**Note méthodologique : preuve de concept**

Table des matières

[Contexte 2](#_Toc169270235)

[1. Dataset retenu 3](#_Toc169270236)

[2. Les concepts de l’algorithme récent 4](#_Toc169270237)

[2.1. Introduction / Améliorations de DeBERTa 4](#_Toc169270238)

[2.2. Fonctionnement de DeBERTa 5](#_Toc169270239)

[2.3. Conclusion 5](#_Toc169270240)

[3. La modélisation 6](#_Toc169270241)

[3.1 Choix des algorithmes de classification 6](#_Toc169270242)

[3.2 Représentation des données textuelles 6](#_Toc169270243)

[3.3 Évaluation et optimisation des modèles 7](#_Toc169270244)

[3.4. Conclusion Intermédiaire 7](#_Toc169270245)

[4. Une synthèse des résultats 8](#_Toc169270246)

[4.1. ARI VS Methods Vs Classifiers 8](#_Toc169270247)

[4.2. Accuracy VS Methods Vs Classifiers 8](#_Toc169270248)

[5. L’analyse de la feature importance globale et locale du nouveau modèle 10](#_Toc169270249)

[6. Les limites et les améliorations possibles 11](#_Toc169270250)

# Contexte

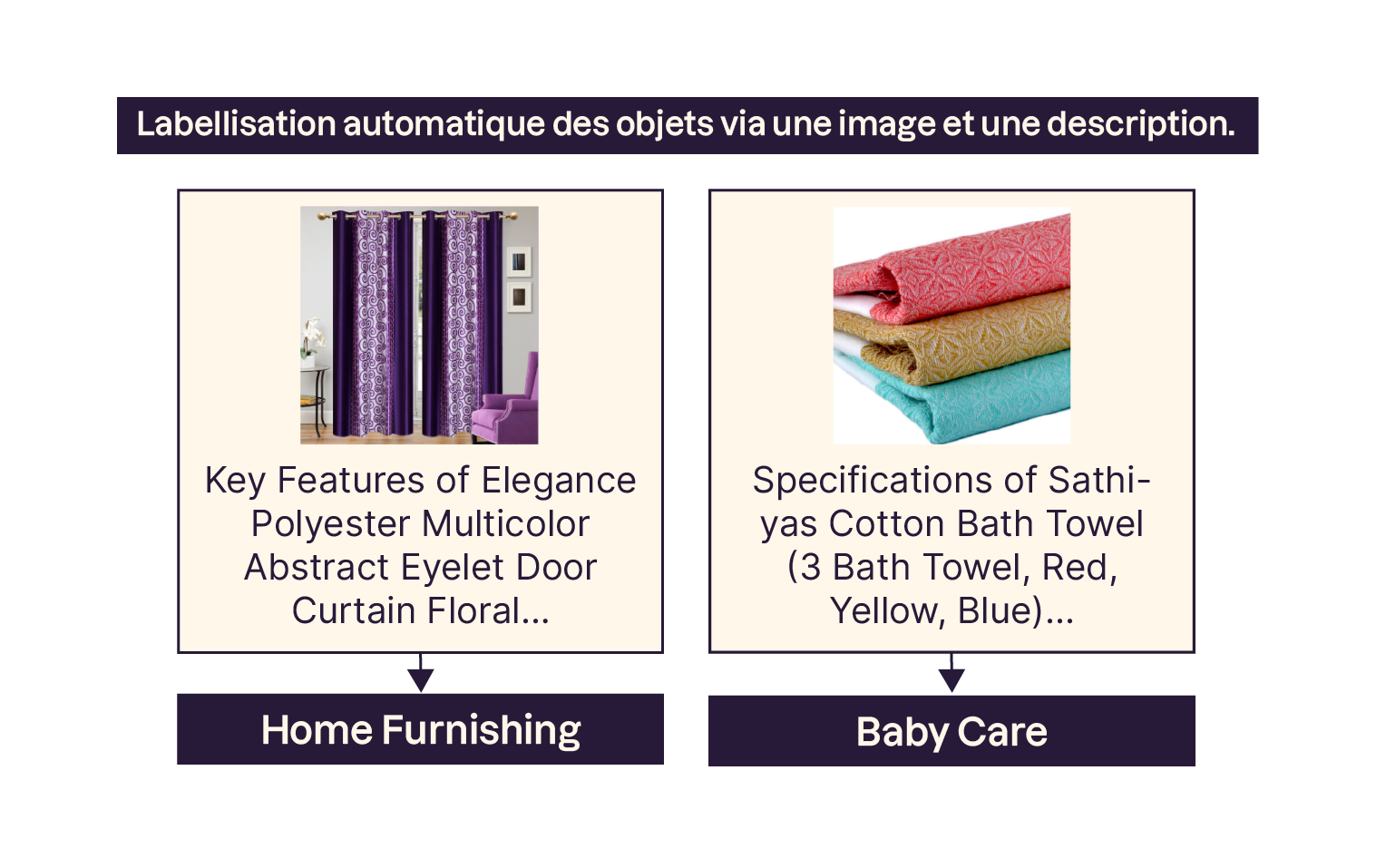
Sur la place de marché anglophone « **Place de marché** », des vendeurs proposent des articles à des acheteurs en postant une photo et une description.

Pour l'instant, l'attribution de la catégorie d'un article est effectuée manuellement par les vendeurs, et est donc peu fiable. De plus, le volume des articles est pour l’instant très petit.

Pour rendre l’expérience utilisateur des vendeurs (faciliter la mise en ligne de nouveaux articles) et des acheteurs (faciliter la recherche de produits) la plus fluide possible, et dans l'optique d'un passage à l’échelle, il devient nécessaire d'automatiser cette tâche d‘attribution de la catégorie.

**Problématique** : Catégorisation peu fiable ET augmentation du volume des articles

**Proposition** : Automatiser la tâche d’attribution de la catégorie via du texte



# **Une image contenant texte, capture d’écran, Police Description générée automatiquement**1. Dataset retenu

Dataset complet sur des produits d’une plateforme de e-commerce : Place de marché.   
Trois indicateurs nous intéressent particulièrement :

* uniq\_id : L’identifiant unique d’un produit
* product\_category\_tree : L’arbre des catégories d’un produit (Top > secondaire > tertiaire)
* description : La description d’un produit.

**Une image contenant texte, diagramme, cercle, capture d’écran

Description générée automatiquement**Pour les Top catégories : 7 classes équilibrées

# 2. Les concepts de l’algorithme récent (6 oct 2021)

**Url de l’article et du code :**

* <https://arxiv.org/pdf/2006.03654>
* <https://github.com/huggingface/transformers/blob/main/src/transformers/models/deberta/modeling_deberta.py>

## 2.1. Introduction / Améliorations de DeBERTa

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) est un modèle de langage basé sur Transformer développé par Microsoft Research. Il s'agit d'une variante de BERT (Bidirectional Encoder Representations from Transformers) qui propose plusieurs améliorations, notamment :

* **Attention désenchevêtrée :**   
  DeBERTa utilise deux types d'attention distincts : une attention locale pour capturer les dépendances à courte portée et une attention globale pour capturer les dépendances à longue portée. Cela permet d'améliorer la capacité du modèle à comprendre le contexte global d'une phrase.  
    
  Chaque mot est représenté par deux vecteurs distincts : {Hi} et {Pi|j} qui encodent respectivement son contenu et sa position relative par rapport au mot à la position j.  
  Le calcul du score d'attention croisée entre les tokens i et j peut être décomposé en quatre composantes :

Une image contenant texte, Police, typographie, blanc

Description générée automatiquement

Ainsi, le poids d'attention d'une paire de mots peut être calculé comme la somme de quatre scores d'attention en utilisant des matrices désenchevêtrées sur leurs contenus et positions comme contenu-contenu, contenu-position, position-contenu et position-position.

* **Décodeur de masque amélioré :**   
  DeBERTa utilise un décodeur de masque amélioré qui prend en compte les sorties des couches d'attention précédentes lors du calcul des représentations des mots. Cela permet d'améliorer la capacité du modèle à générer des séquences de mots cohérentes.

Ces améliorations permettent à DeBERTa d'obtenir des résultats de pointe sur une variété de tâches de traitement du langage naturel (NLP), y compris la classification de texte.

## 2.2. Fonctionnement de DeBERTa

Le fonctionnement de DeBERTa est similaire aux autres modèles de langage basés sur Transformer. Il prend en entrée une séquence de mots, les tokenise et les indexe pour obtenir des représentations vectorielles. Ces représentations sont ensuite utilisées dans l'encodeur DeBERTa avec une attention désenchevêtrée pour générer des représentations contextuelles, puis passées à un classificateur linéaire pour la classification du texte Le processus de classification de texte avec DeBERTa peut être résumé comme suit :

Le prétraitement du texte consiste à découper le texte en mots par tokenisation, puis à convertir chaque mot en un identifiant numérique unique par indexation. Ensuite, les représentations vectorielles des mots sont obtenues en utilisant les index pour rechercher les vecteurs correspondants dans le vocabulaire du modèle, capturant ainsi le sens et les relations sémantiques entre les mots. Ces représentations vectorielles sont ensuite passées dans l'encodeur DeBERTa, qui utilise une attention désenchevêtrée pour générer des représentations contextuelles en tenant compte du contexte des mots environnants. Enfin, la représentation vectorielle finale du texte est obtenue en agrégeant les représentations contextuelles de tous les mots, puis transmise à un classificateur linéaire pour prédire la classe du texte, par exemple, positif/négatif, pertinent/non pertinent

## **2.3. Conclusion**

Les améliorations apportées par l'attention désenchevêtrée et le décodeur de masque amélioré permettent à DeBERTa d'obtenir des résultats de pointe sur une large gamme de tâches de traitement du langage naturel (NLP), notamment :

* **Compréhension du langage naturel (NLU) :** Question-réponse, résumé automatique, reconnaissance d'intention, etc.
* **Génération de langage naturel (NLG) :** Résumé automatique, génération de texte, traduction automatique, etc.

DeBERTa surpasse souvent les modèles de pointe tels que RoBERTa-Large sur divers benchmarks NLP et atteint des performances de pointe sur le benchmark SuperGLUE.

**En résumé, DeBERTa se distingue par son architecture Transformer optimisée avec une attention désenchevêtrée et un décodeur de masque amélioré, ce qui lui permet d'obtenir des performances de pointe dans une large gamme de tâches NLP.**

# 3. La modélisation

## 3.1 Choix des algorithmes de classification

Dans cette étude, plusieurs algorithmes de classification populaires ont été choisis pour la tâche de classification de texte. Ces algorithmes incluent :

* **Forêts aléatoires (Random Forest) :** Un algorithme robuste et efficace qui combine plusieurs arbres de décision pour améliorer la précision.
* **Arbres de décision (Decision Tree) :** Des modèles simples et intuitifs qui permettent d'interpréter les règles de décision.
* **Régression logistique (Logistic Regression) :** Un modèle probabiliste adapté aux problèmes de classification binaire.
* **K plus proches voisins (KNN) :** Un algorithme basé sur la similarité entre les points de données pour prédire la classe d'un nouveau point.
* **SVM linéaire (Support Vector Machine - Linear) :** Un algorithme qui maximise la marge entre les classes pour trouver un hyperplan optimal.
* **SVM RBF (Support Vector Machine - Radial Basis Function) :** Une variante du SVM linéaire qui utilise une fonction de similarité non linéaire pour mieux capturer les relations complexes dans les données.

Le choix de ces algorithmes permet de couvrir une large gamme d'approches et de complexités, offrant ainsi un aperçu complet des performances de la classification de texte dans notre contexte.

## 3.2 Représentation des données textuelles

Différentes techniques de représentation de texte ont été utilisées pour transformer le texte brut des descriptions de produits en données numériques utilisables par les algorithmes de classification. Ces techniques incluent :

* **TF-IDF (Term Frequency - Inverse Document Frequency) :** Une méthode qui pondère chaque mot en fonction de sa fréquence dans un document et son importance relative dans l'ensemble du corpus.
* **BOW (Bag of Words) :** Une méthode simple qui représente chaque document par un vecteur binaire indiquant la présence ou l'absence de chaque mot du vocabulaire.
* **Bag of N-grams :** Des séquences de N mots consécutifs extraites du texte, permettant de capturer des informations locales sur l'ordre des mots.
* **Word2Vec (Skip-grams et CBOW) :** Des modèles de réseaux neuronaux qui apprennent à représenter les mots en fonction de leur contexte, permettant d'obtenir des vecteurs de mots distribués sémantiquement.
* **BERT (Bidirectional Encoder Representations from Transformers) :** Un modèle de langage pré-entraîné puissant qui produit des vecteurs de mots riches en informations contextuelles.
* **USE (Universal Sentence Encoder) :** Un modèle de codage de phrases basé sur BERT, permettant de convertir des phrases en vecteurs de représentation sémantique.
* **DEBERTA :** Un modèle de langage pré-entraîné basé sur RoBERTa, avec une architecture d'attention améliorée pour une meilleure compréhension du contexte.

L'utilisation de ces différentes techniques permet d'explorer les avantages et les limites de chacune d'entre elles pour la tâche de classification de produits.

## 3.3 Évaluation des modèles

Les performances des différents modèles et techniques ont été évaluées à l'aide des métriques de l'ARI (Adjusted Rand Index) et de la précision.

* **L'ARI est une métrique** **adaptée à la comparaison de partitions de données multiclasses**, comme c'est le cas ici avec la classification en 7 catégories de produits. Il prend en compte à la fois les paires de points correctement classés et les paires de points incorrectement classés, en tenant compte de la taille des classes.
* **La précision** est une métrique simple et intuitive qui **mesure la proportion de prédictions correctes** parmi toutes les prédictions effectuées. Elle est particulièrement utile pour **évaluer la capacité du modèle à identifier correctement les classes majoritaires**.

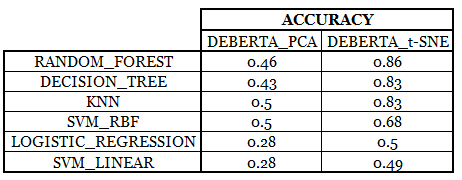
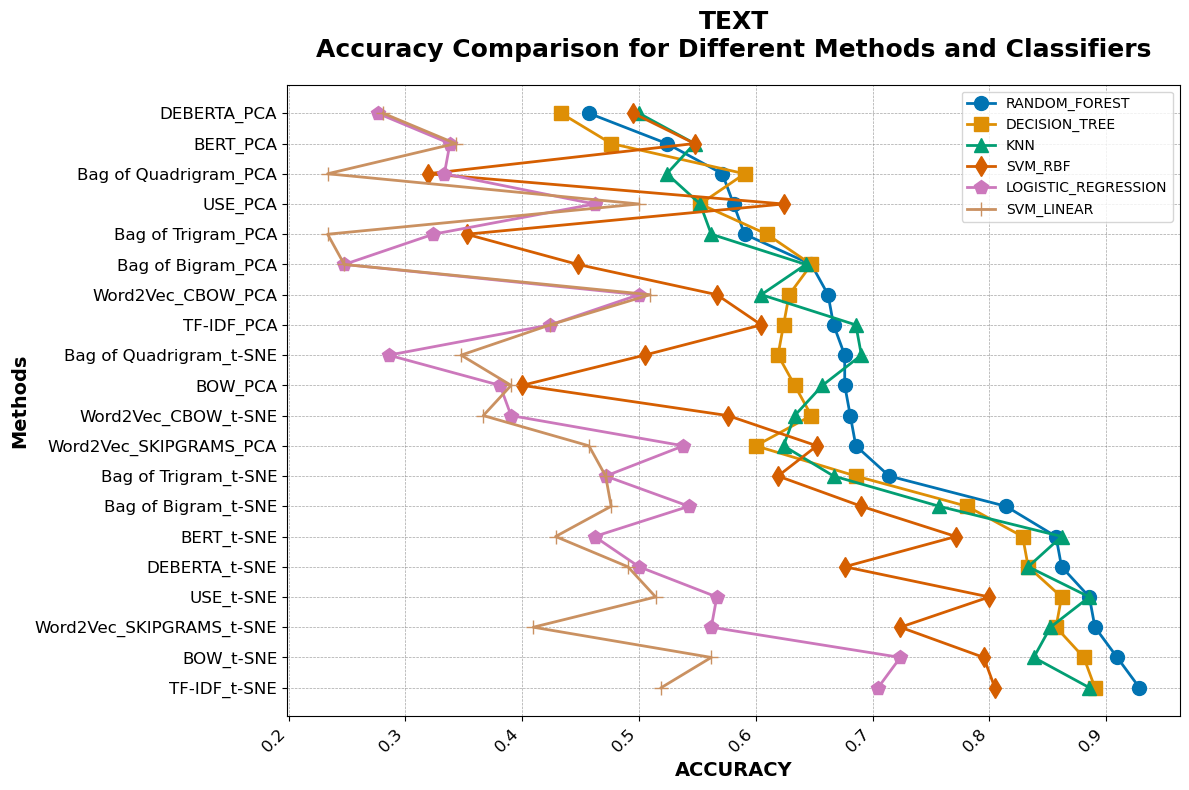
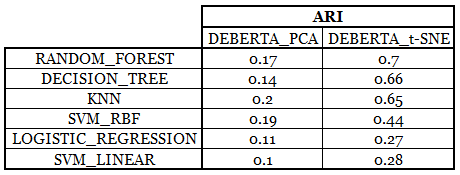
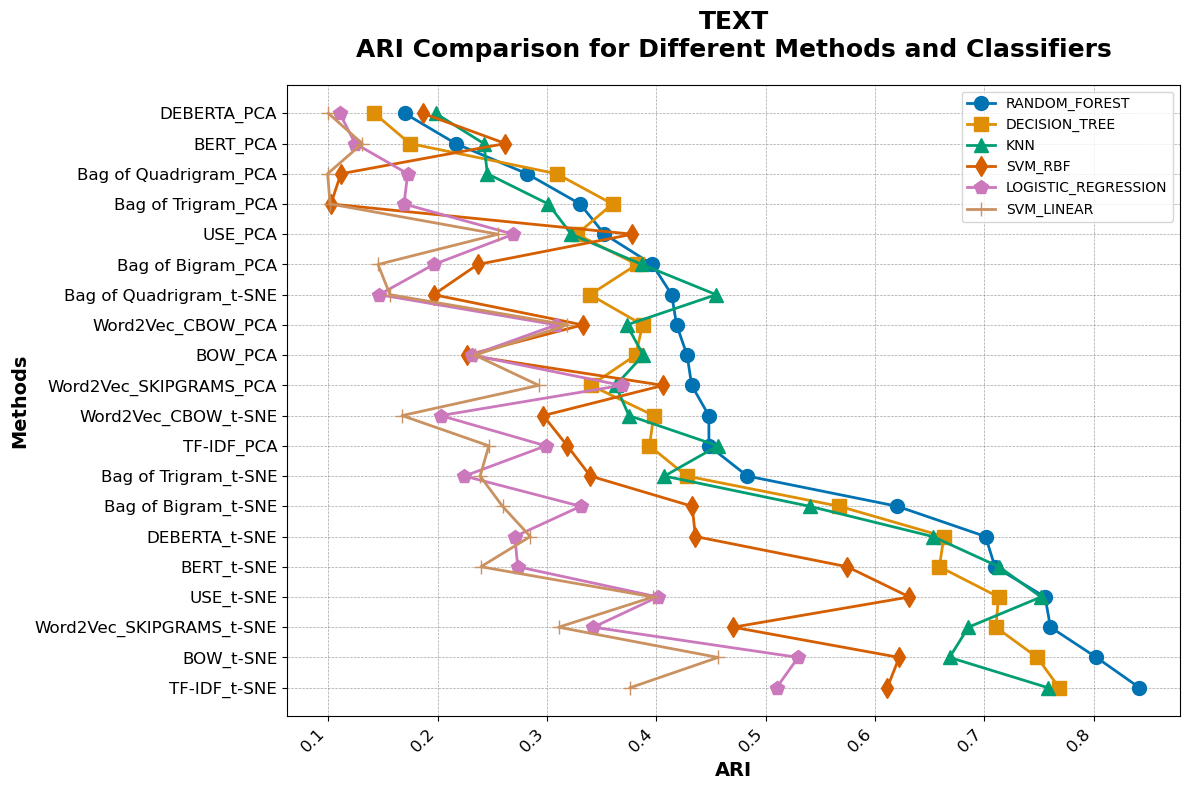
## 3.4. Conclusion Intermédiaire

Ce chapitre a exploré diverses approches pour classifier des descriptions de produits en 7 catégories. On a comparé des algorithmes (Forêts aléatoires, SVM, BERT, DEBERTA) et des techniques de représentation de texte (TF-IDF, BOW, Word2Vec). L'évaluation (ARI et précision) a considéré la distribution des classes. Ces résultats guident le choix des modèles et techniques pour la classification de texte similaire.

# 4. Une synthèse des résultats

## 4.1. ARI VS Methods Vs Classifiers

## 4.2. Accuracy VS Methods Vs Classifiers



# 5. L’analyse de la feature importance globale et locale du nouveau modèle

*Présentez l’analyse de la feature importance globale et locale du nouveau modèle, en 2 pages maximum.*

# 6. Les limites et les améliorations possibles

*Présentez les limites et les améliorations envisageables pour gagner en performance et en interprétabilité de l'approche de modélisation, en 1 page maximum.*

*Mer 13h30*