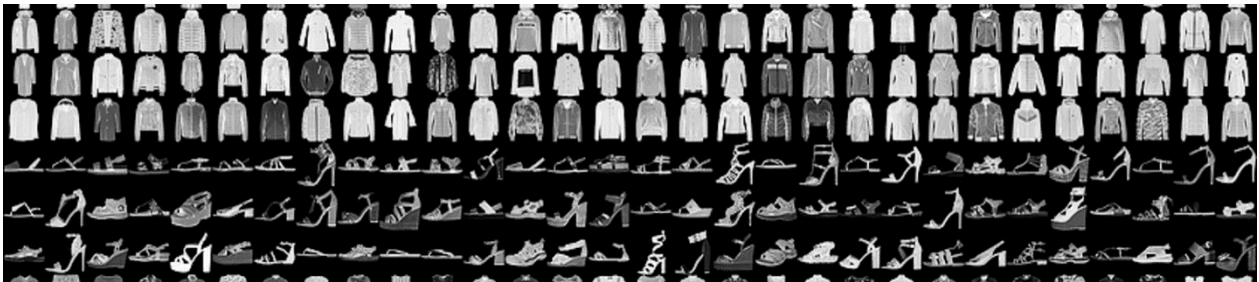


TP N°4 Énoncé (Classification)

Version 1

L'objectif de ce TP est d'appliquer :

- ✓ La régression logistique multiclass sur un jeu de données images



Contexte :

- Fashion-MNIST est un ensemble de données d'images, composé d'un ensemble d'entraînement de 60 000 exemples et d'un ensemble de test de 10 000 exemples. Chaque exemple est une image en niveaux de gris de 28x28 pixels, associée à une étiquette parmi 10 classes. Il partage la même taille d'image et la même structure de répartition entre les ensembles d'entraînement et de test.
- Chaque image mesure 28 pixels de hauteur et 28 pixels de largeur, pour un total de 784 pixels au total.
- Chaque pixel a une seule valeur associée, indiquant la luminosité ou l'obscurité de ce pixel, les valeurs plus élevées signifiant plus sombre. Cette valeur de pixel est un entier compris entre 0 et 255.
- Les ensembles de données d'entraînement et de test ont 785 colonnes.
- La première colonne se compose des étiquettes de classe et représente l'article vestimentaire.
- Le reste des colonnes contient les valeurs de pixels de l'image associée.

- Chaque exemple d'entraînement et de test est assigné à l'une des étiquettes suivantes :

0 T-shirt	5 Sandale
1 Pantalon	6 Chemise
2 Pull	7 Basket
3 Robe	8 Sac
4 Manteau	9 Bottine

Tâche 1 : Appliquer le modèle de la régression logistique sans la transformation des données.

- ✓ **Le fichier associé :** TP4_Classification_Logistique_Multiclass_Méthode1_MNIST
- ✓ **Les étapes essentielles :**
 - Charger les jeux de données `fashion-mnist_train` et `fashion-mnist_test`
 - Séparer des données en ensembles d'entraînement et de test. Autrement, déterminez `X_train`, `Y_train`, `X_test` `Y_test`.
 - Créer une liste de données (nommée `X_train_liste`) à partir de `X_train`
 - Visualiser l'image d'indice 0 dans la liste `X_train_liste`

```
plt.imshow(np.reshape(X_train_list[0],(28,28,1)))
```
 - Créer et entraîner le modèle de régression logistique
 - Calculer des prévisions `Y_pred` sur `X_test`
 - Déterminer la matrice de confusion en utilisant la bibliothèque `searborn`
 - Déterminer les métriques d'évaluation
 - Tester la classe de l'image d'indice 2 dans la liste `X_train_liste`
 - Prédire l'étiquette d'une nouvelle image (`t1.jpg`) via les instructions suivantes :

```
from PIL import Image
# Charger l'image depuis le chemin spécifié
im = Image.open("./datasets/t1.jpg")
# Convertir l'image en niveaux de gris
im_gris = im.convert('L')
# Redimensionner l'image à la taille en 28x28 pixels
im_red = im_gris.resize((28, 28))
# Faire une prédiction avec le modèle Logistic Regression
Y_prevu = model.predict(np.array(im_red).reshape(1, -1))
```

```
print(Y_prevu)
```

Tâche 2 : Refaire la tâche 1 en appliquant la transformation de données (la réduction et normalisation) sur les jeux de données.

- ✓ **Comparer le modèle de tâche 1 avec le modèle de tâche 2**
- ✓ **Le fichier associé : TP4_Classification_Logistique_Multiclass_Méthode2_MNIST**