**Note méthodologique : preuve de concept**

**I- Dataset retenu**

**Source des données :**

Le dataset utilisé dans cette étude provient d'une plateforme de e-commerce anglophone "place de marché", où des vendeurs proposent des articles à des acheteurs en postant des annonces contenant une photo et une description textuelle, accompagnées de leur catégorie principale (main\_category). Actuellement, l'attribution des catégories aux articles est effectuée manuellement. L'objectif est donc de prédire automatiquement la catégorie à partir du texte, dans une logique de classification multi-classe.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Contenu du dataframe :**

* Le dataset comprend 1 050 produits et 15 colonnes

**Classes cibles :** (Variable cible : main\_category (7 classes))

* La colonne "product\_category\_tree" était la catégorie réelle du produit. Nous en avons extrait la catégorie principale c'est à dire la première catégorie affichée dans l’arborescence, de sorte à avoir "main\_category".
* Les catégories principales sont : *Home Furnishing - Baby Care - Watches - Home Decor & Festive Needs - Kitchen & Dining - Beauty and Personal Care – Computers.*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prétraitement :** Le prétraitement des données a consisté en plusieurs étapes :

* Suppression des colonnes non pertinentes (non nécessaires pour la classification des produits ont été supprimées du dataset.
* Fusion des colonnes "product\_name" et "description" pour créer "cleaned\_txt".
* Conversion du texte en minuscules : Les textes ont été convertis en minuscules pour assurer la cohérence.
* Tokenisation et Suppression des stop words : Les descriptions textuelles des produits ont été tokenisées, Les mots communs et non informatifs (stop words) ont été supprimés des textes pour réduire le bruit dans les données.
* Application de la lemmatisation, en spécifiant la catégorie grammaticale des mots.
* Encodage des catégories : Les catégories de produits ont été encodées pour être utilisées comme étiquettes dans les modèles de classification.
* Le texte propre a été Stocké dans "cleaned\_text".
* Application de toute ces étapes sur une phrases TEST (+ Stemming).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**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.

**II - Les concepts de l’algorithme récent utilisé : DeBERTa**

**Introduction à DeBERTa :**

DeBERTa, acronyme de Decoding-enhanced BERT with Disentangled Attention, a été développé en 2020 par l’équipe Microsoft Research dirigée par Heung-Yeung Shum. Ce modèle constitue une évolution de BERT, l’un des modèles phares en traitement automatique du langage naturel (NLP). DeBERTa vise à pallier certaines limites de BERT, notamment dans la manière dont sont encodées les positions des tokens et dans le calcul de l’attention entre eux.

* Papier officiel : *He, P., Gao, J., Chen, W., & Li, Z. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. arXiv:2006.03654*
* Année de publication : 2020
* Équipe : Microsoft Research (Heung-Yeung Shum et collaborateurs)
* Lien ArXiv : <https://arxiv.org/abs/2006.03654>
* Code source : https://github.com/microsoft/DeBERTa

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**L’objectif principal de DeBERTa :**

DeBERTa cherche à mieux saisir les relations entre les mots, même lorsqu’ils sont éloignés dans une phrase. Pour cela, il introduit deux grandes innovations : un mécanisme d’attention "désentrelée" et un encodage positionnel amélioré.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Principes de Fonctionnement :**

• Mécanisme d’Attention Désentrelée (**Disentangled Attention Mechanism)**

Contrairement aux modèles classiques comme BERT où l’attention est calculée globalement sans différencier clairement leur sens et leur position dans la phrase, DeBERTa change cela en séparant les vecteurs d’attention en deux :

* **Contenu :** pour la signification du mot
* **Position :** pour sa place dans la phrase

Cette distinction permet de mieux comprendre le rôle de chaque mot dans son contexte, surtout dans les phrases longues.

**Formule d'Attention Désentrelée :** L'attention dans DeBERTa peut être formalisée de la manière suivante : 

Où :

* ***Q* (Query)** et ***K* (Key)** représentent le contenu des tokens,
* **P** représente la position
* **V (Value)** représente les valeurs des tokens à transmettre.

• Encodage Positionnel Amélioré (**Positionnal Encoding Amélioré)**

BERT encode les positions de manière statique, ce qui peut poser problème si la structure des phrases change par rapport aux données d’entraînement.

Dans DeBERTa, l’information de position est traitée séparément du contenu. Cela rend le modèle plus flexible pour traiter des textes variés, de longueurs différentes, tout en respectant la structure grammaticale.

• Avantages Théoriques et Pratiques

Grâce à ses innovations, DeBERTa offre une meilleure compréhension globale du sens d’un texte, ce qui le rend particulièrement performant pour des tâches telles que la classification fine de textes, la détection de sentiments, ou les systèmes de questions-réponses. Il a surpassé BERT et RoBERTa sur plusieurs benchmarks (GLUE, SQuAD, MNLI).

**III - La modélisation**

**Préparation des Données :**

* **StratifiedKFold**  
  Pour garantir une répartition équilibrée des classes, les données ont été divisées à l’aide de la méthode StratifiedKFold. Cela permet de maintenir les proportions de chaque classe dans les sous-ensembles.
* **Pourquoi ?**  
  DeBERTa est un modèle lourd, donc les données ont été divisées en deux parties pour alléger l’entraînement tout en conservant un bon équilibre.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Extraction des Caractéristiques :**

* **Avec BERT :**
* Modèle utilisé : bert-base-uncased
* Tokenisation et encodage via le tokenizer BERT
* Moyenne des états cachés de la dernière couche pour obtenir des vecteurs de représentation
* Réduction dimensionnelle via PCA, visualisation 2D avec t-SNE
* **Avec DeBERTa :**
* Modèle utilisé : microsoft/deberta-large
* Tokenisation et encodage adaptés
* Extraction via le mécanisme d’attention désentrelée fournissant des représentations améliorées
* Réduction dimensionnelle et visualisation similaires à BERT

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Évaluation et Optimisation :**

* **Métrique principale** : Adjusted Rand Index (ARI), mesurant la qualité des clusters obtenus par rapport aux classes réelles.
* **Pourquoi l’ARI ?**

*L’ARI mesure la qualité des clusters produits, en comparant les prédictions aux vraies classes.*

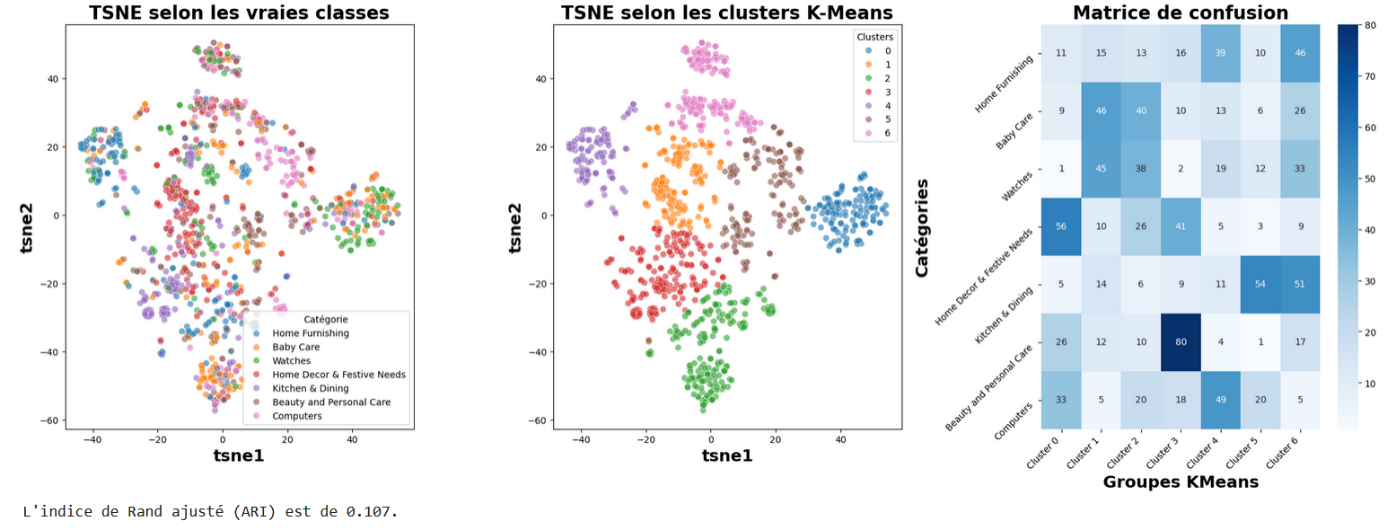
*Score de 1 : correspondance parfaite*

*Score de 0 : résultat aléatoire*

* **Complément : matrice de confusion** pour voir quelles classes sont confondues.
* **Hyperparamètres ajustés :**
* taille des lots (batch size = 16), Augmentée progressivement (valeur finale : 16).
* longueur maximale des séquences (**Max sequence length**) **:** Réglée à 128 pour ne pas tronquer l’information utile
* nombre d’itérations de t-SNE (2000) pour optimiser visualisation.

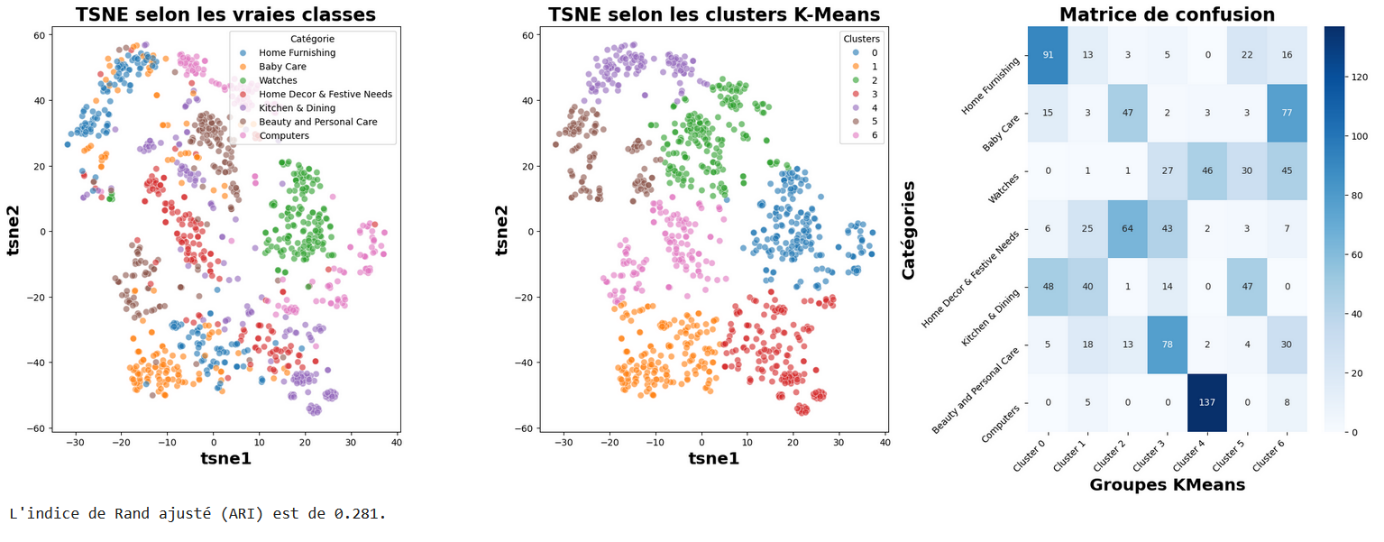
**IV - synthèse des résultats**

**BERT** : **BERT base** génère des représentations qui ne permettent pas une séparation claire entre les classes. L'indice de Rand ajusté (ARI) est de **0,107** montre que les clusters produits par KMeans ne correspondent que très partiellement aux vraies catégories.



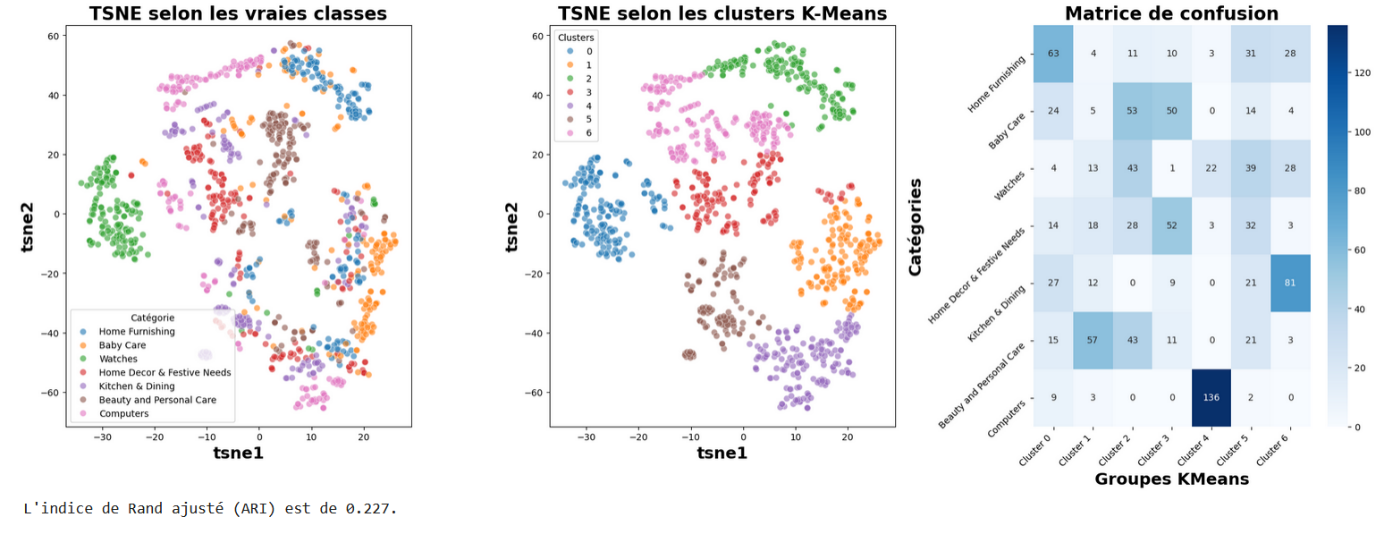
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DeBERTa base** : Ce modèle surpasse BERT avec un **ARI de 0,281**, indiquant des regroupements mieux alignés avec les catégories réelles. Les clusters restent imparfaits mais montrent une meilleure structuration. les groupes sont mieux formés, avec des frontières plus claires.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DeBERTa large** : Malgré des ajustements d’hyperparamètres (taille de lot augmentée, séquences allongées), le score ARI atteint **0,227**, légèrement inférieur à la version base. Cela suggère que l’augmentation de la capacité du modèle ne garantit pas une meilleure performance sans réglages plus poussés ou données supplémentaires. Cela peut également s'expliquer par une complexité plus grande, qui peut nécessiter plus de données pour être pleinement exploitée.

les clusters sont présents mais plus flous que ceux obtenus avec DeBERTa base.



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Cette comparaison révèle que **DeBERTa**, notamment en version base, **offre de meilleures représentations textuelles** que BERT sur ce jeu de données, mais que **la version large demande encore un calibrage fin** pour exploiter tout son potentiel. Il est possible que les embeddings plus complexes captent des nuances qui ne se traduisent pas directement en regroupements plus nets sans traitement complémentaire.

**Conclusion**

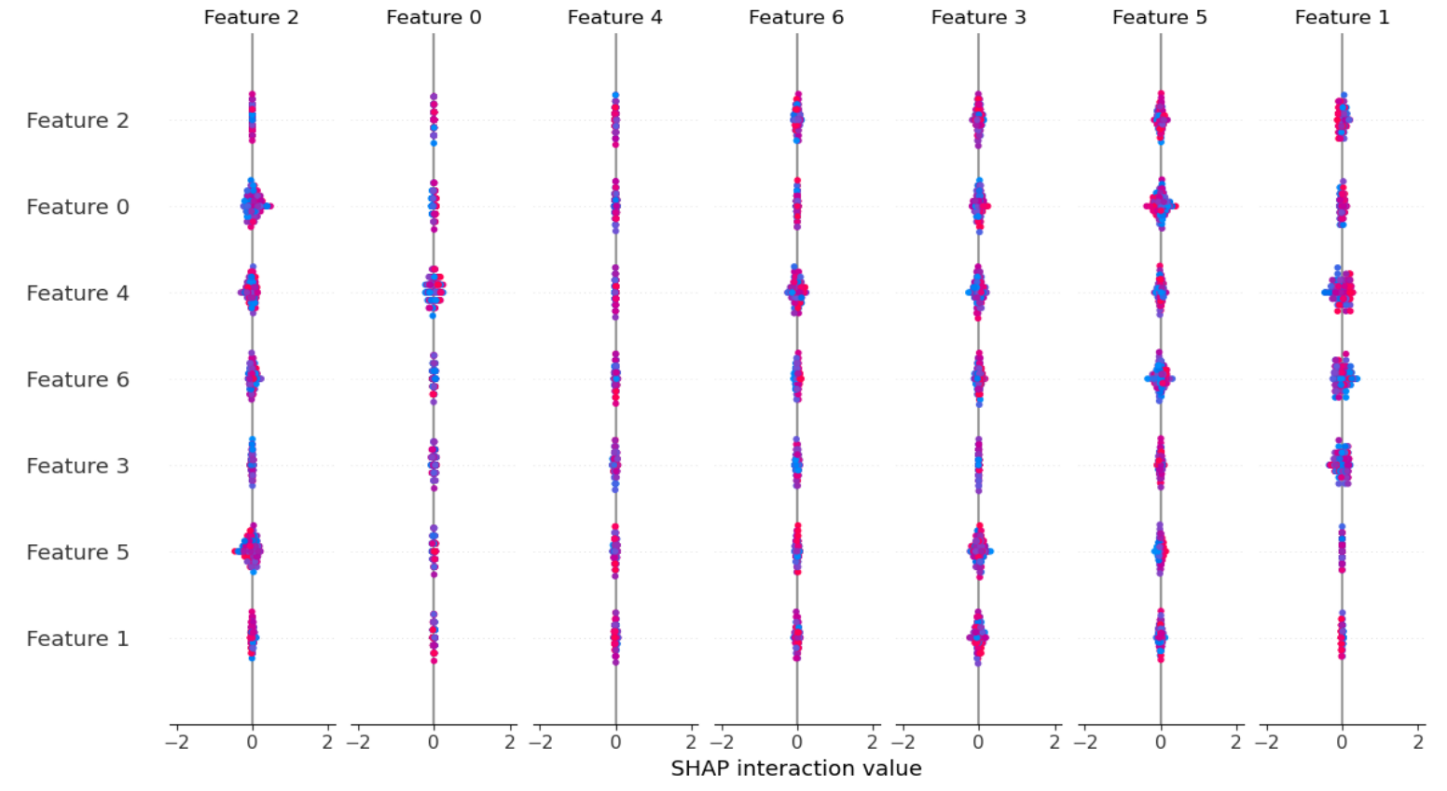
* **DeBERTa base surpasse BERT**, confirmant l'intérêt des améliorations structurelles du modèle pour produire des représentations plus informatives.
* **BERT reste une base solide**, avec des résultats corrects mais moins discriminants.
* **DeBERTa large**, malgré un plus grand potentiel théorique, **n’a pas produit les meilleurs résultats**, possiblement en raison d’un besoin accru en tuning ou d’une plus grande sensibilité à la qualité des données d’entrée.

Des pistes à explorer incluraient :  
– un affinement des hyperparamètres via validation croisée sur des tâches en aval,  
– l’usage de méthodes de réduction de dimensionnalité mieux adaptées aux représentations contextuelles,  
– ou encore l’évaluation de modèles alternatifs comme RoBERTa ou des variantes multilingues selon le jeu de données.

**V - L’analyse de la feature importance globale et locale du nouveau modèle**

**Feature Importance Globale :**

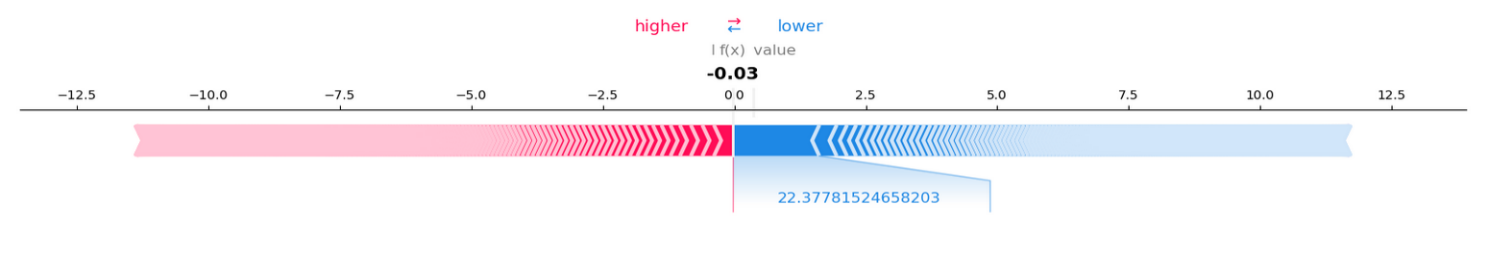
* Ce graphique en violons montre les valeurs d'interaction SHAP des caractéristiques les plus influentes pour le modèle DeBERTa.
* Les valeurs d’interaction SHAP montrent l’impact relatif des caractéristiques sur les prédictions du modèle DeBERTa.
* **Interprétation Globale :** Chaque point représente 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.
* 🔵 Bleu = contribution à une classe (ex : « Watches »),
* 🔴 Rouge = contribution à une autre (ex : « Baby Care »).
* Les features 2, 4 et 6 montrent une forte variabilité d’impact, ce qui indique qu’elles sont influentes mais contextuelles.
* Un impact centré près de zéro reflète une faible contribution moyenne.



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Force de la Prédiction :**

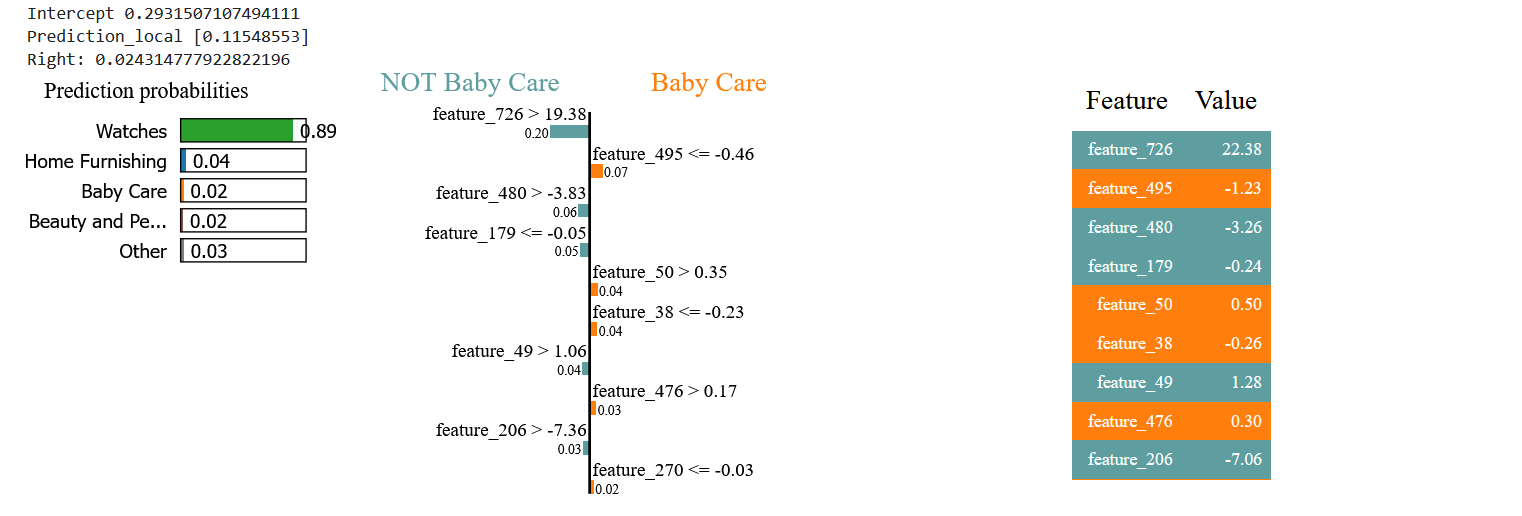
* Ce graphique montre comment la valeur prédite finale (base value) est ajustée par les contributions des différentes caractéristiques. En d’autres mots, il montre **comment les contributions SHAP s’additionnent** à partir d’une valeur de base (base value) pour produire la prédiction finale.
* **Interprétation Locale :** La base value 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.
* Axe horizontal : impact cumulé de chaque feature.
* Les flèches rouges diminuent la prédiction, les flèches bleues l’augmentent.
* Ici, la prédiction finale est -0.03, obtenue par la somme des effets positifs et négatifs



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**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. Il s’agit de l’explication **d’une prédiction précise**.
* **Probabilités de classification :** l’instance est prédite comme appartenant à la classe "Watches" avec une probabilité de **0.89.**
* Les barres horizontales indiquent l'effet de chaque feature sur la classification :
* Vers "Baby Care",
* Vers "NOT Baby Care" (ici : « Watches »).
* À droite, les **valeurs réelles des features** sont colorées selon leur orientation.



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion**

L'analyse des valeurs SHAP révèle l'importance des caractéristiques globales et locales pour les décisions du modèle DeBERTa :

* **Importance Globale :** certaines features (ex. : feature\_2, feature\_6) sont essentielles pour le modèle, mais leur importance varie selon les cas.
* **Importance Locale :** pour cette instance, **feature\_726** a fortement poussé la prédiction vers « Watches », tandis que d’autres (feature\_495, feature\_480…) ont freiné la classe « Baby Care ».

**VI - Les limites et les améliorations possibles**

**Limites de l’Approche Actuelle :**

* **Temps de calcul important**

L’un des principaux inconvénients de l'utilisation de modèles comme DeBERTa-large est leur coût computationnel élevé. Le traitement des données textuelles prend un temps non négligeable, en particulier lors de l’extraction des embeddings.

Par exemple lors de l’extraction avec microsoft/deberta-large, l'utilisation d’un batch size de 16 et d’une longueur maximale de 128 tokens a permis d’accélérer un peu le traitement, mais le modèle reste exigeant en ressources.

Leur utilisation sur de petites machines ou en production temps réel peut poser problème.

* **Performance modérée malgré un modèle avancé**

Bien que DeBERTa soit supposé surpasser BERT sur de nombreuses tâches NLP, les résultats sur ce dataset (avec un ARI max de 0.281) sont modestes.Raisons possibles :

* Complexité excessive du modèle par rapport à la taille des données.
* un tuning d’hyperparamètres limité.
* un surapprentissage potentiel dû à la petite taille du dataset.
* **Taille limitée du dataset et déséquilibre des catégories**

Avec seulement 1 050 exemples et des classes sous-représentées (ex. : "Baby Care"), le modèle manque de données pour bien généraliser, ce qui nuit à la qualité du clustering.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Améliorations Possibles :**

* **Ajuster les hyperparamètres**

Une exploration plus fine de la longueur maximale des séquences ou de la taille des lots pourrait améliorer les résultats (ex. : max\_length=128, batch\_size=16 ont déjà montré un léger gain).

* **Augmenter la taille du dataset**

Plus d’exemples, surtout pour les catégories rares, permettraient de renforcer l’apprentissage et la cohérence des représentations.

* **Réduire la complexité du modèle**

Des alternatives plus légères (comme DistilBERT) peuvent offrir un bon compromis entre qualité et rapidité, surtout sur des petits jeux de données.

* **Adapter le pré-entraînement**

Fine-tuner DeBERTa sur un corpus e-commerce spécifique pourrait améliorer la pertinence des embeddings.

* **Exploiter les images disponibles**

Une approche multi-modale (texte + image) pourrait améliorer la différenciation entre les catégories.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion**

DeBERTa montre un potentiel prometteur, mais son efficacité dépend fortement des ressources disponibles et du contexte d’usage. Avec un meilleur ajustement, un dataset plus riche et des stratégies adaptées, ses performances en clustering pourraient être significativement améliorées.

**Bibliographie**

- Papier officiel : He, P., Gao, J., Chen, W., & Li, Z. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. arXiv:2006.03654.

- He et al. (2021). DeBERTa. https://arxiv.org/abs/2006.03654

- Lundberg & Lee (2017). SHAP. https://arxiv.org/abs/1705.07874

- Ribeiro et al. (2016). LIME. https://arxiv.org/abs/1602.04938