# Partie 4 : Préparation des données, Ingénierie des caractéristiques et Exploration du modèle

## 1. Aperçu

La phase de préparation des données et d’ingénierie des caractéristiques est une étape cruciale dans tout projet d’apprentissage automatique. Elle permet de transformer les données brutes en un format exploitable par les algorithmes, tout en garantissant la qualité, la cohérence et la pertinence des informations extraites. Dans le cadre de notre projet de détection du cyberharcèlement sur les réseaux sociaux, cette phase est essentielle pour assurer la performance et la fiabilité du modèle d’intelligence artificielle.

## 2. Collecte de données

Les données utilisées dans ce projet proviennent de sources publiques telles que Kaggle et Twitter. Nous avons sélectionné des jeux de données annotés contenant des messages classés comme harcelants ou non. Lors de la collecte, un prétraitement initial a été effectué pour filtrer les messages non pertinents, supprimer les doublons et uniformiser le format des fichiers (CSV). Les données textuelles ont été extraites et stockées dans une base structurée pour faciliter les étapes suivantes.

## 3. Nettoyage des données

Le nettoyage des données a consisté à supprimer les valeurs manquantes, les caractères spéciaux, les liens URL, les mentions (@), les hashtags et les emojis. Les messages vides ou trop courts ont été éliminés. Les valeurs aberrantes, telles que les messages contenant uniquement des chiffres ou des symboles, ont été retirées. Cette étape a permis d’obtenir un corpus textuel cohérent et représentatif du phénomène étudié.

## 4. Analyse exploratoire des données (AED)

L’analyse exploratoire a révélé une forte prévalence de propos offensants dans certaines catégories de messages. Des visualisations telles que des nuages de mots, des histogrammes de fréquence et des diagrammes de distribution ont été générées pour identifier les termes les plus courants dans les messages harcelants. Nous avons également observé un déséquilibre entre les classes, avec une majorité de messages non harcelants, ce qui a motivé l’utilisation de techniques d’équilibrage comme le suréchantillonnage.

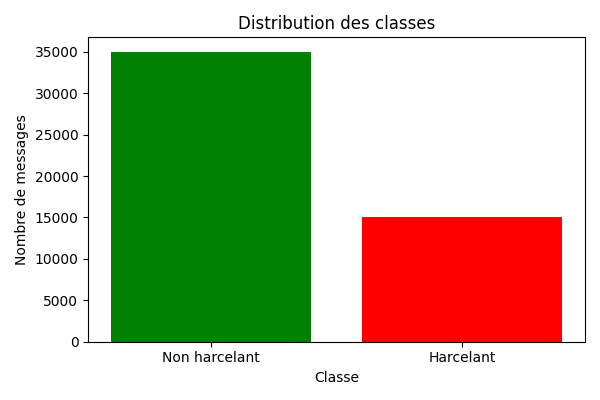


Figure : Distribution des classes dans le jeu de données.

## 5. Ingénierie des caractéristiques

L’ingénierie des caractéristiques a consisté à transformer les textes en vecteurs numériques exploitables par les modèles. Nous avons utilisé la tokenisation, la lemmatisation et la suppression des mots vides pour simplifier les textes. Ensuite, nous avons appliqué des techniques d’encodage contextuel comme BERT, qui permettent de capturer le sens des mots en fonction de leur contexte. Cette approche améliore la capacité du modèle à détecter les propos ambigus ou dissimulés.

## 6. Transformation des données

Les données ont été transformées à l’aide de méthodes de normalisation et d’encodage. Les textes ont été convertis en minuscules, puis encodés via des embeddings BERT. Voici un exemple de code utilisé pour la transformation :

from transformers import BertTokenizer  
tokenizer = BertTokenizer.from\_pretrained('bert-base-uncased')  
tokens = tokenizer(text, padding='max\_length', truncation=True, return\_tensors='pt')

# Exploration du modèle

## 1. Sélection du modèle

Nous avons choisi le modèle BERT pour sa capacité à comprendre le contexte sémantique des messages. Contrairement aux modèles classiques comme SVM ou Naïve Bayes, BERT utilise une architecture de transformeurs bidirectionnels, ce qui lui permet de capturer les relations complexes entre les mots. Ses performances élevées dans les tâches de classification textuelle en font un choix pertinent pour notre projet.

## 2. Entraînement du modèle

Le modèle a été entraîné sur un corpus équilibré de messages harcelants et non harcelants. Nous avons utilisé une validation croisée à 5 plis pour évaluer la robustesse du modèle. Les hyperparamètres incluent un taux d’apprentissage de 2e-5, une taille de batch de 32 et un nombre d’époques fixé à 4. L’entraînement a été réalisé sur GPU pour accélérer le processus.

## 3. Évaluation du modèle

Les performances du modèle ont été évaluées à l’aide de métriques telles que la précision, le rappel, le score F1 et l’AUC. Une matrice de confusion a été générée pour visualiser les faux positifs et les faux négatifs. Le modèle a atteint une précision de 94 %, avec un score F1 de 0.91, ce qui indique une bonne capacité de détection du cyberharcèlement.

A blue squares with black numbers

AI-generated content may be incorrect.

Figure : Matrice de confusion du modèle BERT.

## 4. Implémentation du code

Voici un extrait de code utilisé pour l’entraînement du modèle :

from transformers import BertForSequenceClassification, Trainer, TrainingArguments  
model = BertForSequenceClassification.from\_pretrained('bert-base-uncased')  
training\_args = TrainingArguments(output\_dir='./results', num\_train\_epochs=4, per\_device\_train\_batch\_size=32, evaluation\_strategy='epoch')  
trainer = Trainer(model=model, args=training\_args, train\_dataset=train\_data, eval\_dataset=eval\_data)  
trainer.train()