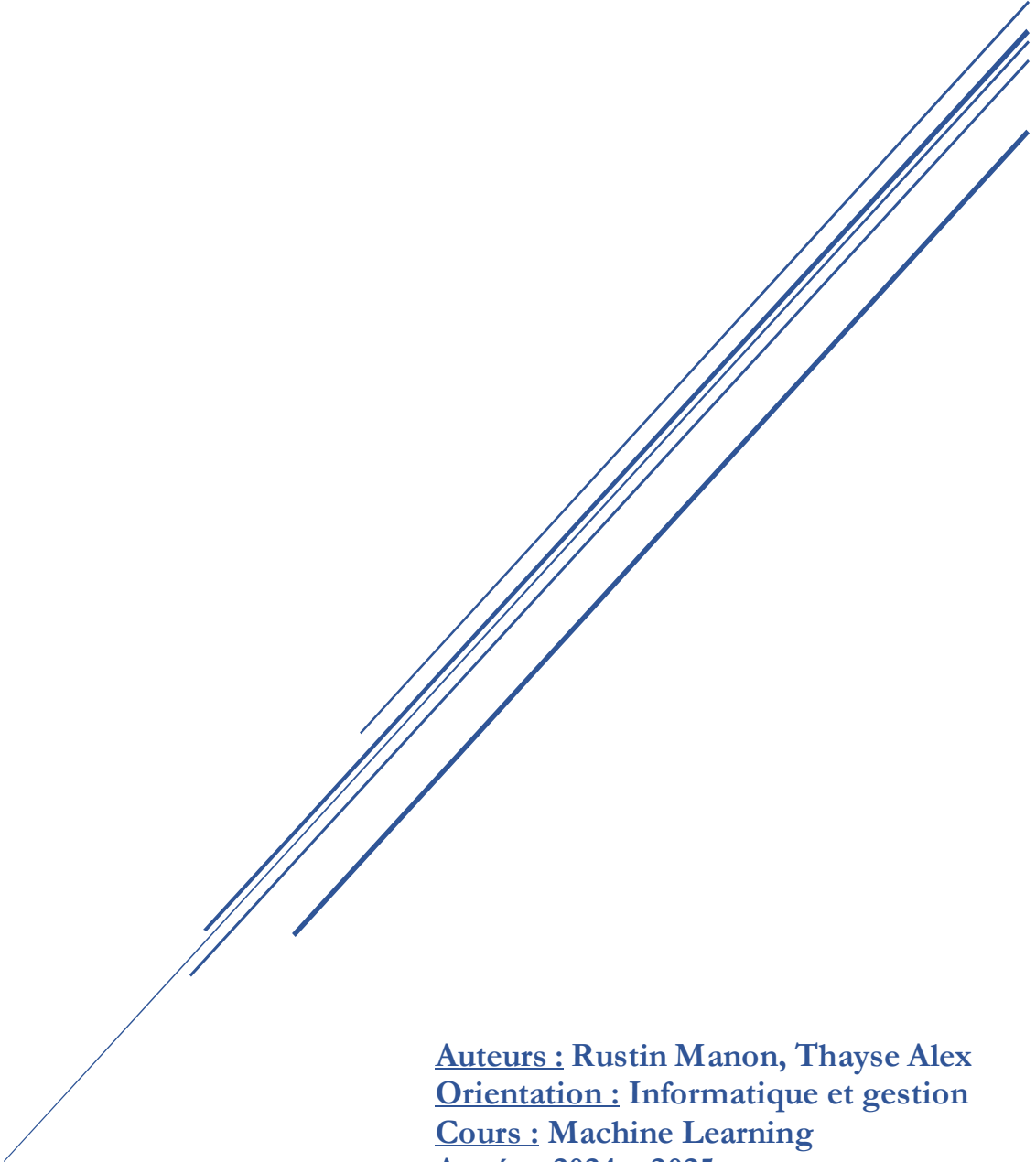


# Mode d'emploi - Classification supervisée appliquée sur un jeu de données de type Chinese MNIST



Auteurs : Rustin Manon, Thayse Alex  
Orientation : Informatique et gestion  
Cours : Machine Learning  
Année : 2024 – 2025  
Professeur : Siebert Xavier  
Assistant : Baeckelant Timothy

# 1. Introduction

Ce mode d'emploi a pour objectif d'expliquer la manière d'utiliser les différents codes implémentés pour ce projet de Machine Learning et de mettre en évidence les modifications possibles.

## 2. Mode d'emploi

### 2.1. Code principal (« *ML\_principale* »)

#### 2.1.1. Séparation des données

Séparation aléatoire des données :

```
X_train_random, X_test_random, y_train_random, y_test_random = train_test_split(X, y, test_size=0.4, random_state=42, stratify=y)
```

Séparation ordonnée des données :

Cette séparation s'effectue en utilisant le code donné sur Moodle.

#### 2.1.2. Modification des modèles

Pour chaque modèle, l'utilisateur peut appliquer les hyperparamètres qu'il souhaite en modifiant cette ligne :

```
model_time = KNeighborsClassifier(n_neighbors=5, weights="uniform", metric="euclidean")
```

Ceci permettra de tester un modèle personnalisé sur le jeu de données en obtenant ses métriques de performance, ses courbes ROC et sa matrice de confusion.

### 2.2. Code sur les données augmentées (« *ML\_données\_augmentées* »)

Ce code implémente l'augmentation des données et applique les modèles sur celles-ci.

### 2.3. Code sur les méthodes d'ensemble (« *ML\_méthodes\_ensemble* »)

Ce code implémente les méthodes de Voting et Stacking sur un ensemble de modèle en les appliquant sur les données augmentées. A noter, que des méthodes de Bagging et de Boosting sont implémentées mais ne sont pas appliquées sur notre jeu de données.

### 2.4. Code en supprimant la classe 1000 (« *ML\_sans\_classe1000* »)

Ce code implémente les méthodes des codes précédents en y supprimant la classe 1000.