Table des matières

Note méthodologique – Preuve de concept

La veille technologique est un processus structuré visant à collecter, analyser et exploiter des informations sur les innovations et évolutions techniques dans un domaine donné. Elle permet aux organisations de rester informées des avancées, d'anticiper les changements, et d'adapter leurs stratégies afin de conserver un avantage compétitif.

Dans le cadre de ce projet, la veille porte sur les techniques récentes de traitement automatique du langage naturel (NLP), appliquées à la classification automatique de produits à partir de leurs descriptions textuelles. L'objectif est d'étudier et de comparer deux modèles de référence, DistilBERT et DeBERTa, en se concentrant sur la qualité des embeddings qu'ils produisent, et leur impact sur des tâches de classification supervisée et non supervisée.

Cette démarche vise à fournir une preuve de concept permettant de mesurer la valeur ajoutée des innovations récentes en NLP dans un contexte métier concret, tout en garantissant la transparence et l'interprétabilité des modèles utilisés.

1) Présentation des données

Le jeu de données utilisé dans cette preuve de concept provient du site e-commerce Flipkart. Il contient 1 050 produits, chacun décrit par 15 colonnes représentant des informations textuelles, numériques et catégorielles.

Parmi ces colonnes, on trouve notamment :

- uniq_id : identifiant unique du produit
- product_name : nom du produit
- product_category_tree : catégorie hiérarchique du produit
- description : description textuelle du produit
- brand : marque (souvent manquante)
- retail_price et discounted_price : prix avant/après remise
- product_rating / overall_rating : notes associées au produit
- product_specifications : spécifications techniques (texte ou JSON)

La variable d'intérêt pour la classification supervisée est le premier niveau de la colonne product_category_tree, représentant la catégorie principale du produit. Ce niveau contient 7 classes équilibrées (150 produits chacune), ce qui en fait un bon candidat pour la modélisation.

Pour simplifier l'analyse, seule la colonne description est utilisée comme variable explicative dans ce projet.

2) Méthodologie comparative

Dans cette veille technique, nous comparons deux modèles NLP préentraînés : DistilBERT (distilbert-base-uncased) et DeBERTa. DistilBERT est une version compacte et rapide de BERT, qui conserve l'essentiel de ses performances tout en étant plus efficace en production.

L'objectif est d'évaluer la qualité des embeddings produits par chaque modèle à partir des descriptions produits, sans recourir à un fine-tuning complet. On s'intéresse uniquement à la capacité des embeddings à capturer des informations utiles pour la tâche.

Deux types de tâches sont testées :

- Classification non supervisée : application de l'algorithme K-Means sur les embeddings pour évaluer leur capacité à regrouper les produits de manière cohérente selon leur similarité sémantique.
- Classification supervisée : entraînement d'un classifieur linéaire (régression logistique) sur les embeddings pour prédire la catégorie principale, avec évaluation selon des métriques classiques (accuracy, F1-score).

Cette approche permet de comparer objectivement la qualité des représentations produites par DistilBERT et DeBERTa, sans introduire de biais liés aux autres étapes du pipeline.

Elle s'inscrit dans une démarche de veille exploratoire, visant à déterminer si les améliorations structurelles de DeBERTa apportent une valeur ajoutée concrète par rapport à DistilBERT dans un usage métier réaliste.

3) Présentation des modèles comparés

Afin de comparer objectivement la qualité des embeddings produits, il est essentiel de présenter brièvement les deux modèles de référence retenus dans cette veille : DistilBERT (version distillée et allégée de BERT, ici distilbert-base-uncased), modèle à l'origine de la révolution des Transformers en NLP, et DeBERTa, évolution récente introduisant des mécanismes d'attention désentrelacée. Les paragraphes suivants synthétisent leurs principes de fonctionnement et leur intérêt pour la représentation des textes produits.

3.1) DistilBERT : principe et extraction d'embeddings

Dans cette étude, nous utilisons DistilBERT (distilbert-base-uncased), une version distillée de BERT, pour l'extraction des embeddings. DistilBERT reprend l'architecture et les principes de BERT, tout en étant plus léger et plus rapide, ce qui le rend particulièrement adapté à des usages nécessitant efficacité et rapidité d'inférence.

Pour rappel, BERT (Bidirectional Encoder Representations from Transformers) est un modèle de langage développé par Google en 2018 qui a marqué un tournant dans le traitement automatique du langage naturel (NLP). Sa principale innovation est de produire des embeddings contextuels, c'est-à-dire des représentations vectorielles des mots et des phrases qui tiennent compte du contexte complet dans lequel ils apparaissent[1].

Architecture et fonctionnement

- **Architecture Transformer** : BERT repose sur l'architecture Transformer, qui utilise des mécanismes d'auto-attention pour analyser simultanément tous les mots d'une séquence et établir leurs relations contextuelles.
- **Contexte bidirectionnel** : Contrairement aux modèles traditionnels qui lisaient le texte de gauche à droite ou de droite à gauche, BERT traite chaque mot en tenant compte de tout le contexte, à la fois avant et après le mot cible.
- **Tokenization WordPiece** : BERT segmente les mots en sous-unités (subwords), ce qui lui permet de gérer efficacement les mots rares ou inconnus.

1 J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, « BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding », *arXiv preprint arXiv:1810.04805*, 2018, https://arxiv.org/abs/1810.04805

Extraction des embeddings

- Embeddings de mots et de phrases : BERT génère, pour chaque token d'une séquence, un vecteur dense (embedding) de grande dimension (souvent 768).
- Embeddings contextuels : Ces vecteurs sont sensibles au contexte : le même mot aura une représentation différente selon la phrase où il apparaît, ce qui permet de capturer les nuances de sens (ex : "bank" dans "river bank" vs "bank account")[2].
- Utilisation pour la classification : Pour représenter une séquence entière (phrase, description produit), on utilise généralement l'embedding du token spécial [CLS] ou la moyenne des embeddings de tous les tokens.

Applications et intérêt pour l'extraction d'embeddings

- Recherche sémantique, classification, recommandation : Les embeddings BERT sont utilisés dans de nombreuses applications pour mesurer la similarité sémantique, regrouper des textes, ou servir de base à des modèles de classification[3].
- Apprentissage par transfert : BERT peut être utilisé tel quel (feature extraction) ou adapté à une tâche spécifique par fine-tuning, mais dans ce projet, il sert uniquement à extraire des représentations avancées des textes produits.

BERT permet de transformer chaque description produit en un vecteur contextuel riche, qui capture la signification globale et les subtilités du texte. Ces embeddings servent ensuite de base à des algorithmes de classification ou de clustering, offrant une représentation bien plus performante que les approches classiques (TF-IDF, word2vec).

3.2) **DeBERTa : principe et innovations**

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) est un modèle de langage développé par Microsoft en 2020, qui s'inscrit dans la lignée des modèles Transformers pré-entraînés comme BERT et RoBERTa. Son objectif est d'améliorer la qualité des représentations textuelles pour les tâches de compréhension du langage naturel, tout en étant plus efficace en termes de données et de précision[4].

Innovations principales :

- Attention désentrelacée (Disentangled Attention) : Contrairement à BERT, où chaque mot est représenté par un unique vecteur combinant contenu et position, DeBERTa sépare explicitement ces deux informations : un vecteur encode le contenu sémantique du mot, un autre encode sa position dans la séquence.

---

2 Nils Reimers, Iryna Gurevych, « Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks », *arXiv preprint arXiv:1908.10084*, 2019, https://arxiv.org/abs/1908.10084

3 C. Wang, M. Li, D. Li, « BERT Goes Shopping: Comparing Distributional Models for Product Representations », *Proceedings of the 4th Workshop on e-Commerce and NLP (ECNLP 4), ACL Anthology*, 2021, https://aclanthology.org/2021.ecnlp-1.1.pdf

4 Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, « DeBERTa: Decoding-enhanced BERT with Disentangled Attention », *arXiv preprint arXiv:2006.03654*, 2020. [https://arxiv.org/abs/2006.03654][10]

- L'attention est alors calculée via des matrices distinctes, permettant au modèle de mieux capturer les relations syntaxiques et sémantiques, même dans des phrases complexes ou longues[5].
- Enhanced Mask Decoder : Le module de décodage masqué de DeBERTa intègre plus finement la position absolue des tokens lors de la tâche de pré-entraînement (Masked Language Modeling). Cela améliore la capacité du modèle à apprendre des relations syntaxiques précises et à généraliser sur des tâches complexes.
- Pré-entraînement optimisé : Les versions récentes (DeBERTa v3) utilisent des stratégies de pré-entraînement inspirées d'ELECTRA, permettant d'atteindre des performances de pointe sur des tâches comme la classification de texte, la compréhension de questions ou l'analyse de sentiments, tout en utilisant moins de données que RoBERTa[6].

Extraction des embeddings
Comme BERT, DeBERTa génère un vecteur contextuel pour chaque token de la séquence, mais ses embeddings sont enrichis par la séparation explicite entre contenu et position. Pour représenter une séquence entière (comme une description de produit), on utilise généralement l'embedding du token spécial [CLS] ou la moyenne des embeddings de tous les tokens. Grâce à ses améliorations architecturales, DeBERTa produit des représentations plus discriminantes et plus robustes, particulièrement utiles pour des tâches de classification ou de regroupement sémantique.

Performances et applications :
DeBERTa surpasse BERT et RoBERTa sur la plupart des benchmarks de compréhension du langage naturel (SQuAD, MNLI, etc.), avec des gains de plusieurs points de pourcentage en précision, même avec moins de données d'entraînement1.
Il est particulièrement adapté aux tâches nécessitant une compréhension fine du contexte, des relations sémantiques et de la structure des phrases, comme la classification de texte, l'analyse de sentiments, la reconnaissance d'entités nommées et la réponse automatique à des questions.

DeBERTa se distingue par sa capacité à séparer explicitement le contenu et la position des mots dans ses représentations, ce qui lui permet de générer des embeddings contextuels plus riches et plus précis que ceux de DistilBERT ou RoBERTa. Cette avancée se traduit par de meilleures performances sur de nombreuses tâches de NLP, tout en restant efficace et adaptable à différents volumes de données.

4) Méthodologie de modélisation

4.1) Modélisation

Prétraitement des données

- Pour DistilBERT :

Nettoyage simple (suppression des caractères spéciaux), conversion en minuscules (modèle uncased), pas de suppression des stopwords ni de lemmatisation. La tokenisation propre à DistilBERT est réalisée lors de l'extraction des embeddings.

5 Papers With Code, « DeBERTa Explained », 2024. [https://paperswithcode.com/method/deberta][7]

6 Y. He et al., « Improving DeBERTa using ELECTRA-style pre-training with gradient penalty », *arXiv preprint arXiv:2111.09543*, 2021. [https://arxiv.org/pdf/2111.09543.pdf][8]

- Pour DeBERTa :

Conservation de la casse (modèle cased), suppression des espaces multiples, pas de suppression des stopwords ni de lemmatisation. La tokenisation spécifique à DeBERTa est effectuée lors de l'extraction des embeddings.

Extraction des caractéristiques

- Utilisation de DistilBERT ou DeBERTa pré-entraîné pour transformer chaque description produit en un vecteur d'embedding contextuel.
- Pour chaque texte, récupération de l'embedding du token [CLS], qui sert de représentation globale de la phrase et est couramment utilisé pour les tâches de classification.

Réduction de dimension et visualisation

- Application d'une ACP (Analyse en Composantes Principales) pour réduire la dimensionnalité des embeddings. Réduction de la dimensionnalité des embeddings avec un seuil de 99 % de variance expliquée conservée.

  Cette valeur élevée garantit que l'essentiel de l'information sémantique est préservé tout en simplifiant les données.
- Utilisation de t-SNE pour la visualisation des données dans un espace 2D.

Modélisation non supervisée (Clustering)

- Application de l'algorithme K-Means sur les embeddings réduits pour regrouper les descriptions produits selon leur similarité.
- Évaluation de la cohérence des clusters.

Modélisation supervisée (Classification)

- Séparation du jeu de données en ensembles d'entraînement et de test de façon stratifiée.
- Entraînement d'un modèle de régression logistique sur les embeddings pour prédire la classe des descriptions produits.
- Évaluation des performances sur l'ensemble de test.

4.2) **Métriques d'évaluation utilisées**

Pour le clustering (modélisation non supervisée) :

- **Homogeneity Score :** mesure si chaque cluster ne contient que des membres d'une seule classe réelle (valeur entre 0 et 1, 1 étant parfait).
- **Completeness Score :** mesure si tous les membres d'une même classe réelle sont regroupés dans le même cluster (0 à 1).
- **V-measure :** moyenne harmonique de l'homogeneity et de la completeness, fournissant un indicateur synthétique de la qualité du clustering.
- **Adjusted Rand Index (ARI) :** compare la similarité entre le clustering obtenu et la répartition réelle des classes, en corrigeant l'effet du hasard (score entre -1 et 1).
- **Silhouette Score :** mesure la cohérence interne des clusters, en évaluant la compacité intra-cluster et la séparation inter-cluster (score entre -1 et 1, plus il est proche de 1, mieux c'est).

Pour la classification supervisée :

- **Accuracy :** proportion de prédictions correctes sur l'ensemble d'entraînement et l'ensemble de test, pour évaluer la capacité de généralisation du modèle.
- **Rapport de classification :** inclut plusieurs métriques pour chaque classe :
  - **Précision (precision) :** part des prédictions correctes parmi toutes les prédictions positives pour une classe donnée.
  - **Rappel (recall) :** part des éléments d'une classe correctement identifiés par le modèle.
  - **F1-score :** moyenne harmonique entre la précision et le rappel, synthétisant la performance pour chaque classe.
  - **Moyennes macro et pondérée :** permettent d'évaluer la performance globale en tenant compte ou non de l'équilibre entre les classes.

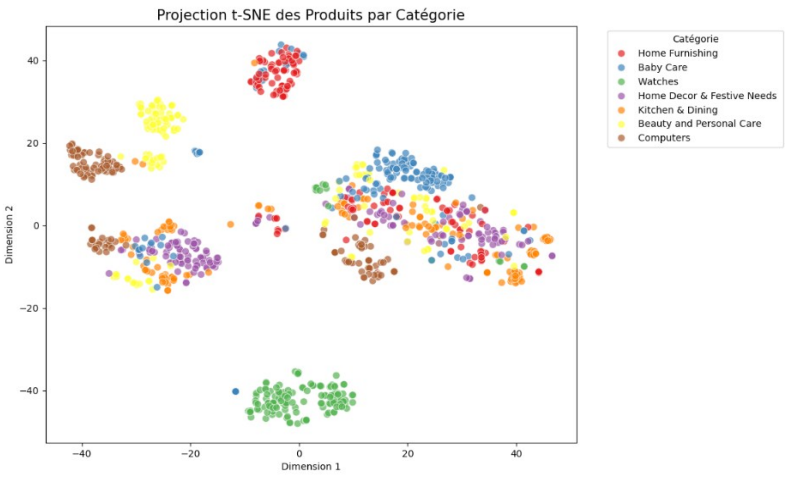
Ces métriques sont choisies car elles permettent d'évaluer à la fois la qualité du regroupement non supervisé (avec et sans connaissance des vraies classes) et la performance détaillée de la classification, notamment en cas de déséquilibre entre les catégories.

5) Synthèse des résultats

5.1) Visualisation des embeddings par t-SNE

La visualisation des embeddings est une étape clé pour explorer la structure des données textuelles dans l'espace généré par les modèles de langage. Les embeddings, qui sont des vecteurs de grande dimension représentant chaque description produit, sont difficilement interprétables directement. Pour faciliter leur analyse, nous utilisons l'algorithme t-SNE (t-distributed stochastic neighbor embedding), une technique de réduction de dimension non linéaire conçue pour projeter des données complexes dans un espace à deux dimensions tout en préservant les relations de proximité entre les points.

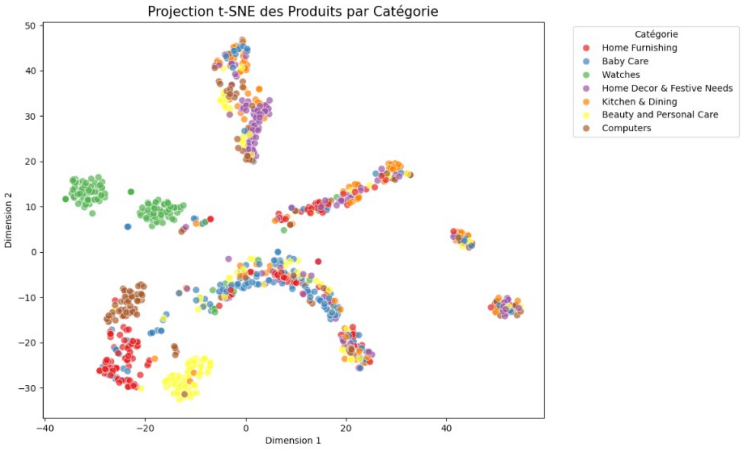
t-SNE attribue à chaque point de l'espace initial une position sur une carte 2D, de sorte que les points proches dans l'espace d'origine restent proches dans la projection, et que les points éloignés restent séparés. Cette méthode permet ainsi de visualiser la répartition naturelle des classes : si les embeddings issus du modèle séparent bien les catégories de produits, cela se traduira par des groupes distincts sur la carte t-SNE. Cette visualisation offre donc un aperçu intuitif de la capacité du modèle à discriminer les différentes classes, avant même toute étape de clustering ou de classification supervisée.



La projection t-SNE ci-dessus montre la répartition des produits dans l'espace des embeddings générés par DistilBERT, chaque point étant colorié selon sa catégorie réelle.

On observe que certaines catégories, comme "Watches" ou "Computers", forment des groupes bien séparés, tandis que d'autres présentent un léger chevauchement.

Cela indique que DistilBERT parvient globalement à distinguer les différentes classes de produits, mais que certaines catégories restent partiellement mélangées dans l'espace de représentation.



La projection t-SNE ci-dessus présente la répartition des produits selon leurs catégories réelles, à partir des embeddings générés par le DeBERTa.

On observe que certaines catégories, comme "Watches" (vert) et "Computers" (marron), forment des groupes bien distincts et compacts. D'autres catégories, en revanche, restent partiellement entremêlées.

Cette visualisation montre que le modèle parvient à séparer certaines classes de façon nette, mais que la distinction entre toutes les catégories n'est pas parfaite.

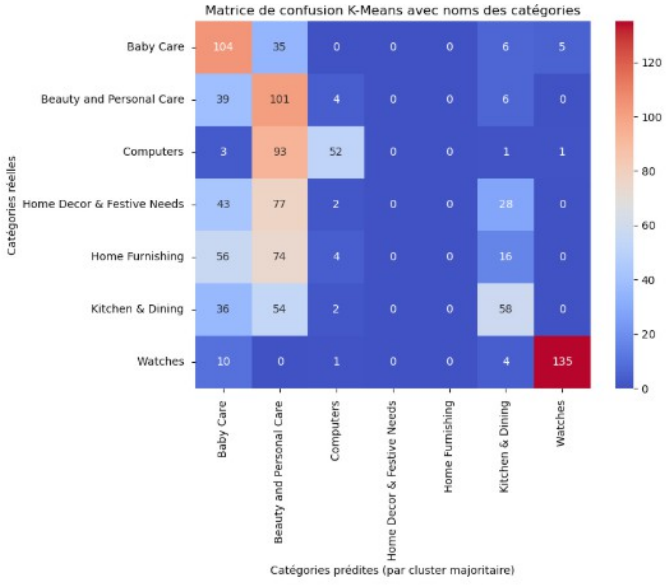
Les visualisations t-SNE ne permettent pas de conclure clairement lequel des deux modèles, DistilBERT ou DeBERTa, offre la meilleure séparation des classes. Elles donnent une intuition sur la structure des données, mais ne suffisent pas à départager les performances des modèles de façon fiable.

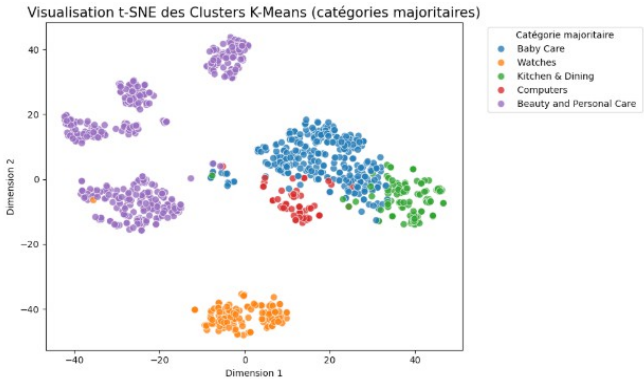
5.2) **Résultats du clustering**

Pour évaluer la capacité des modèles à regrouper les descriptions produits selon leur similarité sémantique, nous avons appliqué l'algorithme K-Means sur les embeddings réduits par ACP. Les résultats obtenus pour DistilBERT et DeBERTa sont comparés à l'aide de plusieurs métriques.

DistilBERT

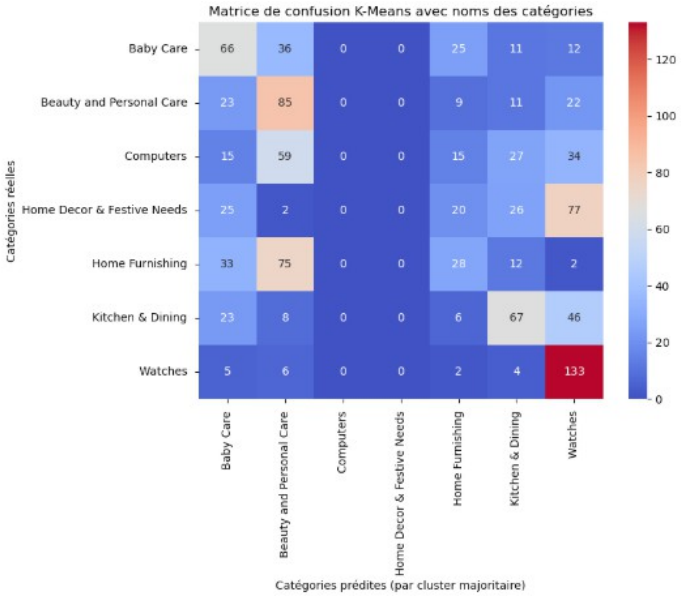
Homogeneity Score	Completeness Score	V-measure	Silhouette Score	Adjusted Rand Index
0.288	0.334	0.31	0.161	0.173

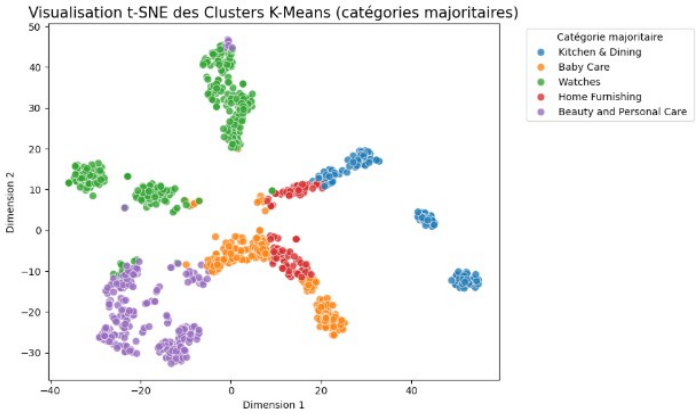




DeBERTa

Homogeneity Score	Completeness Score	V-measure	Silhouette Score	Adjusted Rand Index
0.182	0.201	0.191	0.313	0.144





Les résultats du clustering KMeans montrent que les embeddings DistilBERT produisent des clusters un peu plus cohérents par rapport aux catégories réelles que ceux de DeBERTa, comme l'indiquent les scores d'homogeneity, de completeness, de V-measure et d'Adjusted Rand Index, tous supérieurs pour DistilBERT. Cependant, DeBERTa obtient un score de silhouette plus élevé, ce qui suggère que ses clusters sont plus compacts et mieux séparés dans l'espace des embeddings, même s'ils correspondent moins bien aux vraies catégories.

5.3) Résultats de la classification

Pour évaluer la capacité des embeddings à distinguer les différentes catégories de produits, nous avons entraîné un modèle de régression logistique sur l'ensemble d'entraînement, en utilisant les vecteurs extraits par DistilBERT et DeBERTa. Les performances ont ensuite été mesurées sur l'ensemble de test, afin de comparer l'efficacité des embeddings dans une tâche de classification supervisée.

La régression logistique a été choisie comme classificateur de référence en raison de sa simplicité et de son efficacité pour les tâches de classification supervisée multi-classes. Ce modèle permet d'exploiter directement les embeddings produits par DistilBERT ou DeBERTa pour estimer la probabilité d'appartenance à chaque catégorie. Sa nature linéaire facilite l'interprétation des résultats et limite la complexité du modèle, ce qui est adapté dans un contexte où l'objectif principal est de comparer la qualité des représentations vectorielles extraites.

```
Accuracy sur l'ensemble d'entraînement : 0.9798
Accuracy sur l'ensemble de test : 0.9095

Rapport de classification (test) :
                           precision    recall  f1-score   support

                Baby Care       0.81      0.87      0.84        30
  Beauty and Personal Care       0.93      0.87      0.90        30
                Computers       1.00      1.00      1.00        30
Home Decor & Festive Needs       0.81      0.87      0.84        30
           Home Furnishing       0.87      0.90      0.89        30
          Kitchen & Dining       0.96      0.87      0.91        30
                   Watches       1.00      1.00      1.00        30

                  accuracy                           0.91       210
                 macro avg       0.91      0.91      0.91       210
              weighted avg       0.91      0.91      0.91       210
```

DistilBERT

Avec les embeddings DistilBERT, la régression logistique atteint une accuracy de 0,98 sur l'ensemble d'entraînement et de 0,91 sur l'ensemble de test, ce qui indique une bonne capacité de généralisation.

Le modèle obtient d'excellents résultats sur les catégories « Computers » et « Watches » (f1-score de 1,00), et des scores également élevés pour les autres classes, avec une macro F1-score globale de 0,91 sur le test.

Ces performances montrent que DistilBERT permet de bien distinguer la plupart des catégories de produits.

```
Accuracy sur l'ensemble d'entraînement : 0.6798
Accuracy sur l'ensemble de test : 0.5524

Rapport de classification (test) :
                           precision    recall  f1-score   support

                Baby Care       0.38      0.50      0.43        30
  Beauty and Personal Care       0.94      0.50      0.65        30
                Computers       0.58      0.47      0.52        30
Home Decor & Festive Needs       0.32      0.40      0.36        30
           Home Furnishing       0.71      0.57      0.63        30
          Kitchen & Dining       0.40      0.57      0.47        30
                   Watches       0.96      0.87      0.91        30

                  accuracy                           0.55       210
                 macro avg       0.61      0.55      0.57       210
              weighted avg       0.61      0.55      0.57       210
```

DeBERTa

Avec les embeddings DeBERTa, la régression logistique atteint une accuracy de 0,68 sur l'ensemble d'entraînement et de 0,55 sur l'ensemble de test, ce qui montre une performance nettement inférieure à celle obtenue avec DistilBERT sur ce jeu de données.

Le modèle obtient de bons résultats sur la catégorie « Watches » (f1-score de 0,91), mais les scores sont plus faibles pour les autres classes, avec une macro F1-score de 0,57 sur le test.

Ces résultats indiquent que, dans ce contexte, DeBERTa distingue moins bien les catégories de produits que DistilBERT.

5.4) Analyse de la feature importance globale et locale

Pour mieux comprendre comment le modèle DeBERTa utilise ses embeddings dans la classification, nous analysons l'importance des différentes dimensions des vecteurs à deux niveaux : global, pour identifier les dimensions les plus influentes sur l'ensemble des données, et local, pour expliquer les prédictions sur des exemples individuels.

Cette analyse, réalisée via la permutation feature importance et les valeurs SHAP, permet de vérifier que le modèle s'appuie sur des signaux pertinents et d'améliorer la transparence des décisions.

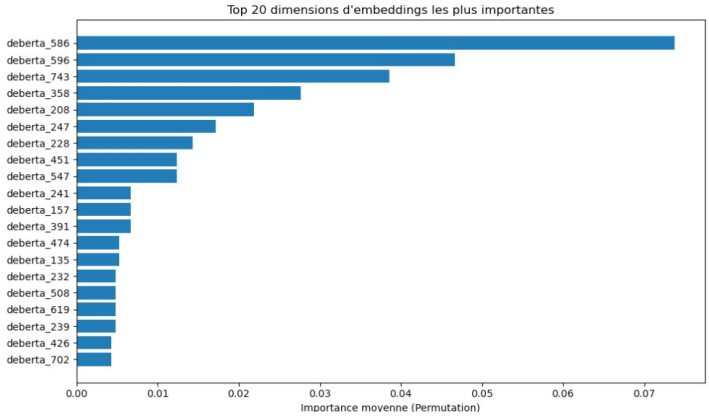
Feature importance globale:

Pour évaluer l'importance de chaque dimension des embeddings DeBERTa, nous utilisons la méthode de permutation importance. Cette approche consiste à perturber, une à une, les colonnes du jeu de test : pour chaque dimension, on mélange aléatoirement ses valeurs entre les exemples, tout en laissant les autres dimensions inchangées. On mesure alors la baisse de performance du modèle (ici l'accuracy).

Si la permutation d'une dimension entraîne une chute notable des performances, cela signifie que le modèle s'appuyait fortement sur cette dimension pour ses prédictions. À l'inverse, si la

permutation n'a pas d'effet, la dimension n'apporte pas d'information utile à la décision.

Cette méthode, automatisée dans scikit-learn, permet ainsi d'identifier quelles composantes de l'espace d'embedding sont réellement exploitées par le modèle, et d'interpréter de façon quantitative la contribution de chaque feature à la classification.



L'analyse de la permutation importance appliquée aux embeddings DeBERTa montre que seules quelques dimensions présentent une importance notable pour la classification, mais leur contribution reste modérée.

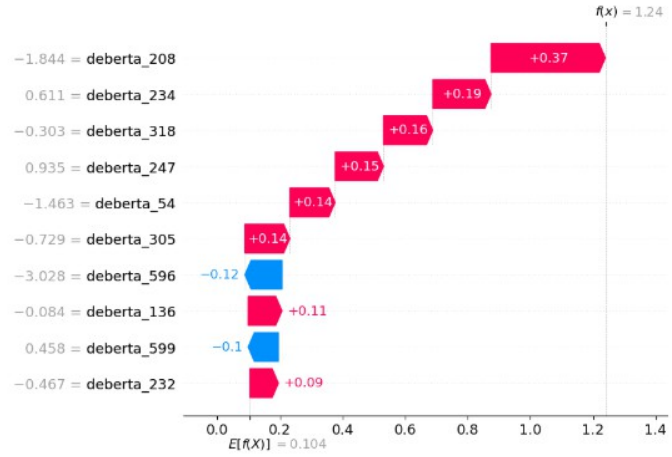
La majorité des dimensions ont une importance très faible, ce qui indique que l'information utile est répartie sur plusieurs composantes : le modèle s'appuie sur une combinaison de signaux faibles plutôt que sur une seule dimension dominante.

Cela suggère qu'une réduction de la dimensionnalité pourrait être envisagée, et confirme que le modèle exploite des signaux pertinents pour ses prédictions.

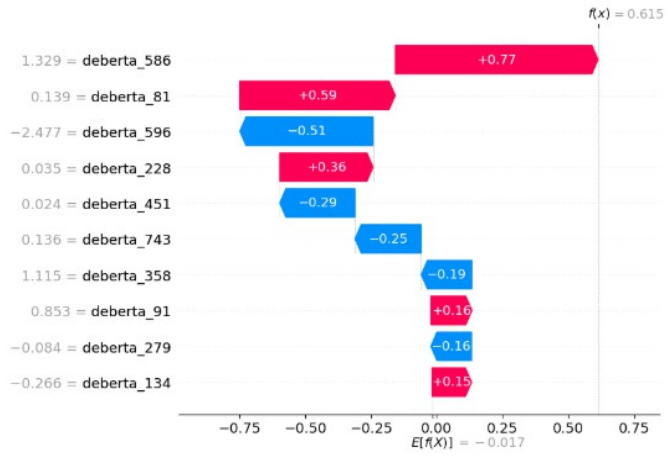
Feature importance locale:

L'analyse locale vise à expliquer, pour chaque prédiction individuelle, quelles dimensions de l'embedding ou quels éléments du texte ont le plus influencé la décision du modèle. Cette approche permet d'ouvrir la « boîte noire » du modèle au niveau d'un exemple précis, en identifiant les facteurs déterminants pour une classification donnée.

```
======== Explication locale de la prédiction ========
Texte analysé :
Key Features of Elegance Polyester Multicolor Abstract Eyelet Door Curtain Floral Curtain,Elegance Polyester Multicolor Abstrac
t Eyelet Door Curtain (213 cm in Height, Pack of 2) Price: Rs. 899 This curtain enhances the look of the interiors.This curtain
is made from 100% high quality polyester fabric.It features an eyelet style stitch with Metal Ring.It makes the room environmen
t romantic and lov...
--------------------------------------------------
Classe réelle : Home Furnishing
Classe prédite : Baby Care
--------------------------------------------------
Top 10 dimensions de l'embedding DeBERTa les plus influentes :
  - deberta_208: 0.365
  - deberta_234: 0.186
  - deberta_318: 0.157
  - deberta_247: 0.155
  - deberta_54: 0.144
  - deberta_305: 0.144
  - deberta_596: -0.119
  - deberta_136: 0.108
  - deberta_599: -0.097
  - deberta_232: 0.092
==================================================
```




```
======== Explication locale de la prédiction ========
Texte analysé :
Key Features of Eurospa Cotton Terry Face Towel Set Size: small Height: 9 inch GSM: 360,Eurospa Cotton Terry Face Towel Set (20
PIECE FACE TOWEL SET, Assorted) Price: Rs. 299 Eurospa brings to you an exclusively designed, 100% soft cotton towels of export
quality. All our products have soft texture that takes care of your skin and gives you that enriched feeling you deserve. Euros
pa has been expor...
--------------------------------------------------
Classe réelle : Baby Care
Classe prédite : Kitchen & Dining
--------------------------------------------------
Top 10 dimensions de l'embedding DeBERTa les plus influentes :
  - deberta_586: 0.771
  - deberta_81: 0.591
  - deberta_596: -0.507
  - deberta_228: 0.355
  - deberta_451: -0.286
  - deberta_743: -0.252
  - deberta_358: -0.194
  - deberta_91: 0.157
  - deberta_279: -0.156
  - deberta_134: 0.150
==================================================
```



Même en cas d'erreur de classification, on observe que certaines dimensions exercent un impact

marqué sur la sortie du modèle, ce qui montre que la décision n'est pas prise au hasard mais repose sur des signaux appris dans l'espace d'embedding. Les dimensions les plus influentes varient selon les textes, suggérant qu'elles capturent des thématiques ou motifs spécifiques. Enfin, on constate que plusieurs dimensions importantes localement correspondent aussi à celles identifiées dans l'analyse globale, ce qui renforce la cohérence de l'interprétation du modèle. Ces analyses mettent également en évidence que certaines erreurs proviennent de proximités sémantiques ou d'ambiguïtés dans les données, plutôt que d'un comportement aléatoire du modèle.

6) Conclusion

Ce POC a permis d'évaluer l'efficacité des embeddings issus de modèles pré-entraînés DistilBERT et DeBERTa pour la classification automatique de descriptions produits. Les résultats montrent que DistilBERT fournit de bonnes performances, avec une capacité satisfaisante à généraliser sur les données de test, tandis que DeBERTa, dans la configuration actuelle, présente des performances plus faibles.

L'analyse de l'importance des features, à la fois globale et locale, a permis de mieux comprendre comment le modèle DeBERTA exploite l'information textuelle et a confirmé que les décisions reposent sur des signaux pertinents, même lorsque des erreurs sont commises. Cette compréhension ouvre la voie à des améliorations ciblées.

7) Limites et améliorations possibles

Limites de l'approche actuelle

- Interprétabilité limitée des embeddings : Les dimensions des vecteurs DeBERTa n'ont pas de signification humaine directe, rendant l'analyse fine des décisions du modèle difficile, même avec des outils d'explicabilité.
- Performance perfectible : Les résultats obtenus avec DeBERTa sont inférieurs à ceux de BERT, ce qui peut s'expliquer par la méthode d'extraction des embeddings ou par le manque d'adaptation du modèle au domaine.
- Erreurs dues à la proximité sémantique : Certaines erreurs de classification proviennent de descriptions de produits ambiguës ou très proches entre catégories, limitant la capacité de discrimination du modèle.
- Limites du classificateur linéaire : La régression logistique ne capture pas les relations non linéaires potentielles entre les dimensions des embeddings.
- Peu de prise en compte du contexte métier :L'approche ne mobilise pas d'informations additionnelles (métadonnées, hiérarchie produit, etc.) qui pourraient enrichir la représentation.

Améliorations envisageables

- Changement de stratégie d'agrégation des embeddings : Tester la moyenne des embeddings de tous les tokens (mean pooling) plutôt que l'embedding [CLS] de DeBERTa, afin de mieux représenter l'information globale du texte et d'améliorer la qualité des features utilisées pour la classification. Comme le modèle n'a pas été fine-tuné spécifiquement pour la tâche de classification, la moyenne des embeddings est souvent plus efficace que l'embedding [CLS], car elle exploite l'information de l'ensemble des tokens plutôt que de s'appuyer sur un résumé qui n'a pas été optimisé pour notre cas d'usage.
- Réduction de la dimensionnalité : Appliquer des techniques comme PCA ou UMAP pour ne

conserver que les dimensions les plus informatives, ce qui peut simplifier le modèle et faciliter l'interprétation.

- Utilisation de modèles de classification plus complexes : Expérimenter avec des modèles non linéaires (Random Forest, SVM, réseaux de neurones) pour mieux exploiter la structure des embeddings.
- Affinage des embeddings (fine-tuning) : Adapter DeBERTa au corpus cible via un entraînement complémentaire pour obtenir des représentations plus pertinentes pour la tâche.
- Enrichissement et nettoyage des données : Ajouter des métadonnées produits, utiliser des descriptions plus détaillées ou nettoyer les données pour réduire les ambiguïtés et améliorer la qualité des entrées.
- Exploration d'autres méthodes d'interprétabilité : Tester des outils comme LIME ou des méthodes d'attention pour relier plus directement les décisions du modèle aux éléments du texte.
- Analyse d'erreurs approfondie : Étudier systématiquement les cas de confusion entre catégories pour mieux comprendre les limites du modèle et guider les améliorations futures.

8) Bibliographie

- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805. https://arxiv.org/abs/1810.04805
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv preprint arXiv:1908.10084. https://arxiv.org/abs/1908.10084
- Wang, C., Li, M., & Li, D. (2021). BERT Goes Shopping: Comparing Distributional Models for Product Representations. Proceedings of the 4th Workshop on e-Commerce and NLP (ECNLP 4), ACL Anthology. https://aclanthology.org/2021.ecnlp-1.1.pdf
- He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. arXiv preprint arXiv:2006.03654. https://arxiv.org/abs/2006.03654
- Papers With Code. (2024). DeBERTa Explained. https://paperswithcode.com/method/deberta
- He, Y., et al. (2021). Improving DeBERTa using ELECTRA-style pre-training with gradient penalty. arXiv preprint arXiv:2111.09543. https://arxiv.org/pdf/2111.09543.pdf