# Stacking Generalisé à 2 Niveaux : Analyse Comparative Multi-Datasets

Techniques Prédictives – Projet de Fin de Module

Mohamed Amine Bellatreche  Soltan Rezegia

Université USTO – ING 4 Data Science

*Module : Techniques Prédictives*

*Encadré par : Pr.* Bouziane *H.*

Année universitaire 2024–2025

# Sommaire

# Pourquoi les méthodes d'ensemble ?

En classification, un seul modèle capture rarement toute la structure des données. Chaque algorithme possède ses propres **biais inductifs** :

- **Random Forest** : découpe l'espace par arbres aléatoires
- **SVM** : cherche un hyperplan de marge maximale
- **Régression logistique** : frontière linéaire probabiliste
- **KNN** : vote par proximité dans l'espace
- **Naïve Bayes** : hypothèse d'indépendance conditionnelle

**Idée fondatrice**

Si ces modèles se trompent sur des observations *différentes*, on peut combiner leurs prédictions pour réduire l'erreur globale.

**Question de recherche**

Le stacking à 2 niveaux améliore-t-il systématiquement les performances par rapport aux modèles individuels ? Sous quelles conditions ?

# Le stacking : principe général

Le **stacking** (ou *stacked generalization*), proposé par Wolpert (1992), repose sur un principe simple mais puissant :

1. Entraîner $M$ modèles de base (niveau 0) sur les données d'entraînement
2. Collecter leurs prédictions comme **méta-features**
3. Entraîner un **méta-modèle** (niveau 1) qui apprend la combinaison optimale

Contrairement au *voting* simple (moyenne), le méta-modèle apprend à pondérer adapté aux forces de chaque modèle.

## Formulation

Soit $h_1, h_2, \ldots, h_M$ les modèles de base. Pour une observation $x$, on construit :

$$z = \big(h_1(x),\ h_2(x),\ \ldots,\ h_M(x)\big)$$

Le méta-modèle $g$ prédit :

$$\hat{y} = g(z) = g\big(h_1(x), \ldots, h_M(x)\big)$$

## Condition clé

Le stacking fonctionne si les modèles de base commettent des erreurs **non corrélées** entre eux.

# OOF vs Blending : deux stratégies

## Out-Of-Fold (OOF)

1. Découper le train en $K$ folds
2. Pour chaque fold : entraîner sur $K-1$ folds, prédire le fold restant
3. On obtient une prédiction OOF pour **chaque** observation du train
4. Sur le test : moyenner les $K$ prédictions

**Avantage** : utilise 100% du train
**Risque** : plus complexe, fuite possible si mal implémenté

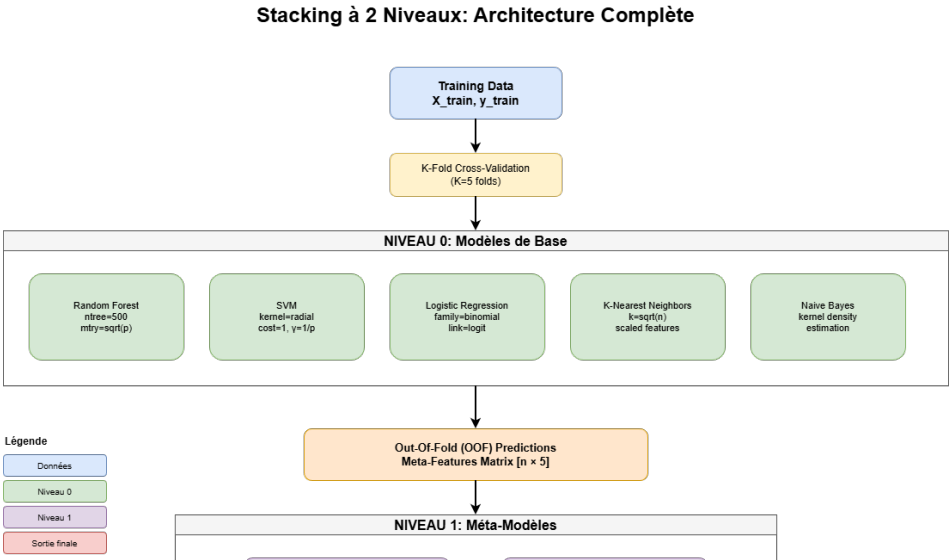
## Blending (holdout)

1. Séparer une portion du train (ex. 25%) comme ensemble de validation
2. Entraîner les modèles sur les 75% restants
3. Prédire sur le holdout $\rightarrow$ méta-features
4. Le méta-modèle s'entraîne sur le holdout uniquement

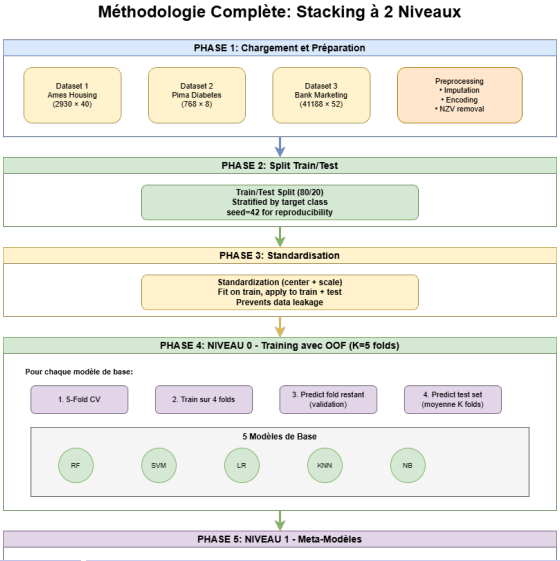
**Avantage** : simple à implémenter
**Risque** : perte de 25% des données pour le méta-modèle

*Dans notre étude, nous implémentons les deux approches pour les comparer directement.*

# Architecture du Stacking à 2 niveaux



# Pipeline expérimental complet



# Choix d'implémentation

## Environnement

- Langage : **R** (version 4.5.2)
- IDE : Jupyter Notebook avec noyau R
- Packages principaux : `caret`, `randomForest`, `e1071`, `glmnet`, `xgboost`, `pROC`

## Préprocessing

- Imputation par médiane
- Encodage one-hot des catégorielles
- Suppression des features à variance quasi-nulle (`nearZeroVar`)
- Standardisation center + scale

## Protocole de validation

- Split train/test : **80%/20%**, stratifié
- Validation croisée interne : **5-fold**
- Graine aléatoire fixée (`seed=42`) pour la reproductibilité

## Métriques d'évaluation

- **Accuracy** : taux de bonne classification
- **AUC** : aire sous la courbe ROC
- **Précision**, **Rappel**, **F1-Score**
- **Temps d'entraînement** (secondes)

# Trois datasets, trois défis

| Propriété | Ames Housing | Pima Diabetes | Bank Marketing |
|---|---|---|---|
| **Source** | AmesHousing (R) | mlbench (R) | UCI Repository |
| **Domaine** | Immobilier | Médical | Financier |
| **Observations** | 2 930 | 768 | 41 188 |
| **Features** | 40 (mixtes) | 8 (num.) | 52 (mixtes) |
| **Cible** | Prix > médiane | Diabète pos/neg | Souscription oui/non |
| **Équilibre** | 50/50 | 65/35 | 89/11 |
| **Split train** | 2 345 | 615 | 32 951 |
| **Split test** | 585 | 153 | 8 237 |

- **Ames** : grand dataset équilibré, features mixtes → cas favorable
- **Pima** : petit dataset, peu de features, classes déséquilibrées → cas difficile
- **Bank** : très grand, fortement déséquilibré (11% positifs) → cas réel

# Pourquoi ces 3 datasets ?

## Ames Housing

- 40 features après encodage
- Features immobilières (qualité, surface, année, etc.)
- Taille suffisante pour l'OOF
- Baseline élevée ($\approx$91%)

## Pima Diabetes

- Seulement 8 features médicales
- 768 observations
- Données bruitées (zéros = NA)
- Teste la robustesse du stacking avec peu de données

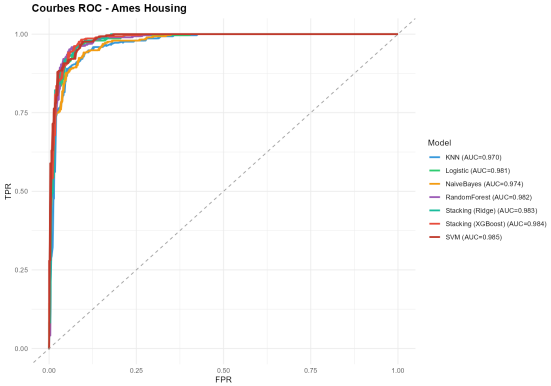
## Bank Marketing

- 52 features après encodage
- Déséquilibre sévère (89/11)
- 41k observations
- Haute dimensionnalité $\rightarrow$ diversité naturelle

L'objectif est de tester le stacking dans des **conditions variées** : taille du dataset, nombre de features, degré de déséquilibre et domaine applicatif.

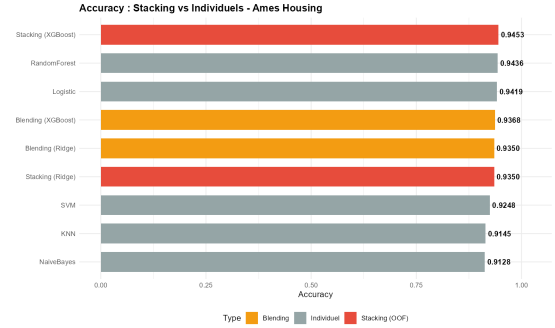
# Dataset 1 : Ames Housing – Résultats

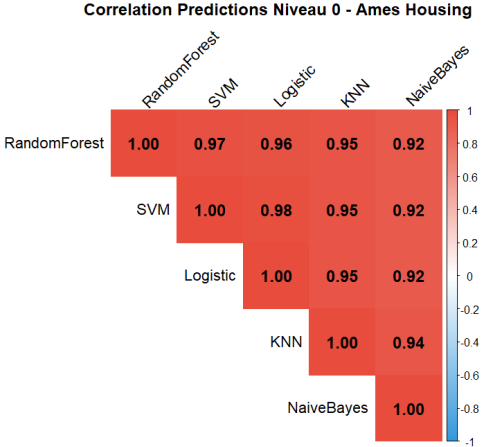
| Modèle | Acc. | AUC | F1 |
|---|---|---|---|
| **Stacking (XGB)** | **0.9453** | **0.9838** | **0.9461** |
| RandomForest | 0.9436 | 0.9818 | 0.9436 |
| Logistic | 0.9419 | 0.9809 | 0.9424 |
| Blending (XGB) | 0.9368 | 0.9837 | 0.9384 |
| Stacking (Ridge) | 0.9350 | 0.9827 | 0.9349 |
| Blending (Ridge) | 0.9350 | 0.9820 | 0.9349 |
| SVM | 0.9248 | 0.9847 | 0.9247 |
| KNN | 0.9145 | 0.9698 | 0.9144 |
| Naïve Bayes | 0.9128 | 0.9743 | 0.9101 |




- Stacking XGBoost en tête : **+0.17 pp** vs RF
- Gain modeste car corrélation très élevée (0.944)
- Tous les modèles dépassent 91% d'accuracy

# Dataset 1 : Ames Housing – Visualisations

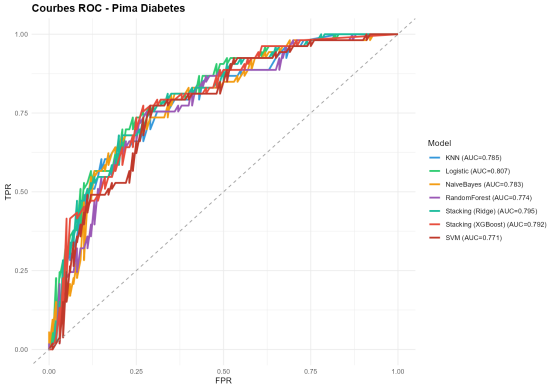



*Gauche : classement par accuracy. Droite : matrice de corrélation des prédictions OOF.*
La corrélation moyenne est de **0.944** : les modèles produisent des prédictions très similaires.

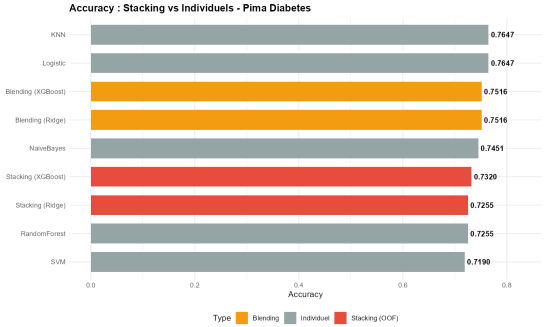
# Dataset 2 : Pima Diabetes – Résultats

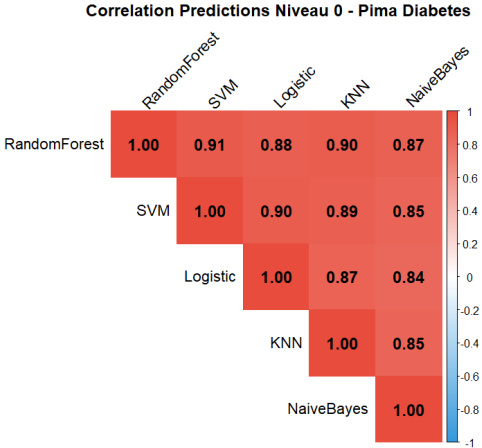
| Modèle | Acc. | AUC | F1 |
|---|---|---|---|
| Logistic | **0.7647** | **0.8070** | 0.6250 |
| KNN | 0.7647 | 0.7855 | 0.6400 |
| Blending (Ridge) | 0.7516 | 0.8011 | 0.6275 |
| Blending (XGB) | 0.7516 | 0.8030 | **0.6607** |
| Naïve Bayes | 0.7451 | 0.7828 | 0.6355 |
| Stacking (XGB) | 0.7320 | 0.7916 | 0.6019 |
| RandomForest | 0.7255 | 0.7741 | 0.5882 |
| Stacking (Ridge) | 0.7255 | 0.7955 | 0.5800 |
| SVM | 0.7190 | 0.7711 | 0.5567 |




- Le stacking **dégrade** les performances : **−3.27 pp**
- Le blending fait mieux (+0.33 pp vs stacking)
- Cause : dataset trop petit (615 obs train) pour l'OOF

# Dataset 2 : Pima Diabetes – Visualisations

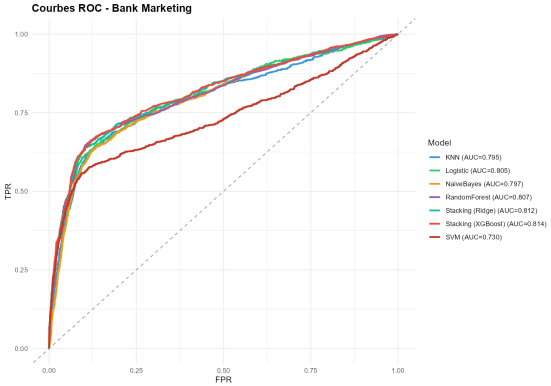



Corrélation moyenne : **0.877**. Avec seulement 8 features, les modèles explorent le même espace et produisent des prédictions semblables. Le méta-modèle n'a pas assez de signal complémentaire pour améliorer la classification.

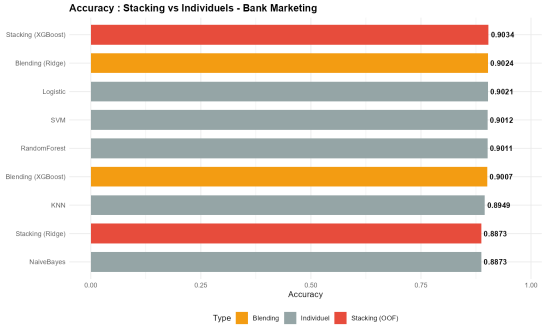
# Dataset 3 : Bank Marketing – Résultats

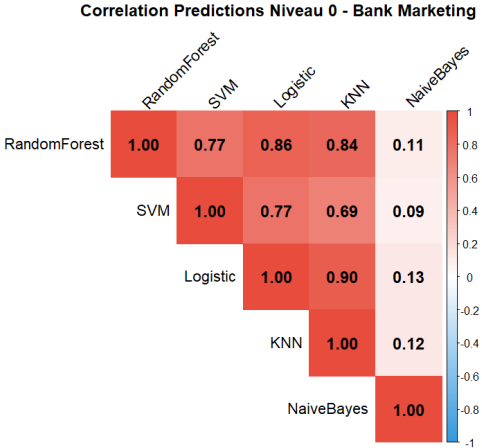
| Modèle | Acc. | AUC | F1 |
|---|---|---|---|
| **Stacking (XGB)** | **0.9034** | **0.8141** | 0.3486 |
| Blending (Ridge) | 0.9024 | 0.8094 | **0.3748** |
| Logistic | 0.9021 | 0.8049 | 0.3623 |
| SVM | 0.9012 | 0.7303 | 0.3361 |
| RandomForest | 0.9011 | 0.8073 | 0.3783 |
| Blending (XGB) | 0.9007 | 0.8104 | 0.3339 |
| KNN | 0.8949 | 0.7946 | 0.1996 |
| Naïve Bayes | 0.8873 | 0.7967 | – |
| Stacking (Ridge) | 0.8873 | 0.8120 | – |




- Stacking XGBoost : meilleur AUC (**+0.92 pp**)
- F1-Scores très faibles → déséquilibre 89/11
- NB et Stacking Ridge n'arrivent pas à prédire la classe minoritaire

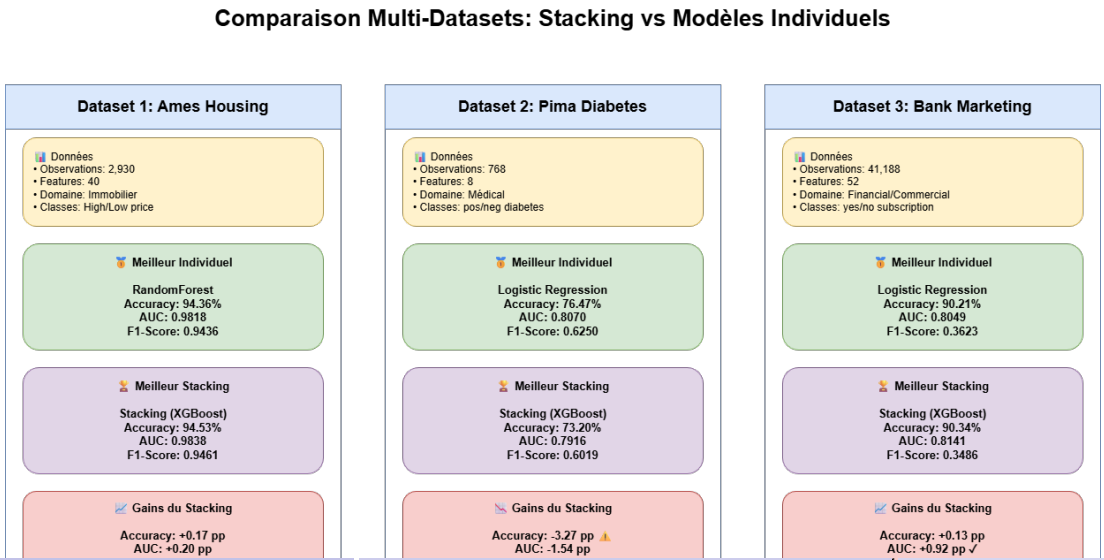
# Dataset 3 : Bank Marketing – Visualisations



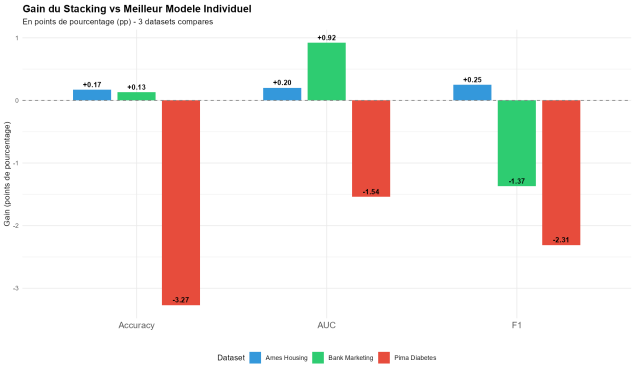




Corrélation moyenne : **0.528**. Naïve Bayes est très décorrélé des autres (≈0.10).
Cette diversité permet au stacking d'exploiter des signaux complémentaires, d'où le gain en AUC malgré le problème de déséquilibre.

# Vue d'ensemble : Stacking vs Individuels



# Gains du stacking par dataset



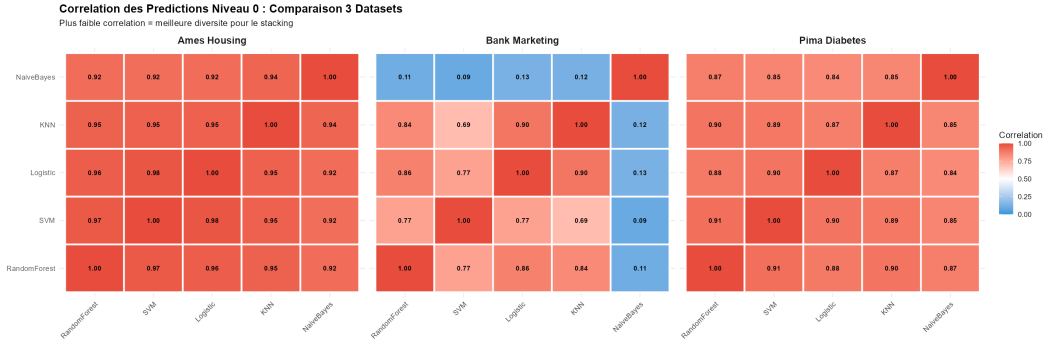

| | Acc. | AUC | F1 |
|---|---|---|---|
| **Ames** | +0.17 | +0.20 | +0.25 |
| **Pima** | −3.27 | −1.54 | −2.31 |
| **Bank** | +0.13 | +0.92 | −1.37 |

Table – *

Gains en points de pourcentage (pp)

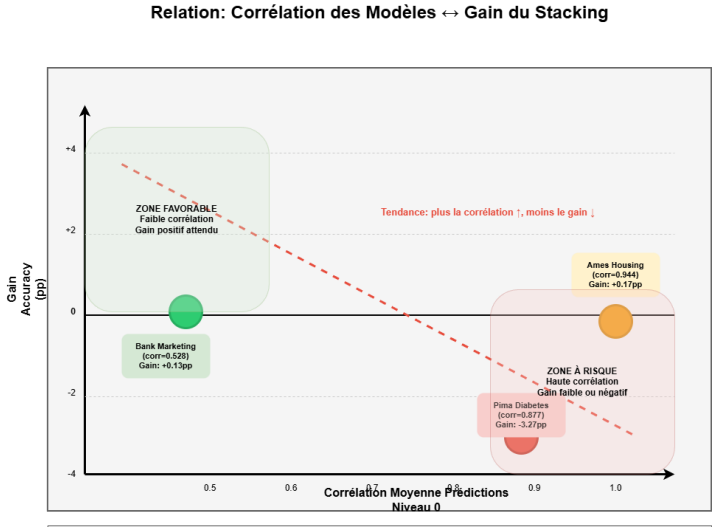
- Ames et Bank : gain en Accuracy et AUC
- Pima : le stacking dégrade sur toutes les métriques
- Le F1 sur Bank est tiré vers le bas par le déséquilibre sévère

# Corrélation des prédictions : facteur déterminant

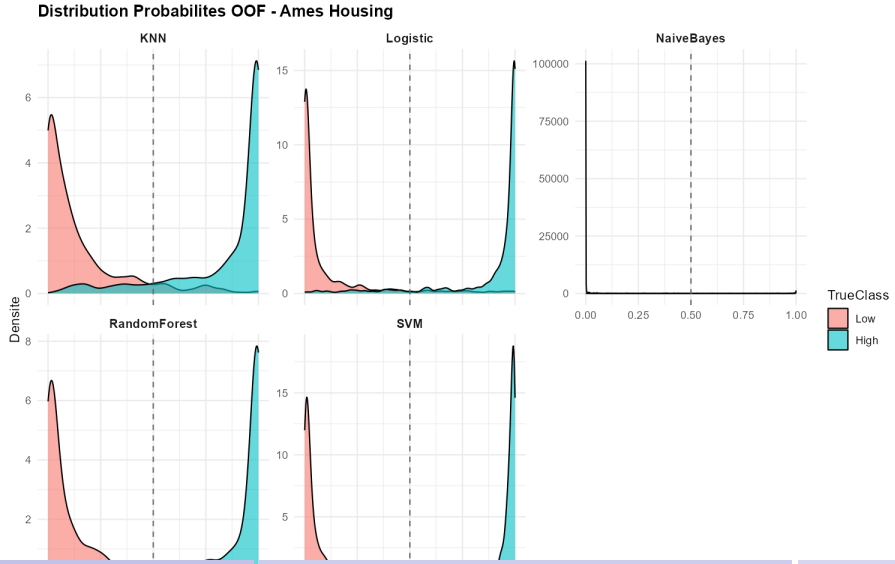



**Ames** : corrélation $\geq 0.92$ partout. **Pima** : corrélation autour de 0.85–0.91. **Bank** : Naïve Bayes quasi-indépendant ($\approx$0.10), diversité marquée.

# Corrélation ↔ gain du stacking



# Distribution des probabilités OOF – Ames Housing



# OOF vs Blending : résultats empiriques

| Dataset | OOF – Blend (pp) |
|---|---|
| Ames Housing | +0.85 |
| Pima Diabetes | −1.96 |
| Bank Marketing | +0.10 |
| **Moyenne** | −0.34 |

L'OOF est généralement supérieur sur les grands datasets car il exploite la totalité des données d'entraînement.

## Quand préférer l'OOF ?

- Dataset de taille suffisante ($> 1\,000$ obs)
- Quand chaque observation compte
- Implémentation rigoureuse (pas de fuite)

## Quand préférer le Blending ?

- Petit dataset où l'OOF risque de sur-apprendre
- Prototypage rapide
- Quand la simplicité prime

# Coût computationnel

| Modèle | Ames | Pima | Bank |
|---|---|---|---|
| RandomForest | 18.1 s | 1.4 s | 450 s |
| SVM | 5.5 s | 0.4 s | 5 275 s |
| Logistic | 0.4 s | 0.1 s | 6.2 s |
| KNN | 1.1 s | 0.3 s | 214 s |
| Naïve Bayes | 0.7 s | 0.1 s | 2.0 s |
| **Stacking (XGB)** | 27.6 s | 3.6 s | 5 951 s |
| **Stacking (Ridge)** | 28.4 s | 3.9 s | 5 973 s |

**Constat**

Le stacking cumule le temps de **tous les modèles de base** (5 modèles $\times$ 5 folds) plus le méta-modèle.

Sur Bank Marketing, le SVM prend à lui seul **1h28** sur les 5 folds, ce qui porte le stacking à près de **1h40** au total.

*Pour un gain de +0.13 pp en accuracy, la question du rapport coût/bénéfice se pose.*

# Insights clés

## Conclusions et Insights Clés

Stacking multi-datasets (Ames, Pima, Bank Marketing)

### 1. Diversité & Corrélation

Corrélation < 0.7 : stacking très rentable
Corrélation 0.7-0.9 : gains modérés
Corrélation > 0.9 : gain faible ou négatif

Exemples : Bank Marketing (corr=0.53, gain AUC +0.92pp) vs Ames Housing (corr=0.94, gain +0.17pp).

### 2. Taille du dataset

Grand dataset (> 2k obs) : OOF stable
Dataset moyen : vigilance sur l'overfit
Petit dataset (< 500) : Blending ou bootstrap

Pima (768 obs) : stacking perd -3.27pp vs meilleur modèle.
Bank (41k obs) : stacking gagne +0.13pp accuracy.

### 3. Haute dimensionnalité

RF sélectionne des features, SVM exploite les noyaux, KNN souffre de la malédiction et NB repose sur des hypothèses fortes.

Ensemble, ces comportements produisent des prédictions complémentaires pour le stacking.

### 4. OOF vs Blending

OOF utilise 100% des données d'entraînement pour produire des meta-features. Blending sacrifie 25%.

Résultats : OOF +0.33pp en moyenne
(Ames +0.85, Bank +0.10, Pima -1.96).

### 5. Recommandations pratiques

1. Vérifier la corrélation des prédictions de base (idéal < 0.8).
2. Valider la taille du dataset (si < 500, envisager Blending).
3. Mixer des algorithmes contrastés (arbre, noyau, linéaire, distance).
4. Surveiller le coût : stacking = 5x temps training + meta-model (Bank = 1.6h vs RF seul 7min).

# Synthèse et recommandations

## Ce que l'on retient

1. La **diversité** des modèles de base est le facteur n°1 du succès du stacking
2. Le stacking n'améliore **pas systématiquement** les performances : il faut vérifier les conditions
3. Sur Pima, un simple modèle logistique bat l'ensemble du pipeline
4. L'OOF surpasse le blending sur les grands datasets, mais pas toujours sur les petits

## Guide pratique

**Avant** de déployer un stacking :

1. Calculer la corrélation entre prédictions de base
   - $\rightarrow$ Si $> 0.9$ : gain négligeable probable
   - $\rightarrow$ Si $< 0.7$ : stacking potentiellement utile
2. Vérifier la taille du dataset (idéalement $> 1\,000$ obs pour l'OOF)
3. Privilégier des algorithmes de natures différentes
4. Évaluer le coût computationnel par rapport au gain espéré

# Limites et perspectives

## Limites de notre étude

- Nombre de datasets limité à 3
- Pas d'optimisation fine des hyperparamètres (grid search) pour les modèles de base
- Le déséquilibre de Bank Marketing n'a pas été traité (SMOTE, sous-échantillonnage)
- Un seul split train/test (pas de répétitions multiples)

## Pistes d'amélioration

- Ajouter des modèles plus divers (réseaux de neurones, LDA, gradient boosting simple)
- Appliquer un resampling (SMOTE) avant stacking
- Optimiser les hyperparamètres de chaque modèle de base
- Tester le stacking à 3 niveaux
- Étendre à des problèmes multi-classes et de régression

# Récapitulatif des livrables

## Code et données

- Notebook R complet (`stacking_dual_dataset.ipynb`)
- Pipeline générique réutilisable (`run_stacking_pipeline`)
- 3 datasets préparés et prétraités

## Résultats (CSV)

- Tables de résultats par dataset
- Matrices de corrélation
- Accuracies par fold
- Table comparative cross-dataset
- Table de synthèse pour LaTeX

## Visualisations (40+ fichiers PNG)

- Courbes ROC par dataset
- Barplots d'accuracy
- Heatmaps de corrélation
- Distributions OOF
- Compromis performance/temps
- Comparaisons multi-datasets

## Diagrammes (Draw.io)

- Architecture du stacking
- Workflow méthodologique
- Comparaison des résultats

# Références


Wolpert, D. H. (1992).
Stacked Generalization.
*Neural Networks*, 5(2), 241–259.

Breiman, L. (1996).
Stacked Regressions.
*Machine Learning*, 24(1), 49–64.

Ting, K. M. & Witten, I. H. (1999).
Issues in Stacked Generalization.
*Journal of Artificial Intelligence Research*, 10, 271–289.

van der Laan, M. J., Polley, E. C. & Hubbard, A. E. (2007).
Super Learner.
*Statistical Applications in Genetics and Molecular Biology*, 6(1).

De Cock, D. (2011).

# Merci pour votre attention

Questions ?

Mohamed Amine Bellatreche & Soltan Rezegia
USTO – ING 4 Data Science
Techniques Prédictives – Pr. Bouziane H.