Rapport des expérimentations pour le TP 1

Trinôme : Nehemia BACQUE & Onsi CHHIMA & Reda LALOU

1. *Simulation avec la fonction **MixSim** :*

Tout d'abord , nous avons commencé à analyser l'article qui explique la fonction **MixSim.**

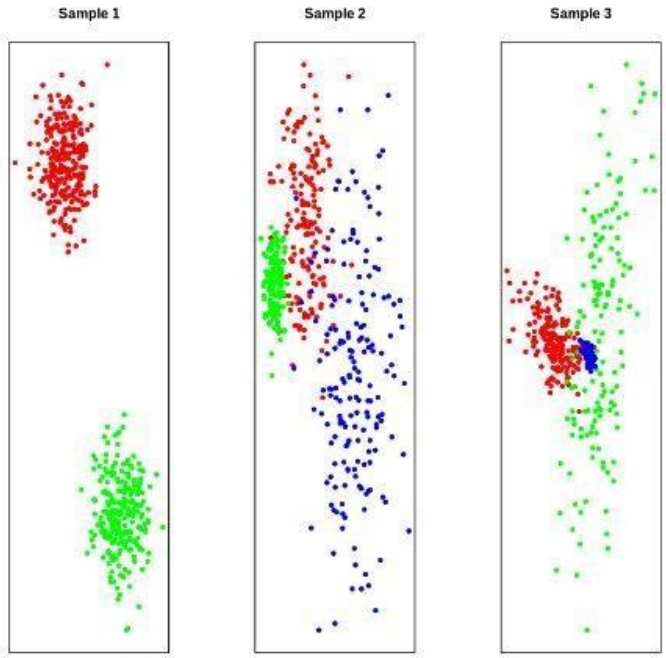
Nous avons compris que **MixSim** nous permet de générer un modèle de mélange Gaussien avec des niveaux prédéfinis de chevauchements, ainsi que de définir la valeur du chevauchement moyen ou maximal, le nombre de classe souhaité, la dimension de nos données et la structurede la matrice de variance – covariance (sphérique ou non sphérique).

Après le traitement avec la bibliothèque **MixSim**, nous avons utilisé **simdataset** afin d'appliquer une simulation à un jeu de données de taille n, et lui attribuer les sorties de la fonction **MixSim** (moyenne, matrice de variance – covariance et le vecteur des proportions de mélange).

Il nous a été demandé dans ce TP de simuler les trois jeux de données ci-dessous :

- **Jeu1** : une matrice de taille 500x2 avec deux classes sphériques et bien séparées.
- **Jeu2** : une matrice de taille 500x2 avec 3 classes sphériques et un degré de mélange différent de zéro.
- **Jeu3** : une matrice de taille 500x2 avec 3 classes non- sphériques et un degré de mélange différent de zéro.

En utilisant les fonctions **MixSim** et **simdataset** nous obtenons les résultats suivants :



- **Jeu1** : données bien séparées et sphériques, donc variables indépendantes.
- **Jeu2** : structure sphérique, donc variables indépendantes mais avec un degré de chevauchement entre les classes.
- **Jeu3** : structure non sphériques, donc variables dépendantes avec chevauchement entre les classes.

2. *Classifieur linéaire et quadratique*

L'objectif est d'implémenter sous R la règle de classification $\boldsymbol{g_i(X)}$ qui est une fonction discriminante permettant de connaitre l'appartenance d'une variable **x** à une classe **i**.

```
G <- -0.5*(t(X-jeu.Q$Mu))%*%solve(Sigma1)%*%(X-jeu.Q$Mu)-0.5*log(det(Sigma1)) + log(jeu.Q$Pi)
```

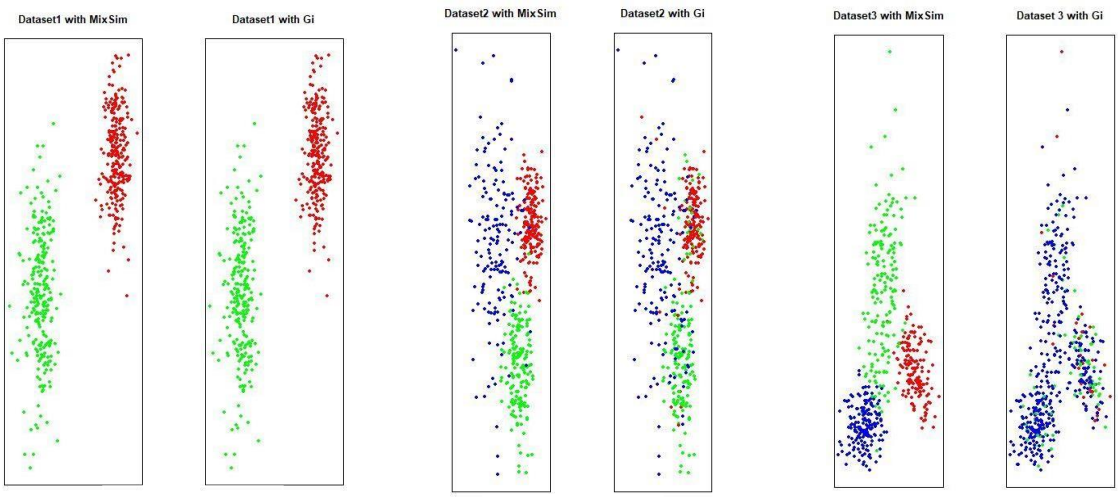
Mu, Sigma et Pi sont respectivement la moyenne, la matrice de variance – covariance et le vecteur des proportions de mélanges pour chaque classe, calculés avec la fonction **MixSim**.

- Cas 1 : $\sum_i = \sigma^2 I$:

La matrice de variance – covariance est une matrice diagonale, de cette manière, nos variables sont indépendantes , cette matrice ne dépend pas de i. La formule de la fonction discriminante sous R devient comme suit :

```
G <- -0.5*(t(X-jeu.Q$Mu))%*%(X-jeu.Q$Mu) + log(jeu.Q$Pi)
```

En appliquant cette formule à nos données, on obtient les résultats suivants :



Dataset 1 Dataset 2 Dataset 3

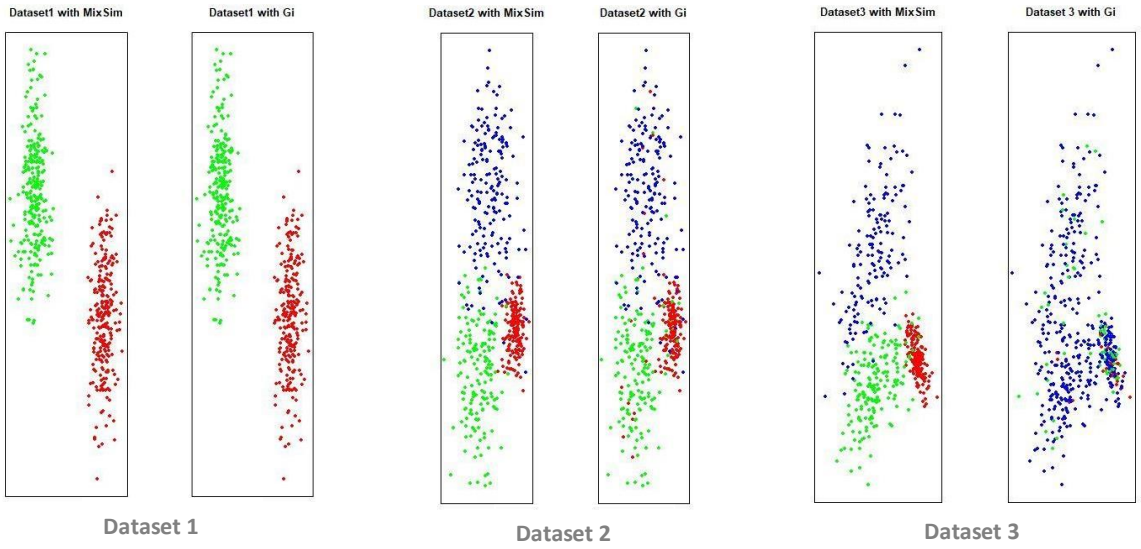
Matrice de confusion	Accuracy	Sensitivity	Specificity
Jeu 1	1	1	1
Jeu 2	0.93	0.92	0.97
Jeu 3	0.30	0.34	0.47

- Cas 2 : $\sum_i = \sum$:

Les matrices de variance – covariance sont égales entre classes mais restent arbitraires. La formule de la fonction discriminante sous R devient comme suit :

```
G <- -0.5*(t(X-jeu.Q$Mu))%*%solve(Sigma1)%*%(X-jeu.Q$Mu) + log(jeu.Q$Pi)
```

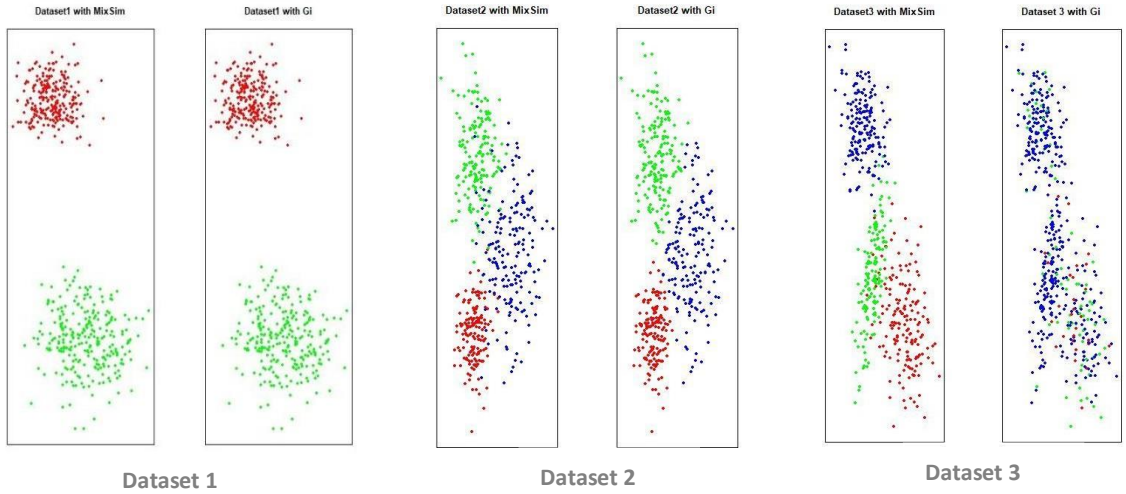
En appliquant cette formule à nos données, on obtient les résultats suivants :



Matrice de confusion	Accuracy	Sensitivity	Specificity
Jeu 1	1	1	1
Jeu 2	0.942	0.942	0.97
Jeu 3	0.23	0.19	0.59

- Cas général où $\sum_i$ est arbitraire :

En appliquant la formule générale à nos trois jeux de données, on obtient les résultats suivants :



Matrice de confusion	Accuracy	Sensitivity	Specificity
Jeu 1	1	1	1
Jeu 2	0.942	0.94	0.97
Jeu 3	0.53	0.55	0.78

Conclusion :

En appliquant la fonction discriminante sur nos trois jeux de données, nous constatons que les deux premiers datasets sont biens classés avec nos 3 fonctions, hors que le troisième dataset n'a eu une meilleure classification qu'en utilisant le cas général de notre fonction discriminante.

Étant donné que les trois cas de notre fonction discriminante prennent en considération les distributions linéaires, une bonne classification est présente dans notre premier jeu de données nos classes sont bien séparées. Même chose pour le deuxième jeu de données, en effet ,
même si nous avons un degré de mélange différent de zéro (ce qui implique que le chevauchement des classes) nous avons aussi des classes sphériques et par conséquent on peut avoir des variables indépendantes ce qui va intégrer une bonne classification.

Pour le troisième dataset, comme les classes sont de forme non sphérique, la classification reste toujours difficile à faire, pour cette raison nous constatons que la fonction discriminante ne fonctionne pas avec nos deux premiers cas, car ils sont basés sur une classification linéaire.