Super titre de TIPE

sous-titre

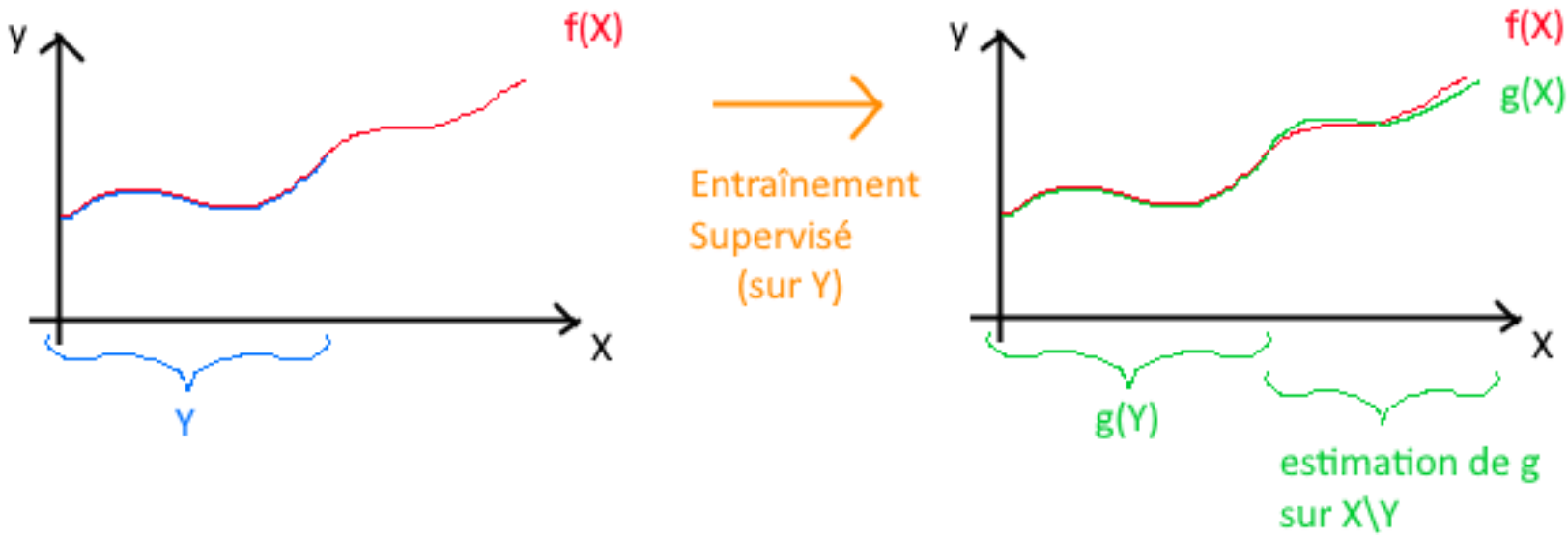
Alexandrine, Pedro
De Carvalho, Enzo

2020-2021

Sommaire

Principe de l'apprentissage supervisée

Donné une fonction $f : X \rightarrow y$ (X et y des données corrélées), un algoritmhe d'apprentissage supervisée cherche alors à obtenir un modèle $g : X \rightarrow y$ qui estime f à partir d'une d'une partie $Y \subset f(X)$ déjà connue de l'algoritme.



Principe de l'apprentissage supervisée

La création du modèle passe par l'optimisation d'une fonction d'objectif, où les paramètres à optimiser sont ceux du modèles. Exemple avec une regression linéaire dont le modèle est de la forme :

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$$

Figure: x les données, w les paramètres

N.B : refaire la figure (celle-ci est volée du manuel scikit)

ElasticNet

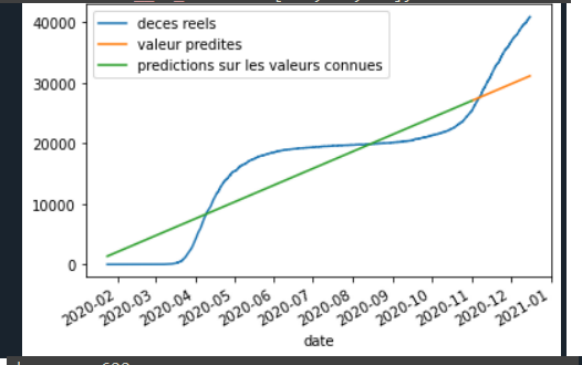
La fonction d'objectif d'ElasticNet minimise : (à l'aide d'un algorithme itératif "Coordinate Descent" (trad?))

$$\min_w \frac{1}{2n_{\text{samples}}} ||Xw - y||_2^2 + \alpha\rho||w||_1 + \frac{\alpha(1-\rho)}{2}||w||_2^2$$

ElasticNet

Figure: α et ρ sont des hyperparamètres définissant le modèle

```
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]}
```

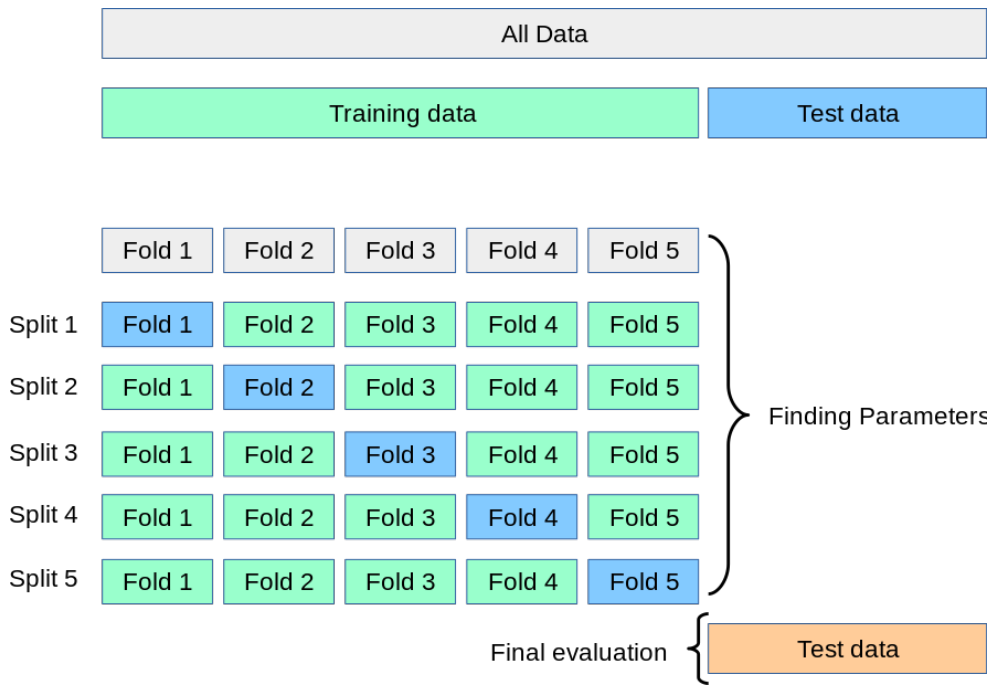



```
deces    622
dtype: int64
{'elasticnet__alpha': 0.1, 'elasticnet__l1_ratio': 0.9}
```

Figure: résultats peu intéressant dans notre cas..

Recherche du meilleur paramètre

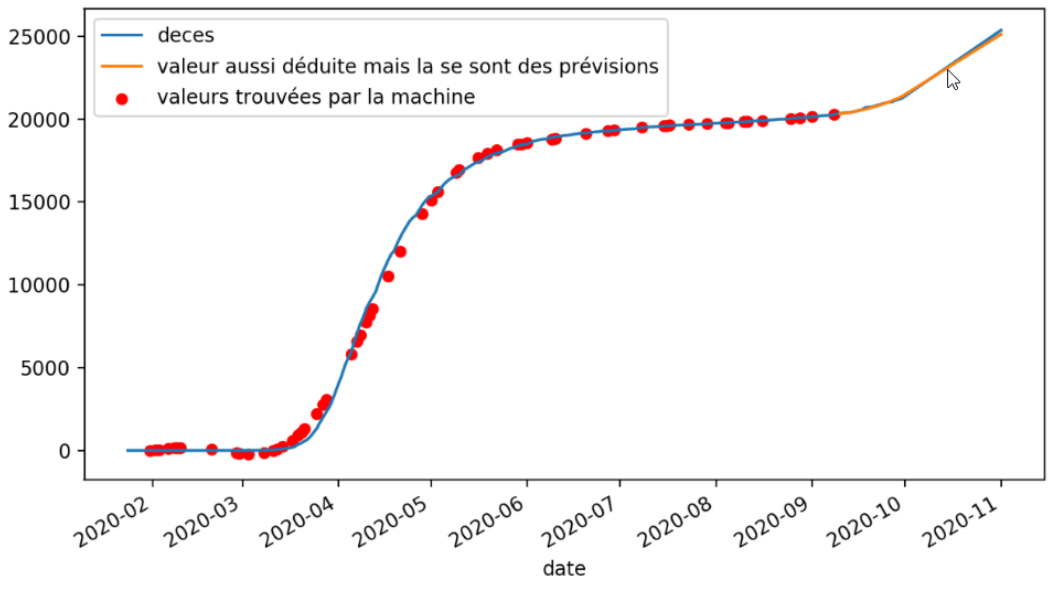
On recherche la meilleur combinaison des paramètres donnés en évaluant successivement chaque modèle pour chacune des combinaisons possibles selon une stratégie de "cross-validation" :



SVR ; premier résultat

Approche à l'aide du modèle SVR (avec le noyau 'rbf') (Support Vector Regressor)

— Modèle SVR (prédit les données entre le 02/12/20 et 16/12/20)




```
{'svr__C': 500000, 'svr__cache_size': 200, 'svr__degree': 0}
0.9980841978694834
```

Figure: Premier résultat avec SVR avec une "mauvaise stratégie" de découpage pour former un modèle prédisant une série temporelle

Découpage adapté pour la CV

Une note moyenne est donnée à un set de paramètres selon un score moyen donné sur chacun des découpages. Pour une évaluation cohérente, il faut un découpage cohérent :

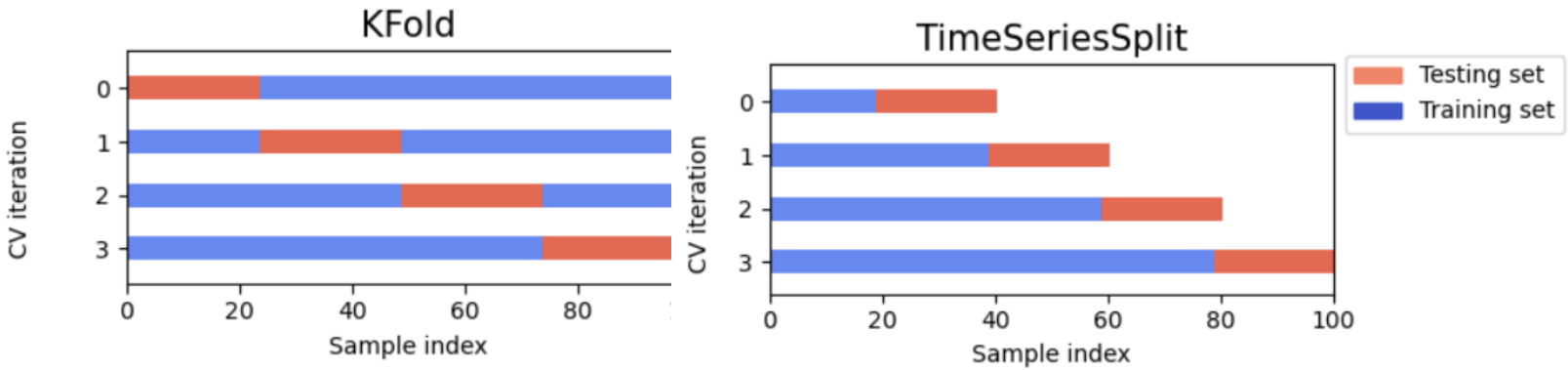
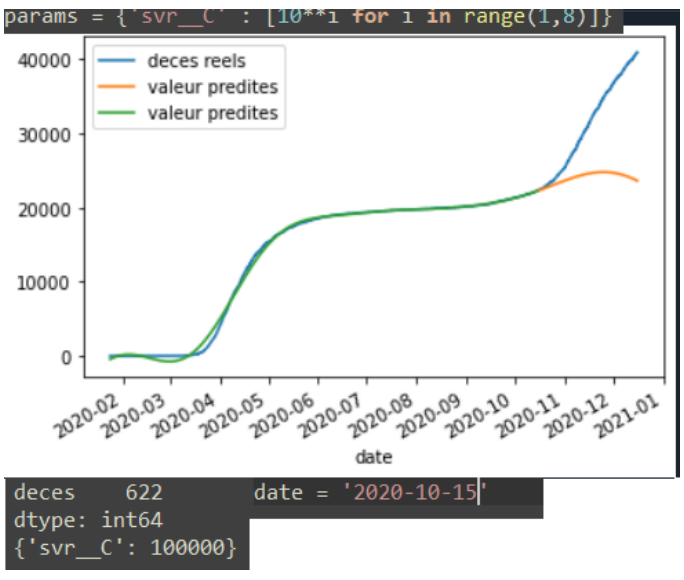


Figure: Comparaison des différents découpages des données employés pour la CV

SVR



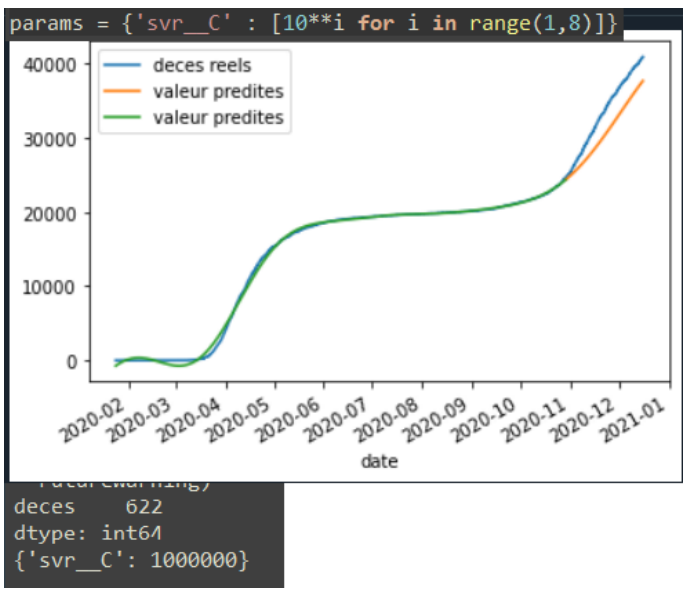


Figure: à gauche la prediction avant le pt d'inflexion, à droite, après.

⇒ Le modèle n'arrive pas à "suivre" sans le point d'inflexion.

fbprophet

Approche avec le module fbprophet (designé pour prédire des séries temporelles)

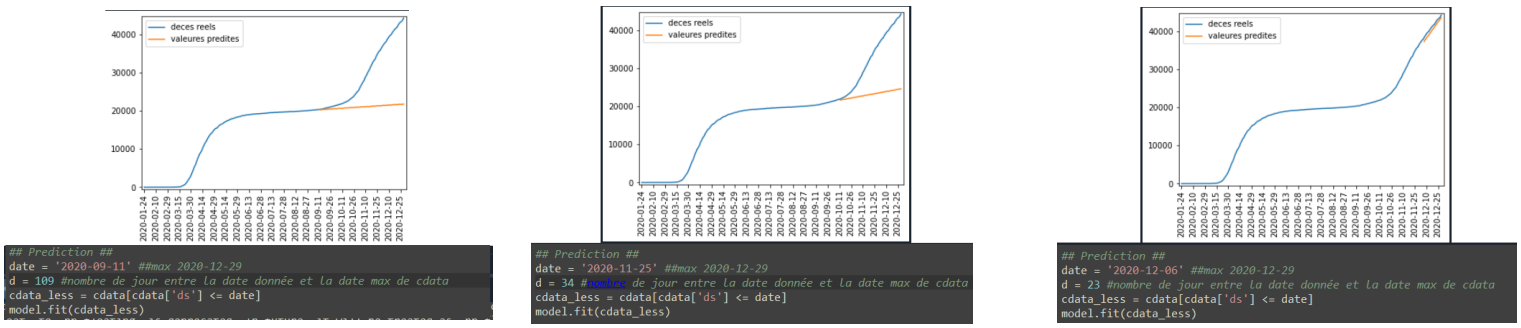


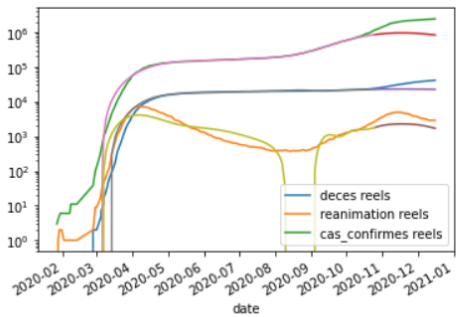
Figure: résultats avec fbprophet (insatisfaisant)

RegressorChain SVR

Approche avec le multiregresseur

(description de scikit : `https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.RegressorChain.html#sklearn.multioutput.RegressorChain`)

"Each model makes a prediction in the order specified by the chain using all of the available features provided to the model plus the predictions of models that are earlier in the chain."



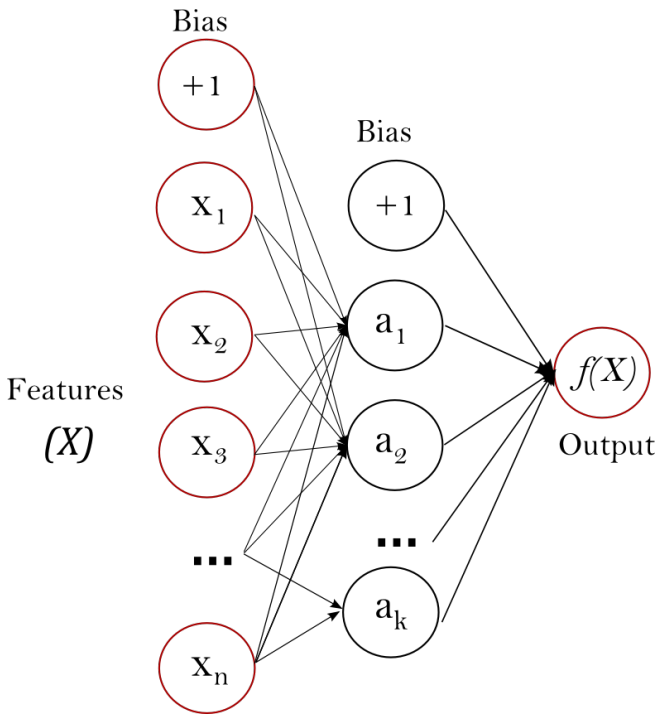
Multiregresseur

```
In [52]: runfile('C:/Users/Anzo/Documents/TIPE/repo/covid_deces_multiregresseur.py', wdir='C:/Users/Anzo/Documents/TIPE/repo')
C:\Users\Anzo\Documents\TIPE\repo\covid_deces_multiregresseur.py:1: DtypeWarning: Columns (17,18) have mixed types.Specify dtype option on import or set low_memory=False.
  ### Multi regresseur avec le modele SVR ###
{'regressorchain__base_estimator__C': 100000, 'regressorchain__base_estimator__epsilon': 0.001}

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.
```

Figure: bug qu'on a pas reussi à corriger sous ces conditions

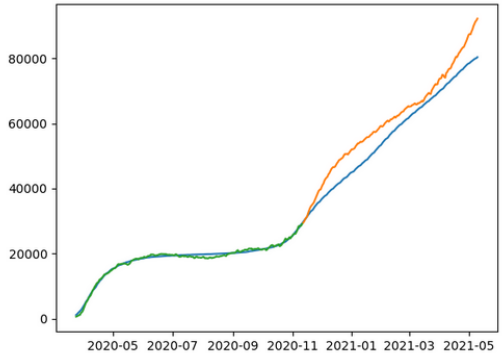
Explications



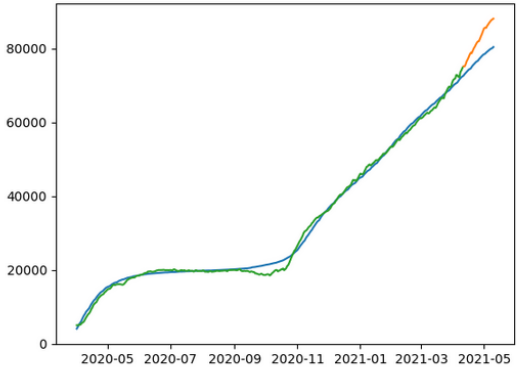
Basé sur le fonctionnement du cerveau. Donne un jeu de donné successivement interprétées par un jeu de hidden layers. Un hidden layer est un jeu de neurones chacun interprétant et transformant les objets donnés pour décrypter le jeu de donnés précédents. À la fin on aboutit au résultat trouvé par la machine. Le but de l'entrainement ici est l'ajustement de chaque hidden layer et de ces neurones pour tomber sur des prédictions les plus proches possible de la réalité. *Maxiter* est alors le nombre de fois que le modèle va s'entrainer au maximum si il n'a pas réussi à aboutir à un résultat satisfaisant la tolérance tol donnée.

N.B trop verbeux !

Résultats



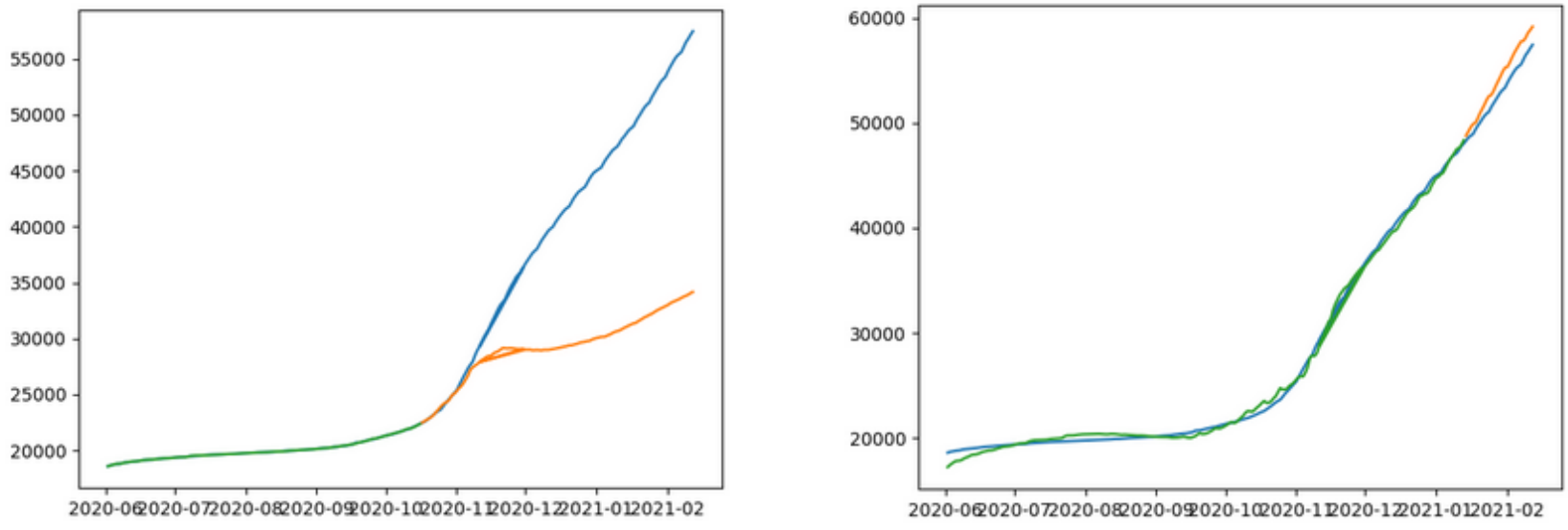
Neural network avec 7 jour de décalage; max_iter=100k; prédit à partir du 11 novembre




```
{'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

Résultats



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

Figure: échec du modèle sans la courbe des cas confirmés

Notes

N.B les diapos sont trop verbeux; il faut encore un effort de concision, ou alors elles doivent être expliquées à l'oral
N.B sourcer les images