# Régression Logistique : Explication et Formules Mathématiques

## Introduction

La **régression logistique** est une méthode de classification utilisée pour prédire une variable binaire, c'est-à-dire une variable qui prend deux valeurs possibles (par exemple, 0 ou 1). Contrairement à la régression linéaire, elle est conçue pour donner des probabilités, plutôt qu'une prédiction continue, et pour être utilisée avec des variables dépendantes discrètes.

## 1. Hypothèse de la régression logistique

En régression logistique, on cherche à prédire la probabilité que la variable cible $y$ prenne une valeur de 1, étant donné les valeurs des variables explicatives $x_1, x_2, \ldots, x_n$.

La probabilité que $y = 1$ peut être exprimée comme suit :

$$P(y = 1|x) = h_\theta(x)$$

où $h_\theta(x)$ est la fonction d'hypothèse, qui est définie par une fonction logistique ou sigmoïde. La fonction sigmoïde transforme une valeur de n'importe quelle étendue en une valeur comprise entre 0 et 1, ce qui permet de l'interpréter comme une probabilité.

La fonction sigmoïde est définie par :

$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n)}}$$

où :

- $\theta_0, \theta_1, \ldots, \theta_n$ sont les paramètres du modèle (ou les coefficients de régression),
- $x_1, x_2, \ldots, x_n$ sont les valeurs des variables explicatives.

## 2. Interprétation de la fonction sigmoïde

La fonction sigmoïde transforme la combinaison linéaire $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$ en une probabilité comprise entre 0 et 1. En d'autres termes, pour des

valeurs de $\theta^T x$ grandes et positives, $h_\theta(x)$ s'approche de 1, et pour des valeurs de $\theta^T x$ grandes et négatives, $h_\theta(x)$ s'approche de 0.

## 1. Définition de la fonction sigmoïde

La fonction sigmoïde est donnée par :

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}}$$

où :

- $\theta$ est le vecteur des coefficients du modèle (paramètres),
- $X$ est le vecteur des prédicteurs (variables explicatives),
- $\theta^T X$ est le produit scalaire entre $\theta$ et $X$, également appelé *logit* ou score.

La fonction sigmoïde transforme une valeur réelle $z = \theta^T X$ en une probabilité comprise entre 0 et 1. Elle a les propriétés suivantes :

- Lorsque $z \to +\infty$, $h_\theta(X) \to 1$,
- Lorsque $z \to -\infty$, $h_\theta(X) \to 0$,
- Lorsque $z = 0$, $h_\theta(X) = 0.5$.

Ainsi, la fonction sigmoïde agit comme une frontière de décision en attribuant une probabilité à chaque observation.

## 2. Pourquoi utiliser une fonction sigmoïde ?

Dans la régression linéaire, la relation entre les prédicteurs $X$ et la cible $y$ est modélisée comme une combinaison linéaire :

$$\hat{y} = \theta^T X$$

Cependant, cette approche ne fonctionne pas bien pour les problèmes de classification, car $\hat{y}$ peut prendre des valeurs en dehors de l'intervalle $[0, 1]$, ce qui n'est pas cohérent avec une probabilité.

La fonction sigmoïde garantit que les prédictions sont toujours comprises entre 0 et 1, ce qui les rend interprétables comme des probabilités.

## 3. Lien avec le logarithme des cotes (log-odds)

La régression logistique modélise le logarithme des cotes (*log-odds*) comme une combinaison linéaire des prédicteurs :

$$\ln\left(\frac{P(y=1|X)}{1 - P(y=1|X)}\right) = \theta^T X$$

où :

- $\frac{P(y=1|X)}{1-P(y=1|X)}$ est le rapport des chances (*odds*) de la classe $y = 1$,
- $\ln(\cdot)$ est la transformation logarithmique.

Cette relation montre que la régression logistique établit un lien direct entre les prédicteurs et les chances d'appartenance à la classe $y = 1$.

### 4. Exemple numérique

Considérons un problème de classification avec un prédicteur $x$, une constante ($\theta_0$) et un coefficient ($\theta_1$) :

$$z = \theta_0 + \theta_1 x$$

et la probabilité :

$$P(y = 1|x) = \frac{1}{1 + e^{-(\theta_0+\theta_1 x)}}$$

Supposons :

- $\theta_0 = -1$ (biais),
- $\theta_1 = 0.5$ (coefficient).

Calculons $P(y = 1|x)$ pour différentes valeurs de $x$ :

1. Si $x = 0$, alors $z = -1$ et :

$$P(y = 1|x = 0) = \frac{1}{1 + e^1} \approx 0.2689$$

2. Si $x = 2$, alors $z = -1 + 0.5 \cdot 2 = 0$ et :

$$P(y = 1|x = 2) = \frac{1}{1 + e^0} = 0.5$$

3. Si $x = 6$, alors $z = -1 + 0.5 \cdot 6 = 2$ et :

$$P(y = 1|x = 6) = \frac{1}{1 + e^{-2}} \approx 0.8808$$

## 3. Fonction de coût (Log-Loss)

Pour estimer les paramètres $\theta$, on utilise une fonction de coût qui mesure la différence entre les prédictions du modèle et les valeurs réelles. En régression logistique, la fonction de coût (aussi appelée log-vraisemblance négative) est définie comme suit :

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[ y^{(i)} \log \left( h_\theta \left( x^{(i)} \right) \right) + \left( 1 - y^{(i)} \right) \log \left( 1 - h_\theta \left( x^{(i)} \right) \right) \right]$$

où :

- $m$ est le nombre total d'exemples d'entraînement,
- $y^{(i)}$ est la valeur réelle de la cible pour le $i$-ème exemple,
- $h_\theta\left(x^{(i)}\right)$ est la prédiction pour le $i$-ème exemple.

La fonction de coût est conçue de manière à pénaliser fortement les prédictions incorrectes, et en minimisant $J(\theta)$, on ajuste les paramètres du modèle pour que les prédictions soient les plus précises possible.

# Explication de la fonction coût en régression logistique

La fonction coût utilisée pour la régression logistique est donnée par :

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(X^{(i)})) + (1-y^{(i)})\log(1-h_\theta(X^{(i)}))\right],$$

où :

- $m$ est le nombre d'exemples dans les données d'entraînement,
- $X^{(i)}$ est le vecteur de caractéristiques pour l'exemple $i$,
- $y^{(i)} \in \{0,1\}$ est la classe réelle pour l'exemple $i$,
- $h_\theta(X^{(i)})$ est l'hypothèse ou prédiction du modèle, définie comme :

$$h_\theta(X^{(i)}) = \sigma(\theta^T X^{(i)}) = \frac{1}{1+e^{-\theta^T X^{(i)}}}.$$

## 1. Intuition derrière la fonction coût

La fonction coût mesure à quel point les prédictions $h_\theta(X^{(i)})$ s'éloignent des valeurs réelles $y^{(i)}$. Elle est basée sur le principe de l'entropie croisée. Examinons les deux termes dans la somme :

- Lorsque $y^{(i)} = 1$, seule la partie $y^{(i)}\log(h_\theta(X^{(i)}))$ contribue au coût. Ce terme pénalise fortement le modèle si $h_\theta(X^{(i)})$ est proche de 0.
- Lorsque $y^{(i)} = 0$, seule la partie $(1-y^{(i)})\log(1-h_\theta(X^{(i)}))$ contribue au coût. Ce terme pénalise fortement le modèle si $h_\theta(X^{(i)})$ est proche de 1.

## 2. Pourquoi cette fonction est-elle appropriée ?

1. **Lien avec la probabilité :** La régression logistique modélise la probabilité conditionnelle $P(y|X)$. Maximiser cette probabilité pour les données d'entraînement revient à minimiser la fonction coût ci-dessus (principe du maximum de vraisemblance).

2. **Convexité :** Pour la régression logistique, la fonction $J(\theta)$ est convexe, ce qui garantit qu'il existe un minimum global et qu'on peut le trouver efficacement en utilisant des algorithmes comme la descente de gradient.

3. **Sensible aux grandes erreurs :** La fonction coût est très sensible lorsque les prédictions $h_\theta(X^{(i)})$ sont très éloignées des valeurs réelles $y^{(i)}$, ce qui force le modèle à bien séparer les classes.

### 3. Cas particuliers

- **Prédictions parfaites :**
  - Si $y^{(i)} = 1$ et $h_\theta(X^{(i)}) = 1$, le coût est nul ($\log(1) = 0$).
  - Si $y^{(i)} = 0$ et $h_\theta(X^{(i)}) = 0$, le coût est également nul ($\log(1) = 0$).
- **Prédictions complètement erronées :**
  - Si $y^{(i)} = 1$ et $h_\theta(X^{(i)}) = 0$, le coût devient infiniment grand ($\log(0) \to -\infty$).
  - Si $y^{(i)} = 0$ et $h_\theta(X^{(i)}) = 1$, le coût devient également infiniment grand ($\log(0) \to -\infty$).

### 4. Résumé mathématique

La fonction coût combine les cas où $y^{(i)} = 1$ et $y^{(i)} = 0$ dans une seule expression :

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[ y^{(i)} \log(h_\theta(X^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(X^{(i)})) \right].$$

Elle est conçue pour être minimisée, ce qui correspond à maximiser la probabilité que le modèle prédit correctement les classes.

## 4. Maximisation de la vraisemblance

L'objectif de la régression logistique est de trouver les valeurs de $\theta$ qui maximisent la vraisemblance des observations. La vraisemblance est définie comme la probabilité des observations données les paramètres $\theta$. En prenant le logarithme de la fonction de vraisemblance, on obtient la log-vraisemblance, qui est maximisée lors de l'entraînement du modèle.

## 5. Algorithme de gradient pour la mise à jour des paramètres

La mise à jour des paramètres se fait généralement par descente de gradient. Pour chaque itération de la descente de gradient, les paramètres $\theta_j$ sont mis à jour selon la règle suivante :

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

où :

- $\alpha$ est le taux d'apprentissage,
- $\frac{\partial J(\theta)}{\partial \theta_j}$ est la dérivée partielle de la fonction de coût par rapport au paramètre $\theta_j$.

En prenant la dérivée de la fonction de coût par rapport à $\theta_j$, on obtient :

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left( h_\theta \left( x^{(i)} \right) - y^{(i)} \right) x_j^{(i)}$$

Ainsi, à chaque itération, le modèle ajuste les paramètres de manière à minimiser la fonction de coût, ce qui conduit à des prédictions de plus en plus précises.

# 6. Résumé des formules clés

## Hypothèse (fonction sigmoïde)

$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n)}}$$

## Fonction de coût (log-loss)

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[ y^{(i)} \log \left( h_\theta \left( x^{(i)} \right) \right) + \left( 1 - y^{(i)} \right) \log \left( 1 - h_\theta \left( x^{(i)} \right) \right) \right]$$

## Mise à jour des paramètres (descente de gradient)

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left( h_\theta \left( x^{(i)} \right) - y^{(i)} \right) x_j^{(i)}$$

Ces formules permettent d'entraîner un modèle de régression logistique pour la classification binaire.

# Pourquoi maximiser la vraisemblance en régression logistique ?

L'objectif de la régression logistique est de trouver les paramètres $\theta$ qui maximisent la vraisemblance des observations. Voici une explication détaillée.

## 1. Concept de vraisemblance

La vraisemblance mesure la probabilité que les données observées aient été générées par le modèle pour un ensemble donné de paramètres $\theta$. En notation mathématique, la vraisemblance est définie par :

$$L(\theta) = P(\{y^{(i)}, X^{(i)}\}_{i=1}^{m}|\theta).$$

Maximiser la vraisemblance revient à trouver les paramètres $\theta$ qui rendent la probabilité des observations la plus grande possible.

## 2. Lien avec l'apprentissage supervisé

En apprentissage supervisé, l'objectif est d'entraîner un modèle qui prédit correctement les étiquettes $y^{(i)}$ à partir des caractéristiques $X^{(i)}$. Maximiser la vraisemblance garantit que le modèle apprend à prédire des probabilités proches de 1 pour les étiquettes correctes $y^{(i)}$, et proches de 0 pour les étiquettes incorrectes.

## 3. Régression logistique et probabilités

En régression logistique, l'hypothèse $h_\theta(X)$ représente la probabilité que $y = 1$ :

$$h_\theta(X) = P(y = 1|X; \theta).$$

Pour une observation $(X^{(i)}, y^{(i)})$, la probabilité associée est donnée par :

$$P(y^{(i)}|X^{(i)}; \theta) = \begin{cases} h_\theta(X^{(i)}) & \text{si } y^{(i)} = 1, \\ 1 - h_\theta(X^{(i)}) & \text{si } y^{(i)} = 0. \end{cases}$$

Cette probabilité peut être écrite de manière compacte :

$$P(y^{(i)}|X^{(i)}; \theta) = \left(h_\theta(X^{(i)})\right)^{y^{(i)}} \left(1 - h_\theta(X^{(i)})\right)^{1-y^{(i)}}.$$

La vraisemblance totale pour l'ensemble des données est alors le produit des probabilités individuelles :

$$L(\theta) = \prod_{i=1}^{m} P(y^{(i)}|X^{(i)}; \theta).$$

## 4. Log-vraisemblance

Pour simplifier les calculs et éviter les problèmes numériques, on utilise le logarithme de la vraisemblance, appelé log-vraisemblance :

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} \log P(y^{(i)}|X^{(i)};\theta).$$

En remplaçant $P(y^{(i)}|X^{(i)};\theta)$, on obtient :

$$\ell(\theta) = \sum_{i=1}^{m} \left[ y^{(i)} \log(h_\theta(X^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(X^{(i)})) \right].$$

## 5. Pourquoi maximiser ?

Maximiser la vraisemblance revient à ajuster les paramètres $\theta$ pour que :

- $h_\theta(X^{(i)}) \approx 1$ lorsque $y^{(i)} = 1$,
- $h_\theta(X^{(i)}) \approx 0$ lorsque $y^{(i)} = 0$.

Cela garantit que le modèle prédit les bonnes classes avec des probabilités élevées, optimisant ainsi ses performances.

## 6. Fonction coût et optimisation

En pratique, on minimise l'opposé de la log-vraisemblance pour trouver les paramètres optimaux $\theta$. La fonction coût est donnée par :

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[ y^{(i)} \log(h_\theta(X^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(X^{(i)})) \right].$$

Minimiser $J(\theta)$ revient à maximiser la log-vraisemblance $\ell(\theta)$.

## 7. Avantages de la log-vraisemblance

- **Facilité de calcul :** Le logarithme transforme un produit en somme, ce qui simplifie les calculs.
- **Stabilité numérique :** Le logarithme évite des produits de probabilités très petites, réduisant les erreurs d'arrondi.
- **Lien avec l'entropie croisée :** La log-vraisemblance est équivalente à l'entropie croisée, une mesure standard en apprentissage automatique.

# Maximisation de la vraisemblance en régression logistique

L'objectif de la régression logistique est de trouver les paramètres $\theta$ qui maximisent la vraisemblance des observations. Voici une explication détaillée.

## 1. Définition de la vraisemblance

En régression logistique, l'hypothèse $h_\theta(X)$ représente la probabilité que $y = 1$ :

$$h_\theta(X) = P(y = 1|X; \theta) = \sigma(\theta^T X),$$

où $\sigma(z) = \frac{1}{1+e^{-z}}$ est la fonction sigmoïde.

Pour un exemple $i$, la probabilité associée à la sortie $y^{(i)}$ est donnée par :

$$P(y^{(i)}|X^{(i)}; \theta) = \begin{cases} h_\theta(X^{(i)}) & \text{si } y^{(i)} = 1, \\ 1 - h_\theta(X^{(i)}) & \text{si } y^{(i)} = 0. \end{cases}$$

Cette probabilité peut être écrite de manière compacte :

$$P(y^{(i)}|X^{(i)}; \theta) = \left(h_\theta(X^{(i)})\right)^{y^{(i)}} \left(1 - h_\theta(X^{(i)})\right)^{1-y^{(i)}}.$$

La vraisemblance totale pour tous les exemples est alors le produit des probabilités individuelles :

$$L(\theta) = \prod_{i=1}^{m} P(y^{(i)}|X^{(i)}; \theta).$$

## 2. Log-vraisemblance

Pour des raisons pratiques, on travaille avec le logarithme de la vraisemblance, appelé log-vraisemblance :

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} \log P(y^{(i)}|X^{(i)}; \theta).$$

En remplaçant $P(y^{(i)}|X^{(i)}; \theta)$, on obtient :

$$\ell(\theta) = \sum_{i=1}^{m} \left[ y^{(i)} \log(h_\theta(X^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(X^{(i)})) \right].$$

Cette log-vraisemblance mesure la qualité des prédictions du modèle.

## 3. Maximisation de la log-vraisemblance

L'objectif est de maximiser $\ell(\theta)$, c'est-à-dire de trouver les paramètres $\theta$ qui rendent les prédictions $h_\theta(X)$ les plus probables pour les données observées.

**a) Fonction coût**

Au lieu de maximiser la log-vraisemblance, on minimise son opposé, ce qui revient au même. La fonction coût $J(\theta)$ est définie comme suit :

$$J(\theta) = -\ell(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(X^{(i)})) + (1-y^{(i)})\log(1-h_\theta(X^{(i)}))\right].$$

**b) Méthode d'optimisation**

Pour minimiser $J(\theta)$, on utilise des algorithmes d'optimisation comme :

- La descente de gradient, où les paramètres $\theta$ sont mis à jour selon :

  $$\theta := \theta - \alpha\nabla J(\theta),$$

  où $\nabla J(\theta)$ est le gradient de $J(\theta)$.

## 4. Intuition derrière la maximisation de la vraisemblance

Maximiser la vraisemblance revient à ajuster les paramètres $\theta$ pour que :

- $h_\theta(X^{(i)})$ soit proche de 1 lorsque $y^{(i)} = 1$,
- $h_\theta(X^{(i)})$ soit proche de 0 lorsque $y^{(i)} = 0$.

Ainsi, le modèle apprend à attribuer des probabilités élevées aux bonnes classes.

## 5. Pourquoi utiliser la log-vraisemblance ?

- **Facilité de calcul :** Le logarithme transforme le produit des probabilités en une somme, simplifiant les calculs.
- **Stabilité numérique :** Le logarithme évite que des produits de probabilités très petites conduisent à des erreurs numériques.
- **Lien avec l'entropie croisée :** La log-vraisemblance est étroitement liée à l'entropie croisée, une mesure standard de divergence entre distributions.

# Comprendre le taux d'apprentissage

Le taux d'apprentissage, noté $\alpha$, est un paramètre clé dans les algorithmes d'optimisation comme la descente de gradient, utilisés pour entraîner un modèle d'apprentissage automatique. Il détermine la taille des pas que l'algorithme effectue à chaque itération pour ajuster les paramètres (ou poids) du modèle, dans le but de minimiser la fonction de coût.

## Fonctionnement pratique

- **Petits pas avec $\alpha$ :** Si $\alpha$ est petit, le modèle effectue des ajustements mineurs à chaque itération. Cela rend l'entraînement plus lent mais potentiellement plus précis, car le modèle est moins susceptible de dépasser le minimum de la fonction de coût. Ce choix est utile pour trouver un minimum global, mais il peut rendre l'entraînement très long.
- **Grands pas avec $\alpha$ :** Si $\alpha$ est plus grand, les pas seront plus importants et l'algorithme atteindra un minimum plus rapidement. Cependant, un $\alpha$ trop grand peut faire en sorte que l'algorithme saute par-dessus le minimum et ne converge jamais, ou bien qu'il converge vers un minimum local sous-optimal.

## Exemple

Supposons que l'on cherche le minimum d'une fonction en forme de vallée. Avec un taux d'apprentissage trop grand, le modèle peut rebondir d'un côté à l'autre de la vallée sans jamais atteindre le point le plus bas. Avec un taux d'apprentissage trop petit, le modèle progresse lentement mais peut mieux se stabiliser près du minimum global.

## En résumé

Le taux d'apprentissage doit être choisi de manière judicieuse :

- **Petit** : précis mais lent.
- **Grand** : rapide mais potentiellement imprécis ou instable.

Trouver le bon taux d'apprentissage est crucial et peut nécessiter plusieurs essais pour obtenir la meilleure performance.

## Comment choisir le taux d'apprentissage

Choisir le bon taux d'apprentissage est crucial pour l'entraînement d'un modèle. Voici quelques méthodes et conseils pour le déterminer :

### 1. Essais et ajustements

- **Commencez avec un taux d'apprentissage faible** : Essayez une valeur relativement basse, comme 0,01 ou 0,001, et observez si le modèle converge (c'est-à-dire si la fonction de coût diminue régulièrement).
- **Augmentez progressivement** : Si la convergence est trop lente, augmentez le taux progressivement (par exemple, de 0,01 à 0,1) et vérifiez l'impact sur la précision et la vitesse de convergence.

### 2. Utilisation de la technique de recherche par grille (grid search)

Cette méthode consiste à tester plusieurs valeurs possibles pour le taux d'apprentissage dans une plage (par exemple, de 0,0001 à 0,1) et à sélectionner celle qui donne le meilleur équilibre entre la précision du modèle et le temps d'entraînement.

### 3. Approche du taux d'apprentissage variable (annealing)

Vous pouvez commencer avec un taux d'apprentissage relativement élevé, puis le réduire progressivement au cours de l'entraînement. Cette technique permet au modèle d'apprendre rapidement au début, puis de se stabiliser pour éviter de manquer le minimum global.

$$\alpha = \alpha_0 \times \frac{1}{1 + k \times \text{epoch}}$$

où $\alpha_0$ est le taux d'apprentissage initial, et $k$ est une constante de réduction.

### 4. Utilisation de méthodes de taux d'apprentissage adaptatif

Certaines variantes de la descente de gradient, comme **Adam**, **RMSprop** ou **AdaGrad**, adaptent automatiquement le taux d'apprentissage au cours de l'entraînement, ce qui peut simplifier la tâche de sélection du bon $\alpha$.

### 5. Visualisation de la courbe d'entraînement

Tracez la fonction de coût en fonction des itérations :

- Si le taux est trop élevé, vous verrez des oscillations importantes, et la fonction de coût pourrait ne pas converger.

- Si le taux est trop faible, la courbe de coût diminuera lentement.

Un bon taux d'apprentissage produit une courbe de coût qui diminue de manière régulière, sans oscillations.

## En résumé

Le choix du taux d'apprentissage est un processus d'ajustement. Commencez avec une petite valeur, observez les courbes d'erreur et de coût, et ajustez en conséquence. Utiliser des techniques comme l'annealing ou des optimisateurs adaptatifs peut aussi vous aider à automatiser cette sélection.