### 1. **Initialisation (méthode** \_\_init\_\_**)**

* **Modèle SpaCy :** Le code tente de charger un modèle préexistant de SpaCy pour la langue française (fr\_core\_news\_md). Si ce modèle n'est pas trouvé, il le télécharge automatiquement. Ce modèle est utilisé pour le traitement du texte, comme l'analyse syntaxique.
* **Pipeline d'analyse de sentiment :** Utilise un modèle de transformers pour analyser les sentiments des textes. Il utilise un modèle multilingue (nlptown/bert-base-multilingual-uncased-sentiment).
* **SentenceTransformer :** Un modèle pour générer des embeddings de phrases à l'aide de distiluse-base-multilingual-cased. Cela permet de convertir des textes en vecteurs de caractéristiques pour des comparaisons sémantiques.
* **Taxonomie du luxe :** Une taxonomie définie dans un dictionnaire Python qui regroupe différents termes associés aux catégories du luxe, comme les matériaux nobles, le savoir-faire, le patrimoine, etc.

### 2. **Extraction de texte depuis des PDFs (méthode** extract\_text\_from\_pdfs**)**

* Cette méthode prend en entrée un dictionnaire de fichiers PDF et utilise la bibliothèque **PyPDF2** pour extraire le texte de chaque page. Les textes sont ensuite prétraités pour supprimer les espaces excessifs et les sauts de ligne inutiles.

### 3. **Prétraitement du texte (méthode** preprocess\_text**)**

* Le texte extrait est converti en minuscules et analysé à l'aide de SpaCy pour enlever les **mots vides** (stop words) et la **ponctuation**, et pour garder uniquement les **tokens pertinents**.

### 4. **Extraction de la fourchette de prix (méthode** extract\_price\_range**)**

* Cette méthode utilise des expressions régulières pour extraire les prix dans les textes. Elle filtre ensuite les prix qui sont réalistes (entre 100 et 10 000 €) et calcule des statistiques comme :
  + Le prix minimum, maximum, moyen et médian
  + La distribution des prix dans différentes catégories : entrée de gamme, milieu de gamme, haut de gamme

### 5. **Analyse des matériaux (méthode** analyze\_materials**)**

* La méthode parcourt les textes extraits et cherche des mots-clés associés aux **matériaux nobles** (comme le cuir, le veau, le métal, etc.) définis dans la taxonomie du luxe. Elle utilise SpaCy pour analyser le texte et cherche ces termes dans chaque token (mots) extrait.

### 6. **Extraction des caractéristiques clés (méthode** extract\_key\_features**)**

* Cette méthode utilise des expressions régulières pour extraire des phrases contenant des mots-clés comme **"signature"**, **"exclusif"**, **"édition"**, **"design"**, etc., qui sont souvent associés aux éléments distinctifs des produits de luxe. Ces phrases sont extraites et filtrées selon leur longueur (entre 10 et 150 caractères).

### 7. **Utilisation du modèle de sentiment (non montré ici mais probablement utilisé ailleurs)**

* Le modèle de sentiment (self.sentiment\_analyzer) est probablement utilisé pour analyser l'attitude (positive, négative, neutre) des textes. Cependant, cette partie n'est pas explicitement utilisée dans ce code. Le modèle nlptown/bert-base-multilingual-uncased-sentiment permet d'analyser les sentiments en fonction des tokens du texte.

### Résumé de la fonctionnalité :

* Le **BrandDNAAnalyzer** permet d'analyser des documents, principalement des fichiers PDF, pour extraire des informations sur les produits de luxe. Cela inclut :
  + Des informations sur les matériaux utilisés.
  + L'extraction de certaines caractéristiques clés des produits (par exemple, des éléments de design distinctifs).
  + L'analyse des prix des produits.
  + La classification des prix en fonction des segments (entrée de gamme, milieu de gamme, etc.).

### Quelques points à noter :

* **SpaCy** est utilisé pour le traitement linguistique (tokenisation, lemmatisation, etc.), tandis que **transformers** est utilisé pour l'analyse des sentiments et les embeddings de phrases.
* La taxonomie du luxe permet de repérer les termes associés à chaque aspect de l'ADN d'une marque de luxe (matériaux, savoir-faire, etc.).
*  **Dédoublonnage et tri par pertinence** : La fonction analyze\_luxury\_markers prend un texte et identifie les marqueurs de luxe selon une taxonomie prédéfinie. Elle renvoie les termes trouvés dans le texte ainsi que leur contexte, en supprimant les doublons pour ne conserver que les termes uniques.
*  **Analyse des marqueurs de luxe** : La méthode analyze\_luxury\_markers parcourt chaque terme de la taxonomie du luxe et extrait des contextes autour des termes correspondants dans le texte (par exemple, "exclusif", "précieux", etc.). Ces termes sont regroupés par catégories (ex : histoire, savoir-faire, etc.).
*  **Contexte autour des mots-clés** : La fonction get\_context extrait un extrait de texte autour d’un mot-clé, sur un "window" de 50 caractères avant et après le mot trouvé. Cela permet de mieux comprendre l’usage du mot dans son contexte.
*  **Analyse des codes visuels distinctifs** : La fonction analyze\_visual\_codes cherche des termes spécifiques aux codes visuels comme des motifs (par exemple, "damier", "rayures") et les associe à un contexte. Ces codes visuels font partie de l'identité d'une marque de luxe.
*  **Analyse de l'innovation** : L'analyse de l'innovation se concentre sur des aspects comme la technologie, les matériaux, les processus et l'expérience de la marque. Chaque terme trouvé dans ces catégories augmente un score d'innovation, et les détails sont retournés.
*  **Analyse de l'héritage et des traditions** : Cette fonction analyse des termes relatifs à l’histoire et à l'héritage de la marque (ex : "artisanat", "tradition"). Elle attribue également un score d’héritage et renvoie les catégories avec leurs termes.
*  **Analyse complète de l'ADN de la marque** : La méthode analyze\_brand\_dna effectue une analyse complète selon plusieurs critères (luxe, visuel, innovation, héritage, etc.). Elle génère un rapport détaillé avec des scores et des listes de marqueurs trouvés dans les documents, et peut également calculer la similarité entre documents.
*  **Matrice de similarité entre les documents** : L'analyse de similarité utilise des embeddings (représentations vectorielles de texte) pour calculer la similarité entre les textes. Ces résultats sont affichés sous forme de matrice et peuvent être visualisés à l’aide d’un graphique de type heatmap (carte thermique).
*  **Rapport final** : La méthode generate\_dna\_report génère un rapport structuré sur l'ADN de la marque, incluant les marqueurs de luxe, les codes visuels, l'innovation, l'héritage, le positionnement prix, et une analyse des sentiments associés.

Le script permet de :

1. **Analyser l'ADN des produits** en extrayant du texte à partir de fichiers PDF et en générant des résumés.
2. **Analyser des documents longs** pour identifier des similitudes et extraire des informations essentielles.
3. **Effectuer des recherches sémantiques** à l'aide d'un index FAISS et de modèles d'embedding.
4.  faiss : Permet d'effectuer des recherches de similarité ultra-rapides.
5.  sentence-transformers : Utilisé pour transformer les textes en embeddings vectoriels.
6.  transformers : Utilisé pour le résumé de texte avec facebook/bart-large-cnn.
7.  google.colab : Permet d'accéder à Google Drive.
8.  rouge\_score et nltk : Évaluent la qualité des résumés générés.
9.  PyPDF2 : Permet d'extraire le texte des fichiers PDF.
10.  matplotlib et seaborn (mentionnés dans le code) : Utilisés pour visualiser les similarités entre documents.

**⚙ 3. Initialisation (\_\_init\_\_ method)**

Lors de l'initialisation :

1. Monte Google Drive pour stocker les documents et vecteurs.
2. Crée des dossiers (LV\_DNA\_System/documents, vectors, summaries).
3. Charge :
   * Un modèle d'embedding (distiluse-base-multilingual-cased).
   * Un modèle de résumé (facebook/bart-large-cnn).
   * Un index FAISS pour stocker les embeddings vectoriels.

**📑 4. Analyse ADN des Produits (analyze\_product\_dna)**

1. **Upload de fichiers PDF** via files.upload().
2. **Extraction des textes** avec un analyseur de marque externe (BrandDNAAnalyzer).
3. **Calcul des similarités** entre les documents et affichage d’une **matrice de similarité**.
4. **Génération d’embeddings** avec SentenceTransformer, puis stockage dans FAISS.
5. **Sauvegarde des résumés et métadonnées** sur Google Drive.

**5. Analyse des Documents Longs (analyze\_documents)**

1. **Extraction et découpage** des textes en segments (chunk\_text).
2. **Résumé automatique** des segments avec self.summarizer.
3. **Évaluation de la qualité** des résumés avec les scores ROUGE et BLEU.
4. **Sauvegarde des résultats** sous forme de fichiers JSON.

**6. Recherche Sémantique (semantic\_search)**

1. **L’utilisateur saisit une requête**.
2. La requête est convertie en **vecteur d’embedding**.
3. **Recherche dans l’index FAISS** pour récupérer les 5 documents les plus similaires.
4. **Affichage des résultats** avec des scores de similarité.

**🛠 7. Utilitaires**

* **extract\_text** : Extrait le texte d’un fichier PDF.
* **chunk\_text** : Découpe un texte en morceaux de 1000 caractères.
* **generate\_summary** : Utilise BART pour générer un résumé.
* **evaluate\_summary** : Évalue la qualité du résumé avec les scores ROUGE et BLEU.

**8. Exécution Principale (main)**

Le programme propose un **menu interactif** :

1. **Analyser l'ADN de fiches produits**.
2. **Analyser l'ADN de documents longs**.
3. **Effectuer une recherche sémantique**.
4. **Quitter**.

### **1. Bibliothèques et outils NLP (Traitement du Langage Naturel) :**

* **spaCy** → Pour le traitement du texte (tokenisation, lemmatisation, analyse syntaxique).
* **Hugging Face Transformers** (si utilisé) → Pour l'analyse des sentiments et l'extraction d'entités.
* **Sentence Transformers** → Pour la création d'embeddings de texte et l'analyse de similarité.

### **2. Algorithmes et méthodes de traitement :**

* **Détection de marqueurs de luxe** → Recherche de termes spécifiques dans le texte (par catégorie).
* **Extraction de contexte** → Identification des phrases autour des mots-clés.
* **Analyse de similarité** → Utilisation de **cosine similarity** pour comparer les textes.
* **Analyse de l'ADN créatif** → Catégorisation des éléments liés au design, à l'innovation et au storytelling.

### **3. Techniques de machine learning :**

* **Embeddings** → Création de représentations vectorielles des textes avec un modèle de sentence embedding.
* **Clustering et classification** → Pour identifier les thèmes et caractéristiques récurrents.

### **4. Manipulation des données :**

* **re (expressions régulières)** → Nettoyage et extraction des informations des textes.
* **datetime** → Pour ajouter des métadonnées comme la date de l'analyse.