Projet du module “Algorithmique pour IA”

Master 1 Data Science, UIE, Bamako

Participants:

Abissa Ag Abdousalam

Hammed bakoroba Boundi

Souleymane Ibrahim Maïga

# Partie I/A : Prédiction de la fréquence cardiaque maximale

Dans cette première partie, nous nous concentrons sur l’étude de la fréquence cardiaque maximale atteinte, la variable thalach dans le jeu de données (voir Figure 1).

## 1. Questions préliminaires

(a) Type des variables

• thalach (int64) : Variable quantitative continue, représentant la fréquence cardiaque maximale atteinte.  
• age : numérique continue  
• sex : variable catégorielle binaire (0 pour féminin, 1 pour masculin)  
• chol : Variable quantitative continue

(b) Vérification du lien entre chol et thalach

Pour vérifier l'existence d'une relation entre chol et thalach, on peut calculer le coefficient de corrélation de Pearson. Ce coefficient indique la force et la direction du lien linéaire entre les deux variables.  
Une autre approche consiste à tracer un nuage de points (scatter plot) pour visualiser une éventuelle tendance.

## 2. Mise en place d'un modèle de régression linéaire

(a) Est-ce que la régression linéaire est adaptée ?

La régression linéaire est adaptée si la relation entre thalach et les variables explicatives (age, sex, trestbps, chol, oldpeak et thal) est linéaire. Pour le vérifier, on peut observer les relations entre ces variables à l'aide de diagrammes de dispersion et calculer les coefficients de corrélation.

(b) Identification de la variable étiquette

La variable que nous cherchons à prédire est thalach, donc c'est notre variable cible, notée y.

(c) Expression mathématique du modèle

L'expression du modèle de régression linéaire pour un patient i est donnée par :  
  
où :  
- yi est la fréquence cardiaque maximale (thalach)  
- x1i, x2i, ... sont les caractéristiques du patient  
- β0, β1, ... sont les paramètres du modèle  
- εi est l'erreur résiduelle

(d) Entraînement du modèle

(i) Objectif de l'entraînement

L'entraînement vise à ajuster les paramètres β de manière à minimiser l'erreur entre les valeurs prédites et les valeurs réelles de thalach.

(ii) Choix de la fonction perte

Nous utilisons l'erreur quadratique moyenne (MSE) comme fonction perte, car elle pénalise fortement les erreurs importantes et est couramment utilisée en régression.

Formule de l’ Erreur quadratique moyenne (MSE):



(iii) Algorithme d'optimisation

L'algorithme de descente de gradient est souvent utilisé pour minimiser la fonction perte en ajustant les paramètres du modèle à chaque itération.

Formule :



Fonctionne de l’algorithme de la descente de gradient:

On part de paramètres initiaux aléatoires;

À chaque étape, on regarde dans quelle direction la fonction de coût augmente (en calculant la dérivée);

On ajuste les paramètres dans la direction opposée, comme si on descendait une pente;

On répète jusqu'à trouver le minimum

(iv) Limite et alternative a cet algorithme:

Les limites de la descente de gradient :

Ça peut rester bloqué dans des minima locaux (comme s'il trouvait une petite cuvette alors qu'il y a une vallée plus profonde ailleurs)

Le choix de la "taille des pas" (learning rate) est crucial : trop grand et ça diverge, trop petit et c'est très lent

Ça marche mal quand les données ont des échelles très différentes

Alternative : Maximum de Vraisemblance (MLE)

Particulièrement adapté quand :

Le modèle a une formulation probabiliste

On cherche à maximiser la probabilité des observations

Les estimateurs ont des propriétés statistiques optimales (ex : sans biais)

(e) Importance de la séparation des données

La séparation en données d'entraînement et de test permet d'éviter le sur-apprentissage et de mesurer la capacité du modèle à généraliser à de nouvelles données.

(f) Indicateur de qualité du modèle

Le coefficient de détermination R² est un bon indicateur, car il mesure la proportion de variance expliquée par le modèle.

(g) Prédiction pour un nouveau patient

On utilise l'équation du modèle entraîné pour prédire thalach d'un nouveau patient en remplaçant les variables x1 à x6 par leurs valeurs et en appliquant les coefficients estimés.

# Partie I/B : Prédiction de la présence de la maladie cardiaque

Dans cette section, nous allons aborder la problématique de la prédiction de la présence d’une maladie cardiaque en utilisant des méthodes de classification en Machine Learning.

## 3. Justification du choix d’un modèle de classification

(a) Sommes-nous d’accord avec l’utilisation d’un modèle de classification ?

Oui, l’utilisation d’un modèle de classification est tout à fait appropriée pour ce problème. En effet, la variable cible **target** est une variable binaire qui prend deux valeurs possibles :

* **0** : absence de maladie cardiaque.
* **1** : présence de maladie cardiaque.

Un modèle de classification permet d’attribuer une classe à un nouvel individu en fonction des caractéristiques observées. Ce type de modèle est adapté pour les problèmes où l’objectif est d’assigner un nouvel élément à l’une des catégories définies. Et dans notre cas, il sagit de dire si oui ou non un patient est malade du coeur.

(b) À quelle classe de problèmes de Machine Learning appartient ce problème ?

Ce problème appartient à la classe des **problèmes de classification supervisée**. En effet :

* Nous disposons d’un ensemble de données labellisé autrement dit étiquetées (les patients dont on connaît déjà l’état de santé).
* L’objectif est d’entraîner un modèle qui apprend à prédire la classe d’un nouvel individu sur la base des exemple de cas passé.

4. Choix de l’algorithme de classification

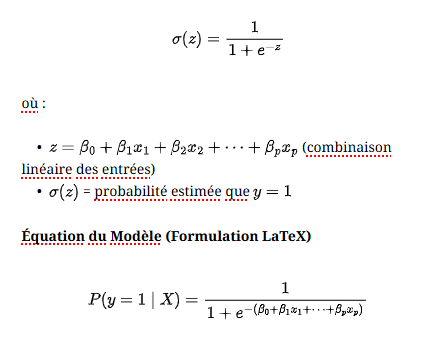
L’algorithme recommandé est **la régression logistique** pour la prédiction de la maladie cardiaque.

#### Pourquoi la régression logistique ?

* **Facile a interpreté** : Il permet d’analyser l’influence de chaque variable explicative sur la probabilité d’être malade.
* **Robuste pour les petits ensembles de données** : Notre dataset contient seulement **303 observations**, ce qui est suffisant pour la régression logistique.
* **Efficace pour les problèmes binaires** : La ***régression logistique*** ou **modèle logit** est un modèle de [régression](https://fr.wikipedia.org/wiki/R%C3%A9gression_(statistiques)) principalement utilisé en [classification binaire](https://en.wikipedia.org/wiki/Binary_classification). La fonction sigmoïde permet de transformer les valeurs en probabilités comprises entre **0 et 1**.

#### Principe de la régression logistique

La régression logistique est un modèle qui s’appuie sur la **fonction sigmoïde** :



où :

* P(Y=1∣X) est la probabilité qu’un patient soit malade.
* β0​ est le biais.
* βi​ sont les coefficients associés aux variables explicatives Xi.
* La classification se fait en appliquant un **seuil** (généralement 0.5) sur la probabilité obtenue.

## 5. Évaluation du modèle de classification

Nous utilisons les ensembles de valeurs suivantes :

ysubset=(1,1,0,0,1,0,1,0,0,1)

PEUT ETRE IMG

y^subset=(1,0,1,0,1,1,1,1,0,1)

(a) Pseudo-code pour le calcul de l’accuracy

L’accuracy est définie comme le rapport entre le nombre de bonnes prédictions et le nombre total d’échantillons :

IMG

**Pseudo-code :**

# Données d'entrée

ysubset = [1, 1, 0, 0, 1, 0, 1, 0, 0, 1]

y\_pred\_subset = [1, 0, 1, 0, 1, 1, 1, 1, 0, 1]

# compteur de prédictions correctes

correct\_predictions = 0

# Boucle pour comparer les prédictions avec les vraies valeurs

for i in range(len(ysubset)):

if ysubset[i] == y\_pred\_subset[i]:

correct\_predictions += 1

# Calcul de l'accuracy

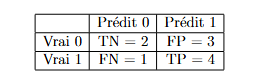
accuracy = correct\_predictions / len(ysubset)

# le resultat retourné

print("L'accuracy est :", accuracy)

(b) Matrice de confusion

La matrice de confusion permet d’évaluer les performances du modèle :



Explication :

* **TP (True Positives) = 4** : Nombre de malades bien prédits.
* **FP (False Positives) = 3** : Nombre de non-malades mal classés comme malades.
* **TN (True Negatives) = 2** : Nombre de non-malades bien prédits.
* **FN (False Negatives) = 1** : Nombre de malades mal classés comme non-malades.

(c) Calcul des métriques de performance

1. **Accuracy :**



→ Le modèle a une **précision de 60%**.

1. **Sensibilité (Recall) :**



→ **80% des malades sont correctement détectés**.

1. **Spécificité :**



→ **Seulement 40% des non-malades sont bien identifiés**.

1. **Spécificité (taux de faux positifs) :**



→ **60% des patients non-malades sont incorrectement diagnostiqués comme malades**.

(d) Méthode de comparaison de plusieurs modèles de classification

Pour comparer plusieurs modèles, on peut utiliser :

* **La validation croisée (k-fold cross-validation)** : Permet d’évaluer la performance en divisant les données en plusieurs sous-ensembles d’entraînement et de test.
* **L'AUC-ROC (Area Under the Curve - Receiver Operating Characteristic)** : Mesure la capacité du modèle à distinguer les classes.
* **Le F1-score** : Moyenne harmonique entre précision et rappel.
* **Le score de log-vraisemblance (Log Loss)** : Utilisé pour évaluer la qualité des probabilités fournies par le modèle.

## Source:

[Régréssion logistique (Wikipedia)](https://fr.wikipedia.org/wiki/R%C3%A9gression_logistique)

[Régréssion lineaire](https://datatab.fr/tutorial/linear-regression)

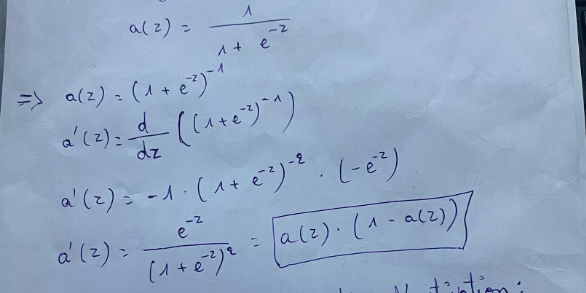
Google Recherche

ChatGPT

# Partie II / A : Théorie de la descente de gradient dans un DNN simple.

**1/** Montrons que la dérivée 𝑎’(𝑧) = 𝑎(𝑧)(1 − 𝑎(𝑧)) :

La fonction sigmoïde est définie par :



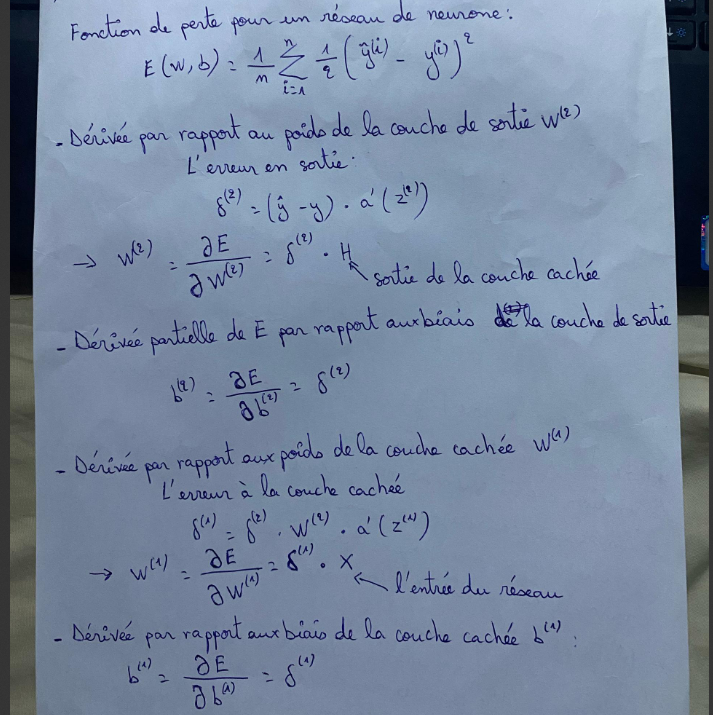
2) Décrivons le rôle / effet d’une fonction d’activation :

**Rôle d’une fonction d’activation :**La fonction d’activation permet d’assurer la non-linéarité dans un réseau de neurones. Sans elle, le réseau ne serait plus qu’une simple combinaison linéaire des entrées, ce qui limiterait sa capacité à apprendre des modèles complexes.

**Effets de la sigmoïde :**

* **Compression :** Prend n’importe quelle valeur réelle et la compresse entre 0 et 1, utile pour la classification binaire.
* **Non-linéarité :** Permet au réseau de neurones d’apprendre des modèles non linéaires.
* **Problème de saturation :** Si la valeur d’entrée est très grande ou très petite, la sigmoïde sature (proche de 0 ou de 1), ce qui peut ralentir l’apprentissage.

**3/** Calcul des dérivées partielles de la fonction de perte E(W, b) :



**4/** Déduction des formules de mise à jour d’un algorithme de descente de gradient stochastique :

