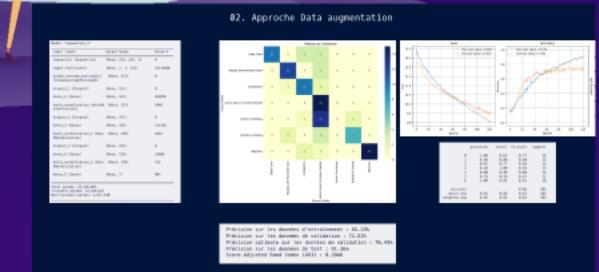
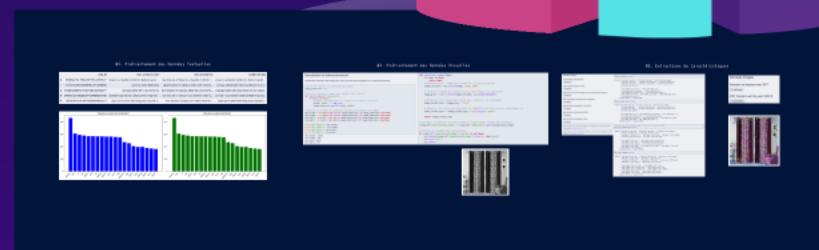


Classifiez automatiquement des biens de consommation



Introduction
"Place de marché" une marketplace e-commerce. Souhaite Automatiser l'attribution des catégories pour améliorer l'expérience utilisateur.

Table des matières		
A. Etude de faisabilité	B. Classification experte	C. Scénario API
• Évaluation des services	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des données	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des utilisateurs	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des catégories	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation du processus	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation du résultat	• Méthode par Projet experte	• Méthode de classification par API



Introduction

"Place de marché" une marketplace e-commerce.
Souhaite Automatiser l'attribution des catégories pour améliorer l'expérience utilisateur.



Classifiez automatiquement des biens de consommation



Introduction
"Place de marché" une marketplace e-commerce. Souhaite Automatiser l'attribution des catégories pour améliorer l'expérience utilisateur.

Table des matières		
A. Etude de faisabilité	B. Classification experte	C. Scénario API
• Évaluation des services	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des données	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des utilisateurs	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des catégories	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation du processus	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation du résultat	• Méthode par Projet experte	• Méthode de classification par API

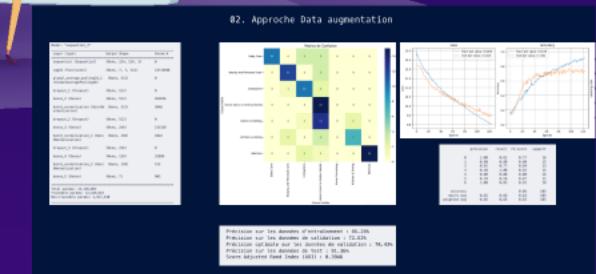


Table des matières

A. Etude de faisabilité

- 1 Prétraitement des Données Textuelles
- 2 Prétraitement des Données Visuelles
- 3 Extraction des Caractéristiques
- 4 Réduction Dimensionnelle et conclusion

B. Classification supervisée

- 1 Approche par Préparation Initiale des Images
- 2 Approche Data augmentation

C. Script API

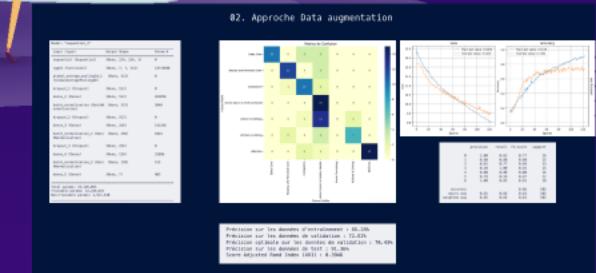
- 1 Teste de la collecte de produits
- 2 Collecte des informations sur les produits à base de "champagne"
- 3 Visualisations des résultats dans un dataframe

Classifiez automatiquement des biens de consommation



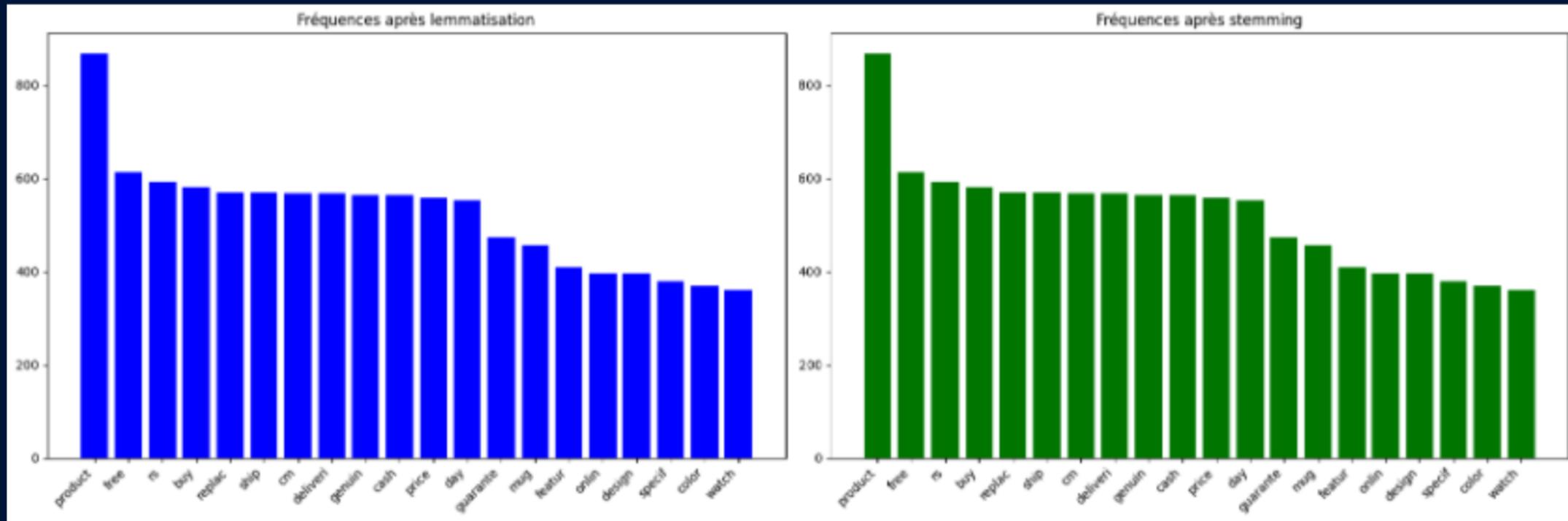
Introduction
"Place de marché" une marketplace e-commerce. Souhaite Automatiser l'attribution des catégories pour améliorer l'expérience utilisateur.

Table des matières		
A. Etude de faisabilité	B. Classification experte	C. Scénario API
• Évaluation des services	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des besoins	• Méthode par Projet experte	• Méthode de classification par API
• Analyse des données	• Méthode par Projet experte	• Méthode de classification par API
• Conclusion	• Méthode par Projet experte	• Méthode de classification par API



01. Prétraitement des Données Textuelles

	uniq_id	text_product_name	text_description	combined_text
0	55b85ea15a1536d46b7190ad6fff8ce7	elegance polyester multicolor abstract eyelet ...	key features of elegance polyester multicolor ...	elegance polyester multicolor abstract eyelet ...
1	7b72c92c2f6c40268628ec5f14c6d590	sathiya s cotton bath towel	specifications of sathiya s cotton bath towel b...	sathiya s cotton bath towel specifications of s...
2	64d5d4a258243731dc7bbb1eef49ad74	eurospa cotton terry face towel set	key features of eurospa cotton terry face tow...	eurospa cotton terry face towel set key featur...
3	d4684dcdc759dd9cdf41504698d737d8	santosh royal fashion cotton printed king size...	key features of santosh royal fashion cotton p...	santosh royal fashion cotton printed king size...
4	6325b6870c54cd47be6ebfbffa620ec7	jaipur print cotton floral king sized double b...	key features of jaipur print cotton floral kin...	jaipur print cotton floral king sized double b...



02. Prétraitement des Données Visuelles

Normalisation et redimensionnement

Analyse des dimensions des images pour choisir une taille cible appropriée pour le redimensionnement.

```
# Liste pour stocker les dimensions des images
image_dimensions = []

# Parcourir toutes les images déjà chargées dans study_df
for img in study_df['image_data']:
    if img is not None:
        # Obtenir les dimensions de l'image et les ajouter à la liste
        height, width, _ = img.shape
        image_dimensions.append((height, width))

# Calculer et afficher les dimensions maximales et minimales des images
max_height = max(dim[0] for dim in image_dimensions) if image_dimensions else None
min_height = min(dim[0] for dim in image_dimensions) if image_dimensions else None
max_width = max(dim[1] for dim in image_dimensions) if image_dimensions else None
min_width = min(dim[1] for dim in image_dimensions) if image_dimensions else None

print("Max Height:", max_height)
print("Min Height:", min_height)
print("Max Width:", max_width)
print("Min Width:", min_width)
```

```
Max Height: 11042
Min Height: 160
Max Width: 8484
Min Width: 145
```

```
def preprocess_image(image):
    if image is None:
        return None

    # Redimensionner l'image pour normaliser la taille d'entrée
    image_resized = cv2.resize(image, (256, 256))

    # Convertir l'image en niveaux de gris pour réduire la complexité
    image_gray = cv2.cvtColor(image_resized, cv2.COLOR_RGB2GRAY)

    # Normaliser les valeurs des pixels pour faciliter le traitement
    image_normalized = image_gray / 255.0

    # Convertir en 8 bits pour maintenir la compatibilité avec certains algorithmes CV
    image_normalized = (image_normalized * 255).astype('uint8')

    return image_normalized

# Appliquer la fonction de prétraitement à chaque image stockée dans study_df
study_df['preprocessed_image'] = study_df['image_data'].apply(preprocess_image)

# Afficher un exemple pour vérification
if study_df['preprocessed_image'].iloc[0] is not None:
    plt.imshow(study_df['preprocessed_image'].iloc[0], cmap='gray')
    plt.axis('off')
    plt.show()
```



03. Extractions de Caractéristiques

Données Textes

Approche Bag of Words (BoW)

+ 3 cells hidden

Latent Dirichlet Allocation (LDA)

+ 2 cells hidden

Approche TF-IDF (Term Frequency-Inverse Document Frequency)

+ 3 cells hidden

NMF (Factorisation de Matrices Non-négatives)

+ 2 cells hidden

Approche Word Embeddings (Word2Vec)

+ 1 cell hidden

Approche Word Embeddings (BERT)

+ 1 cell hidden

Approche Word Embeddings (USE)

+ 1 cell hidden

Vérifier le résultat après l'ajout des features NMF

study_df.columns.tolist()

Vérification du résultat après l'ajout des features LDA

study_df.columns.tolist()

```
df_bow_stemmed.columns  
  
Index(['stem_desc_bow_aa', 'stem_desc_bow_aaa', 'stem_desc_bow_aapno',  
       'stem_desc_bow_aari', 'stem_desc_bow_aarika', 'stem_desc_bow_abil',  
       'stem_desc_bow_abl', 'stem_desc_bow_abod', 'stem_desc_bow_abras',  
       'stem_desc_bow_abroad',  
       ...  
       'stem_combo_bow_zikrak', 'stem_combo_bow_zinc',  
       'stem_combo_bow_zingalala', 'stem_combo_bow_zip',  
       'stem_combo_bow_zipexterior', 'stem_combo_bow_zipper',  
       'stem_combo_bow_zone', 'stem_combo_bow_zoom', 'stem_combo_bow_zora',  
       'stem_combo_bow_zyxel'],  
      dtype='object', length=9227)
```

```
df_bow_lemmatized.columns  
  
Index(['lem_desc_bow_aa', 'lem_desc_bow_aaa', 'lem_desc_bow_aapno',  
       'lem_desc_bow_aari', 'lem_desc_bow_aarika', 'lem_desc_bow_ability',  
       'lem_desc_bow_able', 'lem_desc_bow_abode', 'lem_desc_bow_abrasion',  
       'lem_desc_bow_abroad',  
       ...  
       'lem_combo_bow_zinc', 'lem_combo_bow_zingalala', 'lem_combo_bow_zip',  
       'lem_combo_bow_zipexterior', 'lem_combo_bow_zipper',  
       'lem_combo_bow_zipped', 'lem_combo_bow_zone', 'lem_combo_bow_zoom',  
       'lem_combo_bow_zora', 'lem_combo_bow_zyxel'],  
      dtype='object', length=10376)
```

Vérification du résultat après l'ajout des features TF-IDF

df_tfidf_lemmatized.columns

```
Index(['lem_desc_tfidf_aa', 'lem_desc_tfidf_aaa', 'lem_desc_tfidf_aapno',  
       'lem_desc_tfidf_aari', 'lem_desc_tfidf_aarika',  
       'lem_desc_tfidf_ability', 'lem_desc_tfidf_able', 'lem_desc_tfidf_abode',  
       'lem_desc_tfidf_abrasion', 'lem_desc_tfidf_abroad',  
       ...  
       'lem_combo_tfidf_zinc', 'lem_combo_tfidf_zingalala',  
       'lem_combo_tfidf_zip', 'lem_combo_tfidf_zipexterior',  
       'lem_combo_tfidf_zipper', 'lem_combo_tfidf_zipped',  
       'lem_combo_tfidf_zone', 'lem_combo_tfidf_zoom', 'lem_combo_tfidf_zora',  
       'lem_combo_tfidf_zyxel'],  
      dtype='object', length=10376)
```

```
df_tfidf_stemmed.columns  
  
Index(['stem_desc_tfidf_aa', 'stem_desc_tfidf_aaa', 'stem_desc_tfidf_aapno',  
       'stem_desc_tfidf_aari', 'stem_desc_tfidf_aarika',  
       'stem_desc_tfidf_abil', 'stem_desc_tfidf_abl', 'stem_desc_tfidf_abod',  
       'stem_desc_tfidf_abras', 'stem_desc_tfidf_abroad',  
       ...  
       'stem_combo_tfidf_zikrak', 'stem_combo_tfidf_zinc',  
       'stem_combo_tfidf_zingalala', 'stem_combo_tfidf_zip',  
       'stem_combo_tfidf_zipexterior', 'stem_combo_tfidf_zipper',  
       'stem_combo_tfidf_zone', 'stem_combo_tfidf_zoom',  
       'stem_combo_tfidf_zora', 'stem_combo_tfidf_zyxel'],  
      dtype='object', length=9227)
```

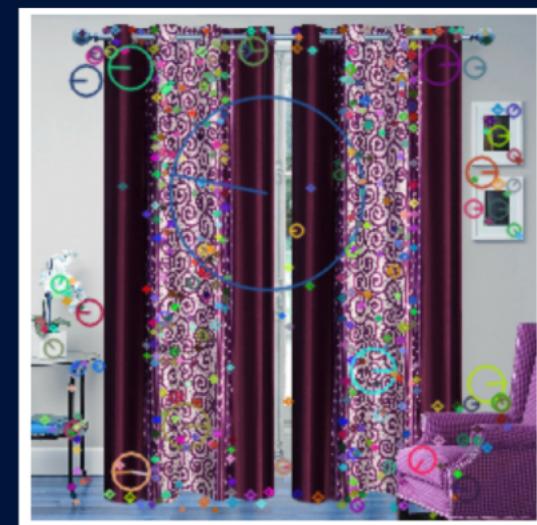
Données Images

Extraction de features avec SIFT

+ 1 cell hidden

CNN Transfer Learning avec VGG16

+ 2 cells hidden

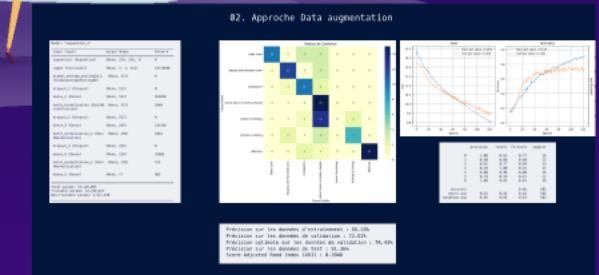
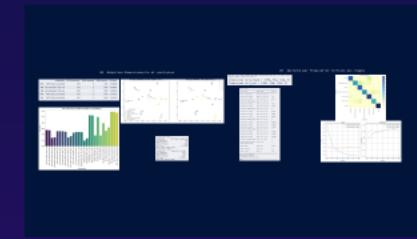


Classifiez automatiquement des biens de consommation



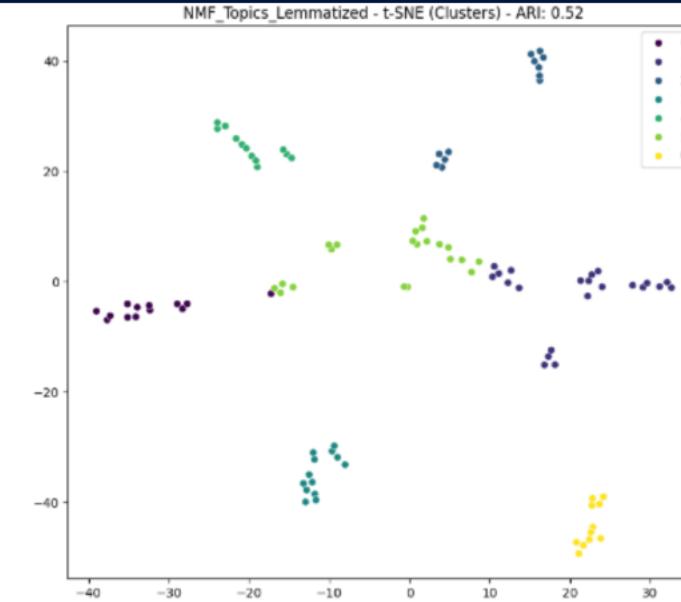
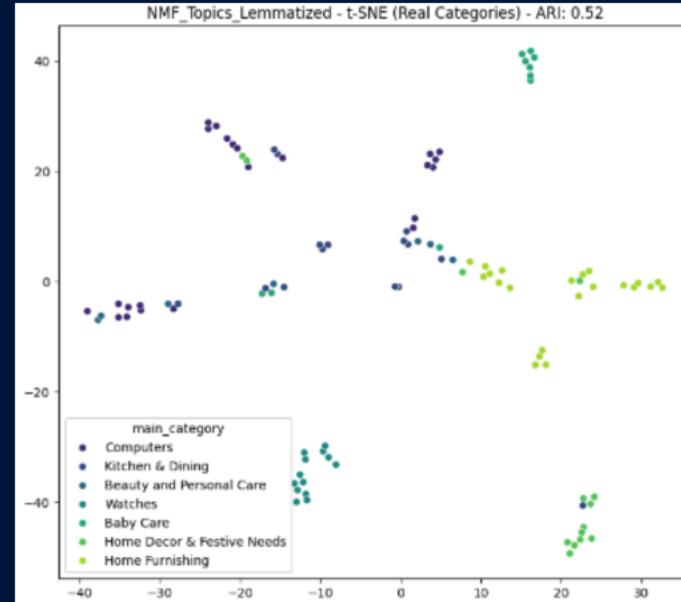
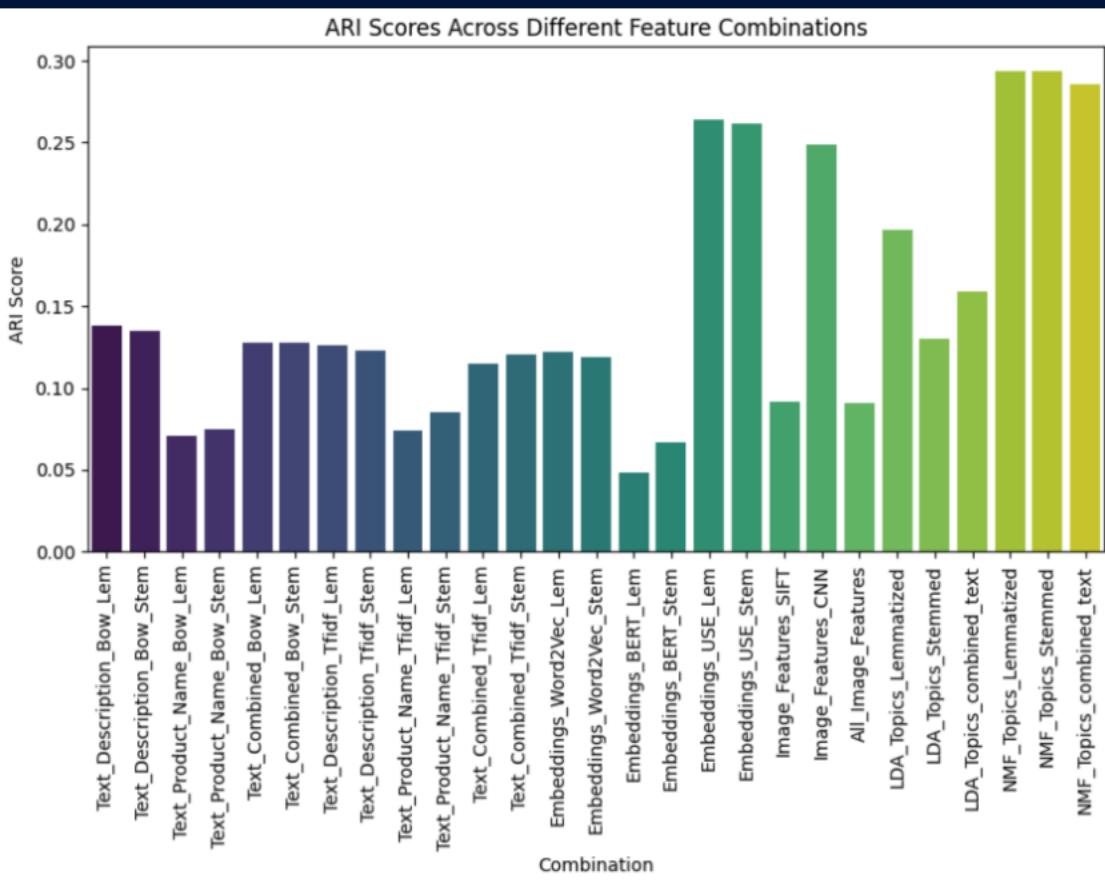
Introduction
"Place de marché" une marketplace e-commerce. Souhaite Automatiser l'attribution des catégories pour améliorer l'expérience utilisateur.

Table des matières		
A. Etude de faisabilité	B. Classification experte	C. Scénario API
• Évaluation des services	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des besoins	• Méthode par Projet experte	• Méthode de classification par API
• Analyse des données	• Méthode par Projet experte	• Méthode de classification par API
• Conclusion	• Méthode par Projet experte	• Méthode de classification par API



04. Réduction Dimensionnelle et conclusion

	Combination	PCA Components	TSNE Perplexity	TSNE Iterations	ARI Score
1188	NMF_Topic_Lemmatized	0.99	5	1000	0.523595
289	Text_Description_Tfidf_Lem	0.85	5	3000	0.450601
290	Text_Description_Tfidf_Lem	0.85	5	5000	0.436434
1189	NMF_Topic_Lemmatized	0.99	5	3000	0.424572
1190	NMF_Topic_Lemmatized	0.99	5	5000	0.424572



ARI max pour text:
 Combination NMF_Topic_Lemmatized
 PCA Components 0.99
 TSNE Perplexity 5
 TSNE Iterations 1000
 ARI Score 0.523595
 Name: 1188, dtype: object

ARI max pour images:
 Combination Image_Features_CNN
 PCA Components 0.99
 TSNE Perplexity 30
 TSNE Iterations 3000
 ARI Score 0.374455
 Name: 952, dtype: object

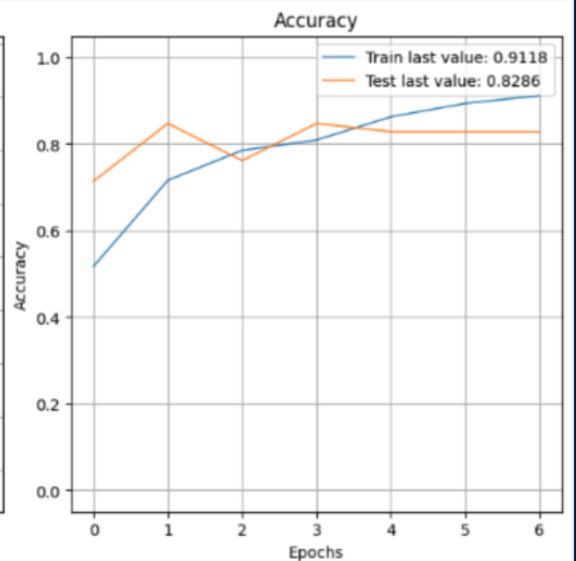
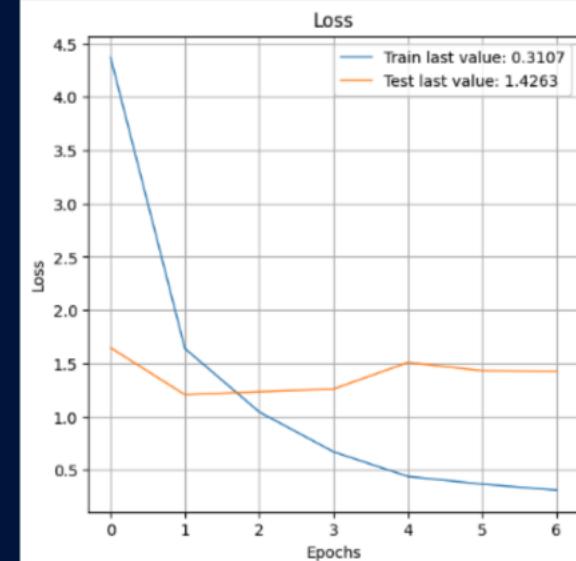
01. Approche par Préparation Initiale des Images

Skipping large image 5518124b75d6c6dfee4c2e4c0cf716a.jpg with size (8484, 11042)
 Images prepared: (1049, 224, 224, 3)

Dimensions de X_train : (839, 224, 224, 3)

Dimensions de X_val : (105, 224, 224, 3)

Model: "model"		
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[None, 224, 224, 3]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 256)	131328
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 7)	1799
<hr/>		
Total params: 14,847,815		
Trainable params: 133,127		
Non-trainable params: 14,714,688		
None		

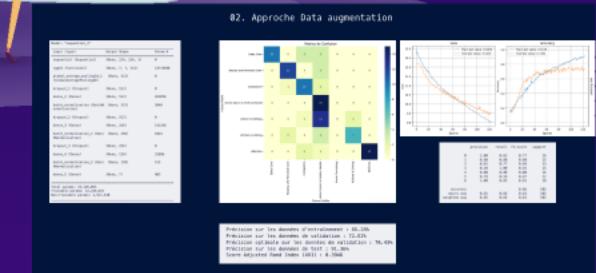


Classifiez automatiquement des biens de consommation



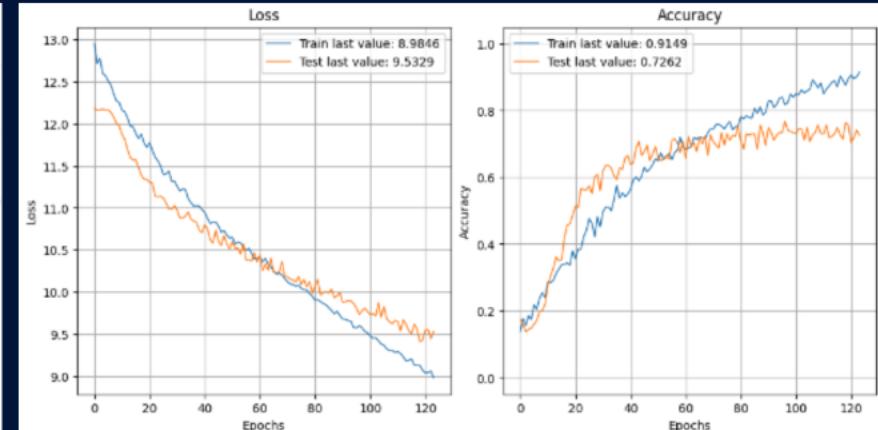
Introduction
"Place de marché" une marketplace e-commerce. Souhaite Automatiser l'attribution des catégories pour améliorer l'expérience utilisateur.

Table des matières		
A. Etude de faisabilité	B. Classification experte	C. Scénario API
• Évaluation des services	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des données	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des utilisateurs	• Méthode de classification par API	• Méthode de classification par API
• Évaluation des coûts		



02. Approche Data augmentation

Model: "sequential_1"		
Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 224, 224, 3)	0
vgg16 (Functional)	(None, 7, 7, 512)	14714688
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 512)	0
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
batch_normalization (BatchNormalization)	(None, 512)	2048
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 256)	131328
batch_normalization_1 (BatchNormalization)	(None, 256)	1024
dropout_3 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 128)	32896
batch_normalization_2 (BatchNormalization)	(None, 128)	512
dense_5 (Dense)	(None, 7)	903
<hr/>		
Total params:	15,146,055	
Trainable params:	12,228,615	
Non-trainable params:	2,917,440	



	precision	recall	f1-score	support
0	1.00	0.62	0.77	16
1	0.80	0.80	0.80	15
2	0.62	0.77	0.69	13
3	0.38	1.00	0.55	15
4	0.00	0.00	0.00	16
5	0.78	0.58	0.67	12
6	1.00	0.83	0.91	18
accuracy				0.66
macro avg	0.65	0.66	0.63	105
weighted avg	0.66	0.66	0.63	105

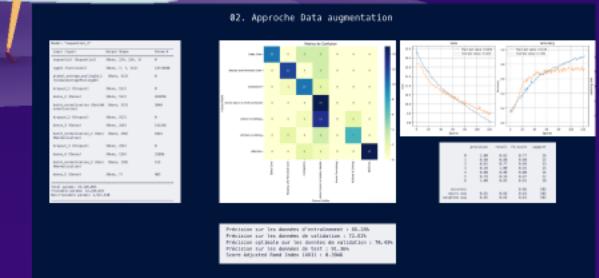
Précision sur les données d'entraînement : 86.28%
 Précision sur les données de validation : 72.62%
 Précision optimale sur les données de validation : 76.43%
 Précision sur les données de test : 91.36%
 Score Adjusted Rand Index (ARI) : 0.3946

Classifiez automatiquement des biens de consommation



Introduction
"Place de marché" une marketplace e-commerce. Souhaite Automatiser l'attribution des catégories pour améliorer l'expérience utilisateur.

Table des matières		
A. Etude de faisabilité	B. Classification experte	C. Scénario API
• Évaluation des services	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des données	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des utilisateurs	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation des catégories	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation du processus	• Méthode par Projet experte	• Méthode de classification par API
• Évaluation du résultat	• Méthode par Projet experte	• Méthode de classification par API



01. Teste de la collecte de produits

Teste de la collecte de produits

```
url = "https://edamam-food-and-grocery-database.p.rapidapi.com/api/food-database/v2/parser"

querystring = {"nutrition-type":"cooking","category[0]":"generic-foods","health[0]":"alcohol-free"}

headers = {
    "X-RapidAPI-Key": "4f03f5d8e3msh960222cd30adde9p17b8abjsn3511ddad03aa",
    "X-RapidAPI-Host": "edamam-food-and-grocery-database.p.rapidapi.com"
}

response = requests.get(url, headers=headers, params=querystring)

print(response.json())

{'text': '', 'parsed': [], 'hints': [{'food': {'foodId': 'food_bwrgmmqau78xrdazxx79obeezumz', 'uri': 'http://www.edamam.com/ontologies/edamam.owl#Food_01001', 'label': 'Butter, Salted', 'knownAs': 'Butter, salted', 'nutrients': {'ENERGY_KCAL': 717.0, 'PROCNT': 0.85, 'FAT': 81.1, 'CHOCDF': 0.06, 'FIBTG': 0.0}, 'category': 'Generic foods', 'categoryLabel': 'food', 'image': 'https://www.edamam.com/food-img/515/515af390107678fce1533a31ee4cc35b.jpeg'}, 'measures': [{"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_serving", "label": "Serving", "weight": 14.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_tablespoon", "label": "Tablespoon", "weight": 14.2}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_cup", "label": "Cup", "weight": 227.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_stick", "label": "Stick", "weight": 113.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_gram", "label": "Gram", "weight": 1.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_ounce", "label": "Ounce", "weight": 28.349523125}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_pound", "label": "Pound", "weight": 453.59237}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_unit", "label": "Whole", "weight": 113.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_pat", "label": "Pat", "weight": 5.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_kilogram", "label": "Kilogram", "weight": 1000.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_default", "label": "Default", "weight": 0.0136}]}}, {'food': {'foodId': 'food_blrpqo1bp94h17apq4prgbhcuhex', 'uri': 'http://www.edamam.com/ontologies/edamam.owl#Food_01001', 'label': 'Butter, Unsalted', 'knownAs': 'Butter, unsalted', 'nutrients': {'ENERGY_KCAL': 772.0, 'PROCNT': 0.85, 'FAT': 81.1, 'CHOCDF': 0.06, 'FIBTG': 0.0}, 'category': 'Generic foods', 'categoryLabel': 'food', 'image': 'https://www.edamam.com/food-img/515/515af390107678fce1533a31ee4cc35b.jpeg'}, 'measures': [{"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_serving", "label": "Serving", "weight": 14.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_tablespoon", "label": "Tablespoon", "weight": 14.2}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_cup", "label": "Cup", "weight": 227.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_stick", "label": "Stick", "weight": 113.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_gram", "label": "Gram", "weight": 1.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_ounce", "label": "Ounce", "weight": 28.349523125}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_pound", "label": "Pound", "weight": 453.59237}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_unit", "label": "Whole", "weight": 113.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_pat", "label": "Pat", "weight": 5.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_kilogram", "label": "Kilogram", "weight": 1000.0}, {"uri": "http://www.edamam.com/ontologies/edamam.owl#Measure_default", "label": "Default", "weight": 0.0136}]}]
```

02. Collecte des informations sur les produits à base de "champagne"

Collecte des informations sur les produits à base de "champagne"

```
import requests
import csv

url = "https://edamam-food-and-grocery-database.p.rapidapi.com/api/food-database/v2/parser"
querystring = {"ingr": "champagne"}
headers = {
    "X-RapidAPI-Key": "4f83f5d8e3esh96822cd38adde9p17b8abjsn351lddad83aa",
    "X-RapidAPI-Host": "edamam-food-and-grocery-database.p.rapidapi.com"
}

response = requests.get(url, headers=headers, params=querystring)
data = response.json()

# Ouverture d'un fichier CSV pour écrire les données des produits
with open('Data/champagne_products.csv', mode='w', newline='', encoding='utf-8') as file:
    writer = csv.writer(file)
    writer.writerow(['foodId', 'label', 'category', 'foodContentsLabel', 'image'])

    # Extraction des données de 'parsed'
    for item in data.get('parsed', []):
        food = item.get('food', {})
        foodId = food.get('foodId', '')
        label = food.get('label', '')
        category = food.get('category', '')
        foodContentsLabel = food.get('foodContentsLabel', '')
        image = food.get('image', '')
        writer.writerow([foodId, label, category, foodContentsLabel, image])

    # Extraction des données de 'hints'
    for item in data.get('hints', []):
        food = item.get('food', {})
        foodId = food.get('foodId', '')
        label = food.get('label', '')
        category = food.get('category', '')
        foodContentsLabel = food.get('foodContentsLabel', '')
        image = food.get('image', '')
        writer.writerow([foodId, label, category, foodContentsLabel, image])

print("Les données ont été enregistrées dans 'champagne_products.csv'")

Les données ont été enregistrées dans 'champagne_products.csv'.

print(data)

{'text': 'champagne', 'parsed': [{ 'food': { 'foodId': 'food_a656mk2a5dmqb2adiamu6beihduu', 'uri': 'http://www.edamam.com/ontologies/edamam.owl#Food_table_white_wine', 'label': 'Champagne', 'knownAs': 'dry white wine', 'nutrients': { 'ENERC_KCAL': 82.0, 'CHOCDF': 2.6, 'FATG': 0.0, 'PROCNT': 0.07, 'FAT': 0.0, 'CHOCDF': 2.6, 'FIBTG': 0.0, 'category': 'Generic foods', 'categoryLabel': 'food', 'image': 'https://www.edamam.com/food-img/a71/a718cf3c52add5221289291f1324d2ab.jpg' }, 'hints': { ('foodId': 'food_a656mk2a5dmqb2adiamu6beihduu', 'uri': 'http://www.edamam.com/ontologies/edamam.owl#Food_table_white_wine', 'label': 'Champagne', 'knownAs': 'dry white wine', 'nutrients': { 'ENERC_KCAL': 82.0, 'PROCNT': 0.07, 'FAT': 0.0, 'CHOCDF': 2.6, 'FIBTG': 0.0, 'category': 'Generic foods', 'categoryLabel': 'food', 'image': 'https://www.edamam.com/food-img/a71/a718cf3c52add5221289291f1324d2ab.jpg' }, 'measures': { ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_serving', 'label': 'Serving', 'weight': 147.0, 'uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_unit', 'label': 'Gram', 'weight': 1.0, 'uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_gram', 'label': 'Gram', 'weight': 1.0, 'uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_ounce', 'label': 'Ounce', 'weight': 28.349523125}, ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_pound', 'label': 'Pound', 'weight': 453.59237}, ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_kilogram', 'label': 'Kilogram', 'weight': 1000.0}, ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_cup', 'label': 'Cup', 'weight': 235.2}, ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_bottle', 'label': 'Bottle', 'weight': 745.599200575638}, ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_fluid_ounce', 'label': 'Fluid ounce', 'weight': 29.4}], 'food': { 'foodId': 'food_b753ithaadb8psbt8w2k9aqu06c', 'uri': 'http://www.edamam.com/ontologies/edamam.owl#Food_FDCBR_744858', 'label': 'Champagne Vinaigrette, Champagne', 'knownAs': 'CHAMPAGNE VINAIGRETTE, CHAMPAGNE', 'nutrients': { 'ENERC_KCAL': 571.0, 'PROCNT': 0.0, 'FAT': 64.29000091552734, 'CHOCDF': 7.13999866485596}, 'brand': 'SoFine Food', 'category': 'Packaged foods', 'categoryLabel': 'food', 'foodContentsLabel': 'OLIVE OIL; BALSAMIC VINEGAR; CHAMPAGNE VINEGAR; GARLIC; DIJON MUSTARD; SEA SALT.', 'servingSizes': { ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_gram', 'label': 'Gram', 'quantity': 14.0}, ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_tablespoon', 'label': 'Tablespoon', 'quantity': 1.0)}, 'measures': { ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_serving', 'label': 'Serving', 'weight': 14.0}, ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_tablespoon', 'label': 'Tablespoon', 'weight': 34.0}, ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_gram', 'label': 'Gram', 'weight': 1.0}, ('uri': 'http://www.edamam.com/ontologies/edamam.owl#Measure_ounce', 'label': 'Ounce', 'weight': 1000.0}), ('food': { 'foodId': 'food_b3dybabj054xobm6r8zbghjqge', 'uri': 'http://www.edamam.com/ontologies/edamam.owl#Food_FDCBR_1069751', 'label': 'Champagne Vinaigrette, Champagne', 'knownAs': 'CHAMPAGNE VINAIGRETTE, CHAMPAGNE', 'nutrients': { 'ENERC_KCAL': 333.0, 'PROCNT': 0.0, 'FAT': 36.66999816894531, 'CHOCDF': 6.670000076293945}, 'brand': 'Maple Grove Farms Of Vermont Inc.', 'category': 'Packaged foods', 'categoryLabel': 'food', 'foodContentsLabel': 'INGREDIENTS: WATER; CANOLA OIL; CHAMPAGNE VINEGAR; SUGAR; OLIVE OIL; SALT; DRIED GARLIC; DRIED SHALLOTS; BLACK PEPPER; XA'} } } ] }
```

03. Visualisations des résultats dans un dataframe

	foodId	label	category	foodContentsLabel	Image
0	food_a656mk2a5dmqb2adiamu6beihduu	Champagne	Generic foods	NaN	https://www.edamam.com/food-img/a71/a718cf3c52...
1	food_a656mk2a5dmqb2adiamu6beihduu	Champagne	Generic foods	NaN	https://www.edamam.com/food-img/a71/a718cf3c52...
2	food_b753ithamdb8psbt0w2k9aquo06c	Champagne Vinaigrette, Champagne	Packaged foods	OLIVE OIL; BALSAMIC VINEGAR; CHAMPAGNE VINEGAR...	NaN
3	food_b3dyababjo54xobm6r8jzbghjgqe	Champagne Vinaigrette, Champagne	Packaged foods	INGREDIENTS: WATER; CANOLA OIL; CHAMPAGNE VINE...	https://www.edamam.com/food-img/d88/d88b64d973...
4	food_a9e0ghsamvoc45bwa2ybsa3gken9	Champagne Vinaigrette, Champagne	Packaged foods	CANOLA AND SOYBEAN OIL; WHITE WINE (CONTAINS S...	NaN
5	food_an4j jueaucpus2a3u1ni8auhe7q9	Champagne Vinaigrette, Champagne	Packaged foods	WATER; CANOLA AND SOYBEAN OIL; WHITE WINE (CON...	NaN
6	food_bmu5dmkazwuvpaa5prh1daa8jxs0	Champagne Dressing, Champagne	Packaged foods	SOYBEAN OIL; WHITE WINE (PRESERVED WITH SULFIT...	https://www.edamam.com/food-img/ab2/ab2459fc2a...
7	food_alpl44taoyv11ra0lic1qa8xculi	Champagne Buttercream	Generic meals	sugar; butter; shortening; vanilla; champagne;...	NaN
8	food_byap67hab6evc3a0f9w1oag3s0qf	Champagne Sorbet	Generic meals	Sugar; Lemon juice; brandy; Champagne; Peach	NaN
9	food_am5egz6aq3fpjlaf8xpdkbc2asis	Champagne Truffles	Generic meals	butter; cocoa; sweetened condensed milk; vanil...	NaN