# Théorie bayésienne de la décision - L3 Info SD

# TP - feuille n°5 - Visualisation et prédiction sur des données à 2 caractéristiques

## 1 Présentation

Le but de ce TP est de visualiser un jeu de données d'entraînement, puis de proposer une frontière de décision et de la tester sur un jeu de données de validation.
Ces jeux contiennent des données suivant les mêmes distributions :

- avec 2 caractéristiques réelles `x1` et `x2` (2 premières colonnes)
- réparties en 2 modalités (0 ou 1) désignées par la classe `y` (3ème colonne)

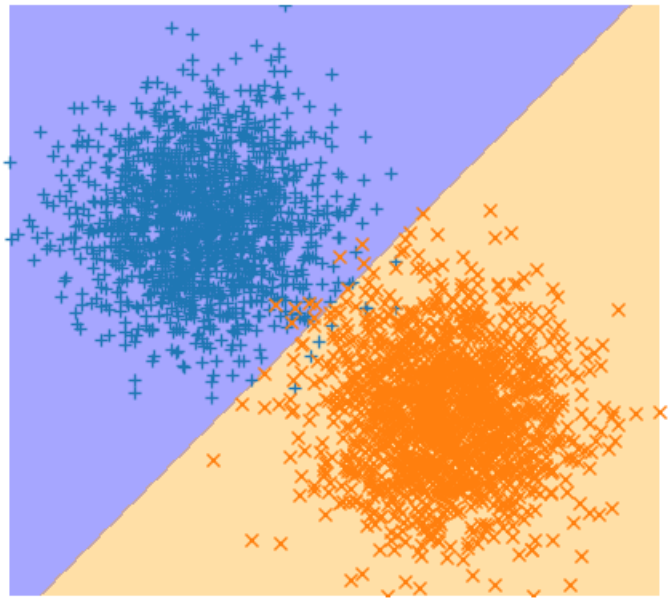
## 2 Affichage des données d'entraînement

1) Ouvrez les données d'entraînement contenues dans le fichier `tp5_data1_train.txt` dans une DataFrame Panda et créez :
   - une matrice `X_train` avec les caractéristiques `x1` et `x2`
   - un vecteur `y_train` avec les classes `y`
2) Utilisez Matplotlib pour afficher ces données sous la forme de deux nuages de points :
   - avec des `+` pour la classe 0
   - avec des `×` pour la classe 1

   Remarque : on utilisera `plt.axis("equal")` pour obtenir la même échelle sur les deux axes.

## 3 Recherche d'une frontière de décision

Dans cette partie, on souhaite créer une fonction de prédiction et visualiser la frontière de décision que cette fonction dessine sur le plan. On pourra obtenir par exemple l'image suivante, sur laquelle la partie bleue du plan (partie supérieure gauche) est classée 0 et la partie orange (partie inférieure droite) est classée 1 :



Remarque : sur cette image, la frontière de décision est une droite, mais on peut afficher n'importe quel type de frontière, comme vous le verrez ultérieurement.

1) Proposez une fonction `prediction(...)` qui :
   - prend en paramètre un vecteur `x` de caractéristiques
   - retourne une classe prédite
2) Utilisez la méthode `plot_decision(...)` fournie dans le fichier `utils.py` pour afficher la frontière de décision et les nuages de points sur un même graphique.

## 4 Test de la fonction de prédiction

1) Créez une matrice `X_valid` et un vecteur `y_valid` à partir des données de validation contenues dans le fichier `tp5_data1_valid.txt`.
2) Créez le vecteur `y_pred` des classes prédites sur ces données de validation.
3) Construisez la matrice de confusion et le taux d'erreur de votre modèle.

## 5 Deuxième jeu de données

Mêmes questions avec `tp5_data2_train.txt` et `tp5_data2_valid.txt`.