

## Mini-projet en IA2

### Exercice 1 : classification de texte

Soit les commentaires sur les films stockés dans le répertoire *data*. Chaque commentaire est mis sur un fichier (lire un exemple). Le sous-répertoire *pos* contient les fichiers des commentaires positifs et le *neg* contient les commentaires négatifs. Le but de cet exercice est d'implémenter un classificateur positif/négatif des commentaires en texte.

1. Charger les fichiers depuis le répertoire *data* en utilisant la méthode *load\_files()* du package *sklearn.datasets*.
2. Diviser les données en train et test avec une répartition de 70-30%.
3. Implémenter le classificateur de Naïf-Bayes
4. Evaluer le modèle et afficher le rapport d'évaluation pour chaque modèle.
5. Importer Keras et créer un réseau
  - `from keras.models import Sequential`
  - `from keras import layers`
  - `model = Sequential()`
  - `model.add(layers.Dense(?, input_dim=??, activation='relu'))`
  - `model.add(layers.Dense(?, activation='sigmoid'))`
6. compiler le modèle
  - `model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])`
  - `model.summary()`
7. entrainer le model (exemple de paramètres)
  - `history = model.fit(X_train, y_train, epochs=100, verbose=False, validation_data=(X_test, y_test), batch_size=10)`
8. évaluer le modèle (
  - a. `loss, accuracy = model.evaluate` (pour le train ensuite pour le test)
9. plotter le loss et l'accuracy du history
10. Faire un tableau pour comparer les résultats

### Exercice 2 : Classification des mobiles

Le but de ce classificateur est de prédire les classes de prix (*price\_range*) (0, 1, 2, 3) des téléphones portables à partir de leurs caractéristiques techniques (puissance de la batterie, couleurs, mémoire, etc.).

1. Lire les données du répertoire *data* dans un dataframe Pandas et afficher les premiers 10 lignes des données de *train*.
2. Enlever depuis les données de *train* les lignes où la colonne *px\_height* est nulle.
3. Implémenter un pipeline SVM avec un *scaler* de votre choix.
4. Afficher le rapport d'évaluation après l'évaluation des données de test.
5. Implémenter un arbre de décision puis un Forest Tree (from `sklearn.ensemble` import `RandomForestClassifier`).
6. Implémenter un réseau de neurone.
7. Faire un tableau pour comparer les résultats.