# PROJET N°8
# NOTE METHODOLOGIQUE

## POC : Proof Of Concept


### Résumé

Ce document a pour vocation de résumer, sous forme de synthèse et note méthodologique, un Proof of Concept, élaborée sur un notebook, mettant en œuvre une nouvelle technique de « Natural Language Processing », NLP.



Justin Vacquerie

OpenClassRooms – 23/09/2024

# Table des matières

I. Les données & la veille technique ..... 2
A. Le base de données retenue ..... 2
B. Veille technique & POC ..... 2
II. Les techniques de NLP ..... 3
A. Modèle classique avec le Word2Vec ..... 3
B. Approche récente avec le MiniLM (version 2) ..... 3
1) Présentation du modèle ..... 3
2) Détails techniques ..... 4
III. La modélisation ..... 5
A. Les deux approches ..... 5
1) Classique avec le Word2Vec ..... 5
2) Récente avec le MiniLMV2 ..... 5
B. Les métriques de performances ..... 6
1) Le score de silhouette ..... 6
2) Le score Rand Ajusté ..... 6
3) L'exactitude ..... 6
IV. Les résultats de performances ..... 7
A. Comparaison sur projection TSNE ..... 7
B. Comparaison pour la modélisation non-supervisée ..... 8
C. Comparaison pour la modélisation supervisée ..... 9
D. Analyse de l'importance globale et locale ..... 10
1) Analyse globale ..... 10
2) Analyse locale ..... 10
V. Conclusion sur le POC ..... 11
VI. Limites et améliorations possibles ..... 11
A. Les limites ..... 11
B. Pour aller plus loin ..... 11
VII. Bibliographie ..... 12

# I. Les données & la veille technique

## A. Le base de données retenue

Pour cette étude comparative, nous nous appuyons sur un projet précédent dans lequel nous avions développé un moteur de classification automatique pour une marketplace e-commerce. Ce moteur visait à catégoriser les produits en fonction de leurs descriptions textuelles, en utilisant des techniques avancées de traitement du langage naturel (NLP).

Nous allons ainsi réutiliser ce dataset pour ce « Proof of Concept », où nous comparerons deux techniques de NLP, dans un contexte de veille sur les dernières avancées dans ce domaine.

Le jeu de données utilisé provient de la marketplace "Place de marché". Il contient les informations suivantes :

- Identifiant unique du produit
- Timestamp de collecte des données
- URL du produit
- Nom du produit
- Catégorie du produit
- Prix de vente conseillé
- Prix après remise
- Image du produit
- Description du produit
- Note du produit
- Marque et spécifications du produit

Le dataset comprend 1 050 lignes et 15 colonnes, avec un taux de valeurs manquantes de 2,17 %.

Pour ce proof of concept (POC), seules les colonnes **product_category** et **description**, qui ne contiennent aucune valeur manquante, seront utilisées.

## B. Veille technique & POC

Dans le cadre de cette veille technique sur le traitement automatique du langage (NLP), je réalise cette étude comparative entre l'évolution des modèles d'apprentissage de représentations textuelles. Cette étude vise à évaluer les performances de Word2Vec, un modèle classique d'embeddings de mots, par rapport à MiniLM, un modèle plus récent, compact et performant pour générer des représentations textuelles.

Un Proof of Concept (POC) est ici un bon exercice, pour comparer ces deux approches et ainsi, valider leur efficacité dans des tâches de classification de textes. Un POC permet de tester la faisabilité d'une solution dans un contexte donné, tout en limitant les risques avant une implémentation plus large par exemple. Il s'inscrit dans une démarche de veille technique, un processus qui consiste à identifier et évaluer les innovations pertinentes.

En comparant MiniLM et Word2Vec, mon objectif est de déterminer, entre les deux techniques de NLP, laquelle présente les meilleures performances, pour classifier des objets en fonction de leurs descriptions textuelles.

# II. Les techniques de NLP

Dans cette partie, nous allons introduire les deux modèles utilisés pour ce POC. En l'occurrence, le Word2Vec, notre technique de référence, où je ferai une brève description de son historique et mode de fonctionnement. A l'inverse, je développerai, un peu plus en détails, le modèle MiniLM qui sera notre approche récente, à évaluer et à tester.

## A. Modèle classique avec le Word2Vec

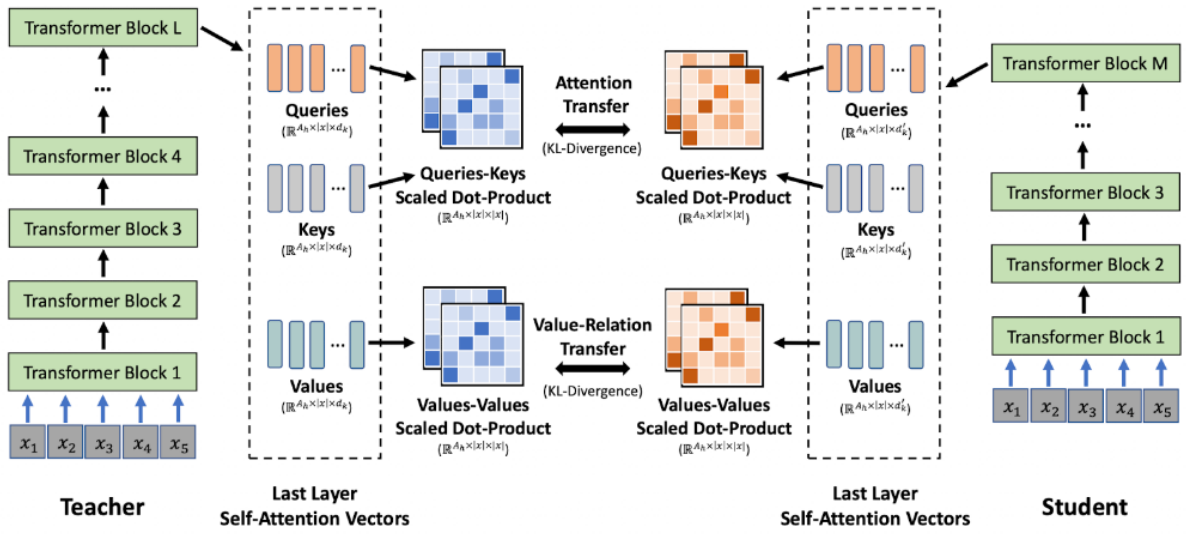
Word2Vec est une technique de modélisation de langage et de représentation vectorielle des mots, développée par une équipe de recherche chez Google sous la direction de Tomas Mikolov en 2013. Elle transforme les mots en vecteurs de nombres réels dans un espace vectoriel de haute dimension, permettant de capturer des relations sémantiques et syntaxiques entre les mots. Word2Vec utilise deux architectures principales : CBOW (Continuous Bag of Words) et Skip-gram. CBOW prédit un mot en fonction d'un contexte, tandis que Skip-gram fait l'inverse en prédisant le contexte à partir d'un mot. L'approche est fondée sur la notion que les mots apparaissant dans des contextes similaires possèdent des significations proches. Les vecteurs résultants facilitent diverses tâches de traitement automatique des langues, telles que la traduction automatique, la reconnaissance vocale et la recherche sémantique.

## B. Approche récente avec le MiniLM (version 2)

### 1) Présentation du modèle

MiniLM est une série de modèles de langage reposant sur l'architecture Transformer, optimisés pour être plus compacts et efficaces que leurs prédécesseurs tels que BERT. Ils exploitent la distillation de connaissances, une méthode où un modèle plus volumineux (le modèle enseignant) transfère son savoir à un modèle plus léger (le modèle étudiant). Cette approche permet de diminuer la taille du modèle tout en préservant d'excellentes capacités sur diverses tâches de traitement du langage naturel (NLP).

<u>Schéma de fonctionnement du modèle :</u>



**Explication du schéma :**

Ce graphique représente un processus de distillation des connaissances entre un modèle enseignant (Teacher) et un modèle étudiant (Student), tous deux basés sur des blocs de transformeurs. L'enseignant a un modèle plus grand (L blocs), tandis que l'étudiant est plus petit (M blocs). La distillation se fait en transférant des informations sur l'attention (Queries-Keys et Values-Values) du modèle enseignant vers l'étudiant en minimisant la divergence de Kullback-Leibler (KL) entre les matrices d'attention des deux modèles. Cela permet à l'étudiant de s'aligner sur les performances du modèle enseignant tout en étant plus léger.

**Le modèle « all-MiniLM-L6-v2 »** : Ce modèle étudiant est spécialement adapté pour transformer des phrases et des paragraphes en vecteurs denses de 384 dimensions. Idéal pour le clustering, la recherche sémantique et l'évaluation de la similarité de phrases, all-MiniLM-L6-v2 excelle grâce à son entraînement sur un vaste corpus de paires de phrases, capturant ainsi des nuances sémantiques complexes et offrant une compréhension textuelle approfondie.

## 2) Détails techniques

**Pré-entraînement** : Basé sur MiniLM-L6-H384-uncased, all-MiniLM-L6-v2 utilise une version allégée de l'architecture Transformer pour générer des embeddings de phrases.

**Fine-tuning** : Le fine-tuning est effectué via un objectif d'apprentissage contrastif, employant la similarité cosinus pour évaluer la proximité entre les paires de phrases et utilisant la perte de cross-entropie pour l'ajustement. Ce processus est optimisé sur plus d'un milliard de paires de phrases avec des TPU v3-8 pour maximiser l'efficacité et la rapidité.

**Corpus d'entraînement** : Le modèle bénéficie d'une formation sur un ensemble diversifié de données incluant des commentaires de Reddit, des citations de S2ORC, des doublons de questions de WikiAnswers, et d'autres, totalisant plus d'un milliard de paires.

**Usage** : Conçu pour coder efficacement les phrases et petits paragraphes en vecteurs sémantiques, all-MiniLM-L6-v2 se prête bien aux applications nécessitant une extraction d'information, du clustering, ou une évaluation de similarité.

En résumé, le Word2Vec génère des vecteurs de mots statiques, ce qui signifie que chaque mot a une seule représentation vectorielle, indépendamment du contexte dans lequel il est utilisé. En revanche, MiniLM (v2 améliorée) produit des embeddings contextuels, où la représentation vectorielle d'un mot peut varier en fonction de son contexte dans une phrase, capturant ainsi des nuances de sens plus complexes et des relations syntaxiques plus fines.

# III. La modélisation

## A. Les deux approches

### 1) Classique avec le Word2Vec

1. Nettoyage des données textuelles :
   - Tokenization
   - Suppression des stop words
   - Lemmatisation
   - Suppression des mots avec une fréquence non significative

2. Embedding
   - Utilisation de Word2Vec pour vectoriser les descriptions de produits.

3. Réduction des données :
   - ACP : pour réduire le bruit dans les données (99% de variance conservée)
   - T-SNE : notamment pour la projection des embeddings et la visualisation

4. Modélisation
   - Modélisation non-supervisée avec l'utilisation du K-means pour faire de la segmentation
   - Modélisation supervisée avec une Logistic Regression multiclasse pour prédire nos catégories

### 2) Récente avec le MiniLMV2

Approche récente :

1. Embedding
   - Utilisation de all-MiniLM-L6-v2 pour vectoriser les descriptions de produits.

2. Réduction des données :
   - T-SNE : notamment pour la projection des embeddings et la visualisation

3. Modélisation
   - Modélisation non-supervisée avec l'utilisation du K-means pour faire de la segmentation
   - Modélisation supervisée avec une Logistic Regression multiclasse pour prédire nos catégories

Le modèle MiniLM, comme d'autres modèles de type Transformer, ne nécessite pas de prétraitement manuel tel que la lemmatisation ou le stemming parce qu'il utilise un tokenizer intégré pour transformer le texte brut en tokens directement exploitables.

Contrairement à Word2Vec, qui dépend d'un prétraitement préalable pour créer des vecteurs, MiniLM apprend les relations contextuelles à partir du texte brut dans sa globalité, ce qui rend les étapes de nettoyage moins nécessaires.

## B. Les métriques de performances

Les métriques de performances sont centrales dans notre POC, en effet ce sont les différents scores obtenus qui nous permettront de juger et comparer les approches sur une base commune et objective.

### 1) Le score de silhouette

**Définition** : Le Score de Silhouette évalue l'efficacité de la séparation entre les clusters. Il se calcule à partir de la moyenne des distances à l'intérieur d'un cluster (distance entre un point et les autres points du même cluster) et des distances entre différents clusters (distance entre un point et les points des autres clusters).

**Interprétation** : Un Score de Silhouette élevé suggère une bonne séparation et regroupement des points, indiquant des clusters bien délimités et compacts. Un score proche de 1 implique une bonne attribution des points aux clusters, tandis qu'un score proche de -1 révèle des clusters mal formés.

**Utilité** : Ce score permet d'évaluer l'efficacité du clustering, notamment pour vérifier si les embeddings obtenus par MiniLM sont nettement distincts de ceux obtenus par le Word2Vec.

### 2) Le score Rand Ajusté

**Définition :** Le Score Rand Ajusté (Adjusted Rand Score - ARS) mesure la similarité entre deux clusters en tenant compte du nombre de clusters et des chances de correspondance aléatoire. Il évalue la concordance des points assignés dans les clusters prédits par rapport aux vrais clusters.

**Interprétation :** Un ARS de 1 signale une correspondance parfaite entre les clusters prédits et réels, un score de 0 une correspondance aléatoire, et un score négatif une discordance.

**Importance :** Le Score Rand Ajusté est crucial pour mesurer la précision avec laquelle l'algorithme de clustering reproduit les vraies catégories des données, utile pour juger la pertinence des embeddings produits par le MiniLM comparativement au Word2Vec

### 3) L'exactitude

**Définition :** L'exactitude représente le pourcentage de prédictions correctes sur le total des prédictions. Pour le clustering, elle peut être estimée en associant chaque cluster à la classe la plus représentée de ses membres, et en comparant ensuite ces attributions aux étiquettes réelles.

**Interprétation :** Une accuracy élevée indique que la majorité des points sont bien attribués à leurs clusters. Toutefois, cette mesure nécessite souvent de réajuster les étiquettes des clusters aux catégories réelles pour une évaluation précise.

**Pourquoi cette mesure :** L'accuracy offre une compréhension simple et directe de la performance de l'algorithme, facilitant la comparaison entre les capacités de MiniLM et de Word2Vec pour classifier correctement les catégories de produits.

Ces trois indicateurs nous fournissent une évaluation complète des techniques de modélisation :

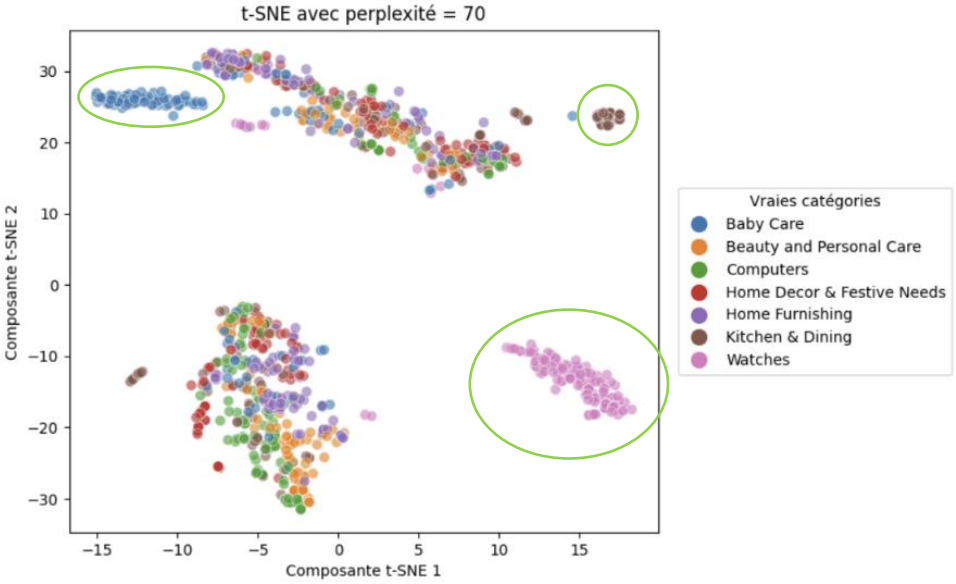
- Le Score de Silhouette renseigne sur la délimitation et la compacité des clusters.
- Le Score Rand Ajusté reflète la concordance entre les clusters formés et les catégories réelles.
- L'Accuracy évalue l'exactitude globale des affectations de cluster.

Ensemble, ces mesures offrent une perspective détaillée sur la qualité du clustering et la performance des modèles, permettant une comparaison efficace entre MiniLM et le Word2Vec.

# IV. Les résultats de performances

## A. Comparaison sur projection TSNE

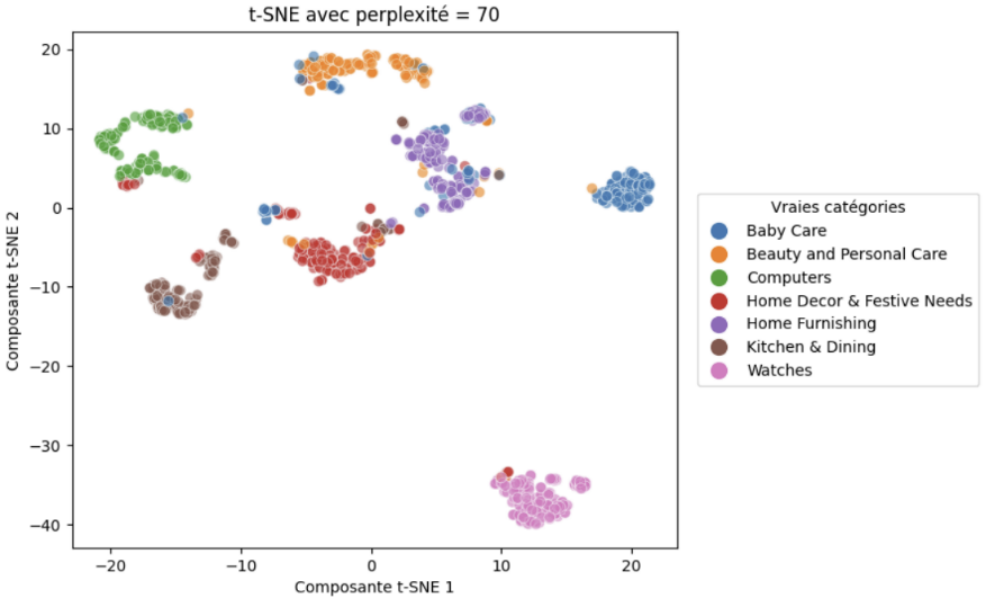
Dans cette première partie, nous allons comparer la projection des embeddings, des deux techniques dans un espace à deux dimensions avec le T-SNE.




Le premier graphique présente les embeddings issu du modèle, Word2Vec.

Si, on commence par le positif, on remarque 3 grappes uniformes, entourées en vert. Notamment, celle pour la catégorie des montres, rose et celle en bleu pour la catégorie BabyCare.

Toutefois, la présence de deux grappes d'une taille importante, complètement hétérogènes, montre que la proximité sémantique des descriptions des produits n'est pas retrouvée car trop subtiles pour le Word2Vec.




Dans le graphique suivant, ce sont les embeddings issus du modèle MiniLM.

On constate ici déjà une bonne segmentation des points. La projection permet de montrer les relations locales, ici sémantiques. Et nous pouvons discerner des grappes de points bien formées et uniformes.

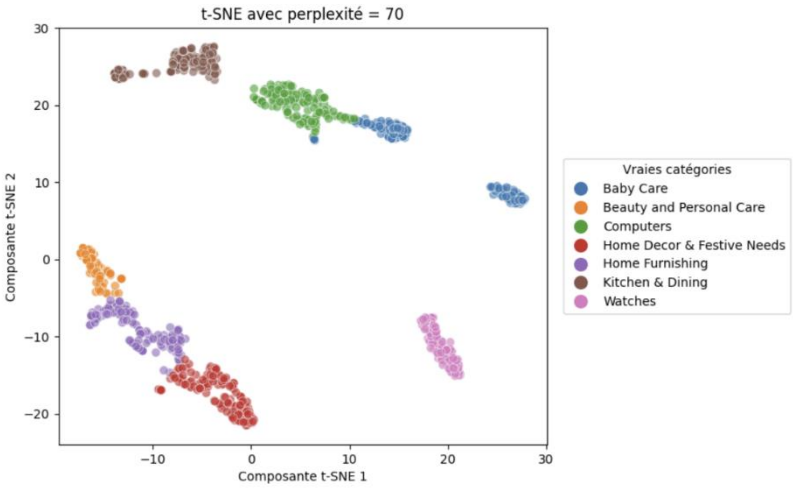
Les embeddings du MiniLM sont donc plus pertinents et performants où nous voyons que les catégories se regroupent assez bien. Malgré, encore quelques points assez éloignés encore de leur centroïde, même si ce niveau de l'étude le k-means n'a pas encore été appliqué.

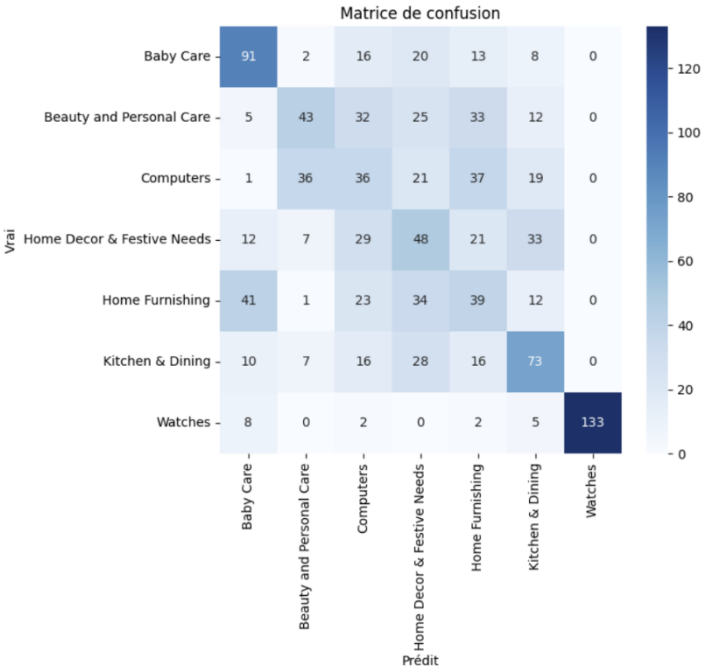
## B. Comparaison pour la modélisation non-supervisée

Cette partie est dédiée à présenter les différents résultats obtenus au cours de notre démarche de modélisation en non-supervisée avec le k-means.

**Le Word2Vec :**

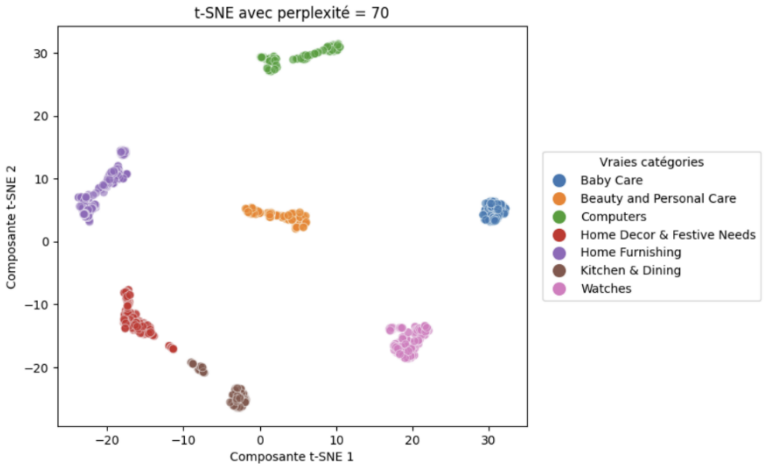
- Silhouette Score: **0.5229**
- Adjusted Rand Score: **0.2153**
- Accuracy: **0.4410**

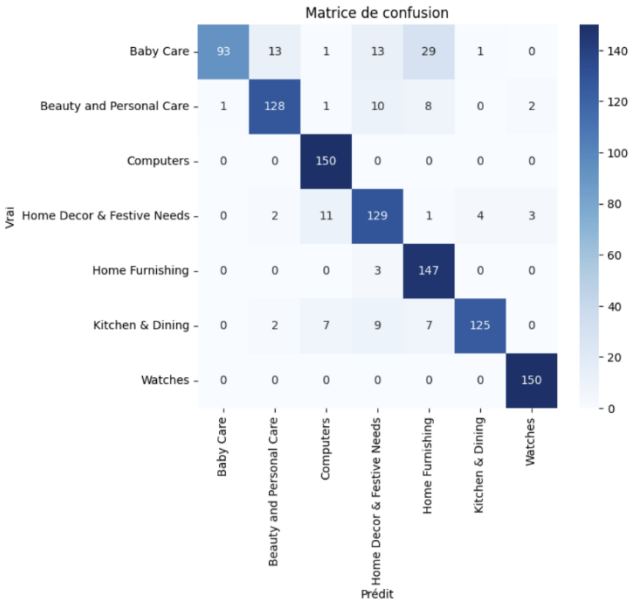



**Le Mini LM :**

- Silhouette Score : **0.6928**
- Adjusted Rand Score : **0.7443**
- Exactitude : **0.8781**




Les résultats montrent que MiniLM dépasse le Word2Vec
sur toutes les métriques évaluées signe d'une meilleure performance l'analyse sémantique du corpus.

Le Silhouette Score plus élevé avec MiniLM indique des clusters mieux définis et séparés. L'Adjusted Rand Score et l'Accuracy plus élevés démontrent une meilleure concordance entre les clusters prédits et les clusters réels, ainsi qu'une plus grande précision du modèle.

## C. Comparaison pour la modélisation supervisée

Enfin, dernier point de comparaison entre nos deux techniques de NLP, avec la modélisation supervisée. Je rappelle que pour l'ensemble des entrainements, il n'y a pas eu de recherche d'optimisation des hyperparamètres, comme nous le faisons souvent dans une approche model-centric. Ici, le classifieur, « Logistic Regression », a été entrainé dans son utilisation « standard ».

**Pour le Word2Vec** : une accuracy globale de **75%**

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Home Furnishing | 0.94 | 0.56 | 0.70 | 27 |
| Baby Care | 0.55 | 0.81 | 0.65 | 21 |
| Watches | 0.86 | 0.79 | 0.82 | 38 |
| Home Decor & Festive Needs | 0.62 | 0.70 | 0.66 | 30 |
| Kitchen & Dining | 0.71 | 0.71 | 0.71 | 35 |
| Beauty and Personal Care | 0.62 | 0.62 | 0.62 | 26 |
| Computers | 1.00 | 1.00 | 1.00 | 33 |
| accuracy | | | 0.75 | 210 |
| macro avg | 0.76 | 0.74 | 0.74 | 210 |
| weighted avg | 0.77 | 0.75 | 0.75 | 210 |

**Word2Vec** : accuracy globale de 75%. Les performances varient fortement selon les classes, avec un bon score pour "Computers" (1.00) mais des résultats moyens pour d'autres comme "Home Furnishing" (0.56 en rappel).

**Pour le MiniLM** : une accuracy globale de **95%**

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Baby Care | 0.88 | 0.78 | 0.82 | 27 |
| Beauty and Personal Care | 0.95 | 0.95 | 0.95 | 21 |
| Computers | 1.00 | 1.00 | 1.00 | 38 |
| Home Decor & Festive Needs | 0.91 | 1.00 | 0.95 | 30 |
| Home Furnishing | 0.92 | 0.94 | 0.93 | 35 |
| Kitchen & Dining | 1.00 | 0.96 | 0.98 | 26 |
| Watches | 1.00 | 1.00 | 1.00 | 33 |
| accuracy | | | 0.95 | 210 |
| macro avg | 0.95 | 0.95 | 0.95 | 210 |
| weighted avg | 0.95 | 0.95 | 0.95 | 210 |

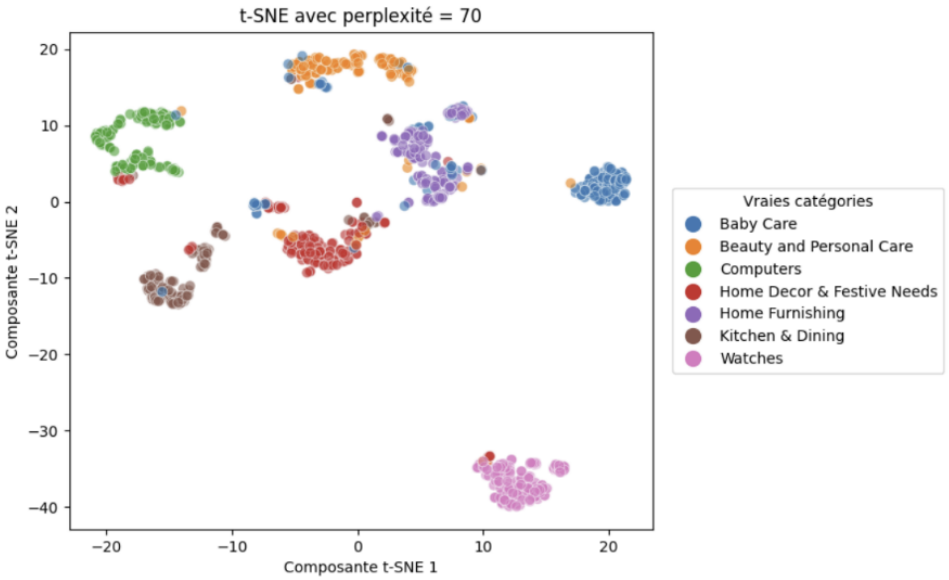
**MiniLM** : accuracy globale de 95%. Il obtient des scores presque parfaits dans toutes les classes, avec une précision et un rappel bien plus constant, démontrant une meilleure capacité à capturer les représentations textuelles et classifier les objets.

Entre les deux modèles, **MiniLM** se distingue par une précision et un rappel beaucoup plus constant et élevé (95% d'accuracy globale), tandis que **Word2Vec** présente des résultats plus variables (75% d'accuracy), avec des performances inégales selon les classes. MiniLM offre donc une meilleure fiabilité pour cette tâche de classification.

## D. Analyse de l'importance globale et locale

Le graphique, ci-dessous, représente une visualisation t-SNE des embeddings générés par MiniLM, comme nous avons pu le voir dans la partie précédente. Et pour rappel, chaque point correspond à un échantillon du jeu de données, coloré selon sa catégorie réelle.

Cette visualisation en 2D permet d'observer comment les embeddings regroupent les données et de détecter des clusters ou des relations entre les catégories.




### 1) Analyse globale

Le t-SNE projette les embeddings dans un espace 2D tout en préservant les relations locales, ce qui permet d'observer des regroupements de points correspondant à des similarités dans l'espace d'origine à haute dimension. Cela offre une vue d'ensemble des relations entre les catégories du jeu de données.

- **Clustering distinct** : Certaines catégories, comme Baby Care (bleu) et Watches (rose clair), forment des clusters bien séparés, indiquant que les embeddings capturent des caractéristiques propres à ces catégories. Les points sont compacts, traduisant une cohérence sémantique.
- **Chevauchement des catégories** : Certaines catégories, comme Home Decor & Festive Needs (rouge) et Kitchen & Dining (marron), sont plus mélangées, suggérant des caractéristiques partagées et rendant la séparation plus complexe.
- **Distances entre les clusters** : Les catégories éloignées, comme Computers (vert) et Watches (rose clair), montrent une distinction sémantique forte, confirmant que les embeddings reflètent bien ces différences.

### 2) Analyse locale

Au niveau local, chaque point représente un échantillon avec des embeddings similaires. L'analyse consiste à comprendre pourquoi des échantillons sont regroupés ou proches dans cet espace 2D.

- **Proximité locale** : Lorsque des points de catégories différentes sont proches, cela peut indiquer des similarités textuelles dans les descriptions. Par exemple, un produit Home Furnishing pourrait avoir une description semblable à un produit Kitchen & Dining si des termes communs liés à la maison sont utilisés.
- **Différences locales** : Un point isolé ou en périphérie d'un cluster peut indiquer que l'échantillon présente des caractéristiques uniques, faisant de lui un outlier ou une exception à la catégorie générale.

# V. Conclusion sur le POC

Word2Vec et MiniLM sont deux techniques de génération d'embeddings pour représenter des mots, mais avec des différences importantes. Word2Vec génère des embeddings statiques, c'est-à-dire que chaque mot a une seule représentation, peu importe le contexte. En utilisant Word2Vec avec K-Means en non-supervisé, on peut obtenir des clusters de mots similaires, mais cela reste limité par le manque d'information contextuelle.

MiniLM, en revanche, génère des embeddings contextuels, ce qui signifie que le même mot peut avoir une représentation différente selon la phrase. Cela améliore grandement les résultats pour des tâches comme la classification supervisée avec une régression logistique, car MiniLM prend mieux en compte les nuances du texte. En conséquence, MiniLM surpasse Word2Vec, offrant de meilleures performances, que ce soit en non-supervisé pour des clusters plus cohérents ou en supervisé pour une classification plus précise.

# VI. Limites et améliorations possibles

## A. Les limites

- **Temps de calcul élevé avec MiniLM** : Lorsqu'on travaille avec de grands ensembles de données, générer des embeddings avec MiniLM peut prendre beaucoup de temps, car ce modèle, bien que plus léger que d'autres Transformers, reste plus complexe que des techniques comme Word2Vec. Ce temps de traitement peut devenir un frein pour des applications à grande échelle.
- **Perte d'informations avec t-SNE** : La réduction de dimension via t-SNE, bien qu'efficace pour la visualisation, peut parfois entraîner une perte de certaines informations importantes dans les embeddings. En effet, lors de la réduction à 2 ou 3 dimensions, les relations contextuelles entre les mots ou phrases peuvent être dégradées, ce qui complique l'interprétation des résultats.
- **Interprétation des clusters** : L'interprétabilité des clusters générés reste un défi. Il est difficile d'attribuer une signification claire à chaque dimension des embeddings, car les représentations produites par des modèles comme MiniLM sont hautement abstraites et multidimensionnelles, rendant l'analyse des groupes de données plus complexe.

## B. Pour aller plus loin

- **Techniques de réduction de dimension avancées** : Pour pallier la perte d'informations observée avec t-SNE, il serait utile d'explorer des techniques de réduction de dimension plus robustes, comme l'Umap ou des méthodes basées sur des auto-encodeurs, qui pourraient mieux conserver les relations contextuelles entre les données tout en réduisant la dimensionnalité.
- **Optimisation du temps de calcul** : Il pourrait être intéressant d'explorer d'autres variantes plus légères de Transformers ou des techniques d'optimisation, comme la quantification de modèle ou la distillation, pour réduire le temps de calcul tout en préservant des performances comparables à MiniLM.
- **Feature importance plus poussé :** Utilisation de librairies comme Shap ou Lime

# VII. Bibliographie

https://mohitmayank.com/a_lazy_data_science_guide/natural_language_processing/minilm/

https://huggingface.co/microsoft/MiniLM-L12-H384-uncased

https://arxiv.org/abs/2002.10957

https://github.com/microsoft/unilm/tree/master/minilm

https://arxiv.org/abs/2012.15828

# VIII. Annexes

## 1) Schéma de la démarche employée lors de ce POC