Projet

Master parcours SSD - UE Apprentissage Statistique II

Cet examen mi-parcours prend la forme d'un devoir maison, à réaliser **en groupe de 2 ou 3**. Il faut soumettre (via la plateforme moodle) :

1. un rapport de 10 pages maximum (sans le code) décrivant les analyses réalisées,
2. un (ou des) notebook(s) Jupyter (commenté un minimum) permettant de reproduire vos analyses.

Contexte

Dans cet exercice vous travaillerez sur une problématique de classification binaire visant à prédire si une bactérie est résistante ou susceptible à divers antibiotiques à partir de trois matrices de descripteurs :

1. une matrice contenant des variables binaires codant pour la présence ou l'absence de différents gènes de résistance dans le génome de la bactérie. Dans certains cas, la présence d'un gène est en effet suffisante pour permettre à une bactérie de résister à un antibiotique.
2. une matrice contenant d'autres variables binaires codant pour la présence de mutations ponctuelles (SNPs) au sein de certains gènes. Certains gènes sont en effet parfois toujours présents dans le génome d'une bactérie, et la résistance à un antibiotique résulte de l'acquisition de mutations ponctuelles pouvant rendre la protéine codée par un gène non fonctionnelle.
3. une matrice contenant des variables quantitatives quantifiant l'expression de différents gènes. Cette information permet de savoir quels gènes sont "actifs" (ou "exprimés") chez la bactérie. Elle reflète par conséquent davantage son état cellulaire que les deux sources d'information précédentes, qui caractérisent son patrimoine génétique et sont donc statiques, et parfois insuffisantes pour prédire efficacement la résistance d'une bactérie.

Vous aurez par ailleurs accès aux phénotypes de résistance (résistant ou susceptible) pour 5 antibiotiques : la ceftazidime, la ciprofloxacine, la colistine, le méropenème et la tobramycine.

Cette problématique offre de grandes perspectives en terme de diagnostic et suscite un intérêt important dans la communauté scientifique, la valeur prédictive apportée par cette information "omique" (génomique et transcriptomique) restant néanmoins encore à démontrer. L'objectif de cette étude sera donc (i) d'évaluer quel niveau de performance peut être atteinte pour prédire la résistance aux différents antibiotiques, et (ii) quelles sources d'information sont les plus pertinentes pour chacun d'entre eux. Vous aurez à disposition un jeu d'apprentissage et réaliserez une étude de validation croisée. Le jeu de données n'étant pas toujours équilibré en terme de nombres de souches susceptibles et résistantes pour les différents antibiotiques, le critère de performance considéré sera la moyenne entre la sensibilité et la spécificité du modèle (aussi appelé "macro recall" dans la terminologie `scikit-learn`), pour que les résultats soient facilement comparables entre antibiotiques.

Vous avez toute liberté quant au choix des modèles à considérer, de leurs hyperparamètres, des éventuels prétraitements à appliquer au jeu de données et de la manière de combiner les différentes sources de données. Vous serez évalués sur la qualité de vos prédictions, sur la pertinence de votre démarche, et sur l'originalité de votre travail. N'hésitez notamment pas à explorer certaines fonctionnalités de `scikit-learn` peu ou pas vues en cours.

Jeu de données

Le jeu de données met en jeu 414 souches bactériennes. Il est stocké dans un fichier "pickle" (`dataset.pkl`) contenant les objets suivants :

- un data-frame `pandas` appelé `pheno` contenant les phénotypes de résistance (0 = susceptible ; 1 = résistant) pour les différents antibiotiques.
- une matrice (binaire) `numpy` appelée `X_gpa` codant pour la présence / absence de 16005 gènes.
- une matrice (binaire) `numpy` appelée `X_snps` codant pour la présence / absence de 72236 SNPs.
- une matrice (quantitative) `numpy` appelée `X_genexp` contenant les mesures d'expression de 6026 gènes.

La syntaxe pour charger le jeu de données est la suivante :

```
— import pickle
— file = open('dataset.pkl','rb')
— DATA = pickle.load(file)
— pheno = DATA['pheno']
— X_gpa = DATA['X_gpa']
— X_snps = DATA['X_snps']
— X_genexp = DATA['X_genexp']
```