

Machine Learning Supervisé: Classification et Régression Logistique

Dr. EL BENANY Mohamed Mahmoud

January 25, 2025

Introduction au Machine Learning Supervisé

- ▶ Objectif: prédire une **étiquette** (y) à partir des **caractéristiques** (X).
- ▶ La Classification consiste à attribuer des catégories discrètes ($y \in \{0, 1\}$).
- ▶ La Régression Logistique est une méthode de classification utilisée pour prédire des probabilités.
- ▶ Étapes principales:
 1. Modélisation avec une fonction hypothèse.
 2. Définition d'une fonction coût.
 3. Optimisation via Gradient Descent.

Fonction Hypothèse

- ▶ La fonction hypothèse de la régression logistique est donnée par:

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

où:

- ▶ $h_{\theta}(X)$ est la probabilité que $y = 1$.
 - ▶ θ est le vecteur des paramètres du modèle.
- ▶ Exemple en Python:

```
1 import numpy as np
2
3 def sigmoid(z):
4     return 1 / (1 + np.exp(-z))
5
6 def hypothesis(theta, X):
7     return sigmoid(np.dot(X, theta))
```

- ▶ Le résultat est une probabilité ($0 \leq h_{\theta}(X) \leq 1$).

Fonction Coût

- ▶ La fonction coût est définie comme:

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

- ▶ Cette fonction mesure l'erreur entre les prédictions $h_{\theta}(X)$ et les valeurs réelles y .
- ▶ Exemple en Python:

```
1 def cost_function(theta, X, y):  
2     m = len(y)  
3     H = hypothesis(theta, X)  
4     return -np.sum(y * np.log(H) + (1 - y) *  
                    np.log(1 - H)) / m
```

- ▶ Minimiser $J(\theta)$ permet d'optimiser le modèle.

Gradient Descent

- ▶ Les paramètres θ sont mis à jour par:

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

où:

- ▶ α : taux d'apprentissage.
- ▶ $\frac{\partial J(\theta)}{\partial \theta_j}$: dérivée partielle de la fonction coût.
- ▶ Gradients pour la régression logistique:

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

Gradient Descent

► Exemple en Python:

```
1 def gradient_descent(X, y, theta, learning_rate,
2   iterations):
3     m = len(y)
4     for _ in range(iterations):
5         H = hypothesis(theta, X)
6         gradient = np.dot(X.T, (H - y)) / m
7         theta -= learning_rate * gradient
8     return theta
```

Limite de Décision

- ▶ La limite de décision sépare les classes $y = 0$ et $y = 1$.
- ▶ Elle est définie par $h_{\theta}(X) = 0.5$, soit:

$$\theta^T X = 0$$

- ▶ Exemple de visualisation en Python:

```
1 import matplotlib.pyplot as plt
2
3 def plot_decision_boundary(theta, X, y):
4     x_boundary = np.linspace(X[:, 1].min(), X
5                              [:, 1].max(), 100)
6     y_boundary = -(theta[0] + theta[1] *
7                     x_boundary) / theta[2]
8     plt.scatter(X[:, 1], X[:, 2], c=y, cmap='
9                 viridis')
10    plt.plot(x_boundary, y_boundary, 'r--')
11    plt.xlabel('Feature 1')
12    plt.ylabel('Feature 2')
13    plt.show()
```

Évaluation du Modèle

► Mesures clés:

1. ****Précision****: $\frac{TP+TN}{Total}$.
2. ****Rappel****: $\frac{TP}{TP+FN}$.
3. ****F1-Score****: Harmonic mean de la précision et du rappel.

► Exemple en Python:

```
1 from sklearn.metrics import
   classification_report
2
3 y_pred = (hypothesis(theta, X) >= 0.5).astype
   (int)
4 print(classification_report(y, y_pred))
```


Résultats et Visualisation

- Coût diminue au fil des itérations.
- Limite de décision correctement définie.

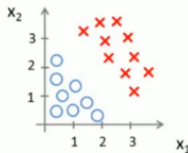


Exemple : $h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

Avec :

- $\theta_0 = -3$
- $\theta_1 = 1$
- $\theta_2 = 1$

$$-3 + x_1 + x_2 \geq 0$$



A retenir

- ▶ La régression logistique est un puissant algorithme pour la classification binaire.
- ▶ Principaux éléments:
 1. Modélisation des probabilités.
 2. Optimisation via Gradient Descent.
 3. Évaluation rigoureuse pour assurer la qualité.
- ▶ Extensible à des cas multi-classes avec Softmax Regression.

Matrice de Confusion

- ▶ Une matrice de confusion résume les performances d'un modèle de classification.
- ▶ Elle compare les prédictions du modèle aux valeurs réelles.

	Prédit Oui	Prédit Non
Réel Oui	TP (Vrai Positif)	FN (Faux Négatif)
Réel Non	FP (Faux Positif)	TN (Vrai Négatif)

Les Principales Métriques

- ▶ **Exactitude (Accuracy)** : $\frac{TP+TN}{P+N}$
- ▶ **Rappel (Recall)** : $\frac{TP}{TP+FN}$
- ▶ **Précision (Precision)** : $\frac{TP}{TP+FP}$
- ▶ **F1-Score** : $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Exemple : Données Équilibrées

Contexte :

- ▶ 50 malades (**Oui**) et 50 sains (**Non**).

	Prédit Oui	Prédit Non
Réel Oui	40 (TP)	10 (FN)
Réel Non	8 (FP)	42 (TN)

Calcul des métriques :

- ▶ **Accuracy** = 82%, **Recall** = 80%
- ▶ **Precision** = 83%, **F1-Score** = 81%

Exemple : Données Déséquilibrées

Contexte :

- ▶ 5 malades (**Oui**) et 95 sains (**Non**).

	Prédit Oui	Prédit Non
Réel Oui	4 (TP)	1 (FN)
Réel Non	10 (FP)	85 (TN)

Calcul des métriques :

- ▶ **Accuracy** = 89%, **Recall** = 80%
- ▶ **Precision** = 29%, **F1-Score** = 42%

Comparaison des Données

Tableau Comparatif :

Données	Accuracy	Recall	Precision	F1-Score
Équilibrées	82%	80%	83%	81%
Déséquilibrées	89%	80%	29%	42%

Conclusion :

- ▶ **Données équilibrées** : Toutes les métriques sont fiables.
- ▶ **Données déséquilibrées** : Attention à l'accuracy, privilégier **Recall** et **F1-Score**.

Qu'est-ce qu'une courbe ROC ?

- ▶ ROC signifie **Receiver Operating Characteristic**.
- ▶ Elle illustre la performance d'un modèle de classification binaire.
- ▶ Elle est tracée en comparant :
 - ▶ L'axe Y : **Taux de Vrais Positifs** (**Sensitivity** ou **Recall**).
 - ▶ L'axe X : **Taux de Faux Positifs** (*FP* rate).

But : Comparer les performances d'un modèle à différents seuils de classification.

Taux de Vrais et Faux Positifs

Formules importantes :

- ▶ **Taux de Vrais Positifs (TPR) :**

$$TPR = \frac{TP}{TP + FN}$$

- ▶ **Taux de Faux Positifs (FPR) :**

$$FPR = \frac{FP}{FP + TN}$$

Interprétation :

- ▶ *TPR* mesure la capacité du modèle à identifier correctement les vrais positifs.
- ▶ *FPR* mesure le taux d'erreurs où un négatif est mal classé comme positif.

Exemple : Prédications avec différents seuils

Scénario : Un modèle prédit des probabilités pour 10 échantillons (5 positifs, 5 négatifs).

Seuil	TPR (Recall)	FPR
0.9	20%	0%
0.7	60%	20%
0.5	80%	40%
0.3	100%	80%

Courbe ROC : Tracer TPR (y) contre FPR (x) pour chaque seuil.

Qu'est-ce que l'AUC ?

- ▶ **AUC** : Area Under the Curve (**Aire sous la courbe ROC**).
- ▶ Mesure la performance globale du modèle.

Interprétation :

- ▶ $AUC = 1$: Modèle parfait.
- ▶ $AUC = 0.5$: Modèle aléatoire (aucune capacité de discrimination).
- ▶ Plus l'AUC est proche de 1, meilleur est le modèle.

Exemple : AUC pour un modèle

Données :

Seuil	TPR (y)	FPR (x)
0.9	0.2	0.0
0.7	0.6	0.2
0.5	0.8	0.4
0.3	1.0	0.8

AUC = Aire des rectangles et triangles sous la courbe.

$$AUC = 0.2 \times 0.2 + 0.2 \times 0.4 + 0.1 \times 0.4 = 0.76$$

Conclusion : Le modèle est performant.

Comparaison des Modèles avec AUC

Exemple : Deux modèles

- ▶ **Modèle A** : $AUC = 0.90$
- ▶ **Modèle B** : $AUC = 0.75$

Interprétation :

- ▶ Le modèle A est meilleur pour discriminer les classes positives et négatives.
- ▶ Utiliser AUC pour choisir un modèle lors de l'entraînement.

A retenir

- ▶ La courbe ROC est un outil puissant pour évaluer les performances des modèles de classification.
- ▶ L'AUC fournit une mesure globale pour comparer différents modèles.
- ▶ Attention aux déséquilibres dans les données : AUC peut ne pas suffire dans certains cas.

Conseil : Toujours considérer le contexte métier avant de choisir un modèle basé uniquement sur l'AUC.