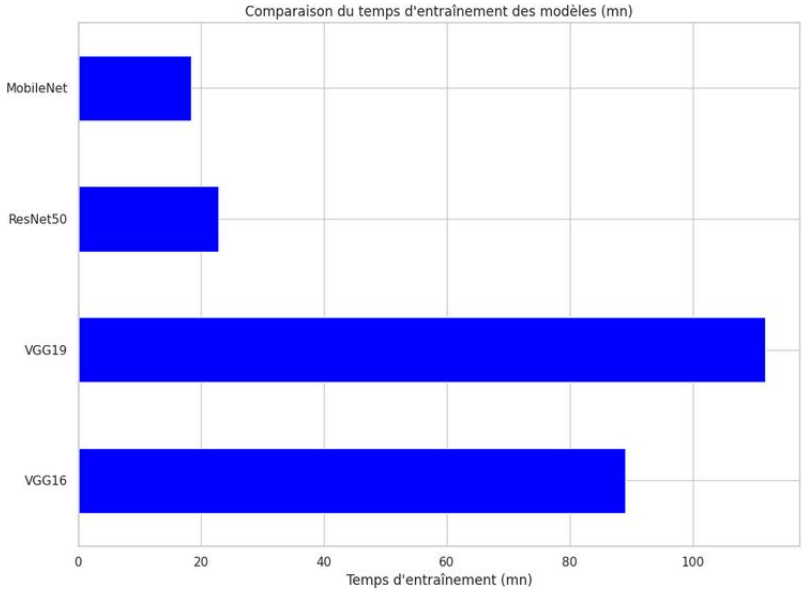


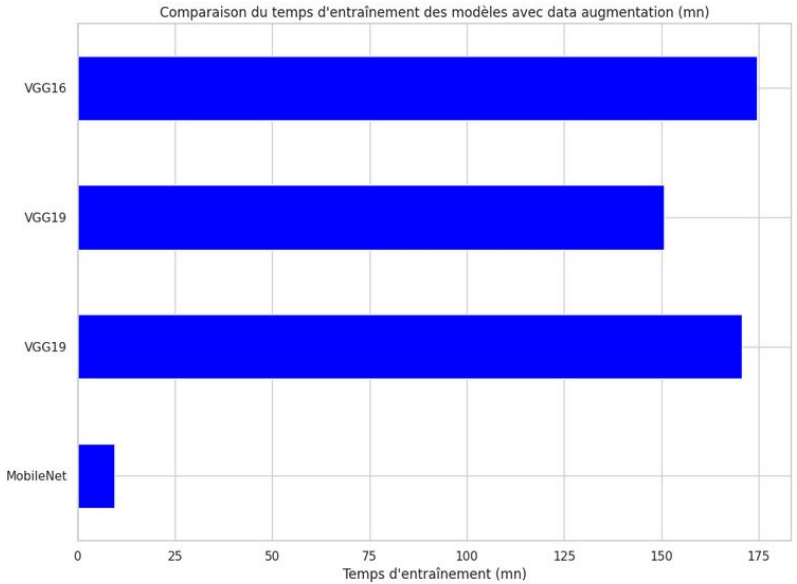


3. Classification supervisée des images

3.3. Synthèse comparative

Comparaison des temps d'entraînement :





3. Classification supervisée des images

3.3. Synthèse comparative

	Sans data augmentation					Avec data augmentation				
Modèles	Val loss opt	Train Accuracy	Val accuracy last	Val accuracy opt	Temps (mn)	Val loss opt	Train Accuracy	Val accuracy last	Val accuracy opt	Temps (mn)
VGG16	1.7684	0.9952	0.7809	0.7904	89	0.7180	0.8905	0.7905	0.7810	175
VGG19	1.8284	1	0.7762	0.7524	112	0.7770	0.8857	0.7524	0.7476	151
Mobile_Net	0.9315	0.9857	0.6952	0.7000	18	0.6351	0.9635	0.8000	0.7857	9
ResNet50	0.9116	0.9937	0.7476	0.7429	23	1.6670	0.4635	0.3714	0.3905	25

3. Classification supervisée des images

3.3. Synthèse comparative

- La mise en place de data augmentation permet de réduire un peu l'overfitting, même s'il reste bien présent pour certains modèles ;
- Chute des performances du modèle ResNet50 ;
- Le **choix se porterait vers le modèle MobileNet** en raison de son **temps d'entraînement nettement inférieur** aux autres pour des résultats comparable ;
- Les **modèles VGG16 et VGG19** donnent également de **bons résultats**, même si leur **temps d'entraînement est très élevé** ;
- Il est sans doute possible d'optimiser les performances et d'améliorer les résultats en poursuivant le finetuning de certains hyperparamètres.

4. Requêtes API

Objectif : Réaliser une requête (API : Edamam Food and Grocery Database)permettant de collecter les 10 premiers produits à base de champagne ayant une sélection de champs renseignés : foodId, label, category, foodContentsLabel, image.

- On a utilisé le script d'exemple fourni par l'API pour réaliser une requête en Python, on s'est assuré de ne limiter notre requête qu'aux ingrédients "Champagne" :
- La 1e requête n'était pas concluante car elle ne ressortait que les résultats de la 1e page fournie par l'API ;
- Après modification et meilleure compréhension du fonctionnement de l'API, nous avons obtenu les résultats voulus :

```
# On ne récupère que les données qui nous intéressent pour le projet -
# on "filtre" sur l'ingrédient "champagne" :
querystring = {"ingr" : "champagne"}
```

	foodId	label	category	foodContentsLabel	image
2	food_b3dyababjo54xobm6r8jzbghjgqe	Champagne Vinaigrette, Champagne	Packaged foods	INGREDIENTS: WATER; CANOLA OIL; CHAMPAGNE VINE...	https://www.edamam.com/food-img/d88/d88b64d973...
5	food_bmu5dmkazwuvpaa5prh1daa8jxs0	Champagne Dressing, Champagne	Packaged foods	SOYBEAN OIL; WHITE WINE (PRESERVED WITH SULFIT...	https://www.edamam.com/food-img/ab2/ab2459fc2a...
16	food_bu12urpbtuo9v6b4jpvk2a1fh4hh	Champagne Simply Dressed Vinaigrette, Champagne	Packaged foods	FILTERED WATER; CANOLA OIL; CHAMPAGNE AND WHIT...	https://www.edamam.com/food-img/736/736a3e27a6...
38	food_b48c55sagj89z4afsne5dar76h4x	Champagne Vinegar	Packaged foods	CHAMPAGNE VINEGAR DILUTED WITH WATER TO 7% ACI...	https://www.edamam.com/food-img/ad8/ad8c8d6ba8...
51	food_bb0nrsgbser5g4ac2enxib1deyh8	Champagne Mustard	Packaged foods	DIJON MUSTARD (VINEGAR; WATER; MUSTARD SEED; S...	https://www.edamam.com/food-img/775/775b39c0b0...
57	food_arm1rlyb5v81v6arurmjqb6xiy9t	Light Champagne Dressing, Light Champagne	Packaged foods	WATER; SOYBEAN OIL; WHITE WINE (PRESERVED WITH...	https://www.edamam.com/food-img/ee0/ee0475f45b...
59	food_bzb3i0lbxkz4nnbv10dsladd965e	Inglenook Champagne	Packaged foods	Dealcoholized Champagne: Water; Grape Juice Co...	https://www.edamam.com/food-img/cb3/cb33b008db...
65	food_byl67wcbbfw82ua6j1n7oa6ago4a	Cola Champagne	Packaged foods	CARBONATED WATER; HIGH FRUCTOSE CORN SYRUP; AR...	https://www.edamam.com/food-img/f82/f82d164536...
72	food_b3iaz2gav39j6abdpavjka0qgz2e	Cola Champagne	Packaged foods	CARBONATED WATER; HIGH FRUCTOSE CORN SYRUP; AR...	https://www.edamam.com/food-img/b4b/b4b747a25b...
79	food_bsck64rasshe3aag46w4dbq3ip81	Champagne Reserve Vinegar	Packaged foods	CHAMPAGNE VINEGAR	https://www.edamam.com/food-img/d2d/d2dcfa4b43...

4. Requêtes API

Respect des 6 grands principes RGPD

Nous trouvons sur le site de la CNIL les 6 grands principes du **RGPD**, à savoir :

1) Ne collectez que les données vraiment nécessaires pour atteindre votre objectif
2) Soyez transparent
3) Organisez et facilitez l'exercice des droits des personnes
4) Fixez des durées de conservation
5) Sécurisez les données et identifiez les risques
6) Inscrivez la mise en conformité dans une démarche continue

Par ailleurs la CNIL définit la donnée personnelle comme : « toute information se rapportant à une personne physique identifiée ou identifiable ».”

Notre travail ne concerne pas ce type de données.

Source : https://www.cnil.fr/fr/comprendre-le-rgpd/les-six-grands-principes-du-rgpd

Conclusion

- Travail de preprocessing de textes et d'images permettant de conclure à la faisabilité d'une démarche de classification supervisée ;
- Classification des produits à partir de leur image, via du transfer learning / utilisation de plusieurs modèles pré-entraînés ;
- Résultats plutôt concluants même si encore optimisables ;
- Réalisation d'une requête sur une API permettant la collecte des informations nécessaires ;
- Respect des normes RGPD ;
- Les textes et images utilisés ne relèvent pas d'une propriété intellectuelle.

Merci de votre attention.