



Projet Python

Analyse du data set
QSAR
biodegradation

Présenté par
Grégoire Caurier &
Mattéo Berodier



Introduction

Dans une société où le développement durable est au cœur des intérêts communs, le caractère biodégradable des matériaux est de plus en plus recherché dans l'industrie et chez les consommateurs.

Ainsi lors de cette présentation, nous allons montrer notre travail sur la data base QSAR biodegradable ayant pour objectif d'identifier la biodégradabilité d'un matériau au regard des différents atomes qui le composent.

Sommaire

01

Data processing & visualization

- Description du dataset et de ses variables
- Corrélation et tri des variables
- Visualisation des variables

02

Modélisations

- Présentation des modélisations
- Comparaison des scores
- Recherche des hyperparamètres

03

Conclusion & API

- Meilleur modèle
- Flask

Variables

Notre data set regroupe 1055 produits chimiques avec, pour chacun, 41 attributs nous informant sur leur constitution moléculaire.

Le data set est également séparé en deux classes : les produits biodégradables (RB) et non-biodégradables (NRB)

Leading eigenvalue from Laplace matrix
Balaban-like index from Barysz matrix weighted by Sanderson electronegativity
Number of heavy atoms
Frequency of N-N at topological distance 1
Frequency of C-N at topological distance 4
Number of atoms of type ssssC
Number of substituted benzene C(sp₂)
Percentage of C atoms
Number of terminal primary C(sp₃)
Number of oxygen atoms
Frequency of C-N at topological distance 3
Sum of dssC E-states
Hyper-Wiener-like index (log function) from Burden matrix weighted by mass
Lopping centric index
Spectral moment of order 6 from Laplace matrix
Frequency of C - O at topological distance 3
Mean atomic Sanderson electronegativity (scaled on Carbon atom)
Mean first ionization potential (scaled on Carbon atom)
Number of N hydrazines
Number of nitro groups (aromatic)
Number of CRX3
Normalized spectral positive sum from Burden matrix weighted by polarizability
Number of circuits
Presence/absence of C - Br at topological distance 1
Presence/absence of C - Cl at topological distance 3
Ar₂NH / Ar₃N / Ar₂N-Al / R..N..R
Leading eigenvalue from adjacency matrix (Lovasz-Pelikan index)
Intrinsic state pseudoconnectivity index - type 1d
Presence/absence of C - Br at topological distance 4
Sum of d0 E-states
Second Mohar index from Laplace matrix
Number of ring tertiary C(sp₃)

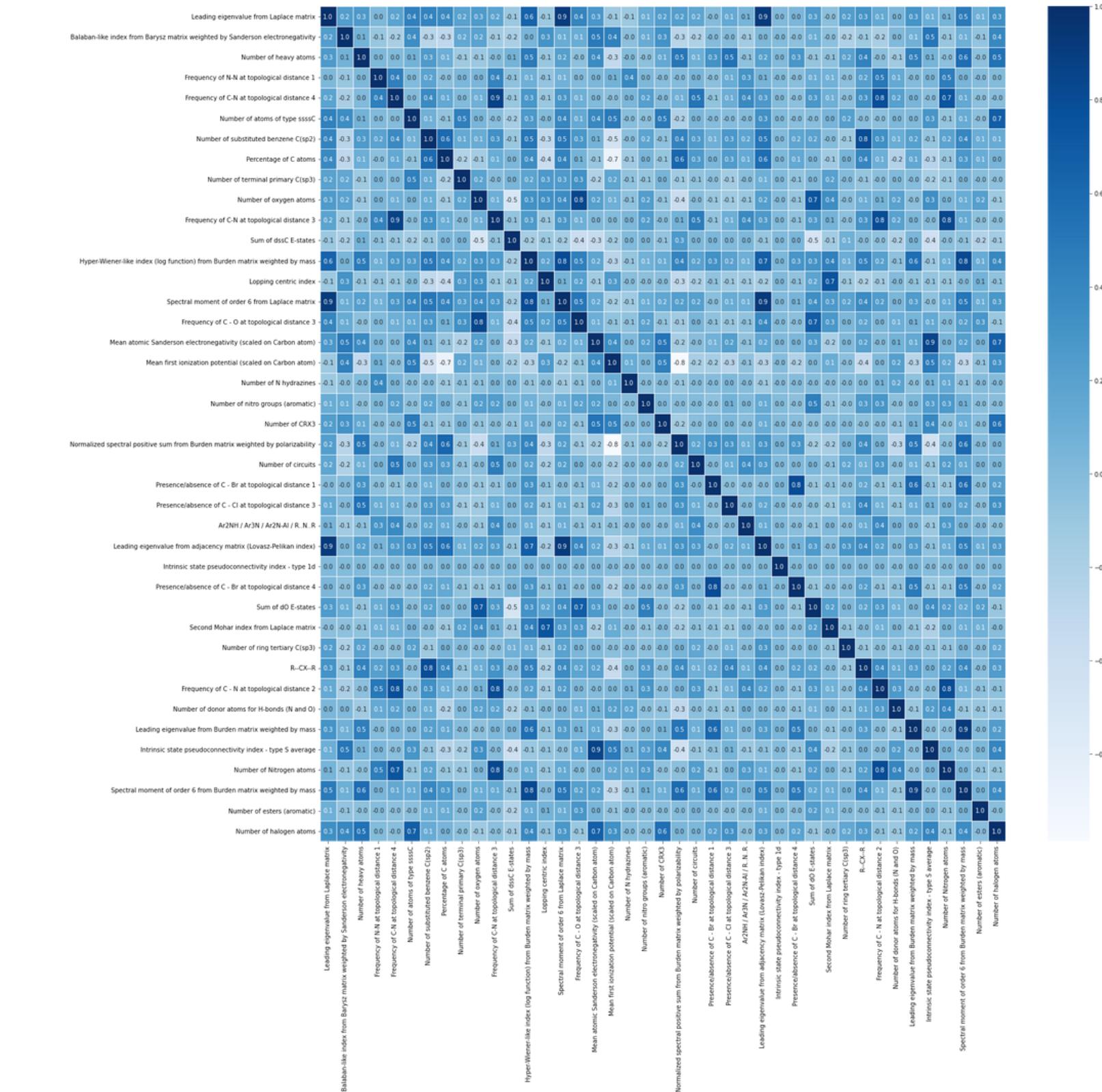
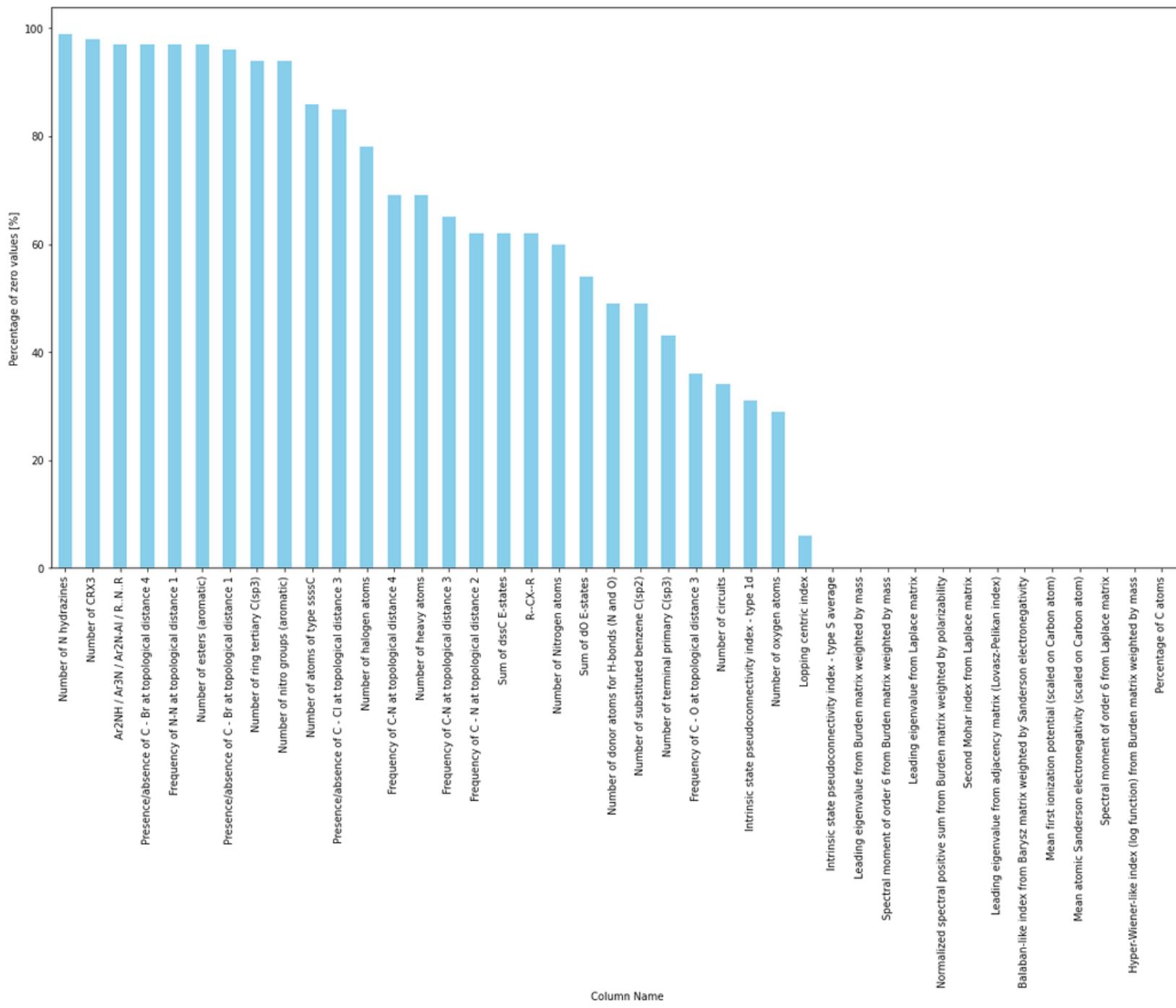
1. Visualisation des données

Afin d'étudier correctement le dataset, il convient de réfléchir et d'étudier si certaines variables ne polluent pas notre analyse du caractère biodégradable ou non d'un produit chimique donné.

Ainsi, nous avons d'abord analysé ce dernier de manière totale pour ensuite étudier de plus près les différentes variables pouvant être jugées inutiles lors de la première visualisation et vérifier si l'on peut les écarter lors du bilan finale

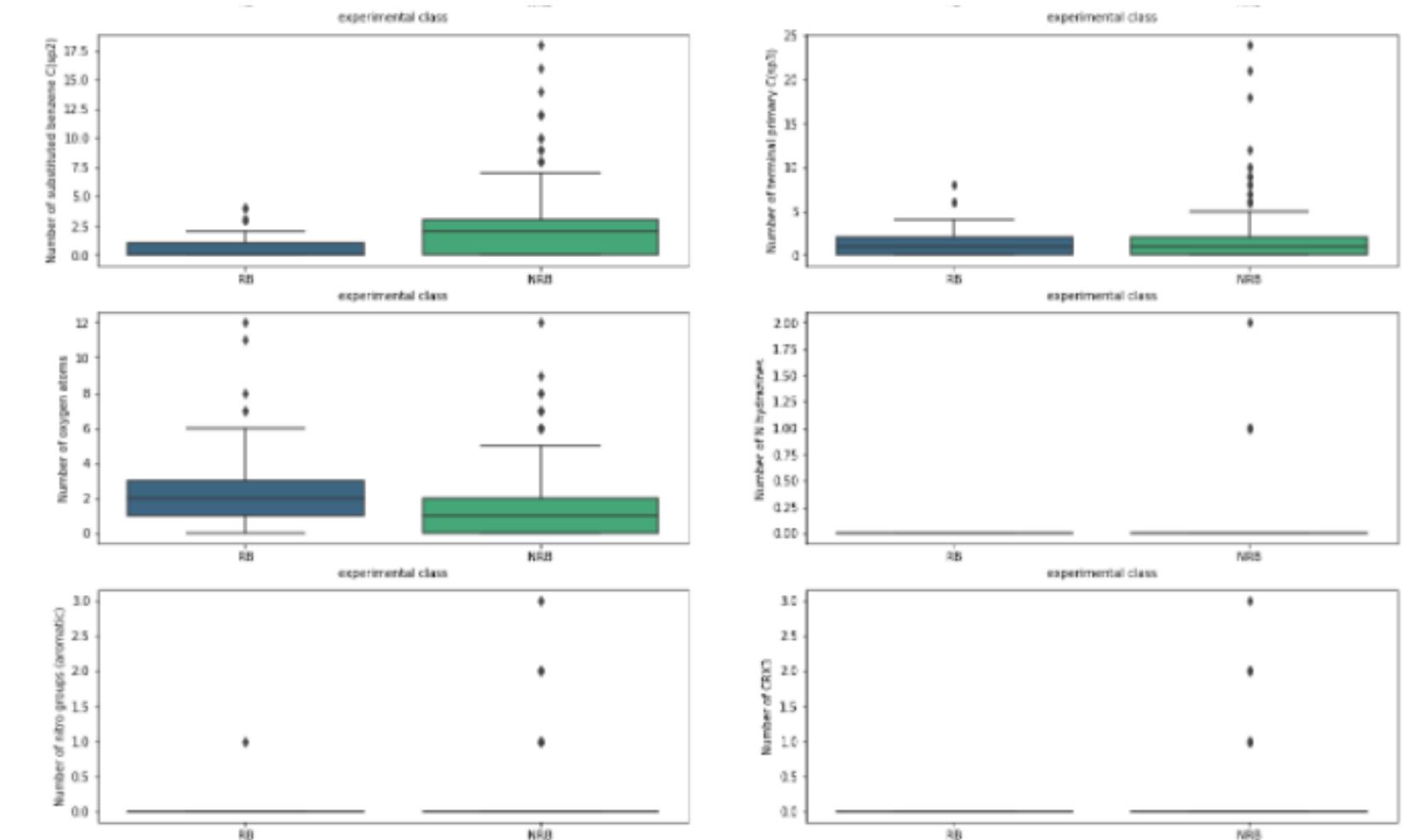
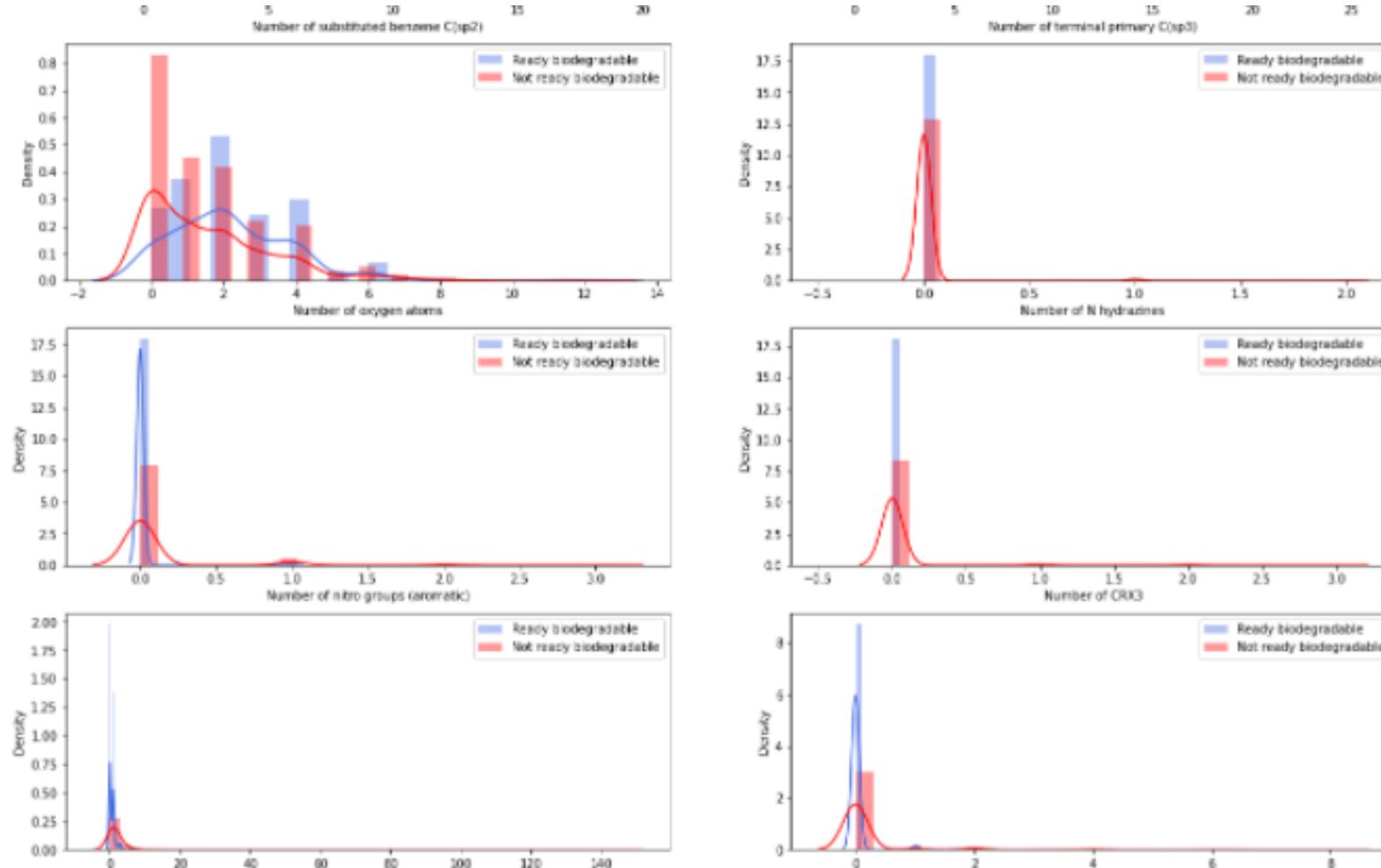
Histogramme des variables nulles

Matrice de corrélation



Visualisation générale

Analyse spécifique



Etude de l'influence de chaque attribut sur le caractère biodégradable des produits

Processing

A la suite de notre visualisation, nous avons chercher utiliser les ressources restantes à leurs plein potentiel.

C'est pourquoi nous avons normalisé les valeurs afin que celles qui soient encore nulles puissent être exploitées.

De plus, nous avons remplacé le caractère biodégradable (string) en binaire dans la même logique.

	Leading eigen...	Number of hea...	Frequency of ...	Frequency of ...
0	-1.123007433167 3774	-0.490511316718 802	-0.166770787933 46328	-0.4205426
1	-1.558691035797 951	-0.490511316718 802	-0.166770787933 46328	-0.4205426
2	-3.264813378872 298	-0.490511316718 802	-0.166770787933 46328	-0.4205426
3	-1.002187610589 1514	-0.490511316718 802	-0.166770787933 46328	-0.4205426
4	-1.002187610589 1514	-0.490511316718 802	-0.166770787933 46328	-0.4205426

2. Modélisation

Une fois le processing terminé, il nous est désormais possible de séparer nos données (apprentissage et test) pour pouvoir les tester avec des algorithmes différents qui nous semblent efficaces dans notre cas.

Ensuite nous cherchons à obtenir le meilleur score possible avec les modèles choisis, en utilisant Cross-val et Grid-Search, afin de trouver les meilleurs hyperparamètres.

```
knn_model = KNeighborsClassifier(n_jobs=-1)
pcpt_model = Perceptron(random_state=seed_num, n_jobs=-1)
lr_model = LogisticRegression(random_state=seed_num, n_jobs=-1)
svc_model = SVC(probability=True, random_state=seed_num)
gnb_model = GaussianNB()
rf_model = RandomForestClassifier(random_state = seed_num, n_jobs=-1)
mlp_model = MLPClassifier(random_state=seed_num)
gbc_model = GradientBoostingClassifier(random_state=seed_num)
ada_model = AdaBoostClassifier(random_state=seed_num)
bag_model = BaggingClassifier(random_state=seed_num, n_jobs=-1)
```

Comparaison des modèles

	Models object	Accuracy float64	Precision float64	Recall float64	F1 score float64	Average float64
	Multi-layer Perceptron ... 10% Perceptron 10% 8 others 80%	0.791666666666666 	0.606060606060606 	0.746987951807289 	0.744186046511647 	0.776442185231412
7	Multi-layer Perceptron...	0.920454545454545 454	0.8780487804878 049	0.8674698795180 723	0.8727272727272 728	0.8846751195469 238
2	Perceptron	0.8901515151515 151	0.78125	0.9036144578313 253	0.8379888268156 425	0.8532511999496 207
4	Support Vector Classifier	0.8977272727272 727	0.85	0.8192771084337 349	0.8343558282208 589	0.8503400523454 666
3	Logistic Regression	0.8863636363636 364	0.7912087912087 912	0.8674698795180 723	0.8275862068965 518	0.8431571284967 629
8	Gradient Boosting...	0.8863636363636 364	0.8192771084337 349	0.8192771084337 349	0.8192771084337 35	0.8360487404162 102
9	AdaBoost Classifier	0.8825757575757 576	0.8023255813953 488	0.8313253012048 193	0.8165680473372 782	0.8331986718783 01
6	Random Forest	0.8863636363636 364	0.8354430379746 836	0.7951807228915 663	0.8148148148148 148	0.8329505530111 753
10	Bagging Classifier	0.8787878787878 788	0.8493150684931 506	0.7469879518072 289	0.7948717948717 949	0.8174906734900 134
1	k-Nearest Neighbors	0.8522727272727 273	0.7391304347826 086	0.8192771084337 349	0.7771428571428 571	0.7969557819079 82
5	Gaussian Naive Bayes	0.7916666666666 666	0.6060606060606 061	0.9638554216867 47	0.7441860465116 28	0.7764421852314 12

Hyperparamètres

Après avoir comparé les différents algorithmes, nous avons sélectionné les 4 meilleurs pour écarter ceux qui ont un score moyen inférieur à 0,84.

Nous avons ensuite utilisé les fonctions `cross_val` et `GridSearch` pour trouver les hyperparamètres recherchés.

Cela nous a donc permis d'obtenir un modèle beaucoup plus fiable pour exprimer les résultats finaux entre ces 4 algorithmes

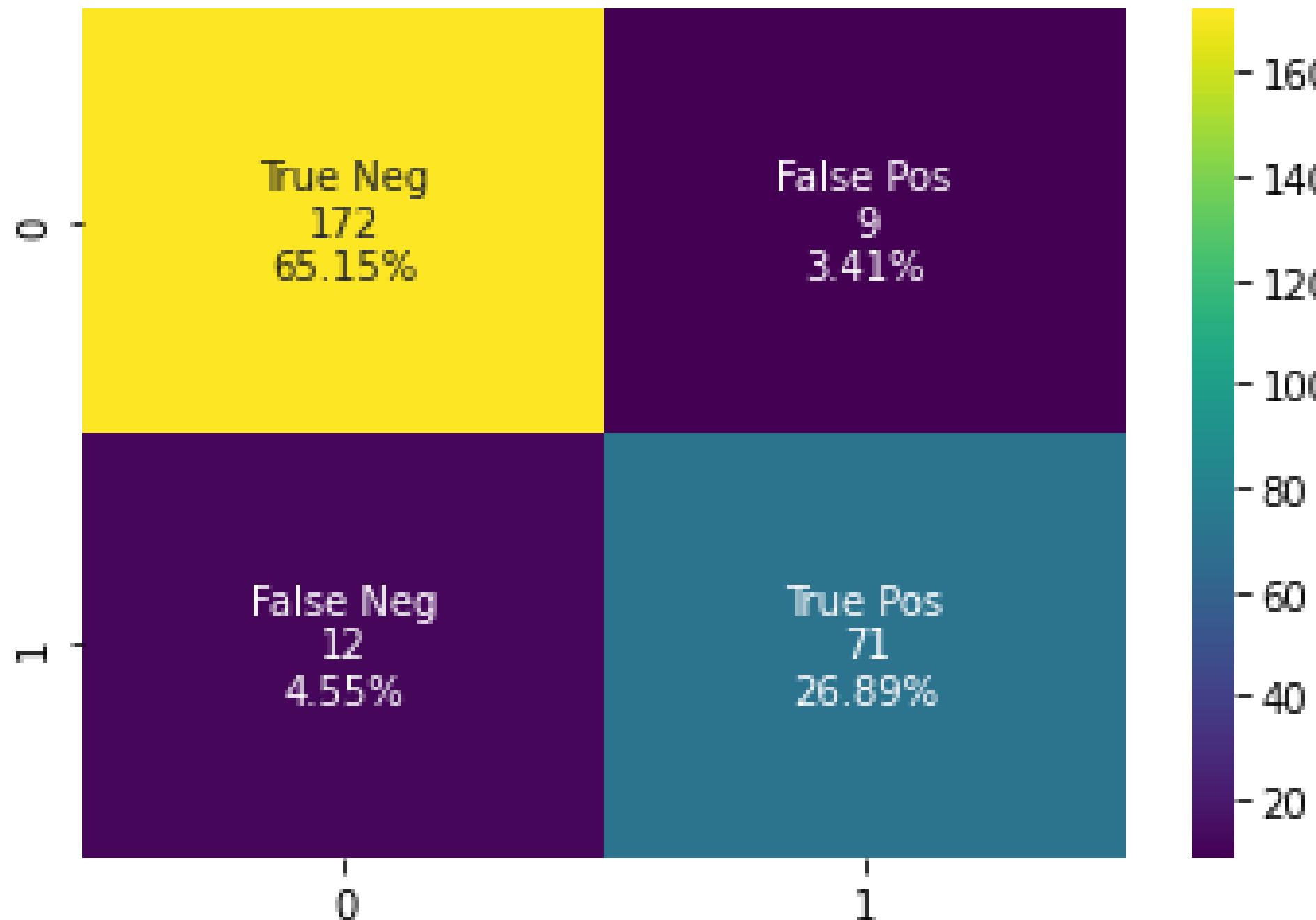
3. Conclusion

Après comparaison avec nos hyperparamètres, l'algorithme MLP s'avère être le plus efficace avec une précision de 0,92. Et notre nouvelle matrice de confusion obtenue confirme ces résultats.

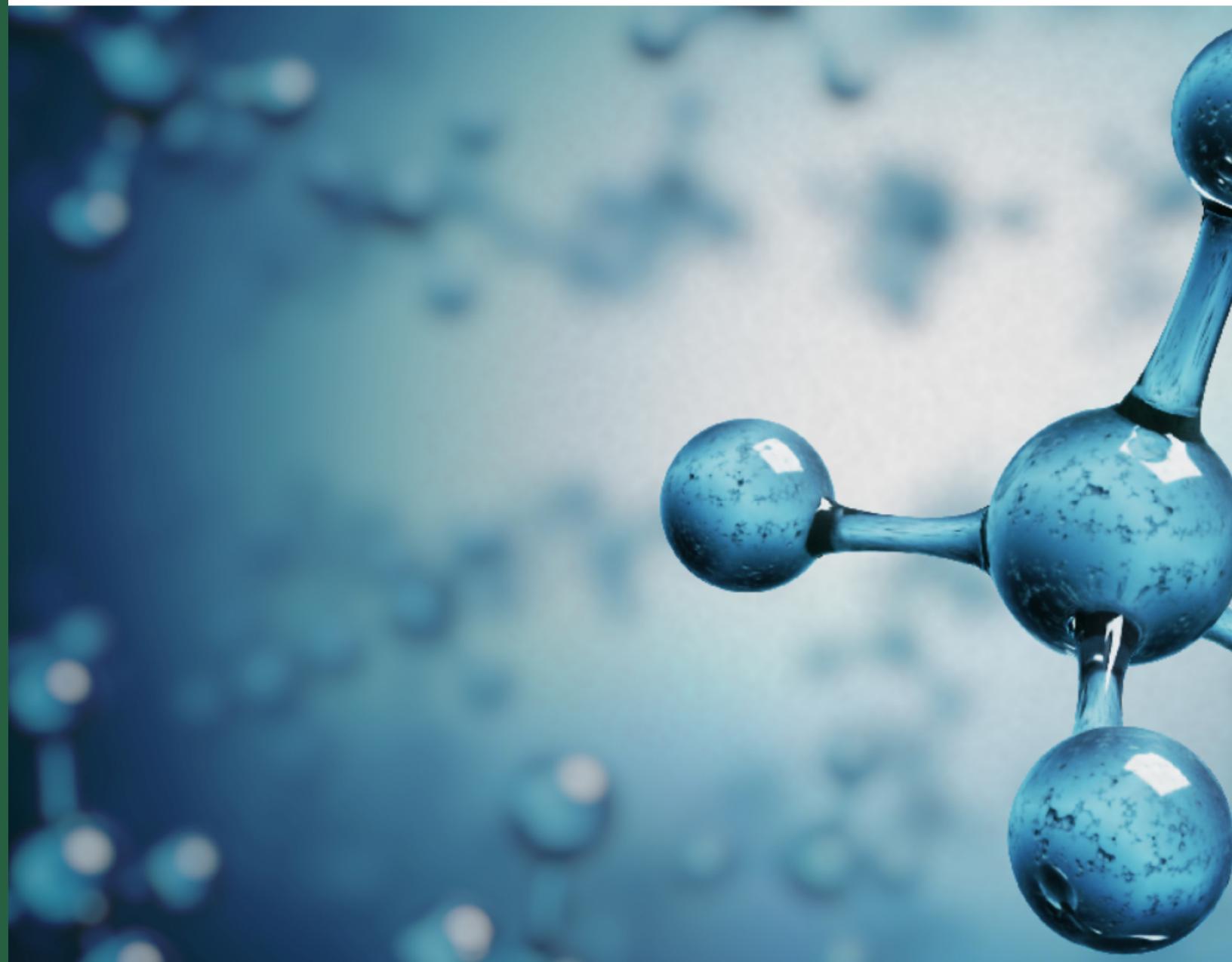
Afin de mettre en valeur l'utilisation de notre modèle, nous avons crée une API avec flask pour que n'importe quel utilisateur puisse savoir le caractère biodégradable de son produit en connaissant ses paramètres.

	Models object	Accuracy float...	Precision floa...	Recall float64	F1 score float...	Average float64
1	Multi-layer Perceptron...	0.92045454545454	0.8875	0.8554216867469879	0.8711656441717791	0.883635469093328
2	Support Vector Classifier	0.9053030303030303	0.8452380952380952	0.8554216867469879	0.8502994011976047	0.8640655533714295
3	Logistic Regression	0.8901515151515151	0.8	0.8674698795180723	0.8323699421965319	0.8474978342165298
4	Perceptron	0.8257575757575758	0.7032967032967034	0.7710843373493976	0.7356321839080461	0.7589427000779307

Nouvelle matrice de confusion



API avec Flask



Predict the biodegradation x +
127.0.0.1:5000

Leading eigenvalue from Laplace matrix :

Number of heavy atoms :

Predict the biodegradation x +
127.0.0.1:5000

Number of nitro groups (aromatic) :

Normalized spectral positive sum from Burden matrix weighted by polarizability :

Number of circuits :

Presence/absence of C - Br at topological distance 1 :

Presence/absence of C - Cl at topological distance 3 :

Leading eigenvalue from adjacency matrix (Lovasz-Pelikan index) :

Second Mohar index from Laplace matrix :

Number of ring tertiary C(sp³) :

R-CX-R :

Frequency of C - N at topological distance 2 :

Leading eigenvalue from Burden matrix weighted by mass :

Intrinsic state pseudoconnectivity index - type S average :

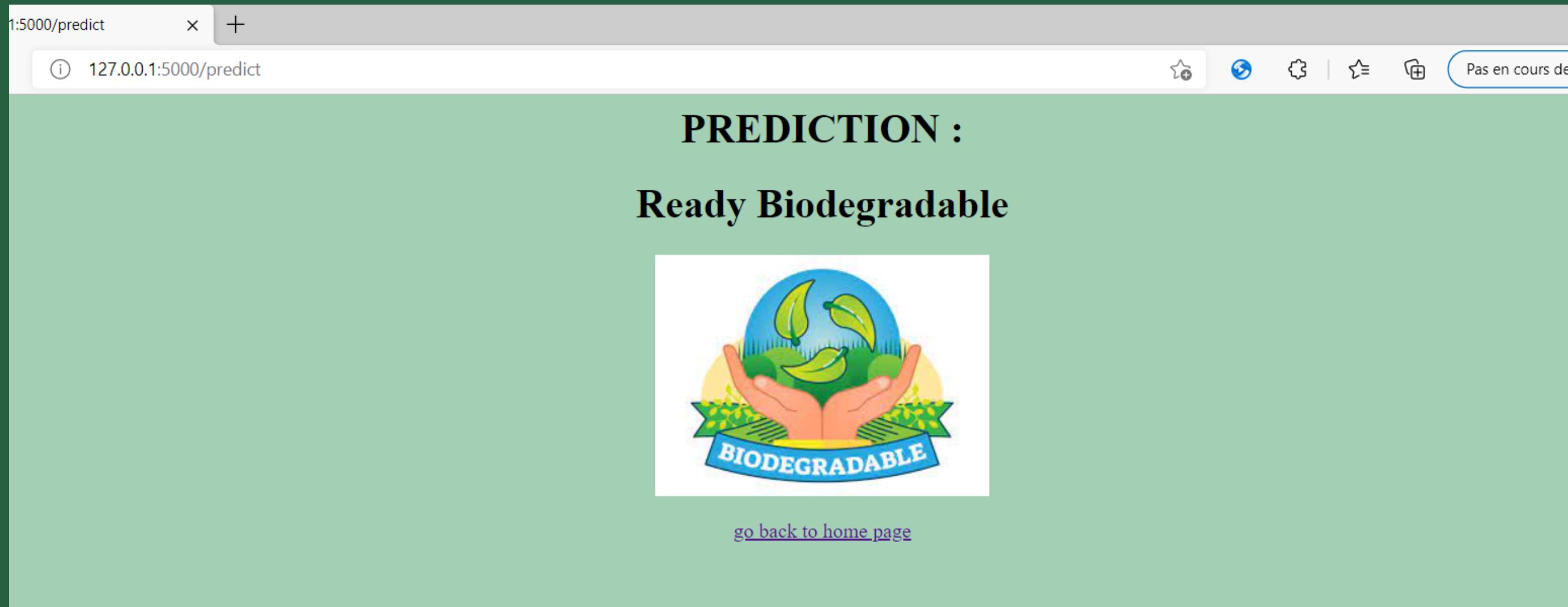
Number of Nitrogen atoms :

Spectral moment of order 6 from Burden matrix weighted by mass :

Number of esters (aromatic) :

Number of halogen atoms :

Exemple de résultat de l'API



Merci pour votre attention !