Algorithme des k plus proches voisins

I. Introduction :

Le projet consiste à étudier une base de données sur les maladies cardiaques. Les données de notre Dataset sont des données étiquetées, donc nous effectuerons un traitement de problème d'apprentissage supervisé.
En apprentissage supervisé, un algorithme reçoit un ensemble de données qui est étiqueté avec des valeurs de sorties correspondantes sur lequel il va pouvoir s'entraîner et définir un modèle de prédiction.
Cet algorithme pourra par la suite être utilisé sur de nouvelles données afin de prédire leurs valeurs de sorties correspondantes.

On retrouve comme caractéristique ces données par :

1. Le nombre de grossesses.
2. Le taux de glycémie 2 heures après l'absorption d'une solution sucrée (g de glucose par L de sang).
3. La pression sanguine diastolique (mmHg).
4. L'épaisseur du pli cutané au niveau du triceps (indicateur de masse graisseuse, mm).
5. La mesure de l'effet de l'insuline (muU/ml).
6. L'indice de masse corporelle (kg/m 2)
7. Le résultat d'une fonction calculant le risque de développer un diabète qui prend en compte l'hérédité.
8. L'âge.

II. Analyse statistique descriptive :

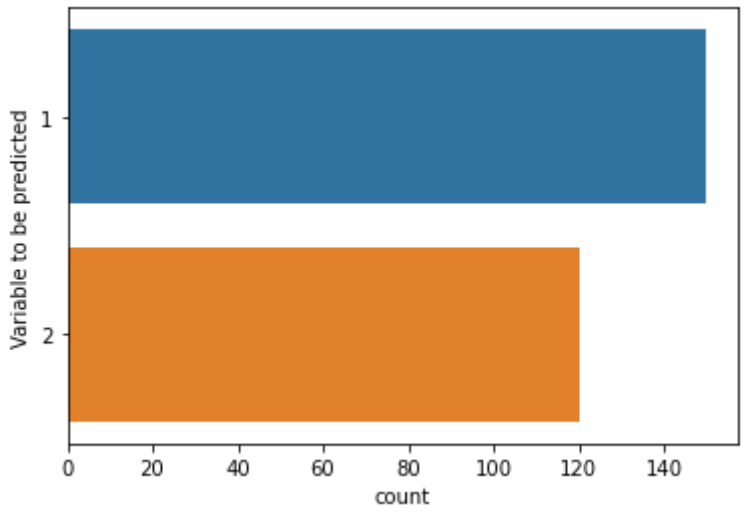
Avant de traiter les données nous devons vérifier qu'il n'y a aucunes valeurs null parmi les données :

```
age                                                 0
sex                                                 0
chest pain type                                     0
resting blood pressure                              0
serum cholestoral                                   0
fasting blood sugar                                 0
resting electrocardiographic results                0
maximum heart rate achieved                         0
exercise induced angina                             0
oldpeak                                             0
the slope of the peak exercise ST segment           0
number of major vessels (0-3) colored by flourosopy 0
thal fixed reversable defect                        0
Variable to be predicted                            0
```

- Histogramme :

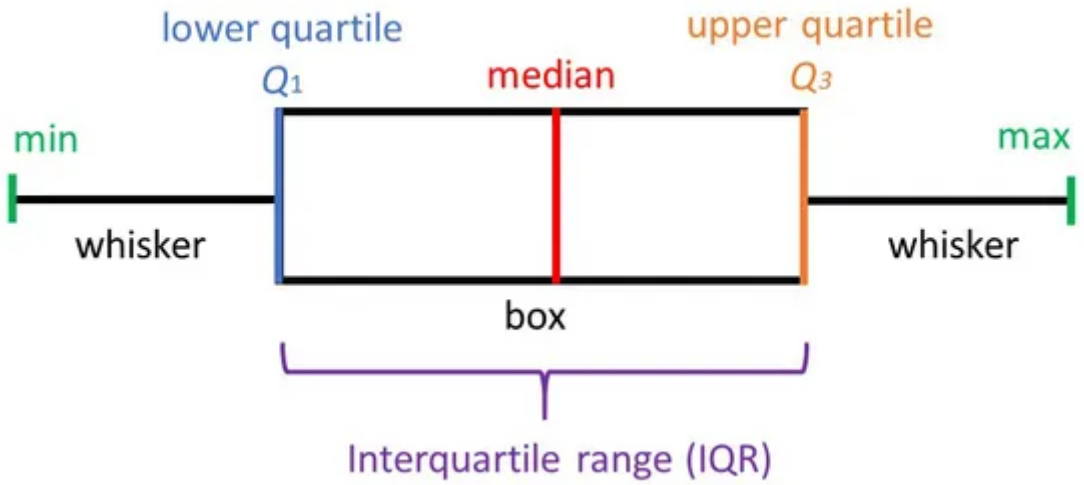
Les variables utilisé pour la prédiction sont composées de 1 et 2 :

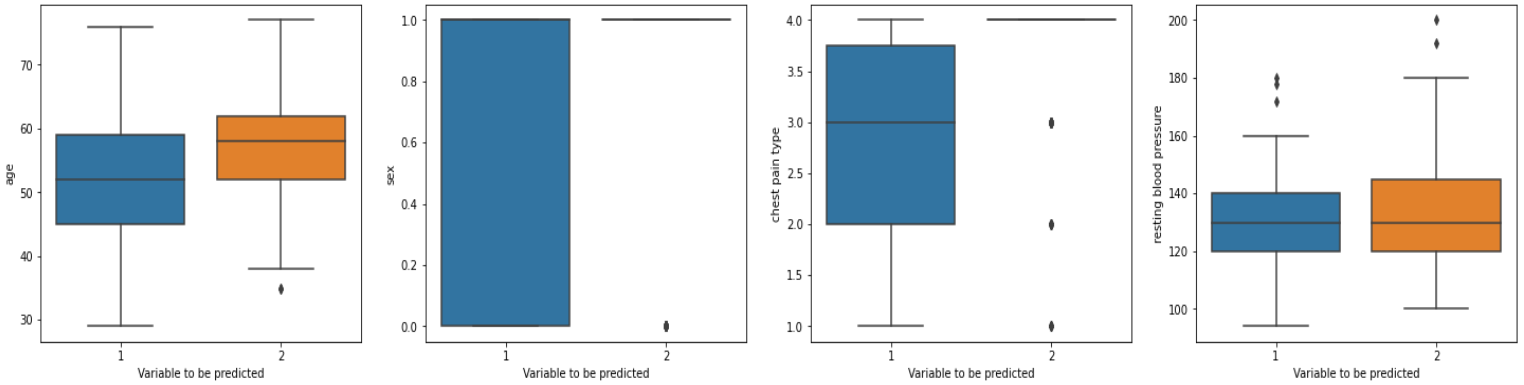
Parmi le nombre de composants, nous avons 120 variables de 2 et 150 composant 1.



- Les boxplots :

Dans les statistiques descriptives, le boxplot montrent visuellement la distribution des données numériques. Le boxplot est résumé à cinq chiffres d'un ensemble de données, y compris la note minimale, le premier quartile (inférieur), la médiane, le troisième quartile (supérieur) et la note maximale.

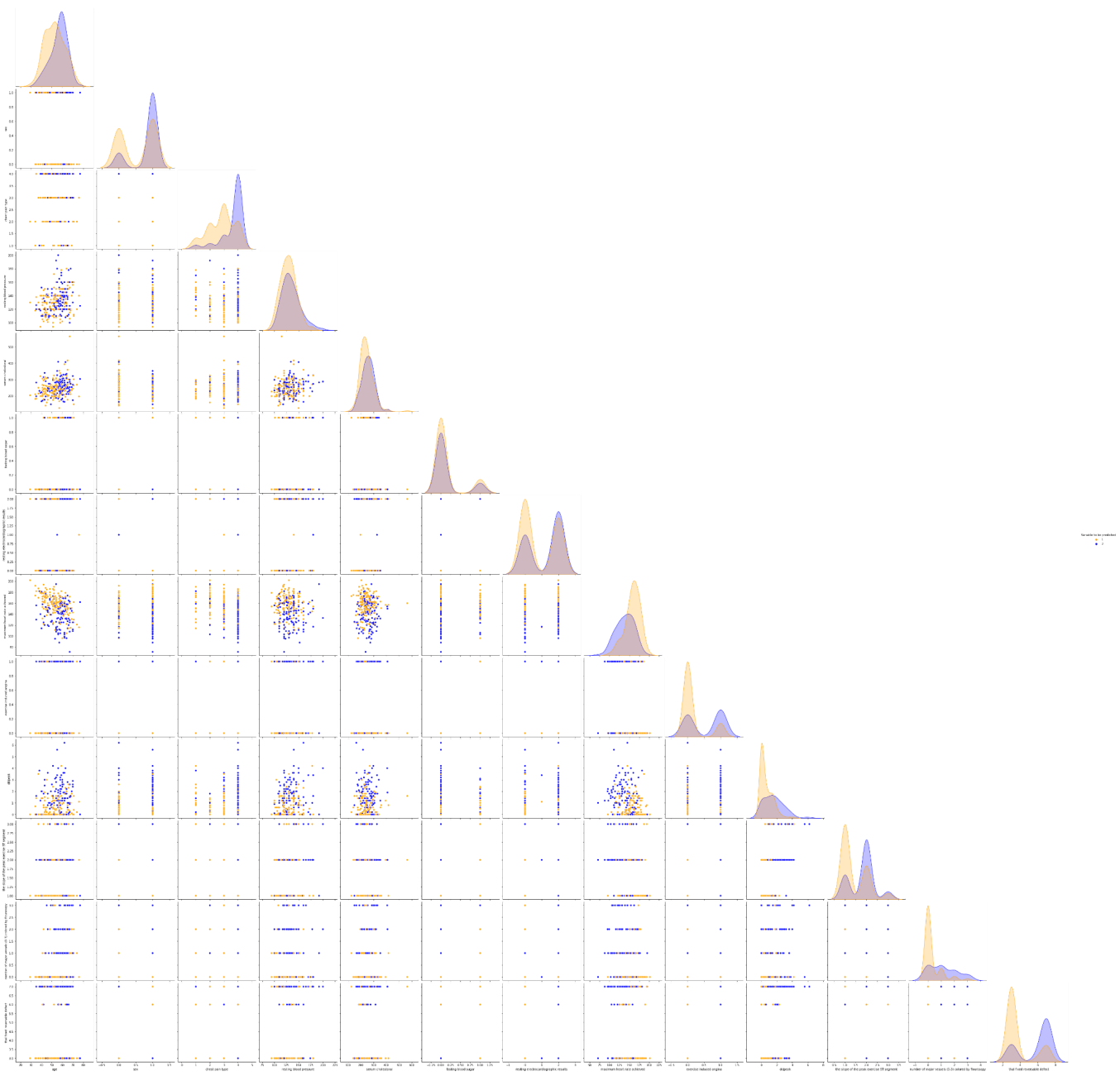




D'après les boîtes à moustache, nous constatons que le boxplot 1 et 4 caractérise mieux les variables de prédiction sur le 'age', 'resting blood pressure' parce que la distribution est moins dispersée. Par contre, les valeurs 'chest pain type' et 'resting blood pressure' sont beaucoup plus dispersées et donc 'Variable to be predicted' n'est pas la caractéristique optimale.

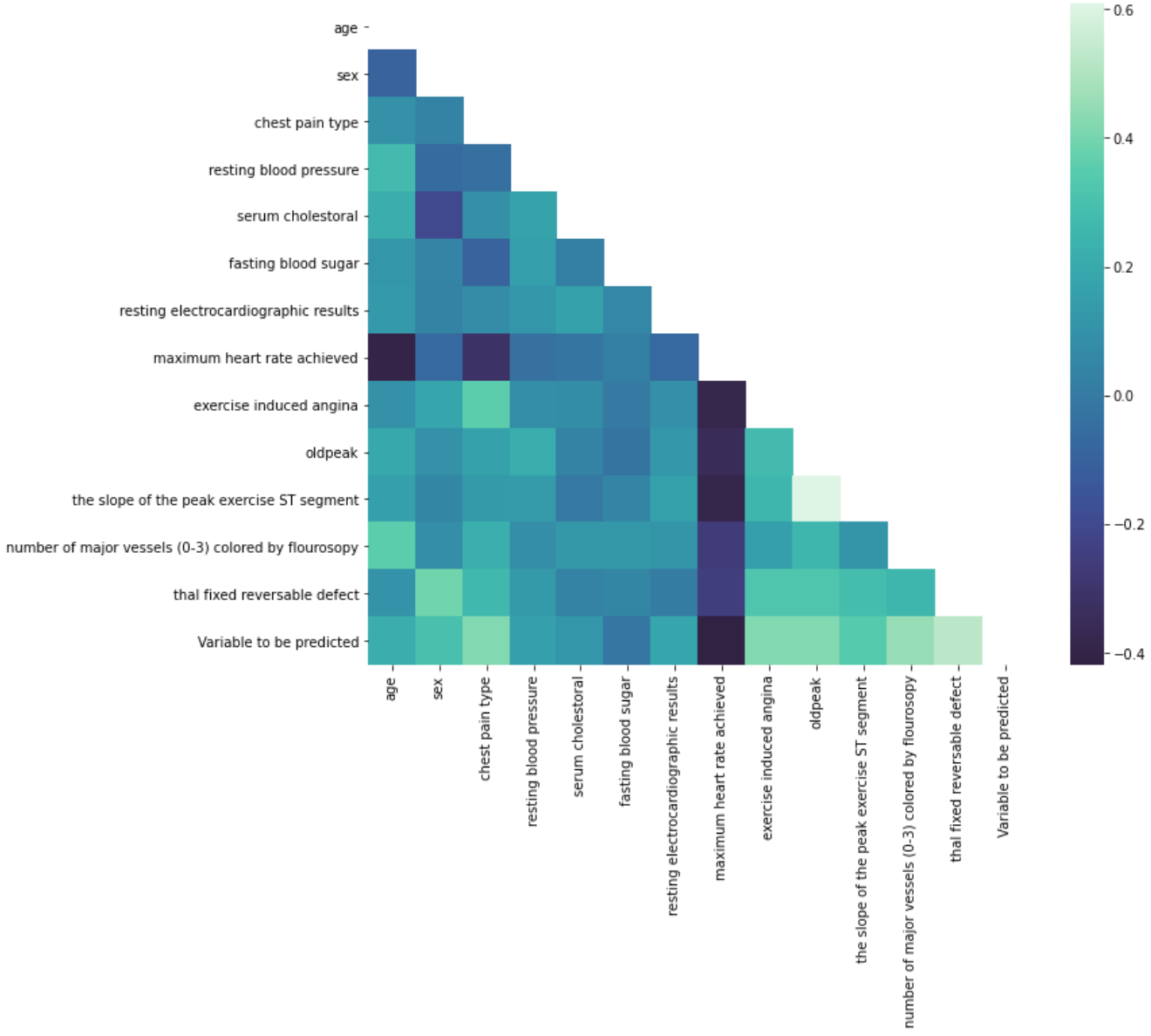
- Pairplot :

Pairplot trace une relation par paires dans un ensemble de données.
La fonction pairplot crée une grille d'axes de telle sorte que chaque variable des données sera partagée dans l'axe des y sur une seule ligne et dans l'axe des x sur une seule colonne. Cela crée des tracés comme indiqué ci-dessous :



- Heatmap :

Une heatmap est une représentation graphique où les valeurs individuelles d'une matrice sont représentées sous forme de couleurs. Cela aide à trouver des modèles et donne une perspective de profondeur.



D'après la figure, on remarque qu'il y a une faible corrélation entre les 'variables maximum heart rate achieved' avec 'exercise induced angina', 'the slope of the peak exercise ST segment' et 'Variable to be predicted'.

Nous constatons que les variables ne sont pas représentées sous la même échelle, il est nécessaire d'effectuer une normalisation.

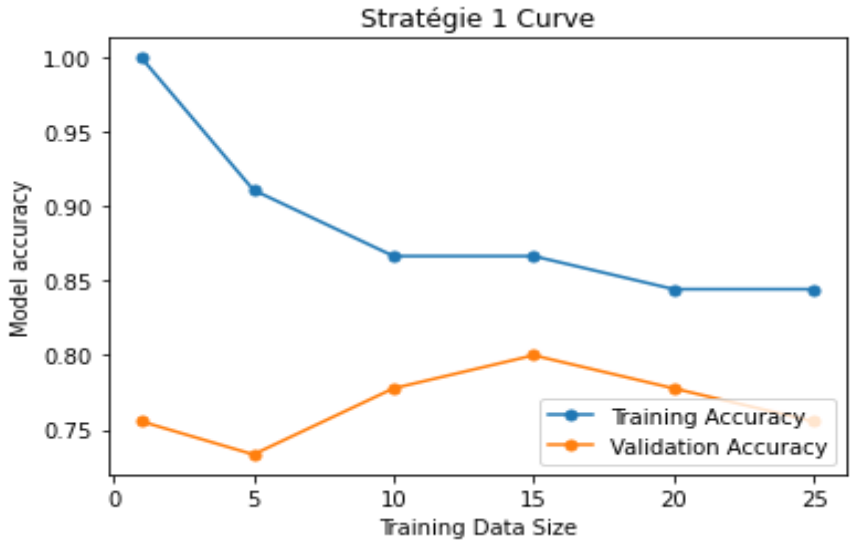
III. Stratégies 1 :

Nous pouvons à présent nous intéresser à un exemple d'utilisation de l'algorithme des K plus proches voisins.

Les paramètres de la fonction train_test_split sont :

- les données : variables prédictives
- les données : variable à prédire
- test_size : proportion de l'échantillon consacré au test
- random_state : graine du générateur aléatoire utilisé pour le découpage

Ensuite, on effectue l'apprentissage :



À la fin de l'étape précédente, on dispose des indices des éléments correspondants aux k plus proches voisins de celui qui nous a servi de référence pour le calcul des distances.

Ces voisins font partie de la base d'apprentissage, et on dispose donc pour eux d'une valeur concernant la variable à prédire. On remarque que notre modèle de validation est plus précis avec k = 15.

Il s'agit maintenant d'agréger ces labels ou ces valeurs pour déterminer ce que l'on peut affecter à ce nouvel élément.

Après évaluation de notre modèle avec k = 15 on obtient les résultats suivants :

```
Confusion Matrix:
[[67  8]
 [14 46]]
Classification Report:
              precision    recall  f1-score   support

           1       0.83      0.89      0.86        75
           2       0.85      0.77      0.81        60

    accuracy                           0.84       135
   macro avg       0.84      0.83      0.83       135
weighted avg       0.84      0.84      0.84       135

For n_neighbors = 15, Accuracy: 0.837037037037037
```

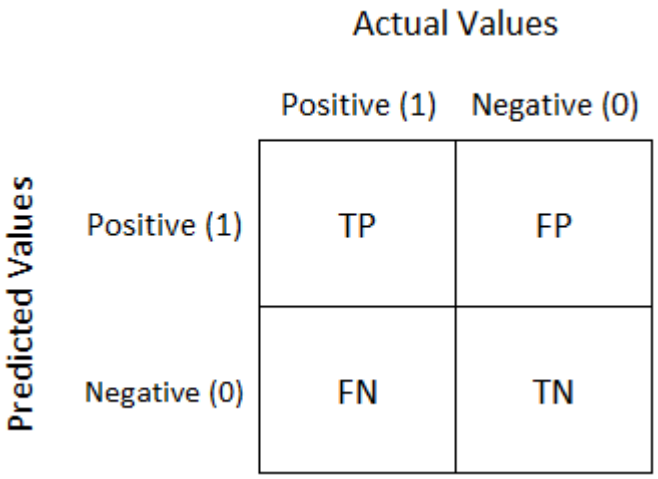
Pour un nombre de 15 voisins nous avons une précision (Accuracy) de 83,7 %

Grace a cette valeur, on peut calculer le

Taux d'erreur = 1 - Précision globale (Accuracy)
= 1 - 0,84
= 0,16

Notre modèle est capable de prédire a hauteur de 84 % les maladies cardiaques, cependant le taux d'erreur ou le taux de précision de notre modèle n'est pas suffisant pour évaluer les maladies cardiaques.

Il faut déduire les indicateurs suivants, que nous déduirons à partir de la matrice de confusion ((67,8), (15,46)) :



Remarque : Vous constaterez que la matrice de coût doit être adaptée : il doit être 5 fois plus couteux de prédire une absence d'attaque cardiaque lorsqu'en réalité cette attaque cardiaque a lieu, que de prédire la présence d'une attaque cardiaque, lorsqu'en réalité celle-ci n'a pas lieu.

On en déduit que FP = 5 x FP.

Le rappel (recall) est la proportion de vrais positifs, elle traduit la sensibilité (sensitivity), ici, il s'agit de la capacité du modèle à détecter tous les chats.

Rappel = TP / (TP + FP) = 67/(67+8×5) = 0,63

La spécificité (specificity), est la proportion de vrais négatifs. Ici, elle traduit la capacité du modèle à ne pas détecter de chat quand ce n'en est pas un.

Spécificité =TP / (TP + FN) = 67/ (67+14) = 0,83

La précision (precision), est la proportion de bonnes prédictions dans les prédictions positives. Ici, la précision traduit la capacité du modèle à ne détecter comme chats que de vrais chats.

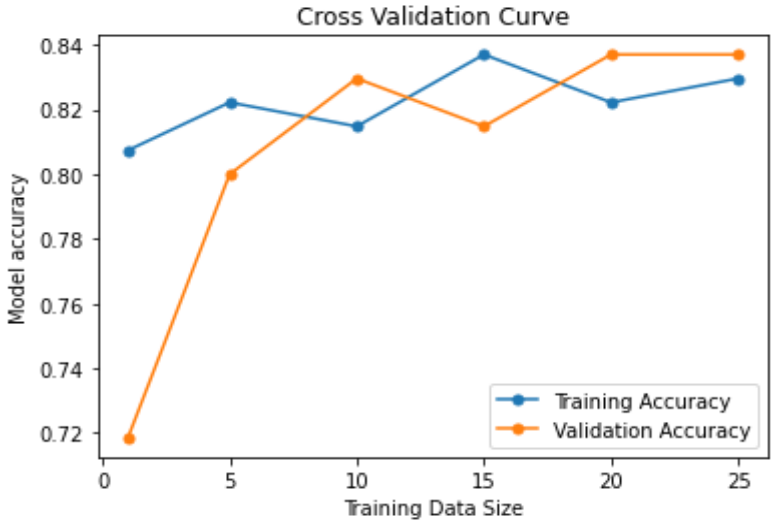
Précision = TN/ (TN + FP) = 46/(46+8x5)= 0,53

Enfin la F -mesure, qui mesurer son rendement en termes d'exactitude.
Il s'agit d'une mesure utile du modèle dans le cas où l'on tente d'optimiser la précision ou le score de rappel, par conséquent la performance du modèle en souffre.

recall-precision : F = 2TP / (TP + FP + FN) = 2x67 / (67+8x5+14) = 1,1

IV. Stratégies 2 :

La différence entre l'approche par validation croisée K-folds et l'approche par découpage apprentissage/test est qu'en validation croisée, toutes les données sont utilisées pour l'apprentissage et pour la prédiction. En validation croisée 5-folds par exemple les données sont toutes utilisées 4 fois pour apprendre et une fois pour prédire. Donc toutes les données sont vues en apprentissage et en test, mais aucune donnée n'est utilisée en même temps pour apprendre et prédire.



Et on calcule un score obtenu par comparaison des prédictions et des vraies valeurs :

```
Confusion Matrix:
[[69  6]
 [15 45]]
Classification Report:
              precision    recall  f1-score   support

           1       0.82      0.92      0.87        75
           2       0.88      0.75      0.81        60

    accuracy                           0.84       135
   macro avg       0.85      0.83      0.84       135
weighted avg       0.85      0.84      0.84       135

For n_neighbors = 20, Accuracy: 0.8444444444444444
```

Pour un nombre de 20 voisins, nous avons une précision (Accuracy) de 84%.

Grace a cette valeur, on peut calculer le

Taux d'erreur = 1 - Précision globale (Accuracy)
= 1 - 0,84
= 0,16

Notre modèle est capable de prédire a hauteur de 84 % les maladies cardiaques, cependant le taux d'erreur ou le taux de précision de notre modèle n'est pas suffisant pour évaluer les maladies cardiaques.

Il faut déduire les indicateurs suivants, que nous déduirons à partir de la matrice de confusion ((69,6), (15,45)) :

Le rappel (recall) est la proportion de vrais positifs, elle traduit la sensibilité (sensitivity), ici, il s'agit de la capacité du modèle à détecter tous les chats.

Rappel = TP / (TP + FP) = 69/(69+6×5) = 0,70

La spécificité (specificity), est la proportion de vrais négatifs. Ici, elle traduit la capacité du modèle à ne pas détecter de chat quand ce n'en est pas un.

Spécificité =TP / (TP + FN) = 69/ (69+14) = 0,83

La précision (precision), est la proportion de bonnes prédictions dans les prédictions positives. Ici, la précision traduit la capacité du modèle à ne détecter comme chats que de vrais chats.

Précision = TN/ (TN + FP) = 45/(45+6x5)= 0,60

Enfin la F -mesure, qui mesurer son rendement en termes d'exactitude.
Il s'agit d'une mesure utile du modèle dans le cas où l'on tente d'optimiser la précision ou le score de rappel, par conséquent la performance du modèle en souffre.

recall-precision : F = 2TP / (TP + FP + FN) = 2x69 / (69+6x5+14) = 1,2

V. Conclusion :

Après comparaison des deux stratégies, on remarque que la deuxième stratégie est meilleure.
En effet, toutes les valeurs d'évaluations sont supérieurs dans la stratégie 2.

En général, un modèle qui surpasse un autre modèle sur le plan de la précision et du rappel est probablement le meilleur modèle.

L'algorithme des k plus proches voisins (kNN) est un algorithme de Machine Learning supervisé simple qui peut être utilisé pour résoudre des problèmes de classification et de régression. Il est facile à mettre en œuvre et à comprendre. Cependant, plus la taille des données est grande plus il devient lent.

kNN recherche les distances entre une requête cible et toutes les observations des données. Ensuite il sélectionne les k observations les plus proches de la requête.

De ce fait, on peut améliorer notre algorithme en cherchant nous même la meilleure valeur de k voisin. Il existe une fonction GridSearchCV qui nous permet de retrouver les meilleurs paramètres de notre modèle.
Pour finir avec la fonction validation_curve de sklearn nous avons la possibilité d'afficher les différentes courbes d'analyse des voisins en apprentissage et en validation.