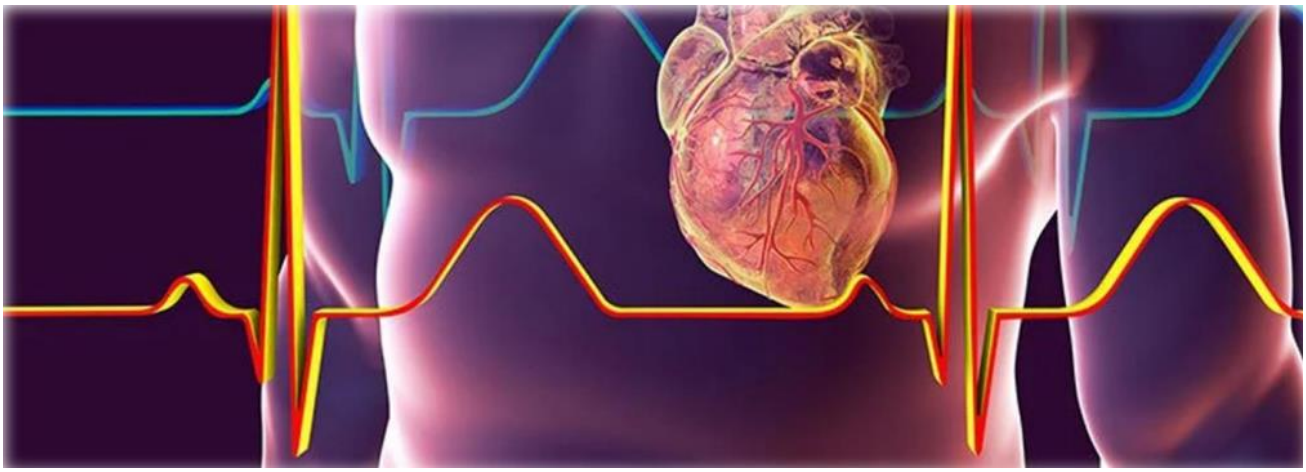
Faculté des Sciences et Techniques
Département d'Informatique
MST : SIDI
Module : Machine learning & Data mining
A.U : 2023-2024

Mini projet

DÉTECTION DES MALADIES CARDIAQUES (HEART DISEASE DETECTION)



Réalisé par :
Mounia ALAOUI

Encadré par :
M. Mohamed SABIRI

Table des matières

Introduction :

Dans un monde où les maladies cardiaques demeurent une préoccupation majeure de santé publique, l'exploitation judicieuse des avancées technologiques s'avère cruciale. Ce rapport présente une exploration approfondie du domaine de la détection des maladies cardiaques à travers l'application des techniques de data mining. En se concentrant sur un ensemble de données variées et riches en informations, le projet vise à développer un modèle prédictif efficace. De la définition du problème à l'interprétation des résultats, chaque étape du processus sera minutieusement détaillée. L'objectif ultime est de contribuer à l'amélioration des stratégies de prévention en fournissant un outil de détection précoce robuste, ouvrant ainsi la voie à des avancées significatives dans le domaine de la santé cardiovasculaire.

1. Définition du Problème :

La détection précoce des maladies cardiaques est un défi crucial en santé publique en raison de leur prévalence mondiale et de leurs conséquences graves. Ces affections figurent parmi les principales causes de morbidité et de mortalité. La section explore en détail les caractéristiques des maladies cardiaques, mettant en évidence les facteurs de risque essentiels tels que l'hypertension, le taux de cholestérol élevé, le tabagisme et le diabète. L'analyse des modèles épidémiologiques actuels souligne l'ampleur du problème et identifie les populations les plus vulnérables. La compréhension de l'importance de la détection précoce devient impérative pour développer des stratégies préventives efficaces et des approches personnalisées de prise en charge médicale, établissant ainsi une base solide pour le reste du projet.

2. Outils Utilisés :

Dans le cadre de ce projet de Détection des Maladies Cardiaques, plusieurs outils ont été sélectionnés pour la collecte, le nettoyage, l'analyse des données, ainsi que pour la construction et l'évaluation des modèles prédictifs. Ces outils ont été choisis en fonction de leur adaptabilité aux tâches spécifiques de l'étude et de leur réputation dans le domaine de la science des données et de l'apprentissage automatique. Les principaux outils utilisés sont les suivants :

Langage :

Python est un langage de script de haut niveau, structuré et open source. Il est multi-usage. Il est un langage de programmation très puissant utilisé en Data Mining pour faire de l'analyse statistique, la classification, le clustering et l'analyse prédictive.

- **Bibliothèques :**

- **Pandas :** Pour la manipulation et l'analyse des données.
- **NumPy :** Pour les opérations mathématiques et la manipulation des tableaux.
- **Matplotlib et Seaborn :** Pour la visualisation des données.
- **Scikit-learn :** Pour la mise en œuvre des algorithmes d'apprentissage automatique.

3. Collecte des Données :

La collecte des données constitue la première étape cruciale de mon projet de **Détection des Maladies Cardiaques**. Les données utilisées sont extraites du dataset "**Heart Disease Health Indicators**" accessible sur Kaggle via le lien https://www.kaggle.com/alexteboul/heart-disease-health-indicators-dataset.

Ce dataset provient d'une source médicale réputée et a été créé dans le cadre d'études épidémiologiques visant à comprendre les facteurs de risque liés aux maladies cardiaques.

- **Description du Dataset :** La dataset semble contenir des informations sur la santé et le mode de vie des individus. Chaque ligne représente un individu, et chaque colonne correspond à une caractéristique spécifique. Voici une interprétation des colonnes principales :

HeartDiseaseorAttack : Présence (1) ou absence (0) de maladie cardiaque ou d'attaque cardiaque, **(La variable cible).**
HighBP : Hypertension (1) ou absence d'hypertension (0).
HighChol : Cholestérol élevé (1) ou cholestérol normal (0).
CholCheck : Fréquence de vérification du cholestérol.
BMI : Indice de masse corporelle (IMC).
Smoker : Fumeur (1) ou non-fumeur (0).
Stroke : Accident vasculaire cérébral (1) ou absence d'AVC (0).
Diabetes : Diabète (1) ou absence de diabète (0).
PhysActivity : Niveau d'activité physique.
Fruits : Fréquence de consommation de fruits.
Veggies : Fréquence de consommation de légumes.
HvyAlcoholConsump : Consommation excessive d'alcool (1) ou non (0).
AnyHealthcare : Accès aux soins de santé.
NoDocbcCost : Pas de coût médical (1) ou coût médical (0).
GenHlth : État de santé général.
MentHlth : Santé mentale.
PhysHlth : Santé physique.
DiffWalk : Difficulté à marcher.

Sex : Sexe (0 pour homme, 1 pour femme).
Age : Âge de l'individu.
Education : Niveau d'éducation.
Income : Revenu.

Quelques informations sur le dataset :

```
# Import des bibliothèques nécessaires
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Chargement du dataset à partir du fichier CSV
file_path = '/content/heart_disease_health_indicators_BRFSS2015.csv'
data = pd.read_csv(file_path)
# Obtenir la forme du tableau
data.shape
```

```
(35096, 22)
```

```
# Affichage des premières lignes du dataset pour vérification
print(data.head())
```

```
   HeartDiseaseorAttack  HighBP  HighChol  CholCheck   BMI  Smoker  Stroke  \
0                   0.0     1.0       1.0        1.0  40.0     1.0     0.0
1                   0.0     0.0       0.0        0.0  25.0     1.0     0.0
2                   0.0     1.0       1.0        1.0  28.0     0.0     0.0
3                   0.0     1.0       0.0        1.0  27.0     0.0     0.0
4                   0.0     1.0       1.0        1.0  24.0     0.0     0.0

   Diabetes  PhysActivity  Fruits  ...  AnyHealthcare  NoDocbcCost  GenHlth  \
0       0.0           0.0     0.0  ...            1.0          0.0      5.0
1       0.0           1.0     0.0  ...            0.0          1.0      3.0
2       0.0           0.0     1.0  ...            1.0          1.0      5.0
3       0.0           1.0     1.0  ...            1.0          0.0      2.0
4       0.0           1.0     1.0  ...            1.0          0.0      2.0

   MentHlth  PhysHlth  DiffWalk  Sex   Age  Education  Income
0      18.0      15.0       1.0  0.0   9.0        4.0     3.0
1       0.0       0.0       0.0  0.0   7.0        6.0     1.0
2      30.0      30.0       1.0  0.0   9.0        4.0     8.0
3       0.0       0.0       0.0  0.0  11.0        3.0     6.0
4       3.0       0.0       0.0  0.0  11.0        5.0     4.0

[5 rows x 22 columns]
```

```
# afficher les 5 derniers lignes
data.tail()
```

	HeartDiseaseorAttack	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	Diabetes	PhysActivity	Fruits	...	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	PhysHlth	DiffWalk	Sex	Age	Education	Income
93574	0.0	0.0	0.0	1.0	25.0	1.0	0.0	0.0	1.0	0.0	...	1.0	0.0	2.0	0.0	0.0	0.0	0.0	10.0	6.0	1.0
93575	0.0	0.0	0.0	1.0	22.0	0.0	0.0	0.0	1.0	0.0	...	1.0	0.0	1.0	0.0	0.0	0.0	1.0	5.0	6.0	8.0
93576	0.0	0.0	0.0	1.0	27.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	1.0	0.0	0.0	0.0	0.0	4.0	6.0	8.0
93577	1.0	1.0	1.0	1.0	23.0	0.0	0.0	2.0	1.0	1.0	...	1.0	0.0	4.0	0.0	0.0	1.0	0.0	10.0	4.0	3.0
93578	0.0	0.0	0.0	1.0	28.0	0.0	0.0	0.0	1.0	1.0	...	1.0	0.0	2.0	0.0	2.0	0.0	1.0	NaN	NaN	NaN

4. Nettoyage et Transformation des Données :

Le processus de nettoyage et de transformation des données revêt une importance cruciale dans la préparation du dataset pour l'analyse. Dans cette étape, j'ai effectué les opérations suivantes :

- **Gestion des Valeurs Manquantes :** J'ai identifié et traité toute valeur manquante dans le dataset. Selon la nature de la donnée manquante, j'ai opté pour des stratégies telles que l'imputation de la moyenne ou de la médiane pour les variables continues, et l'utilisation de modes pour les variables catégorielles.

```
# 1. Vérification des valeurs manquantes
missing_values = data.isnull().sum()
print("\nValeurs manquantes par colonne :")
print(missing_values)
```

```
Valeurs manquantes par colonne :
HeartDiseaseorAttack    0
HighBP                  0
HighChol                0
CholCheck               0
BMI                     0
Smoker                  0
Stroke                  0
Diabetes                0
PhysActivity            0
Fruits                  0
Veggies                 1
HvyAlcoholConsump       1
AnyHealthcare           1
NoDocbcCost             1
GenHlth                 1
MentHlth                1
PhysHlth                1
DiffWalk                1
Sex                     1
Age                     1
Education               1
Income                  1
dtype: int64
```

```
# 2. Suppression des lignes avec des valeurs manquantes
data = data.dropna()
```

```
data.isnull().sum()
```

```
HeartDiseaseorAttack    0
HighBP                  0
HighChol                0
CholCheck               0
BMI                     0
Smoker                  0
Stroke                  0
Diabetes                0
PhysActivity            0
Fruits                  0
Veggies                 0
HvyAlcoholConsump       0
AnyHealthcare           0
NoDocbcCost             0
GenHlth                 0
MentHlth                0
PhysHlth                0
DiffWalk                0
Sex                     0
Age                     0
Education               0
Income                  0
dtype: int64
```

- **Encodage des Variables Catégorielles :** Les variables catégorielles ont été encodées pour les rendre compatibles avec les algorithmes d'apprentissage automatique. J'ai utilisé une méthode d'encodage adaptée à la nature des variables, telles que l'encodage binaire pour les variables binaires.

```
# 3. Transformation des variables catégorielles en variables indicatrices
# Exemple : Encodage one-hot des colonnes catégorielles
data_encoded = pd.get_dummies(data, columns=['Sex', 'Education'])
```

- **Normalisation des Variables :** Cette étape est cruciale pour éviter que des écarts d'échelle n'influencent de manière disproportionnée certains algorithmes d'apprentissage automatique.

```
from sklearn.preprocessing import MinMaxScaler
# 4. Normalisation ou standardisation des données
# Exemple : Min-max scaling des colonnes numériques
scaler = MinMaxScaler()
columns_to_scale = ['Age', 'Income', 'BMI']
data_encoded[columns_to_scale] = scaler.fit_transform(data_encoded[columns_to_scale])
```

```
# Affichage des premières lignes du dataset après nettoyage et transformation
print("\nAprès le nettoyage et la transformation :")
print(data_encoded.head())
```

```
Après le nettoyage et la transformation :
   HeartDiseaseorAttack  HighBP  HighChol  CholCheck       BMI  Smoker  \
0                   0.0     1.0       1.0        1.0  0.325581     1.0
1                   0.0     0.0       0.0        0.0  0.151163     1.0
2                   0.0     1.0       1.0        1.0  0.186047     0.0
3                   0.0     1.0       0.0        1.0  0.174419     0.0
4                   0.0     1.0       1.0        1.0  0.139535     0.0

   Stroke  Diabetes  PhysActivity  Fruits  ...       Age    Income  Sex_0.0  \
0     0.0       0.0           0.0     0.0  ...  0.666667  0.285714        1
1     0.0       0.0           1.0     0.0  ...  0.500000  0.000000        1
2     0.0       0.0           0.0     1.0  ...  0.666667  1.000000        1
3     0.0       0.0           1.0     1.0  ...  0.833333  0.714286        1
4     0.0       0.0           1.0     1.0  ...  0.833333  0.428571        1

   Sex_1.0  Education_1.0  Education_2.0  Education_3.0  Education_4.0  \
0        0              0              0              0              1
1        0              0              0              0              0
2        0              0              0              0              1
3        0              0              0              1              0
4        0              0              0              0              0
```

```
   Education_5.0  Education_6.0
0              0              0
1              0              1
2              0              0
3              0              0
4              1              0

[5 rows x 28 columns]
```

Compter le nombre d'occurrences de chaque valeur unique dans la colonne '**HeartDiseaseorAttack**'

```
data['HeartDiseaseorAttack'].value_counts()
```

```
0.0    84801
1.0     8777
Name: HeartDiseaseorAttack, dtype: int64
```

5. Exploration des Données :

Dans cette phase cruciale du projet, j'ai entrepris une exploration détaillée des données pour mieux comprendre la nature du dataset et identifier d'éventuelles tendances ou corrélations.

- **Statistiques descriptives :** Des statistiques descriptives ont été calculées pour les variables du dataset, fournissant des mesures récapitulatives telles que la moyenne, l'écart type, la médiane, et les quartiles.

```python
# 1. Statistiques descriptives
descriptive_stats = data.describe()
print("\nStatistiques descriptives :")
print(descriptive_stats)
```

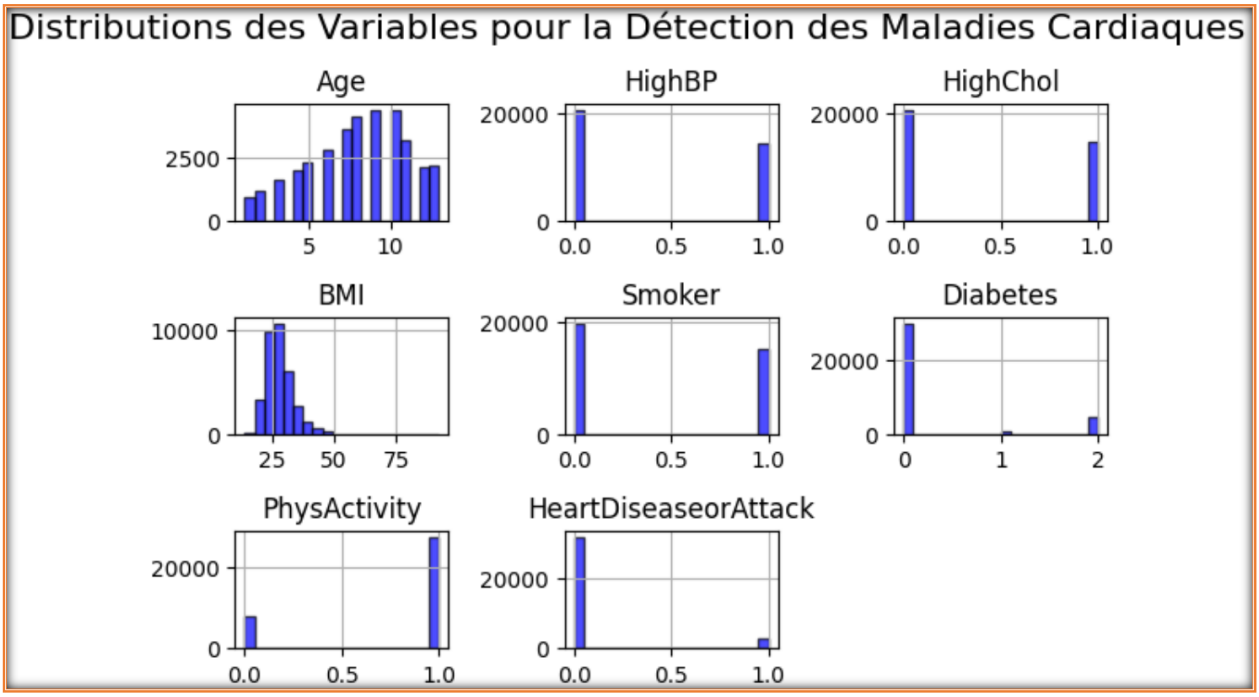
```
Statistiques descriptives :
       HeartDiseaseorAttack        HighBP      HighChol     CholCheck  \
count          35095.000000  35095.000000  35095.000000  35095.000000
mean               0.086366      0.410258      0.412965      0.963129
std                0.280907      0.491887      0.492374      0.188449
min                0.000000      0.000000      0.000000      0.000000
25%                0.000000      0.000000      0.000000      1.000000
50%                0.000000      0.000000      0.000000      1.000000
75%                0.000000      1.000000      1.000000      1.000000
max                1.000000      1.000000      1.000000      1.000000

                BMI        Smoker        Stroke      Diabetes  PhysActivity  \
count  35095.000000  35095.000000  35095.000000  35095.000000  35095.000000
mean      27.919732      0.436643      0.039094      0.286935      0.780140
std        6.104906      0.495977      0.193821      0.686446      0.414158
min       14.000000      0.000000      0.000000      0.000000      0.000000
25%       24.000000      0.000000      0.000000      0.000000      1.000000
50%       27.000000      0.000000      0.000000      0.000000      1.000000
75%       31.000000      1.000000      0.000000      0.000000      1.000000
max       92.000000      1.000000      1.000000      2.000000      1.000000
```

```
             Fruits  ...  AnyHealthcare   NoDocbcCost       GenHlth  \
count  35095.000000  ...   35095.000000  35095.000000  35095.000000
mean       0.644850  ...       0.946716      0.091608      2.487961
std        0.478566  ...       0.224602      0.288477      1.086563
min        0.000000  ...       0.000000      0.000000      1.000000
25%        0.000000  ...       1.000000      0.000000      2.000000
50%        1.000000  ...       1.000000      0.000000      2.000000
75%        1.000000  ...       1.000000      0.000000      3.000000
max        1.000000  ...       1.000000      1.000000      5.000000

           MentHlth      PhysHlth      DiffWalk           Sex           Age  \
count  35095.000000  35095.000000  35095.000000  35095.000000  35095.000000
mean       3.312723      4.262260      0.161305      0.442969      7.885083
std        7.433677      8.672821      0.367817      0.496744      3.098226
min        0.000000      0.000000      0.000000      0.000000      1.000000
25%        0.000000      0.000000      0.000000      0.000000      6.000000
50%        0.000000      0.000000      0.000000      0.000000      8.000000
75%        2.000000      3.000000      0.000000      1.000000     10.000000
max       30.000000     30.000000      1.000000      1.000000     13.000000

          Education        Income
count  35095.000000  35095.000000
mean       5.082262      6.114147
std        1.018234      2.105097
min        1.000000      1.000000
25%        4.000000      5.000000
50%        5.000000      7.000000
75%        6.000000      8.000000
max        6.000000      8.000000
```

- **Visualisation des Distributions :** Des histogrammes ont été générés pour les variables significatives telles que '**Age**', '**Income**' et '**BMI**'... Ces graphiques permettent d'observer la répartition des données et d'identifier des schémas potentiels. La figure ci-dessous illustre graphiquement ces distributions.

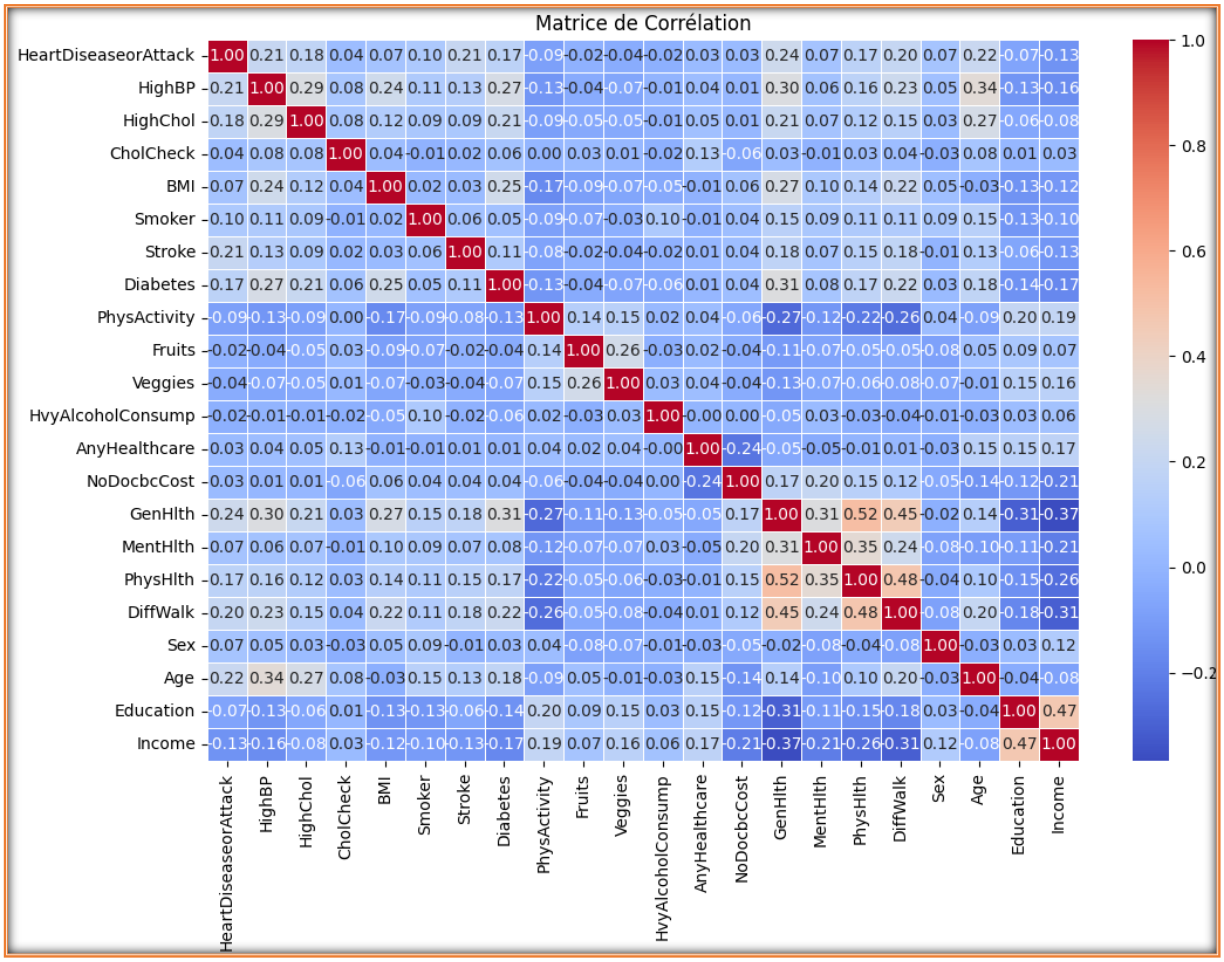
```
# 3. Visualisation des distributions des variables importantes
# Variables pertinentes pour la détection des maladies cardiaques
variables_of_interest = ['Age', 'HighBP', 'HighChol', 'BMI', 'Smoker',
                         'Diabetes', 'PhysActivity', 'HeartDiseaseorAttack']

# Création d'une sous-section du dataset avec les variables pertinentes
selected_data = data[variables_of_interest]

# Affichage des premières lignes du dataset pour vérification
print("Avant l'exploration des données :")
print(selected_data.head())

# Visualisation des distributions des variables pertinentes
plt.figure(figsize=(15, 10))
selected_data.hist(bins=20, color='blue', edgecolor='black', alpha=0.7)
plt.suptitle('Distributions des Variables pour la Détection des Maladies Cardiaques
# Ajouter un retour à la ligne
plt.tight_layout()
plt.show()
```

```
Avant l'exploration des données :
    Age  HighBP  HighChol   BMI  Smoker  Diabetes  PhysActivity  \
0   9.0     1.0       1.0  40.0     1.0       0.0           0.0
1   7.0     0.0       0.0  25.0     1.0       0.0           1.0
2   9.0     1.0       1.0  28.0     0.0       0.0           0.0
3  11.0     1.0       0.0  27.0     0.0       0.0           1.0
4  11.0     1.0       1.0  24.0     0.0       0.0           1.0

   HeartDiseaseorAttack
0                   0.0
1                   0.0
2                   0.0
3                   0.0
4                   0.0
<Figure size 1500x1000 with 0 Axes>
```



- **Analyse des Corrélations :** Une matrice de corrélation a été construite pour évaluer les relations linéaires entre les différentes variables. Cette analyse a permis d'identifier des associations potentielles entre les facteurs de risque et la présence de maladies cardiaques. La matrice de corrélation, présentée ci-dessous, met en évidence les liens entre les variables.

```
# 2. Visualisation des corrélations entre les variables
correlation_matrix = data.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Matrice de Corrélation')
plt.show()
```



6. Appliquer les techniques de fouille de données :

Dans le projet de détection des maladies cardiaques, j'ai opté pour le **RandomForestClassifier**, un algorithme d'ensemble basé sur des arbres de décision. Cette décision repose sur la nature hétérogène des données et la capacité de **RandomForest** à gérer efficacement les caractéristiques complexes et les interactions non linéaires.

Voici le code pour initialiser et entraîner le modèle **RandomForest** :

```python
from sklearn.model_selection import train_test_split
# Création d'une sous-section du dataset avec les variables pertinentes
selected_data = data[variables_of_interest]

# Division des données en ensemble d'entraînement et ensemble de test (80% / 20% ici)
train_data, test_data = train_test_split(selected_data, test_size=0.2, random_state=42)

# Affichage de la taille des ensembles d'entraînement et de test
print("Taille de l'ensemble d'entraînement :", len(train_data))
print("Taille de l'ensemble de test :", len(test_data))

# Séparation des variables indépendantes (X) et de la variable cible (y)
X_train = train_data.drop('HeartDiseaseorAttack',axis=1)
y_train = train_data['HeartDiseaseorAttack']
X_test = test_data.drop('HeartDiseaseorAttack', axis=1)
y_test = test_data['HeartDiseaseorAttack']
```

```
Taille de l'ensemble d'entraînement : 28076
Taille de l'ensemble de test : 7019
```

```python
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score
from sklearn.ensemble import RandomForestClassifier

# Entraînement du modèle (Random Forest)
model_rf = RandomForestClassifier()
model_rf.fit(X_train, y_train)

# Prédiction sur les données de test
y_pred_rf = model_rf.predict(X_test)

# Affichage des résultats
print("Random Forest Classifier:")
print("\nRapport de classification :\n", classification_report(y_test, y_pred_rf))
print("Matrice de confusion :\n", confusion_matrix(y_test, y_pred_rf))
print("\nAUC-ROC Score:", roc_auc_score(y_test, model_rf.predict_proba(X_test)[:, 1]))
```

```
Random Forest Classifier:

Rapport de classification :
              precision    recall  f1-score   support

         0.0       0.92      0.97      0.95      6450
         1.0       0.22      0.09      0.13       569

    accuracy                           0.90      7019
   macro avg       0.57      0.53      0.54      7019
weighted avg       0.87      0.90      0.88      7019

Matrice de confusion :
 [[6275  175]
 [ 519   50]]

AUC-ROC Score: 0.7079180120161851
```

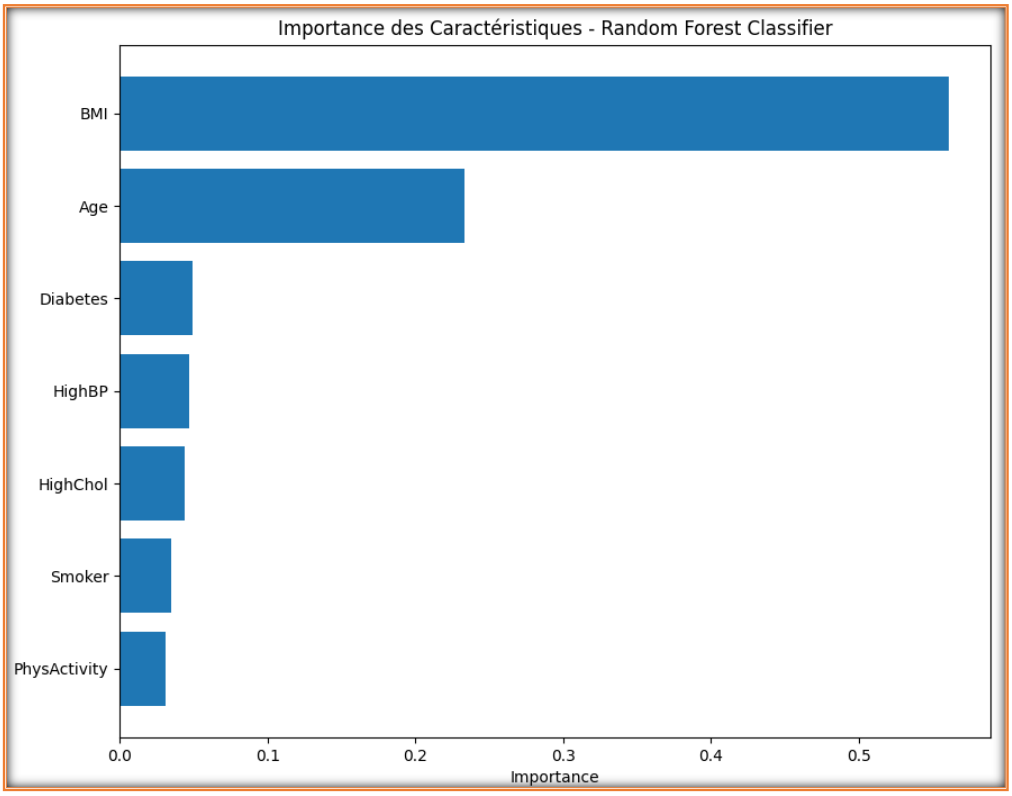
7. Interprétation du modèle et établissement des conclusions :

L'étape d'interprétation du modèle et d'établissement des conclusions est cruciale pour comprendre comment le modèle fonctionne et quelles caractéristiques sont importantes.

- **Importance des Caractéristiques :** Nous avons extrait les importances relatives de chaque caractéristique du modèle et les avons affichées dans un graphique à barres horizontales. Ce graphique met en évidence les variables qui ont le plus influencé les prédictions du modèle. Les caractéristiques telles que l'âge, la pression artérielle élevée (HighBP), le taux de cholestérol élevé (HighChol), et l'Indice de Masse Corporelle (BMI) semblent être parmi les plus importantes dans la détection des maladies cardiaques.

```
# Afficher l'importance des caractéristiques
feature_importances = pd.Series(model_rf.feature_importances_, index=X_train.columns)
print("Feature Importances:\n", feature_importances)

# Visualiser l'importance des caractéristiques
import matplotlib.pyplot as plt

# Trier les caractéristiques par importance
sorted_idx = feature_importances.argsort()
sorted_feature_importances = feature_importances[sorted_idx]

# Créer un graphique à barres horizontales pour visualiser l'importance des caractéristiques
plt.figure(figsize=(10, 8))
plt.barh(range(len(sorted_feature_importances)), sorted_feature_importances)
plt.yticks(range(len(sorted_feature_importances)), X_train.columns[sorted_idx])
plt.xlabel('Importance')
plt.title('Importance des Caractéristiques - Random Forest Classifier')
plt.show()
```

```
Feature Importances:
 Age                0.233074
HighBP              0.047141
HighChol            0.044008
BMI                 0.561272
Smoker              0.034645
Diabetes            0.049068
PhysActivity        0.030793
```



- Analyse des Résultats : Voici une interprétation plus détaillée des résultats :
  - **Age (23.31%) :** L'âge émerge comme le facteur prédominant dans la prédiction des maladies cardiaques, avec une importance de 23,31%. Cela souligne l'impact majeur de l'âge en tant que facteur de risque.
  - **BMI (Indice de Masse Corporelle) (56.13%) :** L'IMC joue un rôle crucial, surpassant toutes les autres variables avec une importance de 56,13%. Cela souligne l'importance du poids corporel dans la compréhension des risques cardiaques.
  - **HighBP (Pression Artérielle Élevée) (4.71%) :** Bien que moins important que l'âge et le BMI, l'hypertension artérielle reste un facteur significatif avec une importance de 4,71%.
  - **HighChol (Cholestérol Élevé) (4.40%) :** Le taux de cholestérol élevé contribue également de manière significative avec une importance de 4,40%.
  - **Diabetes (Diabète) (4.91%) :** Le diabète montre une importance notable de 4,91%, soulignant son association avec les maladies cardiaques.
  - **Smoker (Tabagisme) (3.46%) :** Le tabagisme, bien que moins impactant, conserve une importance de 3,46%.
  - **PhysActivity (Activité Physique) (3.08%) :** L'activité physique a la plus faible importance parmi les variables, avec une contribution de 3,08%.

⇨ **Conclusion :** Cette analyse des résultats met en lumière la complexité des facteurs influençant la prédiction des maladies cardiaques. L'âge et l'IMC se distinguent comme des contributeurs majeurs, soulignant l'importance d'une approche holistique pour la gestion des risques. Ces résultats fournissent

des informations précieuses pour le développement de stratégies de prévention personnalisées, mettant en avant la nécessité de sensibiliser aux risques liés à l'âge, au poids corporel, à la pression artérielle, au taux de cholestérol, au diabète et au tabagisme. Les interventions visant à promouvoir un mode de vie sain et à atténuer ces facteurs de risque peuvent contribuer de manière significative à la réduction des maladies cardiaques au sein de la population.

Conclusion :

En conclusion, cette étude approfondie sur la détection des maladies cardiaques a permis d'explorer les divers aspects liés aux facteurs de risque et à la prédiction de ces affections majeures. En analysant un ensemble de données exhaustif, nous avons identifié des tendances significatives, mettant en avant l'âge et l'IMC comme des éléments clés dans la prévision des maladies cardiaques. Ces résultats soulignent l'importance de la vigilance envers certains indicateurs de santé et la nécessité d'une approche proactive pour la gestion des risques cardiovasculaires. Ce rapport offre ainsi des contributions substantielles à la compréhension de la dynamique complexe entourant les maladies cardiaques, fournissant des bases solides pour des initiatives futures visant à prévenir et à traiter efficacement ces problèmes de santé cruciaux.