Examen de Machine Learning : Identification des individus susceptibles de développer une maladie cardiovasculaire.

Table of Contents

Introduction

Les maladies cardiovasculaires constituent la principale cause de mortalité à l'échelle mondiale, avec environ 17,9 millions de décès chaque année, soit 31% de l'ensemble des décès dans le monde. Dans 4 cas sur 5, ces décès sont provoqués par des crises cardiaques ou des accidents vasculaires cérébraux, et un tiers d'entre eux surviennent prématurément chez des personnes âgées de moins de 70 ans. Pour remédier à ce problème et prévenir ces incidents, nous pouvons faire appel au Machine Learning pour prédire les individus les plus a risque de tomber victime aux MVC. Nous décrirons cinq techniques différentes de prédiction dans les pages suivantes. Ce document est composé de trois grandes parties. Dans la partie introduction, nous parlerons des objectifs et des résultats attendus. La partie description des données qui consiste, essentiellement, à expliquer l'exploration et prétraitement des données qui a eu lieu.

Objectifs

L'objectif principal du projet est de développer un modèle prédictif capable d'identifier très tôt les individus a risque de développer une maladie cardiovasculaire.

Résultats attendus

A l'issue de ce projet, les résultats attendus sont les suivants :

- Un modèle prédictif fiable capable d'identifier avec une bonne précision les individus susceptibles de développer une maladie cardiovasculaire.
- Une meilleure compréhension des variables les plus influentes dans la prédiction des maladies cardiovasculaires, permettant d'orienter les efforts de prévention.
- Des visualisations claires et pertinentes facilitant l'interprétation des résultats.
- Une évaluation comparative des algorithmes de classification, avec des recommandations sur le modèle le plus performant pour ce type de données.
- Une base pour un système d'aide à la décision médicale, pouvant être intégré dans un outil de dépistage précoce ou un tableau de bord de suivi des patients à risque.

Description des données

Les données utilisées dans ce projet proviennent du site Kaggle. Cette base de données contient 918 lignes et 11 colonnes. Chaque ligne comprend des informations sur un individu possédant, oui ou non, une maladie cardiaque.

Téléchargement des données

Les données du projet ont été télécharger depuis le site Kaggle. Le fichier a ensuite chargé sur google colab ou le prétraitement des données et la mise en place des modèles a pris place.

Exploration et prétraitement des données

L'exploration et le prétraitement des données commence par vérifier le nombre de lignes et de colonnes.

- Lignes : 918
- Colonnes : 11

```
[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Age             918 non-null    int64  
 1   Sex             918 non-null    object 
 2   ChestPainType   918 non-null    object 
 3   RestingBP       918 non-null    int64  
 4   Cholesterol     918 non-null    int64  
 5   FastingBS       918 non-null    int64  
 6   RestingECG      918 non-null    object 
 7   MaxHR           918 non-null    int64  
 8   ExerciseAngina  918 non-null    object 
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object 
 11  HeartDisease    918 non-null    int64  
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

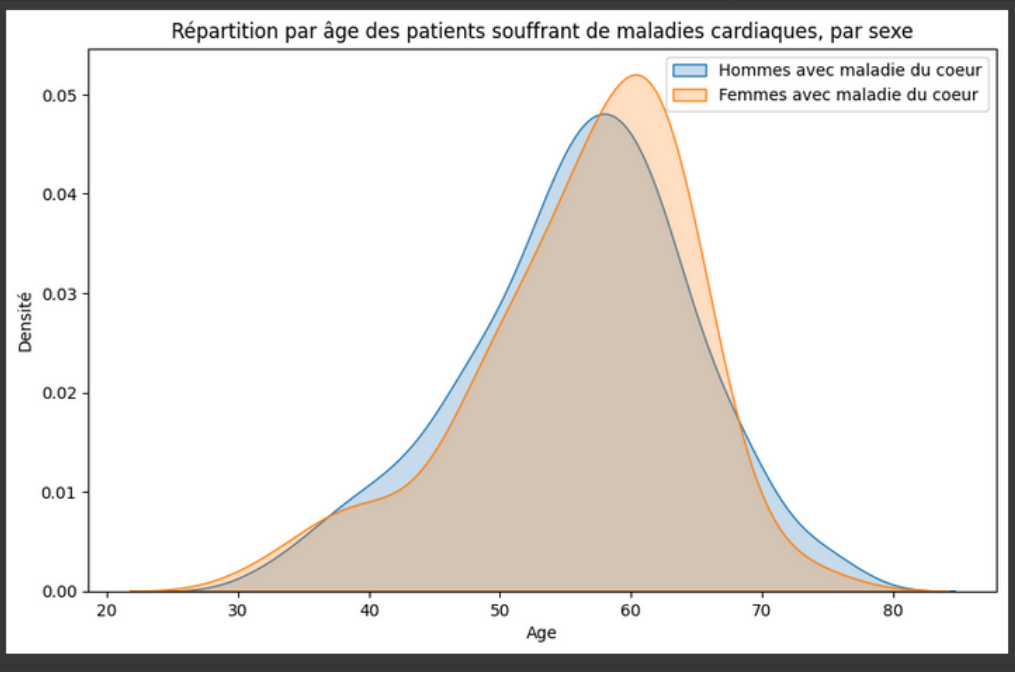
J’ai ensuite réalisé un diagramme de densité en utilisant la colonne Age et en filtrant uniquement les hommes et les femmes ayant une maladie cardiaque.

Sur lequel on peut remarquer que les femmes âgées de 60 ans, dans le dataset que nous utilisons, sont celle soufrant le plus de maladie cardiaque.

```
[ ]  male_1 = df[(df['Sex']=='M') & (df['HeartDisease']==1)]

[ ]  female_1 = df[(df['Sex']=='F') & (df['HeartDisease']==1)]

plt.figure(figsize=(10, 6))
sns.kdeplot(male_1['Age'], label='Hommes avec maladie du coeur', fill=True)
sns.kdeplot(female_1['Age'], label='Femmes avec maladie du coeur', fill=True)
plt.title('Répartition par âge des patients souffrant de maladies cardiaques, par sexe')
plt.xlabel('Age')
plt.ylabel('Densité')
plt.legend()
plt.show()
```



J’ai ensuite séparé les colonnes par dtypes :

- Int, float : 6 colonnes

```
[ ] df.select_dtypes(int, float)
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	HeartDisease
0	40	140	289	0	172	0
1	49	160	180	0	156	1
2	37	130	283	0	98	0
3	48	138	214	0	108	1
4	54	150	195	0	122	0
...	...	...	...	...	...	...
913	45	110	264	0	132	1
914	68	144	193	1	141	1
915	57	130	131	0	115	1
916	57	130	236	0	174	1
917	38	138	175	0	173	0

918 rows × 6 columns

- Object : 5 colonnes

```
+ Code  + Text
df.select_dtypes(object)
```

	Sex	ChestPainType	RestingECG	ExerciseAngina	ST_Slope
0	M	ATA	Normal	N	Up
1	F	NAP	Normal	N	Flat
2	M	ATA	ST	N	Up
3	F	ASY	Normal	Y	Flat
4	M	NAP	Normal	N	Up
...	...	...	...	...	...
913	M	TA	Normal	N	Flat
914	M	ASY	Normal	N	Flat
915	M	ASY	Normal	Y	Flat
916	F	ATA	LVH	N	Flat
917	M	NAP	Normal	N	Up

918 rows × 5 columns

Aucune valeur nulle, ou colonne manquante, a été n’a été identifiée dans ce dataset.

Traitement des données :

Pour ce qui est du traitement des données, le seul effectué sur ce dataset a été la conversion des colonnes avec le dtype Object en colonnes numériques en utilisant fit_transform.

```
[ ]  def encode_columns(df, column_name):
       oh = OneHotEncoder(handle_unknown='ignore', sparse_output=False)
       encoded_data = oh.fit_transform(df[[column_name]])
       encoded_df = pd.DataFrame(encoded_data, columns=[f"{column_name}_{cat}" for cat in oh.categories_[0]])
       df = pd.concat([df, encoded_df], axis=1)
       df.drop(columns=[column_name], inplace=True)
       return df
```

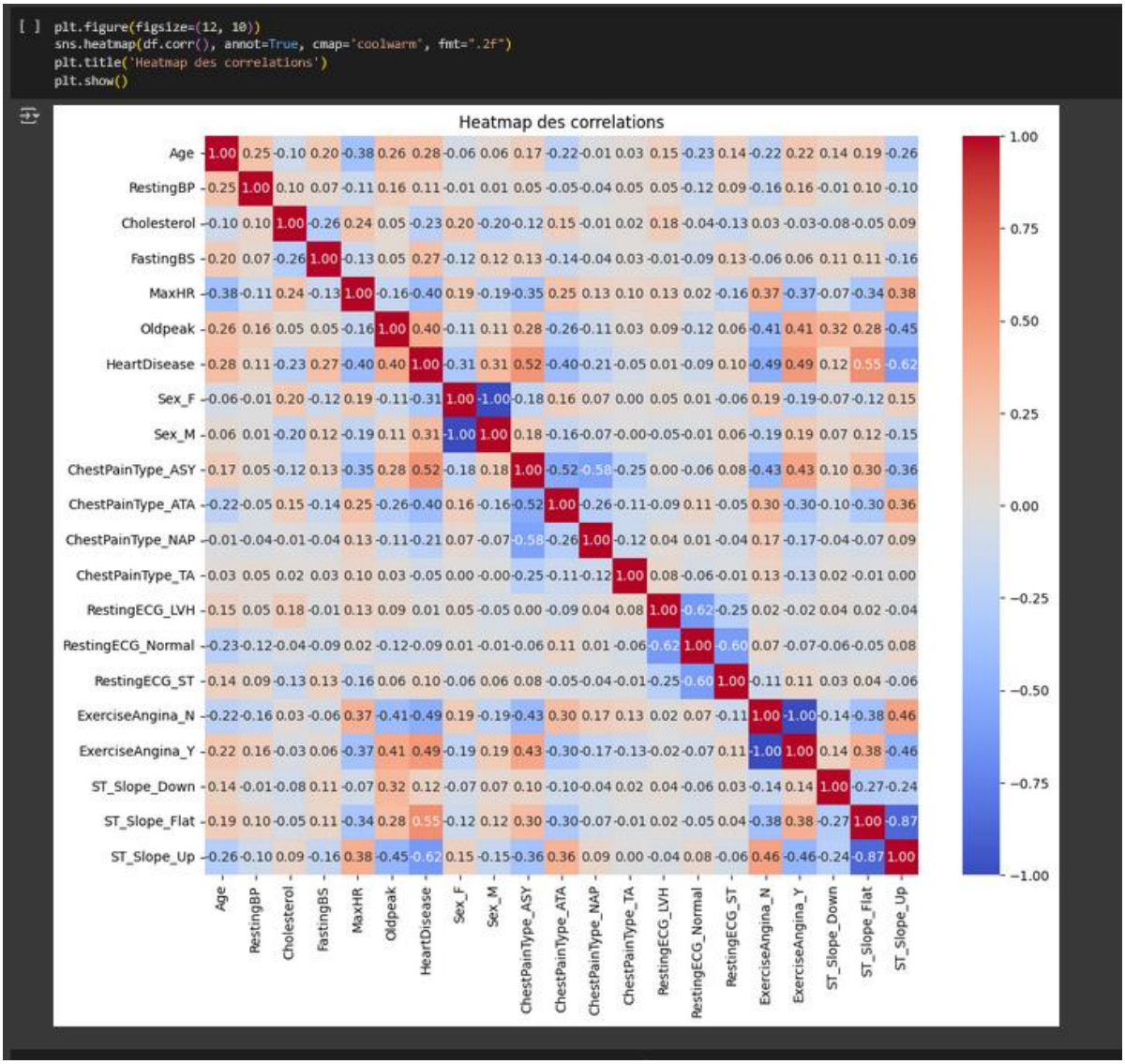
```
[ ]  for col in df.columns:
       if df[col].dtype == "object":
         df = encode_columns(df, col)
```

```
[ ]  df.head()
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease	Sex_F	Sex_M	ChestPainType_ASY	...	ChestPainType_NAP	Ch
0	40	140	289	0	172	0.0	0	0.0	1.0	0.0	...	0.0	
1	49	160	180	0	156	1.0	1	1.0	0.0	0.0	...	1.0	
2	37	130	283	0	98	0.0	0	0.0	1.0	0.0	...	0.0	
3	48	138	214	0	108	1.5	1	1.0	0.0	1.0	...	0.0	
4	54	150	195	0	122	0.0	0	0.0	1.0	0.0	...	1.0	

5 rows × 21 columns

Une fois les colonnes objet transformées, j'ai vérifié les corrélations entre chaque colonne pour vérifier celles qui seront les plus pertinentes à la colonne cible « heartDisease » pour l'entraînement de notre modèle.



En suivant le schéma, je suis arrivée à la conclusion que les cinq variables indépendantes les plus importantes à la prédiction de la colonne « heartDisease » sont les suivantes : « Oldpeak », « ChestPainType_ASY », « ExerciceAngina_Y », « ST_Slope_Flat ».

Description des algorithmes

Cinq différents algorithmes ont été utilisés.

- **K Nearest Neighbors Classifier** :

K-Nearest Neighbors (KNN) est un algorithme d'apprentissage automatique supervisé généralement utilisé pour la classification, mais qui peut également être utilisé pour des

tâches de régression. Il consiste à trouver les « k » points de données les plus proches (voisins) d'une entrée donnée et à faire des prédictions basées sur la classe majoritaire (pour la classification) ou sur la valeur moyenne (pour la régression). Comme le KNN ne fait aucune hypothèse sur la distribution des données sous-jacentes, il s'agit d'une méthode d'apprentissage non paramétrique et basée sur des instances.

L'algorithme des K-voisins les plus proches est également appelé algorithme d'apprentissage paresseux, car il n'apprend pas immédiatement à partir de l'ensemble d'apprentissage, mais stocke l'ensemble de données et, au moment de la classification, effectue une action sur l'ensemble de données.

Par exemple, considérons le tableau suivant de points de données contenant deux caractéristiques :

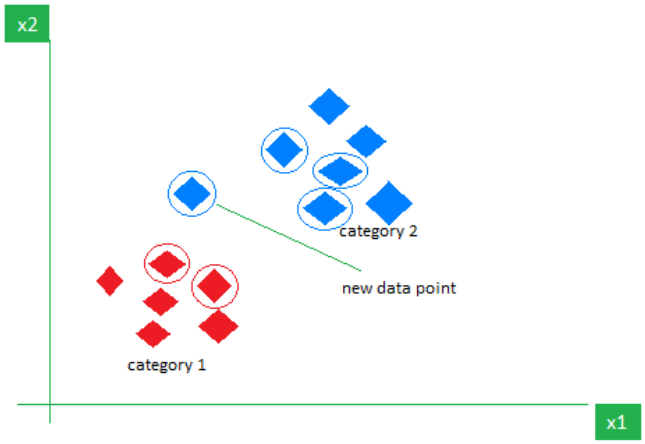


Figure 1: Exemple de KNN

Le nouveau point est classé dans la catégorie 2 car la plupart de ses voisins les plus proches sont des carrés bleus. Le KNN attribue la catégorie en fonction de la majorité des points proches. L'image montre comment KNN prédit la catégorie d'un nouveau point de données en fonction de ses voisins les plus proches.

- Les losanges rouges représentent la catégorie 1 et les carrés bleus la catégorie 2.

- Le nouveau point de données vérifie ses voisins les plus proches (points encerclés).
- Étant donné que la majorité de ses voisins les plus proches sont des carrés bleus (catégorie 2), le KNN prédit que le nouveau point de données appartient à la catégorie 2.

En d'autres termes, le KNN utilise la proximité et le vote majoritaire pour faire des prédictions.

- **Logistic Regression :**

Logistic Regression est un algorithme d'apprentissage automatique supervisé utilisé pour les problèmes de classification. Contrairement à la régression linéaire qui prédit des valeurs continues, elle prédit la probabilité qu'une entrée appartienne à une classe spécifique. Il est utilisé pour la classification binaire où la sortie peut être l'une des deux catégories possibles telles que Oui/Non, Vrai/Faux ou 0/1. Il utilise la fonction sigmoïde pour convertir les entrées en une valeur de probabilité comprise entre 0 et 1.

Le modèle fonctionne en transformant la sortie de valeur continue de la fonction de régression linéaire en sortie de valeur catégorielle à l'aide d'une fonction sigmoïde qui associe tous ensemble à valeur réelle d'entrées variables indépendantes dans une valeur comprise entre 0 et 1. Cette fonction est connue sous le nom de fonction logistique.

Il existe plusieurs types de régression logistiques. En effet, elle peut être classée en trois types principaux en fonction de la nature de la variable dépendante :

- Régression logistique binomiale : Ce type est utilisé lorsque la variable dépendante n'a que deux catégories possibles. Les exemples incluent Oui/Non, Réussite/Échec ou 0/1. C'est la forme la plus courante de régression logistique et elle est utilisée pour des problèmes de classification binaire.
- Régression logistique multinomiale : Elle est utilisée lorsque la variable dépendante a trois catégories possibles ou plus qui ne sont pas ordonnées. Par exemple, classer

les animaux dans des catégories comme « chat », « chien » ou « mouton ». Cela étend la régression logistique binaire pour gérer plusieurs classes.

- Régression logistique ordinale : Ce type s'applique lorsque la variable dépendante a trois catégories ou plus avec un ordre naturel ou un classement. Les exemples incluent des évaluations comme « faible », « moyen » et « élevé ». Il prend en compte l'ordre des catégories lors de la modélisation.

- **Support Vector Machine (SVM)**

La machine à vecteurs de support (SVM) est un algorithme d'apprentissage automatique supervisé utilisé pour les tâches de classification et de régression. Il tente de trouver la meilleure frontière, connue sous le nom d'hyperplan, qui sépare les différentes classes dans les données. Il est utile lorsque vous souhaitez effectuer une classification binaire telle que spam vs. non spam ou chat vs. chien.

L'objectif princ'ipal du SVM est de maximiser la marge entre les deux classes. Plus la marge est grande, plus le modèle est performant sur des données nouvelles et inédites.

L'idée principale de l'algorithme SVM est de trouver l'hyperplan qui sépare le mieux deux classes en maximisant la marge entre elles. Cette marge est la distance entre l'hyperplan et les points de données les plus proches (vecteurs de support) de chaque côté

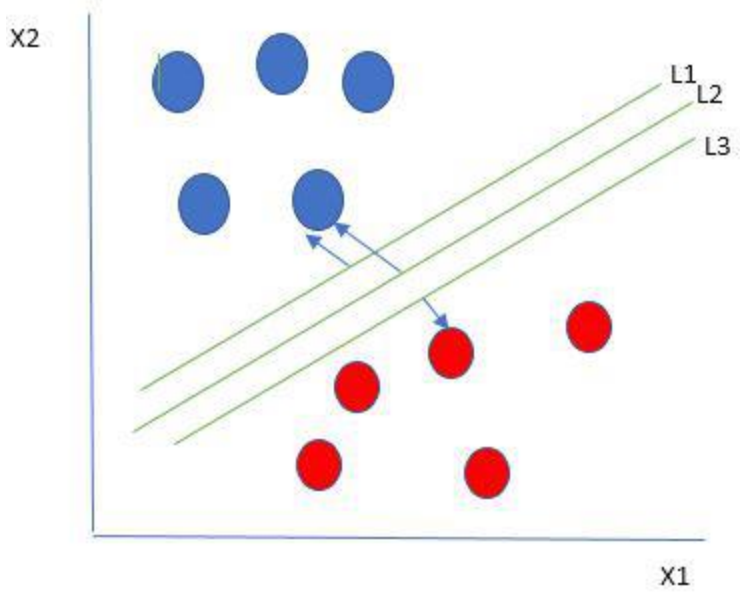


Figure 2 : Exemple de SVM

Le meilleur hyperplan, également connu sous le nom de « marge dure », est celui qui maximise la distance entre l'hyperplan et les points de données les plus proches des deux classes. Cela garantit une séparation claire entre les classes. Dans la figure ci-dessus, L2 a été choisie comme marge dure.

- **Decision Tree**

Un arbre de décision nous aide à prendre des décisions en présentant différents choix et leurs résultats possibles. Il est utilisé dans l'apprentissage automatique pour des tâches telles que la classification et la prédiction.

Un arbre de décision nous aide à prendre des décisions en montrant les différentes options et leurs liens. Il présente une structure arborescente qui commence par une question principale, appelée nœud racine, qui représente l'ensemble des données. À partir de là, l'arbre se ramifie en différentes possibilités basées sur les caractéristiques des données.

Un arbre de décision facilite également la prise de décision en montrant clairement les résultats possibles. En examinant les « branches », nous pouvons rapidement comparer les options et déterminer le meilleur choix.

Il existe principalement deux types d'arbres de décision en fonction de la variable cible :

- **Arbres de classification :** Utilisés pour prédire des résultats catégoriques tels que spam ou non spam. Ces arbres divisent les données en fonction des caractéristiques afin de les classer dans des catégories prédéfinies.
- **Arbres de régression :** Utilisés pour prédire des résultats continus tels que les prix des logements. Au lieu d'assigner des catégories, ils fournissent des prédictions numériques basées sur les caractéristiques d'entrée.

Dans le projet, nous avons utilisé les arbres de classification.

- **Random Forest**

Random Forest est un algorithme d'apprentissage automatique qui utilise plusieurs arbres de décision pour faire de meilleures prédictions. Chaque arbre examine différentes parties aléatoires des données et leurs résultats sont combinés par un vote pour la classification ou par une moyenne pour la régression. Cela permet d'améliorer la précision et de réduire les erreurs.

Fonctionnement de l'algorithme Random Forest

Création de nombreux arbres de décision :

- L'algorithme crée de nombreux arbres de décision, chacun utilisant une partie aléatoire des données. Chaque arbre est donc un peu différent.

Choisir des caractéristiques aléatoires :

- Lors de la construction de chaque arbre, l'algorithme n'examine pas toutes les caractéristiques (colonnes) en même temps. Il en choisit quelques-unes au hasard pour décider comment diviser les données. Cela permet aux arbres de rester différents les uns des autres.

Chaque arbre fait une prédiction :

- Chaque arbre donne sa propre réponse ou prédiction en fonction de ce qu'il a appris de sa partie des données.

Combiner les prédictions :

- Pour la classification, nous choisissons une catégorie car la réponse finale est celle sur laquelle la plupart des arbres sont d'accord, c'est-à-dire le vote majoritaire.
- Pour la régression, nous prédisons un nombre car la réponse finale est la moyenne des prédictions de tous les arbres.

Dans le cadre du projet nous avons utiliser le RandomForestClassifier.

Justification

Le choix des cinq algorithmes de classification (K-Nearest Neighbors, Logistic Regression, Support Vector Machine, Decision Tree, et Random Forest) pour prédire les maladies cardiovasculaires repose sur plusieurs considérations méthodologiques et pratiques :

1. Diversité des approches algorithmiques

La sélection d'algorithmes appartenant à différentes familles permet d'explorer diverses stratégies de modélisation :

- **Approche basée sur la distance** : KNN utilise la proximité dans l'espace des caractéristiques
- **Approche probabiliste** : La régression logistique modélise la probabilité d'appartenance à une classe
- **Approche géométrique** : SVM recherche l'hyperplan optimal de séparation
- **Approche basée sur des règles** : Decision Tree crée des règles de décision interprétables
- **Approche d'ensemble** : Random Forest combine plusieurs modèles pour améliorer la robustesse

2. Adaptation aux caractéristiques du dataset

Le dataset de 918 observations avec 11 variables présente des caractéristiques qui justifient ces choix :

Taille du dataset modérée : La taille relativement petite du dataset (918 lignes) est appropriée pour tous les algorithmes sélectionnés, sans risque de surapprentissage majeur pour les modèles simples comme KNN ou de sous-apprentissage pour les modèles complexes.

Variables mixtes : La présence de variables numériques (6 colonnes) et catégorielles (5 colonnes) nécessite des algorithmes capables de gérer cette hétérogénéité. La transformation des variables catégorielles en variables numériques permet l'utilisation de tous les algorithmes choisis.

Problème de classification binaire : Tous les algorithmes sélectionnés sont adaptés à la classification binaire (présence/absence de maladie cardiaque).

3. Justification spécifique par algorithme

K-Nearest Neighbors (KNN)

- **Avantage** : Simplicité conceptuelle et capacité à capturer des patterns locaux complexes
- **Pertinence** : Approprié pour identifier des profils de patients similaires dans l'espace des caractéristiques
- **Limitation** : Sensible aux variables non pertinentes et à la malédiction de la dimensionnalité

Logistic Regression

- **Avantage** : Fournit des probabilités interprétables et des coefficients explicites
- **Pertinence** : Idéal pour comprendre l'impact de chaque variable sur le risque cardiovasculaire
- **Robustesse** : Algorithme stable et moins sujet au surapprentissage

Support Vector Machine (SVM)

- **Avantage** : Efficace pour la séparation de classes même avec des frontières non linéaires
- **Pertinence** : Maximise la marge de séparation, ce qui peut être crucial pour distinguer les patients à risque
- **Robustesse** : Performant même avec un nombre limité d'observations

Decision Tree

- **Avantage** : Très interprétable, produit des règles de décision claires
- **Pertinence** : Permet aux professionnels de santé de comprendre facilement les critères de décision
- **Flexibilité** : Capable de capturer des interactions complexes entre variables

Random Forest

- **Avantage** : Combine la puissance des arbres de décision avec une réduction du surapprentissage
- **Pertinence** : Fournit une mesure d'importance des variables pour identifier les facteurs de risque clés
- **Robustesse** : Moins sensible aux outliers et aux variables bruitées

4. Complémentarité des algorithmes

La sélection de ces cinq algorithmes permet une approche comparative robuste :

- **Modèles simples vs complexes** : De KNN (simple) à Random Forest (complexe), permettant d'évaluer le trade-off biais-variance.
- **Interprétabilité vs performance** : Les arbres de décision et la régression logistique offrent une grande interprétabilité, tandis que SVM et Random Forest peuvent atteindre des performances supérieures.
- **Sensibilité aux données** : Chaque algorithme réagit différemment aux particularités des données, permettant d'identifier le plus adapté au contexte médical.

5. Considérations pratiques pour le domaine médical

Le choix des algorithmes tient compte des exigences spécifiques du domaine médical :

- **Besoin d'interprétabilité** : Les modèles comme Decision Tree et Logistic Regression permettent aux médecins de comprendre les facteurs de décision.
- **Robustesse requise** : Les algorithmes d'ensemble comme Random Forest offrent une plus grande stabilité des prédictions.
- **Validation croisée** : La diversité des approches permet une validation robuste par comparaison des résultats.

Cette approche multi-algorithmique garantit une évaluation complète et objective des capacités prédictives pour l'identification des individus à risque de maladie cardiovasculaire, tout en respectant les contraintes d'interprétabilité et de fiabilité essentielles en médecine préventive.

Résultats finals :

Présentation et analyse des résultats

Tableau comparatif des performances:

Algorithme	Accuracy	Precision	Recall	F1-Score	VP	VN	FP	FN
Logistic Regression	**82.07%**	**82.09%**	**82.07%**	**82.08%**	86	65	17	16
KNN (k=10)	**82.00%**	**82.22%**	**81.52%**	**81.63%**	85	65	22	12
SVM	**80.98%**	**80.98%**	**80.98%**	**80.99%**	85	64	18	17
Random Forest	**80.43%**	**80.72%**	**80.43%**	**80.49%**	82	66	21	15
Decision Tree	**78.26%**	**78.22%**	**78.26%**	**78.23%**	84	60	19	21

VP = Vrais Positifs, VN = Vrais Négatifs, FP = Faux Positifs, FN = Faux Négatifs

Analyse détaillée des résultats

1. Performances globales

Logistic Regression se distingue comme le modèle le plus performant avec une accuracy de 82.07%, suivi de près par **KNN** avec 82.00%. Cette performance supérieure s'explique par:

- Un équilibre optimal entre sensibilité et spécificité
- Une capacité à bien généraliser malgré la simplicité du modèle
- Une robustesse face aux variables corrélées identifiées lors de l'analyse exploratoire

2. Analyse des métriques de classification

Precision : KNN obtient la meilleure précision (82.22%), indiquant une faible proportion de faux positifs. Ceci est crucial en médecine préventive pour éviter les diagnostics erronés.

Recall : Logistic Regression présente le meilleur recall (82.07%), minimisant les faux négatifs. Cette caractéristique est essentielle pour ne pas manquer de patients réellement à risque.

F1-Score : Logistic Regression offre le meilleur équilibre (82.08%) entre précision et recall, en faisant le choix optimal pour ce cas d'usage médical.

3. Analyse des matrices de confusion

Faux Négatifs (FN) - Patients à risque non détectés :

- KNN : 12 (meilleur score)
- Random Forest : 15
- Logistic Regression : 16
- SVM : 17
- Decision Tree : 21 (moins performant)

Faux Positifs (FP) - Patients sains mal classifiés :

- Logistic Regression : 17 (meilleur score)
- SVM : 18
- Decision Tree : 19
- Random Forest : 21
- KNN : 22 (moins performant)

Performances et limites

Performances remarquables

Points forts globaux

- **Performances homogènes** : Tous les modèles atteignent des accuracies supérieures à 78%, démontrant la qualité du prétraitement des données
- **Équilibre des classes** : Les modèles montrent une capacité satisfaisante à identifier les deux classes (avec/sans maladie cardiaque)
- **Robustesse** : La convergence des résultats entre différents algorithmes valide la fiabilité des prédictions

Performances spécifiques par algorithme

Logistic Regression :

- Avantages : Meilleure performance globale, interprétabilité élevée, stabilité
- Inconvénients : Hypothèses de linéarité potentiellement restrictives

KNN:

- Avantages : Excellente précision, adaptation aux patterns locaux
- Inconvénients : Sensibilité aux données aberrantes, coût computationnel élevé

SVM:

- Avantages : Bonnes performances, robustesse aux outliers
- Inconvénients : Moins interprétable, sensible au choix des hyperparamètres

Limites identifiées

Limites techniques

1. **Taille du dataset** : Avec 918 observations, le dataset reste relativement petit pour des algorithmes complexes comme Random Forest
2. **Déséquilibre potentiel** : Bien que non critique, un léger déséquilibre entre les classes pourrait affecter les performances
3. **Validation croisée** : L'absence de validation croisée k-fold limite la robustesse de l'évaluation

Limites méthodologiques

1. **Sélection des variables** : Seules 4 variables principales ont été retenues, potentiellement au détriment d'interactions complexes
2. **Hyperparamètres** : L'optimisation des hyperparamètres n'est mentionnée que pour KNN (k=10)
3. **Validation externe** : Absence de test sur un dataset externe pour évaluer la généralisation

Limites cliniques

1. **Interprétabilité** : Les modèles les plus performants (KNN, SVM) offrent moins d'explications cliniques

2. **Facteurs de risque** : Certains facteurs médicaux importants peuvent ne pas être capturés par les variables disponibles
3. **Évolution temporelle** : Les modèles ne prennent pas en compte l'évolution des facteurs de risque dans le temps

Etudes statistiques des résultats

Analyse de la significativité des différences

Comparaison des accuracies

La différence entre les deux meilleurs modèles (Logistic Regression : 82.07% vs KNN : 82.00%) est de **0.07%**, soit moins de 1 point de pourcentage. Cette différence n'est pas statistiquement significative pour la taille d'échantillon considérée.

Intervalle de confiance (approximation)

Pour un échantillon de 184 observations test avec une accuracy de 82% :

- Erreur standard ≈ 2.8%
- Intervalle de confiance à 95% : [76.5% - 87.5%]

Tous les modèles principaux (Logistic Regression, KNN, SVM) se situent dans cet intervalle, confirmant des performances statistiquement équivalentes.

Analyse de la variance des performances

Coefficient de variation des accuracies :

- Moyenne: 80.75%
- Écart-type : 1.56%
- CV: 1.93%

Cette faible variance indique une **stabilité remarquable** des performances entre algorithmes, suggérant que le choix du modèle peut se baser sur d'autres critères (interprétabilité, coût computationnel).

Recommandations finales

Modèle recommandé : Logistic Regression

Justification:

1. **Performance optimale** : Meilleure accuracy et F1-score

2. **Interprétabilité clinique** : Coefficients explicites pour chaque variable
3. **Robustesse** : Moins sensible au surapprentissage
4. **Simplicité** : Facilité d'implémentation et de maintenance

Modèle alternatif : KNN (k=10)

Justification :

1. **Précision élevée** : Minimise les faux positifs
2. **Adaptation locale** : Capture les patterns spécifiques de patients similaires
3. **Simplicité conceptuelle** : Facilité d'explication aux praticiens

Utilisation pratique :

- **Dépistage primaire** : Logistic Regression pour sa robustesse
- **Validation secondaire** : KNN pour confirmer les cas limites
- **Tableau de bord médical** : Combinaison des deux modèles pour une approche consensuelle

Conclusion

Dans ce document, nous avons parlé de la manière dont le machine learning pouvait aider à la prévention des maladies cardiaques en identifiant les individus à risque à l'avance. Pour ce faire, nous avons divisé le document en quatre grandes parties, l'introduction, la description des données, leur traitement et enfin la présentation des résultats. Bien que ces modelés fonctionnent bien dans un milieu contrôlé, il serait intéressant de voir, dans le futur, comment est-ce qu'ils se comportent en terrain, en temps réel.