

# Prédiction de décès du Covid19 grâce au machine learning.

Enzo De Carvalho

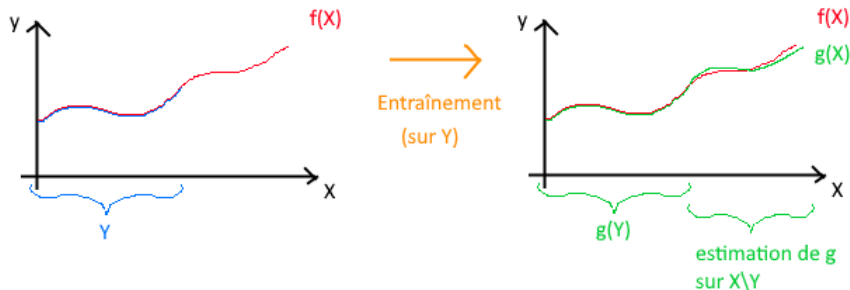
Numéro d'inscription : 29448  
2020-2021

# Sommaire

1 Première approche : simple regression

2 Approche multivariées

# Principe de la démarche



↪ 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)

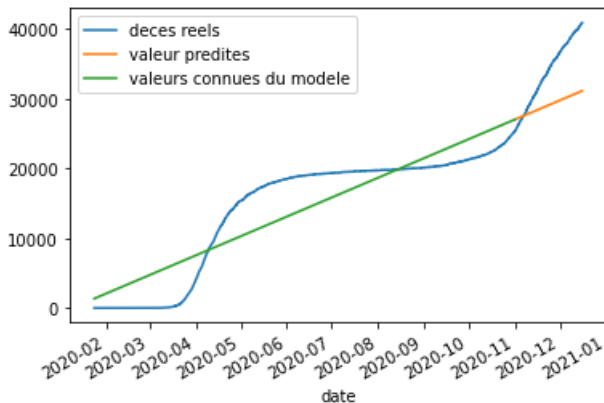
$$\omega = \begin{pmatrix} \omega_0 \\ \omega_1 \end{pmatrix} \text{ un paramètre à déterminer}$$

le modèle **ElasticNet** détermine alors  $\omega$ .

Le résultat dépend des hyperparamètres :  $\alpha$  et  $\rho$

# 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  $\rho$ )

# SVR ; Premier résultat

Modèle **SVR**

Hyperparamètres :

- $C$  le paramètre de régularisation

- $\epsilon$  la taille du tube de « non-pénalité »

- $\gamma$  paramètre du noyau (rbf ici)

## 6 / 15



# Validation croisée

Pour une combinaison d'hyperparamètre :

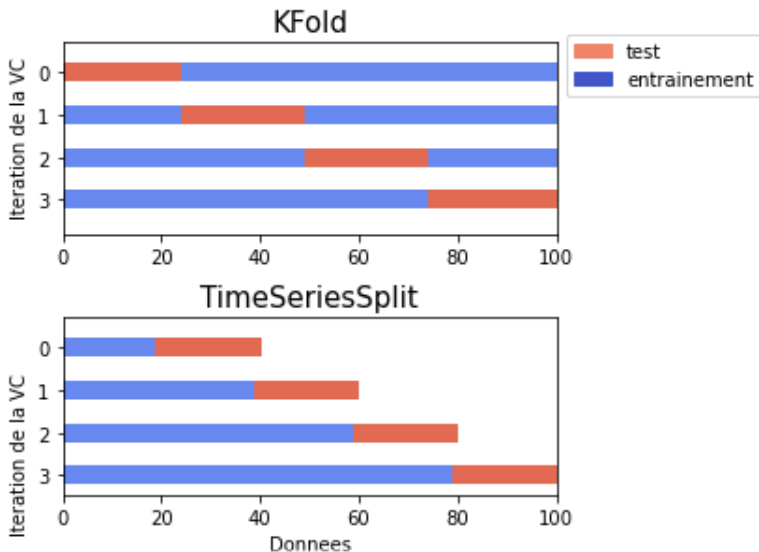
Données d'entraînement totale

| Découpage<br>1 | Découpage<br>2 | Découpage<br>3 | Découpage<br>4 |
|----------------|----------------|----------------|----------------|
|----------------|----------------|----------------|----------------|

|        |        |        |        |           |
|--------|--------|--------|--------|-----------|
| Test 1 |        |        |        | → Score 1 |
|        | Test 2 |        |        | → Score 2 |
|        |        | Test 3 |        | → Score 3 |
|        |        |        | Test 4 | → Score 4 |

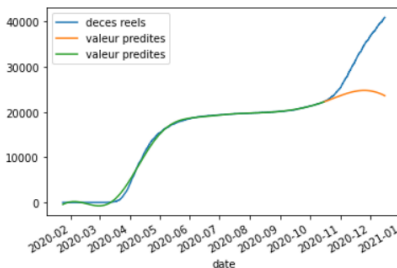
→ Score final

# Stratégie pour la Validation croisée

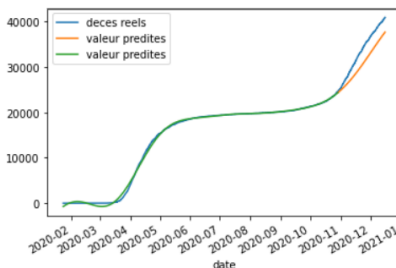


# Application avec SVR

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



{'svr\_\_C' : 100000}



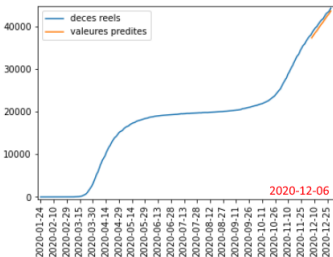
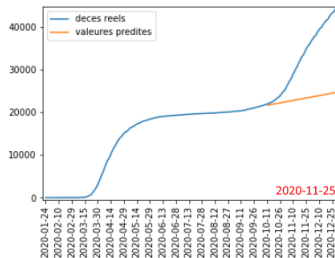
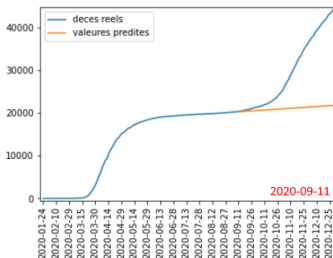
{'svr\_\_C' : 1000000}

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]}
```

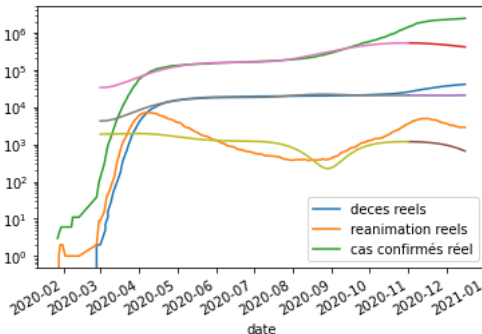
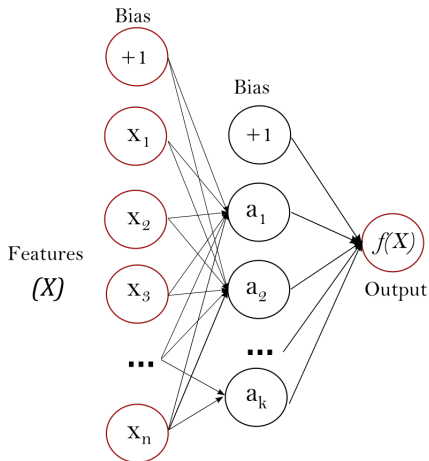


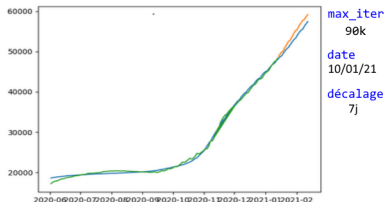
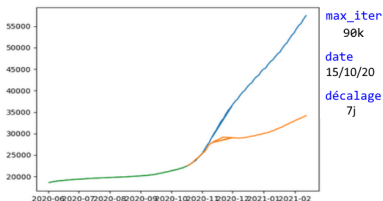
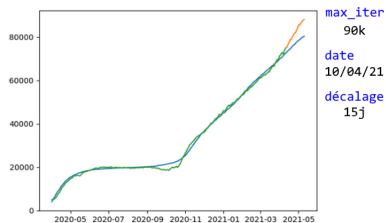
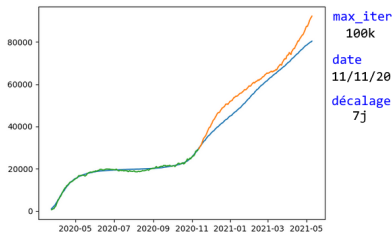
Figure – RegressorChain avec  $C = 10000$  et  $\epsilon = 0.001$

# Réseau neuronal

Approche avec un **réseau de neurone**



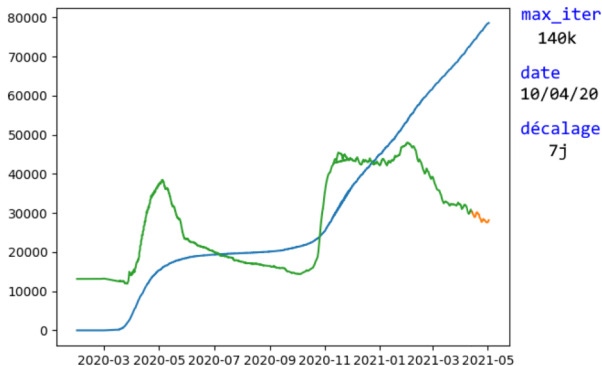
# Application avec réseau neuronal



# Application avec réseau neuronal

taux de corrélation entre

total\_cas\_confirmes et total\_deces\_hopital : 0.977939



⇒ échec du modèle sans la courbe des cas confirmés



# Conclusion

⇒ Difficultés de généralisation des modèles proposés, trop dépendant du point d'inflexion.

# fonction d'objectif d'ElasticNet

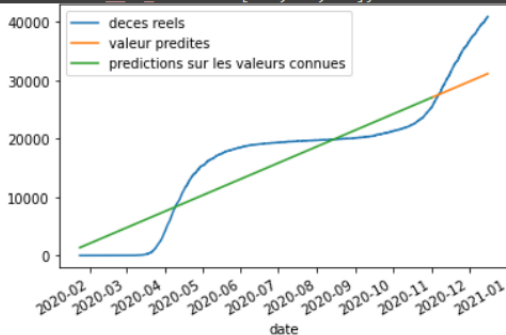
ElasticNet cherche  $\omega$  tel que :

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

$\alpha$  et  $\rho$  les hyperparamètres définissant le modèle,  
 $f$  la courbe réelle des décès.

# Détails sur la regression lineaire

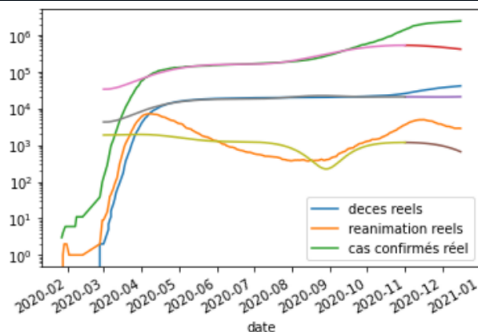
```
params = {'lasso_alpha' : [i/100 for i in range(10,25)]  
          #'svr_degree': [0],  
          #'svr_C': [600000]  
          'elasticnet_alpha' : [i/100 for i in range(10,100)],  
          'elasticnet_l1_ratio' : [0.5,0.8,0.9]}
```



```
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'}
```