**Note méthodologique : preuve de concept**

# **Dataset retenu**

Le dataset utilisé pour ce projet se concentre sur des données e-commerces liés à des produits de consommation variés. Ce dataset inclut plusieurs caractéristiques pertinentes pour la classification de produits, notamment les colonnes suivantes :

* main\_category : la catégorie principale du produit (ex : "Home Decor", "Baby Care").
* product\_name : le nom du produit.
* brand : la marque associée au produit.
* cleaned\_description : une description nettoyée et prétraitée pour être utilisée dans les modèles NLP.

*Le dataset compte environ 1 050 échantillons.*

# **Les concepts de l’algorithme récent**

DeBERTa (Decoding-enhanced BERT with disentangled attention) est une amélioration de BERT et RoBERTa. Son architecture introduit deux concepts clés :

Mécanisme d’attention désassemblé : Contrairement à BERT où chaque mot est représenté par un seul vecteur, DeBERTa représente chaque mot avec deux vecteurs distincts : un vecteur de contenu (content embedding) et un vecteur de position (position embedding). Le vecteur de contenu capture la signification sémantique du mot tandis que le vecteur de position encode la position relative du mot dans la phrase. Cette approche permet au modèle d'apprendre des relations plus fines entre les mots et leur contexte, ce qui améliore la compréhension des relations syntaxiques et sémantiques complexes.

Enhanced Mask Decoder (EMD) : DeBERTa introduit un décodeur de masque amélioré (EMD) pour traiter les positions absolues. Contrairement à l’attention relative classique, l’EMD permet d’intégrer des informations de position absolue lors de la décodification des masques, ce qui renforce la capacité à comprendre les relations entre les mots distants dans le texte.

# **La modélisation**

La modélisation s'appuie sur deux approches principales pour l'analyse des descriptions des produits e-commerce : une approche classique basée sur TF-IDF et une approche avancée utilisant le modèle DeBERTa. Ces deux approches permettent de comparer les performances de méthodes traditionnelles et modernes de traitement du langage naturel (NLP).

**Approche Classique (TF-IDF + K-Means)**

L'approche classique repose sur l'utilisation de la méthode TF-IDF combinée à un clustering K-means. Cette méthodologie se décompose en plusieurs étapes clés :

Standardisation des Données : Les colonnes numériques et la matrice TF-IDF des descriptions de produits sont combinées en un seul ensemble de données. Pour assurer l'homogénéité des échelles, les données sont standardisées à l'aide de la méthode StandardScaler(), ce qui normalise les valeurs afin qu'elles aient une moyenne de 0 et un écart-type de 1.

Réduction de Dimension : Pour réduire la complexité computationnelle et faciliter la visualisation, la méthode de l'Analyse en Composantes Principales (PCA - Principal Component Analysis) est appliquée. Cette méthode réduit les dimensions tout en préservant l'essentiel de la variance des données. L'espace d'origine est projeté sur un sous-espace de 50 dimensions, ce qui permet de conserver les informations les plus significatives.

T-SNE (t-Distributed Stochastic Neighbor Embedding) : Afin de visualiser les relations entre les produits, la méthode T-SNE est appliquée pour réduire l'espace des 50 dimensions issues de la PCA à seulement 2 dimensions. Contrairement à la PCA, qui cherche à préserver la variance, T-SNE se concentre sur la préservation des relations de proximité locale, ce qui le rend particulièrement adapté pour la visualisation des données. Le résultat est une représentation visuelle permettant de mieux comprendre la distribution et la séparation des catégories.

Clustering avec K-means : L'algorithme K-means est utilisé pour segmenter les produits en groupes ou "clusters" similaires. En utilisant les représentations 2D obtenues via T-SNE, K-means forme des clusters distincts de produits. Le nombre de clusters est déterminé à l'avance (par exemple, 7 clusters dans notre cas). Chaque produit est alors affecté à un cluster spécifique, ce qui permet d'analyser et de visualiser les similitudes au sein du dataset.

**Approche Avancée (DeBERTa)**

La modélisation avec DeBERTa commence par la préparation des données, où la colonne cleaned\_description est nettoyée et transformée pour créer un texte d'entrée sous la forme "classify: [texte]". Cette étape permet de formater les descriptions pour qu'elles soient compatibles avec le modèle DeBERTa. Chaque description est ensuite tokenisée à l'aide du tokenizer DeBERTa, générant trois tenseurs : input\_ids, attention\_mask et les labels. Ces tenseurs servent d'entrée au modèle pour l'entraînement.

Le modèle DeBERTa (Disentangled BERT) est chargé depuis la bibliothèque HuggingFace Transformers, configuré pour la classification multi-classe avec 7 labels, correspondant aux 7 catégories de produits identifiées. Le modèle est affiné (fine-tuned) sur les descriptions de produits, en utilisant un algorithme d'optimisation AdamW avec un taux d'apprentissage de 2e-5. L'entraînement s'effectue sur 3 époques et un suivi du temps d'exécution est mis en place. Ce suivi permet de mesurer la performance temporelle de chaque itération d'entraînement et de suivre l'évolution de la perte à chaque époque.

Une fois le modèle DeBERTa entraîné, les embeddings (vecteurs de caractéristiques) sont extraits de la dernière couche du modèle. Ces embeddings, qui capturent les relations sémantiques et contextuelles des descriptions, sont ensuite convertis en DataFrame pour faciliter leur manipulation et leur analyse.

Les embeddings générés sont ensuite utilisés pour des analyses de clustering. Une réduction de dimension est effectuée via T-SNE pour visualiser les embeddings dans un espace bidimensionnel. Cela permet de visualiser la répartition des produits selon leurs relations sémantiques. Deux visualisations distinctes sont produites :

* Visualisation des vraies classes : Chaque produit est coloré en fonction de sa vraie catégorie.
* Visualisation des clusters K-means : Les embeddings sont utilisés pour appliquer l'algorithme de K-means afin de créer des groupes similaires. Les points sont colorés selon les clusters K-means, ce qui permet de comparer la cohérence des clusters avec les catégories réelles.

**Métriques utilisées : Précision, Rappel et Score F1.**

Ces trois métriques permettent d'évaluer la performance des modèles sur la tâche de classification multi-classe.

* Précision : Mesure la proportion de prédictions correctes parmi les prédictions positives effectuées.
* Rappel : Évalue la capacité du modèle à identifier toutes les instances positives dans les données.
* Score F1 : C'est la moyenne harmonique de la précision et du rappel, ce qui permet de trouver un équilibre entre les faux positifs et les faux négatifs.

# **Une synthèse des résultats**

Les deux approches ont été évaluées selon plusieurs métriques, notamment l'ARI (Adjusted Rand Index), la précision, le rappel, le F1-score, et le temps d'exécution.

|  |  |  |
| --- | --- | --- |
| Métrique | TF-IDF | DeBERTa |
| ARI | 0.22 | 0.86 |
| Précision moyenne | 47 % | 94 % |
| Rappel moyen | 50 % | 94 % |
| F1-score moyen | 48 % | 94 % |
| Temps d'exécution (sec) | 6.28 | 4161.68 |

**Analyse de TF-IDF :**

La méthode TF-IDF combinée à K-Means montre une séparation des données correcte, mais reste globalement imprécise lorsqu'il s'agit de refléter les catégories réelles. Le faible score ARI met en évidence une correspondance limitée entre les clusters générés par K-Means et les classes attendues. Par ailleurs, la matrice de confusion révèle une confusion marquée entre plusieurs catégories, indiquant un manque de précision dans le regroupement. Cependant, cette méthode présente un avantage significatif : son temps d'exécution rapide, ce qui en fait une option adaptée aux scénarios nécessitant des résultats rapides.

**Analyse de DeBERTa :**

La méthode DeBERTa surpasse largement l'approche classique dans toutes les métriques d'évaluation. Les clusters formés après application du T-SNE montrent une séparation plus nette et plus cohérente avec les catégories réelles. La matrice de confusion souligne une classification extrêmement précise avec très peu d'erreurs. Toutefois, cette précision accrue a un coût : un temps d'exécution nettement plus élevé pour l'entraînement du modèle, ce qui peut représenter une contrainte dans des contextes où des mises à jour fréquentes ou une exécution rapide sont nécessaires.

**Conclusion :**

DeBERTa surpasse de loin l'approche classique TF-IDF en termes de performance globale, comme le démontrent les métriques ARI, précision, rappel, et F1-score. Cependant, cette supériorité se fait au prix d'un temps d'exécution beaucoup plus élevé pour l'entraînement. Pour des applications nécessitant une classification très précise et justifiable, DeBERTa est recommandé. En revanche, pour des tâches nécessitant une réponse rapide et légère, l'approche TF-IDF peut être privilégiée.

# **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.*

# **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.*