ABABII Anişoara
GAUTIER Baptiste

UNIVERSITÉ PARIS I
PANTHÉON SORBONNE

# Projet Machine Learning

Le projet consiste à prédire le diagnostic d'un patient, afin de savoir si ce dernier est atteint de schizophrénie ou non. A cet égard, nous disposons d'une variable binaire $y \in [0; 1]$ qu'il convient de prédire. Nous sommes donc face à un problème de classification binaire. Dès lors, les métriques d'évaluation que nous avons utilisées sont la **ROC AUC** ainsi que la **Balanced Accuracy**.

Nous disposons pour ce projet d'un dataset composé de 513 lignes et de 331 979 colonnes. De fait, attaquer la modélisation de brut en blanc nous expose clairement au fléau de la dimension, et *ipso facto* à un danger d'*overfitting*. Qui plus est, le théorème de VC dimension nous garantit d'obtenir avec un classifieur linéaire simple une performance parfaite sur le jeu d'entrainement dû au surapprentissage.

Pour pallier à ce problème, étant donné que nous ne pouvons pas obtenir d'autres observations, il nous faut réduire la dimension de nos données. Nous pourrions passer par une méthode de machine learning pour ce faire, cependant nous disposons de **284** variables présélectionnées par une connaissance métier, les *Regions of Interest* (ROIs). Nous avons donc décidé de laisser de côté les voxels et de ne garder que ces dernières.

Nous avons séparé notre jeu de données en un jeu d'entrainement et en un jeu de test. Pour le choix des paramètres des modèles présentés ci-après, nous avons décidé d'opter pour une cross-validation, scindant notre jeu de test en 5 *folds*. Ce choix est motivé par le faible nombre d'observations dont nous disposons, nous avons estimé qu'il ne nous resterait pas assez de données si on rescindait le jeu d'entrainement pour obtenir un jeu de validation.

Concernant la modélisation, nous avons détecté que certaines des variables étaient (fortement) corrélées entre elles. De ce fait, nous avons choisi pour notre modèle linéaire d'utiliser une régularisation. Nous avons choisi de retenir un ElasticNet, incluant à la fois un **Lasso** pour la sélection de variables ainsi qu'une régularisation **L2** afin de réduire de façon plus douce les problèmes de multi colinéarités.

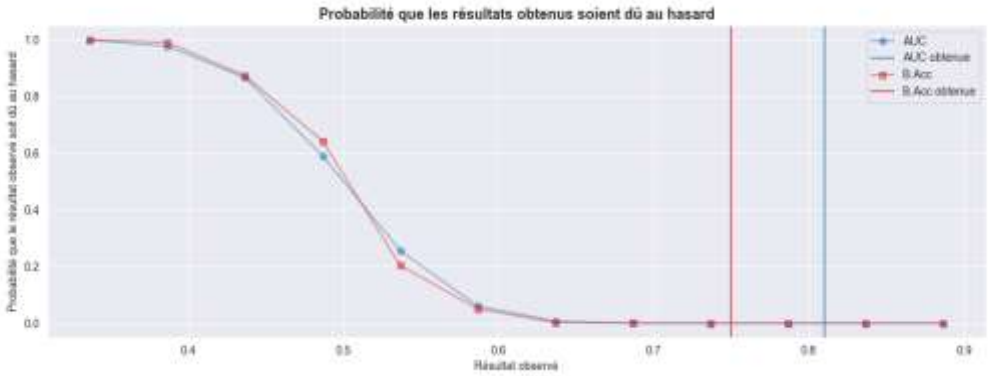
En parallèle à l'ElasticNet, nous avons décidé de tester d'autres modèles non linéaires issues des différentes grandes familles de modèles : Un *support vector machine* à noyau gaussien, un Random Forest, un Gradient Boosting, et un MLP. A noter qu'au vu du faible nombre de variables, nous avons estimé pertinent de garder un Lasso en amont de ces modèles – même si le Random Forest par exemple n'est pas forcément sensible au Lasso. Nous avons testé pour chaque modèle une combinaison de paramètres via la cross-validation décrite plus haut. Nous avons finalement choisi la combinaison nous donnant le meilleur résultat moyen pour les 5 *folds* pour chaque modèle. Voici un tableau récapitulatif des résultats :

| Modèle | ROC AUC CV | B. Acc CV | ROC AUC test | B. Acc test |
|---|---|---|---|---|
| ElasticNet | 0,89 | 0,80 | 0,81 | 0,71 |
| SVM | 0,86 | 0,76 | 0,83 | 0,73 |
| Random Forest | 0,82 | 0,72 | 0,81 | 0,75 |
| Gradient Boosting | 0,83 | 0,75 | 0,84 | 0,75 |
| MLP | 0,77 | 0,71 | 0,63 | 0,74 |

Au regard des résultats, nous avons choisi de garder comme modèle final le **Gradient Boosting**, meilleur challenger lorsque l'on fait la moyenne des deux métriques. Pour chaque modèle, nous avons testé le fait que le résultat obtenu sur le test set soit dû au hasard (en réévaluant le modèle sur un jeu

d'entrainement permuté aléatoirement 1000 fois). Nous obtenons pour chaque modèle une probabilité quasiment nulle.

Voici un graphique illustratif pour les résultats du Gradient Boosting.




Note : Pour les modèles ensemblistes, nous avions initialement retenu un Random Forest. Par acquis de conscience, nous avons voulu tester également un Gradient Boosting. Il se trouve que les résultats obtenus furent légèrement meilleurs sur le second. C'est donc lui que nous avons finalement gardé (mais vous trouverez les 2 sur RAMP avec et sans sélection de variables par Lasso).