Classifiez automatiquement des biens de consommation

Objectifs et problématique

- Problématique:

Classification automatique des biens de consommation à partir de leur image ou leur description.

Un test de faisabilité doit être réaliser avant de mettre en place un modèle de classification automatique.

Réaliser un script pour collecter les informations des produits à base de champagne via une API.

- Contraintes:

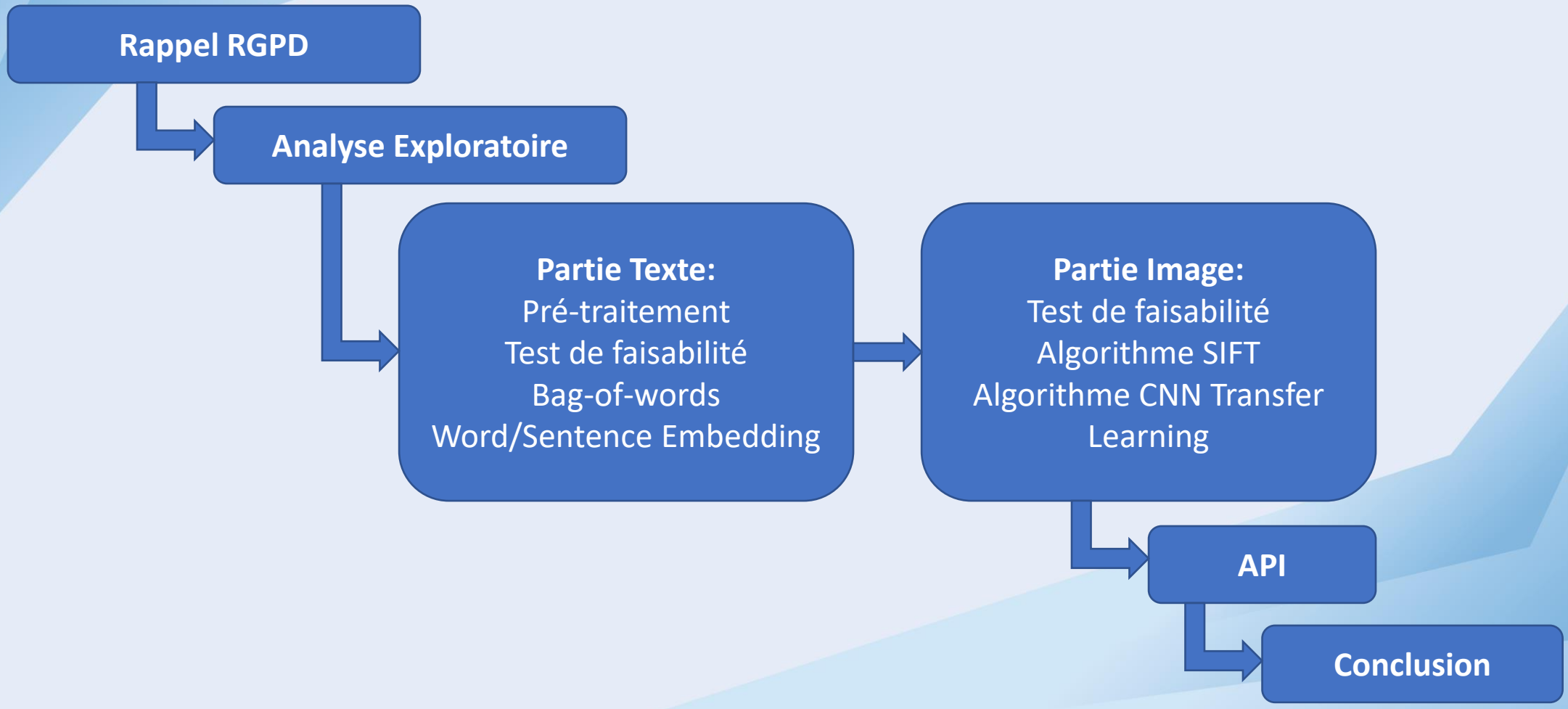
Traitement données texte:

- 2 approches de type « bag-of-words »
- 3 approches de type Word/Sentence Embedding avec Word2Vec, BERT et USE

Traitement données image :

- Un algorithme de type SIFT
- Un algorithme de type CNN Transfer learning

Sommaire


Rappel RGPD
Analyse Exploratoire
Partie Texte:
Pré-traitement
Test de faisabilité
Bag-of-words
Word/Sentence Embedding
Partie Image:
Test de faisabilité
Algorithme SIFT
Algorithme CNN Transfer Learning
API
Conclusion

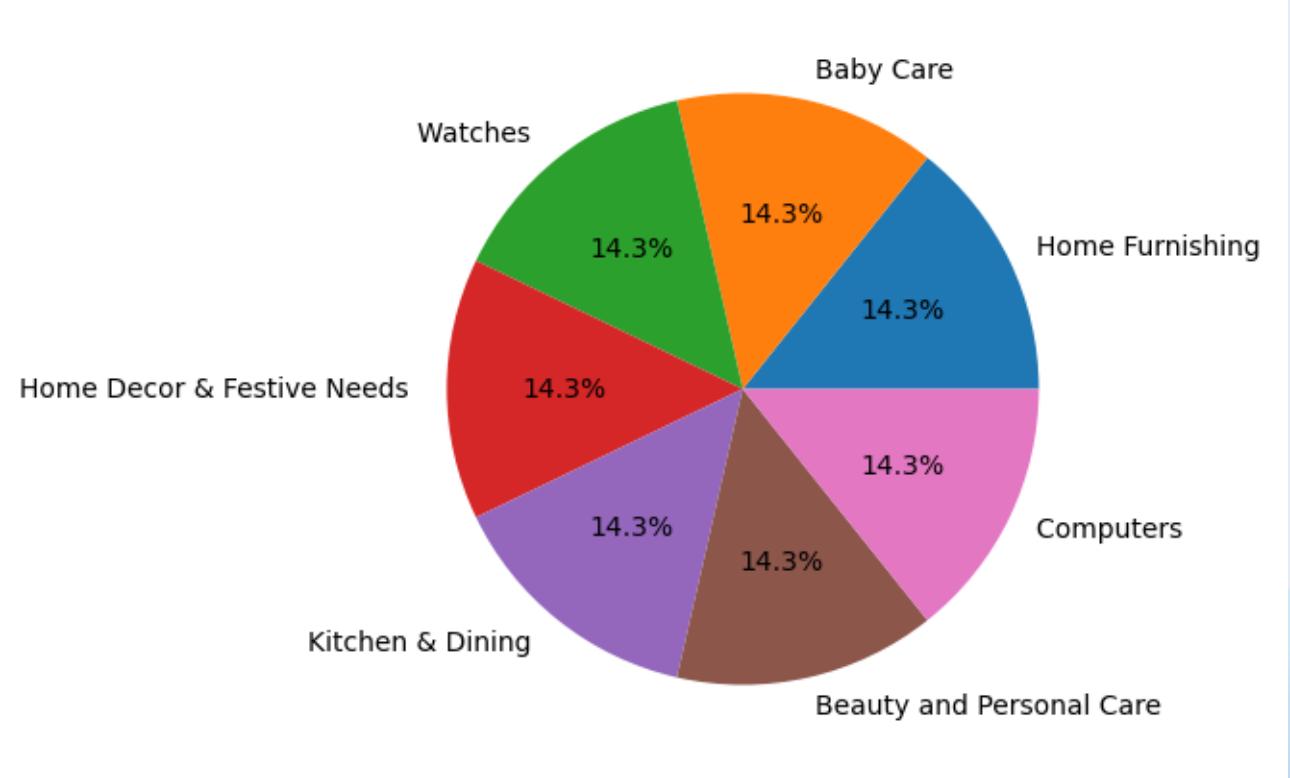
RGPD

Pour simplifier, le Règlement a pour ligne de conduite de :

1. Protéger la vie privée des citoyens européens notamment leurs données et propriétés intellectuelles et numériques
2. Responsabiliser les Directions et sanctionner en cas de manquement
3. Répondre à une volonté citoyenne
4. Protéger les personnes vulnérables

ANALYSE EXPLORATOIRE

Le dataset présente 1050 articles qui sont répartis équitablement en 7 catégories (150 articles par catégorie)



Partie Texte

Méthodologie:

- Pré-traitement des données texte
- Approches Bag-of-words
- Test de faisabilité
- Topic Modeling
- Approches Word Embedding via Doc2Vec, BERT et USE
- Comparaison des différentes approches

• Pré-traitement des données texte

- Fusion du nom du produit et de la description du produit
- Tokenisation et suppression des stopwords

- Lemmatisation:

Normalise les mots d'une même famille à leur forme canonique

- Stemming:

Normalise les mots, en enlevant les affixes des mots pour en garder la racine

Mots d'origine	Lemming	Stemming
Jouer	jouer	jou
Jouant	jouer	jou
Jouerai	jouer	jouer
beautiful	beauty	beauti
beauty	beauty	beauti

Suppression des dropwords

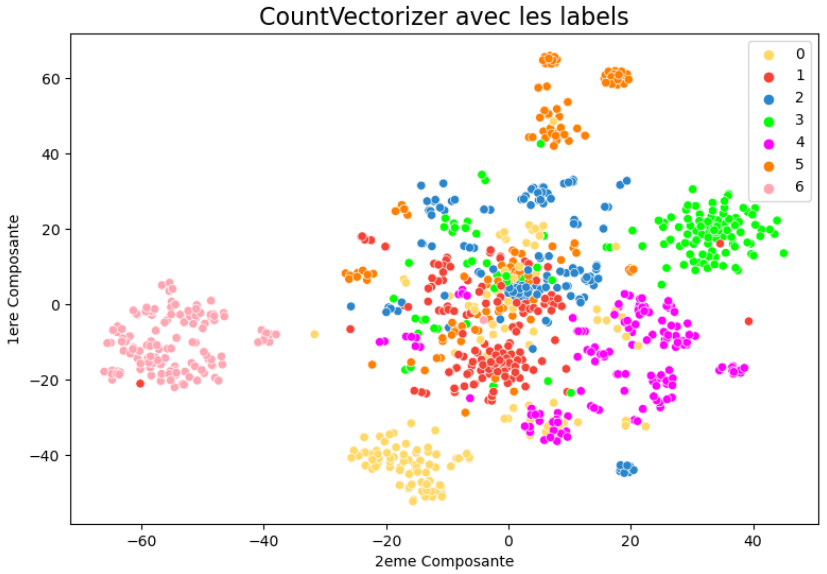
- Méthodologie:

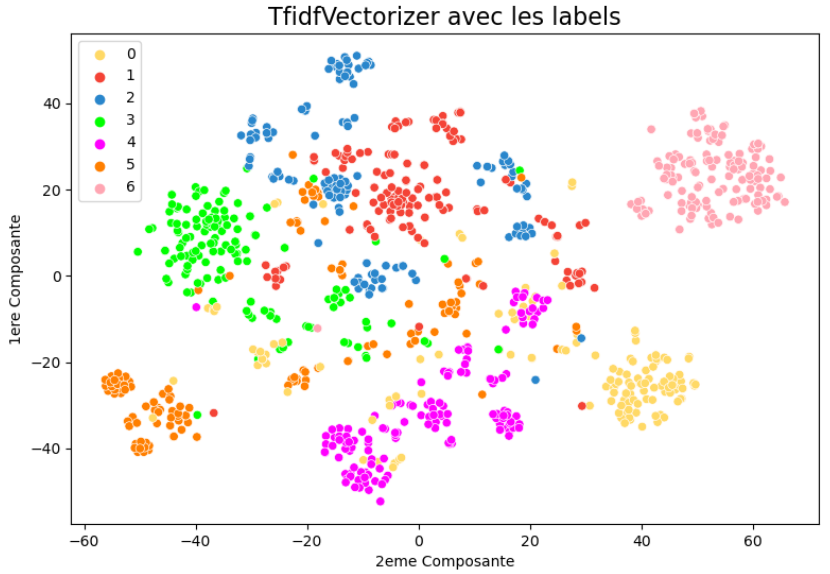
Objectif: Comparaison des différents prétraitements de données et valider la faisabilité d'une classification supervisée

- Comparaison des traitements via Bag-of-ngrams

Traitements	Fréquence	TF-Idf
BoW_wo_sw	0.354169	0.490681
BoW_wo_sw_dw	0.376539	0.226567
BoW_stem	0.342008	0.383303
BoW_stem_dw	0.366856	0.308739
BoW_lem	0.457319	0.498597
BoW_lem_dw	0.33005	0.222708

Les valeurs du tableau sont les résultats de l'ARI entre les clusters via K-means et les labels réels

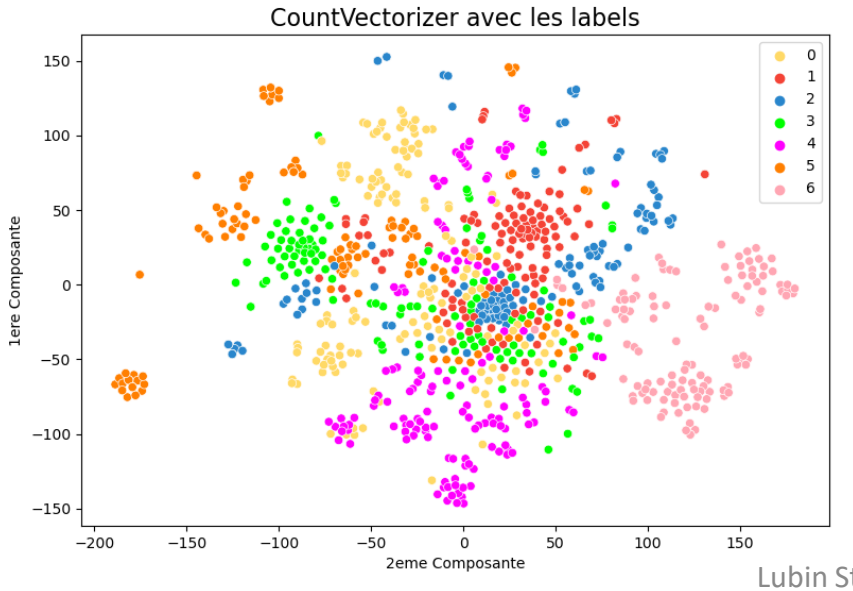


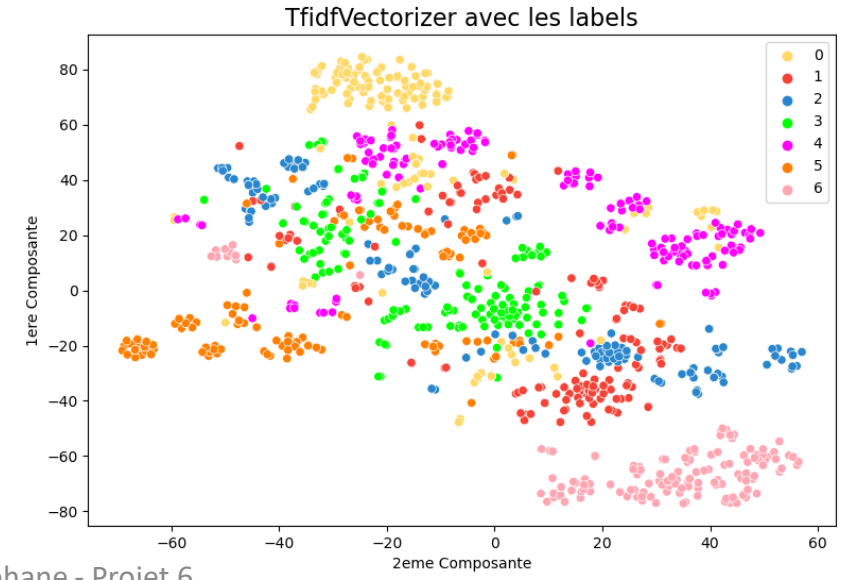


• Comparaison des traitements via Bag-of-ngrams

Traitements	Fréquence	TF-Idf
BoN_wo_sw_bi	0.199699	0.330899
BoN_wo_sw_tri	0.1689	0.29494
BoN_stem_bi	0.219618	0.336687
BoN_stem_tri	0.166546	0.352308
BoN_lem_bi	0.2673	0.353139
BoN_lem_tri	0.162889	0.335781

2 types de n-grams:
- Les bigrammes
- Les trigrammes



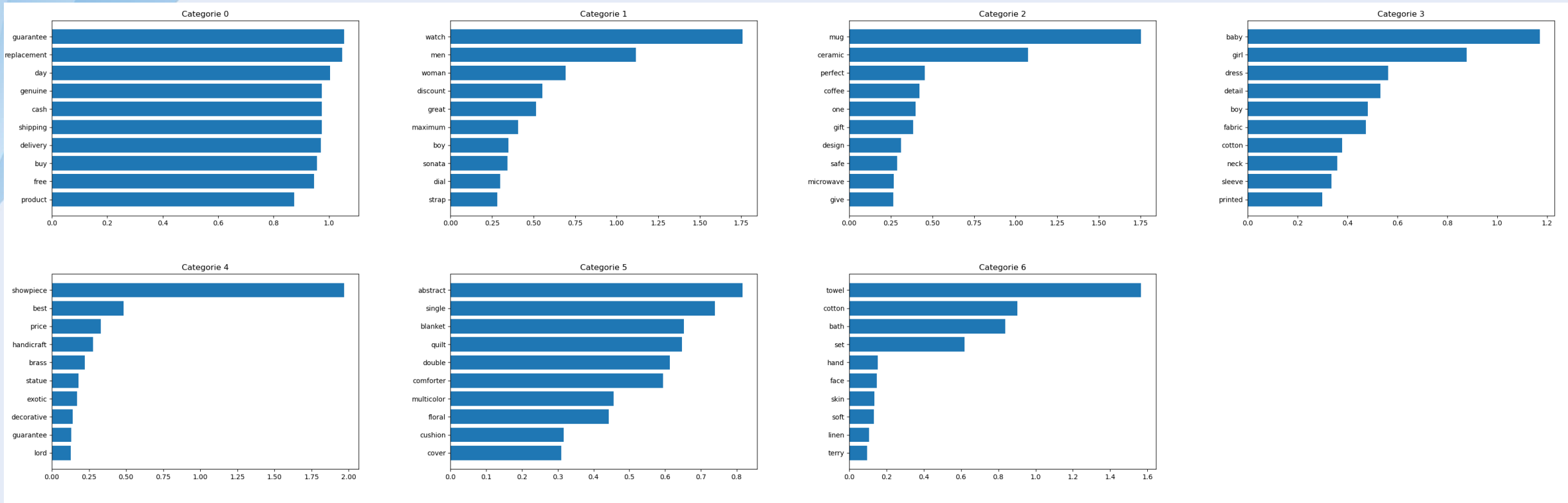


- Observations:

Le traitement le plus adapté semble être la lemmatisation sans l'élimination des dropwords, avec un approche bag-of-words, l'extraction de features via TF-idf est plus adapté à la situation.

Les résultats de l'ARI et les visuels précédents montrent qu'une classification supervisée est faisable.

Mais pour avoir une autre preuve de la faisabilité j'ai réalisé du Topic Moddeling:



• Comparaison des différentes approches

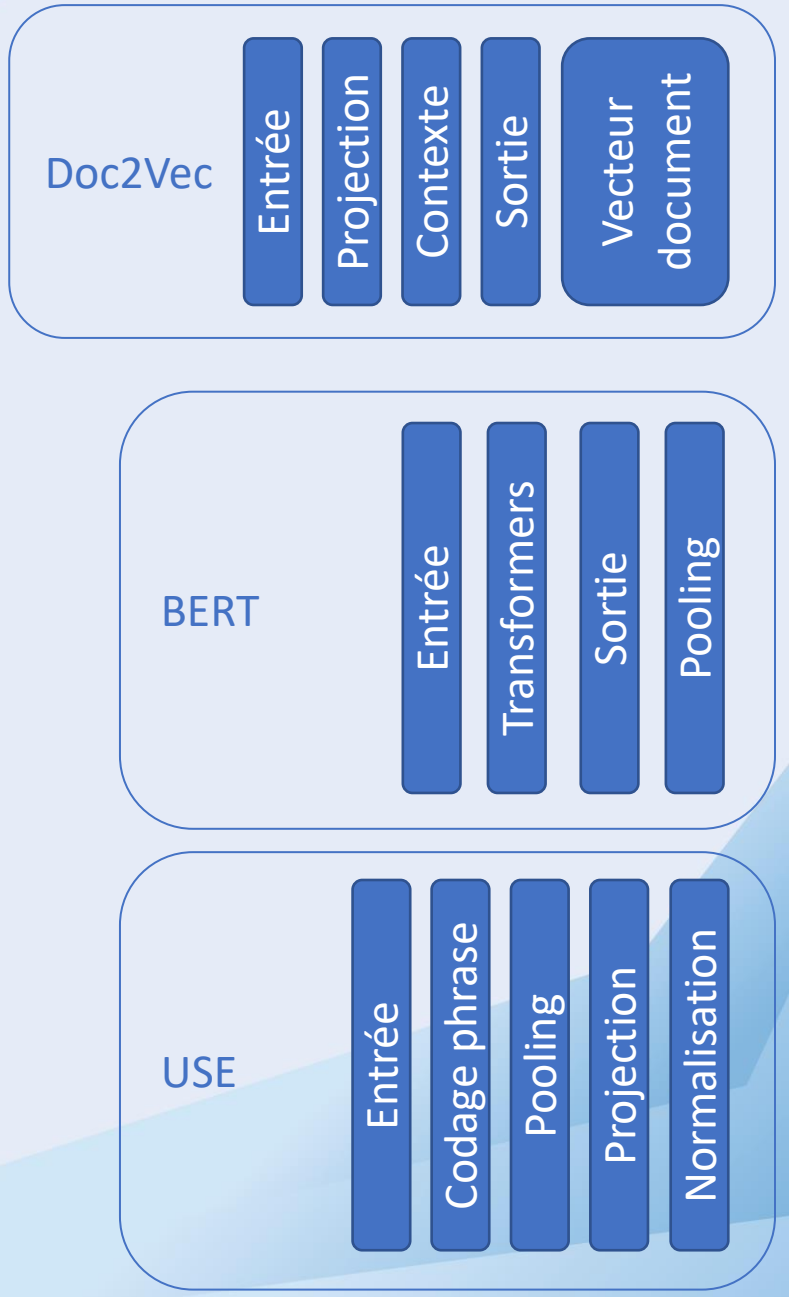
Features extraction:

- Approches Bag-of-Words (Comptage simple et TF-IDF)
- Approche word embledding Word2Vec (Doc2Vec):
  - Une couche d'entrée
  - Une couche de projection
  - Une couche de contexte
  - Une couche de sortie
  - Une couche de vecteur document
- Approche avec BERT:
  - Une couche d'entrée
  - Des couches de transformers
  - Une couche de sortie
  - Une couche de pool
- Approche avec USE :
  - Une couche d'entrée
  - Une couche de codage de phrase
  - Une couche de pooling
  - Une couche de projection
  - Une couche de normalisation

Puis classification supervisée et comparaison des précisions

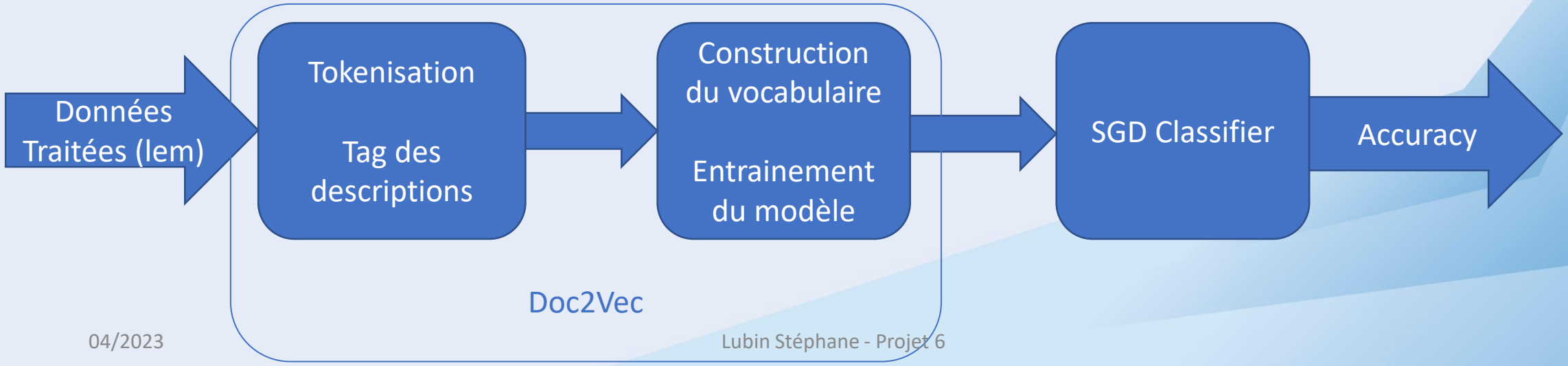
Réalisation de la matrice de confusion

Affichage du temps d'entrainement et celui du test

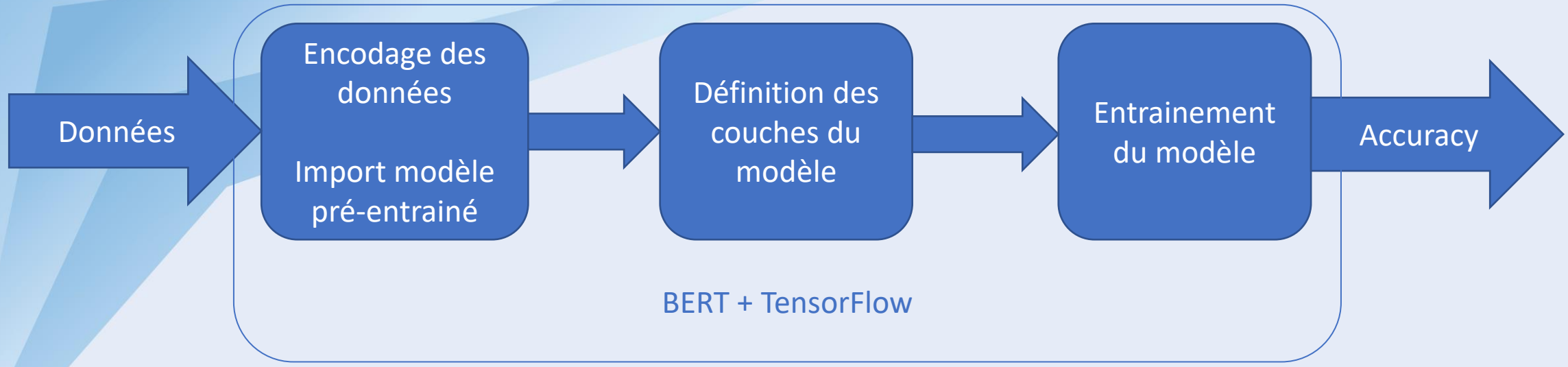


Approches bag-of-words

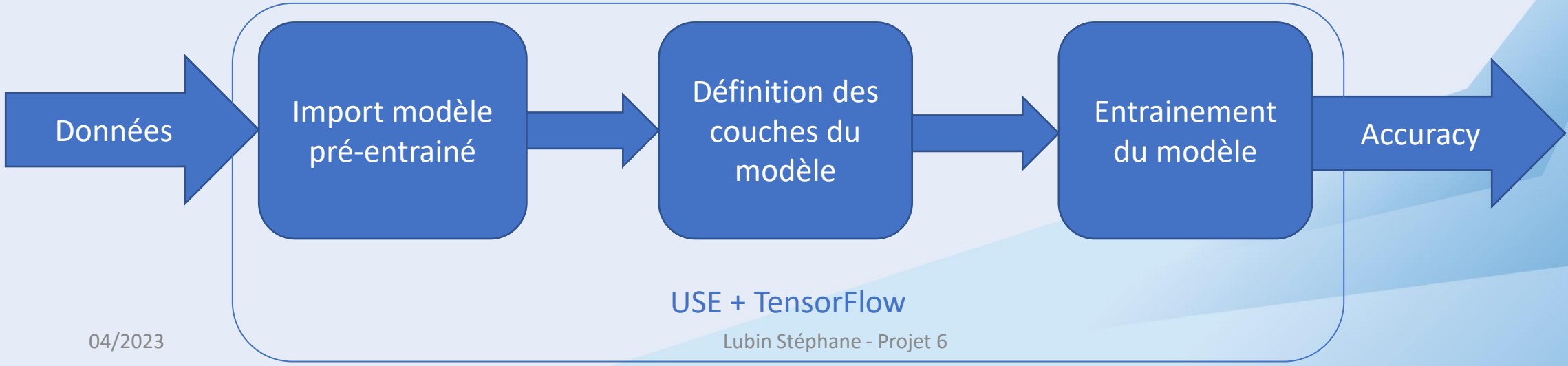
Approche word embledding Doc2Vec



Approches word embledding BERT



Approche word embledding USE

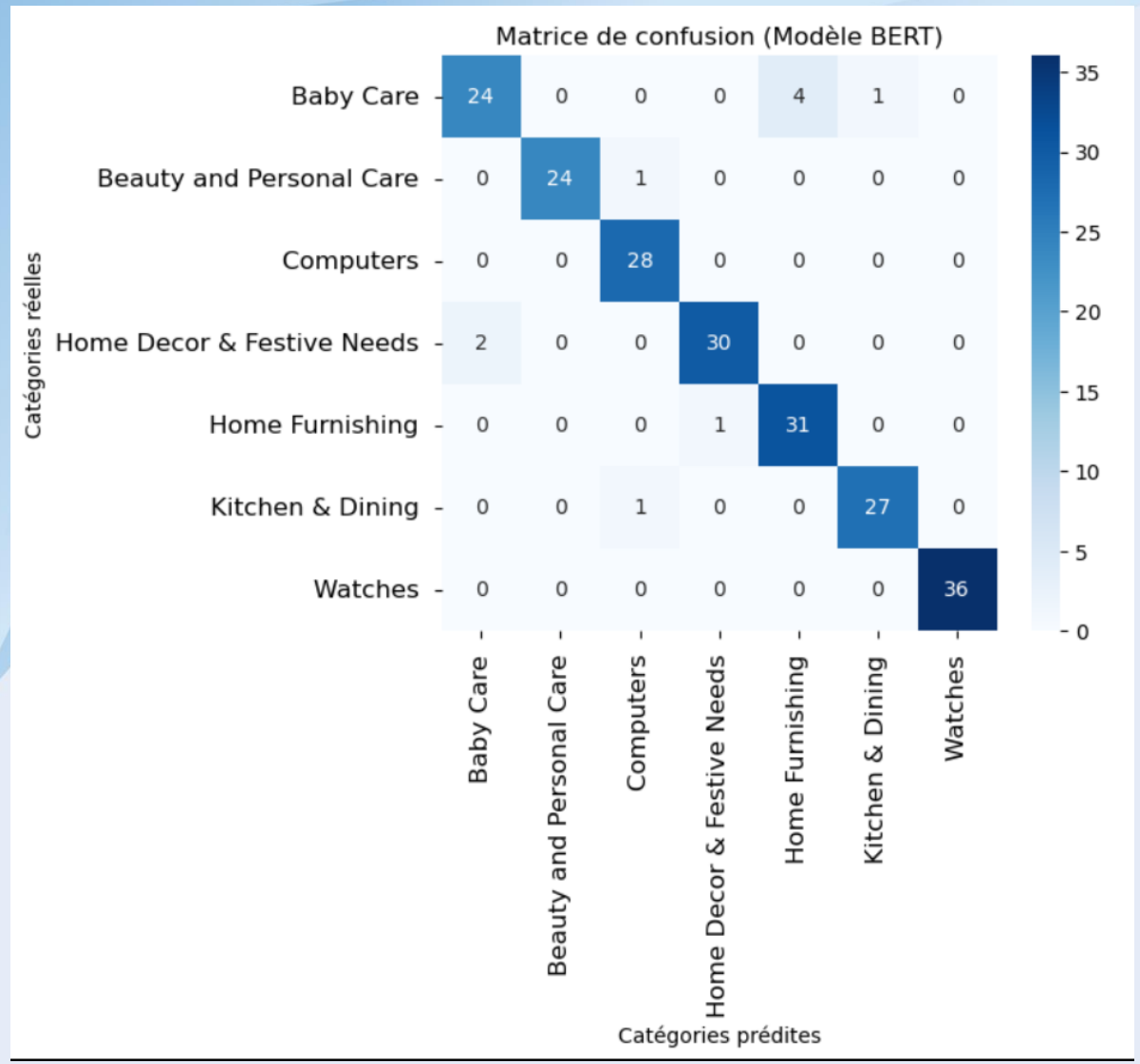


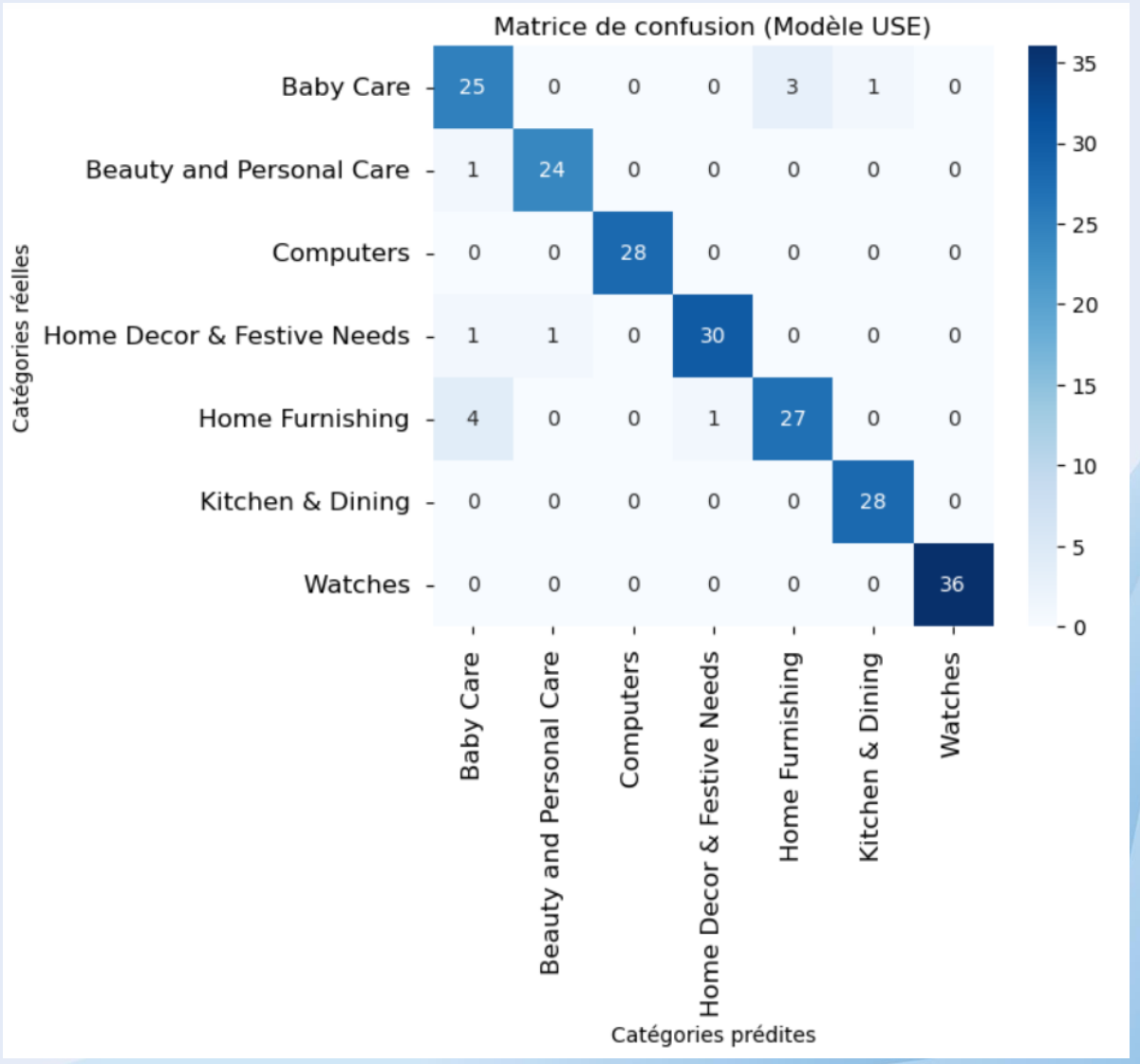
- Comparaison des modèles

	Train_Accuracy	Test_Accuracy	Train_Time (en s)	Test_Time (en s)
CountVectorizer	0.671429	0.314286	19.497707	0.022903
TfidfVectorizer	0.620238	0.114286	12.906792	0.018948
Doc2Vec	0.821429	0.666667	345.167994	1.172099
BERT	0.901786	0.952381	7961.198062	58.864084
USE	1.0	0.942857	496.365633	0.495489

Les modèles les plus performants sont BERT et USE.

Mais nécessite beaucoup plus de temps pour réaliser l'entrainement du classifier


Matrice de confusion (Modèle BERT)
Baby Care
Beauty and Personal Care
Computers
Home Decor & Festive Needs
Home Furnishing
Kitchen & Dining
Watches
24 0 0 0 4 1 0
0 24 1 0 0 0 0
0 0 28 0 0 0 0
2 0 0 30 0 0 0
0 0 0 1 31 0 0
0 0 1 0 0 27 0
0 0 0 0 0 0 36
Catégories réelles
Catégories prédites
35
30
25
20
15
10
5
0


Matrice de confusion (Modèle USE)
Baby Care
Beauty and Personal Care
Computers
Home Decor & Festive Needs
Home Furnishing
Kitchen & Dining
Watches
25 0 0 0 3 1 0
1 24 0 0 0 0 0
0 0 28 0 0 0 0
1 1 0 30 0 0 0
4 0 0 1 27 0 0
0 0 0 0 0 28 0
0 0 0 0 0 0 36
Catégories réelles
Catégories prédites
35
30
25
20
15
10
5
0

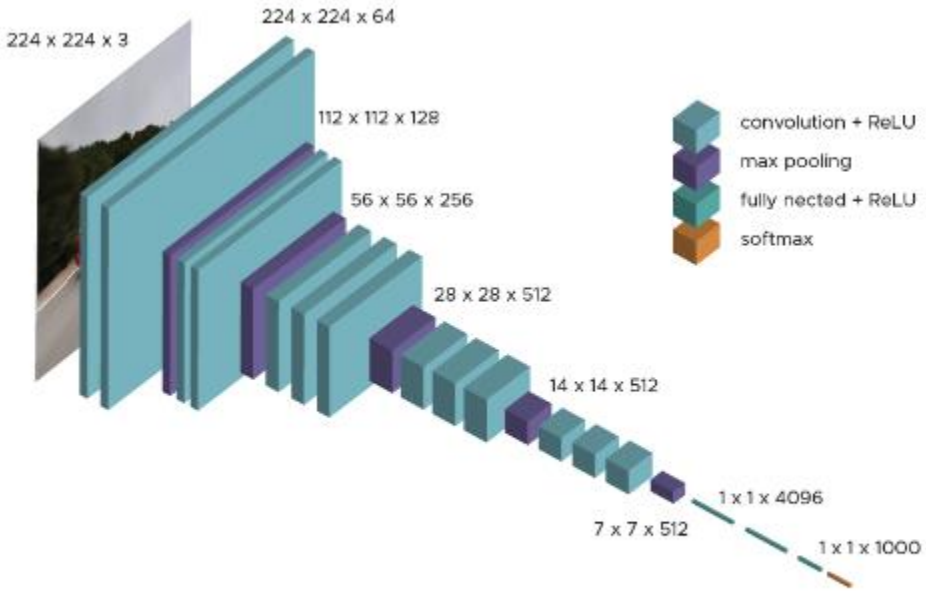
Partie Image

Méthodologie:

- Test de faisabilité
- Approche SIFT
- Approche CNN sans Data Augmentation
- Approche CNN avec Data Augmentation

- Test de faisabilité

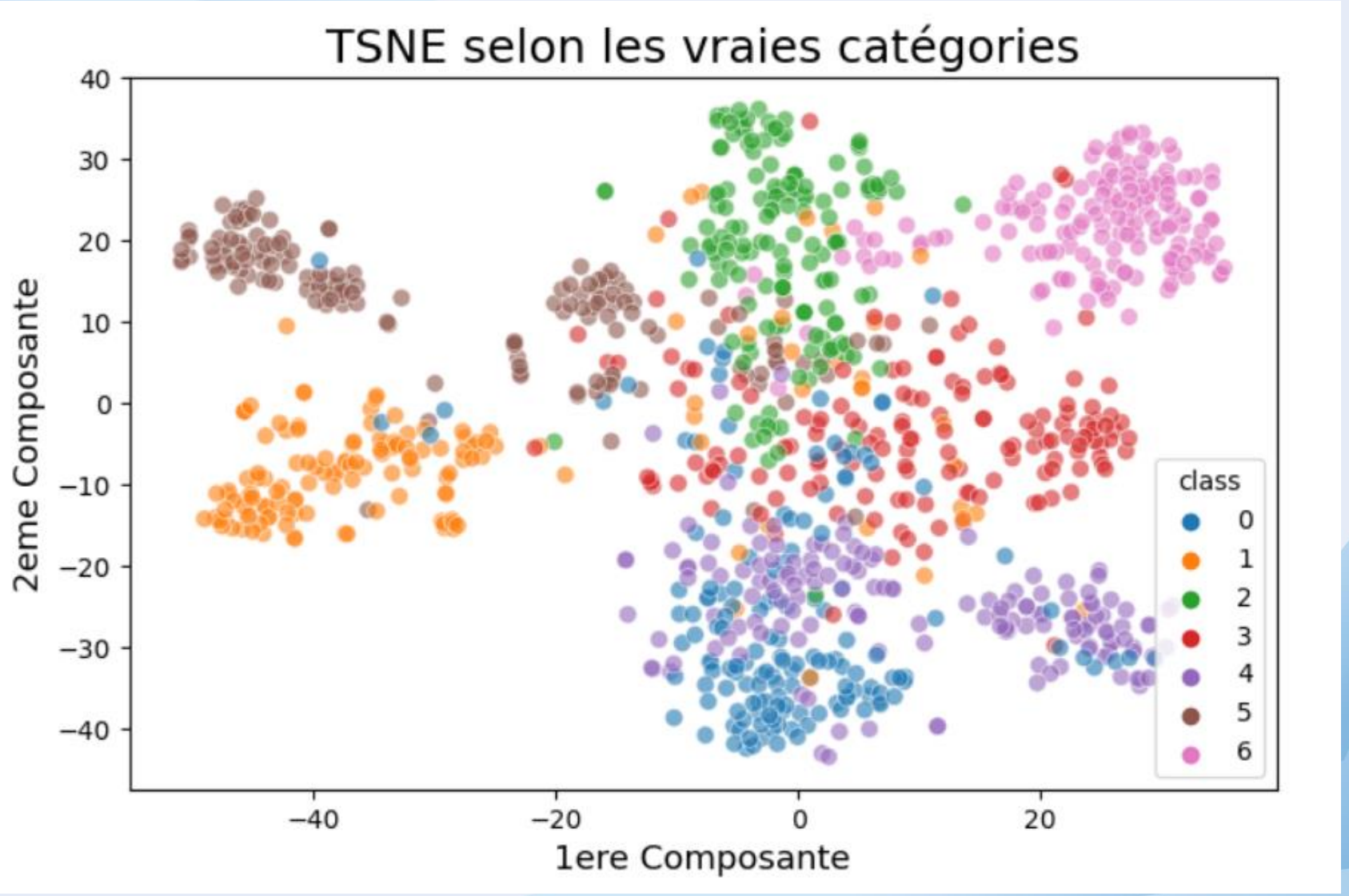
Architecture Algoritme VGG16



Structure Algoritme VGG16

- Observations:

ARI = 0,45



• Approche avec SIFT

J’ai fait varier le threshold de sift pour en observer l’effet

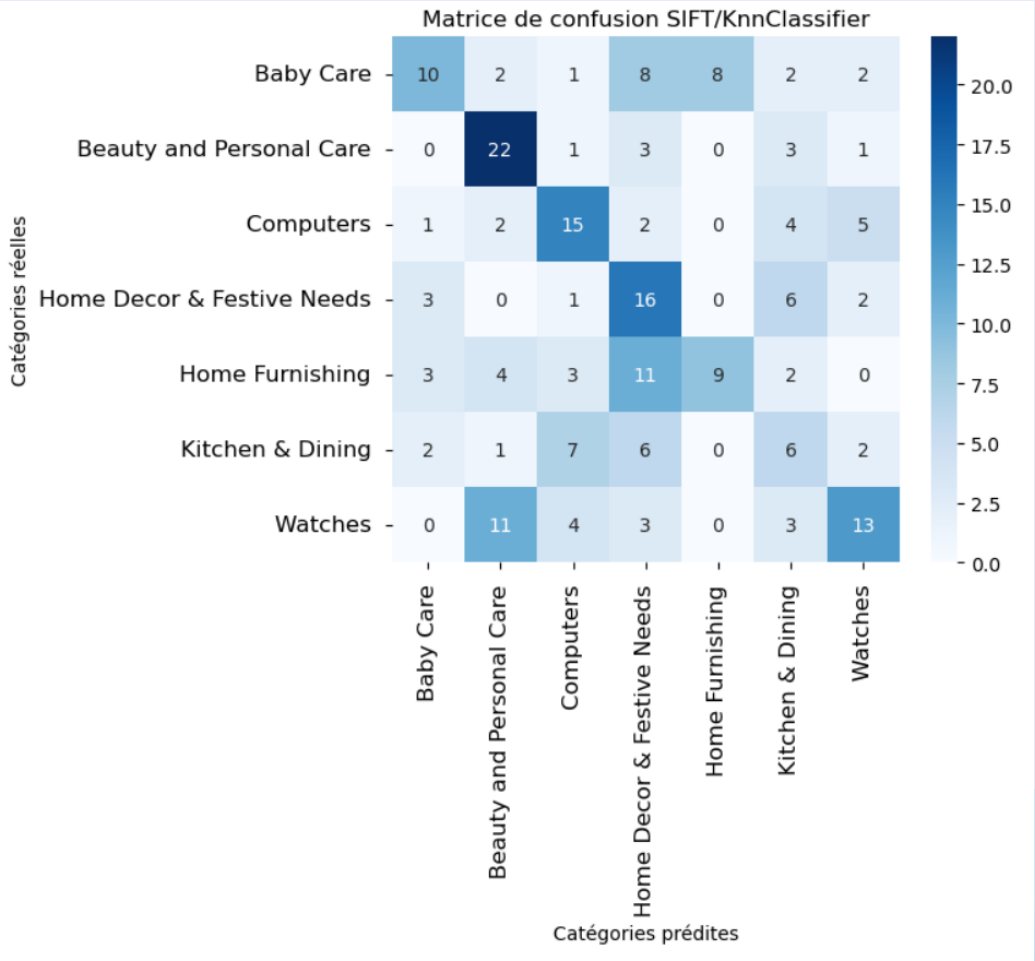
K-means clustering a été utilisé pour la conception des centroïdes du bag-of-features

3 Classifiers différents sont utilisés:

- SVM Classifier
- Logistic Regression
- Knn Classifier

- Observations:

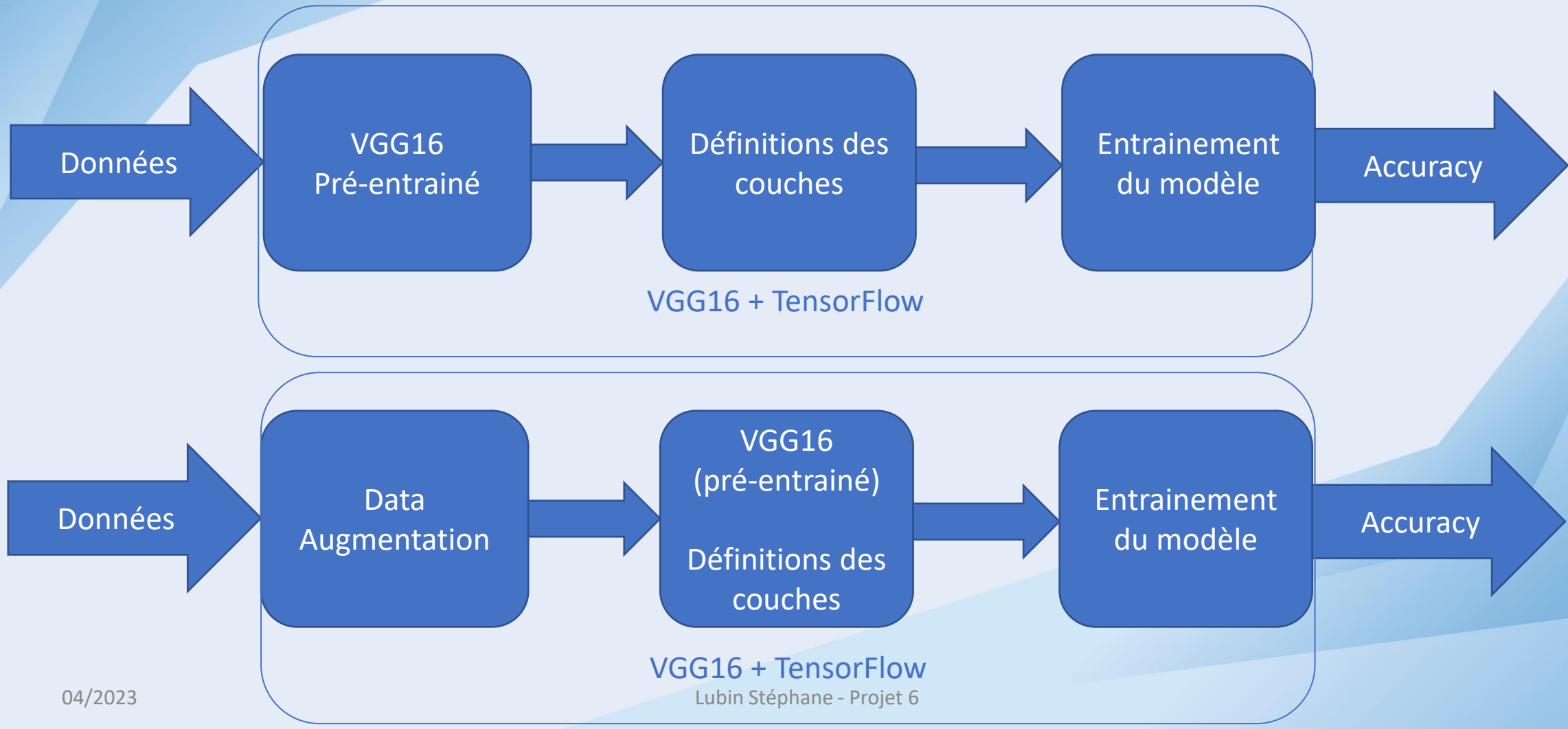
La précision qui est aux alentour de 45% (sur les données test) nous permet de dire que l'approche n'est pas la plus adapté pour notre problème de classification



- Approche CNN

Un modèle sans data augmentation (avec un batch size de 32)

Puis 3 modèles avec data augmentation (Flip, Rotation et Zoom) , en faisant varier le batch size de 1, 32 et 64



- Observations

	Train_accuracy	Validation_accuracy	Test_accuracy
BatchSize_32	0.915873	0.766667	0.780952
BatchSize_64	0.912698	0.766667	0.771429
BatchSize_1	0.901587	0.766667	0.77619
Sans Data Augmentation	0.953968	0.77619	0.766667

La Data Augmentation remplit bien son rôle puisque le modèle est plus performant sur le jeu test

Le batch size le plus adapté est 32, c'est avec celui-ci que l'on obtient les meilleurs résultats

API

Pour élargir la gamme des produits proposées, un test de collecte de produits à base de champagne via une API (edamam-food-and-grocery-database) a été réalisé.

Seulement 3 produits répondent à tous les critères

Conclusion

- L’utilisation de TensorFlow pour faire du Transfer Learning à faciliter grandement le travail effectué.
- TensorFlow est une méthode récente qui prouve que les approches qui donnent les meilleurs résultats ne sont pas dépassé.

MERCI DE VOTRE ATTENTION