Classe : CI1
Matière : MACHINE LEARNING

Enseignant : Y.Haddad

Mini-Projet

A. Objectif du projet

Le but de ce projet est de **détecter automatiquement si un texte (tweet en anglais)** est **sarcastique ou non**, en utilisant des techniques de **Machine Learning (ML)**. Chaque groupe travaillera sur **le même jeu de données textuelles**, et devra concevoir un **système de classification** capable de reconnaître le sarcasme à partir du contenu des messages.

B. Jeu de données : *iSarcasmEval_En*

Le dataset choisi est **iSarcasmEval_En**, disponible sur GitHub :
🔗 https://github.com/iabufarha/iSarcasmEval

Deux fichiers vous seront fournis :

- *train.En.csv* → utilisé pour **entraîner et valider** vos modèles
- *task_A_En_test.csv* → utilisé pour **tester** vos modèles finaux

Contenu typique des colonnes:

Colonne	Description	Utilisation
tweet	**Texte du tweet à analyser**	**Variable d'entrée principale**
sarcastic	**1 = sarcastique, 0 = non sarcastique**	**Variable cible (label à prédire)**
rephrase	Reformulation du tweet sans sarcasme	Optionnelle (pour comparaison ou prétraitement)
sarcasm, **irony**, **satire**, **understatement**, **overstatement**, **rhetorical_question**	Indiquent la présence d'autres formes d'humour ou de style	Variables optionnelles pour analyse avancée
Unnamed: 0	Index automatique	À ignorer

Travailler sur les colonnes *tweet* et *sarcastic*. Les autres colonnes peuvent servir pour une analyse avancée. Vérifier les déséquilibres de classes (plus de tweets sarcastiques ou non sarcastiques).

C. Étapes de travail à réaliser

1. Exploration et Analyse des Données (EAD)

- Charger le fichier *train.En.csv* avec **pandas**
- Analyser les colonnes disponibles (*.info(), .head(), .describe()*)
- Vérifier la distribution des classes (label)
- Mesurer la longueur des textes (nombre de mots, caractères)
- Visualiser les données :
    - Histogrammes (répartition des classes, longueur des textes)
    - WordCloud (mots fréquents dans les tweets sarcastiques vs non sarcastiques)

2. Prétraitement du texte

Avant d'entraîner les modèles, il faut **nettoyer et transformer** les textes :

- Convertir en minuscules
- Supprimer la ponctuation, les liens, les hashtags et les mentions (@user)
- Supprimer les **stopwords** (mots vides comme "the", "is", "and"…)
- Optionnel : lemmatisation (réduction des mots à leur forme de base)
- Créer une colonne *text_clean* avec le texte nettoyé

3. Vectorisation (transformation en nombres)

Les algorithmes de ML ne comprennent que des nombres. Il faut donc convertir les textes en vecteurs numériques avec :

- **TF-IDF Vectorizer** (classique et efficace)
- ou **Bag of Words**
- ou **Word Embeddings (BERT, Word2Vec)** pour les plus avancés

4. Application d'au moins 5 algorithmes de Machine Learning

Chaque groupe doit tester **au moins cinq modèles** différents parmi les suivants :

Tester au moins **5 modèles différents** :

- Régression Logistique
- SVM (Support Vector Machine)
- Random Forest
- Naive Bayes
- K-Nearest Neighbors (KNN)
- (Optionnel : modèles avancés comme LSTM ou BERT)

Pour chaque modèle:

- Diviser les données en **train/test** (*par exemple 80% / 20%)*
- Entraîner le modèle
- Évaluer les performances : **accuracy**, **precision**, **recall**, **F1-score**
- Afficher la **matrice de confusion**

5. Comparaison et sélection du meilleur modèle

- Comparer la précision, le rappel, le F1-score et l'AUC des modèles.
- Présenter les résultats dans un tableau comparatif.

6. Test final et sauvegarde

- Appliquer le modèle choisi sur le fichier *task_A_En_test.csv*
- Sauvegarder le modèle entraîné et les prédictions finales.

D. Résultats attendus

Chaque groupe doit :

- Présenter le notebook Jupyter (.ipynb) avec toutes les étapes clairement commentées
- Fournir les visualisations (histogrammes, WordClouds, matrice de confusion)
- Fournir un **tableau comparatif** des modèles testés
- Identifier et justifier **le modèle le plus performant**
- (Optionnel) Ajouter une petite discussion sur les erreurs fréquentes ou les limites du modèle.

E. Structure de la présentation

1. **Introduction** : présentation du problème et du dataset
2. **Analyse exploratoire (EAD)** : observations, graphiques
3. **Prétraitement** : nettoyage du texte
4. **Vectorisation et Modélisation** : choix et entraînement des modèles
5. **Évaluation et comparaison** : résultats, tableau des performances
6. **Conclusion** : modèle final et perspectives

F. Livrables à remettre

1. Le notebook (.ipynb) du travail complet
2. La version HTML du notebook (générée via cet outil : **https://htmtopdf.herokuapp.com/ipynbviewer/)**