Assembler la diversité des modèles classiques et “deep learning” pour développer un pipeline de calibration SPIR performant et générique

Denis Cornet, Lucienne Desfontaines, Fabien Cormier, Carine Marie-Magdeleine, Gemma Arnau, Karima Meghar, Fabrice Davrieux, Gregory Beurier

▶ **To cite this version:**

Denis Cornet, Lucienne Desfontaines, Fabien Cormier, Carine Marie-Magdeleine, Gemma Arnau, et al.. Assembler la diversité des modèles classiques et “deep learning” pour développer un pipeline de calibration SPIR performant et générique. Rencontres HélioSPIR Session 5, Association HelioSPIR, Oct 2019, Montpellier, France. hal-02958421

HAL Id: hal-02958421
https://hal.inrae.fr/hal-02958421

Submitted on 5 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Assembler la diversité des modèles classiques et «deep learning» pour développer un pipeline de calibration SPIR performant et générique.

Cornet D, Beurier G, Cormier F, Meghar K, Davrieux F, Arnau G (Cirad)
Desfontaines L, Marie-Magdeleine C (INRA)

Rencontres HelioSPIR, Session 5 - Place aux « jeunes » !
15 octobre 2019

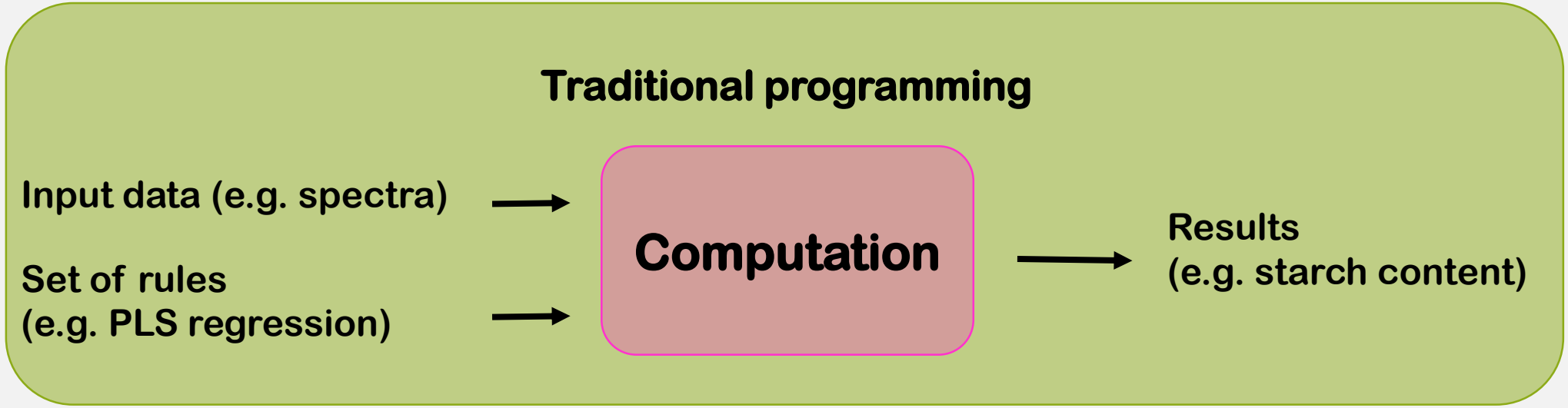
Contenu

- Justification des choix méthodologique
  - Apprentissage automatisé
  - Pipeline générique
  - Assemblage de modèle
- Assemblage de modèles
- Analysis pipeline under development
- Perspectives

Pourquoi l'apprentissage automatisé ?

- calibration SPIR = développement du meilleur modèle **prédictif**
  - Pas/peu d'intérêt de comprendre pourquoi et comment cela fonctionne
  - > 1000 variables explicatives et taille échantillon << variables explicatives
  - Relations non linéaires

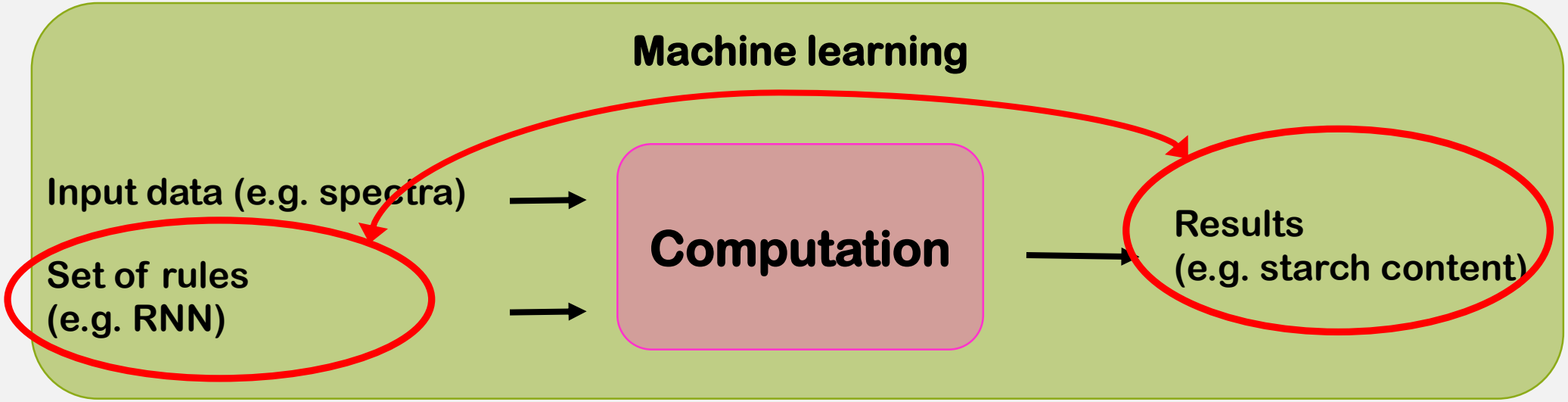
⇒ l'apprentissage automatisé se révèle plus performant



Pourquoi l'apprentissage automatisé ?

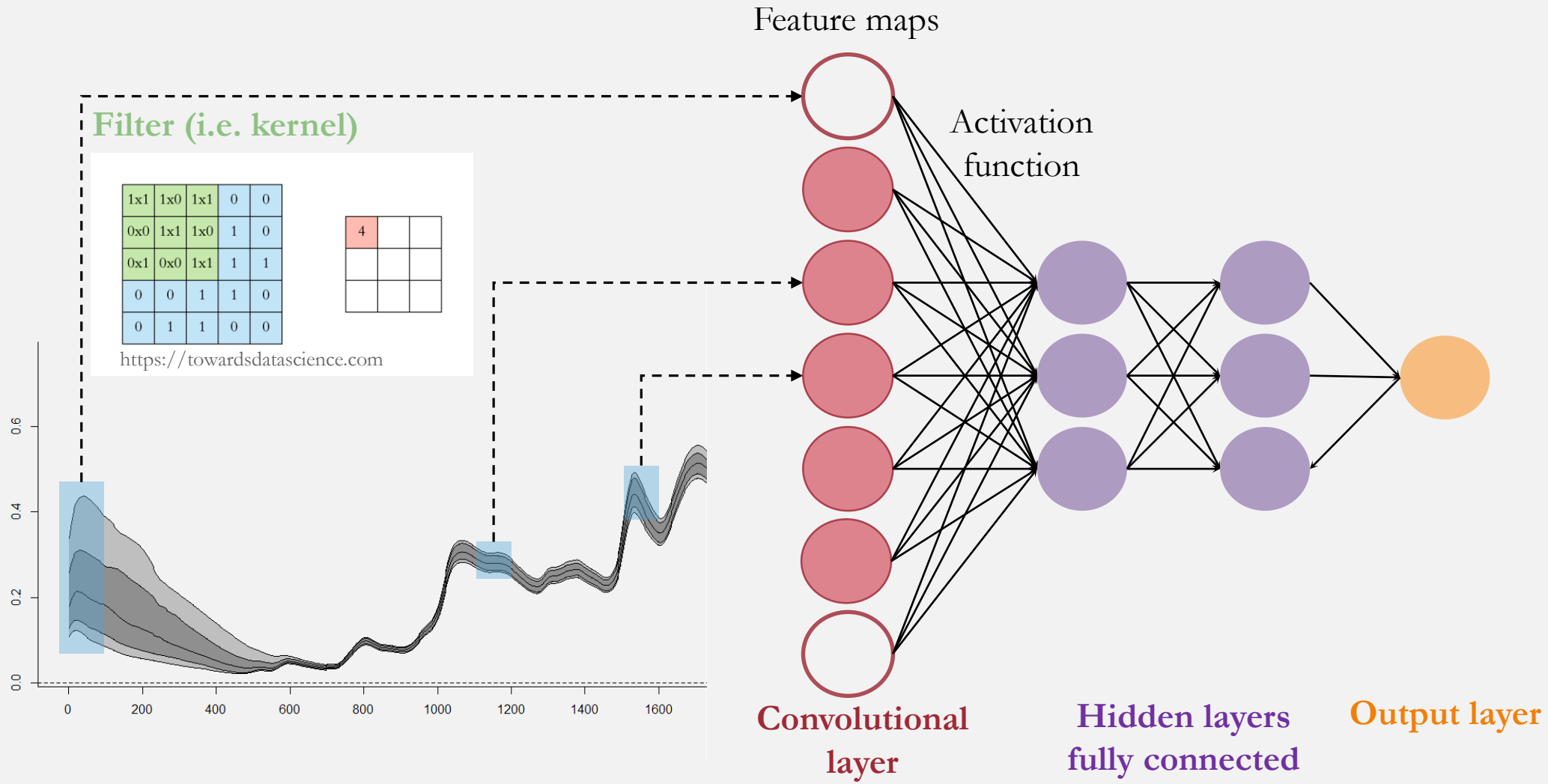
- calibration SPIR = développement du meilleur modèle **prédictif**
  - Pas/peu d'intérêt de comprendre pourquoi et comment cela fonctionne
  - > 1000 variables explicatives et taille échantillon << variables explicatives
  - Relations non linéaires

⇒ l'apprentissage automatisé se révèle plus performant



Exemple d'application

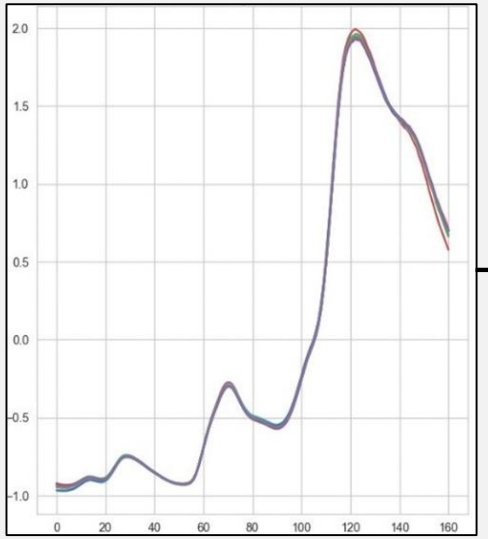
Convolutional network



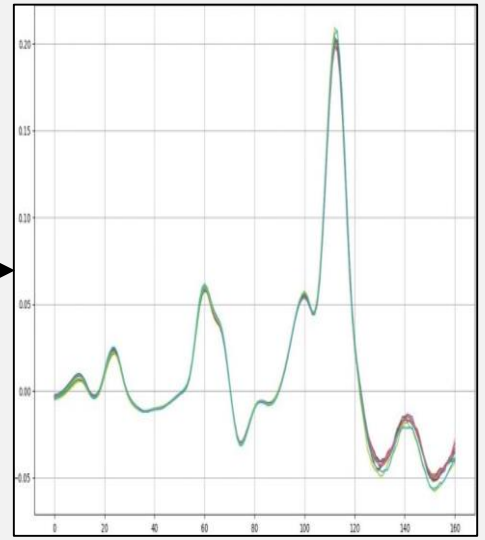
Exemple d'application

Convolutional network

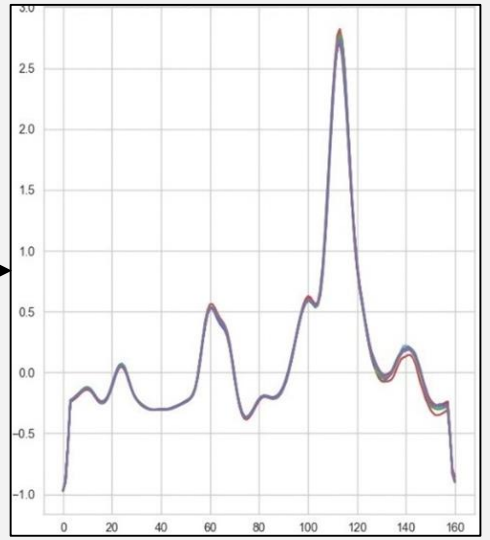
Raw spectra



Savistky-Golay
first derivative

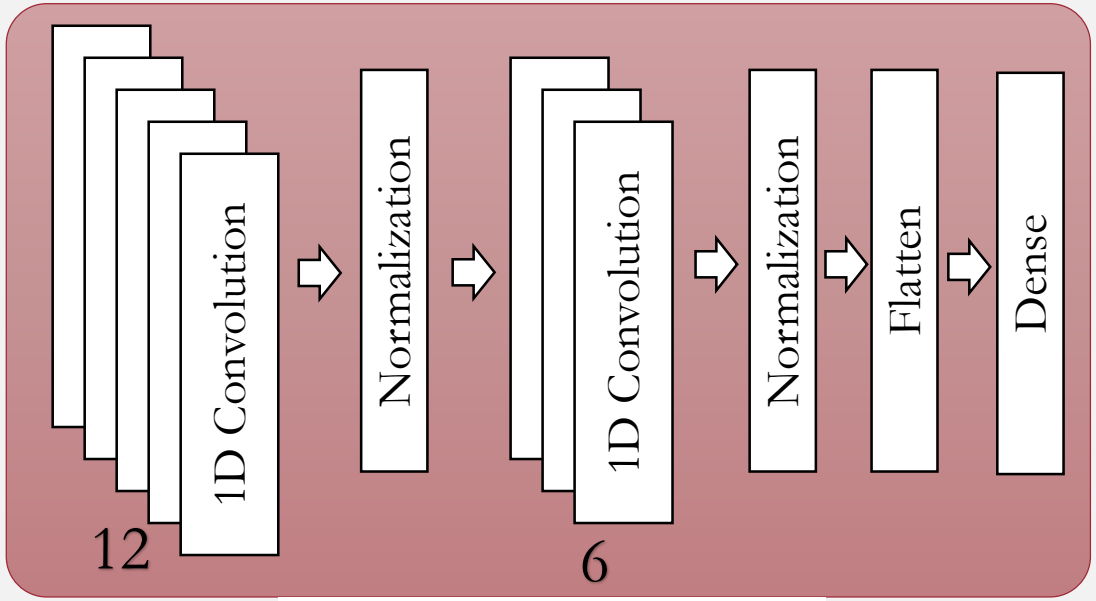


Convolutional
preprocessed



Source: adapted from Cui et Fearn 2018

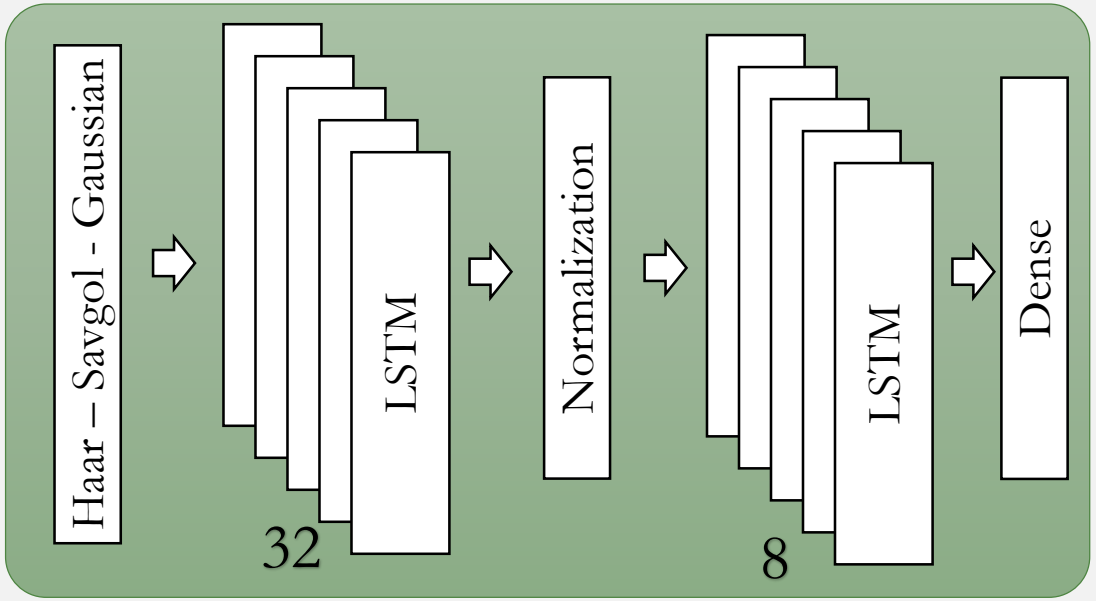
Exemple d'application : texture et caisson du manioc



Deep 1D Convolutional

KFold – 5 Fold 4 Repeats
1500 epochs – **No Hyperparametrization**
40 % Dropout on convolutions

Cooking Time	Texture
Validation Set MAE **19.49 %** ± 2.77%	Validation Set MAE **13.39 %** ± 2.91%
Global Set MAE **10.92 %** ± 0.82%	Global Set MAE **7.67 %** ± 1.74%



Bidirectional Long Short Term Memory

KFold – 5 Fold 4 Repeats
800 epochs – **No Hyperparametrization**
50 % Dropout on input

Cooking Time	Texture
Validation Set MAE **20.32 %** ± 1.85%	Validation Set MAE **10.31 %** ± 0.60%
Global Set MAE **20.74 %** ± 2.26%	Global Set MAE **11.93 %** ± 2.14%

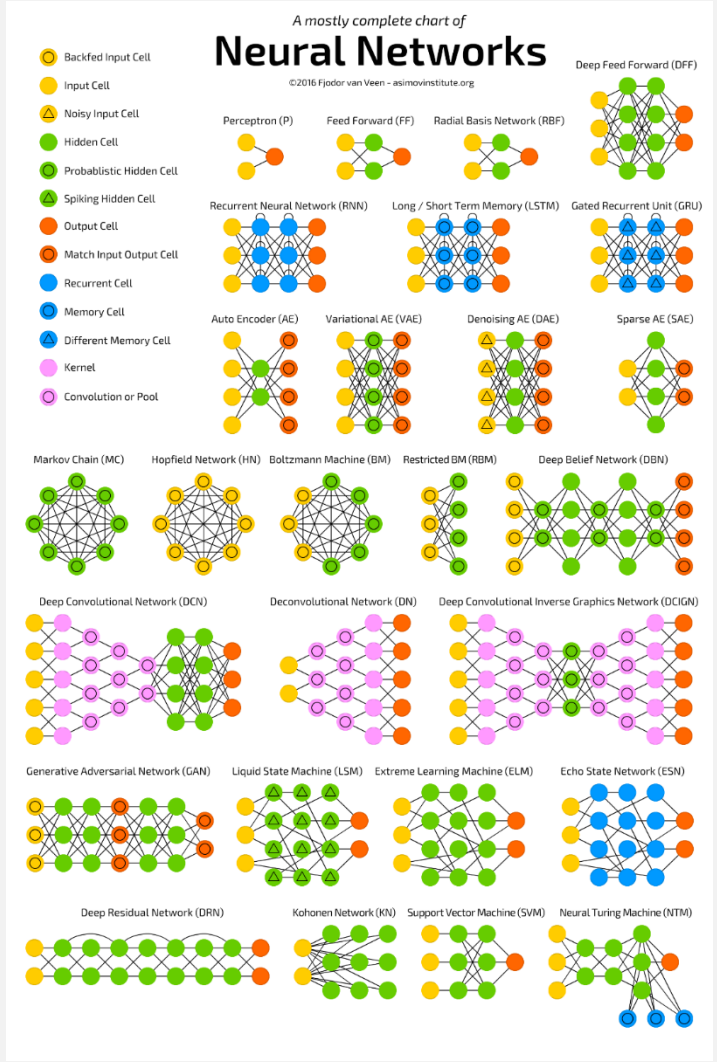
Etude préliminaire avec plusieurs machines

KFold – 5 Fold 4 Repeats
1500 epochs – No Hyperparametrization
40 % Dropout on convolutions

Training Set	**CNN** Spectro 1	**CNN** Spectro 2	**BiLSTM** Spectro 1	**BiLSTM** Spectro 2
Spectro 1 & Spectro 2	7.57% ± 0.19%	13.74% ± 0.45%	7.95% ± 0.49%	9.95% ± 0.66%
Spectro 1 only	7.16% ± 1.05%	25.81% ± 1.51%	6.30% ± 0.73%	26.68% ± 2.76%
Spectro 2 only	17.40% ± 0.44 %	2.93% ± 0.20%	13.28% ± 0.37%	4.44% ± 0.12%
Spectro 1 then Spectro 2	7.28% ± 0.40%	4.45% ± 0.39%	6.64% ± 1.27%	3.87% ± 0.36%
Spectrums Auto-Encoding then Spectro	12.28% ± 1.48%	16.57% ± 1.03%		

Pourquoi un pipeline générique ?

- Démocratisation/vulgarisation de la SPIR apporte de + en + d'utilisateurs naïfs
- La majorité des publications s'attachent à démontrer la supériorité d'une méthode
  - Applique 1 stratégie de modélisation
  - Utilise 1 combinaison spécifique de prétraitement
  - Optimise la calibration pour 1 analyte
- L'utilisateur
  - Dispose souvent de plusieurs analytes (e.g. sucre, amidon, protéine)
  - Se heurte à un mur grandissant de possibiliés de combinaisons de prétraitements/modèles
- Si la combinaison optimale différe d'une étude à l'autre, la manière de l'identifier peut être généralisé



Pourquoi un pipeline générique ?

Construction : les prétreatements

- Noise correction (additive and multiplicative effects)
  - **Multi scatter correction** (remove some physical effects like particle size or Shining Light Reflection)
  - **Savitsky-Golay smoothing and derivative**
- Enhance contrast
  - Discret Wavelet Transform: **Haar transform**
  - **SD feature selection**
- Baseline correction
  - **Detrend**
  - **Continuum removal** (albedo normalization)
  - **Derivatives**

⇒616 combinations of 7 pretreatments

Pourquoi un pipeline générique ?

Construction : les modèles

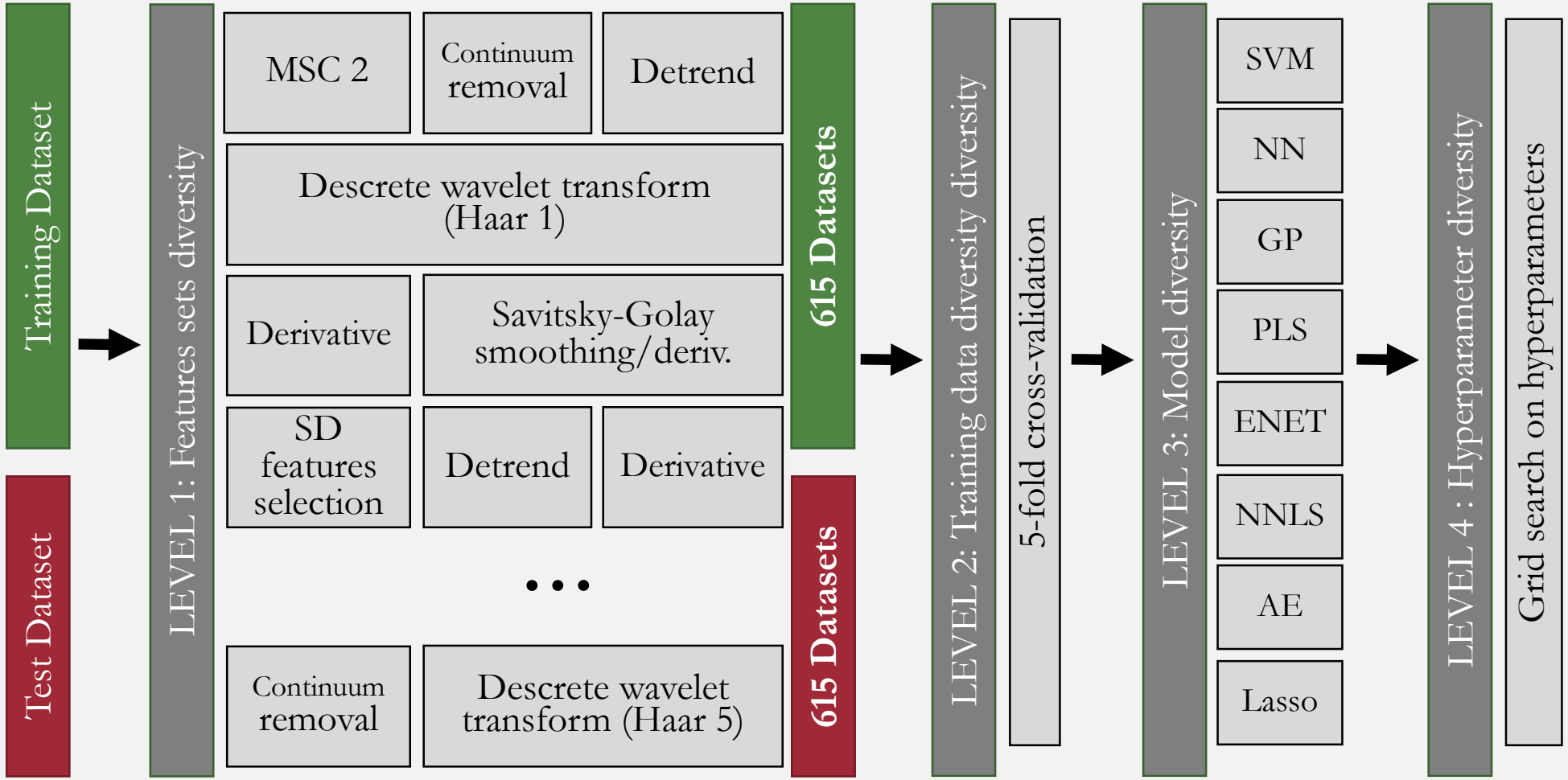
- Traditional programming
  - **Partial least squares regression** (PLS)
  - **Principal component regression** (PCR)
  - **Non negative least squares** (NNLS)
- Machine learning
  - Regularization
    - **Lasso regression** (least absolute shrinkage and selection operator, LASS)
    - **Elastic net regularization** (ENET)
  - Bayesian
    - **Gaussian process** (GP)
  - Support vector machine
    - **Linear support vector regression** (SVMl)
    - **Radial support vector regression** (SVMr)
  - Neural networks
    - **Neural network** (NNET)
    - **LSTM, RNN…**

Hyperparameter tuning:

- Grid selection
- Random selection

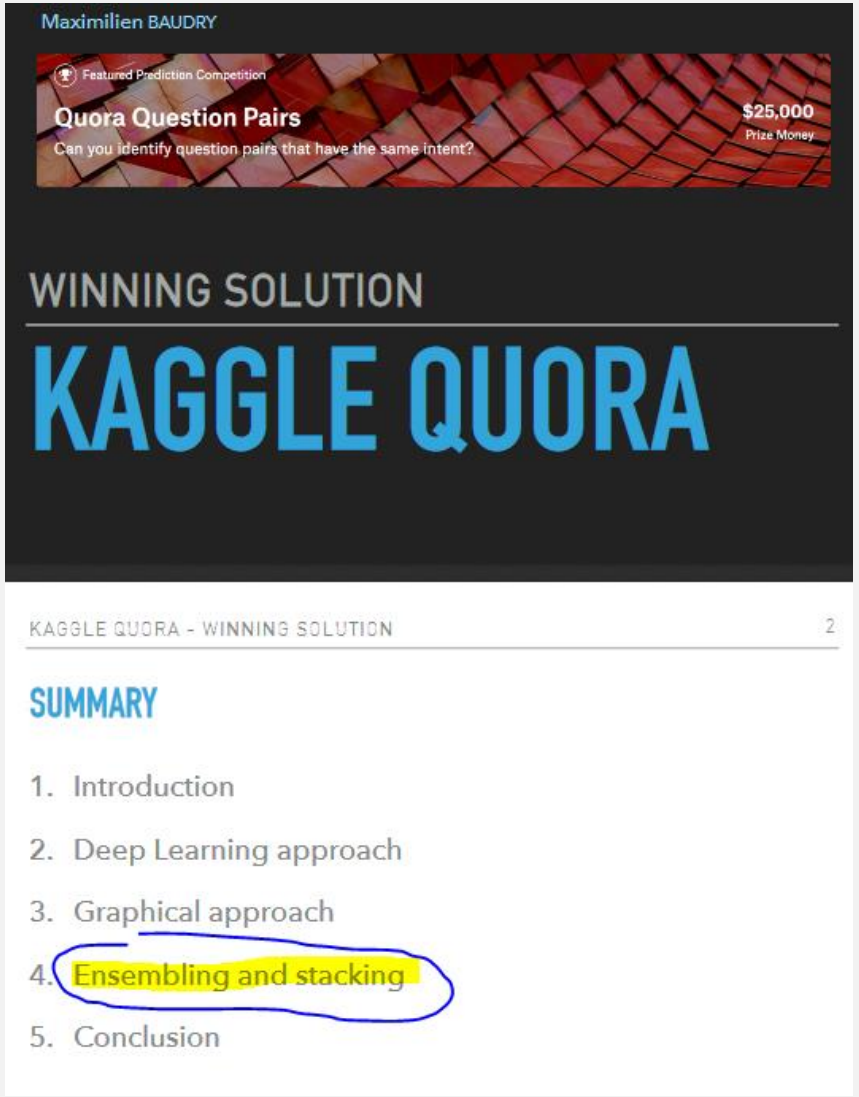
Pipeline d'analyse en cours de construction

Stacking avec R, python, keras et tensorflow



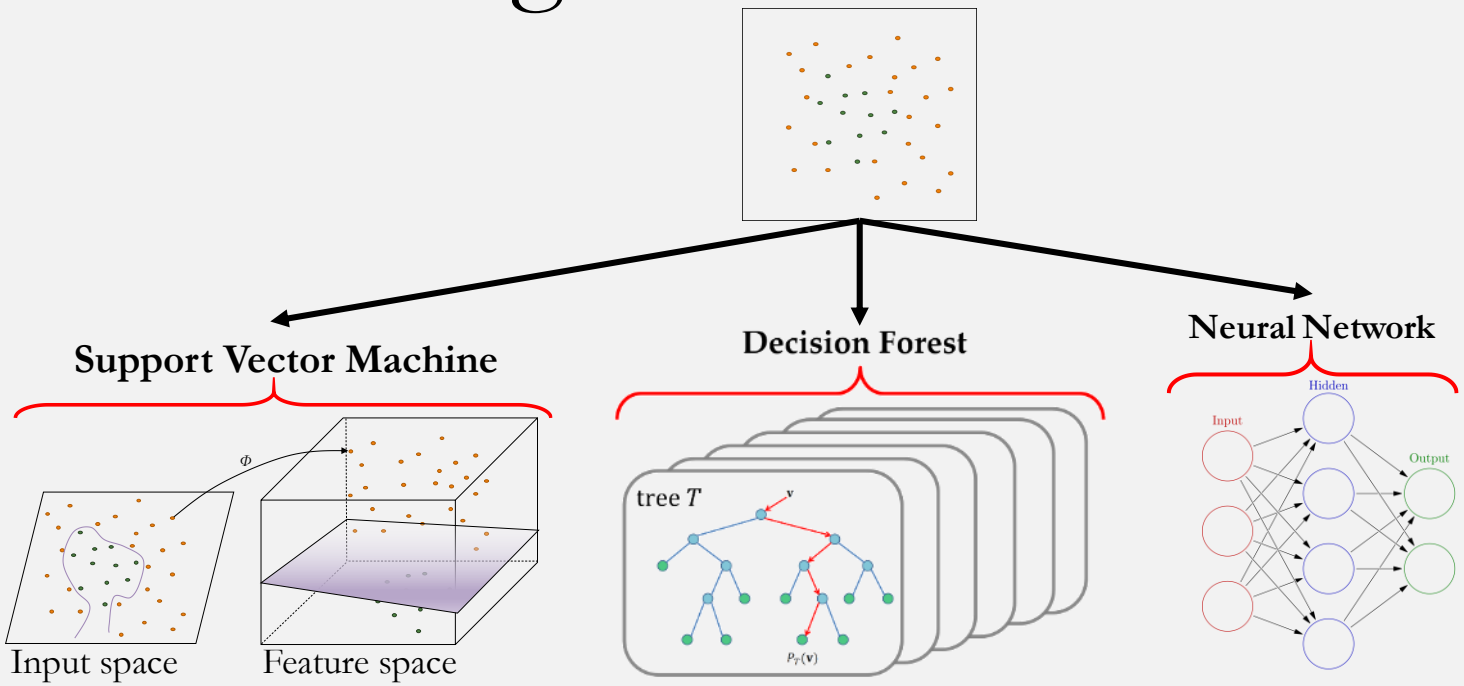
Pourquoi l'assemblage de modèles ?

- Diversité des combinaisons de prétraitements/modèles/analytes
- A un seul modèle
  - Ne bénéficie pas de toute l'information générée pour le sélectionner
  - Risque de s'adjuster au bruit lorsque très/trop spécifique
- Dans le monde singulier des competitions de données (kaggle), l'immense majorité des équipes gagnantes utilise une combinaison de modèle plutôt qu'un modèle optimal
- L'assemblage de modèles de base au sein d'un métamodèle s'appellee l'assemblage de modèle
- Utilisé pour la première fois en 1992 par Wolpert mais il faut attendre 2007 pour avoir une démonstrtion mathématique de son avantage (van de Laar *et al.*)



Techniques d'assemblage

Moyenne

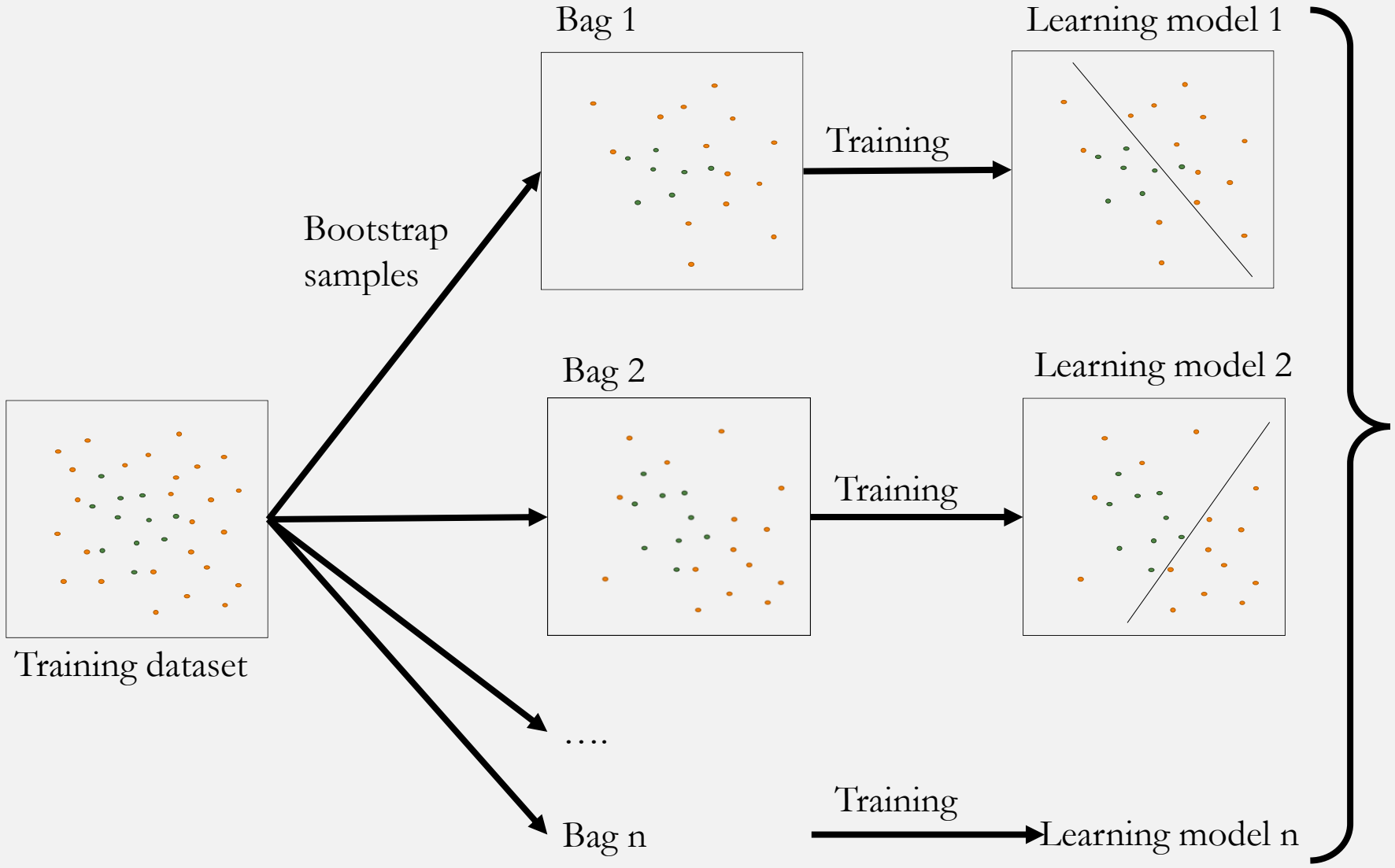


- Simple (Pred SVM + Pred RF + Pred NN) / 3
- Weighted 0,15 Pred SVM + 0,5 Pred RF + 0,35 Pred NN

Techniques d'assemblage

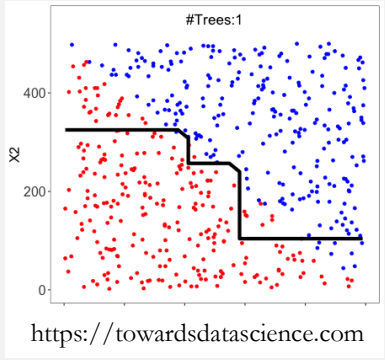
Bagging

Bagging (bootstrap aggregation, e.g. randomforest)



Aggregation (mean…)

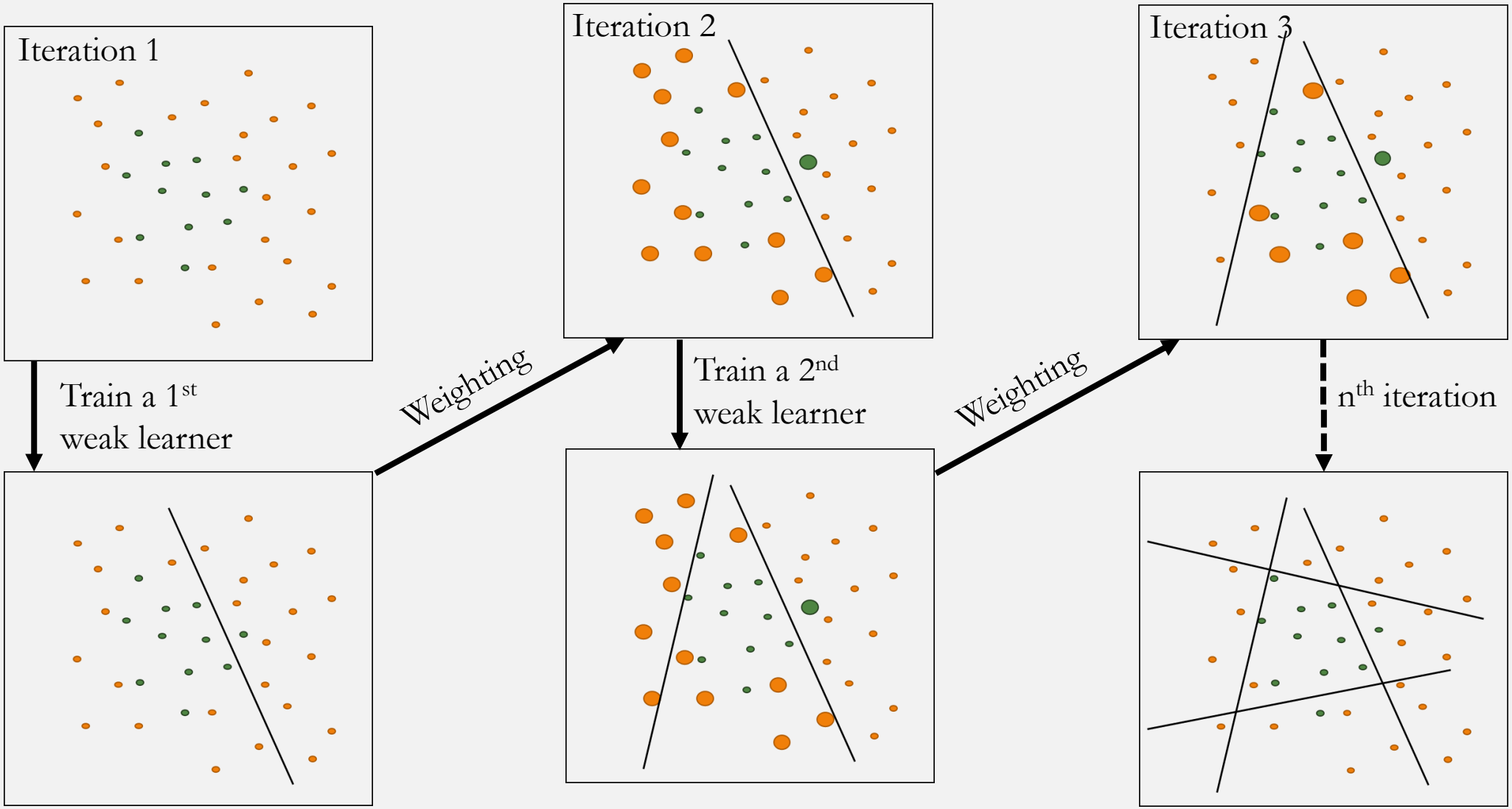
Prediction



https://towardsdatascience.com

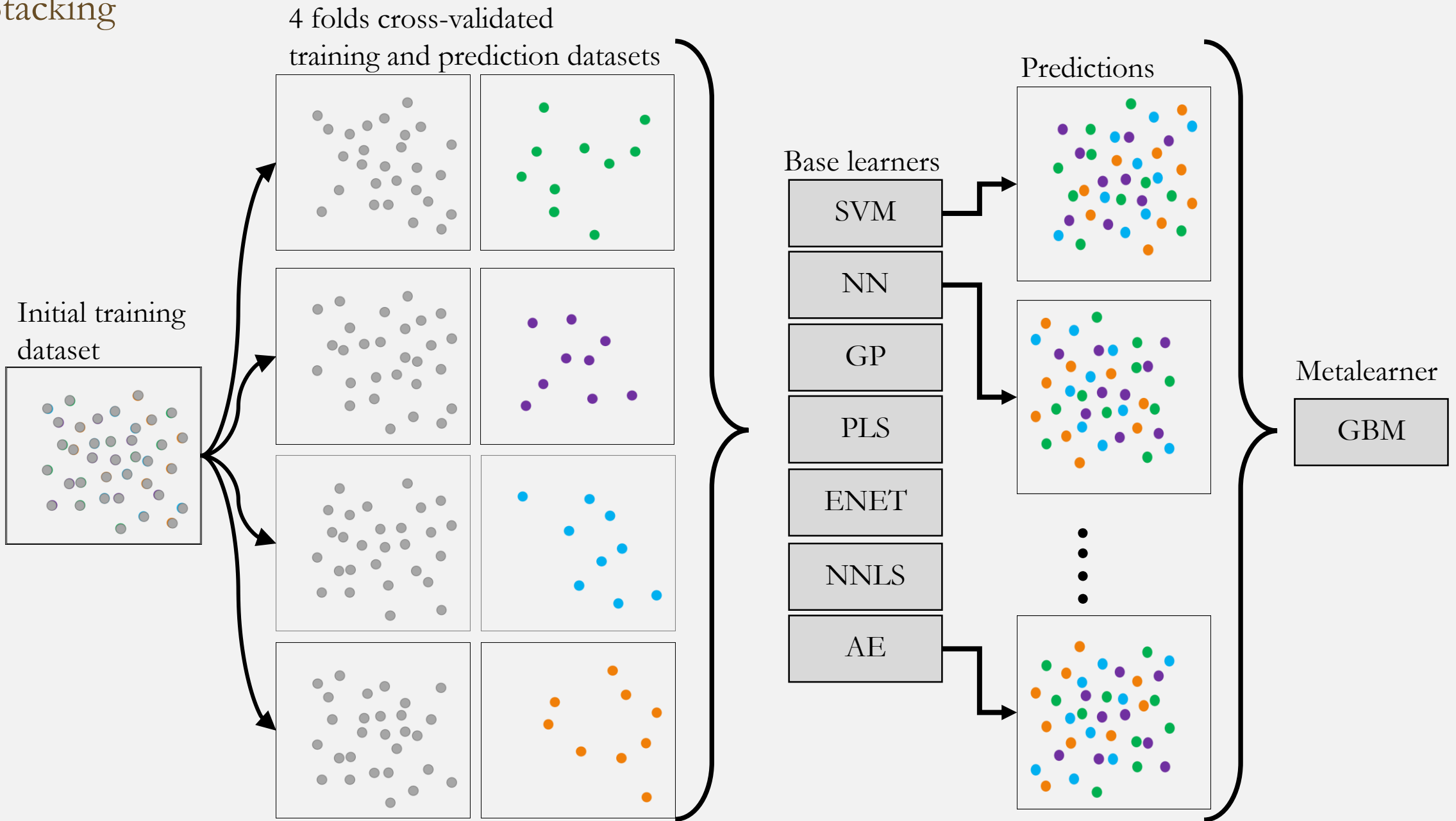
Techniques d'assemblage

Boosting



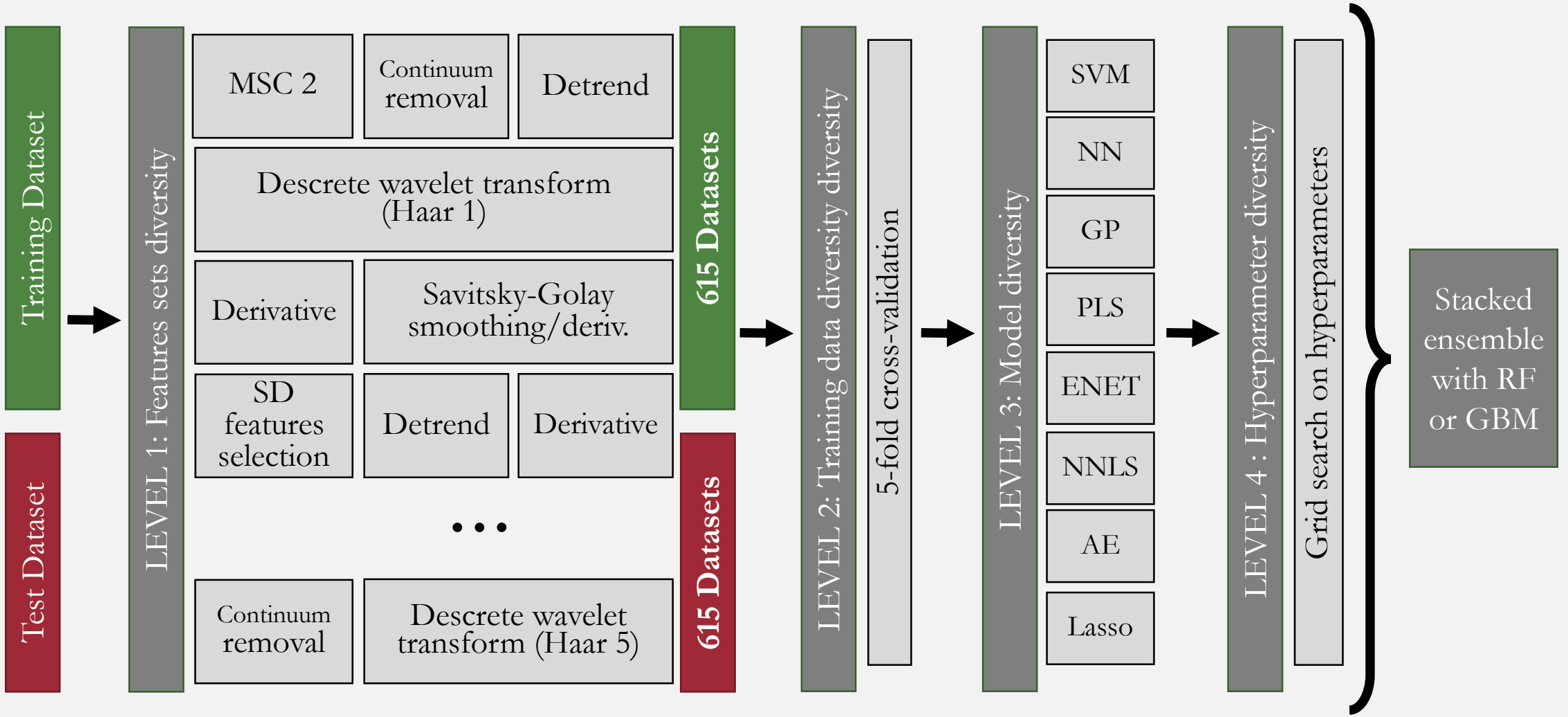
Technique d'assemblage

Stacking



Pipeline d'analyse en cours de construction

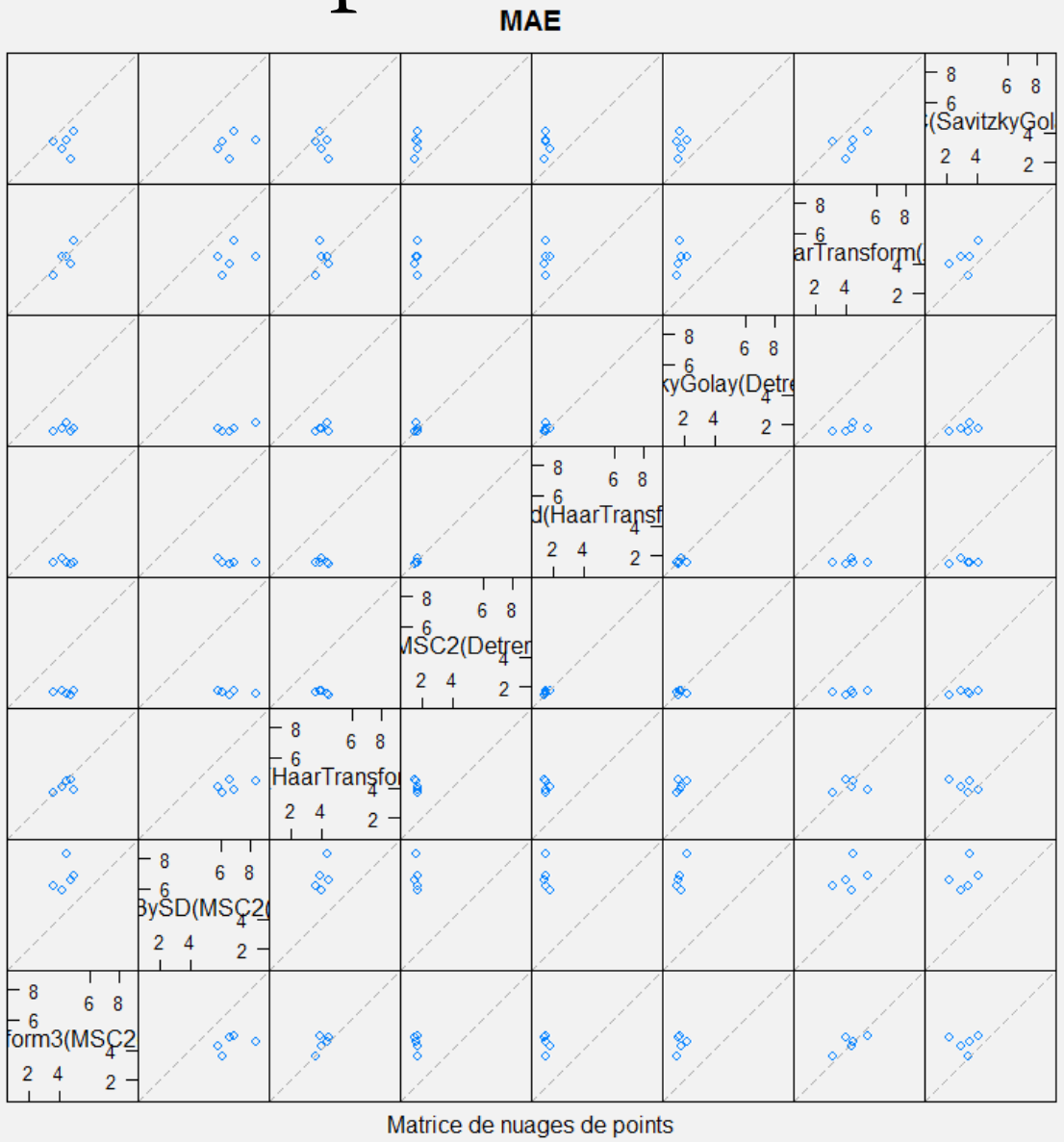
Stacking avec R, python, keras et tensorflow



Analysis pipeline under development

Base model choice

- Conditions
  - Performant
    - ⇒Strong learners (MAE<1)
    - ⇒Steady learners (low variation between cross validated samples MAE)
  - Diverse
    - ⇒Correlation: Pearson < 0.9
    - ⇒Distribution: Kolmogorov-Smirnov > 0.1
- 6150 models → 8 base learners



Perspectives

- Finaliser la construction du pipeline
- Jouer avec les résultats d'analyse (performance relative, analyse de sensibilité….)
- Tester la généricité du pipeline (produits, des tailles de jeux de données, specromètres…) => projet RTBfoods
- Tester la plus-value de travailler sur un vecteur d'analytes plutôt qu'un seul analyte à la fois (meilleure gestion des outliers, …)
- Tester la nouvelle generation d'algorithmes d'apprentissage profond (e.g. RNN, CNN, LSTM, attention based NN)











Merci