Rapport données

Résumé des données à dispositions :

4 jeux de données sont à notre disposition. Plusieurs données sont du Type NaN → à prendre en compte

Données d'entrainements :

- x_train : table de 34515 échantillons de dimension 13
- y_train : table de sortie des 34515 échantillons | 0 → good sample / 1 → bad sample

Données de test :

- x_test : table de 8001 échantillons de dimension 13
- y_test : table de sortie des 8001 échantillons

Répartitions des données à dispositions :

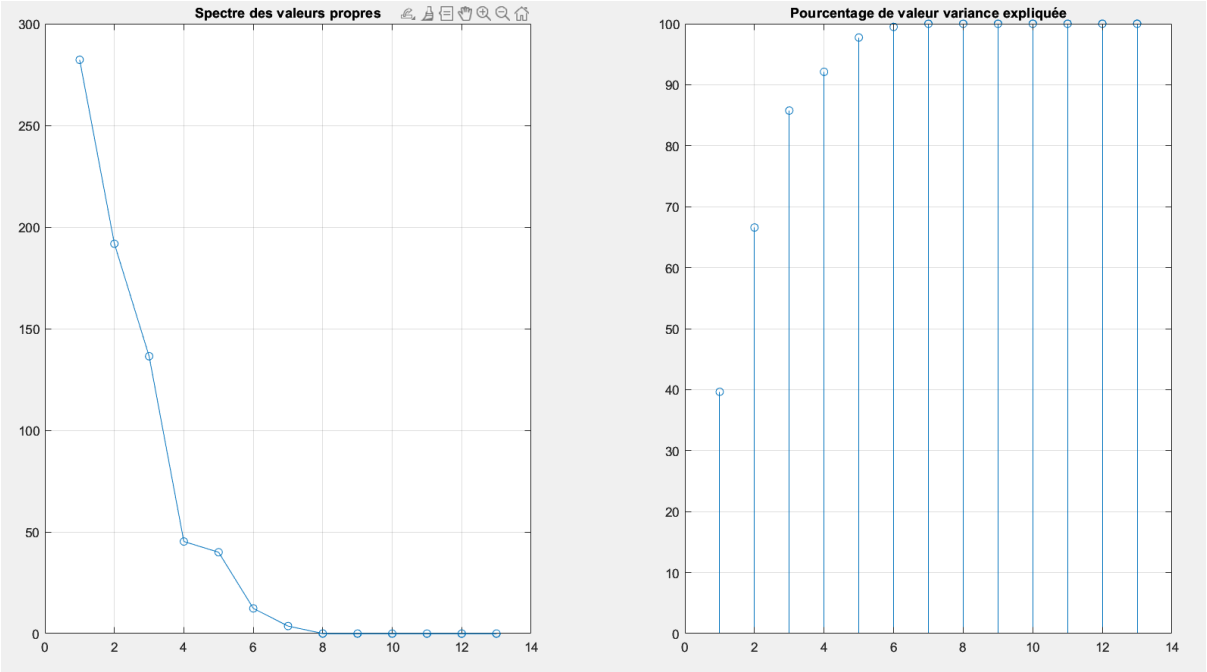
Pour la table y_train, compte les éléments des classes 0 et 1

Nbr d'éléments de la classe 0	Nbr d'éléments de la classe 1
34 210	305

Il y a un fort déséquilibre de classe, la classe 0 contient beaucoup plus d'échantillons que la 1. Il faudra prendre cela en compte pour les apprentissages.
→ Se renseigner sur l'impact potentiel et la sensibilité de chaque méthode à ce problème

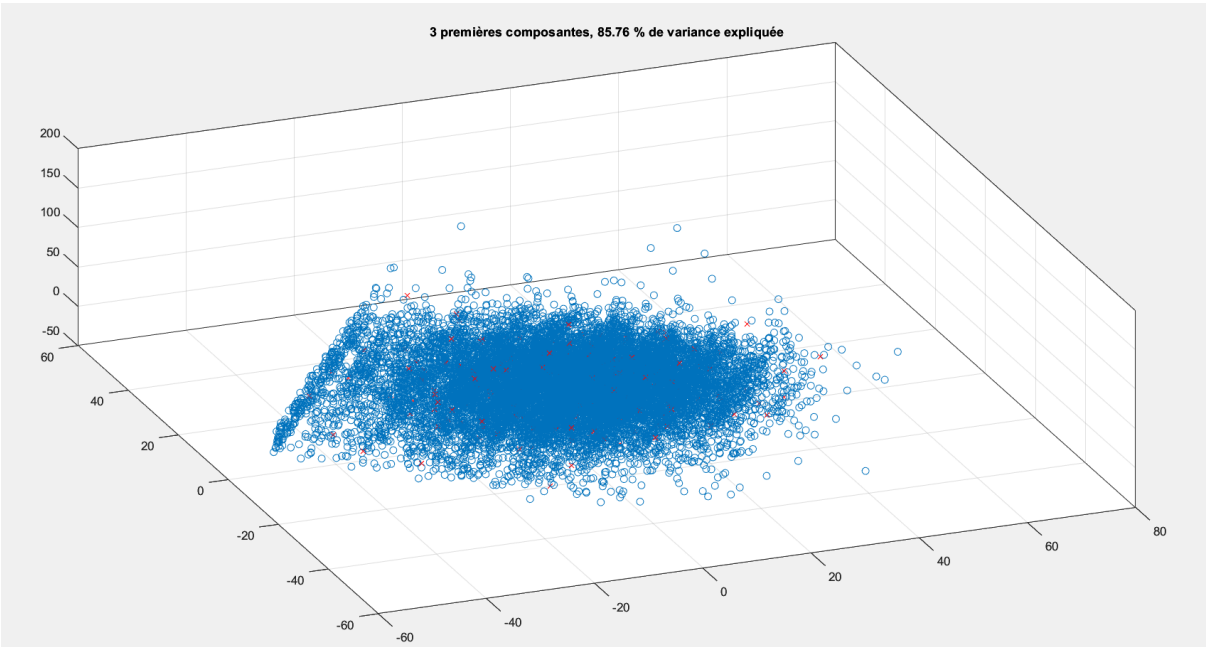
Poids des données :

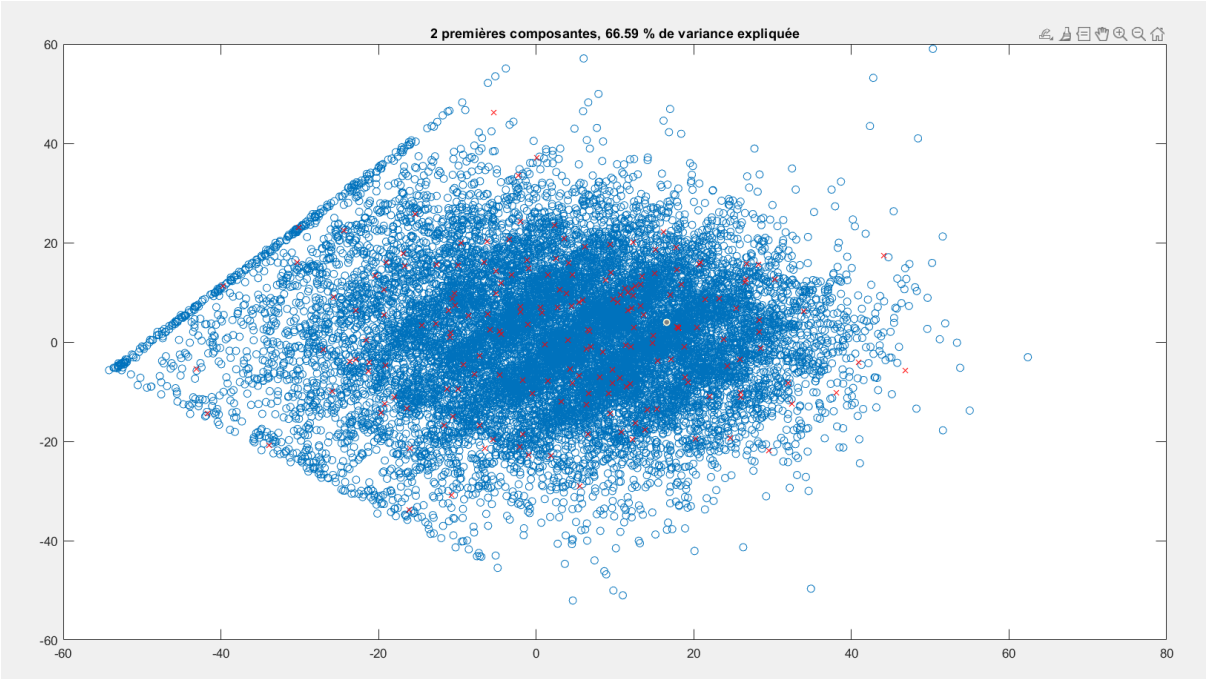
Pour cette partie je réalise un ACP afin d'exprimer le "poids" de chaque dimension des échantillons. Ces résultats serviront juste pour un premier affichage des données afin de comprendre les jeux; cela peut ne pas être représentatif pour les apprentissages à cause du déséquilibre de classe ainsi que des données de Type NaN qui ont été ignorés pour ces calculs.



Avec ce graphe on détermine qu'avec 2 composantes, l'on peut exprimer environ 67% de la variance des données, et avec 3 l'on peut exprimer environ 86%.

Affichage des données:





Les x rouges représentent les données de la classe 1. Dans les espaces réduits de dimensions 2 et 3, les données ne sont pas du tout séparables, il faudra donc surement travailler sur l'espace complet.

Ces résultats ne sont qu'à titre indicatif bien évidemment; je pense qu'il faudra d'abord régler le problème de déséquilibre de classe ainsi que la gestion des données manquantes (NaN).

Méthodes à étudier :

- Classifieur bayésien Naïf
- Noyaux de Parzen
- Knn classification
- Kmeans
- SVM
- Analyse discriminante de Fisher
- Réseau de neurones profonds (Modèle Perceptron)

Pistes pour le déséquilibre de données :

Premièrement, l'on pourrait sélectionner de manière aléatoire un certain nombre de données parmi celles de la classe 0 afin d'équilibrer la répartition des classes.

Rechercher si des méthodes de ML sont moins sensibles que d'autre au déséquilibre de classe.