

## IMSP

### Master en Data science

#### TD 03

#### Exercice 1 : Calcul du Support et de la Confiance

On dispose du tableau suivant représentant des transactions dans un supermarché :

Transaction ID	Articles achetés
1	{Lait, Pain, Beurre}
2	{Lait, Pain}
3	{Pain, Beurre}
4	{Lait, Beurre}
5	{Lait, Pain, Beurre}

1. Calculez le **support** de la règle  $\{\text{Lait}\} \rightarrow \{\text{Pain}\}$ .
2. Calculez la **confiance** de la règle  $\{\text{Lait}\} \rightarrow \{\text{Beurre}\}$ .
3. La règle  $\{\text{Pain}\} \rightarrow \{\text{Beurre}\}$  est-elle forte si le seuil de confiance minimum est de 60 % ?

#### Exercice 2 : Analyse des Règles avec le Lift

On extrait trois règles d'association :

1.  $\{X\} \rightarrow \{Y\}$  avec **confiance** = 80 % et **support** = 30 %
2.  $\{X\} \rightarrow \{Z\}$  avec **confiance** = 50 % et **support** = 20 %
3.  $\{Y\} \rightarrow \{Z\}$  avec **confiance** = 70 % et **support** = 40 %

Sachant que les supports individuels sont :

- $\text{Support}(X) = 50 \%$
- $\text{Support}(Y) = 60 \%$
- $\text{Support}(Z) = 30 \%$

1. Calculez le **lift** des trois règles.
2. Déterminez si elles sont intéressantes (lift > 1 signifie une corrélation positive).

### Exercice 3 : Implémentation d'un CNN simple (MNIST)

Objectif : Construire un réseau CNN en utilisant Keras et TensorFlow pour classifier des chiffres manuscrits du dataset MNIST.

Instructions :

1. Charger le dataset MNIST.
2. Normaliser les images (les valeurs des pixels entre 0 et 1).
3. Construire un CNN avec :
  - Une couche Conv2D (32 filtres 3×3, ReLU).
  - Une couche MaxPooling2D (2×2).
  - Une couche Fully Connected.
4. Compiler et entraîner le modèle.

### Exercice 4 : Amélioration de la Précision avec Data Augmentation

Objectif : Utiliser Data Augmentation pour améliorer la généralisation du CNN sur CIFAR-10.

Instructions :

1. Charger **CIFAR-10** et normaliser les images.
2. Ajouter des transformations :
  - Rotation, zoom, miroir, translation.
3. Comparer la précision avec et sans Data Augmentation.

### Exercice 5 : Implémentation de Dropout et Batch Normalization

Objectif : Comparer l'impact du **Dropout** et de la **Batch Normalization** sur l'apprentissage du modèle.

Instructions :

1. Entraîner un CNN sur **CIFAR-10** sans régularisation.
2. Ajouter **Batch Normalization** après chaque couche convolutionnelle.
3. Ajouter **Dropout** (0.5) avant la couche fully connected.
4. Comparer les performances et analyser les courbes d'apprentissage.

### Exercice 6 : Utilisation d'un CNN pré-entraîné (Transfer Learning)

**Objectif** : Utiliser **ResNet50** pré-entraîné sur **ImageNet** pour classer des images personnalisées.

**Instructions** :

1. Charger **ResNet50** sans la dernière couche de classification.
2. Remplacer la dernière couche par une nouvelle adaptée à votre dataset.
3. Entraîner uniquement cette couche avec un petit dataset.
4. Comparer les performances avec un entraînement **from scratch**.

### Exercice 7 : Extraction de Features pour du Clustering

**Objectif** : Extraire les **features** d'un CNN pour effectuer du **clustering** avec **K-Means**.

**Instructions** :

1. Entraîner un CNN sur **CIFAR-10** (ou utiliser un modèle pré-entraîné).
2. Extraire les **features latentes** (sortie d'une couche Fully Connected).
3. Appliquer **K-Means** sur ces features.
4. Visualiser les clusters formés.

### Exercice 8 : Classification du Cancer du Sein (Mammographies)

**Objectif** : Construire un **CNN** pour classer les mammographies en **bénignes ou malignes**.

**Instructions** :

1. Charger un dataset de mammographies (ex: **BCDR, CBIS-DDSM**).
2. Prétraiter les images :
  - Normalisation des pixels
  - Redimensionnement
  - Augmentation des données (flip, rotation, contraste)
3. Construire un CNN avec :
  - 3 couches **Conv2D + MaxPooling**
  - 2 couches **Fully Connected**
  - Fonction d'activation **Softmax** (classification binaire)

4. Entraîner et évaluer le modèle.
5. Bonus : Tester un modèle pré-entraîné (ResNet, VGG16) et comparer les résultats.