Théorie bayésienne de la décision - L3 Info SD

TP - feuille n°1 - Création d'un modèle et test

1 Utilisation de Python, compte-rendus et bibliothèques

Les TPs seront réalisés à l'aide de Python :

- dans des fichiers `.py`
- mais **pas** dans des notebooks

On pourra utiliser un éditeur intégré (comme par exemple Visual Studio Code) pour faciliter le développement.

Les comptes-rendus seront à faire en séance, sur la base des travaux réalisés durant les séances précédentes ; ils seront composés :

- d'une présentation synthétique des données étudiées (histogrammes et autres résumés statistiques)
- d'une analyse et d'une comparaison des différents résultats selon les méthodes de prédiction utilisées et selon les données sur lesquelles ces méthodes sont appliquées

On ne demande pas (ou très peu) d'extraits de code Python.

Les bibliothèques utilisables sont :

- matplotlib `https://matplotlib.org`
- pandas `https://pandas.pydata.org`
- NumPy `https://numpy.org`

Il ne devrait pas y avoir besoin d'autres bibliothèques ; en particulier l'utilisation de scikit-learn est interdite !

2 Jeu de données du TP 1 : phase d'entraînement

Le fichier `tp1_data_train.txt` contient des données correspondant à un phénomène aléatoire, avec deux colonnes :

- la première colonne contient une caractéristique réelle `x`
- la deuxième colonne contient la classe `y` (0 ou 1)

Ce seront les données d'entraînement (ou d'apprentissage) pour ce TP.

1) Ouvrez ce fichier dans une DataFrame Panda.
   Indication : on utilisera la méthode `read_csv(...)` de Panda.
2) À partir de cette DataFrame, définissez :
   - une matrice `X_train` avec les caractéristiques `x`
   - un vecteur `y_train` avec les classes `y`

   Indication : si `df` est une DataFrame Panda :
   - `df["nom"].values` permet d'extraire la colonne de nom `nom` et de la convertir en vecteur NumPy
   - `df[["nom1","nom2","nom3"]].values` permet d'extraire les colonnes de noms `nom1`, `nom2`, `nom3` et de les convertir en matrice NumPy
3) Combien de données sont dans la classe 0 ? dans la classe 1 ?
4) Sur le même graphique, affichez :
   - l'histogramme de répartition des caractéristiques `x` pour lesquelles la classe `y` est égale à 0
   - et l'histogramme de répartition des caractéristiques `x` pour lesquelles la classe `y` est égale à 1.

   On utilisera la fonction `hist(...)` de Matplotlib.
5) Visuellement, décidez de la valeur d'une frontière de décision `Delta` tel que, pour une valeur quelconque de `x` :
   - si `x < Delta` alors la classe prédite est la classe `0`
   - sinon la classe prédite est la classe `1`
6) On utilise cette frontière de décision pour la prédiction de notre modèle : écrivez une fonction python `prediction(...)` prenant en entrée une valeur `x` et donnant en sortie la classe prédite selon la frontière de décision `Delta`.

3 Phase de validation

Le fichier `tp1_data_valid.txt` contient des données qui correspondent au même phénomène aléatoire que les données d'apprentissage. Ce seront les données de validation pour ce TP.

1) Ouvrez ce fichier dans une DataFrame panda.
2) À partir de cette DataFrame, définissez une matrice `X_valid` contenant les valeurs `x` de ce jeu de données
3) À partir de cette matrice, construisez un vecteur `y_pred` contenant la prédiction de classe pour chaque valeur de `x`, en utilisant la prédiction définie à la partie précédente.
4) À partir de la DataFrame de validation, définissez un vecteur `y_valid` contenant les classes `y` de ce jeu de données, puis calculez :
   - le nombre d'erreurs de prédiction
   - le taux d'erreur de prédiction

   Comparez vos résultats avec vos camarades.
5) On appelle matrice de confusion binaire la matrice suivante, regroupant pour la classe `0` les vrais positifs, faux positifs, faux négatifs et vrais négatifs :

		Vérités terrain (`y_valid`)	
		positifs (classe 0)	négatifs (classe 1)
Prédictions (`y_pred`)	positifs (classe 0)	vrais positifs	faux positifs
	négatifs (classe 1)	faux négatifs	vrais négatifs

   Construisez la matrice de confusion pour ce modèle sur le jeu de données de validation.
   Comparez vos résultats avec vos camarades.