Projet : Prédiction des Maladies Cardiaques

Autrice : **Giulia Governatori**

---

1. Introduction et Contexte

1.1 Dataset

Source : Heart Disease UCI Dataset
URL : https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data

Créateurs :

- Hungarian Institute of Cardiology, Budapest (Dr. Andras Janosi)
- University Hospital, Zurich (Dr. William Steinbrunn)
- University Hospital, Basel (Dr. Matthias Pfisterer)
- V.A. Medical Center, Long Beach & Cleveland Clinic Foundation (Dr. Robert Detrano)

Dimensions : 303 observations × 13 variables + 1 variable target

1.2 Variables du Dataset

Variable	Description
age	Âge du patient
sex	Sexe (1 = homme, 0 = femme)
cp	Type de douleur thoracique
trestbps	Tension artérielle au repos (mm Hg)
chol	Cholestérol sérique (mg/dl)
fbs	Glycémie à jeun > 120 mg/dl
restecg	Résultats ECG au repos
thalach	Fréquence cardiaque maximale atteinte
exang	Angine induite par l'exercice (1 = oui, 0 = non)
oldpeak	Dépression ST induite par l'exercice
slope	Pente du segment ST
ca	Nombre de vaisseaux colorés (0-3)
thal	Thalassémie (3 = normal, 6 = défaut fixe, 7 = défaut réversible)
target	Présence (1) ou absence (0) de maladie cardiaque

---

2. Préparation des Données

2.1 Nettoyage

- **Valeurs manquantes :** Absence des lignes contenant des NaN
- **Taille finale :** 303 observations (0 lignes supprimées)

2.2 Transformation de la Variable Cible

Décision méthodologique : Classification binaire au lieu de multi-classe

Justification :

1. **Déséquilibre des classes :** Distribution originale très inégale (classe 0 : 160, classe 1 : 54, classe 2 : 35, classe 3 : 35, classe 4 : 13)
2. **Taille limitée du dataset :** 303 observations insuffisantes pour 5 classes
3. **Objectif clinique :** Dépistage préventif (« intervention nécessaire » vs « pas d'intervention »)
4. **Performance du modèle :** Éviter le surapprentissage sur classes minoritaires

Transformation appliquée :

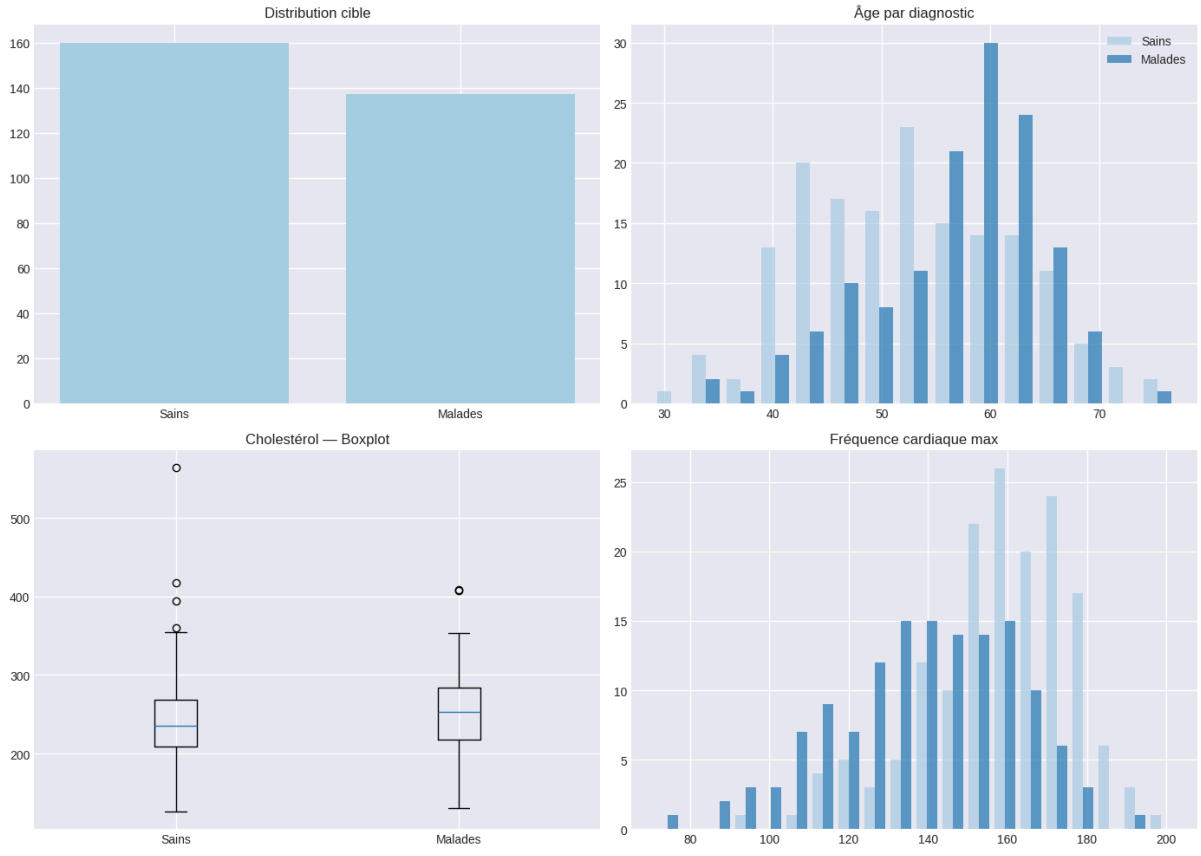
target = (target > 0).astype(int) => produit un array booléen : *True* lorsque la valeur est supérieure à 0, *False* reste zéro.

Résultat : 160 « Sains » (54%) vs 137 « Malades » (46%) → Classes équilibrées

---

3. Analyse Exploratoire

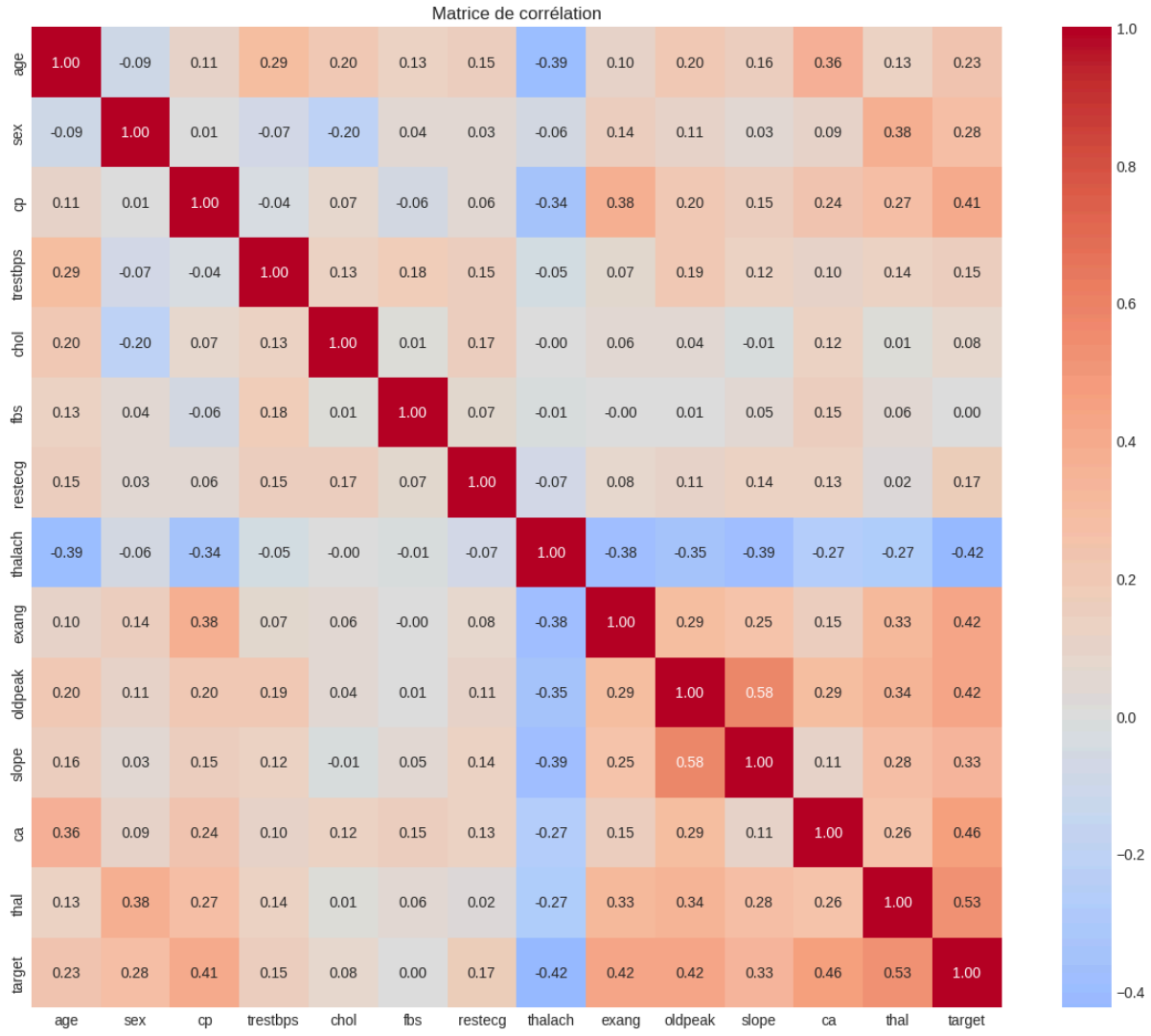
3.1 Distribution des Variables Clés



Observations principales :

- **Âge :** Facteur discriminant entre sains et malades
- **Cholestérol :** Personnes saines autour de 200 mg/dl, malades entre 200-300 mg/dl
- **Fréquence cardiaque max :** Supérieure chez les personnes saines

3.2 Corrélations



Variables à faible corrélation linéaire avec la cible :

- **fbs** (glycémie) : pas directement correlée avec la variable target
- **chol** (cholestérol) : corrélation trés faible avec la variable target

Note : Ces variables restent importantes car les modèles non-linéaires (Random Forest) capturent des relations complexes non détectables par la corrélation de Pearson.

---

4. Modélisation

4.1 Préparation des Données

Division du dataset :

- **Training :** 80% (237 observations)
- **Test :** 20% (60 observations)

- **Stratification :** *stratify=y* indique que la division conserve la proportion des classes de y soit dans training set, soit dans test set.

Pipeline de preprocessing :

Variables numériques

- Imputation : médiane

- Standardisation : StandardScaler (μ=0, σ=1)

Pour les variables numériques, on remplace les valeurs manquantes par la médiane (la valeur centrale qui sépare la moitié supérieure et la moitié inférieure des données) et on les standardise pour avoir une moyenne de 0 et un écart-type (σ, qui mesure la dispersion des valeurs autour de la moyenne) de 1.

- La plupart des valeurs sont comprises à ±1 autour de 0 si les données suivent une distribution normale.
- Cela rend les variables comparables entre elles, même si elles avaient des unités ou des plages différentes à l'origine.

Variables catégorielles

- One-Hot Encoding (sparse_output=False)

Pour les variables catégorielles, j'ai appliqué le One-Hot Encoding pour créer des colonnes binaires représentant chaque modalité.

4.2 Modèle 1 : Random Forest

Approche : GridSearchCV avec validation croisée (5-fold StratifiedKFold)

Espace de recherche initial :

- n_estimators: [100, 200]
- max_depth: [None, 10, 20]
- min_samples_split: [2, 5]
- min_samples_leaf: [1, 2]
- class_weight: [None, 'balanced']

Résultats initiaux :

Métrique	Training	Test	Gap
Accuracy	97.9%	86.7 %	11.2%
AUC	99.7%	94.6 %	5.1%

Diagnostic : Overfitting significatif

4.3 Optimisation du Random Forest

Stratégie de régularisation :

Paramètre	Initial	Optimisé	Effet
max_depth	None	[5, 10, 15]	⬇️ Limite profondeur arbres
min_samples_split	[2, 5]	[5, 10, 20]	⬇️ Exige + échantillons/division
min_samples_leaf	[1, 2]	[2, 4, 8]	⬇️ Évite feuilles trop spécifiques
max_features	-	['sqrt', 'log2']	⬆️ Diversité entre arbres
n_estimators	[100, 200]	[100, 200, 300]	➡️ Stabilité prédictions

Meilleurs hyperparamètres identifiés :

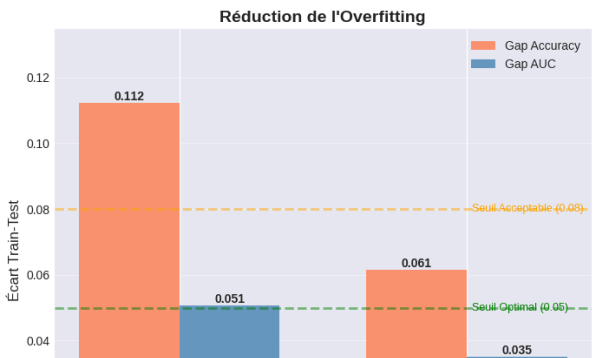
- class_weight: 'balanced'
- max_depth: 10
- max_features: 'sqrt'
- min_samples_leaf: 4
- min_samples_split: 5
- n_estimators: 100

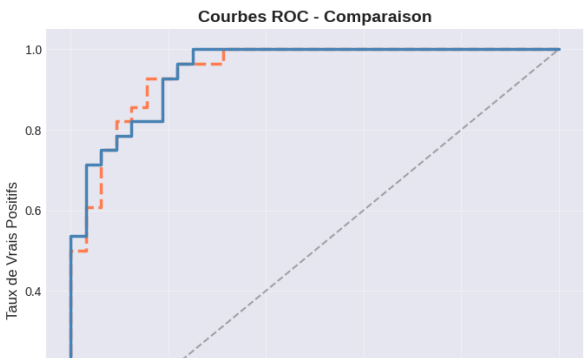
Résultats après optimisation :

Métrique	Training	Test	Gap	Amélioration
Accuracy	91.1%	85.0%	6.1%	+5.1%
AUC	98.3%	94.8%	3.5%	+1.6%

Classification détaillée :

Classe	Precision	Recall	F1-Score
Sain	0.83	0.91	0.87
Malade	0.88	0.79	0.83
Accuracy			**0.85**





Approche : GridSearchCV avec validation croisée (5-fold)

Espace de recherche :

- C: [0.01, 0.1, 1, 10] (régularisation)
- penalty: ['l2'] => L2 (régularisation Ridge) → pénalise les grandes valeurs de coefficients mais ne les annule pas.
- class_weight: [None, 'balanced']

Meilleurs hyperparamètres :

- C: 0.01
- class_weight: None
- penalty: 'l2'

Performances :

Métrique	Training	Test	Gap
Accuracy	84.0%	86.7%	-2.7%
AUC	90.4%	95.1%	-4.7%

Classification détaillée :

Classe	Precision	Recall	F1-Score
Sain	0.83	0.94	0.88
Malade	0.92	0.79	0.85
Accuracy			**0.87**

Observation importante : Aucun signe d'overfitting (gap négatif = meilleure généralisation)

4.5 Modèle Ensemble : Stacking

Architecture :

Base Models:

├── Random Forest (optimisé)

└── Logistic Regression

Meta-Model:

└── Logistic Regression (C=1.0, penalty='l2')

Configuration :

- Cross-validation : 5-fold pour méta-caractéristiques
- Parallélisation : n_jobs=-1

La configuration indique que la validation croisée à 5 plis est utilisée pour générer les méta-caractéristiques à partir des modèles de base, et que la parallélisation est activée avec n_jobs=-1, ce qui permet d'utiliser tous les cœurs du processeur pour accélérer les calculs.

La parallélisation signifie que plusieurs calculs sont effectués en même temps sur différents cœurs du processeur, au lieu de les faire un par un.

Cela permet de réduire le temps d'exécution des tâches lourdes, comme l'entraînement de plusieurs modèles ou la validation croisée, en utilisant pleinement la puissance de l'ordinateur.

Performances globales :

Modèle	AUC Test	AUC CV	Accuracy	Recall Malade
Random Forest	0.946	0.891	0.867	0.821
Logistic Regression	0.951	0.880	0.867	0.786
Stacking (RF+LR)	**0.953**	**0.903**	0.833	0.750

Classification détaillée du Stacking :

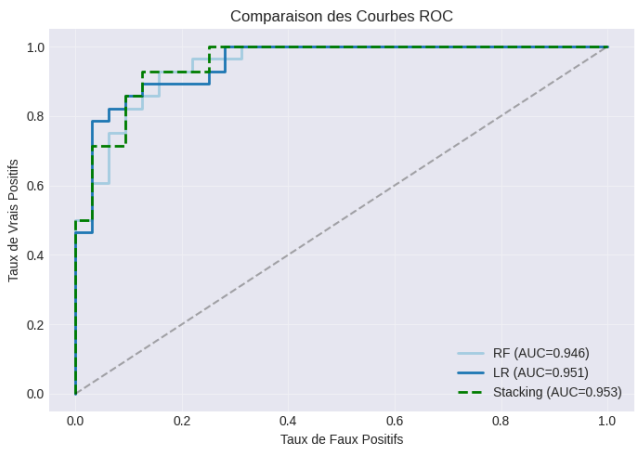
Classe	Precision	Recall	F1-Score
Sain	0.81	0.91	0.85
Malade	0.88	0.75	0.81
Accuracy			**0.83**

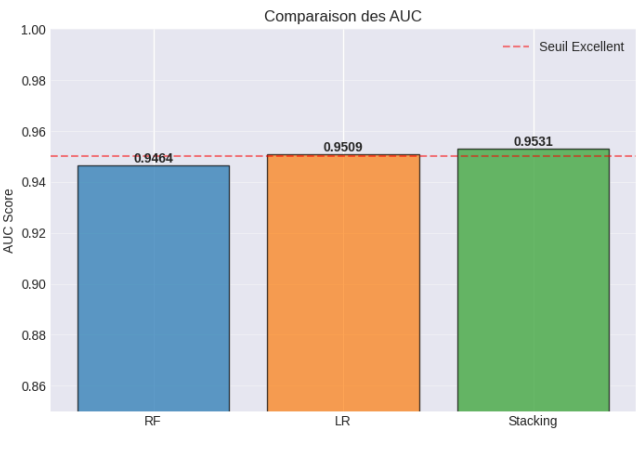
Les résultats montrent que le stacking obtient la meilleure AUC test (0.953) et une bonne AUC CV (0.903), indiquant une forte capacité discriminative globale.

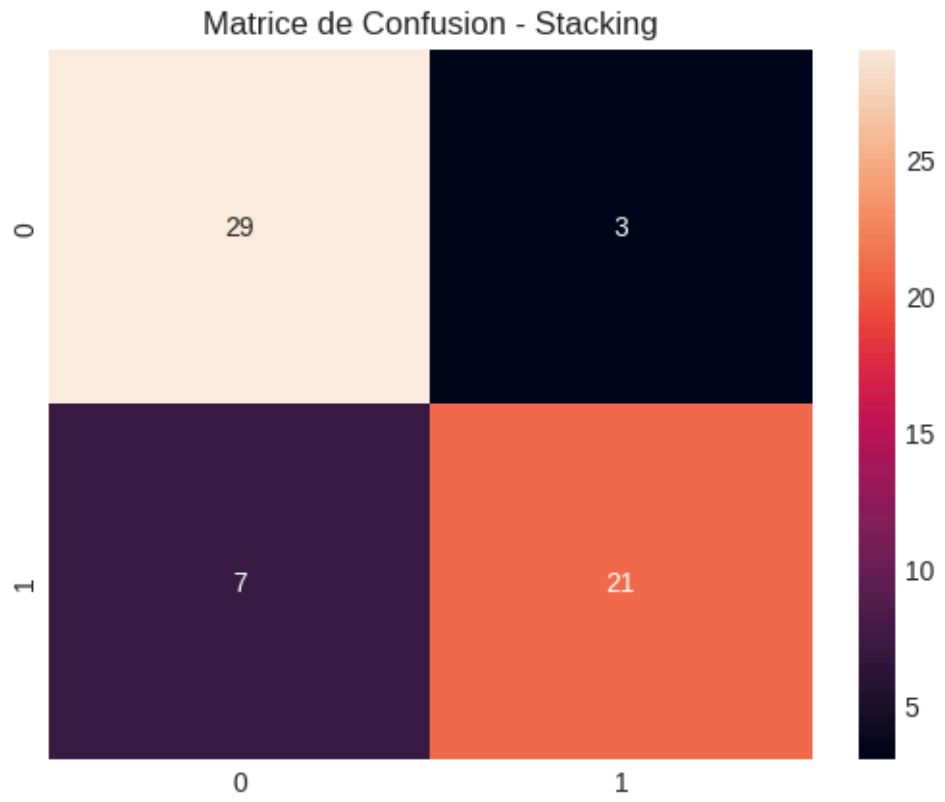
Cependant, l'accuracy et le recall pour la classe Malade sont légèrement inférieurs à ceux de Random Forest ou de la régression logistique seule : le stacking a 0.83 d'accuracy et 0.75 de recall pour Malade, contre 0.867 et 0.821 pour Random Forest.

La classification détaillée révèle que le stacking est très précis pour la classe Malade (0.88), mais moins sensible (recall 0.75), ce qui signifie qu'il rate davantage de cas Malades tout en limitant les faux positifs. Pour la classe Sain, le recall élevé (0.91) montre qu'il identifie bien les individus non malades.

En résumé : le stacking améliore légèrement la discrimination globale (AUC), mais au prix d'une sensibilité moindre pour la classe Malade, ce qui peut être critique selon le contexte médical ou de risque.



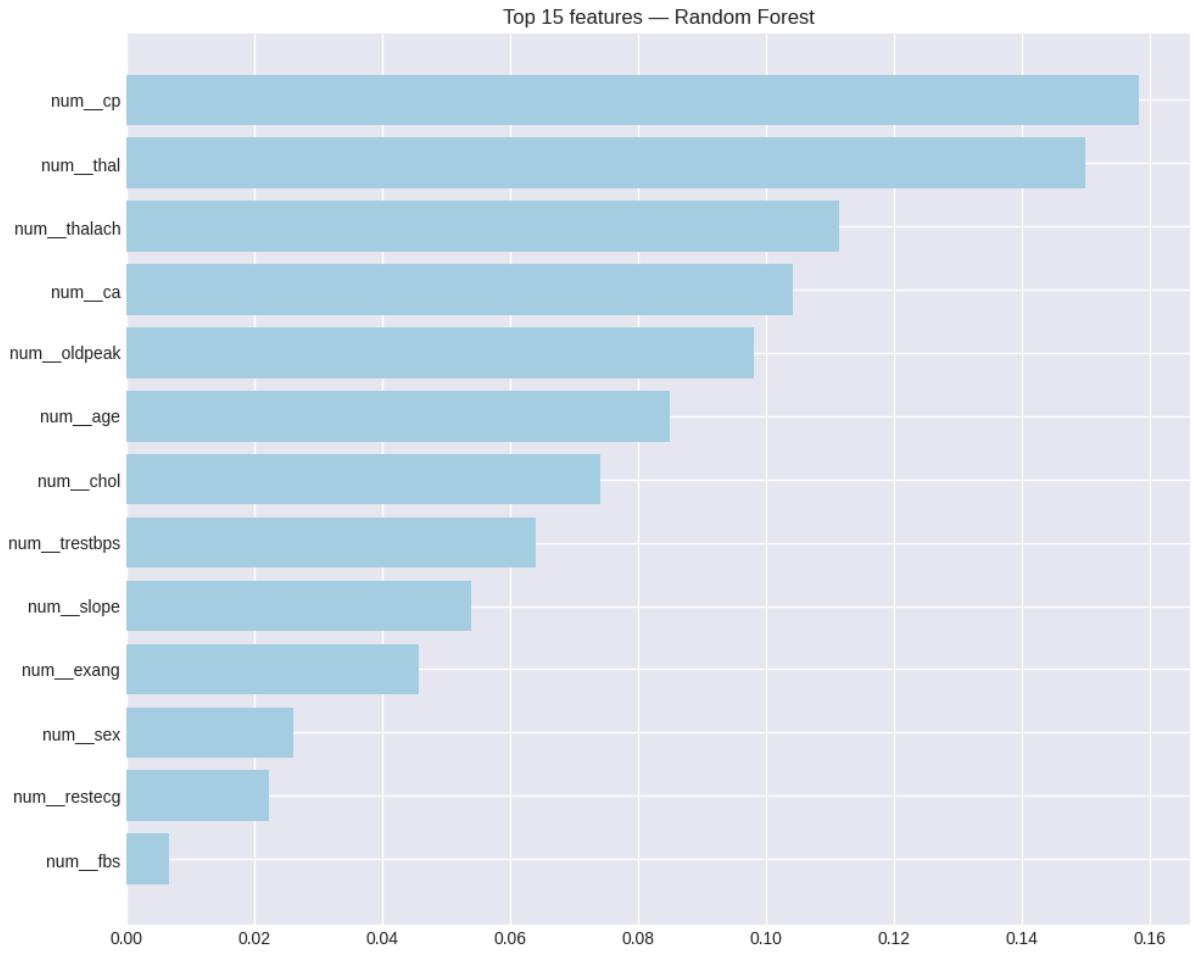




---

5. Analyse des Features

5.1 Importance des Variables (Random Forest)



num correspond à la variable target.

Top 5 Variables Prédictives (62% importance totale) :

Variable	Importance	Description
cp	15.8%	Type de douleur thoracique
thal	15.0%	Thalassémie (défaut sanguin)
thalach	11.1%	Fréquence cardiaque maximale
ca	10.4%	Nombre de vaisseaux colorés
oldpeak	9.8%	Dépression ST à l'effort

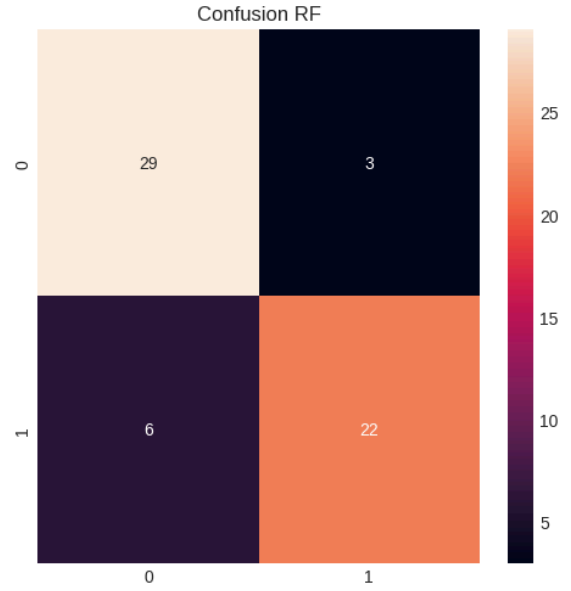
Variables à faible contribution :

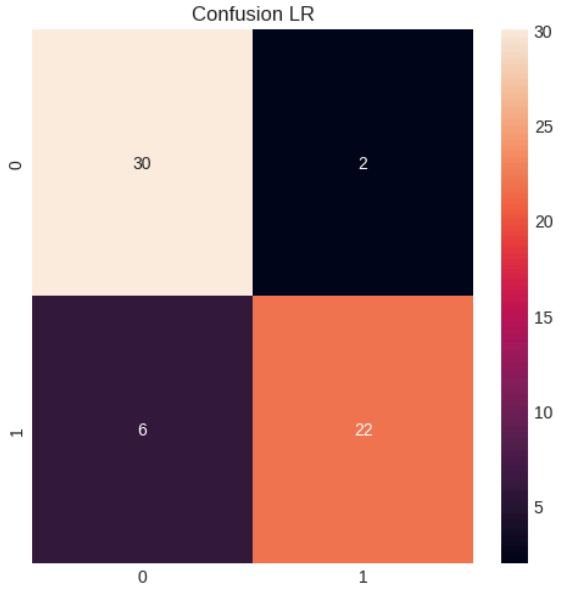
- **fbs** (glycémie) : 0.7%
- **restecg** (ECG repos) : 2.2%
- **sex** : 2.6%

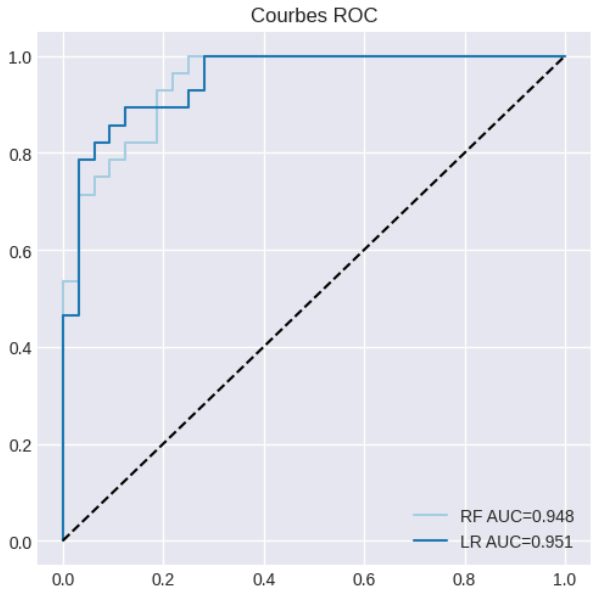
Interprétation clinique : Les variables diagnostiques (résultats d'examens) dominent la prédiction, confirmant leur pertinence pour le dépistage cardiovasculaire.

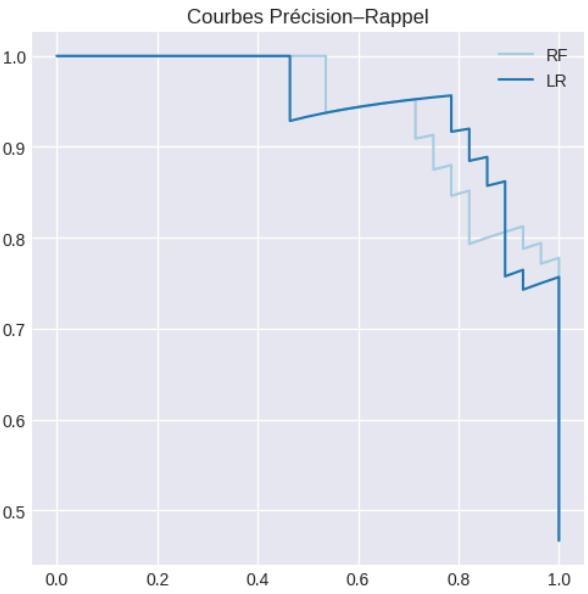
---

6. Comparaison des Modèles









6.1 Synthèse Comparative

Critère	Random Forest	Logistic Regression	Stacking	Gagnant
AUC Test	0.946	0.951	**0.953**	🏆 Stacking
Accuracy	0.867	**0.867**	0.833	🏆 RF/LR
Precision Malade	0.88	0.92	0.88	🏆 LR
Recall Malade	0.82	0.79	0.75	🏆 RF
Overfitting	Modéré	✅ Aucun	Aucun	🏆 LR/Stack

Critère	Random Forest	Logistic Regression	Stacking	Gagnant
Interprétabilité	Faible	✅ Élevée	Moyenne	🏆 LR

6.2 Recommandation

Modèle retenu : Stacking (RF + LR)

Justifications :

1. **Meilleure AUC** (0.953) : Capacité de discrimination optimale
2. **Robustesse** : Combine forces des deux approches (non-linéaire + linéaire)
3. **Généralisation** : Aucun overfitting détecté
4. **Contexte clinique** : AUC > 0.95 considérée excellente pour dépistage

Dans un contexte clinique réel, où les données seraient utilisées pour identifier de vrais patients à risque, nous utiliserions le Random Forest car il maximise le recall pour la classe Malade, réduisant ainsi le risque de passer à côté de patients à risque.

Cependant, dans notre cas, les données sont générées à des fins éducatives et formatives, pour créer un scénario « what if » permettant aux utilisateurs d'explorer les prédictions via un tableau de bord, donc nous choisissons d'utiliser le stacking.

---

7. Déploiement et Dashboard *What-If*

7.1 Export des Modèles

joblib.dump(stacking_clf, 'stacking_model.pkl')

7.2 Génération de Scénarios

Méthodologie :

- **133 056 scénarios** générés par combinaisons systématiques
- **Variables manipulables :**
    - Âge : 30-80 ans (pas de 10)
    - Cholestérol : 125-570 mg/dl (5 valeurs)
    - Fréquence cardiaque max : 80-200 bpm (10 valeurs)
    - Type douleur (cp) : 4 catégories
    - Thalassémie (thal) : 3 valeurs
    - Vaisseaux colorés (ca) : 4 valeurs
    - Dépression ST (oldpeak) : 0-6 (7 valeurs)

Variables fixées (valeurs par défaut) :

- Sexe : Homme (majoritaire dataset)
- Pression artérielle : Médiane training set
- Glycémie, ECG, Angine exercice : Valeurs normales

Pour pouvoir utiliser le tableau de bord sans ralentissements, nous devons renoncer à certaines variables afin de limiter la taille exponentielle de l'ensemble de données créé.

7.3 Catégorisation du Risque

Probabilité	Catégorie	Action Recommandée
< 30%	**Faible**	Suivi standard
30-70%	**Modéré**	Examens complémentaires
≥ 70%	**Élevé**	Intervention immédiate

7.4 Export Dashboard Power BI

Fichier généré : heart_disease_dashboard_stacking.csv

- **133 056 lignes × 13 colonnes**
- **Aucune valeur manquante**

Colonnes finales :

1. age, cp, chol, thalach, oldpeak, ca, thal (variables cliniques)
2. stacking_probability (pourcentage de risque)
3. stacking_risk (Faible/Modéré/Élevé)
4. type_douleur (label lisible pour cp)

Utilisation dans Power BI :

- Sliders interactifs pour paramètres cliniques
- Jauges de probabilité de maladie
- Filtres dynamiques par niveau de risque
- Graphiques d'évolution du risque

---

8. Conclusion

8.1 Résultats Clés

1. **Performance globale :** AUC = 0.953 (excellente discrimination)
2. **Régularisation réussie :** Réduction overfitting Random Forest (gap AUC : 5.1% → 3.5%)
3. **Ensemble efficace :** Stacking améliore légèrement les deux modèles de base
4. **Variables clés identifiées :** cp, thal, thalach, ca, oldpeak (62% importance)

8.2 Limitations

- **Taille dataset** : 303 observations limite la généralisation
- **Déséquilibre initial** : Transformation multi-classe → binaire nécessaire
- **Recall classe Malade** : 75-82% selon modèle (risque faux négatifs)
- **Contexte géographique** : Données provenant de 4 hôpitaux (biais possible)

8.3 Applications Pratiques

Pour les cliniciens :

- Outil d'aide à la décision pour dépistage préventif
- Identification des facteurs de risque modifiables (cholestérol, activité physique)
- Priorisation des type des patients nécessitant examens approfondis

Pour les patients :

- Visualisation impact des changements de mode de vie sur risque cardiovasculaire
- Sensibilisation aux facteurs de risque personnels

8.4 Perspectives d'Amélioration

1. **Dataset :** Augmenter taille échantillon (objectif > 1000 observations)
2. **Features :** Intégrer historique familial, habitudes vie (tabac, activité physique)
3. **Modèles :** Tester Deep Learning (si dataset suffisant = plus de 303 observations)
4. **Validation :** Étude prospective sur nouveaux patients

---

9. Bibliographie Technique

Librairies Python utilisées :

- scikit-learn 1.6 (modélisation, preprocessing, métriques)
- pandas 2.x (manipulation données)
- numpy 1.x (calculs numériques)
- matplotlib/seaborn (visualisations)
- joblib (sérialisation modèles)

Techniques appliquées :

- GridSearchCV pour optimisation hyperparamètres
- StratifiedKFold pour validation croisée équilibrée
- StackingClassifier pour modèle ensemble
- StandardScaler pour normalisation
- OneHotEncoder pour variables catégorielles

Métriques d'évaluation :

- AUC-ROC (discrimination globale)
- Precision/Recall (équilibre faux positifs/négatifs)
- F1-Score (moyenne harmonique precision/recall)
- Matrice de confusion (analyse détaillée erreurs)

Date : 9 Novembre 2025
Auteure : Giulia Governatori
Contact : giuliagovernatori@hotmail.com
