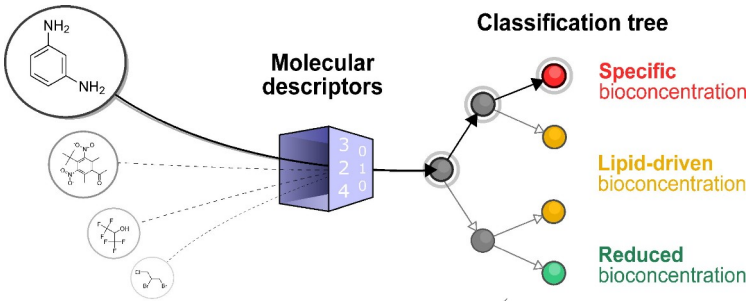
Étude sur le facteur de bioconcentraion (BCF, poisson) et les classes mécanistes pour la modélisation QSAR.

Réalisateurs :

- ZAOUAM Sirageddine
- FARHI Abdellah
- HADJ-SAID Mohand



Nous allons présenter ici un exemple d'utilisation de Python à travers une simple étude d'un ensemble de données de BCF organisé manuellement pour 779 produits chimiques. Le but est de prédire de quelle **Class** (1, 2 ou 3) appartient le composant chimique : **problème de classification.**
Source : https://archive.ics.uci.edu/ml/datasets/QSAR+Bioconcentration+classes+dataset

1. Présentation des données :

- Nombre des produits chimiques : 779
- Nombre de variables : 12 quantitatives (explicatives) et 2 qualitatives (expliquées)
  - ✔ **3 Identificateurs composés** : CAS (Identifiant du produit chimique), SMILES et SET (fractionnement Train/Test : 584 composants chimiques pour le train et 195 pour le test)
  - ✔ **9 descripteurs moléculaires (variables indépendantes) :** nHM, piPC09, PCD, X2AV, MlogP, ON1V, N0-72,B02[C-N], F04[C-0].
  - ✔ **2 réponses expérimentales : Facteur de bioconcentration (BCF)** dans les unités de notation (régression) - **Classe de bioaccumulation** (trois classes)

Dans notre cas on va classifier les produits chimiques selon leur classes de bioaccumulation (1, 2 ou 3) : c'est donc de prédire si un produit chimique est de classe 1 (est principalement stocké dans les tissus lipidiques) , de classe 2 (s'il a d'autres sites de stockages) et de la classe 3 (s'il est métabolisé ou éliminé). Ici notre variable explicative à prédire est **Class.**

2. Analyse univariée :

Nous allons commencer par représenter l'histogramme puis les boîtes à moustaches, pour chacune des quatres variables quantitatives.

2.1 Histogramme :

La distribution de **Class** n'est pas uniforme sur l'ensemble des données

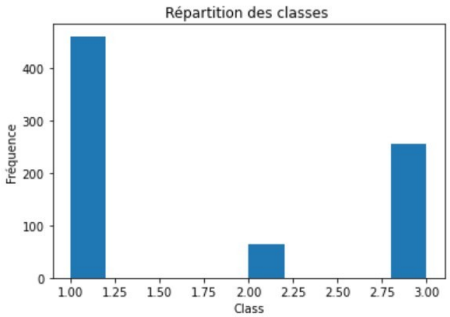
Il ya 460 produits chimiques de classe 1, 64 de classe 2 et 255 de classe 3

```
data_class1 = data[data.Class==1]
data_class1.shape
```

```
(460, 14)
```

```
plt.hist(data['Class'])
plt.xlabel("Class")
plt.ylabel("Fréquence")
plt.title("Répartition des classes ")
```

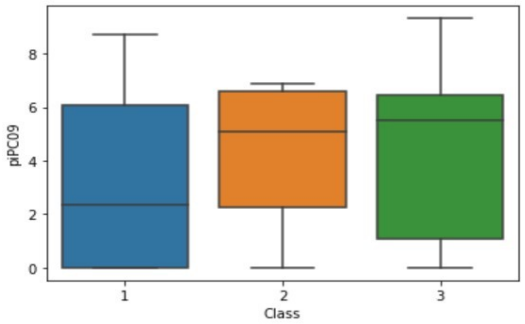
```
Text(0.5, 1.0, 'Répartition des classes ')
```



2.1 Boites à moustaches :

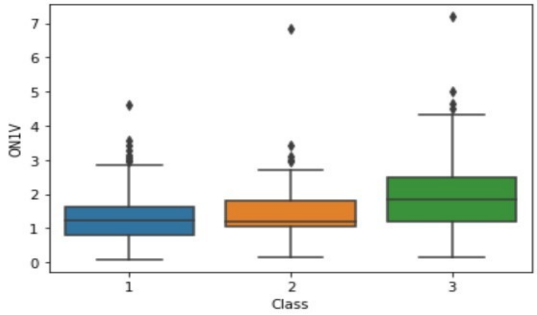
```
sns.boxplot(x="Class", y="piPC09", data=data)
```

```
<AxesSubplot:xlabel='Class', ylabel='piPC09'>
```




```
sns.boxplot(x="Class", y="ON1V", data=data)
```

```
<AxesSubplot:xlabel='Class', ylabel='ON1V'>
```



la moyenne des valeurs de piPC09 pour la classe 1 est celle représenter par la médiane qui est se trouve entre 2 et 2.5.

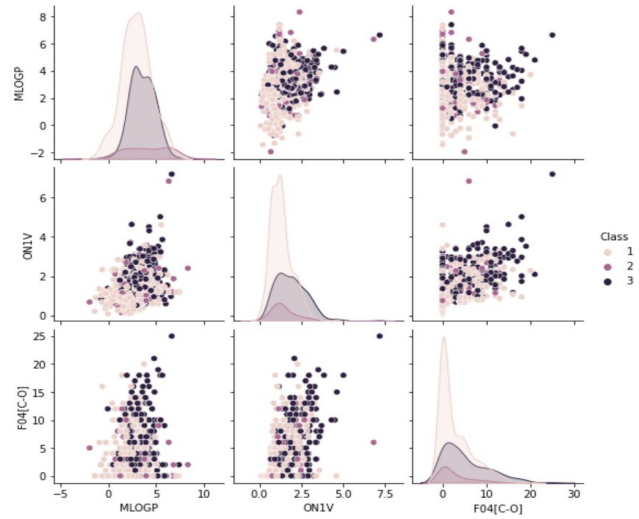
3. Analyse bivariée :

Représentons les nuages de points par paires de variables quantitatives puis ensuite on étudiera les corrélations entre toutes les variables.

3.1 Les nuages de points :

```
column =["MLOGP","ON1V","F04[C-O]","Class"]
data_new = data[column]
sns.pairplot(data_new, hue="Class")
```

```
<seaborn.axisgrid.PairGrid at 0x27acb324160>
```



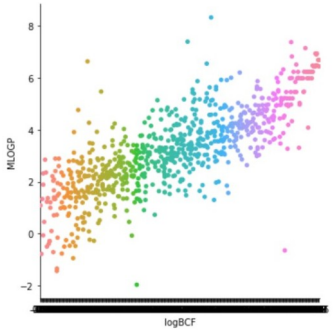
3.2 la matrice de corrélation :

```
data.corr(method="spearman").style.format("{:.2}").background_gradient(cmap=plt.get_cmap("coolwarm"))
```

	nHM	piPC09	PCD	X2Av	MLOGP	ON1V	N-072	B02[C-N]	F04[C-O]	Class	logBCF
nHM	1.0	0.078	0.01	0.43	0.31	-0.26	-0.042	-0.22	-0.071	-0.015	0.37
piPC09	0.078	1.0	0.81	-0.36	0.53	0.53	0.087	0.02	0.2	0.22	0.45
PCD	0.01	0.81	1.0	-0.59	0.46	0.21	-0.024	0.12	0.17	0.12	0.39
X2Av	0.43	-0.36	-0.59	1.0	-0.039	-0.031	-0.062	-0.32	-0.21	-0.0064	0.034
MLOGP	0.31	0.53	0.46	-0.039	1.0	0.27	-0.21	-0.35	-0.15	0.21	0.8
ON1V	-0.26	0.53	0.21	-0.031	0.27	1.0	0.28	0.17	0.45	0.3	0.2
N-072	-0.042	0.087	-0.024	-0.062	-0.21	0.28	1.0	0.4	0.26	-0.00053	-0.19
B02[C-N]	-0.22	0.02	0.12	-0.32	-0.35	0.17	0.4	1.0	0.27	-0.11	-0.32
F04[C-O]	-0.071	0.2	0.17	-0.21	-0.15	0.45	0.26	0.27	1.0	0.22	-0.13
Class	-0.015	0.22	0.12	-0.0064	0.21	0.3	-0.00053	-0.11	0.22	1.0	-0.051
logBCF	0.37	0.45	0.39	0.034	0.8	0.2	-0.19	-0.32	-0.13	-0.051	1.0

On utilise la matrice de corrélation pour déterminer si deux variables sont dépendantes (il existe une relation entre ces deux variables ou ces deux variables sont corrélées). Comme le montre la matrice ci dessus on constate qu'il ya une une forte corrélation entre :

- **Class et ON1V (0 .3)**, **Class et F04[C-O]** , **Class et piPC09**
- logBCF et MLOGP (0.8) : pour cela il existe un lien linéaire
  : plus **logBCF** augmente, plus **MLOGP** augmente aussi et vice versa.



4. Détection des valeurs manquantes : « aucune »

```
missing_values = ["n/a", "na", "--"]
data = pd.read_csv("Grisoni_et_al_2016_EnvInt88.csv", na_values = missing_values)
print(data.isnull().sum().sum())
```

```
0
```