

Note méthodologique : preuve de concept

Dataset retenu	1
1.Source et Nature des Données	1
1.1 Taille du Dataset	1
1.2 Classes cibles	2
1.3 Prétraitement	3
Les Concepts de l'Algorithme Récent (DeBERTa)	4
1. Introduction à DeBERTa	4
2. Principes de Fonctionnement de DeBERTa	4
3. Avantages Théoriques et Pratiques	6
La Modélisation	7
1. Méthodologie de Modélisation	7
1.1. Préparation des Données	7
1.2. Extraction des Caractéristiques	7
1.3. Métrique d'Évaluation	8
1.4. Démarche d'Optimisation	8
Synthèse des Résultats	10
1. Comparaison des Résultats Entre BERT et DeBERTa	10
1.1. Résultats de BERT	10
1.2. Résultats de DeBERTa (Modèle de Base - deberta-base)	10
1.3. Résultats de DeBERTa (Modèle Plus Grand - deberta-large)	11
2. Analyse des Résultats	12
Analyse de la Feature Importance Globale et Locale du Modèle DeBERTa	13
1. Feature Importance Globale	13
2. Visualisation de la Valeur de Base du Modèle (Baseline Value)	14
3. Feature Importance Locale	14
Les Limites et Améliorations Possibles	16
1. Limites de l'Approche Actuelle	16
2. Améliorations Envisageables	16

Dataset retenu

1. Source et Nature des Données

Le dataset utilisé dans cette étude provient d'une plateforme de e-commerce anglophone où des vendeurs proposent des articles à des acheteurs en postant une photo et une description textuelle. Actuellement, l'attribution des catégories aux articles est effectuée manuellement par les vendeurs, ce qui introduit des incohérences et des erreurs en raison de la subjectivité humaine. De plus, le volume actuel des articles est encore relativement faible. Pour améliorer l'expérience utilisateur et préparer un passage à l'échelle, il devient crucial d'automatiser cette tâche de classification des articles dans les différentes catégories.

Labellisation automatique des objets via une image et une description.



1.1 Taille du Dataset

Le dataset comprend 1 050 enregistrements et 15 colonnes, comme le montre l'extrait ci-dessous :

- **uniq_id** : Identifiant unique de chaque produit.
- **product_name** : Nom du produit.
- **product_category_tree** : Arbre des catégories auquel appartient le produit.
- **description** : Description textuelle du produit, principale source d'information pour la classification.
- **retail_price et discounted_price** : Prix du produit, avant et après réduction.
- **image** : Lien vers l'image associée au produit.
- **other features** : D'autres caractéristiques comme le rating, la marque, etc.

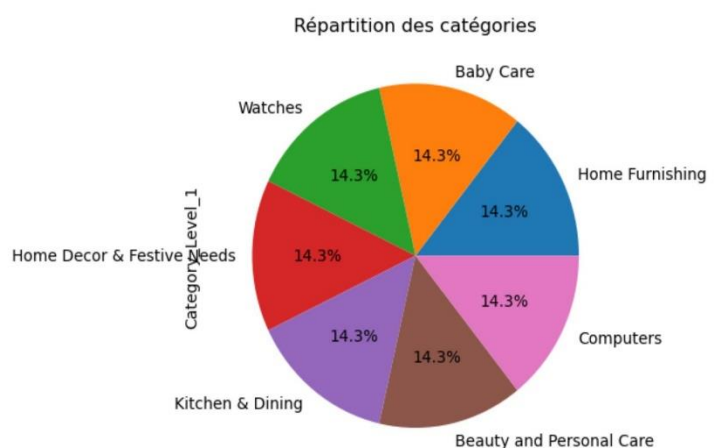
uniq_id	55b85ea15a1536d46b7190ad6fff8ce7	7b72c92c2f6c40268628ec5f14c6d590	64d5d4a258243731dc7bbb1eef49ad74	d4684dc759c
crawl_timestamp	2016-04-30 03:22:56 +0000	2016-04-30 03:22:56 +0000	2016-04-30 03:22:56 +0000	2016-04-30 03:22:56 +0000
product_url	http://www.flipkart.com/elegance-polyester-multicolor-abstract-eyelet-curtains-p/55b85ea15a1536d46b7190ad6fff8ce7	http://www.flipkart.com/sathiyas-cotton-bath-towel/p/7b72c92c2f6c40268628ec5f14c6d590	http://www.flipkart.com/eurospa-cotton-terry-face-towel-set/p/64d5d4a258243731dc7bbb1eef49ad74	http://www.flipkart.com/santos-pajamas/p/d4684dc759c
product_name	Elegance Polyester Multicolor Abstract Eyelet ...	Sathiyas Cotton Bath Towel	Eurospa Cotton Terry Face Towel Set	SANTOS
product_category_tree	["Home Furnishing >> Curtains & Accessories >>...]	["Baby Care >> Baby Bath & Skin >> Baby Bath T...]	["Baby Care >> Baby Bath & Skin >> Baby Bath T...]	["Home I...]
pid	CRNEG7BKMFFYHQ8Z	BTWEGFZHGBXPZUH	BTWEG6SHXTDB2A2Y	
retail_price	1899.0	600.0	NaN	
discounted_price	899.0	449.0	NaN	
image	55b85ea15a1536d46b7190ad6fff8ce7.jpg	7b72c92c2f6c40268628ec5f14c6d590.jpg	64d5d4a258243731dc7bbb1eef49ad74.jpg	d4684dc759c
is_FK_Advantage_product	False	False	False	
description	Key Features of Elegance Polyester Multicolor ...	Specifications of Sathiyas Cotton Bath Towel (...)	Key Features of Eurospa Cotton Terry Face Towe...	Key Fea...
product_rating	No rating available	No rating available	No rating available	
overall_rating	No rating available	No rating available	No rating available	
brand	Elegance	Sathiyas	Eurospa	S...
product_specifications	{"product_specification"=> [{"key"=>"Brand", "v...	{"product_specification"=> [{"key"=>"Machine Wa...	{"product_specification"=> [{"key"=>"Material", ...	

1.2 Classes cibles

Les classes cibles pour cette étude sont les catégories de produits, représentées par la colonne `product_category_tree`. Le dataset comprend 7 classes équilibrées, ce qui rend l'analyse plus robuste et fiable. Ces catégories représentent différents niveaux de l'arbre des catégories d'un produit sur la plateforme.

Les catégories principales sont :

- **Home Furnishing**
- **Baby Care**
- **Watches**
- **Home Decor & Festive Needs**
- **Kitchen & Dining**
- **Beauty and Personal Care**
- **Computers**



1.3 Prétraitement

Le prétraitement des données a consisté en plusieurs étapes :

- **Suppression des colonnes non pertinentes** : Les colonnes qui n'étaient pas nécessaires pour la classification des produits ont été supprimées du dataset.
- **Tokenisation et mise en minuscules (Lower)** : Les descriptions textuelles des produits ont été tokenisées, et tous les tokens ont été convertis en minuscules pour assurer la cohérence.
- **Suppression des stop words** : Les mots communs et non informatifs (stop words) ont été supprimés des textes pour réduire le bruit dans les données.
- **Stemming et lemmatisation** : Les mots ont été réduits à leur racine (stemming) et à leur forme canonique (lemmatisation) pour un meilleur alignement sémantique.
- **Encodage des catégories** : Les catégories de produits ont été encodées pour être utilisées comme étiquettes dans les modèles de classification.
- **Extraction et description des caractéristiques** :
 - **Bag of Words** : Techniques comme CountVectorizer et TF-IDF ont été utilisées pour représenter les textes en vecteurs numériques.
 - **Word Embeddings** : Des méthodes plus avancées comme Word2Vec, BERT, et Universal Sentence Encoder (USE) ont été employées pour capturer les relations sémantiques entre les mots.
- **Réduction de dimension** : Des techniques comme TruncatedSVD, t-SNE, et ACP (Analyse en Composantes Principales) ont été utilisées pour réduire la dimensionnalité des vecteurs tout en conservant l'information la plus pertinente.
- **Clustering** : Un algorithme de classification non supervisée, KMeans, a été utilisé pour créer des clusters à partir des vecteurs réduits.
- **Visualisation** : Les clusters ont été visualisés à l'aide de t-SNE pour mieux comprendre la répartition des données dans l'espace des caractéristiques réduites.
- **Évaluation** : L'efficacité du clustering a été mesurée en utilisant l'Indice de Rand Ajusté (ARI), une métrique robuste pour évaluer la qualité des clusters par rapport aux classes réelles.

Les Concepts de l'Algorithme Récent (DeBERTa)

1. Introduction à DeBERTa

Origine et Contexte :

DeBERTa, qui signifie *Decoding-enhanced BERT with Disentangled Attention*, a été développé par l'équipe de Microsoft Research, dirigée par Heung-Yeung Shum, et publié en 2020. Ce modèle est une amélioration de BERT (*Bidirectional Encoder Representations from Transformers*), qui est un des modèles les plus influents dans le domaine du *Natural Language Processing* (NLP). DeBERTa a été conçu pour résoudre certaines limitations de BERT, en particulier sur la façon dont les positions des tokens sont encodées et comment l'attention est calculée entre ces derniers.

- **Année de publication :** 2020
- **Équipe de création :** Microsoft Research (Heung-Yeung Shum et son équipe)
- **Lien ArXiv :** <https://arxiv.org/abs/2006.03654>
- **Lien GitHub :** <https://github.com/microsoft/DeBERTa>

Objectifs de DeBERTa :

L'objectif principal de DeBERTa est d'améliorer la compréhension contextuelle des textes, en capturant mieux les relations entre les mots dans une phrase, même lorsqu'ils sont éloignés. Cela se fait grâce à deux innovations principales : l'attention désentrelée et un encodage des positions plus sophistiqué.

2. Principes de Fonctionnement de DeBERTa

2.1. Disentangled Attention Mechanism

Explication de l'Attention Désentrelée :

Dans les modèles traditionnels comme BERT, l'attention est calculée de manière globale, c'est-à-dire que les relations entre les tokens (mots) sont apprises sans distinction claire entre leurs rôles spécifiques (par exemple, sujet, verbe, objet). DeBERTa introduit un mécanisme d'attention désentrelée où les vecteurs d'attention sont séparés en deux parties distinctes :

- **Contenu :** Qui capture la signification du token lui-même.
- **Position :** Qui encode la position relative ou absolue du token dans la phrase.

Ce mécanisme permet à DeBERTa de modéliser de manière plus précise les relations complexes entre les tokens, en tenant compte de leur rôle contextuel et de leur position dans la phrase. Cette amélioration est particulièrement bénéfique pour les tâches qui nécessitent de comprendre des dépendances longues.

Formule d'Attention Désentrelée :

L'attention dans DeBERTa peut être formalisée de la manière suivante :

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} + P \right) V$$

Où :

- **Q** (Query) et **K** (Key) représentent le contenu des tokens,
- **P** représente les informations de positionnement,
- **V** (Value) représente les valeurs des tokens.

Avantages :

Ce mécanisme améliore la capacité du modèle à comprendre les dépendances à longue portée dans les textes, ce qui est particulièrement utile pour des tâches comme la classification de texte, la reconnaissance d'entités nommées, et la compréhension de documents.

2.2. Positionnal Encoding Amélioré**Problème avec BERT :**

Dans BERT, les positions des tokens sont encodées de manière statique, ce qui limite la capacité du modèle à généraliser correctement lorsqu'il rencontre des phrases de longueur ou de structure différentes de celles qu'il a vues pendant l'entraînement.

Solution DeBERTa :

DeBERTa utilise un encodage des positions plus sophistiqué, où la position des tokens est modélisée indépendamment de leur contenu. Cela permet au modèle de mieux gérer des phrases de différentes longueurs et de différents types, en capturant plus précisément la structure syntaxique des phrases.

Formule d'Encodage de Position :

$$P_i = \text{Position}_i \cdot \text{Embedding}_i$$

où P_i est le vecteur de position désentrelée pour le token i , permettant une meilleure gestion des dépendances longues.

Avantages :

Cela rend DeBERTa plus robuste et plus performant sur une large variété de tâches NLP, en particulier celles qui nécessitent une compréhension fine des relations entre les tokens dans des phrases complexes.

3. Avantages Théoriques et Pratiques**Meilleure Compréhension Contextuelle :**

Grâce à ses innovations, DeBERTa excelle dans la compréhension du contexte global des phrases. Cela en fait un modèle puissant pour des tâches où la précision du contexte est cruciale, comme la classification fine des textes, la détection de sentiments, et les systèmes de question-réponse.

Résultats Empiriques :

Dans plusieurs benchmarks, DeBERTa a surpassé BERT et même RoBERTa (une autre variante améliorée de BERT), en obtenant des scores plus élevés sur des tâches comme GLUE, SQuAD, et MNLI, montrant ainsi son efficacité sur une large gamme de tâches NLP.

La Modélisation

1. Méthodologie de Modélisation

1.1. Préparation des Données

Pour garantir que la répartition des classes soit équilibrée lors de la division des données, nous avons utilisé la technique de **StratifiedKFold**. Cette méthode permet de diviser le dataset en plusieurs sous-ensembles tout en conservant la proportion des classes dans chaque pli. Voici les étapes suivies :

- **Division en 2 parties** : Nous avons divisé le dataset en deux parties pour rendre l'exécution du modèle DeBERTa plus efficace, étant donné sa lourdeur computationnelle.
- **StratifiedKFold** : L'algorithme StratifiedKFold a été utilisé pour diviser les données en conservant l'équilibre des classes dans chaque partie.
- **Équilibre des classes** : Chaque partie du dataset maintient une répartition équilibrée des classes, ce qui permet au modèle d'être formé et testé de manière représentative de la distribution globale des données.

1.2. Extraction des Caractéristiques

Pour la comparaison entre BERT et DeBERTa, les caractéristiques textuelles ont été extraites en utilisant les deux modèles. Voici le processus suivi pour chaque modèle :

- **Extraction avec BERT** :
 - **Modèle Utilisé** : bert-base-uncased.
 - **Tokenisation** : Les textes ont été tokenisés et encodés en utilisant le tokenizer BERT.
 - **Extraction des États Cachés** : Les états cachés de la dernière couche de BERT ont été moyennés pour obtenir une représentation vectorielle de chaque description de produit.
 - **Réduction de Dimension** : Les représentations vectorielles extraites ont ensuite été réduites en dimensionnalité en utilisant PCA, et les résultats ont été visualisés en 2D avec t-SNE.
- **Extraction avec DeBERTa** :
 - **Modèle Utilisé** : microsoft/deberta-large.
 - **Tokenisation et Préparation des Données** : Comme pour BERT, les textes ont été tokenisés, et les tokens ont été encodés pour être utilisés dans le modèle DeBERTa.

- **Extraction des Caractéristiques** : Les caractéristiques textuelles ont été extraites en utilisant le mécanisme d'attention désentrelée de DeBERTa, fournissant une représentation vectorielle améliorée des textes.
- **Réduction de Dimension** : Les caractéristiques extraites ont été réduites en dimensionnalité avec PCA, puis projetées en 2D avec t-SNE pour la visualisation et l'analyse des clusters.

1.3. Métrique d'Évaluation

ARI (Adjusted Rand Index) :

- **Justification du Choix** : L'ARI est une métrique robuste pour évaluer la qualité des clusters générés par un algorithme de clustering. Elle mesure la similarité entre les clusters prévus et les classes réelles, tout en prenant en compte les permutations possibles. Un score ARI de 1 indique une correspondance parfaite entre les clusters et les classes, tandis qu'un score de 0 indique une correspondance aléatoire.
- **Interprétation** : L'ARI nous permet de comparer directement la performance de BERT et DeBERTa en termes de capacité à regrouper les descriptions de produits dans les catégories correctes.

Autres Métriques :

- **Matrice de Confusion** : En plus de l'ARI, une matrice de confusion a été utilisée pour évaluer la précision des clusters en termes de correspondance avec les classes réelles. La matrice de confusion permet d'identifier les catégories où le modèle a le plus de mal à discriminer correctement les classes.

1.4. Démarche d'Optimisation

Pour maximiser les performances de DeBERTa et BERT, plusieurs ajustements d'hyperparamètres ont été effectués :

- **Taille des Lots (Batch Size)** : La taille des lots a été augmentée progressivement pour permettre au modèle de traiter plus de données à chaque itération, tout en surveillant l'utilisation de la mémoire pour éviter les dépassements. Une taille de lot de 16 a été retenue comme étant un bon compromis entre performance et utilisation de la mémoire.
- **Longueur Maximale des Séquences (max_length)** : La longueur maximale des séquences a été ajustée pour capturer l'essentiel de l'information textuelle sans tronquer les descriptions importantes. Une longueur maximale de 128 tokens a été choisie après des tests pour assurer une bonne couverture des descriptions tout en restant dans des limites raisonnables de calcul.

- **Nombre d'itérations (n_iter) pour t-SNE :** Pour améliorer la visualisation des clusters, le nombre d'itérations du t-SNE a été fixé à 2000, permettant une convergence plus stable et une meilleure séparation des clusters.

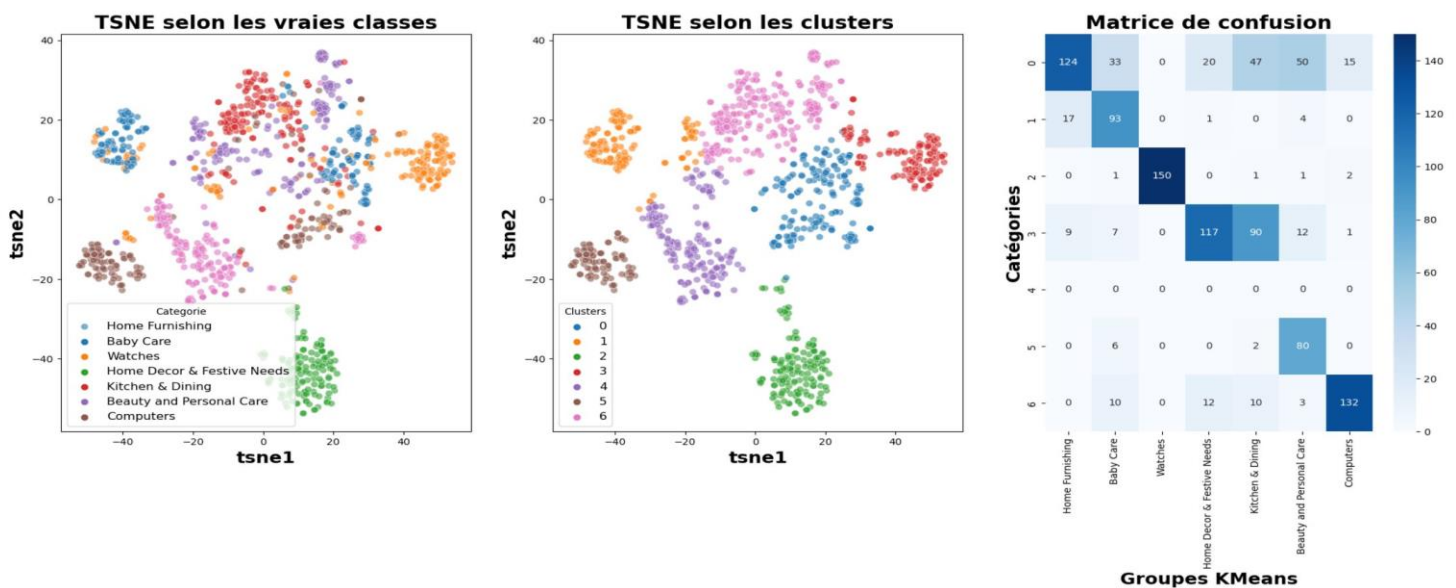
Ces ajustements ont été faits pour maximiser la performance des modèles dans le cadre de cette tâche spécifique, tout en assurant une utilisation optimale des ressources disponibles.

Synthèse des Résultats

1. Comparaison des Résultats Entre BERT et DeBERTa

1.1. Résultats de BERT

- **Visualisation des Clusters avec t-SNE** : Le modèle BERT montre une répartition relativement claire des classes lorsqu'elles sont projetées en 2D avec t-SNE. Les clusters sont bien séparés pour certaines catégories, comme "Home Furnishing" et "Computers".
- **Matrice de Confusion** : La matrice de confusion montre que certaines catégories, telles que "Computers" et "Watches", sont bien identifiées, tandis que d'autres, comme "Baby Care" et "Home Decor & Festive Needs", sont plus souvent confondues avec d'autres classes.
- **Indice de Rand Ajusté (ARI)** : L'ARI obtenu avec BERT est de 0,468, ce qui indique une correspondance modérée entre les clusters prévus par le modèle et les classes réelles.

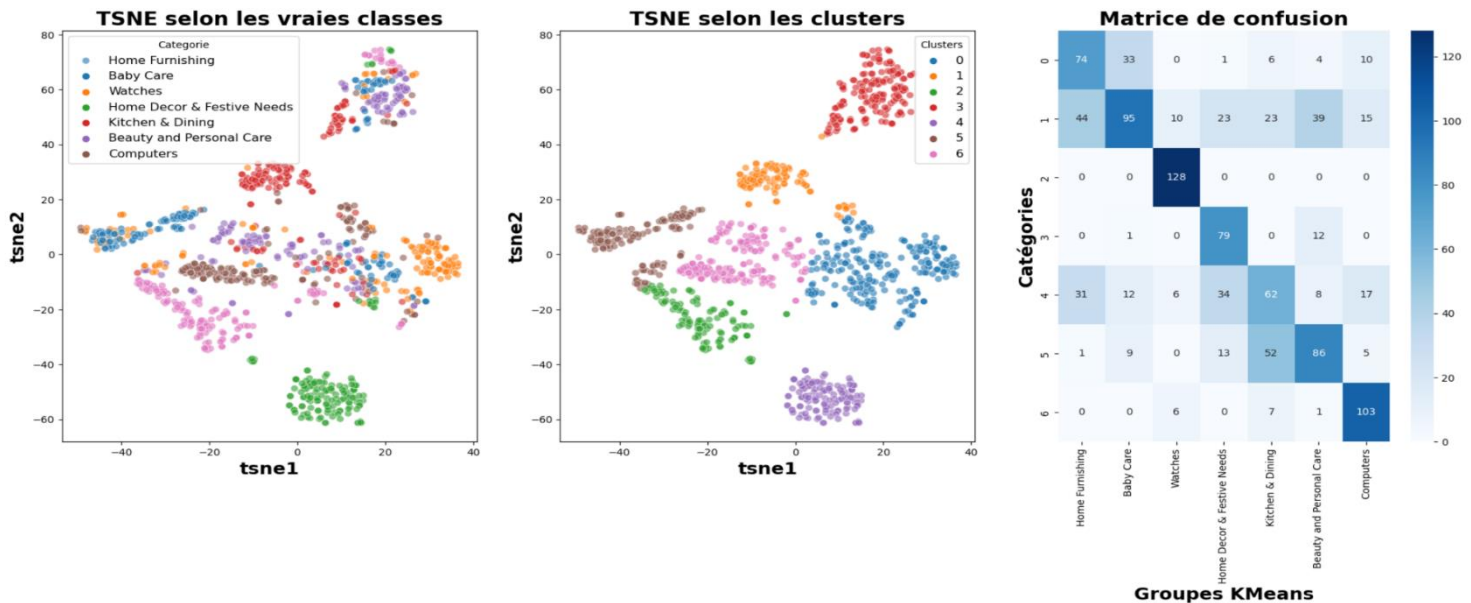


L'indice de Rand ajusté (ARI) pour ce modèle d'extraction de caractéristiques est de 0.468.

1.2. Résultats de DeBERTa (Modèle de Base - deberta-base)

- **Visualisation des Clusters avec t-SNE** : Comparé à BERT, les clusters générés par le modèle DeBERTa de base semblent moins bien séparés. Les catégories "Home Furnishing" et "Computers" sont toujours bien représentées, mais d'autres classes montrent des chevauchements plus importants.

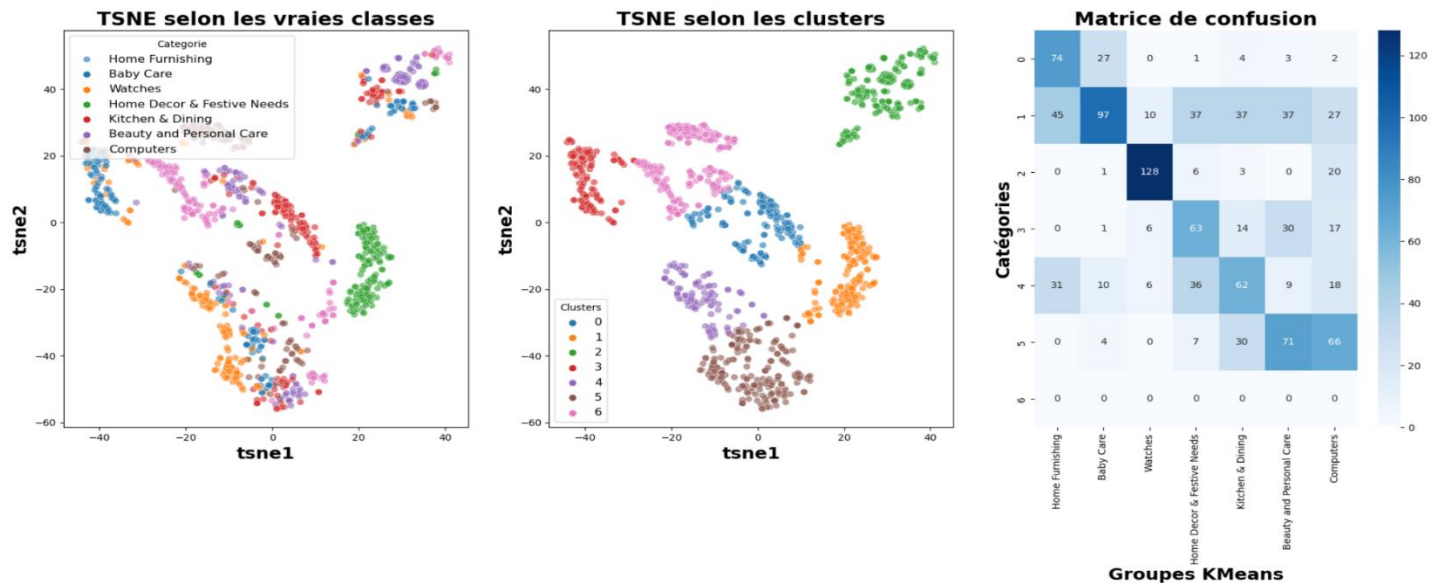
- **Matrice de Confusion** : La matrice de confusion pour DeBERTa base montre une performance légèrement inférieure, avec davantage de confusion entre certaines catégories.
- **Indice de Rand Ajusté (ARI)** : L'ARI obtenu avec le modèle DeBERTa base est de 0,324, ce qui est inférieur à celui de BERT, indiquant une correspondance moins bonne entre les clusters et les classes réelles.



L'indice de Rand ajusté (ARI) pour ce modèle d'extraction de caractéristiques est de 0.324.

1.3. Résultats de DeBERTa (Modèle Plus Grand - deberta-large)

- **Visualisation des Clusters avec t-SNE** : Le modèle DeBERTa plus grand montre une amélioration par rapport au modèle de base, avec des clusters légèrement mieux définis, bien que certains chevauchements persistent.
- **Matrice de Confusion** : La matrice de confusion pour DeBERTa large montre une amélioration par rapport au modèle de base, mais elle reste inférieure à BERT dans certaines catégories.
- **Indice de Rand Ajusté (ARI)** : L'ARI obtenu avec le modèle DeBERTa large est de 0,23, ce qui est également inférieur à BERT. Cela suggère que malgré les améliorations théoriques de DeBERTa, il n'a pas surpassé BERT sur ce jeu de données spécifique.



L'indice de Rand ajusté (ARI) pour ce modèle d'extraction de caractéristiques est de 0.23.

2. Analyse des Résultats

- Performances Relatives :** Bien que DeBERTa soit théoriquement plus avancé que BERT, les résultats montrent que BERT a mieux performé sur ce dataset particulier en termes de clustering des catégories de produits. Cela peut être dû à plusieurs facteurs, tels que la complexité du modèle DeBERTa, qui peut nécessiter un réglage plus fin des hyperparamètres ou une architecture spécifique plus adaptée pour ce type de données.
- Temps de Traitement :** De plus, DeBERTa, en particulier dans sa version large, a nécessité un temps de traitement considérablement plus long que BERT, ce qui peut également être un facteur à considérer dans le choix du modèle pour des applications en production.
- Matrice de Confusion :** Les matrices de confusion révèlent que certains types de produits sont systématiquement mieux classés que d'autres, quel que soit le modèle utilisé. Cela pourrait indiquer la nécessité d'un prétraitement supplémentaire ou d'une adaptation du modèle pour ces catégories spécifiques.

Conclusion

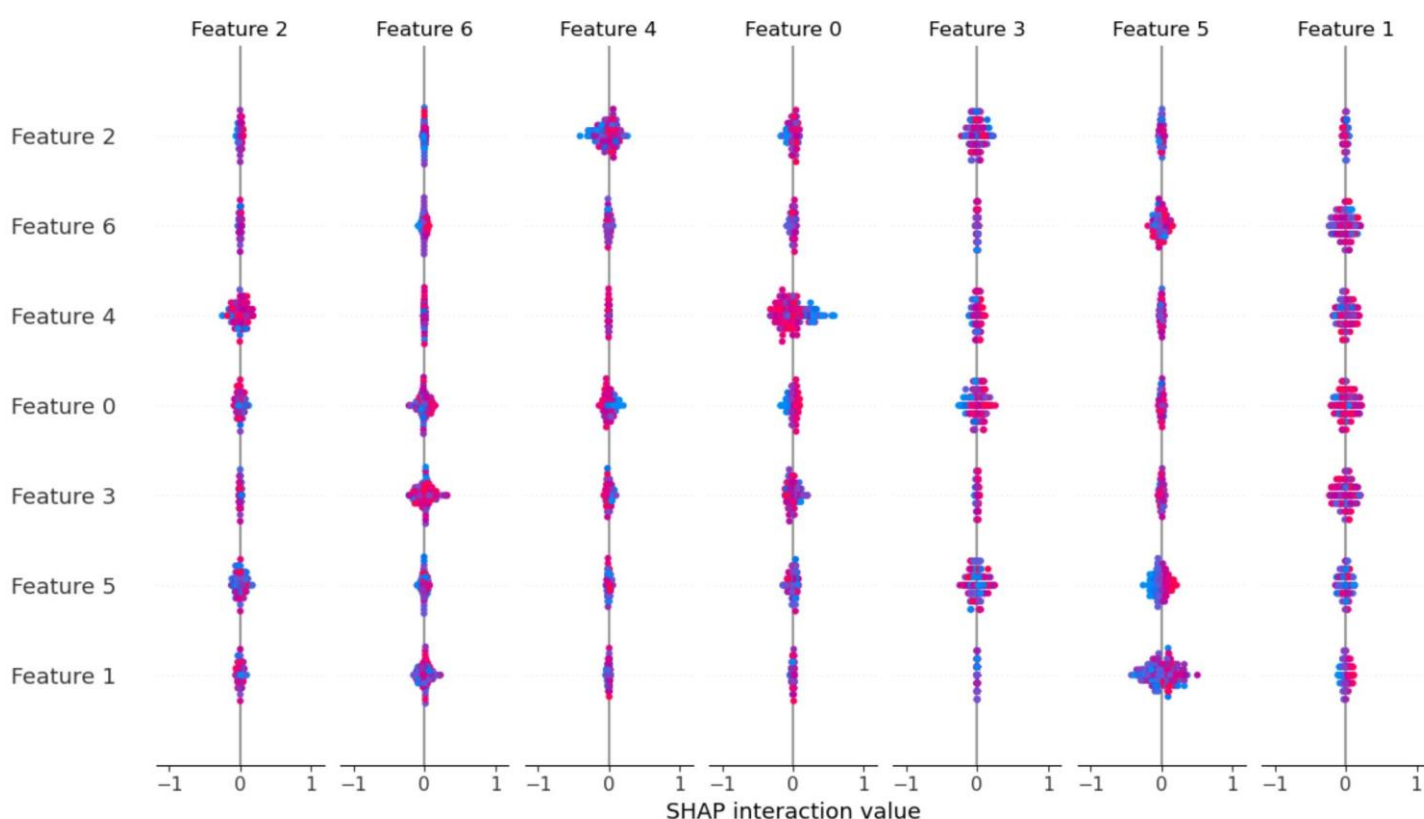
La comparaison des performances entre BERT et DeBERTa sur ce dataset montre que BERT reste un modèle très compétitif, avec des performances supérieures à DeBERTa dans ce cas précis. DeBERTa, malgré ses avancées théoriques, n'a pas atteint les performances escomptées, ce qui suggère que des ajustements supplémentaires pourraient être nécessaires pour exploiter pleinement son potentiel. En fonction de ces résultats, il pourrait être intéressant d'explorer d'autres modèles ou d'améliorer l'optimisation de DeBERTa pour mieux répondre aux spécificités de ce dataset.

Analyse de la Feature Importance Globale et Locale du Modèle DeBERTa

1. Feature Importance Globale

Ce graphique présente les valeurs d'interaction SHAP des caractéristiques les plus influentes pour le modèle DeBERTa. Les valeurs SHAP mesurent l'impact de chaque caractéristique sur les prédictions du modèle.

- Interprétation Globale :** Chaque point représente l'impact d'une caractéristique sur une observation donnée. Les points bleus indiquent que la caractéristique pousse la prédiction vers une classe spécifique, tandis que les points rouges la poussent vers une autre. L'étalement des points montre la variabilité de l'importance des caractéristiques selon les observations.
 - Les **features 2, 6 et 4** semblent avoir une grande influence sur les prédictions, comme le montre la dispersion des points dans ces colonnes.
 - Les **features 1 et 5** ont un impact plus concentré, avec moins de variabilité, ce qui signifie qu'elles sont plus systématiquement importantes dans la décision du modèle.



2. Visualisation de la Valeur de Base du Modèle (Baseline Value)

Ce graphique montre comment la valeur prédite finale (base value) est ajustée par les contributions des différentes caractéristiques.

- **Interprétation Locale :** La base value (0,917 ici) est le point de départ de la prédiction, et chaque caractéristique apporte une contribution positive (vers le bleu) ou négative (vers le rouge) à cette prédiction.
- Cela montre bien que certaines caractéristiques influencent la décision finale dans des directions opposées selon leur valeur et leur interaction avec d'autres caractéristiques.

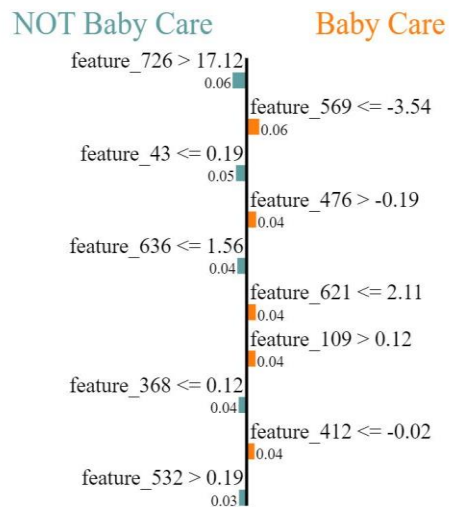
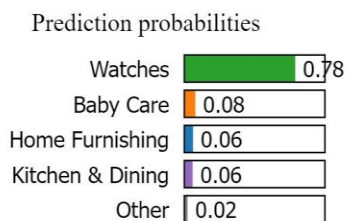


3. Feature Importance Locale

Ce graphique montre une prédiction locale spécifique pour un exemple donné, ainsi que les probabilités de classification. Il détaille également l'importance des caractéristiques individuelles pour cette prédiction.

- **Prédiction :** Le modèle prédit avec une probabilité de 0,78 que l'article appartient à la catégorie "Watches".
- **Explication Locale :** Les caractéristiques qui ont eu le plus grand impact sur cette prédiction sont feature_726, feature_569, et feature_476. Ces caractéristiques sont les plus influentes pour pousser la prédiction vers la classe "Baby Care" ou la classe "Watches".
- **Contributions des Caractéristiques :**
 - feature_726 (valeur 18.42) a poussé fortement la prédiction vers "Watches".
 - feature_569 et d'autres ont contribué à éviter une classification dans la catégorie "Baby Care".

Intercept 0.2757277892293717
 Prediction_local [0.26864529]
 Right: 0.07759109981695678



Feature	Value
feature_726	18.42
feature_569	-3.81
feature_43	0.01
feature_476	-0.17
feature_636	1.56
feature_621	2.04
feature_109	0.14
feature_368	0.12
feature_412	-0.10
feature_532	0.27

Conclusion

L'analyse des valeurs SHAP révèle l'importance des caractéristiques à la fois globalement et localement pour les décisions du modèle DeBERTa :

- **Importance Globale :** Certaines caractéristiques, comme feature_2 et feature_6, jouent un rôle essentiel dans le modèle, avec un impact significatif sur les prédictions pour la plupart des observations.
- **Importance Locale :** En fonction des caractéristiques individuelles, les prédictions peuvent varier fortement d'un exemple à l'autre, comme on le voit dans l'exemple prédictif où certaines caractéristiques ont poussé la prédiction vers "Watches", tandis que d'autres ont aidé à éviter "Baby Care".

Les Limites et Améliorations Possibles

1. Limites de l'Approche Actuelle

- **Temps de Calcul Élevé** : Comme le montrent les captures de temps d'exécution, DeBERTa, en particulier dans sa version large, a pris beaucoup de temps à traiter les données. Cela peut être un problème dans des environnements à ressources limitées ou pour des déploiements en temps réel.
- **Performance Modérée du Modèle** : Bien que DeBERTa soit théoriquement plus performant que BERT, les résultats obtenus (avec un ARI de 0,324 et 0,23) sont inférieurs à ceux de BERT sur ce dataset particulier. Cela peut s'expliquer par :
 - Un besoin de réglage plus fin des hyperparamètres.
 - Une plus grande complexité du modèle qui peut être suradaptée à des jeux de données plus petits ou spécifiques comme celui-ci.
- **Variabilité des Catégories** : Certaines catégories, comme "Baby Care" ou "Home Decor & Festive Needs", sont systématiquement mal classées. Cela peut être dû à une représentation textuelle plus ambiguë ou à une taille d'échantillon insuffisante dans ces classes.
- **Taille du Dataset** : Le dataset utilisé comporte 1 050 exemples. Pour des modèles de la taille de DeBERTa, il est possible que cela soit insuffisant pour obtenir de bons résultats. De plus, le déséquilibre dans les classes peut exacerber cette faiblesse.

2. Améliorations Envisageables

- **Optimisation des Hyperparamètres** :
 - **Longueur maximale des séquences** : Ajuster la longueur maximale des séquences pourrait améliorer la qualité des représentations textuelles. Une longueur trop courte risque de tronquer des informations importantes, tandis qu'une longueur trop longue ajoute du bruit.
 - **Taille des lots (batch size)** : Une meilleure gestion de la taille des lots en fonction de la mémoire disponible peut également accélérer le processus sans sacrifier la performance.
- **Augmentation de la Taille du Dataset** : L'ajout de plus d'exemples, surtout pour les catégories sous-représentées, pourrait améliorer les performances du modèle. DeBERTa est conçu pour gérer de grands volumes de données et pourrait montrer une amélioration significative avec un dataset plus important.

- **Approche Multi-Modalités** : Actuellement, seule la description textuelle des produits est utilisée. L'ajout de la modalité image (disponible dans le dataset) pourrait permettre au modèle de mieux capturer les différences entre les produits et ainsi améliorer la classification.
- **Réduction de la Complexité Modèle** : Pour ce type de dataset relativement petit, il peut être intéressant d'explorer des versions allégées de DeBERTa ou d'utiliser des techniques de distillation de modèles (comme DistilBERT) pour obtenir un modèle plus léger tout en maintenant une performance comparable.
- **Utilisation de Méthodes de Pré-Training Personnalisées** : Au lieu d'utiliser un modèle DeBERTa pré-entraîné sur des données générales, il serait pertinent de fine-tuner un modèle sur un corpus textuel plus pertinent pour les descriptions de produits e-commerce. Cela permettrait d'ajuster le modèle aux spécificités de ce type de langage.

Conclusion des Limites et Améliorations

Les résultats obtenus avec DeBERTa, bien qu'inférieurs à BERT dans ce cas particulier, montrent tout de même le potentiel du modèle avec des ajustements adéquats.

L'amélioration des hyperparamètres, l'augmentation de la taille du dataset, et l'ajout de l'analyse d'images pourraient permettre d'exploiter pleinement la puissance de DeBERTa. Par ailleurs, une réduction de la complexité du modèle ou l'utilisation d'un corpus plus spécialisé pourrait améliorer la généralisation et la rapidité des résultats.