

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

De Carvalho, Enzo

numéro d'inscription : 29448

2020-2021

Sommaire

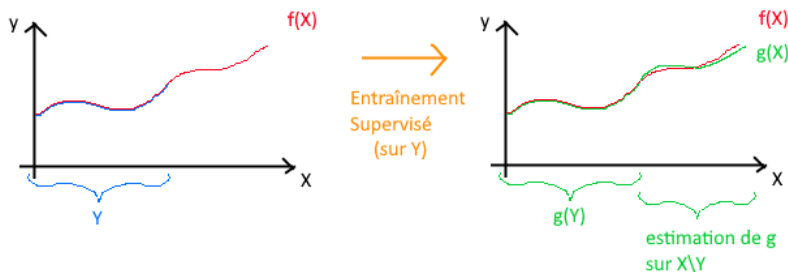
1 Première approche : simple regression

- Principe
- Regression Linéaire
- Optimisation d'hyperparamètres
- Résultats avec SVR
 - Cross-validation

2 Approche multivariées

- Multiregresseur : 'RegressorChain'
- Réseau neuronal

Principe de la demarche



Entraînement supervisé :

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

Première approche : Regression linéaire

En utilisant le modèle [ElasticNet](#) du module `scikit-learn`.

Modèle :

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

t les données (le temps)

ω un paramètre à déterminer

Première approche : Regression linéaire

En utilisant le modèle [ElasticNet](#) du module `scikit-learn`.

Modèle :

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

t les données (le temps)
 ω un paramètre à déterminer

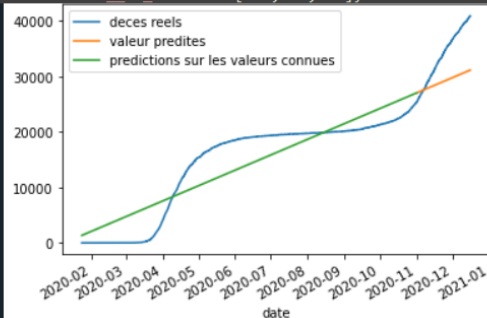
Pour trouver ω le modèle [ElasticNet](#) résout :

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

α et ρ les hyperparamètres définissant le modèle,
 f la courbe réelle des décès.

Première application

```
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}
```

Figure: Résultats peu satisfaisant... (ici $l1_ratio$ est ρ)

SVR ; Premier résultat

Modèle **SVR** du module `scikit-learn`.

Hyperparamètres :

- C le paramètre de régularisation

- ϵ la taille du tube de "non-pénalité"

- γ paramètre du noyau (rbf ici)

SVR ; Premier résultat

Modèle **SVR** du module `scikit-learn`.

Hyperparamètres :

C le paramètre de régularisation

ϵ la taille du tube de "non-pénalité"

γ paramètre du noyau (rbf ici)

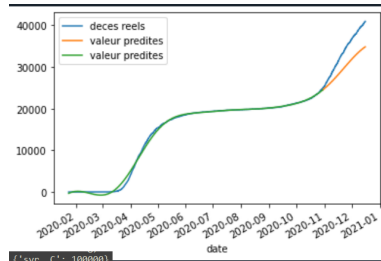
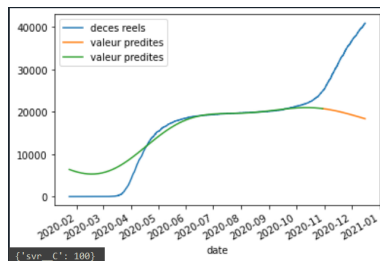
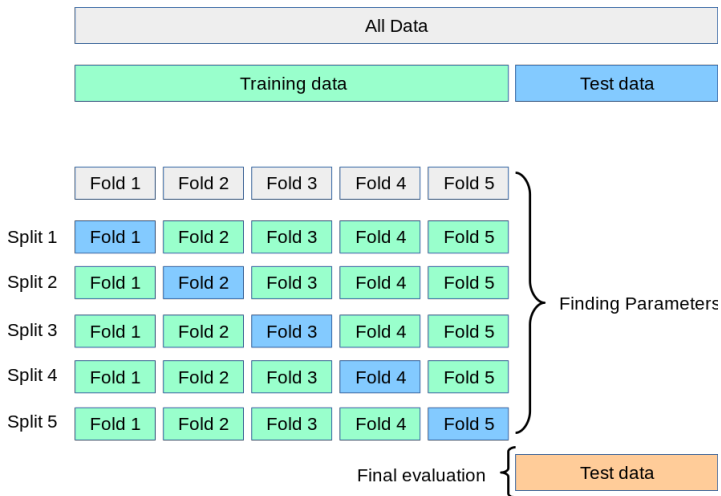
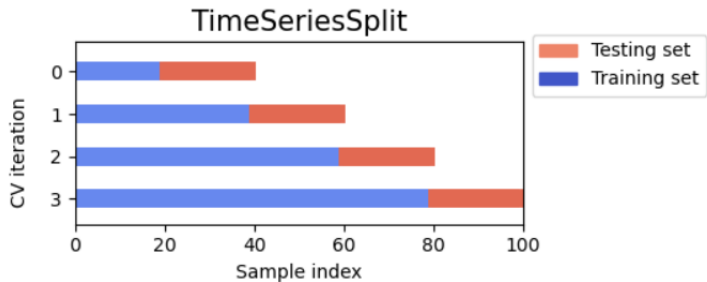
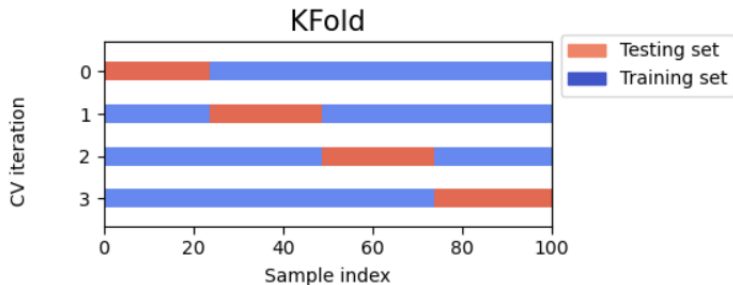


Figure: **SVR** avec $C = 100$ puis $C = 100000$

Cross-Validation

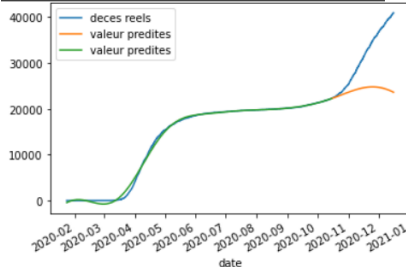


Stratégie pour la CV

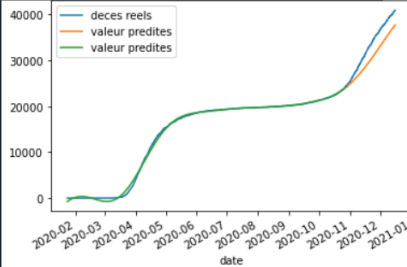


Application avec SVR

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



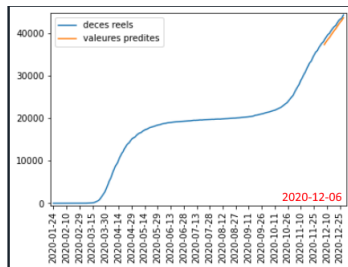
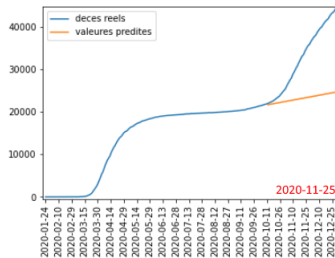
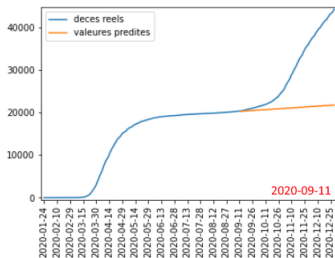
```
params = {'svr__C': [10**i for i in range(1,8)]}
```



⇒ Échec de généralisation.

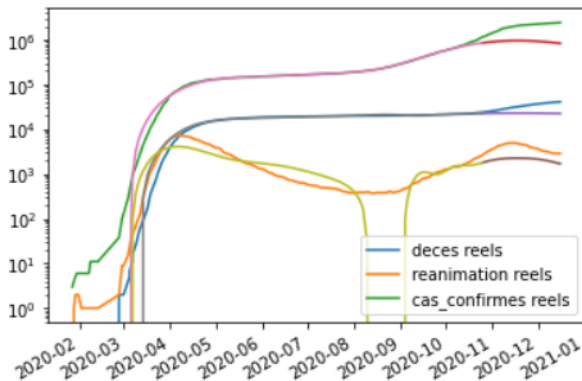
Prophet

Approche avec le module Prophet de Facebook



RegressorChain SVR

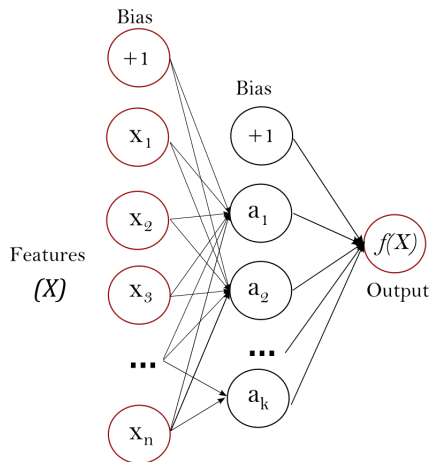
Approche multivariés avec RegressorChain



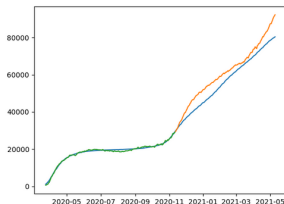
```
{'regressorchain_base_estimator_C': 100000, 'regressorchain_base_estimator_epsilon': 0.001}
```

Réseau neuronal

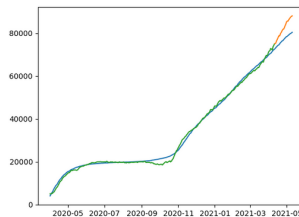
Approche avec un **réseau de neurone**



Application avec réseau neuronal

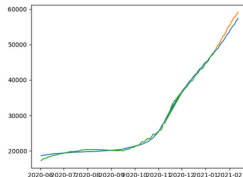
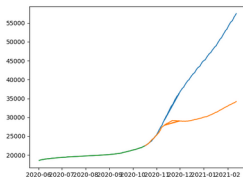


Neural network avec 7 jour de décalage; max_iter=100k; prédit à partir du 11



`{'mlpregressor__max_iter': 90000, 'mlpregressor__n_iter_no_change': 3, 'mlpregressor__tol': 0.0001}`

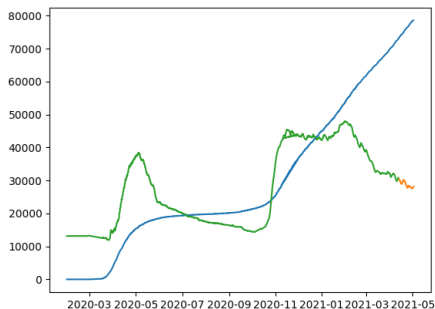
NN: décès prédits 30 jours avec les données de 15 jours avant



Neural network avec 7 jour de décalage; max_iter=90k

Application avec réseau neuronal

	total_cas_confirmes	total_deces_hopital
total_cas_confirmes	1.000000	0.977939
total_deces_hopital	0.977939	1.000000



⇒ échec du modèle sans la courbe des cas confirmés