# DEFI-IA 2022

## JASY

Jiawen / Adja / Salamata / Younes
5MA

# Sommaire

# Imputation des valeurs manquantes

- On considère la moyenne des mesures par jour pour les variables ff, t, td, hu et dd.
- Utilisation du module geopy pour trouver la distance entre deux stations en se servant de leurs longitudes et de leurs latitudes.
- Calcul d'une matrice de distance par paires de stations.
- Imputation des valeurs manquantes par interpolation

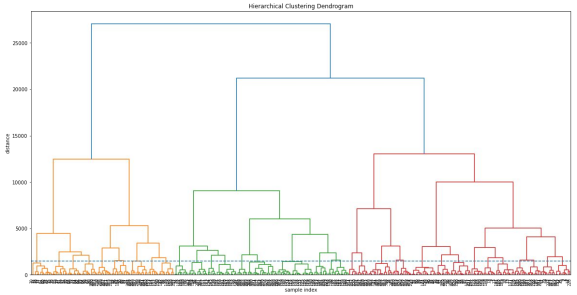
# Imputation des valeurs manquantes

- Sur l'échantillon d'apprentissage :

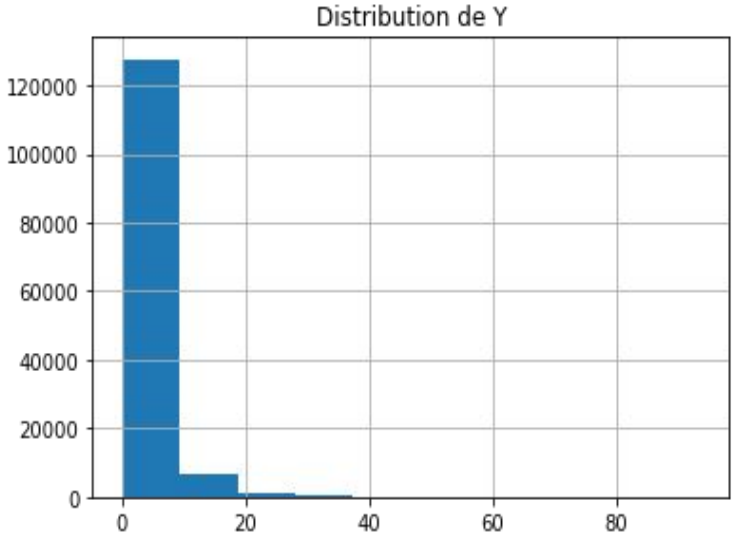
  Remplacement d'un NaN par la valeur de la station la plus proche (disponibilité de la valeur sur un rayon de cinq stations)

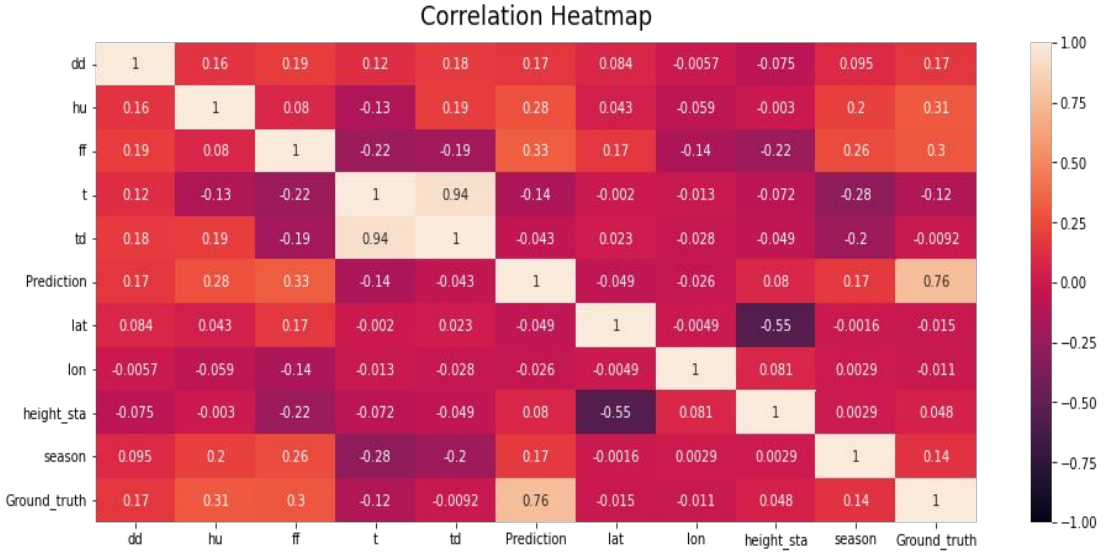
- Sur l'échantillon test :
  - ❏ Construction de clusters de stations
  - ❏ Calcul de la moyenne par jour pour chaque cluster
  - ❏ Remplacement d'un NaN par la moyenne de son cluster le même jour ou la date la plus récente si cette valeur n'est pas disponible.

| | | |
|---|---|---|
| number_sta | 0.00 | 0.00 |
| date | 0.00 | 0.00 |
| td | 32.13 | 1.28 |
| hu | 32.07 | 10.52 |
| ff | 39.48 | 10.54 |
| t | 4.92 | 18.12 |
| dd | 39.50 | 18.12 |



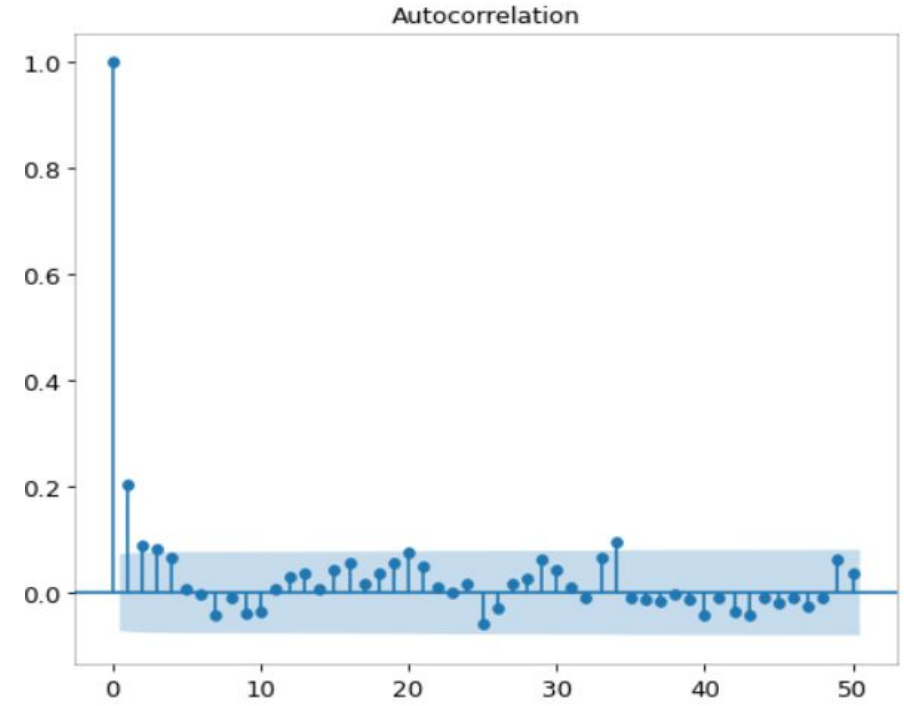
# Analyse descriptive

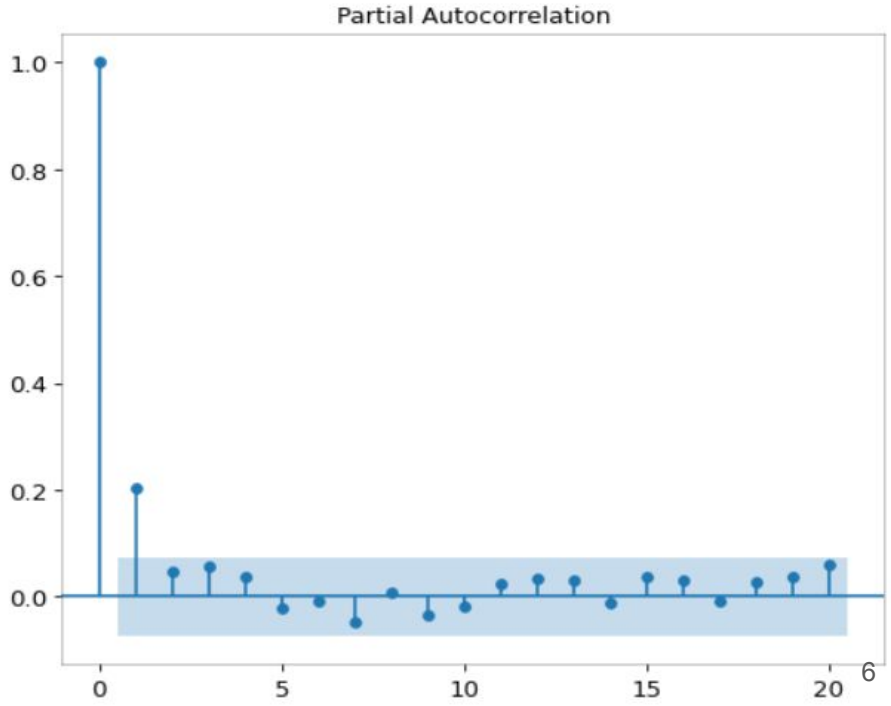


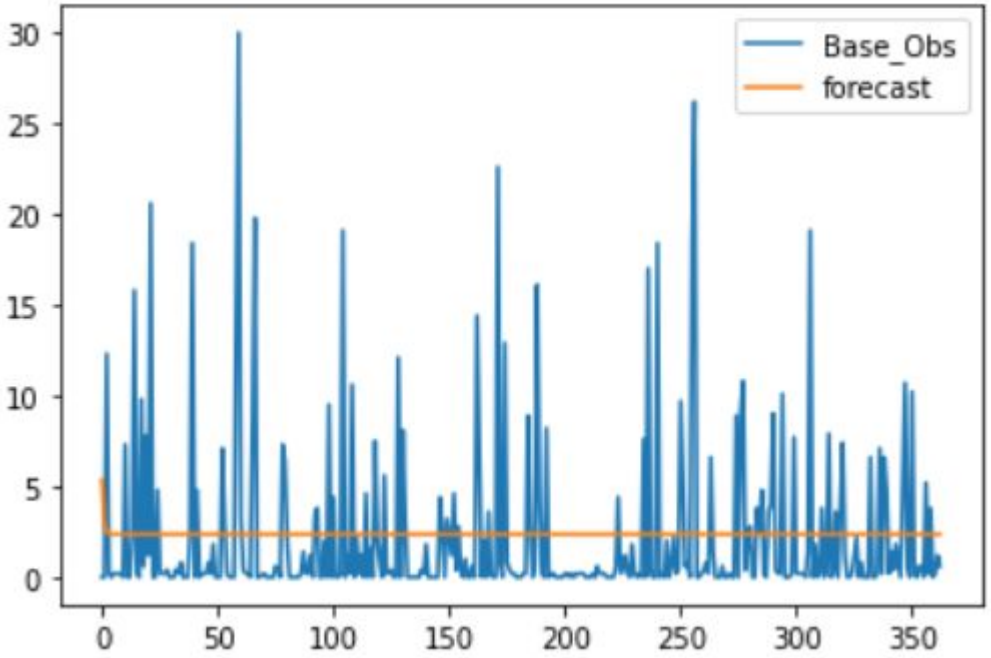
# Arima:

Test Dickey-Fuller: p-valeur = 0.0005



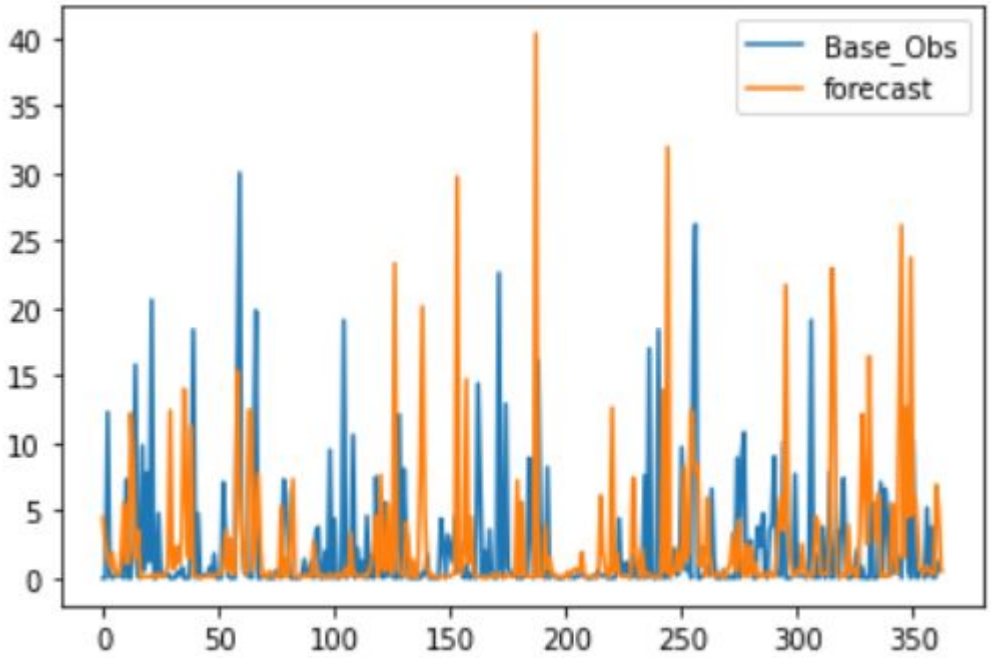



# Modèle de régression-Séries temporelles




MAPE = 154.04

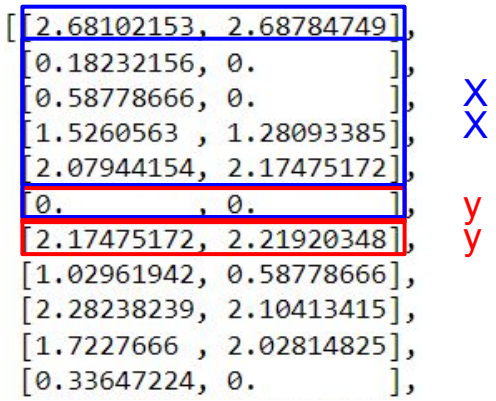
# Modèle de régression-Séries temporelles




MAPE = 187.80

# Méthodologie autour des méthodes de Deep learning

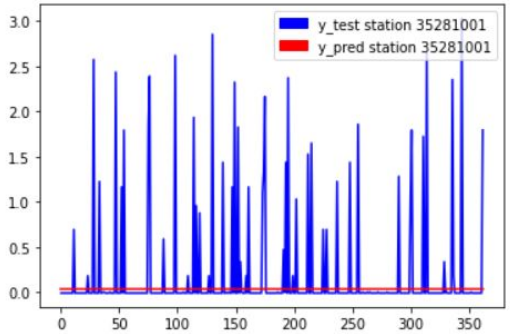
1. Construction d'une matrice 1093x325 (pluviométrie par jour et par station)
2. Transformation logarithmique
3. Séparation en des matrices X (de taille 60x325) et un vecteur y (de taille 325)
4. Constitution des échantillons d'apprentissage et de test
5. Implémentation du réseau de neurone
6. Récupération des Id à prédire
7. Seuillage des valeurs négatives
8. Calcul du MAPE

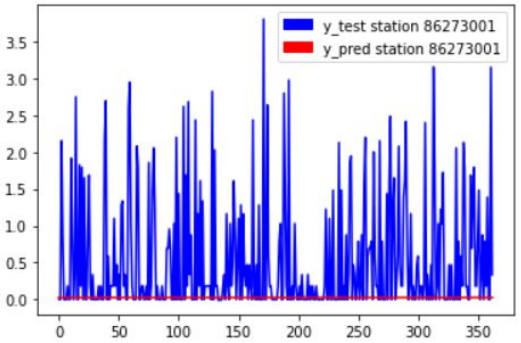


# RNN-LSTM bidirectionnels

```
Mean Absolute Percentage Error (MAPE): 32.47 %
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 masking (Masking)           (None, 60, 325)           0

 bidirectional (Bidirectiona  (None, 400)              841600
 l)

 repeat_vector (RepeatVector  (None, 1, 400)           0
 )

 bidirectional_1 (Bidirectio  (None, 1, 200)           400800
 nal)

 dropout (Dropout)           (None, 1, 200)            0

 time_distributed (TimeDistr  (None, 1, 325)           65325
 ibuted)

=================================================================
Total params: 1,307,725
Trainable params: 1,307,725
Non-trainable params: 0
_________________________________________________________________
```

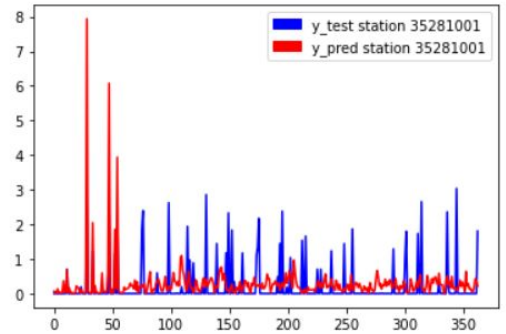


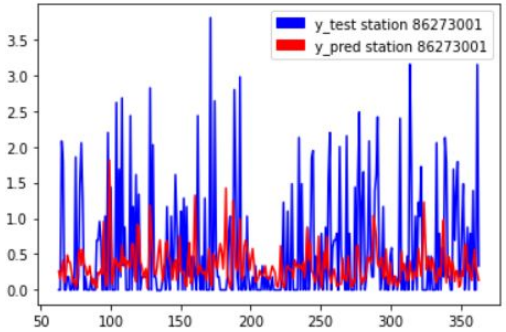



# CNN

Mean Absolute Percentage Error (MAPE): 38.94 %

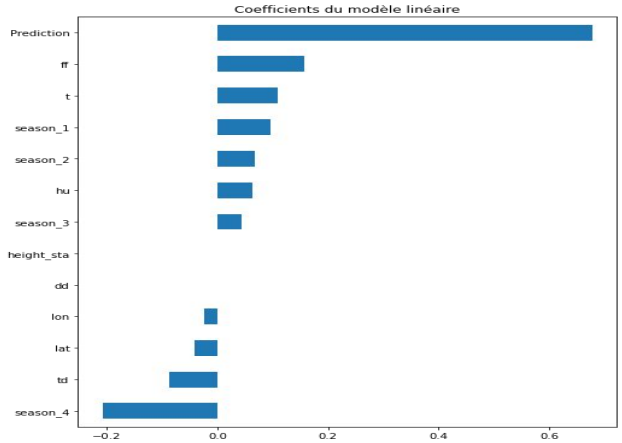
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 masking_1 (Masking)         (None, 60, 325)           0

 conv1d_2 (Conv1D)           (None, 59, 64)            41664

 conv1d_3 (Conv1D)           (None, 58, 32)            4128

 max_pooling1d_1 (MaxPooling  (None, 29, 32)           0
 1D)

 flatten_1 (Flatten)         (None, 928)               0

 dense_2 (Dense)             (None, 600)               557400

 dropout_1 (Dropout)         (None, 600)               0

 dense_3 (Dense)             (None, 325)               195325

=================================================================
Total params: 798,517
Trainable params: 798,517
Non-trainable params: 0
_________________________________________________________________
```



# Modèles linéaires et non linéaires

| Algorithmes | Meilleur MAPE sur les quatre modèles |
|---|---|
| Régression linéaire | 50 |
| CART | 49 |
| Forêts aléatoires | 52 |
| Perceptrons | 47 |



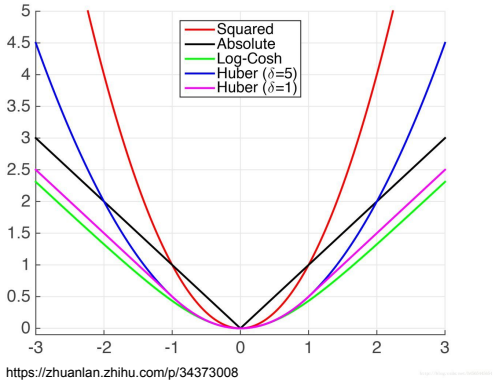

Modèle 1: t, td, hu, dd et ff
Modèle 2: t, td, hu, dd, ff et forecast
Modèle 3: t, td, hu, dd, ff, forecast, coordonnées et saison
Modèle 4: t, hu, dd, ff, forecast, coordonnées et saison

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Dense_n1 (Dense)             (None, 64)                384
_________________________________________________________________
Dense_n2 (Dense)             (None, 64)                4160
_________________________________________________________________
Dense_n3 (Dense)             (None, 32)                2080
_________________________________________________________________
Output (Dense)               (None, 1)                 33
=================================================================
Total params: 6,657
Trainable params: 6,657
Non-trainable params: 0
_________________________________________________________________
```

# XGBoost

Nous approchons le gradient et l'hessienne de MAPE au moyen de Huber loss.

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta\,|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$




https://zhuanlan.zhihu.com/p/34373008

MAPE = 31.417

# Réseau de neurones

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1          [-1, 136000, 32]             448
              Tanh-2          [-1, 136000, 32]               0
            Linear-3          [-1, 136000, 16]             528
              Tanh-4          [-1, 136000, 16]               0
            Linear-5           [-1, 136000, 1]              17
================================================================
Total params: 993
Trainable params: 993
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 6.74
Forward/backward pass size (MB): 100.65
Params size (MB): 0.00
Estimated Total Size (MB): 107.40
----------------------------------------------------------------
```

MAPE = 30.213 sur TrainSet
MAPE = 31.328 sur TestSet

# Classifier

| Classifier | accuracy_score(onDataset2) | accuracy_score(onDataset3) |
|---|---|---|
| ExtraTreesClassifier | 0.789 | 0.741 |
| RandomForestClassifier | 0.818 | 0.725 |
| AdaBoostClassifier | **0.834** | 0.727 |
| BaggingClassifier | 0.715 | 0.715 |
| DecisionTreeClassifier | 0.720 | 0.666 |
| GradientBoostingClassifier | 0.806 | 0.729 |
| XGBClassifier | 0.812 | 0.724 |

# Regressor

```
Avec XGBoost et Dataset2
MAPE = 31.417 -> 31.260   AMELIORATION!
```