PYTHON FOR DATA ANALYSIS

Projet final par Julien PYTEL et Audrey POISSON

Biodegradation Dataset

PRÉSENSATION DU DATA SET

L'objectif de ce dataset est de déterminer la biodégradabilité des produits chimiques sans avoir recours à des tests coûteux.

La composition de chaque molécule est détaillée (nombre d'atomes d'oxygène, nombre d'halogènes, pourcentage de carbone...) avant d'indiquer si oui ou non celles-ci sont facilement biodégradables.

SOMMAIRE

Notre projet est divisé en trois parties.

Tout d'abord nous avons importé, analysé et visualisé les données de notre data set.
Puis nous passons à la partie modélisation en essayant plusieurs algorithmes et en changeant les hyperparamètres.
Et enfin nous avons transformé ce modèle en API Flask.

- DATA VISUALISATION
- MODÉLISATION
- API

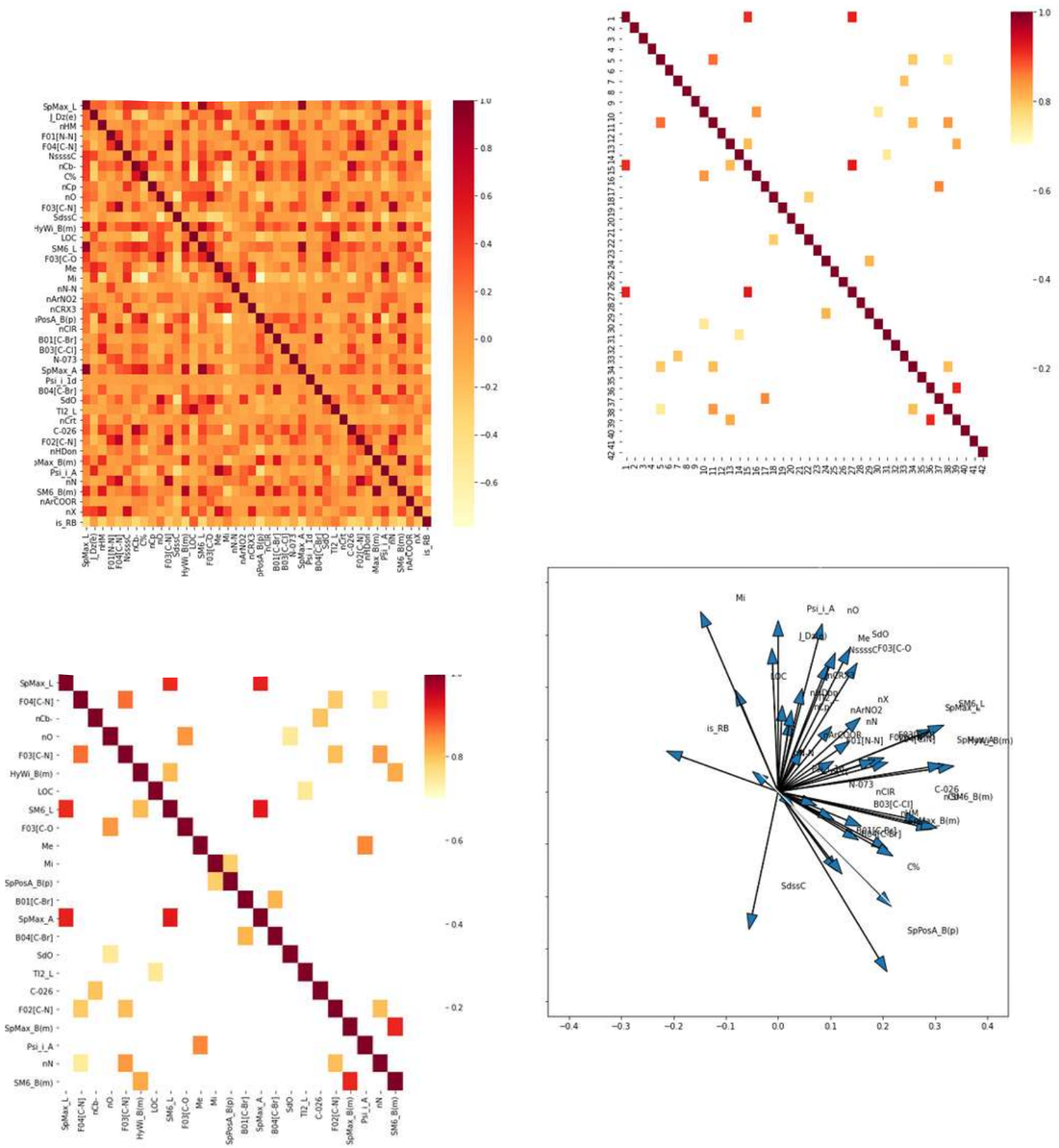
DATA VISUALISATION

VARIABLES

Pour commencer, nous importons le data set puis nous y ajoutons des noms de colonnes. Ensuite, nous récupérons les informations du data set, nombre de ligne, type de variable, descriptions des variables.

Ces informations alliées à celles du dataset nous permettent d'en tirer plusieurs choses. Tout d'abord, la majorité des variables sont numériques, à l'exception de la dernière (facilement biodégradable ou non), que l'on transformera par la suite. De plus trois des ces variables numériques sont binaires.
Ensuite, certaines variables suivent une nomenclature, c'est le cas de :

- B00[__] : pour les variables binaires (présence ou absence de), avec 00 la distance topologique et entre [__] la liaison chimique
- F00[__] : pour la fréquence d'une liaison, avec 00 la distance topologique et entre [__] la liaison chimique
- N____ : pour le nombre d'atomes de type ____.



CORRÉLATIONS

Nous construisons une matrice de corrélation pour mieux comprendre les résultats du dataset. On peut y voir la corrélation entre le résultat "facilement biodégradable ou non" et les différentes variables, mais aussi observer la redondance d'informations (corrélation des variables entre elle). Pour plus de lisibilité sur ce dernier paramètre, nous avons simplifié la matrice pour n'y afficher plus que les variables fortement corrélées. Enfin, pour observer cela sous une autre perspective nous avons également réalisé un cercle de corrélation.

MODÉLISATION

Sachant l'objectif du dataset, à savoir prédire si une molécule est facilement biodégradable ou non, nous nous sommes donc concentrés sur cette cible. Pour cela nous avons organisé notre travail en trois parties.

PRÉPROCESSING

DATASET SPLITTING

HYPERPARAMÈTRES

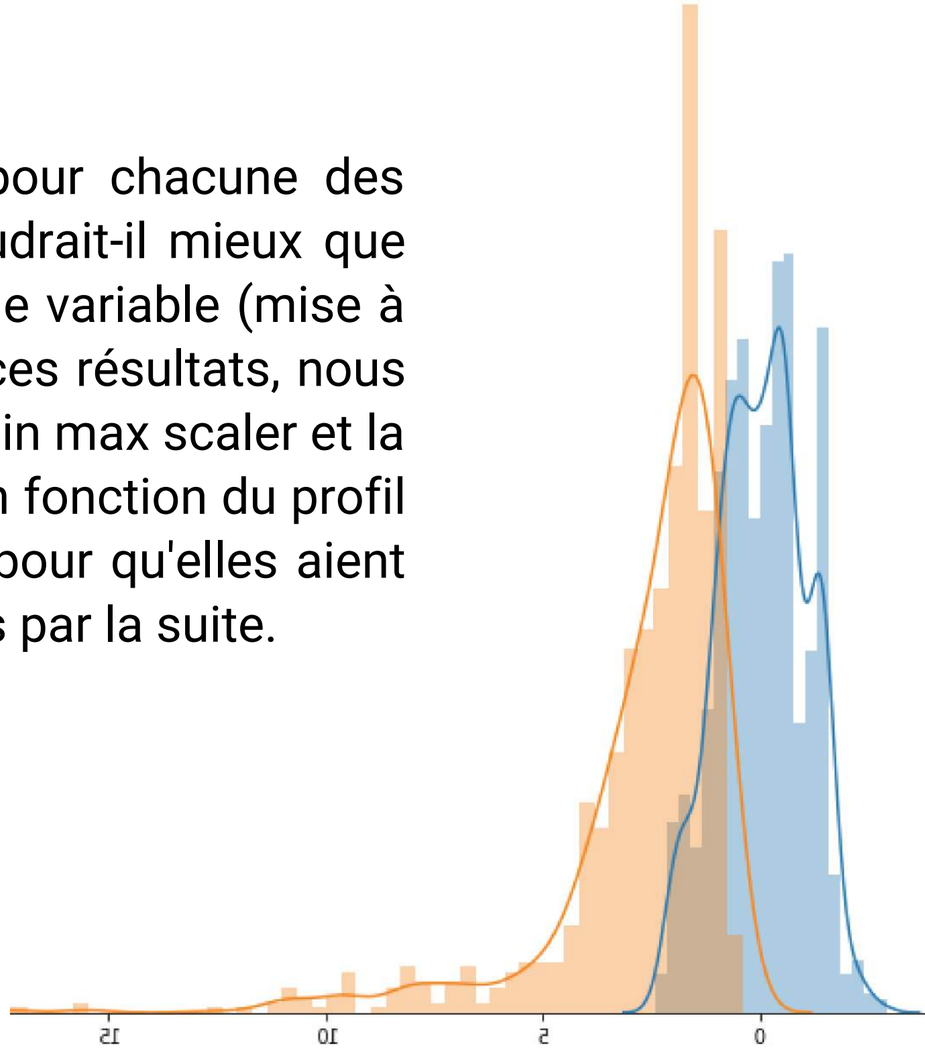
PRÉPROCESSING

Feature scaling

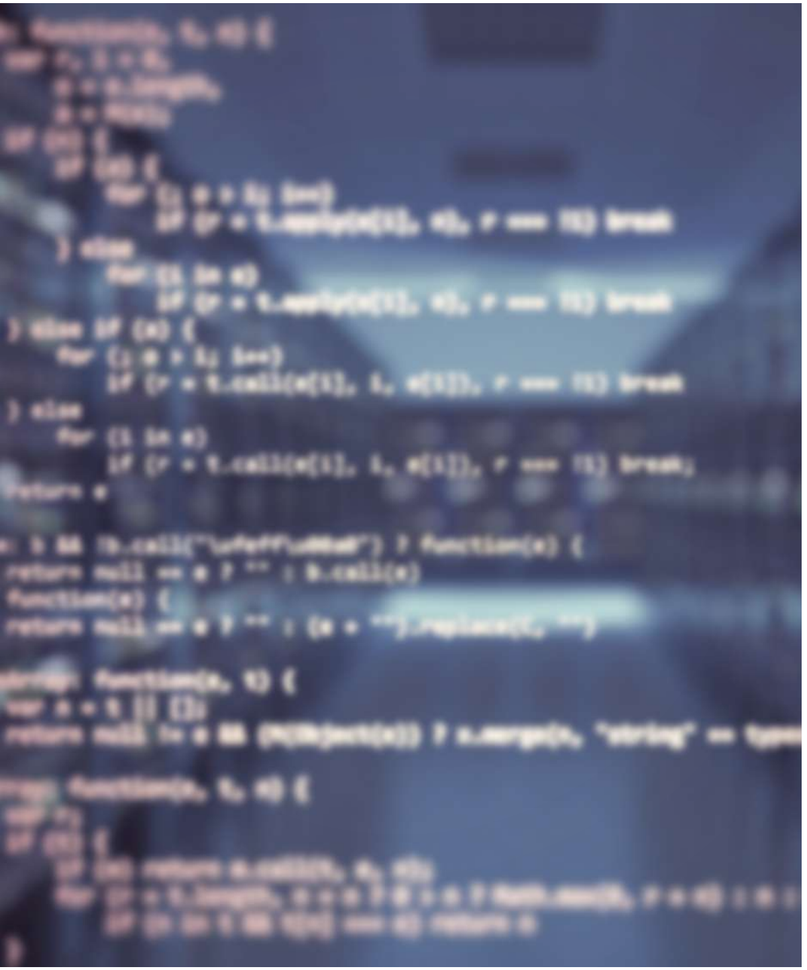
Tout d'abord, nous nous sommes posé la question suivante : pour chacune des variables de notre dataset, quelle méthode de feature scaling vaudrait-il mieux que l'on utilise ? Pour cela nous avons observé la distribution de chaque variable (mise à part la cible.) puis, après avoir tester la power transformation sur ces résultats, nous avons choisi d'appliquer au final 2 méthodes différentes : celle de min max scaler et la power transformation proposée par scikit learn, que l'on applique en fonction du profil de features. Cela nous permet de préparer au mieux les données pour qu'elles aient une forme plus adapté convenant aux algorithmes que l'on utiliseras par la suite.

Power transformation

En statistique, une power transformation est une famille de fonctions que l'on applique pour stabiliser la variance, rendre les données plus proches de la distribution normale et améliorer la validité des mesures.



DATASET SPLITTING



Dans cette partie, nous avons regardé la répartition des observations du dataset dans chacune des classes de la cible. Ceci pourra nous orienter dans notre choix de méthode de "splitting" du dataset. Il en ressort que 66.3% de nos données sont "NRB" ("Non Ready Biodegradable", c'est à dire pas facilement biodégradable).

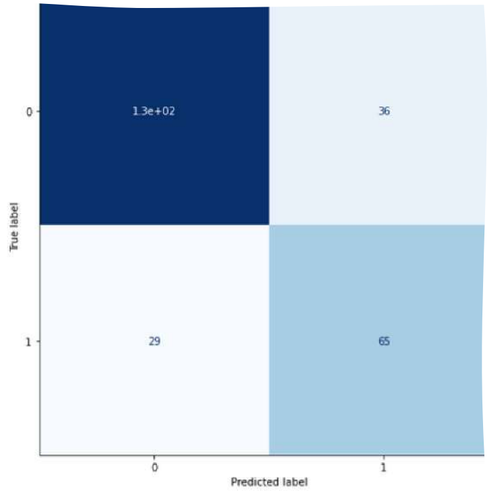
Nous pourrions donc faire le choix d'utiliser un jeu d'entrainement comprenant le même ratio de RB/NRB, cependant notre dataset étant assez réduit nous avons fait le choix de garder le ratio de base, l'aléatoire se chargeant de prendre au hasard des données du dataset d'origine pour les répartir dans deux jeux, un de test et un d'entrainement. Au final nous obtenons :

NOMBRE D'OBSERVATION DU JEU D'ENTRAINEMENT : 527
NOMBRE D'OBSERVATION DU JEU DE VALIDATION : 264
NOMBRE D'OBSERVATION DU JEU DE TEST : 264

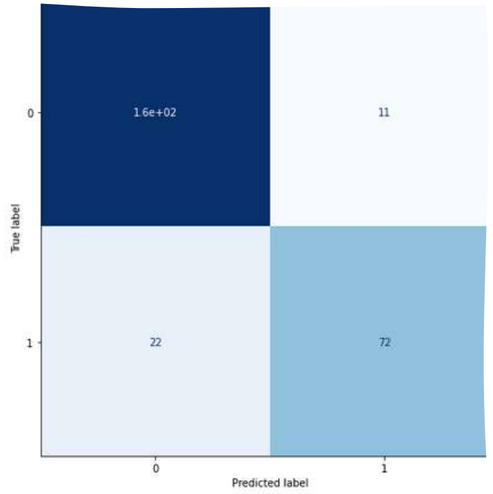
HYPERPARAMÈTRES

Pour finir sur la partie modélisation, nous décidons de mettre en œuvre les algorithmes suivants :

- Régression logistique
- Arbre de décision
- Foret Aléatoire
- Machines à vecteurs de support



Pour chaque algorithme nous réalisons des matrices de confusion. Dans notre cas, il faut faire attention, en effet, si l'on considère une molécule comme étant facilement biodégrable alors qu'elle ne l'ait pas, cela peut être dangereux. Les matrices nous permettent donc de choisir le plus efficacement possible notre modèle.



On retient donc au final celui qui a la meilleure performance globale (tout type d'erreur confondu) : une Fôret Aléatoire.

API

Pour cette dernière partie, nous avons utilisé Flask avec ngrok qui va nous permettre de rendre disponible notre API au delà du local, c'est à dire que ngrok fournira un accès indirect à notre ordinateur sur un port défini.

On a exploré les possibilités de stringification et destringification pour retenir ensuite les méthodes "to_json" et "read_json" de pandas avec le paramètre "orient = 'table' " ce qui permet d'envoyer une DataFrame pandas complète.