

Rapport Projet Python for data analyst

Jean-louis Delebecque – Simon Hervé

Dataset (1/2)



UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems

[About](#) [Citation Policy](#) [Donate a Data Set](#) [Contact](#)

☒ Repository ☐ Web



[View ALL Data Sets](#)

QSAR biodegradation Data Set

Download: [Data Folder](#), [Data Set Description](#)

Abstract: Data set containing values for 41 attributes (molecular descriptors) used to classify 1055 chemicals into 2 classes (ready and not ready biodegradable).

Data Set Characteristics:	Multivariate	Number of Instances:	1055	Area:	N/A
Attribute Characteristics:	Integer, Real	Number of Attributes:	41	Date Donated	2013-06-21
Associated Tasks:	Classification	Missing Values?	N/A	Number of Web Hits:	50650

Source:

Kamel Mansouri, Tine Ringsted, Davide Ballabio (davide.ballabio@unimib.it), Roberto Todeschini, Viviana Consonni, Milano Chemometrics and QSAR Research Group (<http://michem.disat.unimib.it/chm/>), Università degli Studi Milano - Bicocca, Milano (Italy)

Data Set Information:

The QSAR biodegradation dataset was built in the Milano Chemometrics and QSAR Research Group (Università degli Studi Milano - Bicocca, Milano, Italy). The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under Grant Agreement n. 238701 of Marie Curie ITN Environmental Chemoinformatics (ECO) project. The data have been used to develop QSAR (Quantitative Structure Activity Relationships) models for the study of the relationships between chemical structure and biodegradation of molecules. Biodegradation experimental values of 1055 chemicals were collected from the webpage of the National Institute of Technology and Evaluation of Japan (NITE). Classification models were developed in order to discriminate ready (356) and not ready (699) biodegradable molecules by means of three different modelling methods: k Nearest Neighbours, Partial Least Squares Discriminant Analysis and Support Vector Machines. Details on attributes (molecular descriptors) selected in each model can be found in the quoted reference: Mansouri, K., Ringsted, T., Ballabio, D., Todeschini, R., Consonni, V. (2013). Quantitative Structure - Activity Relationship models for ready biodegradability of chemicals. Journal of Chemical Information and Modeling, 53, 867-878.

Nous avons récupéré ce dataset sur le site UCI. On peut remarquer dans la description, que le problème est une classification binaire. De plus, ce dataset est composé de 1055 lignes et 42 colonnes avec la cible incluse.

Dataset (2/2)

Attribute Information:

41 molecular descriptors and 1 experimental class:
1) SpMax_L: Leading eigenvalue from Laplace matrix
2) J_Dz(e): Balaban-like index from Barysz matrix weighted by Sanderson electronegativity
3) nHM: Number of heavy atoms
4) F01[N-N]: Frequency of N-N at topological distance 1
5) F04[C-N]: Frequency of C-N at topological distance 4
6) NssssC: Number of atoms of type ssssC
7) nCb-: Number of substituted benzene C(sp2)
8) C%: Percentage of C atoms
9) nCp: Number of terminal primary C(sp3)
10) nO: Number of oxygen atoms
11) F03[C-N]: Frequency of C-N at topological distance 3
12) SdssC: Sum of dssC E-states
13) HyWi_B(m): Hyper-Wiener-like index (log function) from Burden matrix weighted by mass
14) LOC: Lopping centric index
15) SM6_L: Spectral moment of order 6 from Laplace matrix
16) F03[C-O]: Frequency of C - O at topological distance 3
17) Me: Mean atomic Sanderson electronegativity (scaled on Carbon atom)
18) Mi: Mean first ionization potential (scaled on Carbon atom)
19) nN-N: Number of N hydrazines
20) nArNO2: Number of nitro groups (aromatic)
21) nCRX3: Number of CRX3
22) SpPosA_B(p): Normalized spectral positive sum from Burden matrix weighted by polarizability
23) nCIR: Number of circuits
24) B01[C-Br]: Presence/absence of C - Br at topological distance 1
25) B03[C-Cl]: Presence/absence of C - Cl at topological distance 3
26) N-073: Ar2NH / Ar3N / Ar2N-AI / R...R
27) SpMax_A: Leading eigenvalue from adjacency matrix (Lovasz-Pelikan index)
28) Psi_i_1d: Intrinsic state pseudoconnectivity index - type 1d
29) B04[C-Br]: Presence/absence of C - Br at topological distance 4
30) SdO: Sum of dO E-states
31) TI2_L: Second Mohar index from Laplace matrix
32) nCrT: Number of ring tertiary C(sp3)
33) C-026: R--CX--R
34) F02[C-N]: Frequency of C - N at topological distance 2
35) nHDon: Number of donor atoms for H-bonds (N and O)
36) SpMax_B(m): Leading eigenvalue from Burden matrix weighted by mass
37) Psi_i_A: Intrinsic state pseudoconnectivity index - type S average
38) nN: Number of Nitrogen atoms
39) SM6_B(m): Spectral moment of order 6 from Burden matrix weighted by mass
40) nArCOOR: Number of esters (aromatic)
41) nX: Number of halogen atoms
42) experimental class: ready biodegradable (RB) and not ready biodegradable (NRB)

Sur la page du dataset nous avons une description de chaque variable. On remarque que la variable cible est experimental class, qui correspond à une molécule biodégradable ou non.

Il s'agit donc d'un problème de classification binaire avec des données supervisées.

Exploration (1/3)

La première chose que l'on remarque, c'est que toutes les colonnes sont numériques hormis la colonne cible.

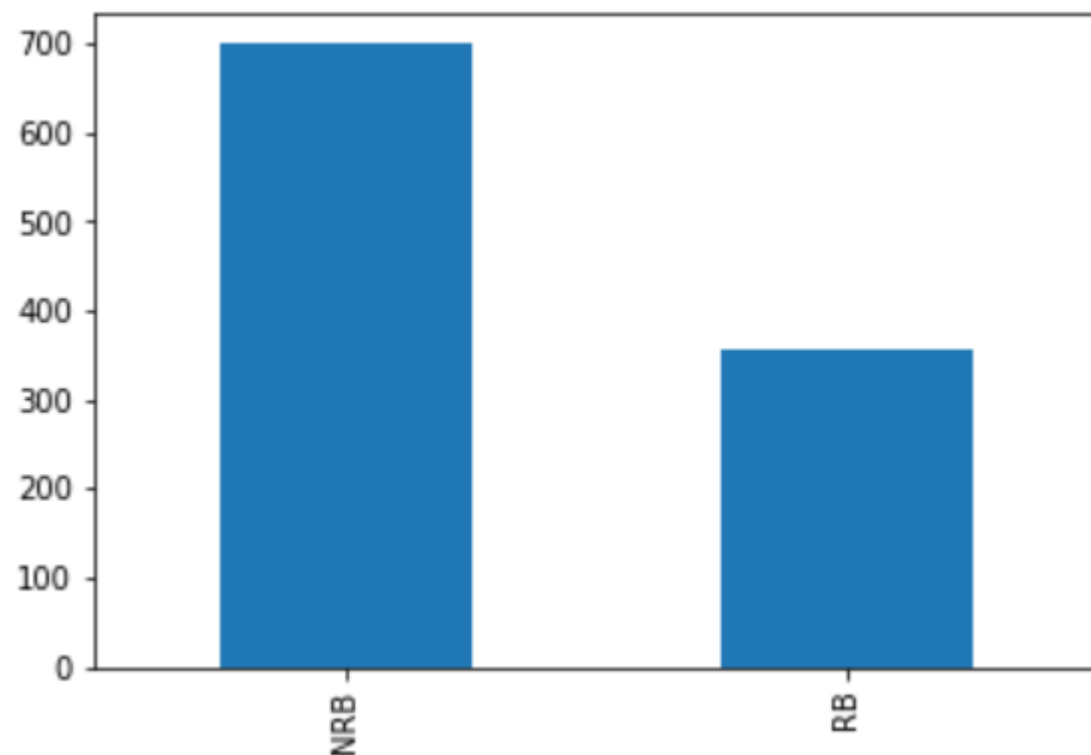
Cela implique que nous ne pourrions pas faire de feature engineering dans ce projet.

En effet ils nous aient impossible de réellement donner un sens à toutes ces features (ce sont des données chimiques et physiques) et il y en a déjà beaucoup. Nous n'avons donc pas crée de variable.

#	Column	Non-Null Count	Dtype
0	SpMaxL	1055 non-null	float64
1	JDze	1055 non-null	float64
2	nHM	1055 non-null	int64
3	F01NN	1055 non-null	int64
4	F04CN	1055 non-null	int64
5	NssssC	1055 non-null	int64
6	nCb	1055 non-null	int64
7	C%	1055 non-null	float64
8	nCp	1055 non-null	int64
9	nO	1055 non-null	int64
10	F03CN	1055 non-null	int64
11	SdssC	1055 non-null	float64
12	HyWiBm	1055 non-null	float64
13	LOC	1055 non-null	float64
14	SM6L	1055 non-null	float64
15	F03CO	1055 non-null	int64
16	Me	1055 non-null	float64
17	Mi	1055 non-null	float64
18	nNN	1055 non-null	int64
19	nArNO2	1055 non-null	int64
20	nCRX3	1055 non-null	int64
21	SpPosABp	1055 non-null	float64
22	nCIR	1055 non-null	int64
23	B01CBr	1055 non-null	int64
24	B03CCl	1055 non-null	int64
25	N073	1055 non-null	int64
26	SpMaxA	1055 non-null	float64
27	Psii1d	1055 non-null	float64
28	B04CBr	1055 non-null	int64
29	SdO	1055 non-null	float64
30	TI2L	1055 non-null	float64
31	nCrt	1055 non-null	int64
32	C026	1055 non-null	int64
33	F02CN	1055 non-null	int64
34	nHDon	1055 non-null	int64
35	SpMaxBm	1055 non-null	float64
36	PsiiA	1055 non-null	float64
37	nN	1055 non-null	int64
38	SM6Bm	1055 non-null	float64
39	nArCOOR	1055 non-null	int64
40	nX	1055 non-null	int64
41	experimentalclass	1055 non-null	object

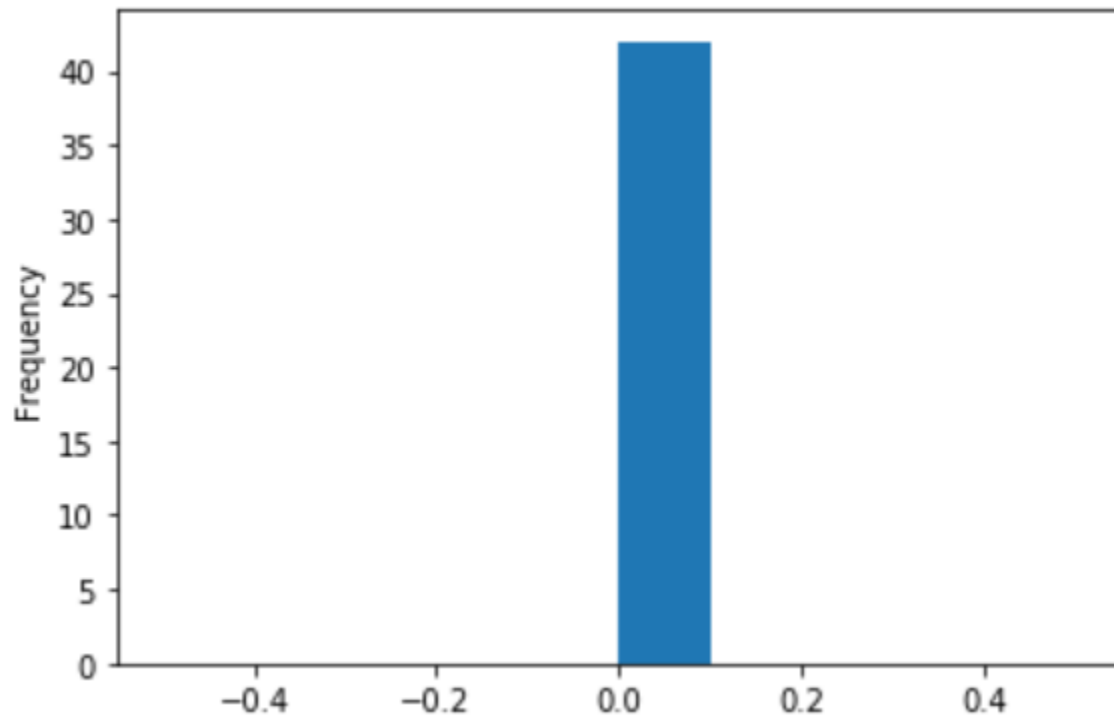
dtypes: float64(17), int64(24), object(1)

Exploration (2/3)



On remarque que le dataset n'est pas équilibré et qu'il y a une majorité de NRB, qui correspond aux molécules non biodégradable.

Exploration (3/3)

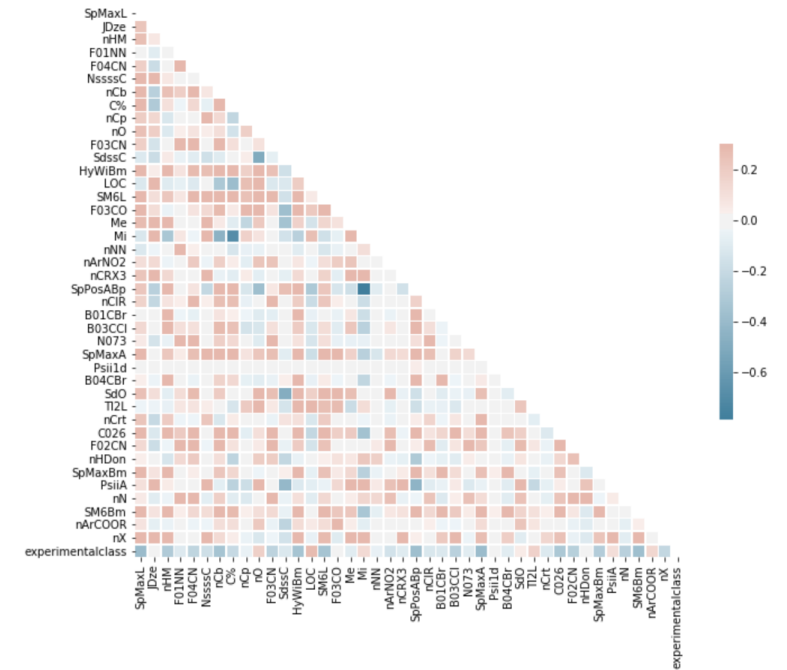


On remarque qu'il n'y a pas de données manquantes, ce qui implique qu'il n'y a pas de nettoyage à faire. Cela est plutôt logique, dans la mesure où le dataset vient d'un site universitaire.

Modélisation (1/4)

On a commencer notre modélisation par une matrice de corrélation. En effet, nous avons après transformation uniquement des colonnes numériques. Mais l'analyse de celle-ci ne donne rien. On fera alors de la feature selection.

Notre stratégie est de mettre en concurrence plusieurs algorithmes scikit-learn, qui sont au nombre de 11.



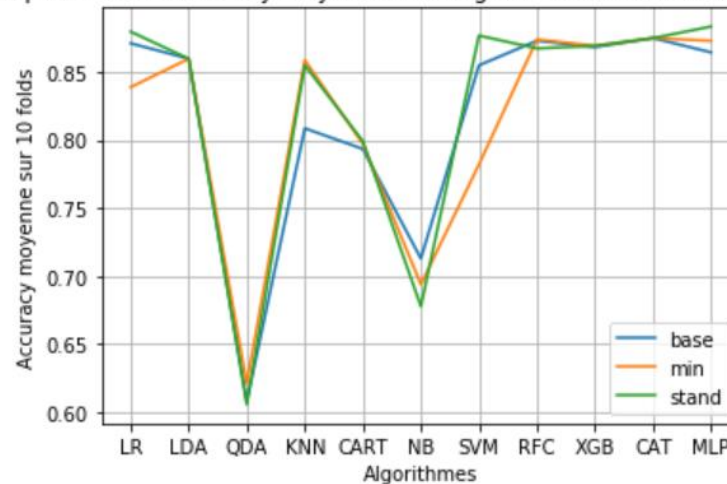
```
models.append(("LR", LogisticRegression(solver="lbfgs", multi_class="auto")))
models.append(("LDA", LinearDiscriminantAnalysis()))
models.append(("QDA", QuadraticDiscriminantAnalysis()))
models.append(("KNN", KNeighborsClassifier()))
models.append(("CART", DecisionTreeClassifier()))
models.append(("NB", GaussianNB()))
models.append(("SVM", SVC(gamma="auto")))
models.append(("RFC", RandomForestClassifier()))
models.append(("XGB", XGBClassifier()))
models.append(("CAT", CatBoostClassifier()))
models.append(("MLP", MLPClassifier()))
```

Modélisation (2/4)

On test chaque algorithme, en faisant de la cross-validation sur 10 folds.

On répète cette tâche trois fois. En effet, nous testons les algorithmes avec une normalisation et une standardisation en plus. On sait que certain algorithme sont plus sensibles que d'autre lorsque l'on les normalise.

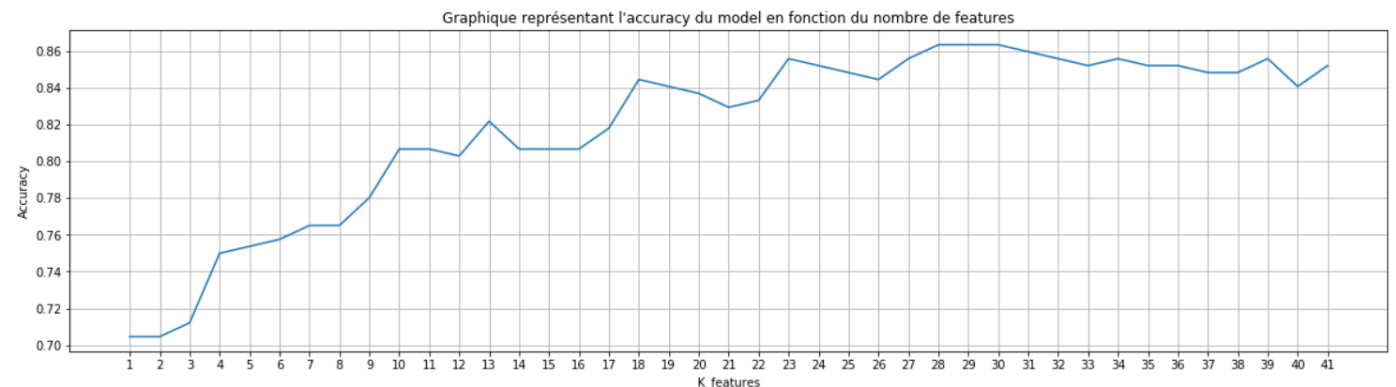
Graphique représentant l'accuracy moyenne d'un algorithme en fonction de son preprocessing



Modélisation (3/4)

Du graphique ci-dessus, on sélectionne trois algorithmes qui sont LR, SVM et MLP avec une standardisation car c'est ceux avec les meilleures moyennes de cross-validation.

Ensuite nous faisons de la feature selection. L'idée est de voir les performances des algorithmes en faisant varier le nombre de colonnes. En effet, un modèle avec trop d'information peut être moins performant. Voir exemple ci-dessous avec LR.

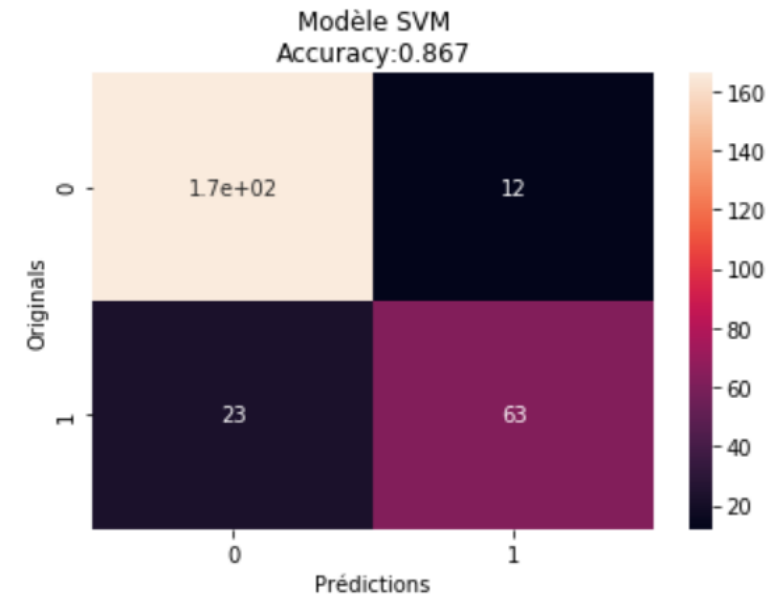


Modélisation (4/4)

Une fois que nous avons une optimisation du nombre de colonnes pour chaque algorithme, nous faisons un GridSearchCV pour les optimiser.

Ensuite on fait une matrice de confusion pour voir les résultats des algorithmes avec les meilleurs paramètres.

Au final nous choisissons SVM.



Exportation

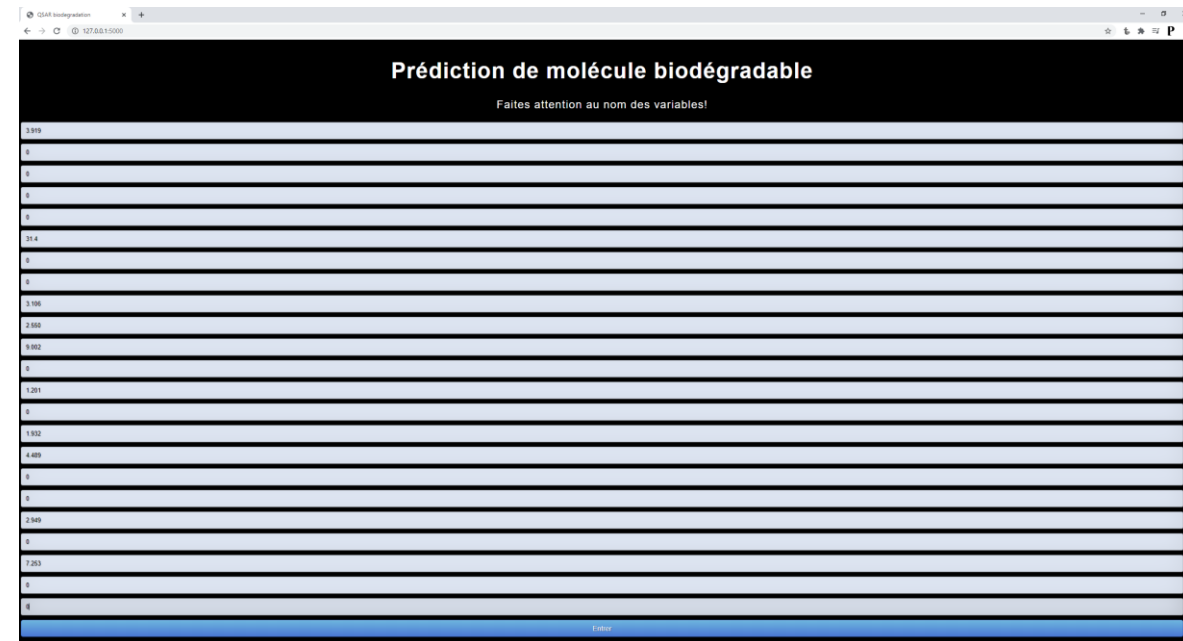
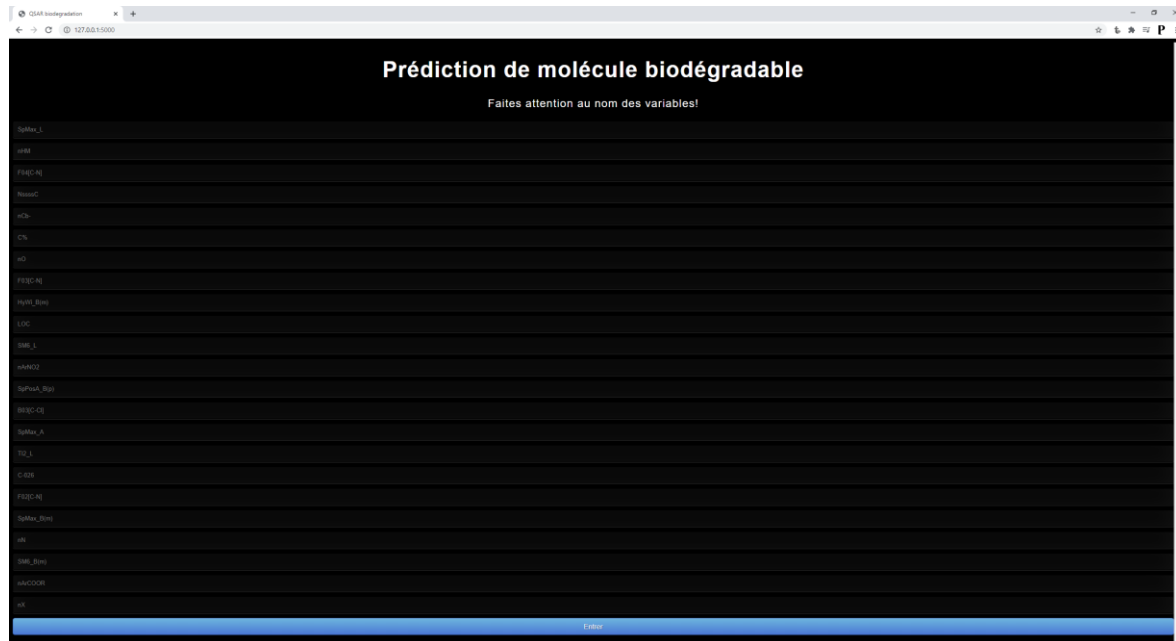
Avant d'exporter nous recalculons un modèle SVM sur l'ensemble des données avec les meilleurs hyper-paramètres que nous avons trouvé et une standardisation.

Ensuite nous enregistrons le model et le scaler avec Pickle.

```
pickle.dump(model, open('final_model.pickle', 'wb'))  
pickle.dump(scaler, open("scaler.pickle", "wb"))
```

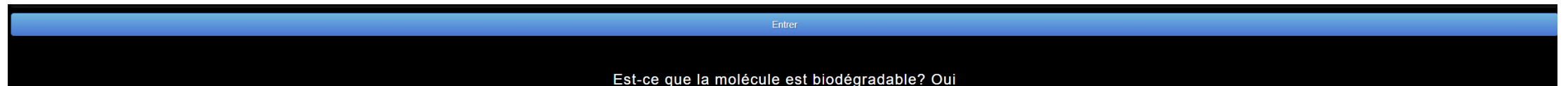
API (1/2)

Nous avons créé une application Flask, où l'utilisateur entre certaines variables de sa molécule. En effet, notre modèle fonctionne sur 23 features.



API (2/2)

Une fois que l'utilisateur a entré ses valeur, il lui reste à cliquer sur Entrer pour savoir si sa molécule est biodégradable ou non.



Entrer

Est-ce que la molécule est biodégradable? Oui