Prédiction des décès dus au Covid19 à l'aide de machine learning.

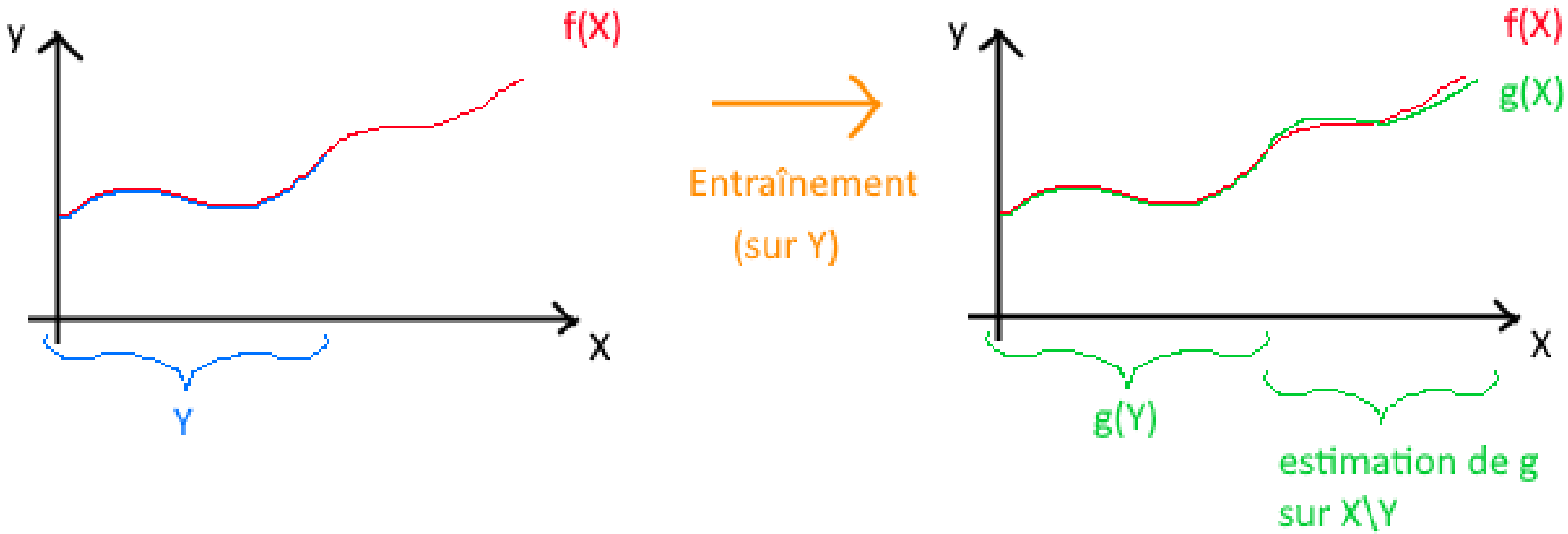
Enzo De Carvalho

Numéro d'inscription : 29448

2020-2021

Sommaire

Principe de la démarche



$\hookrightarrow$ le modèle $\hat{g}$ généralise les données connues Y fournies.

Première approche : regression linéaire

En utilisant ElasticNet

Modèle :

$$\hat{g}_{deces}(\omega, t) = \omega_0 + \omega_1 t$$

t les données (le temps)

$\omega = \begin{pmatrix} \omega_0 \\ \omega_1 \end{pmatrix}$ un paramètre à déterminer

Première approche : regression linéaire

En utilisant ElasticNet

Modèle :

$$\hat{g}_{deces}(\omega, t) = \omega_0 + \omega_1 t$$

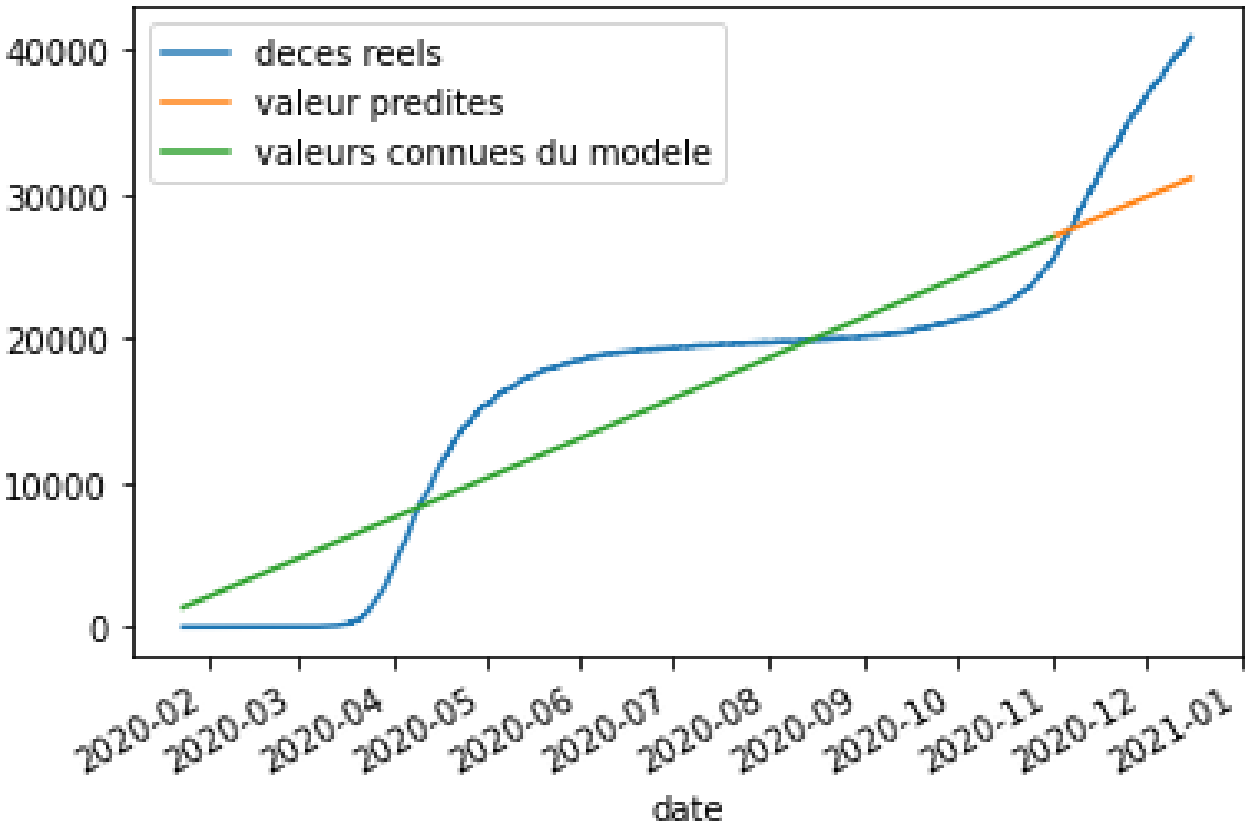
t les données (le temps)

$\omega = \begin{pmatrix} \omega_0 \\ \omega_1 \end{pmatrix}$ un paramètre à déterminer

le modèle ElasticNet détermine alors ω.
Le résultat dépend des hyperparamètres : α et ρ

Première application

Prédictions entre le 2020/11/01 et 2020/12/16.



$\rho = 0.9$

$\alpha = 0.1$

Figure – Résultats peu satisfaisant... (ici `l1_ratio` est ρ)

SVR ; Premier résultat

Modèle SVR

Hyperparamètres :

- C le paramètre de régularisation
- ϵ la taille du tube de « non-pénalité »
- γ paramètre du noyeau (`rbf` ici)

SVR ; Premier résultat

Modèle SVR

Hyperparamètres :

- C le paramètre de régularisation
- ϵ la taille du tube de « non-pénalité »
- γ paramètre du noyeau (`rbf` ici)

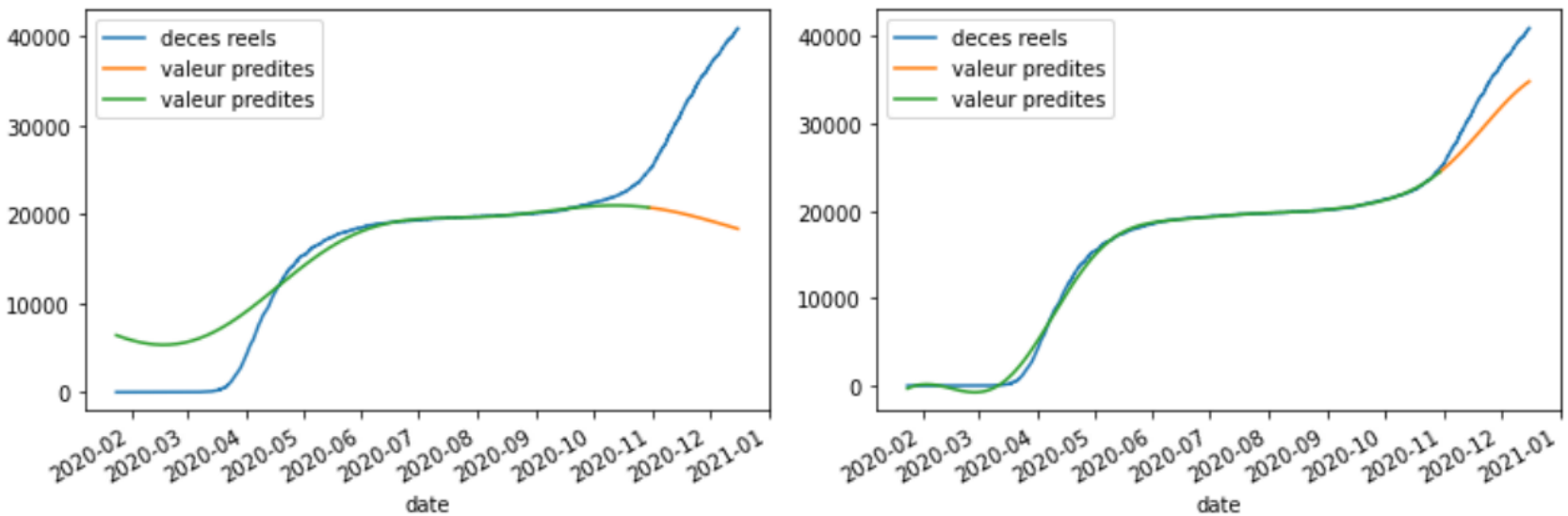
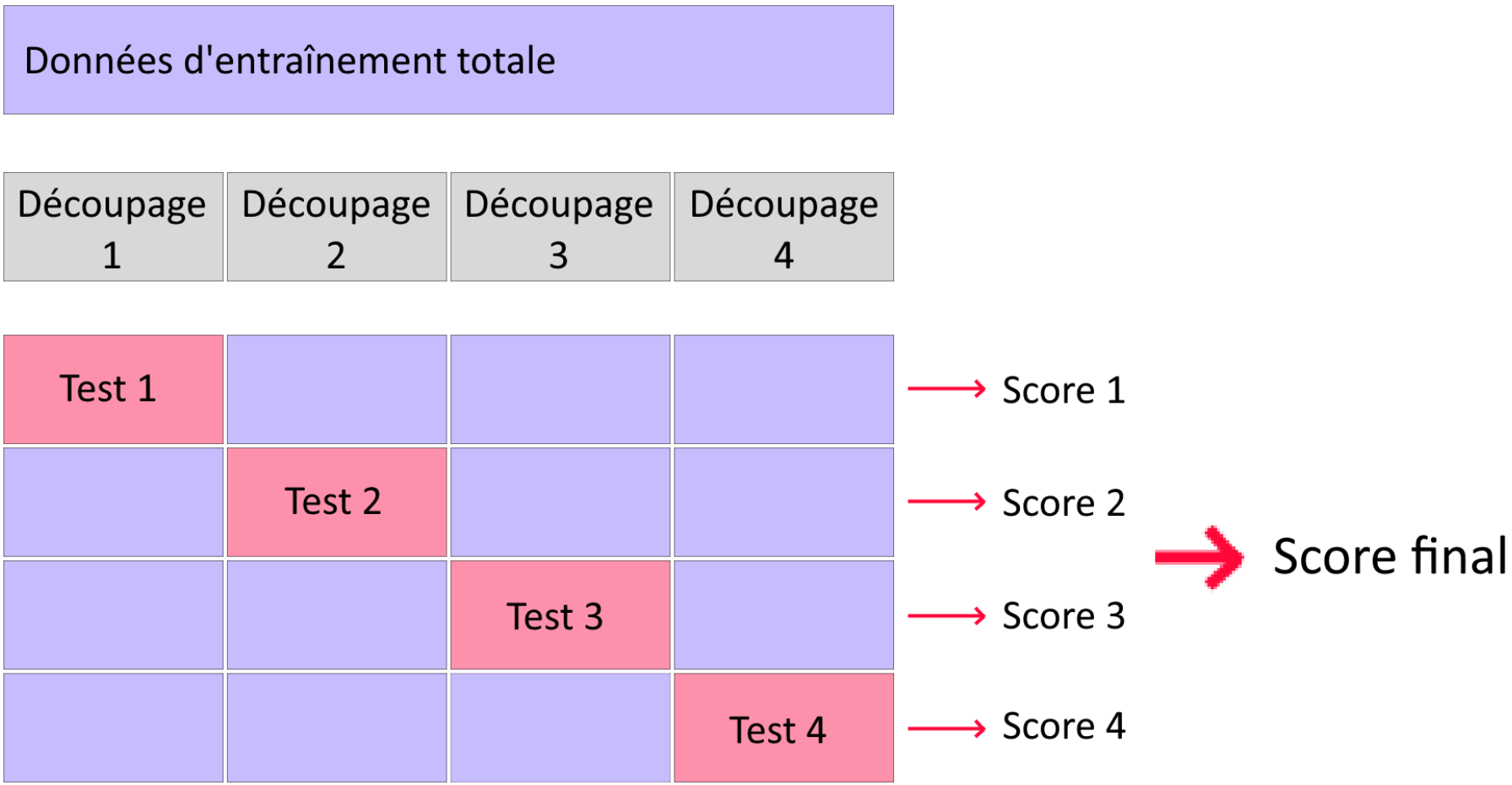


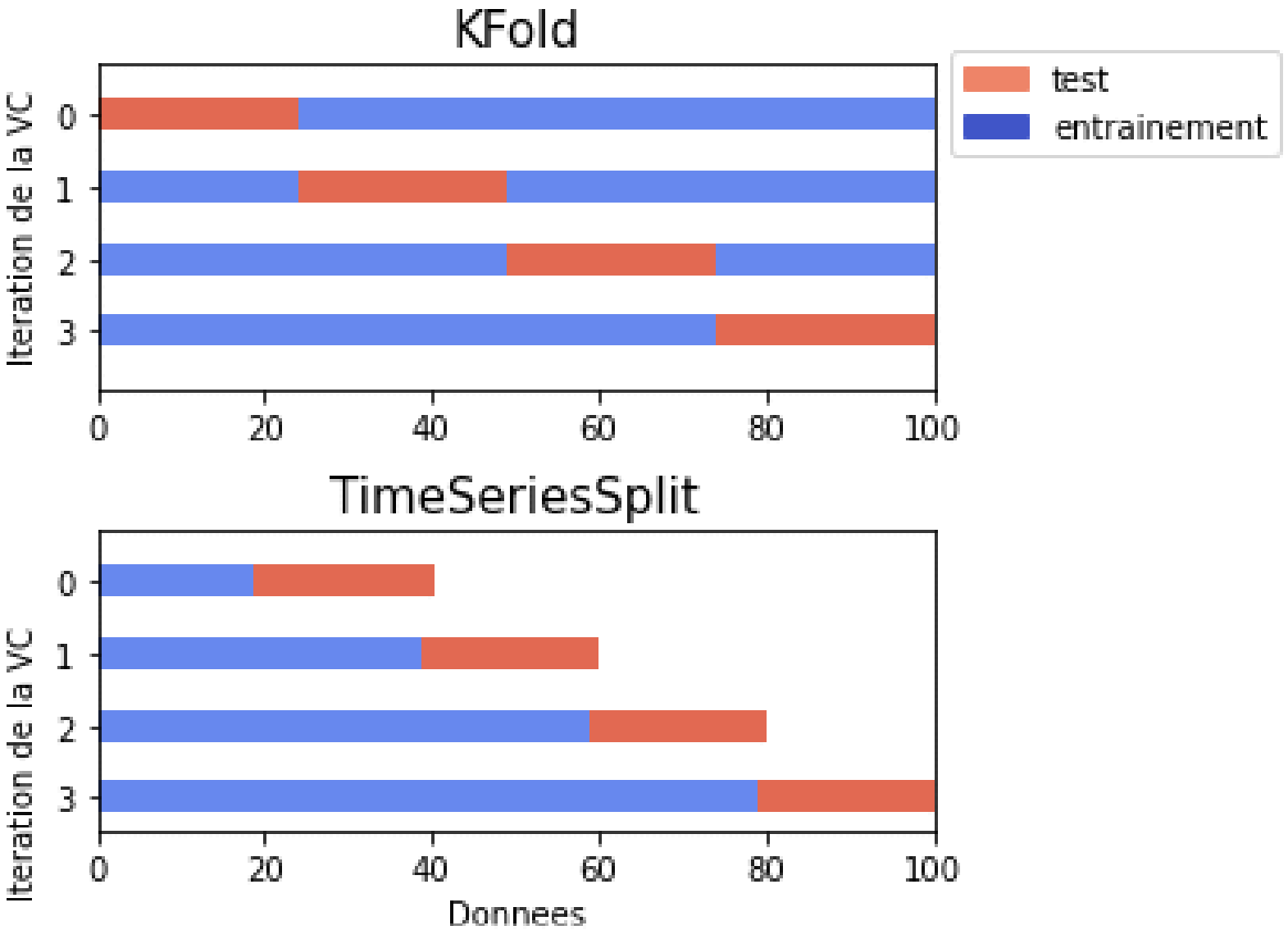
Figure – SVR avec $C = 100$, puis $C = 100000$

Validation croisée

Pour une combinaison d'hyperparamètre :

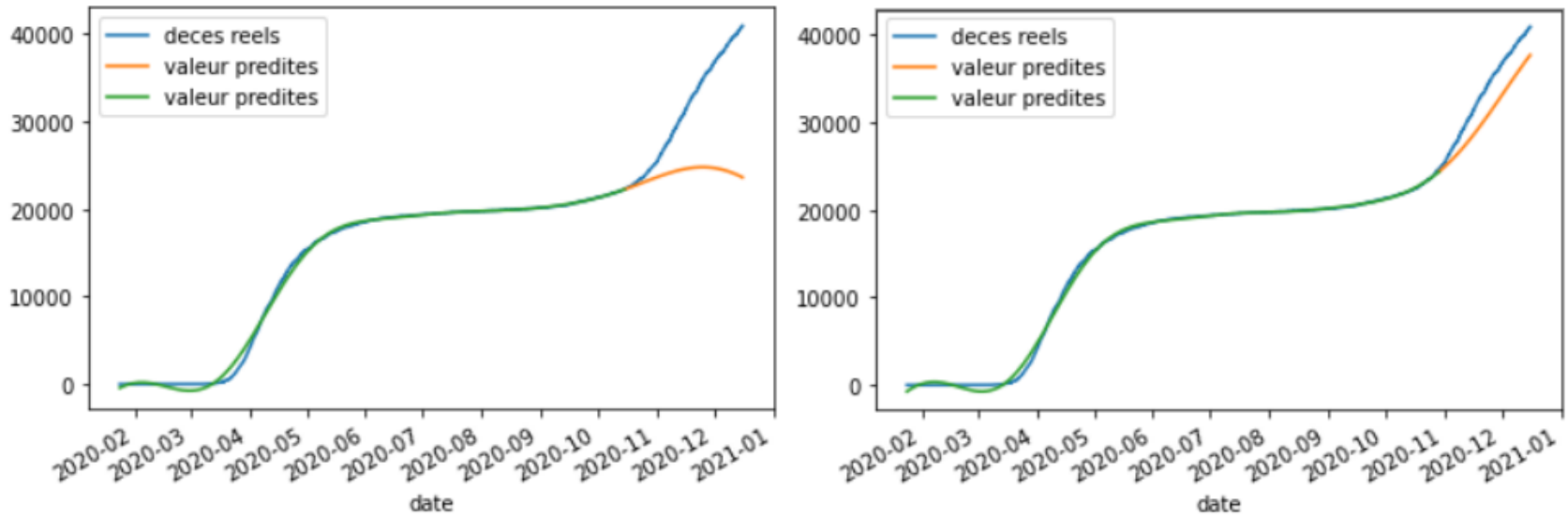


Stratégie pour la Validation croisée



Application avec SVR

```
params = {'svr__C' : [10**i for i in range(1,8)]}
```



{'svr__C' : 100000}

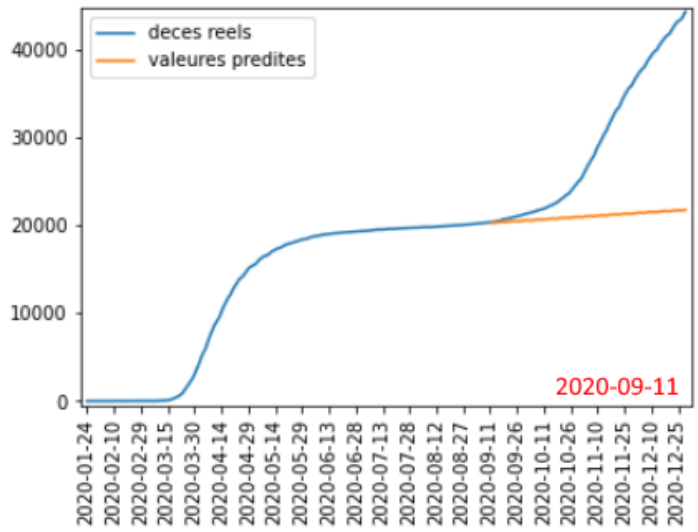
{'svr__C' : 1000000}

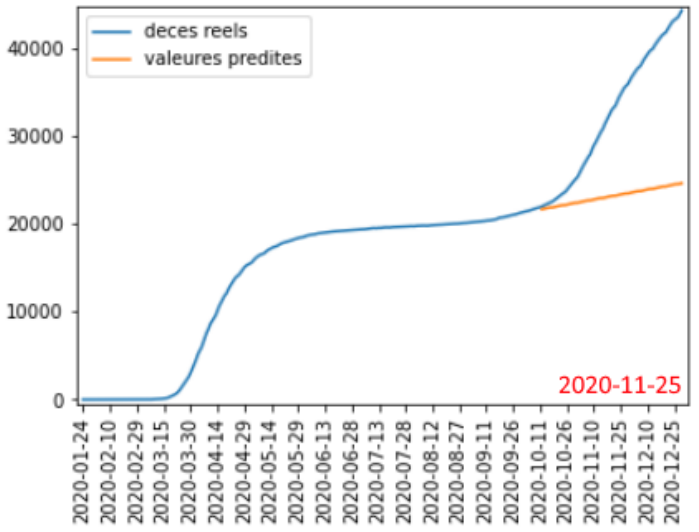
Figure – à partir du 2020/10/15, puis du 2020/11/01

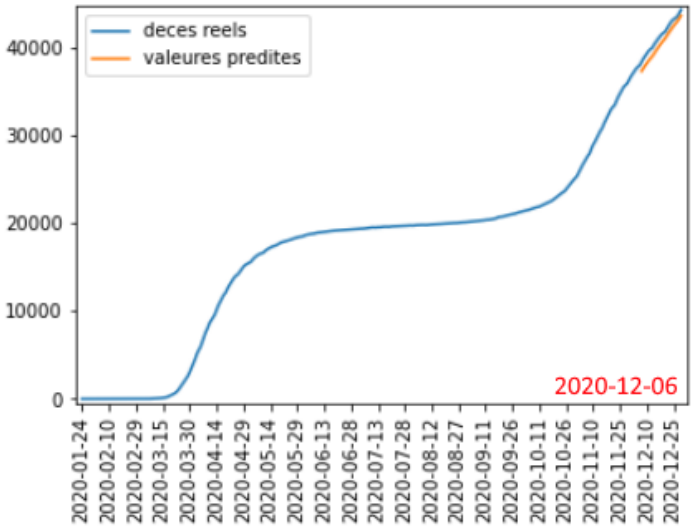
⇒ Échec de généralisation.

Prophet

Approche avec le modèle Prophet de Facebook







RegressorChain SVR

Approche multivarié avec RegressorChain

```
params = {
'svr__C' : [10**i for i in range(1,8)],
'svr__epsilon': [0.1,0.01,0.001]}
```

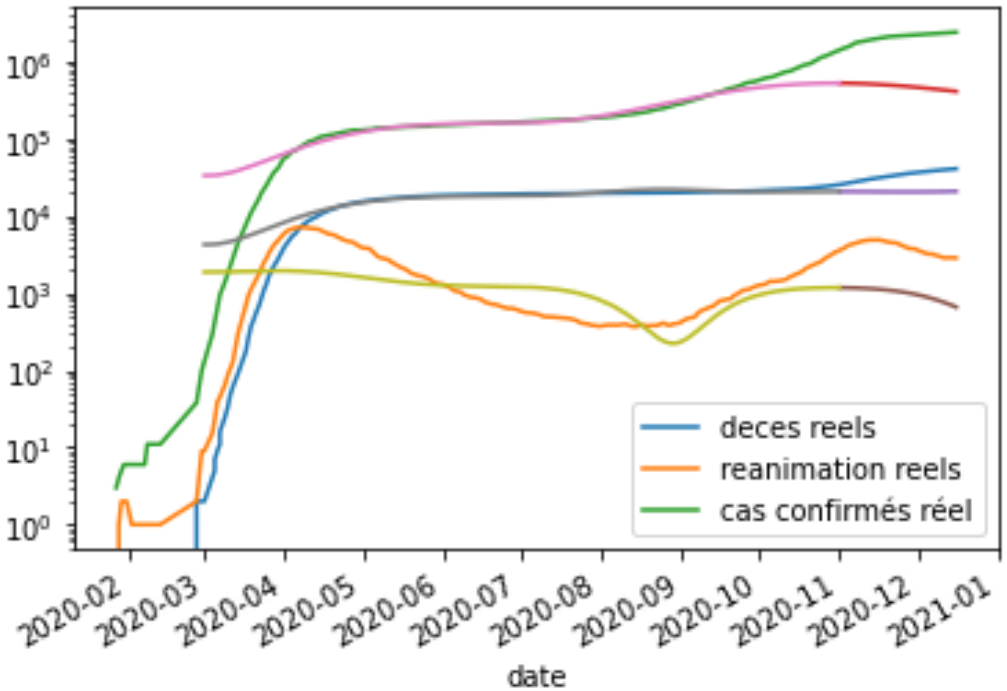
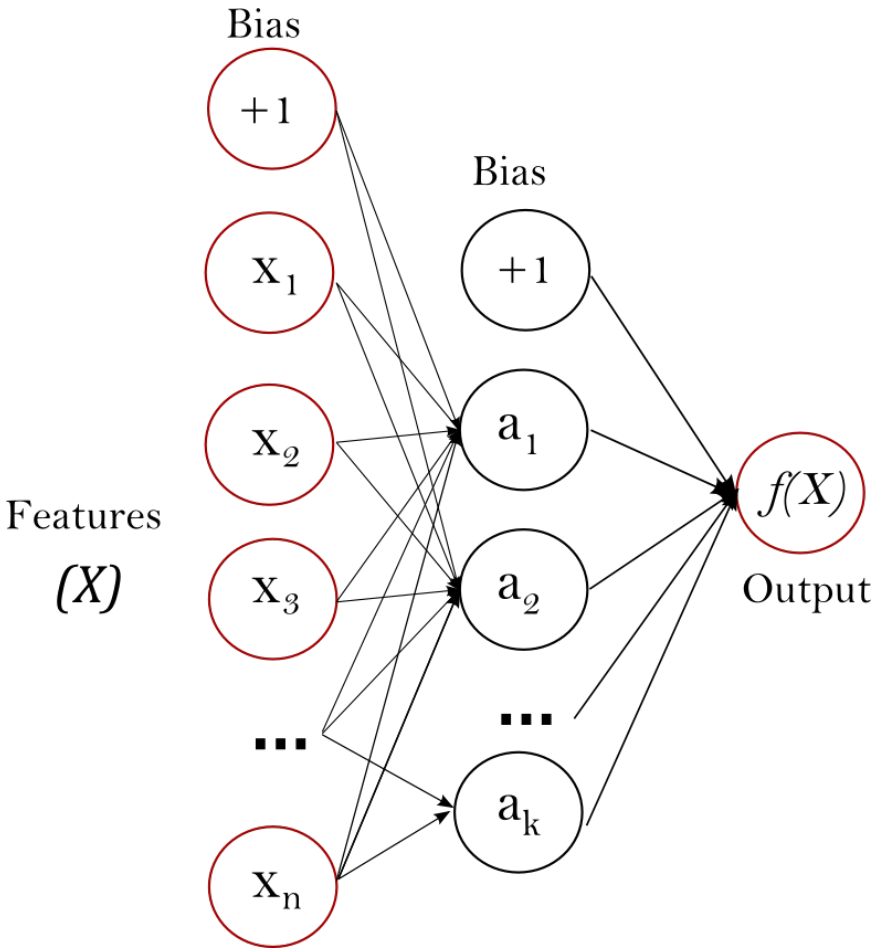


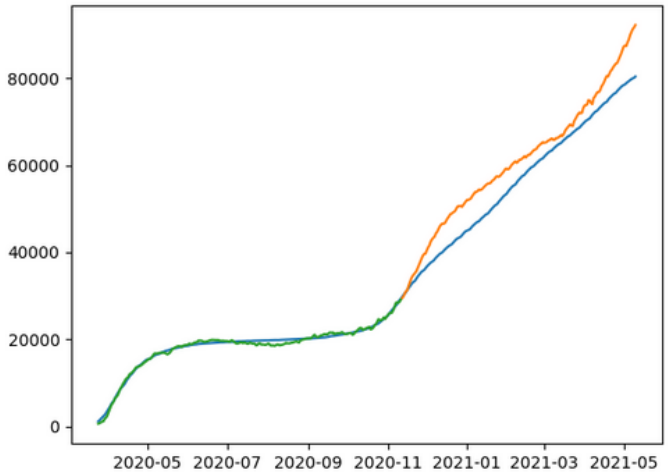
Figure – RegressorChain avec $C = 10000$ et $\epsilon = 0.001$

Réseau neuronal

Approche avec un réseau de neurone



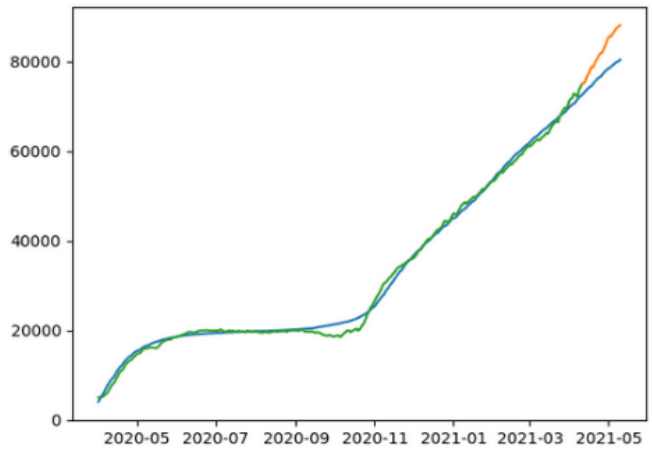
Application avec réseau neuronal



max_iter
100k

date
11/11/20

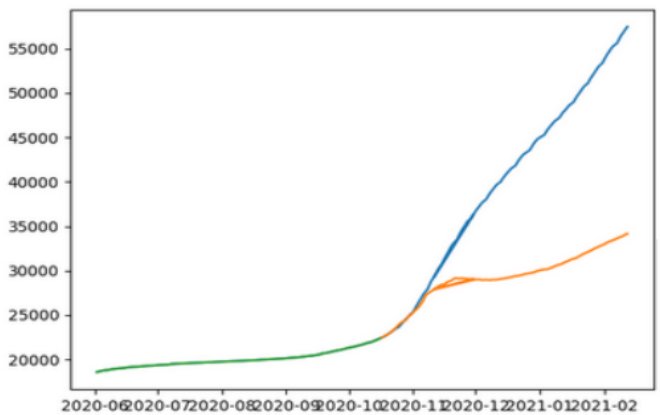
décalage
7j



max_iter
90k

date
10/04/21

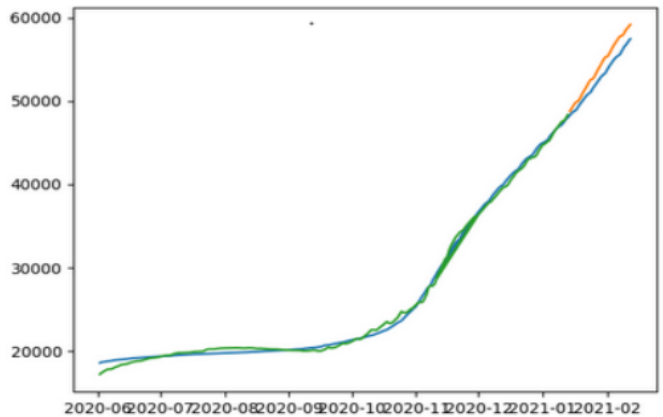
décalage
15j



max_iter
90k

date
15/10/20

décalage
7j



max_iter
90k

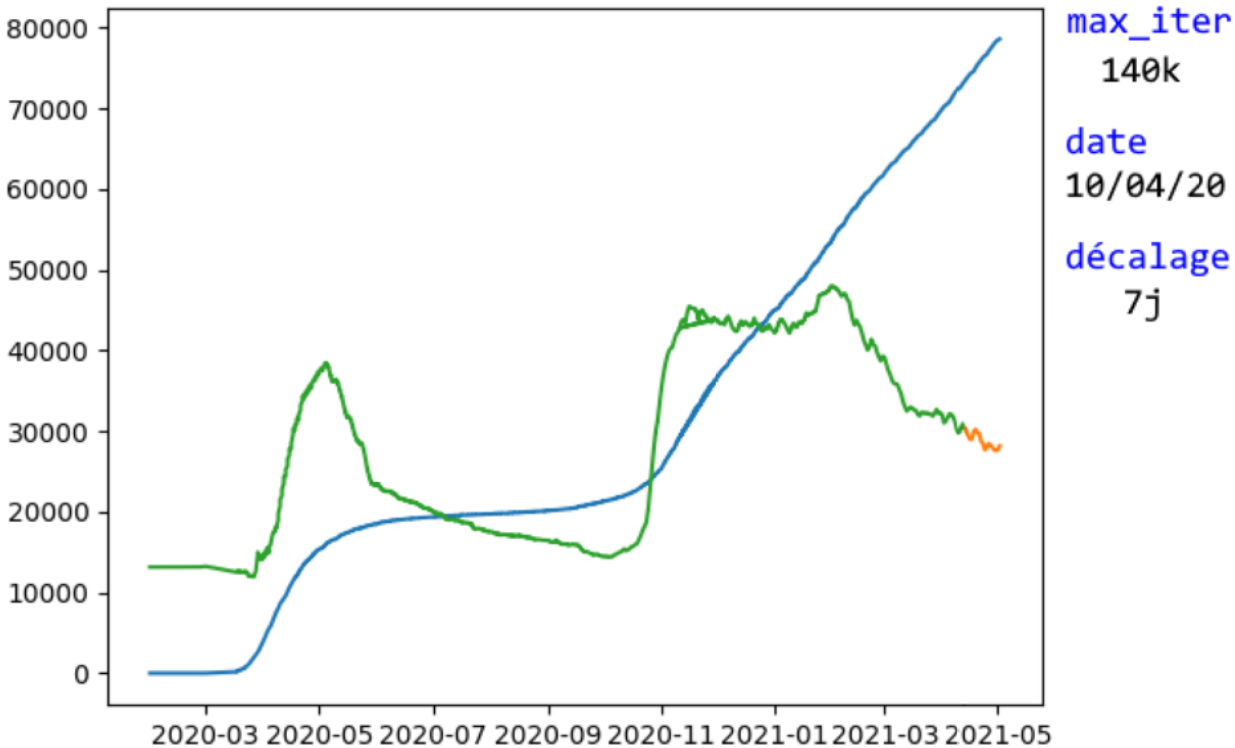
date
10/01/21

décalage
7j

Application avec réseau neuronal

taux de corrélation entre

`total_cas_confirmes` et `total_deces_hopital` : `0.977939`



$\Rightarrow$ échec du modèle sans la courbe des cas confirmés

Conclusion

$\Rightarrow$ Résultats peu convaincants.

fonction d'objectif d'ElasticNet

ElasticNet cherche ω tel que :

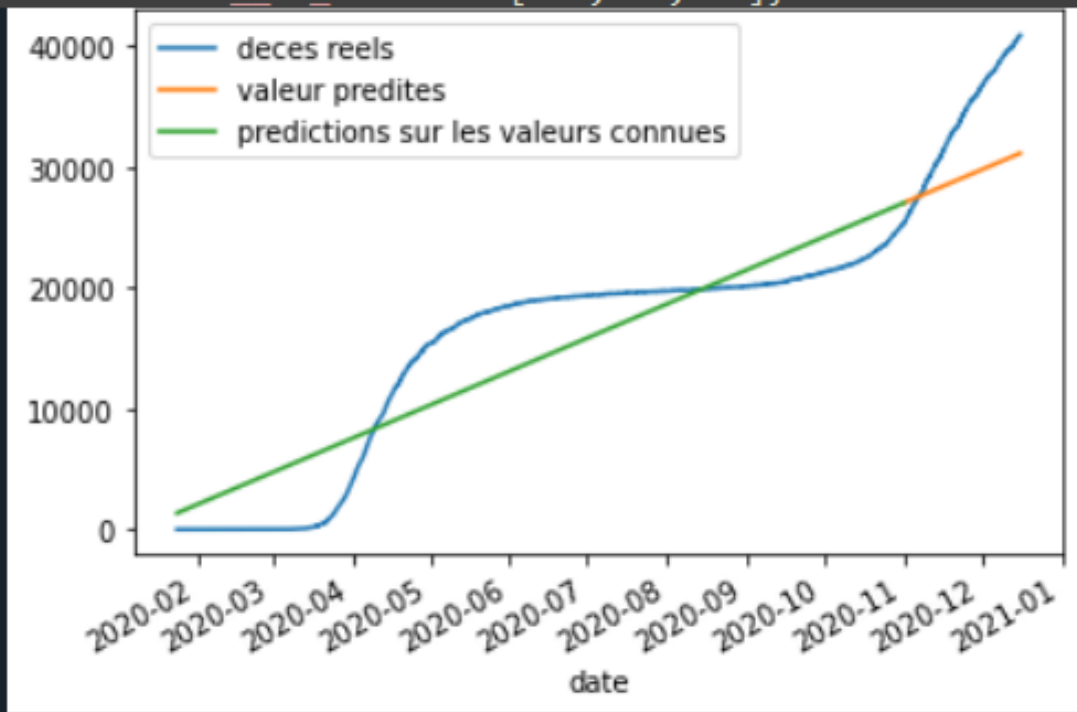
$$\min_{\omega} \frac{1}{2n_{deces}} ||\hat{g}_{deces}(\omega, t) - f(t)||_2^2 + \alpha\rho|\omega| + \frac{\alpha(1-\rho)}{2}||\omega||_2^2$$

α et ρ les hyperparamètres définissant le modèle,

f la courbe réelle des décès.

Détails sur la regression linéaire

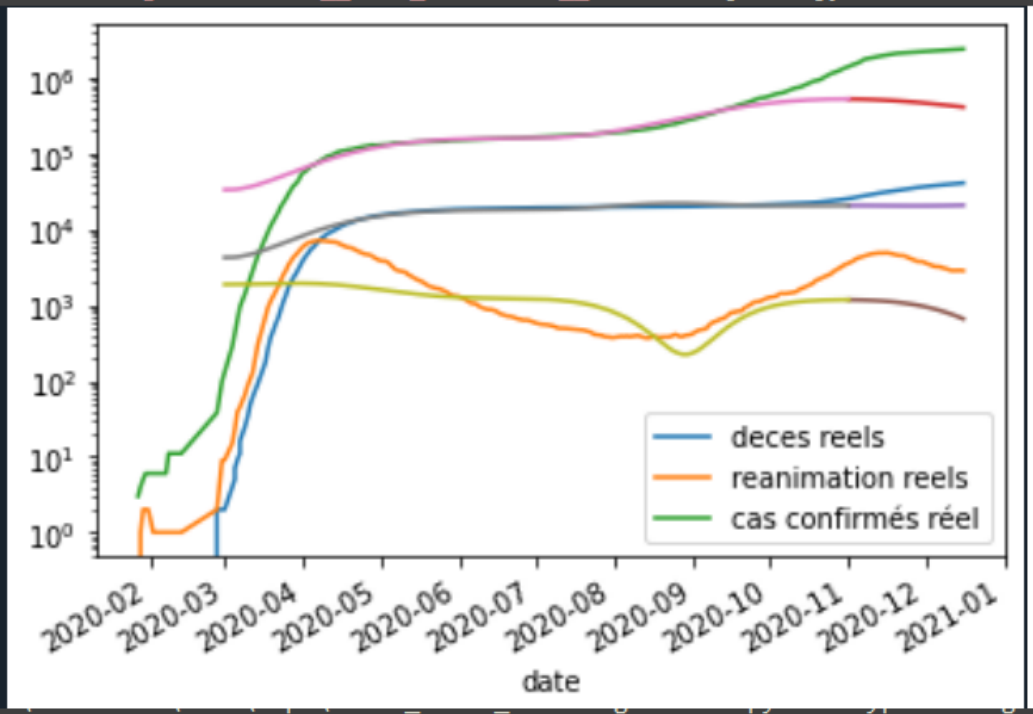
```
params = {#'lasso__alpha' : [i/100 for i in range(10,25)]
    #'svr__degree': [0],
    #'svr__C': [600000]
    'elasticnet__alpha' : [i/100 for i in range(10,100)],
    'elasticnet__l1_ratio' : [0.5,0.8,0.9]}
```




```
deces    622
dtype: int64
{'elasticnet__alpha': 0.1, 'elasticnet__l1_ratio': 0.9}
```

Détails sur la regression multivarié

```
model = make_pipeline(StandardScaler(), RegressorChain(SVR(), order=[0,2,1]))
params = {'regressorchain__base_estimator__C': [10**i for i in range(2,8)],
          'regressorchain__base_estimator__epsilon': [0.001,0.1,0.01],
          'regressorchain__base_estimator__kernel': ['rbf']}
```




```
types.Specify dtype option on import or set low_memory=False.
  ### Multi regresseur avec le modele SVR ###
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
Fitting 15 folds for each of 18 candidates, totalling 270 fits
[Parallel(n_jobs=1)]: Done 270 out of 270 | elapsed:  2.2min finished
{'regressorchain__base_estimator__C': 10000, 'regressorchain__base_estimator__epsilon': 0.001,
'regressorchain__base_estimator__kernel': 'rbf'}
```