On considère un échantillon indépendant $((\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n))$ d'individus $\boldsymbol{X}_i$ décrits par d variables réelles ($\boldsymbol{X}_i \in \mathbb{R}^d$) d'une population de K classes hétérogènes telle que $Y_i \in [\![1, K]\!]$ est la classe de l'individu $\boldsymbol{X}_i$. On dispose d'un échantillon observé datatrain-bis $((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$. L'objectif est de prédire les classes pour un échantillon de test issu de cette même population sur la base d'un modèle probabiliste paramétrique de paramètres $\boldsymbol{\theta}$. La prédiction peut ainsi s'effectuer par la règle du maximum a posteriori (MAP) qui consiste à maximiser la probabilité a posteriori pour avoir une prédiction $\widehat{y}_i$ de la classe de l'individu $\boldsymbol{x}_i$, étant donné un modèle de paramètres $\hat{\boldsymbol{\theta}}$ appris à partir de l'échantillon observé :

$$\widehat{y}_i = \arg \max_{y_i \in [\![1,K]\!]} \mathbb{P}(Y_i = y_i | \boldsymbol{X} = \boldsymbol{x}_i; \boldsymbol{\theta}). \tag{1}$$

Modélisation discriminative : Régression logistique :

On considère un problème à deux classes ($Y_i \in [\![0, 1]\!]$, i.e., régression logistique binaire). Dans ce cas le modèle est défini par les probabilités

$$\pi(\boldsymbol{x}; \boldsymbol{\theta}) = \mathbb{P}(Y = 1 | \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\theta}) = \frac{\exp(\beta_{10} + \boldsymbol{\beta}_1^T \boldsymbol{x})}{1 + \exp(\beta_{10} + \boldsymbol{\beta}_1^T \boldsymbol{x})} \tag{2}$$

et $\mathbb{P}(Y = 0 | \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\theta}) = 1 - \pi(\boldsymbol{x}; \boldsymbol{\theta})$.
Le vecteur paramètre du modèle défini par $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\beta}_1^T)^T \in \mathbb{R}^{d+1}$ est estimé en maximisant par rapport à $\boldsymbol{\theta}$ la log-vraisemblance conditionnelle qui dans ce cas de classification binaire prend la forme :

$$\begin{aligned}
\log L(\boldsymbol{\theta}; \mathbf{X}, \boldsymbol{y}) &= \log \prod_{i=1}^{n} \mathbb{P}(Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{\theta})^{y_i} \mathbb{P}(Y_i = 0 | \boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{\theta})^{1-y_i} \\
&= \sum_{i=1}^{n} y_i \log \mathbb{P}(Y_i = 1 | \boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{\theta}) + (1 - y_i) \log \mathbb{P}(Y_i = 0 | \boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{\theta}) \\
&= \sum_{i=1}^{n} y_i \log \pi(\boldsymbol{x}_i; \boldsymbol{\theta}) + (1 - y_i) \log (1 - \pi(\boldsymbol{x}_i; \boldsymbol{\theta})) \\
&= \sum_{i=1}^{n} y_i (\beta_{10} + \boldsymbol{\beta}_1^T \boldsymbol{x}_i) - \log(1 + \exp(\beta_{10} + \boldsymbol{\beta}_1^T \boldsymbol{x}_i)).
\end{aligned} \tag{3}$$

On montre que dans le cas de ce modèle, l'algorithme de Nerwton-Raphson consiste à partir d'un modèle initial de paramètres $\boldsymbol{\theta}^{(0)}$ et à mettre à jour, à chaque itération t, les paramètres selon l'équation de mise à jour suivante, jusqu'à ce qu'il n'y ait plus d'augmentation significative au sens d'un seuil préfixé de la log-vraisemblansce conditionnelle (3) :

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} + (\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^T (\boldsymbol{y} - \boldsymbol{p}^{(t)}) \\
&= (\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}^{(t)} \widetilde{\boldsymbol{y}}
\end{aligned} \tag{4}$$

où :

- $\mathbf{X}$ est la matrice de dimensions $n \times (d+1)$ de lignes $(1, \boldsymbol{x}_i^T)$ pour $i = 1, \ldots, n$;
- $\boldsymbol{y}$ est le vecteur colonne de dimension $n \times 1$ des labels y_i : $\boldsymbol{y} = (y_1, \ldots, y_n)^T$;

— $\boldsymbol{p}$ est le vecteur colonne de dimension $n \times 1$ des probabilités : $\boldsymbol{p} = (\pi(\boldsymbol{x}_1; \boldsymbol{\theta}), \ldots, \pi(\boldsymbol{x}_n; \boldsymbol{\theta}))^T$ avec $\pi(\boldsymbol{x}_i; \boldsymbol{\theta}) = \mathbb{P}(Y_i = 1 | \boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\beta}_k^T \boldsymbol{x}_i)}{1+\exp(\boldsymbol{\beta}_k^T \boldsymbol{x}_i)}$;

— $\mathbf{W}$ est la matrice diagonale de dimension $n \times n$ d'élements $\pi(\boldsymbol{x}_1; \boldsymbol{\theta})\,(1 - \pi(\boldsymbol{x}_n; \boldsymbol{\theta}))$: $\mathbf{W} = \text{diag}(\boldsymbol{p})$.

— $\widetilde{\boldsymbol{y}} = \mathbf{X}\boldsymbol{\theta}^{(t)} + (\mathbf{W}^{(t)})^{-1}(\boldsymbol{y} - \boldsymbol{p}^{(t)})$

De part la forme de (4) qui correspond à celle de l'estimateur des moindres carrés pondérés (avec une matrice de poids $\mathbf{W}$) on appelle l'algorithme Netwton-Raphson dans ce cas l'algorithme IRLS (Iterative Reweighted Least Squares pour moindres carrés pondérés itératifs).

Travail demandé :

1. Implémenter et tester l'algorithme sur les données du TP LDA. Pour cela, créer une fonction `train_LogReg.m` pour l'apprentissage et une fonction `predict_LogReg.m` pour la prédiction.
2. Calculer le taux d'erreur sur l'échantillon d'apprentissage
3. Maintenant considérer un échantillon de teste (utiliser un cadriallage uniforme de l'espace des données comme dans le TP précédent ou dans le TP1)
4. Afficher, sur le même graphique, les données d'apprentissage et les données de test classées
5. Comparer vos résultats à ceux obtenus en se basant sur les fonctions `mnrfit` et `mnrval` de Matlab.
6. Comparer vos résultats à ceux obtenus en se basant sur les fonctions `glmfit` et `glmval` de Matlab.