# 1

# Qu'est - ce qu'un score ?

L'objectif de ce chapitre est de présenter la nécessité des systèmes de scoring. Nous présenterons également les principes de construction d'un score ainsi que les principes de mesure de performance d'un score. Ce chapitre est divisé en sept sections qui vont indtroduire les concepts et présenter de manière à la fois quantitative mais également appliquée sur des cas concrets l'intégralité de ces concepts.

## 1.1 Introduction au scoring du risque de crédit

Tout d'abord, rappelons le rôle des banques.

### 1.1.1 Le rôle des banques

Les banques ont pour rôle de collecter l'épargne et de financer l'économie ; mettre en relation les prêteurs d'une part et les emprunteurs d'autre part. Ces prêteurs et emprunteurs ne sont pas aux mêmes endroits et n'opèrent pas forcément avec la même dévise, ni sur les mêmes horizons de temps. Les banques sont là également pour transformer l'argent à la fois dans l'espace et dans le temps.

Les principaux acteurs intervenants dans cette opération sont : les particuliers ; les professionnels ; les Petites et Moyennes Entreprises (PME), les Grandes Entreprises (GE) ; les Collectivités Locales et Territoriales ; l'Etat ; les marchés financiers ; le financement de projet (infrastructures, énergie, télécommunications, transport de marchandises, ...).

### 1.1.2 Les instruments de financement

Il y a principalement deux types d'instruments financiers : l'obligation et le prêt.

#### 1.1.2.1 Obligation

Les obligations sont des titres émis sur les marchés financiers. Elles sont cotées sur une bourse et échangeables.

#### 1.1.2.2 Le prêt

Il s'agit d'un accord bilatéral entre un prêteur et un emprunteur. Pour prêter à un client, une institution doit être dotée d'un agrément délivré par une banque centrale. Et donc, le prêt est le resultat de cet accord bilatéral entre une banque qui dispose d'un agrément et un emprunteur. Ce prêt est souvent assorti de clauses particulières (communément appelés covenants), de garanties personnelles (par exemple la garantie des parents) ou de collatéraux (par exemple un actif qui peut être saisi et revendu par le prêteur). Un autre exemple de collatéral est celui des prêts immobiliers : c'est la propriété qui est l'objet du financement.

### 1.1.3 Pourquoi quantifier le risque de crédit ?

Pourquoi chercher à quantifier le risque de crédit ? Premièrement, pour savoir si on accorde ou non un crédit à un client et pour en définir les conditions de taux, de maturité, les options comme par exemple les options de remboursement anticipés.

Deuxièmement, pour piloter la politique d'octroi de prêt de la banque. C'est un élément important pour la banque. Les critères d'octroi de prêt sont déterminés en fonction du couple (rendement, risque) que la banque vise.

Troisièmement, pour suivre les portefeuilles dans le temps, mesurer le risque sur un prêt ou un portefeuille de prêts.

Quatrièmement, pour mesurer la consommation en fonds propres et les provisions. Nous savons que le montant des fonds propres des banques est régulé par la Banque Centrale Européenne (BCE) et que les provisions sont mesurées par les modèles statistiques dans le cadre des normes dites IFRS 9 (*International Financial Reporting Standards*).

Enfin quantifier le risque permet de prendre les bonnes actions en cas de dégradation du risque et de prendre les bonnes décisions.

### 1.1.4 Evénement de défaut

L'événement de défaut est ce qu'on cherche à mesurer dans l'évaluation du risque de crédit (estimation du risque que l'emprunteur ne rembourse pas le crédit). Il survient lorsque l'emprunteur ne rembourse pas le crédit soit en intégralité, soit dans les temps définis par le contrat de prêt. Le risque est donc évalué par rapport au risque de défaut et l'événement de défaut répond à une définition très précise de l'Autorité Bancaire Européenne (EBA) qui est inscrite dans la réglementation prudentielle (notamment Bâle III) des établissements de crédit. Il s'agit d'un docuement de plus de 150 pages qui décrit tous les détails définissant le défaut de crédit.

**Définition du défaut de crédit**

Un client est dit en défaut lorsque les trois principaux points suivants sont observés :

- premièrement, lorsqu'il est en retard de paiement de 90 jours ou plus ;
- deuxièmement, lorsque nous avons des éléments tangibles sur le fait qu'il ne va pas payer : on dit alors qu'il est « *unlikely to pay* » ;
- troisièmement, lorsque les montants en jeu dépassent certains seuils de matérialité. Pour un particulier, on ne déclenche pas l'événement de défaut si les montants en jeu sont inférieurs à 100€. Ce seuil de matérialité est égal à 500€ pour une entreprise.

### 1.1.5 Typologie des approches

Il y a deux typologies d'approche de mesure du risque de crédit : la différentiation du risque *(risk differentiation)* et la quantification du risque *(risk quantification)*.

#### 1.1.5.1 Risk differentiation

La différentiation du risque ou *risk differentiation* consiste à segmenter les sous - portefeuilles d'emprunteurs par niveau de risque relatif et ensuite à classer ces emprunteurs par niveau de risque.

#### 1.1.5.2 Risk quantification

La quantification du risque ou *risk quantification* est l'estimation de la probabilité que l'événement de défaut survienne.

En général, ces deux approches sont réalisées de manière séquentielle : d'abord la différentiation du risque et ensuite la quantification du risque.

### 1.1.6 Autres types de scores

Le score est ce qui sert à la différentiation du risque. Outre le score de défaut, il y a des scores de fraude, de retard de paiement ou de tout autre type d'événement qui met la banque en risque sur le remboursement du prêt par rapport aux termes du contrat. Il y a aussi des scores d'acceptation d'une offre de prêt ; des scores sur le tirage du crédit parce que celui - ci a été octroyé et des scores mesurant la possibilité qu'un client quitte la banque (ce qu'on appelle le *crunch*).

Il y a donc une grande diversité du type de scores et pas uniquement sur l'événement de défaut, mais également sur un certains nombres d'événements liés soit au crédit octroyé à un client, soit au client lui - même.

**En résumé**

**Score de crédit**

Le score de crédit permet de classer les emprunteurs selon leur niveau de risque notamment leur niveau de risque relatif.

**Score de défaut**

Le score de défaut représente la probabilité que l'emprunteur ne rembourse pas ses dettes en intégralité ou dans les délais définis dans le contrat de prêt.

**Utilisation**

Le score est utilisé dans de nombreux processus métiers tels que :
- L'octroi de crédit
- La mesure et le suivi des risques des prêts et des portefeuilles
- La gestion du risque de crédit
- La gestion de la relation clientèle

Nous avons introduit dans cette section les grands principes de ce qu'on entend par score de crédit. Passons à présent à la quantification du risque par des probabilités.

## 1.2 Quantification du risque par des probabilités

### 1.2.1 But du scoring

L'objectif du score de crédit est de classer les emprunteurs par ordre de risque de crédit croissant. Le risque étant évalué à partir des observations des deux états possibles pour les emprunteurs :

- soit l'emprunteur va au défaut (situation que nous notons D=défaut) : situation des emprunteurs qui ont un défaut de paiement sur une période donnée (1 an par exemple) ;
- soit l'emprunteur survit (situation que nous notons ND = non défaut) : situation des emprunteurs qui n'ont pas de défaut de paiement sur cette période.

La méthode de consruction d'un score consiste à observer un groupe d'emprunteur sur une période (1 an par exemple) et à constater (à la fin de la période d'observation) s'ils sont dans le groupe des défauts (D) ou celui des survivants (ND).

Le score a pour but de **classer** les emprunteurs pour **prédire** la classe (D ou ND) dans laquelle nous allons ensuite les observer.

### 1.2.2 Prédictions et observations

Si nous prédisons qu'un emprunteur sera dans le groupe des survivants, nous allons le classer dans la classe des bons emprunteurs (G pour *good*). A l'inverse, si nous prédisons qu'il fera défaut, nous le classerons dans le groupe des mauvais emprunteurs (B pour *bad*). La classe G regroupe les emprunteurs pour lesquels nous prédisons la survie et la classe B celle pour lesquels nous prédisons le défaut. Ainsi, les classes G (*good*) et B (*bad*) sont celles dans lesquelles nous **faisons des prédictions** tandis que

les classes D (*default*) et ND (*not default*) sont celles dans lesquelles nous **observons** les emprunteurs.

Si le modèle de prédiction est parfait, alors les emprunteurs classés G sont tous observés dans la classe ND et les emprunteurs classés B sont tous dans la classe D. A l'inverse, si le modèle de prédiction est aléatoire, les emprunteurs classés G seront observés dans la classe D et dans la classe ND avec le même taux que les emprunteurs classés B.

### 1.2.3 Variables explicatives (features)

La probabilité pour un emprunteur d'etre en défaut dépend des caractéristiques de l'emprunteur qui influencent sa capacité de remboursement. A titre illustratif, ces caractéristiques sont : les ratios financiers (revenus, charges, niveau d'endettement,...) ; les caractéristiques socio - économiques (âge, statut marital, CSP,...) ; la performance du crédit passée ; la nature des prêts souscrits. Les variables macroéconomiques ne sont généralement pas incluses dans les modèles de score mais pourraient l'être.

L'ensemble des variables explicatives dans un modèle sont appelées les « **features** » et sont notées sous la forme d'un veceur :

$$X = (X_1, ..., X_m) \tag{1.1}$$

Ici, le vecteur $X$ a un nombre $m$ de composantes ; c'est - à - dire que nous avons pris en compte dans le score un nombre $m$ de variables qui sont censées représenter la capacité de remboursement ou les probabilités de défaut des emprunteurs.

### 1.2.4 Rappel sur les probabilités conditionnelles

**Définition 1.1** (Probabilité conditionnelle)**.** *Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espacé probabilisé. Soient $A$ et $B$ deux événements (sous - ensemble de $\Omega$) tels que $\mathbb{P}(B) \neq 0$.*

*On définit la probabilité conditionnelle de A quand B (ou sachant B) par :*

$$\mathbb{P}_B(A) = \mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \tag{1.2}$$

**Exemple 1.1.** *Portefeuille des prêts immobiliers de la banque De Vinci*

La banque De Vinci dispose d'un portefeuille de 1150 prêts immobiliers. Les données sont consignées dans le tableau (1.1) :

**Table 1.1 –** Portefeuille de 1150 prêts immobiliers selon le statut de l'emprunteur

| Statut | Non défaut (ND) | Défaut (D) | Total |
|---|---|---|---|
| Propriétaire | 600 | 30 | 630 |
| Locataire | 200 | 70 | 270 |
| Investisseur | 225 | 25 | 250 |
| Total | 1025 | 125 | 1150 |

Les empreunteurs sont répartis en trois catégories : Propriétaire, Locataire et Investisseur. Au sein de chacune des ses catégories, un certain nombre d'emprunteurs font défaut (30 propriétaires, 70 locataires et 25 investisseurs). Sur les 1150 emprunteurs, le nombre total de défaut est égal à 125 et le nombre total de survivant est 1025.

Notons par $\mathbb{P}(D)$ (resp. $\mathbb{P}(ND)$) la probabilité de défaut (resp. de survie) sur le portefeuille global. On a alors :

$$\mathbb{P}(D) = \frac{\text{Card}(D)}{\text{Card}(\Omega)} = \frac{125}{1025+125} = 0.109 = 10.9\%$$
$$\mathbb{P}(ND) = \frac{\text{Card}(ND)}{\text{Card}(\Omega)} = \frac{1025}{1025+125} = 1 - \mathbb{P}(D) = 0.891 = 89.1\%$$

Le taux de probabilité de défaut est de $10.9\%$ et celui de survie est de $89.1\%$.

Nous pouvons calculer directement la probabilité jointe de survie et d'être locataire. Nous pouvons le faire, soit par le théorème de Bayes, soit par le calcul direct des probabilités conditionnelles. Soit $\mathbb{P}(ND, \text{locataire})$ la probabilité pour un emprunteur d'être en survie et locataire.

**Approche 1 : Théorème de Bayes**

$$\begin{aligned}\mathbb{P}(NB \cap \text{Locataire}) &= \mathbb{P}(NB|\text{Locataire}) \times \mathbb{P}(\text{Locataire}) \\ &= \frac{200}{200+70} \times \frac{200+70}{1025+125} = \frac{200}{1150} = 0.174 = 17.4\%\end{aligned}$$

La probabilité d'être locataire, c'est le nombre total de locataire (ici $200 + 70 = 270$) divisé par le nombre total d'emprunteurs ($1150$).

**Approche 2 : Calcul direct**

$$\mathbb{P}(NB \cap \text{Locataire}) = \frac{\text{Card}(ND \cap \text{Locataire})}{\text{Card}(\Omega)} = \frac{200}{1025+125} = 0.174 = 17.4\%$$

### 1.2.5 Probabilités et côtes

Nous introduisons une nouvelle notion qui est la **côte**. Tout d'abord, rappelons la condition de normalisation.

**Définition 1.2** (Condition de normalisation)**.** *Soient $D$ et $ND$ deux événements indépendants. Soit $X$ le vecteur des variables explications. On appelle **condition de normalisation**, la condition suivante :*

$$\mathbb{P}(D|X) + \mathbb{P}(ND|X) = 1 \tag{1.3}$$

**Définition 1.3.** *La **côte** représente la chance qu'un événement survienne divisée par la chance qu'il ne survienne pas. En d'autres termes, on a :*

— *la côte du survie sachant $X$ notée $o(ND|X)$ est :*

$$o(ND|X) = \frac{\mathbb{P}(ND|X)}{\mathbb{P}(D|X)} \tag{1.4}$$

— *la côte du défaut sachant $X$ notée $o(D|X)$ est :*

$$o(D|X) = \frac{\mathbb{P}(D|X)}{\mathbb{P}(ND|X)} \tag{1.5}$$

**Propriété 1.1** (Propriété sur la côte)**.** *On a la relation suivante :*

$$o(ND|X) \times o(D|X) = 1 \tag{1.6}$$

**Exemple 1.2.** *Jeu de pile ou face équilibré*

Prenons le jeu de pile ou face équilibré. La probabilité d'obtenir pile ou face est la même et vaut $1/2$. Ainsi, la côte de pile ou de face vaut $1$.

$$o(\text{pile}) = \frac{\mathbb{P}(\text{pile})}{\mathbb{P}(\text{face})} = \frac{1/2}{1/2} = 1$$
$$o(\text{face}) = \frac{\mathbb{P}(\text{face})}{\mathbb{P}(\text{pile})} = \frac{1/2}{1/2} = 1$$

**Exemple 1.3.** *Jeu de pile ou face déséquilibré*

Considérons à présent un jeu de pile ou face déséquilibré où la probabilité d'obtenir pile est égale à $0.2$. Alors la côte de pile est de :

$$o(\text{pile}) = \frac{\mathbb{P}(\text{pile})}{\mathbb{P}(\text{face})} = \frac{0.2}{0.8} = \frac{1}{4}$$

La côte de pile vaut $1/4$ c'est - à - dire $4$ contre $1$ [1].

**Exemple 1.4.** *Côte sur le portefeuille des prêts immobiliers de la banque De Vinci*

Considérons l'exemple du tableau 1.1 et calculons la côte de survie sachant la variable explicative (ici le statut) :

$$\mathbb{P}(ND|\text{Propriétaire}) = \frac{600}{600+30} = 0.952 \quad \Longrightarrow \quad o(ND|\text{Propriétaire}) = \frac{600}{30} = 20$$
$$\mathbb{P}(ND|\text{Locataire}) = \frac{200}{200+70} = 0.74 \quad \Longrightarrow \quad o(ND|\text{Locataire}) = \frac{200}{70} = 2.9$$
$$\mathbb{P}(ND|\text{Investisseur}) = \frac{225}{225+25} = 0.9 \quad \Longrightarrow \quad o(ND|\text{Investisseur}) = \frac{225}{25} = 9$$

La côte $o(ND|\text{Propriétaire})$ vaut $20$ c'est - à - dire $1$ contre $20$ et non $20$ contre $1$.

1. La côte ici s'exprime comme dans les courses de chevaux.

### 1.2.6 Fonction d'information

Il existe une côte particulière qui est **la côte de la population des défauts**. C'est le rapport entre la probabilité de défaut et la probabilité de survie, soit :

$$o_{\text{pop}} = \frac{\mathbb{P}(D)}{1 - \mathbb{P}(D)} = \frac{\mathbb{P}(D)}{\mathbb{P}(ND)} \tag{1.7}$$

Cette côte représente la chance qu'un emprunteur soit D a priori, avant d'avoir une information plus spécifique. La côte d'un emprunteur en défaut sachant l'information $X$ est :

$$o(D|X) = \frac{\mathbb{P}(D|X)}{\mathbb{P}(ND|X)} = \frac{\mathbb{P}(D)}{\mathbb{P}(ND)} \times \frac{\mathbb{P}(X|D)}{\mathbb{P}(X|ND)} = o_{\text{pop}} \times \mathbb{I}(X) \tag{1.8}$$

Ainsi, la côte du défaut $D$ sachant $X$ se décompose en un produit de deux termes :

— la côte du portefeuille dans sa globalité qui ne dépend pas de $X$, $o_{\text{pop}}$
— et un terme qui lui dépend de $X$

Ce deuxième terme est appelé **fonction d'information** et noté $\mathbb{I}(X)$.

**Propriété 1.2.** *Propriétés sur la fonction d'information*

— *Si $\mathbb{I}(X) > 1$ : l'emprunteur a plus de chance d'être $D$ que la moyenne des emprunteurs.*
— *Si $\mathbb{I}(X) < 1$ : l'emprunteur a plus de chance d'être $ND$ que la moyenne des emprunteurs.*

**Exemple 1.5.** *Fonction d'information et portefeuille des prêts immobiliers de la banque De Vinci*

Pour les données du tableau 1.1, on a la côte de la population :

$$o_{\text{pop}} = \frac{\mathbb{P}(D)}{\mathbb{P}(ND)} = \frac{0.109}{0.891} = 0.12$$

La fonction d'information est la suivante :

$$\begin{aligned}
\mathbb{I}(\text{Propriétaire}) &= \frac{\mathbb{P}(\text{Propriétaire}|D)}{\mathbb{P}(\text{Propriétaire}|ND)} = \frac{30/125}{600/1025} = 0.4 \\
\mathbb{I}(\text{Locataire}) &= \frac{\mathbb{P}(\text{Locataire}|D)}{\mathbb{P}(\text{Locataire}|ND)} = \frac{70/125}{200/1025} = 2.86 \\
\mathbb{I}(\text{Investisseur}) &= \frac{\mathbb{P}(\text{Investisseur}|D)}{\mathbb{P}(\text{Investisseur}|ND)} = \frac{25/125}{225/1025} = 0.91
\end{aligned}$$

**Interprétation :**

Pour les propriétaires, la fonction d'information est inférieure à 1 ; ce qui signifie que les propriétaires ont une probabilité de défaut qui est inférieure à la moyenne du portefeuille. C'est l'inverse pour les locataires : leur probabilité de défaut est supérieure à

la moyenne du portefeuille. Enfin pour les invesstisseurs, on voit que la fonction d'information est proche de 1, c'est - à - dire que la probabilité de défaut est sensiblement la même que celle de la moyenne du portefeuille.

### 1.2.7 Prédictions avec plusieurs variables

La fonction d'information quantifie le risque relatif et nous pouvons généraliser cette fonction d'information dans le cas où nous avons plusieurs variables explicatives. Par exemple, avec deux variables $X_1$ et $X_2$, on a (en utilisant le théorème de Bayes) :

$$\begin{aligned} o(D|X_1,X_2) &= \frac{\mathbb{P}(D)}{\mathbb{P}(ND)} \times \frac{\mathbb{P}(X_1,X_2|D)}{\mathbb{P}(X_1,X_2|ND)} \\ &= \frac{\mathbb{P}(D)}{\mathbb{P}(ND)} \times \frac{\mathbb{P}(X_1|D)}{\mathbb{P}(X_1|ND)} \times \frac{\mathbb{P}(X_2|X_1,D)}{\mathbb{P}(X_2|X_1,ND)} \end{aligned}$$

Nous pouvons décomposer la côte du défaut sachant le couple $(X_1, X_2)$ en un produit de trois termes :

- $\frac{\mathbb{P}(D)}{\mathbb{P}(ND)}$ qui représente la côte du portefeuille dans sa globalité
- $\frac{\mathbb{P}(X_1|D)}{\mathbb{P}(X_1|ND)}$ qui représente la fonction d'information de $X_1$
- $\frac{\mathbb{P}(X_2|X_1,D)}{\mathbb{P}(X_2|X_1,ND)}$ qui représente une information conditionnelle de $X_2$ sachant $X_1$

Ainsi, pour calculer les côtes dans un problème avec plusieurs *features*, nous allons devoir calculer des probabilités conditionnelles.

Dans la réalité, les banques ont plusieurs dizaines de variables explicatives. La combinatoire des probabilités conditionnelles devient très élevée. Par exemple, avec $10$ variables explicatives ayant chacune $5$ modalités, il y a $5^{10} \sim 9.7$ millions de probabilités conditionnelles à calculer. Or le nombre de clients est très inférieur à ce nombre. En général l'estimation des probabilités conditionnelles ne peut se faire par simple comptage. C'est pour cela que les modèles de score sont indispensable en dimension élevée.

**En résumé**

- L'observation directe des événements de défaut $(D)$ ou de survie $(ND)$ sur un portefeuille permet d'obtenir une information sur le niveau de risque de crédit.
- Si cette information est disponible en fonction de certaines caractéristiques des emprunteurs (statut résidentiel, âge,...), on dispose d'une information plus précise encore.
- L'estimation par simple comptage n'est malheureusement pas possible lorsqu'on cherche à lier le défaut à de nombreuses variables explicatives.

## 1.3 Scores simples

### 1.3.1 Scoring

On rappelle que le risque de crédit dépend des features de chacun des emprunteurs. Les features étant représentés par un vecteur $X$ et dont la fonction de score est une fonction $s(X)$. Cette fonction $s(X)$ sert à évaluer la probabilité que l'emprunteur soit en défaut sachant ses caractéristiques propres modélisées dans le vecteur $X$.

**Définition 1.4.** *La fonction de* ***score*** *est une fonction $s(X)$ des features d'un emprunteur qui sert à évaluer une estimation de la probabilité que l'emprunteur soit D.*

Nous émettons les deux hypothèses suivantes :

1. la fonction de score contient toute l'information pour déterminer la probabilité de défaut (PD, *Probability of Default*) de l'événement de défaut ou la probabilité d'être dans la classe D.
2. le score est une fonction monotone de la probabilité de défaut. Le score préserve le rang. Le score permet de classer les emprunteurs par ordre croissant de risque. Le score conduit à un classement ordinal des emprunteurs selon leur niveau de risque et non à un classement cardinal.

A partir de là, nous obtenons la relation suivante :

$$\mathbb{P}(D|s(X)) = \mathbb{P}(D|s(X), X) = \mathbb{P}(D|X) \tag{1.9}$$

### 1.3.2 Score à partir des côtes

Le logarithme des côtes a les bonnes propriétés d'un score car il est **monotone** avec le risque mesuré par $\mathbb{P}(D|X)$. Ainsi, on a :

$$s(X) = \ln o(D|X) = \ln\left(\frac{\mathbb{P}(D|X)}{1-\mathbb{P}(D|X)}\right) \tag{1.10}$$

Cette fonction est bien croissante en fonction de $\mathbb{P}(D|X)$. En inversant cette relation très simple, nous obtenons que la probabilité de défaut sachant $X$, $\mathbb{P}(D|X)$, est une **fonction logistique** du score, soit :

$$\mathbb{P}(D|X) = \frac{1}{1+\mathrm{e}^{-s(X)}} \tag{1.11}$$

Ce score $s(X)$ est la somme d'une composante portefeuille et d'une composante spécifique (dépendant des features de l'emprunteur) :

$$s(X) = \ln o(D|X) = \ln o_{\text{pop}} + \ln \mathbb{I}(X) = s_{\text{pop}} + \omega(X) \tag{1.12}$$

avec $\omega(X) = \ln \mathbb{I}(X) = \ln\left(\frac{\mathbb{P}(X|D)}{\mathbb{P}(X|ND)}\right)$.

Comme nous venons de le voir, le logarithme de la côte est un score possible. Il en existe bien d'autres comme la côte elle - même par exemple. Le choix du logarithme

de la côte donne au score une autre propriété utile pour l'interprétation, celle de se décomposer en une somme de deux termes :

— un premier terme, $s_{\text{pop}}$, qui ne dépend que de la population globale du portefeuille ;
— et un second terme terme, $\omega(X)$, qui dépend uniquement des features

Ce second terme est spécifique à l'emprunteur. Cette fonction $\omega(X)$ correspond au logarithme de la fonction d'information.

Cette fonction de score possède certains avantages parmi lesquelles on peut citer le lien avec la fonction logistique ou score logistique et également la séparabilité des effets de portefeuille,$s_{\text{pop}}$ et des effets spécifiques à chaque emprunteur, $\omega(X)$.

### 1.3.3 Interprétation de la fonction $\omega(X)$

Afin de donner une interprétation à la fonction d'information $\omega(X)$, nous appliquons le théorème de Bayes aux deux termes composants cette fonction. Ainsi :

$$\mathbb{P}(X|D) = \mathbb{P}(D|X) \times \frac{\mathbb{P}(X)}{\mathbb{P}(D)}$$
$$\mathbb{P}(X|ND) = \mathbb{P}(ND|X) \times \frac{\mathbb{P}(X)}{\mathbb{P}(ND)}$$

En substituant ces deux expressions dans la formule de la fonction d'information, $\omega(X)$, nous obtenons :

$$\omega(X) = \ln\left(\frac{\mathbb{P}(D|X)/\mathbb{P}(ND|X)}{\mathbb{P}(D)/\mathbb{P}(ND)}\right) \tag{1.13}$$

Dans cette expression, le numérateur est la côte du défaut sachant $X$, et le dénominateur est la côte du défaut dans tout le portefeuille, soit :

$$\omega(X) = \ln\left(\frac{o(D|X)}{o_{\text{pop}}}\right) \tag{1.14}$$

Ainsi, lorsque $\omega(X) > 0$, l'emprunteur est plus risqué que la moyenne des emprunteurs du portefeuille.

**Exemple 1.6.** *Exemple de la banque De Vinci (statut résidentiel)*

Considérons les données du tableau 1.1 sur le portefeuille des prêts immobiliers. Calculons $s_{\text{pop}}$ :

$$s_{\text{pop}} = \ln\left(\frac{\mathbb{P}(D)}{\mathbb{P}(ND)}\right) = \ln\left(\frac{\text{Card}(D)}{\text{Card}(ND)}\right) = \ln\left(\frac{125}{1025}\right) = -2.10$$

Calculons $\omega(X)$ pour $X$ prenant la modalité propriétaire :

$$\omega_1(\text{Propriétaire}) = \ln\left(\frac{\mathbb{P}(\text{Propriétaire}|D)}{\mathbb{P}(\text{Propriétaire}|ND)}\right) = \ln\left(\frac{30/125}{600/1025}\right) = -0.89$$

Ainsi, le score pour un propriétaire est :

$$s(\text{Propriétaire}) = s_{\text{pop}} + \omega_1(\text{Propriétaire}) = -2.10 - 0.89 = -2.99$$

**Exemple 1.7.** *Exemple de la banque De Vinci (Age)*

Considérons le même portefeuille que celui du tableau 1.1 mais dont l'analyse est faite selon un autre axe, celui de l'âge des emprunteurs. Les données sont consignées dans le tableau 1.2 :

**Table 1.2 –** Portefeuille de 1150 prêts immobiliers selon l'âge de l'emprunteur

| Age | Non défaut (ND) | Défaut (D) | Total |
|---|---|---|---|
| $<30$ ans | 200 | 30 | 230 |
| $<40$ ans | 200 | 40 | 240 |
| $<50$ ans | 200 | 15 | 215 |
| $>50$ ans | 425 | 40 | 465 |
| Total | 1025 | 125 | 1150 |

Calculons la fonction $\omega_1(<30 \text{ ans})$ :

$$\omega_1(<30 \text{ ans}) = \ln\left(\frac{\mathbb{P}(<30 \text{ ans}|D)}{\mathbb{P}(<30 \text{ ans}|ND)}\right) = \ln\left(\frac{30/125}{200/1025}\right) = 0.21$$

Le score pour les moins de 30 ans est égal à :

$$s(<30 \text{ ans}) = s_{\text{pop}} + \omega_2(<30 \text{ ans}) = -2.10 + 0.21 = -1.89$$

### 1.3.4 Score simple combinant plusieurs variables : score bayésien « naïf »

Nous avons donc analysé le portefeuille selon deux axes séparément : $X$ d'une part (statut des emprunteurs) et $Y$ d'autre part (âge des emprunteurs). Si nous voulons développer un score combinant ces deux features ensemble, c'est - à - dire un score qui prend en compte à la fois le statut et l'âge, nous allons procéder de la même façon que précédemment, mais avec deux variables au lieu d'une.

**Propriété 1.3.** *Dans le cas de deux variables $X$ et $Y$, la côte s'écrit :*

$$o(D|X,Y) = \frac{\mathbb{P}(D|X,Y)}{\mathbb{P}(ND|X,Y)} = \frac{\mathbb{P}(D)}{\mathbb{P}(ND)} \times \frac{\mathbb{P}(X,Y|D)}{\mathbb{P}(X,Y|ND)} \tag{1.15}$$

**Proposition 1.1.** *Si les variables $X$ et $Y$ sont indépendantes, alors la côte devient :*

$$o(D|X,Y) = o_{pop} \times \frac{\mathbb{P}(X|D)}{\mathbb{P}(X|ND)} \times \frac{\mathbb{P}(Y|D)}{\mathbb{P}(Y|ND)} \tag{1.16}$$

En passant au logarithme, on voit que le score se décompose en trois termes :

$$s(X,Y) = s_{\text{pop}} + \omega_1(X) + \omega_2(Y) \tag{1.17}$$

— Un premier terme lié au portefeuille, $s_{\text{pop}}$, indépendant de $X$ et de $Y$ ;
— un deuxième terme, $\omega_1(X)$, dépendant de $X$ uniquement (ici représente le statut)
— et un troisième terme, $\omega_1(Y)$, dépendant de $Y$ uniquement (ici représente l'âge).

**Exemple 1.8.** *Exemple de la banque De Vinci*

Nous avons déja calculé les trois composantes du score :

— composante liée au portefeuille : $s_{\text{pop}} = -2.10$
— composante liée au statut résidentiel : $\omega_1(\text{Propriétaire}) = -0.89$
— composante liée à l'âge : $\omega_2(< 30 \text{ ans}) = 0.21$.

Le score est égal à :

$$\begin{aligned} s(\text{Propriétaire}, < 30 \text{ ans}) &= s_{\text{pop}} + \omega_1(\text{Propriétaire}) + \omega_2(< 30 \text{ ans}) \\ &= -2.10 - 0.89 + 0.21 = -2.78 \end{aligned}$$

Ce score bayésien est appelé « naïf » car il s'obtient en supposant les features indépendantes entre elles, ce qui est une approximation généralement simplificatrice mais fausse, et se calcule par simple comptage des populations vers les différentes sous catégories d'emprunteurs.

Il y a un lien entre la probabilité de défaut sachant les features et le score bayésien. Ainsi, la probabilité de défaut pour les propriétaires de moins de 30 ans est donc :

$$\mathbb{P}(\text{Propriétaire}, < 30 \text{ ans}) = \frac{1}{1 + \mathrm{e}^{-s(\text{Propriétaire}, < 30 \text{ ans})}} = \frac{1}{1 + \mathrm{e}^{2.78}} = 0.0584$$

**Propriété 1.4** (Convention sur les scores)**.**
— *Il faut noter que la probabilité de défaut est une fonction croissante du score :* $\dfrac{\partial}{\partial s} PD(s) > 0$.
— *Si on pose* $\overline{s} = -s$*, alors* $\overline{s}$ *est aussi un score et la fonction* $PD(\overline{s})$ *est décroissante.*
— *Selon la convention retenue, la probabilité de défaut est croissante ou décroissante en fonction du score. Il s'agit d'une convention qui n'a pas d'impact sur le résultat.*
— *Ce qui est important, c'est que la probabilité de défaut soit une* ***fonction monotone du score****.*

**En résumé**

— Un score doit être une fonction monotone de la probabilité de défaut (PD)
— Le logarithme de la côte a les propriétés d'un score. Lorsque les features du modèle sont indépendantes, cette fonction sépare les composantes systématiques et spécifiques à chaque feature et permet de calculer la contribution de chacune des features au score global.
— Un modèle est nécessaire pour établir un score lorsque les features sont en dimension élevée.

## 1.4 Bases de la construction des scores

### 1.4.1 Principes de construction

En effet, le score bayésien naïf a plusieurs problèmes. Tout d'abord, le problème d'indépendance des variables dans l'approche bayésienne. Deuxièmement, le problème de la dimension des variables explicatives dans l'approche fréquentiste (par comptage). C'est pour cela que l'approche modelisée, la construction d'un modèle de score, est indispensable.

Pour rappel, en dimension $10$ c'est - à - dire avec $10$ features, et chacune prenant $5$ valeurs ou modalités, il y a $5^{10}$ combinaisons de ces modalités pour lesquelles nous devons estimer la probabilité de défaut. Ceci n'est pas à la portée de l'approche bayésienne naïve qui repose sur une approche par simple comptage des événements observés. Comment y procéder ?

Tout d'abord la méthode de classification car un score a pour objectif de classifier les emprunteurs par ordre de risque croissante. Deuxièmement, la nécessité d'avoir des bases de données riches et de bonne qualité. Troisièmement, la nécessité d'avoir un nombre suffisant de données caractéristiques de la qualité de crédit des emprunteurs ("features").

### 1.4.2 Score comportemental

Pour évaluer la qualité de crédit des particuliers (des emprunteurs particuliers), on utilise souvent la notion de score comportemental.

**Définition 1.5.** *Un **score comportemental** a des features basés sur le comportement de l'emprunteur.*

Les données disponibles pour mesurer le risque de crédit des clients particuliers sont nombreuses et variées. Tout d'abord, **les données comportementales** telles que le niveau moyen du compte courant, le nombre de retards de paiement observé sur le compte dans les $12$ ou les $24$ derniers mois ou encore l'ampleur des retards de paiements, etc. Deuxièmement, **les données socio - démographiques** comme l'âge, le sexe, la situation matrimoniale, la profession etc. Enfin, **les données financières** telles que les revenus, les charges, les ratios d'endettement, etc. A noter que pour les entreprises (les corporates), les institutions financières, les données comportementales ne sont pas pertinentes. Elles ne le sont que pour les particuliers et pour les toutes-petites entrepises.

**Exemple 1.9.** *Exemple de variables comportementales*

- Solde moyen du compte
- Ratio du solde du compte fin de période / début de période c'est - à - dire la variation du niveau de compte moyen entre le début et la fin du mois, c'est aussi la volatilité du solde du compte.
- Prêts déja accordés : est - ce que l'emprunteur a déja des crédits revolvings [2] ?
- Fréquence et ampleurs des découverts
- Retards de paiements

2. Crédit revolving, crédit permanent ou crédit reconstituable, toutes ces expressions désignent le crédit renouvelable. Il s'agit d'un crédit à la consommation caractérisé par une grande liberté d'utilisation et de remboursement.

— Emprunteur est ou a été en retard de paiement sur les 12 ou 24 derniers mois. Combien de temps ont duré ces retards de paiements ? Et combien de temps l'emprunteur a t-il ramené son compte à la normale ?

Voici autant de variables comportementales qui sont pertinentes dans les modèles de score pour les particuliers et pour les petites entreprises.

### 1.4.3 Segmentation

La première étape dans la construction d'un score est d'identifier les différents segments sur lesquels on construit ces scores. En effet, il peut y avoir des cas de sous - portefeuilles qui sont suffisamment hétérogènes pour qu'il soit indispensable de faire des scores différents sur ces deux sous - populations. Par exemple, s'il existe une différence d'information disponible entre deux sous - portefeuilles. Pour illustrer cela, les jeunes (les moins de 25 ans en général) ont un historique bancaire très limité et pour lequel l'historique de remboursement de crédit n'est pas une variable pertinente. Alors qu'il l'est pour les emprunteurs sur lesquels on a une relation qui est plus longue dans la durée. Il peut être pertinent de faire un modèle de score dédié aux jeunes.

Deuxièmement, les différences de politique commerciale entre deux types de populations. Par exemple, une politique orientée vers les jeunes nécessitera la construction d'un score spécifique par rapport aux autres populations car les conditions d'acceptation de crédit ne seront pas les mêmes sur les deux populations.

Troisièmement, des segments différents peuvent avoir des features différentes ou des sensibilités différentes aux mêmes features.

### 1.4.4 Features du modèle

Le nombre de features est en général réduit et compris généralement entre 10 et 20 fetaures. Plus on a de features et plus elles sont corrélées. Ces corrélations pouvant d'ailleurs être fortuites ou instables.

Tout d'abord, il faut sélectionner les features dans l'univers des variables disponibles et qui peuvent expliquer le risque. D'abord, il y a l'évaluation de la dépendance de chacune des features avec les observations du défaut (D/ND). Il faut répondre à la question : y a - t - il une corrélation visible entre l'événement de défaut/survie et la valeur de chacune des features ? Ensuite, il y a des procédures statistiques de sélection des features (forward vs. backward). Ces procédures pouvant faire l'objet à elles seules d'un cours dédié. Enfin, une autre méthode efficace pour sélectionner des features consiste à développer des scores en mode "boîte noire" avec un certain type d'algorithme type réseaux de neurones et à sélectionner les features qui ressortent le plus dans ces modèles pour ensuite développer le score lui - même selon une approche logistique par exemple, afin de conserver une meilleure explicabilité des résultats de ce score.

En résumé

— Le score permet de calculer les probabilités de défaut des emprunteurs en fonction de leurs caractéristiques propres
— La qualité des segmentations va permettre de réaliser des scores ayant une meilleure performance sur chacun des sous-segments. On rappelle que ces segmentations sont nécessaires car il peut y avoir des sous-portefeuilles ou portefeuilles que l'on considère suffisamment hétérogènes pour nécessiter la modélisation par deux scores différents
— La dimension du vecteur de features n'est pas un obstacle à la construction des scores. Nous pouvons très bien faire un score logistique, un score par réseaux de neurônes, par méthode d'arbre avec des centaines de features. Cependant l'interprétation du score par les analystes crédit est un élément important dans l'utilisation du score et dans son implémentation. C'est le pourquoi on réduit en général la dimension, c'est - à - dire le nombre de features que l'on intègre dans un modèle de score.

## 1.5 Distribution des scores

### 1.5.1 Notations

Introduisons un certain nombre de notations. Tout d'abord celle de la fonction de repartition d'une variable aléatoire G.

**Définition 1.6.** *Un fonction de répartition G est croissante :* $G(-\infty) = 0$ *et* $G(+\infty) = 1$.

**Proposition 1.2.** *Pour une fonction de distribution G, la fonction* $G^{-1}$ *désigne son inverse ou sa fonction quantile.*

La densité de la loi normale standard est donnée par :

$$n(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-x^2/2\right) \tag{1.18}$$

et la fonction de répartition normale standard est :

$$N(x) = \int_{-\infty}^{x} du\, n(u) \tag{1.19}$$

où $n(u)$ est la fonction de densité de la loi normale centrée réduite.

Nous introduisons également un couple de variables aléatoires $(S, Z)$. $S$ est la valeur du score. C'est une variable continue dans le cadre d'un modèle de score et discrète si elle correspond à un rating puisqu'elle ne prend qu'un nombre fini de valeurs. Par convention, les faibles valeurs de $S$ correspondent à une mauvaise qualité de crédit (B, *Bad*) et les valeurs élevées de S à une bonne qualité de crédit de l'emprunteur (G, *Good*).

La variable Z désigne l'état de l'emprunteur à la fin de la phase d'observation :

— $Z = 0$ signifie que l'emprunteur reste solvable et est en survie.
— $Z = 1$ signifie que l'emprunteur est en défaut de crédit.

Donc le défaut D correspond à l'événement $Z = 1$ et la survie ND à l'événement $Z = 0$. on a :

— D = $\{Z = 1\}$
— ND = $\{Z = 0\}$

Enfin, la distribution marginale de la variable de Bernoulli Z est caractérisée par le paramètre $p$ qui est la probabilité de défaut :

$$p = \mathbb{P}(D) = \mathbb{P}(Z = 1)$$

### 1.5.2 Distribution conditionnelle du score

Notons par $S_{ND}$ et $S_D$ les variables aléatoires ayant les fonctions de distribution $F_{ND}(.)$ et $F_D(.)$ respectivement, c'est - à - dire :

$$F_{ND}(s) = \mathbb{P}(S \leq s|ND) \quad \text{et} \quad F_D(s) = \mathbb{P}(S \leq s|D)$$

Les variables $S_D$ et $S_{ND}$ représentent le score conditionnel au défaut et à la survie respectivement.

```
# Courbe de densité conditionnelle
import matplotlib.pyplot as plt
import scipy.stats as st
import numpy as np
import pandas as pd

mu_D = 5
sigma_D = 1.96
mu_ND = 8.5
sigma_ND = 2
p = 0.3
pas = np.linspace(0,15,50)

# Densité conditionnelle
F_D = np.array([st.norm.pdf(x,loc=mu_D,scale=sigma_D) for x in pas])
F_ND = np.array([st.norm.pdf(x,loc=mu_ND,scale=sigma_ND) for x in pas])

# Loi conditionnelle du score
fig, axe =plt.subplots(figsize=(16,6))
axe.plot(pas,(1-p)*F_ND, color = "green",label=r"$(1-p)F_{ND}(s)$");
axe.fill_between(pas,(1-p)*F_ND,color="green",alpha=0.2);
axe.text(x=10,y=0.13,s="$F^{'}_{ND}(s)$",color = "green",fontsize=14);
axe.plot(pas,p*F_D, color = "red",label=r"$p F_{D}(s)$");
axe.fill_between(pas, p*F_D,color = "red",alpha=0.2);
axe.text(x=2,y=0.04,s=r"$F^{'}_{D}(s)$",color = "red",fontsize=14);
axe.set(xlabel="Score",ylabel="Densité de probabilité")
axe.text(x=8.5,y=0.10,s="$1-p$",color="green",fontsize=14);
axe.text(x=5,y=0.04,s="$p$",color="red",fontsize=14);
plt.show()
```

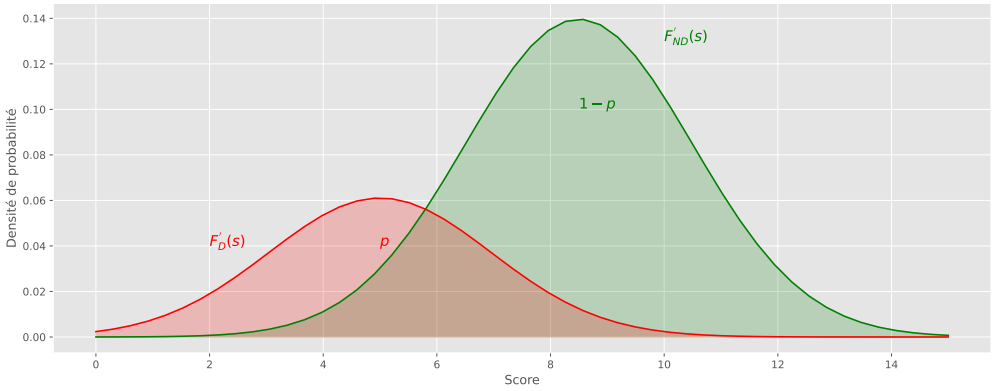


**Figure 1.1 –** Courbes des densités conditionnelles

La figure 1.1 représente les courbes de densité des variables aléatoires $S_D$ et $S_{ND}$. En vert, nous avons la fonction de densité du score sachant survie $F_{ND}$ et en rouge la fonction de densité du score sachant défaut $F_D$. Neanmoins, nous observons que ces deux fonctions ont été pondérées : nous avons en réalité représenté $1-p$ fois la densité de $S_{ND}$, c'est - à - dire le taux de survie du portefeuille fois la fonction de densité des emprunteurs en survie. Et ici en rouge, $p$ fois la densité de $S_D$ de sorte que la surface interceptée par la courbe rouge est égale à la probabilité de défaut $p$. La surface interceptée par la courbe verte est égale à la probabilité de survie $1-p$. Et la surface interceptée par la courbe rouge est égale à $p$.

**Exemple 1.10.** *Exemple pour des scores normalement distribués*

Nous prendrons régulèrement l'exemple des scores distribués normalement pour illustrer notre propos c'est - à - dire le cas où les variables conditionnelles de score $S_D$ et $S_{ND}$ suivent des lois normales. Nous supposons que :

$$\begin{cases} S_D \sim \mathcal{N}(\mu_D, \sigma_D^2) \\ S_{ND} \sim \mathcal{N}(\mu_{ND}, \sigma_{ND}^2) \end{cases}$$

Pour les applications numériques, nous retiendrons les paramètres suivants :

$$\begin{cases} \mu_D = 5 \\ \sigma_D = 1.96 \end{cases} \quad \text{et} \quad \begin{cases} \mu_{ND} = 8.5 \\ \sigma_{ND} = 2 \end{cases}$$

La probabilité de défaut inconditionnelle est choisie à $p = 30\%$.

```
# Définition des variables
mu_D = 5
sigma_D = 1.96
mu_ND = 8.5
sigma_ND = 2
p = 0.3
```

**Loi conditionnelle du score**

$$F(s) = pF_D(s) + (1-p)F_{ND}(s)$$

où $F_j^{'}(s) = f_j(s) = \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{s-\mu_j}{\sigma_j}\right)^2\right\}$ avec $j = D, ND.$

```
# Pas - sequence
pas = np.linspace(start=0,stop=15,num=50)

# Densité conditionnelle
f_D = np.array(list(map(lambda x:st.norm.pdf(x=x,loc=mu_D,scale=sigma_D),pas)))
f_ND = np.array(list(map(lambda x:st.norm.pdf(x=x,loc=mu_ND,scale=sigma_ND),pas)))

# Loi conditionnelle du score
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(pas,p*f_D, color = "red",label=r"$p. F_{D}^{'}(s)$");
axe.plot(pas,(1-p)*f_ND,color = "green", label = r"$(1-p).F_{ND}^{'}(s)$");
axe.plot(pas,p*f_D + (1-p)*f_ND, color = "blue",label = r"$F^{'}(s)$");
axe.set(xlabel="Score",ylabel="Densité de probabilité");
axe.legend();
plt.show()
```

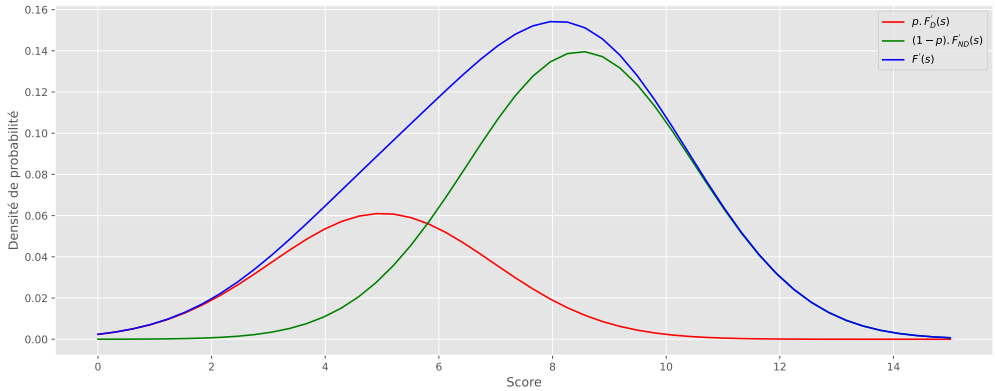


**Figure 1.2 –** Loi conditionnelle du score

Nous représentons les deux courbes, fonction de densité des scores conditionnelles, $S_D$ et $S_{ND}$ (figure 1.2 avec en rouge pour $S_D$ et en vert pour $S_{ND}$). Comme pour la figure 1.1, chacune est pondérée par le taux de défaut et le taux de survie respectivement $p$ et $1-p$. La somme de ces deux courbes nous donne la courbe en bleu, qui est la fonction de distribution inconditionnelle du score sur l'ensemble du portefeuille et que nous notons $F(s)$.

$$F(s) = pF_D(s) + (1-p)F_{ND}(s)$$

A noter que la fonction de répartition $F(s)$ est la somme des deux fonctions de répartition gaussienne, elle - même n'est pas gaussienne.

**Remarque 1.1.** *Rappelons que la somme de deux variables gaussiennes est une gaussienne, mais la somme des fonctions des répartitions des fonctions de densités gaussiennes n'est pas gaussienne.*

Cela se voit sur le caractère asymétrique de la courbe bleue qui a une asymétrie sur la gauche ici.

### 1.5.3 Probabilités de défaut conditionnelles

Par le théorème de Bayes, nous calculons la probabilité de défaut sachant le score. C'est la probabilité que la variable $Z$ prenne la valeur 1 en fonction du score :

$$\mathbb{P}(Z|S) = \frac{\mathbb{P}(Z \cap S)}{\mathbb{P}(S)} = \frac{\mathbb{P}(Z) \times \mathbb{P}(S|Z)}{\mathbb{P}(S)} = \frac{p \times \mathbb{P}(S|Z)}{\mathbb{P}(S)} \tag{1.20}$$

Ainsi, la probabilité de défaut conditionnelle pour l'événement $D = \{Z = 1\}$ sachant le score $S$ s'exprime comme suit :

$$\mathbb{P}(D|S = s) = \frac{p \times f_D(s)}{pf_D(s) + (1-p)f_{ND}(s)} \tag{1.21}$$

Ici les fonctions $f_D$ et $f_{ND}$ indiquent les densités du score conditionnel à $D$ et $ND$ respectivement.

**Exemple 1.11.** *Probabilités de défaut conditionnelles*

Nous représentons graphiquement cette fonction :

$$\mathbb{P}(D|S = s) = \frac{p \times f_D(s)}{pf_D(s) + (1-p)f_{ND}(s)}$$

```
# Probabilités de défaut conditionnelles
from plotnine import *
df = pd.DataFrame({"pas" : pas, "function" : p*f_D/(p*f_D+(1-p)*f_ND)})
p1 = (ggplot(df,aes(x="pas",y="function"))+geom_line(color="red")+
      annotate("text",x=[1,1.155,1.71],y=[0.9,0.825,0.75],
               label=["$s \\rightarrow -\infty$",
                      "$f_{ND}(s)\\rightarrow 0$",
                      "$P(D|S=s)=1$"],color="red")+
      annotate("text",x=[12.5,12.7,13.3],y=[0.2,0.125,0.05],
               label=["$s \\rightarrow +\infty$",
                      "$f_{ND}(s)\\rightarrow 1$",
                      "$P(D|S=s)=0$"],color="red")+
      labs(x="Score",y="Probabilité de défaut"))
print(p1)
```

Lorsque $s$ tend vers $-\infty$, cela signifie que le score correspond à des emprunteurs très risqués, c'est - à - dire que la densité de probabilité conditionelle, $f_{ND}(s)$, est proche de 0. Ainsi, le ratio tend vers 1, puisque les termes dominants au numérateur et au dénominateur sont égaux. Inversement, lorsque le score devient très grand, les

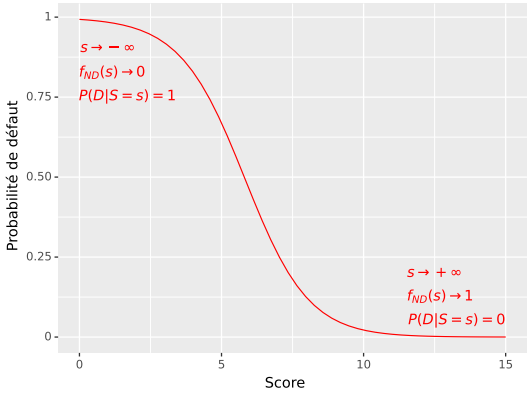


**Figure 1.3 –** Probabilités de défaut conditionnelles

probabilités conditionnelles du score sachant défaut tendent vers $0$, c'est - à - dire que le numérateur tend vers $0$ et le dénominateur est dominé par le terme de survie de sorte que la probabilité de défaut sachant le score tend vers $0$ lorsque le score $s$ tend vers $+\infty$.

**En résumé**

— Si les distributions conditionnelles des scores pour les emprunteurs en défaut (D) et en survie (ND) sont identiques, cela signifie que le modèle de score est aléatoire. Il est impossible de discriminer les emprunteurs qui vont faire défaut des emprunteurs qui font rester en survie. Et donc, les distributions conditionnelles seront identiques.
— Si les distributions conditionnelles sont parfaitement disjointes, alors le modèle est parfait. Il sera très facile de discriminer ceux qui irons au défaut de ceux qui resteront en survie, simplement en connaissant la valeur de leur score.
— Enfin, le modèle sera d'autant plus performant que les distributions conditionnelles seront séparées.

Donc, l'objectif dans le développement des modèles de score, c'est de faire en sorte que les distributions conditionnelles soient suffisamment séparées pour pouvoir discriminer les bons emprunteurs des mauvais emprunteurs.

## 1.6 Choix du seuil de prédiction

Nous avons vu dans la section précédente que les distributions conditionnelles de score étaient des éléments importants pour évaluer la performance d'un score. Nous allons dans cette section montrer comment nous pouvons faire des prédictions avec un modèle de score.

### 1.6.1 Généralités sur le choix du seuil de prédiction

Les prédictions sont faites à l'aide d'un seuil appelé **seuil de prédiction** : lorsque le score est inférieur au seuil de prédicion, nous classons l'emprunteur en B (*bad*) c'est -

à - dire que nous prédisons qu'il va aller au défaut. A l'inverse, si le score est supérieur au seuil de prédiction, nous prédisons la survie de l'emprunteur et nous le classons en G (*good*). Une autre application de seuil de prédiction concerne le choix du seuil dans l'octroi de crédit : à quel niveau de score décide - t - on d'accorder un crédit à un client et au-dessous duquel lui refuse - t - on le crédit ? Le seuil dans ce cas s'appelle un **cut - off**. Le seuil de prédiction est également un élément important pour la mesure et le suivi des risques des clients et des portefeuilles.

Concernant l'analyse des mauvais classements, l'identification des types de clients qui ressortent mal classés est une analyse importante à conduire. De même que l'identification des features qui aideraient à améliorer le classement ainsi que la comparaison des taux d'erreurs en fonction du seuil retenu. Ce sont autant d'éléments d'analyse qui seront utilisés par les modélisateurs et les analystes de crédit au quotidien pour suivre, gérer, piloter les risques de crédit de la banque.

Nous tirons votre attention sur les échantillons qui sont très déséquilibrés c'est - à - dire lorsque le taux de défaut est très faible par rapport au taux de survie (exemple des déclenchements d'alarme intempestifs). Dans ce cas, une analyse statistique dédiée est indispensable et nécessite de tenir compte de la rareté des événements des défauts.

### 1.6.2 Types d'erreurs possibles

Lorsqu'on fait de la prédiction, il y a deux types d'erreurs possibles que nous représentons dans une matrice appelée **matrice de contingence** (tableau 1.3).

**Table 1.3 –** Matrice de contingence

| Prédiction | Défaut (D) | Non défaut (ND) | Total |
|---|---|---|---|
| Bad (B) | VP | FP | Nb B |
| Good (G) | FN | VN | Nb G |
| **Total** | **Nb Défauts** | **Nb Survivants** | **Nb total** |

Par exemple, lorsque les emprunteurs sont dans la catégorie B (les mauvais emprunteurs), la prédiction est qu'ils vont allés au défaut (D). Le nombre d'emprunteurs classés B qui vont au défaut s'appelle **les vrais positifs** : c'est le nombre qui figure dans la première ligne et la première colonne du tableau 1.3. En revanche, ceux qui ne vont pas au défaut et qui ont été classés dans la catégorie B sont appelés **les faux positifs** : ligne un - colonne deux. Le total des vrais positifs et des faux positifs est égal au nombre d'emprunteurs classés B : ligne un - dernière colonne.

De même, pour les emprunteurs classés G, nous prédisons la survie (ND) : cela correspond au nombre de **vrais négatifs** (ligne deux - colonne deux). En revanche, ceux qui vont au défaut (D) alors qu'ils ont été classés dans la catégorie des bons emprunteurs sont ceux qu'on appelle **les faux négatifs** (ligne deux - colonne un). Le total des faux négatifs (FN) et des vrais négatifs représente le nombre total d'emprunteurs classés G (ligne deux - dernière colonne). Le total des vrais positifs (VP) et des faux négatifs (FN) représente le nombre total de défaut (D). Le total des faux positifs (FP) et des vrais négatifs (VN) représente le nombre total de survie (ND).

### 1.6.3 Table de contingence en fonction des distributions des scores

Nous pouvons aussi représenter cette matrice de contingence sous la forme de proportion du nombre total (tableau 1.4)

**Table 1.4 –** Matrice de contingence (en % du nombre total)

| Prédiction | Défaut (D) | Non défaut (ND) | Total |
|---|---|---|---|
| Bad (G) | $p\mathbb{P}(B\|D)$ | $(1-p)\mathbb{P}(B\|ND)$ | $\mathbb{P}(B)$ |
| Good (G) | $p\mathbb{P}(G\|D)$ | $(1-p)\mathbb{P}(G\|ND)$ | $\mathbb{P}(G)$ |
| Total | $p$ | $1-p$ | 1 |

Qu'est - ce que la proportion des emprunteurs classés B et qui sont en défaut (D) ? Eh bien, c'est le taux de défaut $(p)$ multipliée par la probabilité d'être classé B sachant qu'on est en défaut D : $p\mathbb{P}(B|D)$. De même, la proportion des emprunteurs classés G qui vont au défaut est égale à la probabilité de défaut $(p)$ multipliée par la probabilité conditionnelle de G sachant D : $p\mathbb{P}(B|D)$. Il en est de même pour les emprunteurs classés B qui vont à la survie dont la proportion est égale à $(1-p)$ multipliée par la probabilité conditionnelle de B sachant ND. Enfin, la proportion des emprunteurs classés G et qui survivent est égale à $1-p$ (taux de survie) multiplié par la probabilité d'être classée G sachant qu'on est en survie.

La somme des probabilités sur la première ligne est égale à la probabilité d'être classé B. Sur la deuxième ligne, cette même somme est égale à la probabilité d'être classée G. Donc, le total, probabilité d'être classé B plus probabilité d'être classé G, est bien égal à $100\%$. Si on somme à présent ces probabilités sur les deux colonnes, la somme des deux probabilités dans la colonne des défauts (égale à la probabilité d'être en défaut qui vaut $p$), et de même la somme des probabilités dans la deuxième colonne (est égale à probabilité de survie $1-p$). On voit également que la somme de ces deux probabilités est bien égale à $100\%$.

Nous pouvons aisément exprimer cette matrice de contingence en fonction des distributions conditionnelles des scores. Appelons le seuil de prédiction ou cut - off $c$ et considérons que tous les emprunteurs ayant un score inférieur à $c$ seront classés B. Et tous les emprunteurs ayant un score supérieur à $c$ seront classés G. Ainsi, la probabilité conditionnelle $\mathbb{P}(B|D)$ se réécrit comme $f_D$ prise au seuil de prédiction $c$ $f_D(c)$. Il en est de même pour toutes les autres cellules de la matrice de prédiction.

**Table 1.5 –** Matrice de contingence (en fonction des densités conditionnelles)

| Prédiction | Défaut (D) | Non défaut (ND) | Total |
|---|---|---|---|
| Bad (G) | $pf_D(c)$ | $(1-p)f_D(c)$ | $\mathbb{P}(B)$ |
| Good (G) | $p(1-f_{ND}(c))$ | $(1-p)(1-f_{ND}(c))$ | $\mathbb{P}(G)$ |
| Total | $p$ | $1-p$ | 1 |

**Exemple 1.12.** *Exemple des scores normalement distribués*

Dans le graphique 1.4, est représentée, en vert, la densité du score conditionnellement à la survie, $F'_{ND}(s)$, pondérée par le taux de survie $1-p$. De même, en rouge, la densité de score conditionnellement au défaut, $F'_D(s)$, pondérée par le taux de défaut $p$ de sorte que l'aire interceptée sous la courbe rouge est égale à la probabilité de défaut $p$ du portefeuille et l'aire interceptée sous la courbe verte est égale à sa probabilité de survie $1-p$.

```
# Lois conditionnelles du score avec cut - off
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(pas,p*f_D, color = "red");
axe.plot(pas,(1-p)*f_ND,color = "green");
axe.text(1.8,0.04,r"$p.F^{'}_{D}(s)$",color="red");
axe.text(11,0.08,r"$(1-p).F^{'}_{ND}(s)$",color="green");
```

```
axe.axvline(x=6,color="black");
axe.axvline(x=8,color="blue");
axe.set(xlabel="Score",ylabel="Densité de probabilité")
axe.fill_between(pas, (1-p)*f_ND,color = "green",alpha=0.3);
axe.text(8,0.08,r"$1-p$",color="green");
axe.fill_between(pas, p*f_D,color = "red",alpha=0.3);
axe.text(4.5,0.04,r"$p$",color="red");
axe.text(5.65,-0.005,r"$c=6$",color="black");
plt.show()
```

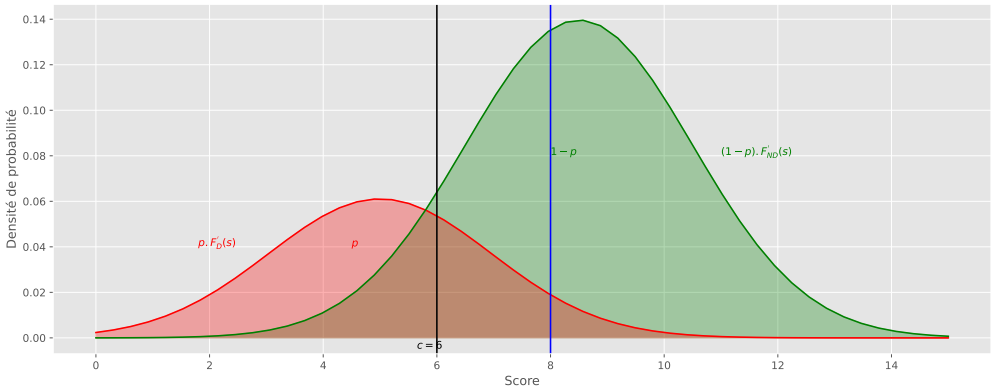


**Figure 1.4 –** Lois conditionnelles du score avec cut - off

Choisissons, à titre illustratif, un seuil égal à 6 $(c = 6)$, représenté graphiquement par la droite verticale noire. Nous classons en *bad* tous les emprunteurs qui ont un score inférieur à 6 et en *good* tous ceux qui ont un score supérieur à 6.

```
# Lois conditionnelles du score avec cut - off Sous Python
fig, axe = plt.subplots(figsize=(16,6));
axe.plot(pas,p*f_D, color = "red");
axe.plot(pas,(1-p)*f_ND,color = "green");
axe.text(1.8,0.04,r"$p.F^{'}_{D}(s)$",color="red");
axe.text(11,0.08,r"$(1-p).F^{'}_{ND}(s)$",color="green");
axe.axvline(x=6,color="black");
axe.axvline(x=8,color="blue");
axe.set(xlabel="Score",ylabel="Densité de probabilité")
axe.fill_between(pas, (1-p)*f_ND,color = "green",alpha=0.3);
axe.text(8,0.08,r"$1-p$",color="green");
axe.fill_between(pas, p*f_D,color = "red",alpha=0.3);
axe.text(4.5,0.04,r"$p$",color="red");
axe.text(5.65,-0.005,r"$c=6$",color="black");
axe.fill_betweenx([0,0.14],0,6,color='red', alpha=.4);
axe.fill_betweenx([0,0.14],6,15,color='green', alpha=.4);
axe.text(0,0.13,"B",color="red",fontsize=12,fontweight="bold");
axe.text(14.5,0.13,"G",color="green",fontsize=12,fontweight="bold");
plt.show()
```

Calculons les probabilités conditionnelles, par exemple la probabilité conditionnelle

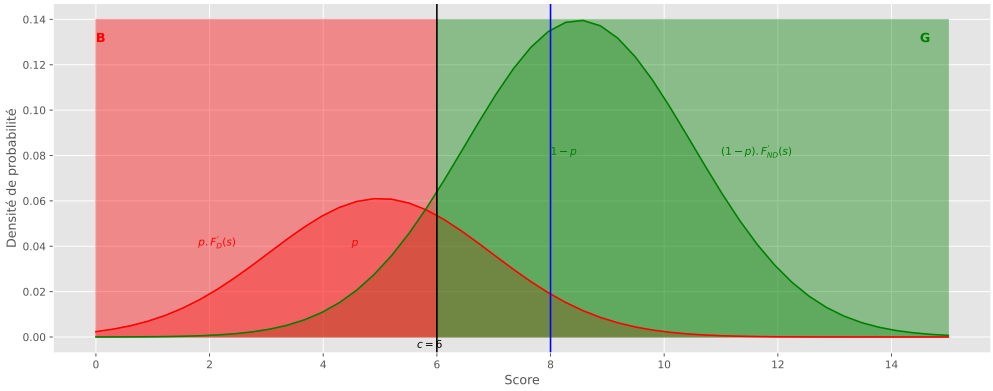


**Figure 1.5 –** Lois conditionnelles du score avec cut - off

d'être *bad* sachant le défaut, qui est l'aire interceptée sous la courbe rouge jusqu'au seuil de cut - off $c$.

```
# Lois conditionnelles du score avec cut - off
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(pas,p*f_D, color = "red");
axe.plot(pas,(1-p)*f_ND,color = "green");
axe.text(1.8,0.04,r"$p.F^{'}_{D}(s)$",color="red");
axe.text(11,0.08,r"$(1-p).F^{'}_{ND}(s)$",color="green");
axe.axvline(x=6,color="black");
axe.axvline(x=8,color="blue");
axe.set(xlabel="Score",ylabel="Densité de probabilité")
axe.text(5.65,-0.005,r"$c=6$",color="black");
axe.fill_between(pas,p*f_D,where=(pas<=6),color = 'red',alpha=0.3);
plt.show()
```

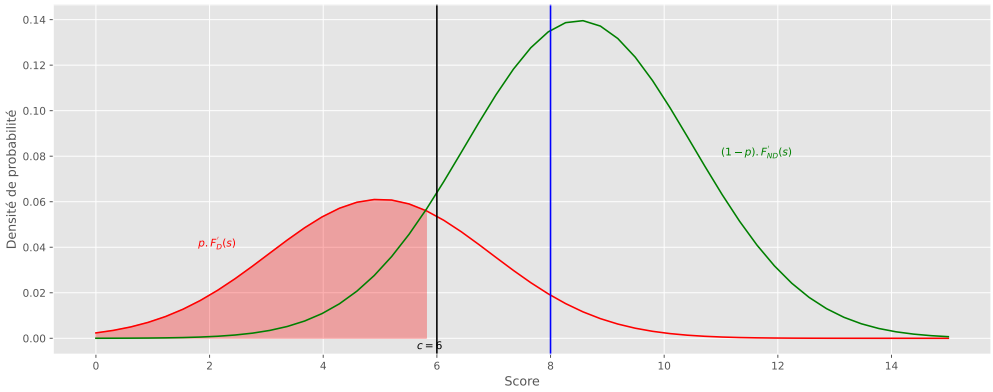


**Figure 1.6 –** Lois conditionnelles du score avec seuil de prédiction fixé à 6.

**Exemple 1.13.** *Exemple numérique pour différens seuils*

Si on suppose que les scores sont normalement distribués, nous pouvons calculer les matrices de contingence pour la valeur du seuil égale à 6 et ces matrices de contingence sont les suivantes :

**Table 1.6 –** Table de contingence pour un seuil égal à 6

| Prédiction | Défaut (D) | Non défaut (ND) | Total |
|---|---|---|---|
| Bad (B) | 0.2085137 | 0.0739548 | 0.2824686 |
| Good (G) | 0.0914863 | 0.6260452 | 0.7175314 |
| Total | 0.3000000 | 0.7000000 | 1.0000000 |

De même, si on change le seuil et qu'on le porte à $8$ $(c = 8)$.

```
# Lois conditionnelles du score avec cut - off
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(pas,p*f_D, color = "red");
axe.plot(pas,(1-p)*f_ND,color = "green");
axe.text(1.8,0.04,r"$p.F^{'}_{D}(s)$",color="red");
axe.text(11,0.08,r"$(1-p).F^{'}_{ND}(s)$",color="green");
axe.axvline(x=6,color="black");
axe.axvline(x=8,color="blue");
axe.set(xlabel="Score",ylabel="Densité de probabilité")
axe.text(7.6,-0.005,r"$c=8$",color="black");
axe.fill_between(pas,p*f_D,where=(pas<=8),color = 'red',alpha=0.3);
plt.show()
```

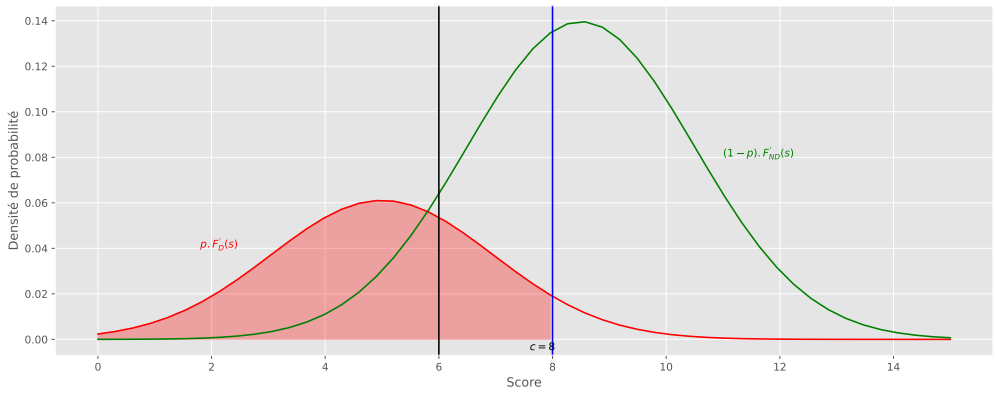


**Figure 1.7 –** Lois conditionnelles du score avec seuil de prédiction fixé à 8.

Nous voyons que l'aire interceptée sous la courbe rouge jusqu'au seuil (maintenant $8$), ici, va être plus importante, c'est - à - dire que la probabilité $p$ fois la probabilité d'être *bad* sachant le défaut va augmenter. C'est bien ce que nous observons (tableau 1.7) :

**Table 1.7 –** Table de contingence pour un seuil égal à 8

| Prédiction | Défaut (D) | Non défaut (ND) | Total |
|---|---|---|---|
| Bad (B) | 0.2811202 | 0.2809056 | 0.5620258 |
| Good (G) | 0.0188798 | 0.4190944 | 0.4379742 |
| Total | 0.3000000 | 0.7000000 | 1.0000000 |

La probabilité d'être *bad* sachant le défaut passe à $0.28$ alors qu'elle était à $0.21$ précédemment (tableau 1.6). Evidemment, la probabilité se somme en colonne pour donner, la probabilité de défaut totale, c'est - à - dire $0.3$ et la probabilité de survie totale, c'est - à - dire $0.7$. Et quel que soit le seuil, la somme de ces probabilités reste inchangée. En revanche, les probabilités d'être *bad* sont à $0.28$ pour un seuil égal à $6$, $0.56$ pour un seuil égal à $8$. Et donc ces probabilités changent en fonction du seuil de cut - off.

### 1.6.4 Indicateurs de performance en fonction du seuil

Nous introduisons à présent des indicateurs de performance qui vont dépendre du seuil.L'objectif es donc de déterminer de manière optimale le choix du seuil de cut - off $c$.

#### 1.6.4.1 Précision

Le premier indicateur est la **précision**, c'est - à - dire la proportion de B qui donnent la bonne prédiction (D). La précision est donc le ratio entre les vrais positis (VP) et le nombre total de B, c'est - à - dire les vrais positifs (VP) plus les faux positifs (FP) :

$$P = \frac{VP}{VP + FP} \tag{1.22}$$

#### 1.6.4.2 Recall

Le **recall** est le deuxième indicateur, appelé également **sensibilité**. C'est la proportion de défaut (D) correctement prédits. Le recall est donc le ratio entre les vrais positifs et le nombre total de défaut du portefeuille, c'est - à - dire les vrais positifs (VP) plus les faux négatifs (FN) :

$$R = \frac{VP}{VP + FN} \tag{1.23}$$

**Exemple 1.14.** *Indicateurs de performance en fonction du seuil*

Considérons notre exemple précédent avec les seuils $c = 6$ et $c = 8$. On a les résultats suivants :

**Table 1.8 –** Précision et recall en fonction du cut - off

| Indicateur | $c = 6$ | $c = 8$ |
|---|---|---|
| Précision (P) | 0.7381838 | 0.5001910 |
| Recall (R) | 0.6950458 | 0.9370674 |

Ce que nous observons c'est que lorsque le seuil augmente, la précision diminue mais la sensibilité s'améliore. Donc nous avons deux indicateurs de performance qui varient dans des sens opposés. C'est une règle général. Il est difficile de choisir un seuil qui va maximiser les deux indicateurs simultanément.

### 1.6.5 F - score

Pour déterminer un critère de choix pour le cut - off $c$, nous prenons un indicateur composite qui est appelé le **f - score**.

**Définition 1.7.** *Le **F - score** est la moyenne harmonique de la précision et de la sensibilité.*

$$F = 2\frac{P \times R}{P + R} \quad \longrightarrow \quad \frac{1}{F} = 2\left(\frac{1}{P} + \frac{1}{R}\right) \tag{1.24}$$

Il est composite car il combine la précision et le recall en une seule expression qui est une croissante de ces deux indicateurs.

**Exemple 1.15.** *Indicateurs de performance plus F - score*

Considérons notre exemple précédent avec les seuils $c = 6$ et $c = 8$. On a les résultats suivants :

**Table 1.9 –** Précision et recall en fonction du cut - off

| Indicateur | $c = 6$ | $c = 8$ |
|---|---|---|
| Précision (P) | 0.7381838 | 0.5001910 |
| Recall (R) | 0.6950458 | 0.9370674 |
| F - score | 0.7159656 | 0.6522316 |

Dans notre exemple, le F - score est meilleur pour un score de $6$ que pour un seuil de $8$. Donc un seuil égal à $6$, en terme de performance, est plus adapté à la prédiction qu'un seuil égal à $8$.

### 1.6.6 Formules analytiques dans le cas des scores normalement distribués

La précision, le recall ainsi que le F - score peuvent être calculés analytiquement lorsqu'on connaît la distribution, la forme analytique des distributions conditionnelles du score $F_D(s)$ et $F_{ND}(s)$.

**Table 1.10 –** Formules analytiques dans le cas des scores normalement distributés

| Prédiction | Défaut (D) | Non défaut (ND) | Total |
|---|---|---|---|
| Bad (B) | $pF_D(c)$ | $(1-p)F_D(c)$ | $\mathbb{P}(B)$ |
| Good (G) | $p(1-F_D(c))$ | $(1-p)(1-F_D(c))$ | $\mathbb{P}(G)$ |
| Total | $p$ | $1-p$ | 1 |

En utilisant les formules contenues dans les cellules de la matrice de contingence (dans le cas de loi conditionnelle normale), nous obtenons une forme analytique pour la précision et pour la sensibilité en fonction du seuil de prédiction :

— **Précision** :

$$P(c) = \frac{pN\left(\frac{c-\mu_D}{\sigma_D}\right)}{pN\left(\frac{c-\mu_D}{\sigma_D}\right) + (1-p)N\left(\frac{c-\mu_{ND}}{\sigma_{ND}}\right)} \tag{1.25}$$

— **Recall**

$$R(c) = N\left(\frac{c-\mu_D}{\sigma_D}\right) \tag{1.26}$$

### 1.6.7 Courbes des indicateurs de performance

On peut tracer aisément ces courbes avec un logiciel de choix.

```
# Précision
def Precision(x,p,mu1,sigma1,mu2,sigma2):
   F1 = st.norm.cdf(x=x,loc=mu1,scale=sigma1)
   F2 = st.norm.cdf(x=x,loc=mu2,scale=sigma2)
   return p*F1/(p*F1+(1-p)*F2)

# Recall
def Recall(x,mu,sigma):
  return st.norm.cdf(x=x,loc=mu,scale=sigma)

# F - score
def Fscore(x,p,mu1,sigma1,mu2,sigma2):
  P = Precision(x,p,mu1,sigma1,mu2,sigma2); R = Recall(x,mu1,sigma1)
  return 2*P*R/(P+R)

# Application
precision = np.array(list(
  map(lambda x : Precision(x=x,p=p,
                        mu1=mu_D,sigma1=sigma_D,mu2=mu_ND,sigma2=sigma_ND),pas)))
recall = np.array(list(map(lambda x : Recall(x=x,mu=mu_D,sigma=sigma_D),pas)))
fscore = np.array(list(
  map(lambda x : Fscore(x=x,p=p,
                        mu1=mu_D,sigma1=sigma_D,mu2=mu_ND,sigma2=sigma_ND),pas)))

# Représentation graphique
data = pd.DataFrame({"pas":pas,"func1":precision,"func2":recall,"func3":fscore})
p2 = (ggplot(data)+geom_line(aes(x="pas",y="func1"),color="green")+
        geom_line(aes(x="pas",y="func2"),color="red")+
        geom_line(aes(x="pas",y="func3"),color="blue")+
        geom_vline(xintercept=6,color="black")+
        annotate("text",x=[6,0,14,14],y=[-0.005,0.95,0.95,0.5],
                  label = ["$c=6$","P(c)","R(c)","F(c)"],
                  color=["black","green","red","blue"])+
        labs(x="Seuil de score",y="Indicateur"))
print(p2)
```

Nous avons la précision, qui est la courbe en vert, en fonction du cut - off $c$ ; la sensibilité est la courbe en rouge $R(c)$. Ce que nous voyons c'est que le F - score, $F(c)$, est la courbe en bleu qui a un point maximum autour de $c$ égal 6. Donc, afin de maximiser au mieux les deux critères, de précision et de sensibilité, il fait se placer à l'optimum de la courbe du F - score en fonction du seuil de prédiction.

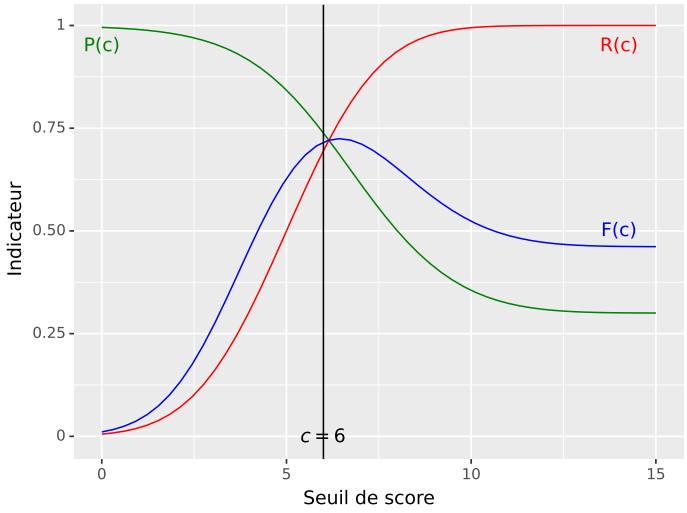


**Figure 1.8 –** Courbes des indicateurs de performance

En résumé

— Le score a permis de classer les emprunteurs en deux catégories : G (*good*) et B (*bad*).
— Ces deux catégories correspondent à une prédiction de l'état futur de l'emprunteur qui sera D ou ND.
— Un seuil de score (cut - off) est choisi pour déterminer la prédiction. La matrice de concordance permet de mesurer la précision des prédictions du score. Nous utilisons plusieurs indicateurs pour cela : la précision, la sensibilité ou recall et le F - score.
— Un modèle imparfait va conduire à deux types d'erreurs : les erreurs de type I (**faux positifs**) et les erreurs de types II (**faux négatifs**).

## 1.7 Pouvoir discriminant

Dans les deux sections précédentes, nous avons introduits les notions de distributions conditionnelles de score et de seuil de prédiction. Nous allons achever la présentation des concepts pour l'évaluation des modèles de score et introduire la notion de pouvoir discriminant, de courbe ROC et de courbe CAP.

### 1.7.1 Courbe ROC et courbe CAP

#### 1.7.1.1 Courbe ROC

La courbe des vrais positifs en fonction des faux négatifs est ce qu'on appelle la courbe ROC (*Receiver Operating Characteristic*). Il s'agit d'une courbe définie de manière paramétrique à partir du seuil de score $s$. C'est une courbe pour laquelle on représente en abscisse $F_{ND}(s)$ et en ordonnée $F_D(s)$.

$$gROC = \{(F_{ND}(s), F_D(s)) : s \in \mathbb{R} \cup \{\pm\infty\}\} \tag{1.27}$$

```
# Courbe ROC
fig, axe = plt.subplots(figsize=(16,6))
axe.set(xlabel="$F_{ND}(s)$",ylabel="$F_{D}(s)$")
plt.show()
```

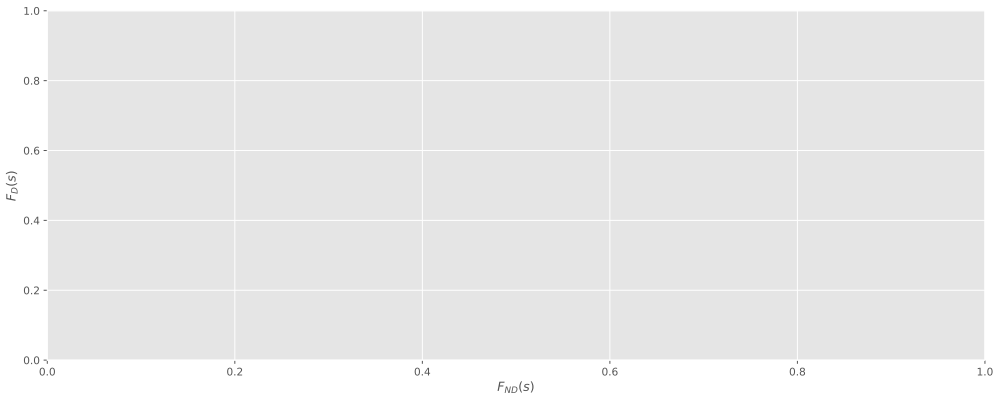


**Figure 1.9 –** Repère d'une courbe ROC

Cette courbe est définie de manière paramétrique, c'est - à - dire que l'abscisse $x$ des points de cette courbe ainsi que l'ordonnée $y$ sont chacun une fonction du score que nous faissons varier de $-\infty$ à $+\infty$.

#### 1.7.1.2 Courbe CAP

De même, la courbe CAP (*Cumulative Accuracy Profile*) ou courbe de Lorenz est une courbe paramétrique du score $s$ qui représente les vrais positifs au seuil $s$ en fonction de la taille du sous - portefeuille capté au seuil $s$. Il s'agit dionc d'une courbe qui représente en abscisse $F(s)$ et en ordonnée $F_D(s)$, c'est - à - dire la probabilité cumulée conditionnelle du score sachant le défaut.

$$gCAP = \{(F(s), F_D(s)) : s \in \mathbb{R} \cup \{\pm\infty\}\} \tag{1.28}$$

```
# Courbe CAP
fig, axe = plt.subplots(figsize=(16,6))
axe.set(xlabel="$F(s)$",ylabel="$F_{D}(s)$")
plt.show()
```

**Exemple 1.16.** *Exemple des fonctions de score continues*

Pour avancer, plaçons nous dans le cadre d'un score continu. Dans ce cas, les masses de probabilités que $S_{ND} = s$, que $S_D = s$ ou encore $S = s$ sont toutes égales à $0$ :

$$\mathbb{P}(S_{ND} = s) = \mathbb{P}(S_D = s) = \mathbb{P}(S = s) = 0 \tag{1.29}$$

Nous supposons également que :

$$F_D^{-1} \leq F_{ND}^{-1} \tag{1.30}$$

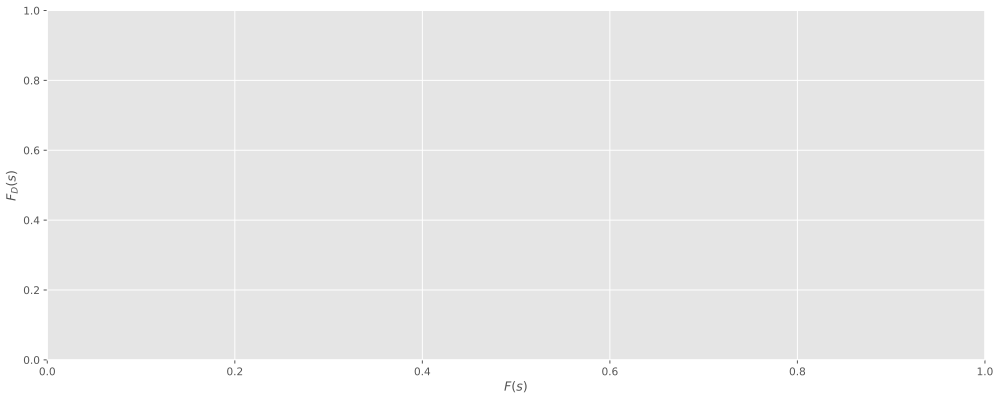


**Figure 1.10 –** Repère d'une courbe CAP

Alors nous déduisons une formule explicite de la fonction ROC et de la fonction CAP en fonction des distributions conditionnelles de score $F_D$ et $F_{ND}$ :

$$ROC(u) = F_D(F_{ND}^{-1}(u)) \tag{1.31}$$
$$CAP(u) = F_D(F^{-1}(u)) \tag{1.32}$$

**Propriété 1.5.** *1.* $CAP(0) = ROC(0) = 0$
*2.* $CAP(1) = ROC(1) = 1$

#### 1.7.1.3 Score avec un pouvoir discriminant nul

Le score ayant un pouvoir discriminant nul est tel que les distributions conditionnelles du score sachant défaut et sachant survie sont identiques c'est - à - dire qu'on ne peut les distinguer :

$$F_D(s) = F_{ND}(s) \tag{1.33}$$

Il est impossible dans ce cas de discriminer pour chaque valeur du score $s$ un emprunteur de bonne ou de mauvaise qualité de crédit. Alors dans ce cas, la fonction ROC et la fonction CAP sont égales à la fonction identité :

$$CAP(u) = ROC(u) = u \quad \text{(fonction identité)} \tag{1.34}$$

IL suffit d'injecter la condition d'égalité des distributions conditionnelles dans les définitions des fonctions ROC et CAP pour s'en assurer. En effet, en prenant $F_D(s) = F_{ND}(s)$, nous obtenons immédiatement que :

$$ROC(u) = F_D(F_D^{-1}(u)) = u \tag{1.35}$$

Pour la fonction CAP, il faut remarquer que $F_D(s) = F_{ND}(s)$, $\forall s$.

#### 1.7.1.4 Score parfait

Pour un score parfait, il existe une valeur du score $s_0$ qui sépare les supports des fonctions de répartition conditionnelles de défaut et de survie, c'est - à - dire que :

$$F_D(s_0) = 1 \quad \text{et} \quad F_{ND}(s_0) = 0 \tag{1.36}$$

Dans ce cas de figure, la fonction ROC vaut :

$$ROC(u) = \begin{cases} 0 & u = 0 \\ 1 & 0 < u \leq 1 \end{cases} \tag{1.37}$$

De même la fonction CAP vaut :

$$CAP(u) = \begin{cases} u/p & 0 < u < p \\ 1 & p \leq u \leq 1 \end{cases} \tag{1.38}$$

### 1.7.2 AUC et AR

Nous introduisons deux indicateurs de performance du modèle de score : AUC et AR

#### 1.7.2.1 AUC

L'AUC (*Area Under the Curve*) est l'aire interceptée par la courbe ROC, c'est - à - dire

$$AUC = \int_0^1 ROC(u)\mathrm{d}u \tag{1.39}$$

Plus le modèle de score est performant, plus l'aire interceptée par la courbe ROC va être élevée.

#### 1.7.2.2 AR

l'AR (*Accuracy Ratio*) est l'aire comprise entre la courbe CAP et la première diagonale, c'est - à - dire la droite d'équation $F_D(s) = F(s)$ divisée par l'aire comprise entre la courbe CAP du modèle parfait et la première diagonale.

$$AR = \frac{\displaystyle\int_0^1 (CAP(u) - u)\,\mathrm{d}u}{1 - p/2 - 1/2} = \frac{2\displaystyle\int_0^1 CAP(u)\mathrm{d}u - 1}{1 - p} \tag{1.40}$$

On montre aisément qu'il y a une relation linéaire entre ces deux indicateurs $AUC$ et $AR$.

**Propriété 1.6.** *On a la relation suivante :*

$$AR = 2AUC - 1 \tag{1.41}$$

**Proposition 1.3.** *Cas particuliers*
— *Modèle aléatoire :* $AUC = 1/2$ *et* $AR = 0$
— *Modèle parfait :* $AUC = AR = 1$

**Démonstration de** $AR = 2AUC - 1$

$$AUC = \int_0^1 ROC(u)\mathrm{d}u = \int_0^1 F_D\left(F_{ND}^{-1}(u)\right)\mathrm{d}u = \int_0^1 F_D(v) f_{ND}(v)\mathrm{d}v$$

$$\begin{aligned} AR &= \frac{2\displaystyle\int_0^1 CAP(u)\mathrm{d}u - 1}{1-p} = \frac{2\displaystyle\int_0^1 F_D\left(F^{-1}(u)\right)\mathrm{d}u - 1}{1-p} = \frac{2\displaystyle\int_0^1 F_D(v) f(v)\mathrm{d}v - 1}{1-p} \\ &= \frac{2\displaystyle\int_0^1 F_D(v)\left(p f_D(v) + (1-p) f_{ND}(v)\right)\mathrm{d}v - 1}{1-p} = \frac{p\left[F_D(v)\right]_0^1 + (1-p)AUC - 1}{1-p} \\ &= 2AUC - 1 \end{aligned}$$

#### 1.7.2.3 Courbe ROC dans le cas de scores normalement distribués

Lorsque les scores sont normalement distribués, la fonction ROC est une fonction simple des paramètres $\mu_D$, $\mu_{ND}$, $\sigma_D$ et $\sigma_{ND}$ et fait intervenir la fonction de répartition $N$ de la loi normale. Nous avons alors une forme analytique explicite pour la fonction ROC

$$ROC(u) = N\left(\frac{\sigma_{ND}\mathcal{N}^{-1}(u) + \mu_{ND} - \mu_D}{\sigma_D}\right) \tag{1.42}$$

De même, nous avons une forme analytique pour la valeur de l'AUC :

$$AUC = N\left(\frac{\mu_{ND} - \mu_D}{\sqrt{\sigma_{ND}^2 + \sigma_D^2}}\right) \tag{1.43}$$

**Exemple 1.17.** *Application*

```
# Valeur de l'AUC
auc = st.norm.cdf((mu_ND - mu_D)/np.sqrt(sigma_ND**2+sigma_D**2))
print(auc)
```

```
## 0.8943269309645037
```

La formule explicite de l'AUC s'obtient par intégration de la fonction ROC en opérant le changement de variable $x = \mathcal{N}^{-1}(u)$ puis en réalisant une intégration par parties.

```
# Courbe ROC
s = np.linspace(start=-15,stop=15,num=100)
F_D = np.array(list(map(lambda x : st.norm.cdf(x=x,loc=mu_D,scale=sigma_D),s)))
F_ND = np.array(list(map(lambda x : st.norm.cdf(x=x,loc=mu_ND,scale=sigma_ND),s)))

# Courbe ROC
x = np.linspace(0,1,100)
y = np.linspace(0,1,100)
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(F_ND, F_D,color ="red");
axe.plot(x,y,color="blue");
axe.set(xlabel="$F_{ND}(s)$",ylabel="$F_{D}(s)$")
plt.show()
```

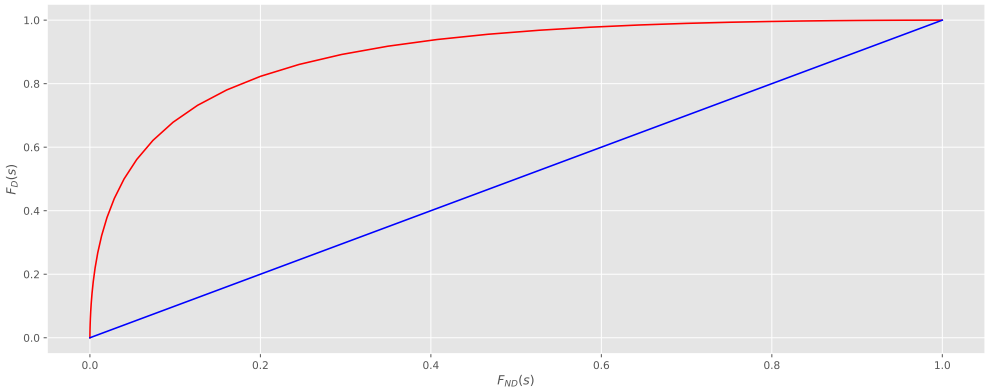


**Figure 1.11 –** Courbe ROC

En rouge, nous avons la courbe ROC dans le cas des scores normalement distribués et en bleu la ROC dans le cas du modèle aléatoire $y = x$.

```
# Courbe ROC
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(F_ND, F_D,color ="red");
axe.plot(x,y,color="blue");
axe.set(xlabel="$F_{ND}(s)$",ylabel="$F_{D}(s)$")
axe.fill_between(x,y,where=(x>=0),color="blue",alpha=0.3)
plt.show()
```

On voit que l'aire sous la courbe du modèle aléatoire (en bleu) et l'aire du triangle interceptée par la courbe bleue est égale à $1/2$ (figure 1.12). L'aire sous la courbe ROC est quant à elle supérieure à $1/2$ (figure 1.13).

```
# Courbe ROC
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(F_ND, F_D,color ="red");
axe.plot(x,y,color="blue");
axe.set(xlabel="$F_{ND}(s)$",ylabel="$F_{D}(s)$")
axe.fill_between(F_ND,F_D,color="red",alpha=0.3)
plt.show()
```

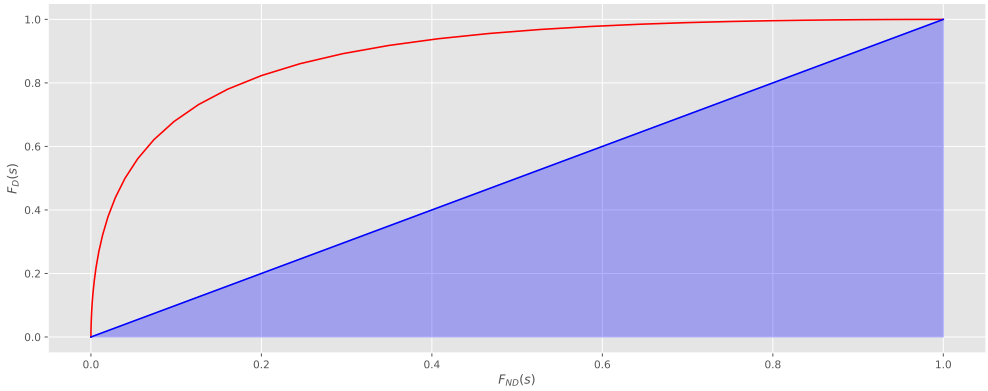


**Figure 1.12 –** Courbe ROC et aire sous la droite $y = x$

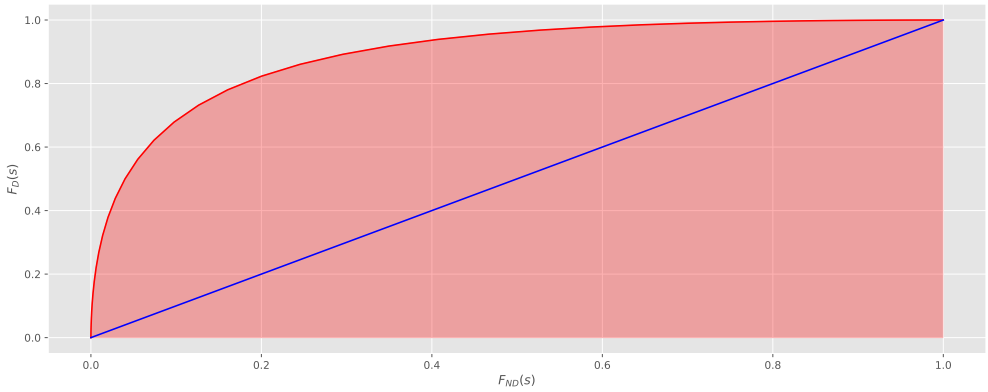


**Figure 1.13 –** Courbe ROC et aire sous la courbe ROC

Plus l'AUC est élevée et tend vers 1, plus le pouvoir discriminant du score est élevé.

#### 1.7.2.4 Courbe CAP dans le cas de scores normalement distribués

De même, nous avons une formule analytique pour la fonction CAP en fonction des paramètres du problème. La fonction CAP est une fonction de la fonction de répartition des scores $F$.

$$CAP(u) = N\left(\frac{F^{-1}(u) - \mu_D}{\sigma_D}\right) \tag{1.44}$$

On rappelle que dans le cas du scores normalement distribués, les scores eux - mêmes ne sont pas distribués normalement, mais sont un mélange de lois normales.

L'*Accuray Ratio* (AR) se calcule de manière explicite. Il est donné par cette formule :

$$AR = 2 \times \mathcal{N}\left(\frac{\mu_{ND} - \mu_D}{\sqrt{\sigma_{ND}^2 + \sigma_D^2}}\right) - 1 \tag{1.45}$$

Comme pour l'AUC, le calcul de l'AR se fait en faisant le changement de variables $x = F^{-1}(u)$, puis en intégrant par parties.

**Exemple 1.18.** *Calcul de l'AR dans le cas de scores normalement distribués*

```
# Valeur de l'AR
print((2*auc - 1))
```

```
## 0.7886538619290073
```

```
# Courbe CAP
def CAP(u,p):
  if (u>0) & (u<p) : return u/p
  if (u>=p) & (u<=1) : return 1

u = np.linspace(0,1,100)
cap = np.array(list(map(lambda x : CAP(u=x,p=0.3),u)))
F = p*F_D + (1-p)*F_ND
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(F, F_D, color = "red");
axe.plot(u, cap, color = "black");
axe.plot(x, y, color="blue");
axe.set(xlabel="$F(s)$",ylabel="$F_{D}(s)$")
axe.axvline(x=0.3,linestyle = "--", color="black");
axe.text(0.29,-0.005,"$30\%$",color="black");
plt.show()
```

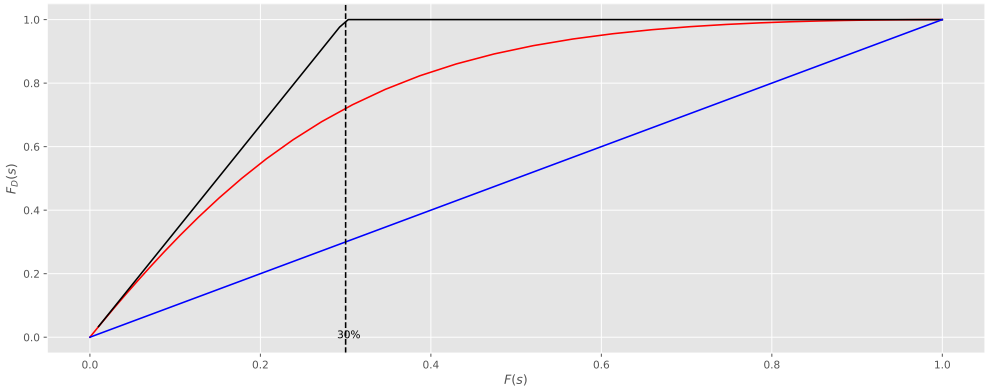


**Figure 1.14 –** CAP curve

Pour le modèle parfait (courbe noire), la courbe CAP est linéaire jusqu'à $u = 30\%$ et elle atteint pour cette valeur le niveau de $100\%$ : tous les défauts sont observés à ce seuil dans le modèle parfait. La courbe CAP est plate ensuite. Pour le modèle aléatoire, la courbe CAP, àl'instar de la courbe ROC est la droite d'équation $y = x$ représentée en bleue. Enfin, pour un modèle réaliste, la courbe rouge représente la fonction CAP.

```
# Représentation graphique
fig, axe = plt.subplots(figsize=(16,6))
axe.plot(F, F_D, color = "red");
axe.plot(u, cap, color = "black");
axe.plot(x, y, color="blue");
```

```
axe.set(xlabel="$F(s)$",ylabel="$F_{D}(s)$")
axe.axvline(x=0.3,linestyle = "--", color="black");
axe.text(0.29,-0.005,"$30\%$",color="black");
axe.fill_between(F,F_D,color="red",alpha=0.3);
plt.show()
```

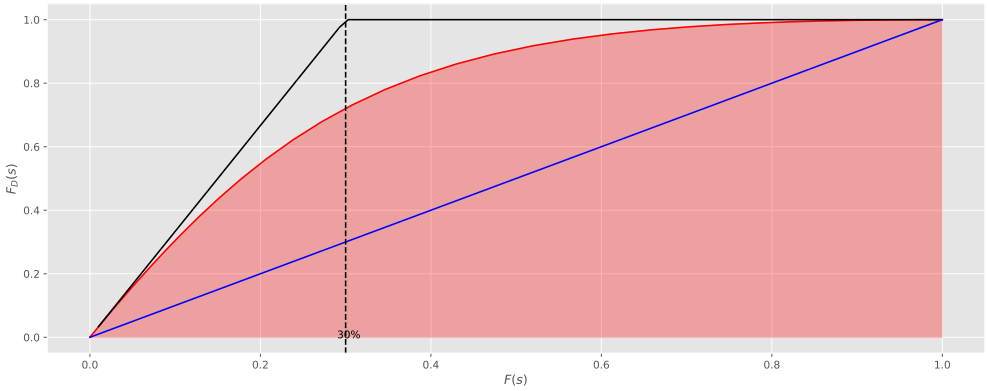


**Figure 1.15 –** Cap Curve

L'*Accuracy Ratio* se déduit de l'aire interceptée par la courbe rouge au dessus du modèle aléatoire. On voit dont que plus l'accuracy ratio est elevé, plus la courbe CAP se rapproche de la courbe CAP du modèle parfait. Donc l'accuracy ratio c'est l'aire interceptée ici divisée par l'aire totale sous la courbe du modèle parfait.

**En résumé**

- Les courbes ROC et CAP contiennent l'information détaillée de la performance d'un modèle de score
- Les indicateurs AUC et AR sont des indicateurs agrégés de la performance d'un modèle de score : plus ces indicateurs dont élevés, plus le modèle de score est discriminant.
- Ces deux indicateurs sont reliés entre eux par la formule : $AR = 2AUC - 1$.

# 2

# Evaluation des modèles de scoring

Ce chapitre a pour finalité de fire appliquer les notions vues au chapitre 1 afin de vous permettre de monter progressivement en compétences. Il porte sur l'évaluation d'un classifieur dans le cas d'un problème de classification binaire et sur la généralisation des métriques d'évaluation dans le cas multi - classes.

## 2.1 Quelques éléments de vocabulaire et métriques

Les termes **variable réponse**, **variable indépendante**, **label** font référence à la variable qu'on cherche à prédire

Les termes **probabilités prédites**, **output du modèle** sont les probabilités calculées en « sortie » du modèle

Le ***recall***, la ***sensitivity***, le **taux de vrais positifs** font référence à la même mesure de performance

La ***specificity***, le **taux de vrais positifs** font référence à la même mesure.

On trouve couramment les notations suivantes dans la littérature : **TP** = **T**rue **P**ositive, **TN** = **T**rue **N**egative, **FP** = **F**alse **P**ositive et **FN** = **F**alse **N**egative.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

$$\text{Misclassification rate} = \frac{FP + FN}{TP + TN + FP + FN} = 1 - \text{Accuracy} \tag{2.2}$$

$$\text{Specificity} = \frac{TN}{FP + TN} \tag{2.3}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.5}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{recall}}{\text{Precision} + \text{recall}} \tag{2.6}$$

## 2.2 Evaluation d'un classifieur dans le cas d'un problème de classification binaire

Dans cette section, nous considérons un cas de classification binaire ($Y = 0$ ou $Y = 1$ où $Y$ est la variable indicatrice que nous souhaitons prédire).

Les probabilités prédites sont données par la fonction logistique

$$\sigma(X) = \frac{1}{1 + \mathrm{e}^{-X}} \tag{2.7}$$

Nous disposons d'un jeu de données (fichier `Data_TD1_partie_1.csv`) contenant : la variable cible (variable « $Y$ ») qui peut prendre les valeurs $0$ et $1$ (on parle de **variable indicatrice** ou de **variable binaire**) et les probabilités prédites par une régression logistique (variable « proba ») pouvant prendre un continuum de valeurs dans $[0, 1]$.

**L'objectif est d'évaluer ce classifieur à l'aide des données qui nous sont fournies**

Pour cela, nous allons utiliser les métriques présentées ci - dessus :

### 2.2.1 Définissez une fonction "confusion_matrix"

Cette fonction calcule le nombre de vrai positifs, vrai négatifs, faux positifs et faux négatifs. Elle prend en arguments la variable réponse, la vecteur des probabilités prédictes, et valeur seuil.

```
# Matrice de confusion
import numpy as np
import pandas as pd
def confusion_matrix(y_true,y_prob,threshold = 0.5):
  y_true, y_prob = np.array(y_true), np.array(y_prob)
  proba_s=(y_prob > threshold)*1
  FP=np.sum((proba_s == 1)*(y_true == 0))
  TP=np.sum((proba_s == 1)*(y_true == 1))
  TN=np.sum((proba_s == 0)*(y_true == 0))
  FN=np.sum((proba_s == 0)*(y_true == 1))
  return {"TP" : TP,"FP" :FP,"TN" : TN, "FN" : FN}
```

### 2.2.2 Pour chaque métrique (accuracy, misclassification rate, sensitivity, specificity, precision, F1), créez une fonction qui prend les valeurs de la matrice de confusion en input et retourne la valeur de la métrique

```
# Métriques sous Python
def accuracy(conf_matrix):
  num = conf_matrix["TN"]+conf_matrix["TP"]
  den = conf_matrix["TN"]+conf_matrix["TP"]+conf_matrix["FN"]+conf_matrix["FP"]
  return num/den

def misclass_rate(conf_matrix):
```

```
  return 1 - accuracy(conf_matrix)

def specificity(conf_matrix):
  return conf_matrix["TN"]/(conf_matrix["TN"]+conf_matrix["FP"])

def recall(conf_matrix):
  return conf_matrix["TP"]/(conf_matrix["TP"]+conf_matrix["FN"])

def precision(conf_matrix):
  return conf_matrix["TP"]/(conf_matrix["TP"]+conf_matrix["FP"])

def f1(conf_matrix):
  recall_val = recall(conf_matrix)
  precision_val = precision(conf_matrix)
  return 2*recall_val*precision_val/(recall_val+precision_val)
```

### 2.2.3 Runnez les fonctions définies précédemment avec un seuil de $0.5$ sur les données qui vous sont fournies. Pourquoi avons nous utilisé le seuil de $0.5$ ? Que pensez vous de la qualité du classifieur ?

```
# Chargement des données
import pandas as pd
Data1 = pd.read_csv("./donnee/data_td1_partie1.csv",sep=";",decimal=',')
Data1.info()
```

```
## <class 'pandas.core.frame.DataFrame'>
## RangeIndex: 1000 entries, 0 to 999
## Data columns (total 2 columns):
##  #   Column  Non-Null Count  Dtype
## ---  ------  --------------  -----
##  0   Y       1000 non-null   int64
##  1   proba   1000 non-null   float64
## dtypes: float64(1), int64(1)
## memory usage: 15.8 KB
```

```
# Matrice de confusion avec un seuil de 0.5
cm = confusion_matrix(Data1.Y,Data1.proba,0.5)
# Retourne un DataFrame
def confusion_matrix_df(conf_matrix):
  return pd.DataFrame({
        "Accuracy" : accuracy(conf_matrix),
        "Misclassification rate" : misclass_rate(conf_matrix),
        "Specificity" : specificity(conf_matrix),
        "Recall" : recall(conf_matrix),
        "Precision" : precision(conf_matrix),
        "F1" : f1(conf_matrix)},index = ["valeur"])
# Application
cm_df = confusion_matrix_df(cm)
```

**Table 2.1 –** Métriques de classification binaire - Seuil 0.5

| | Accuracy | Misclassification rate | Specificity | Recall | Precision | F1 |
|---|---|---|---|---|---|---|
| valeur | 0.743 | 0.257 | 0.7946667 | 0.588 | 0.4883721 | 0.5335753 |

### 2.2.4 Runnez de nouveau le code avec un seuil de $0.7$. Que constatez vous ?

```
# Matrice de confusion avec seuil fixé à 0.7
cm2 = confusion_matrix(Data1.Y,Data1.proba,0.7)
cm2_df = confusion_matrix_df(cm2)
```

**Table 2.2 –** Métriques de classification binaire - Seuil 0.7

| | Accuracy | Misclassification rate | Specificity | Recall | Precision | F1 |
|---|---|---|---|---|---|---|
| valeur | 0.79 | 0.21 | 0.9133333 | 0.42 | 0.6176471 | 0.5 |

### 2.2.5 Graphique des densités conditionnelles

Un outil utile à l'évaluation de notre modèle est l'utilisation des courbes conditionnelles.

Idéalement, les courbes de densité sont séparées de telle sorte que les scores faibles correspondent aux individus peu risqués $(y = 0)$ et les scores élevés aux individus les plus risqués $(y = 1)$.

Classiquement, on utilise une estimation à noyau (KDE, *Kernel Density Estimation*) pour afficher ces courbes. Dans le cas d'un noyaux gaussien, on a la fonction de densité suivante (qui somme à 1) :

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \tag{2.8}$$

On défini la fonction noyau où $h$ correspond au paramètre sigma de la fonction de densité gaussienne et $n$ au nombre d'observations.

$$f(x) = \frac{1}{nh} \sum_{i=1}^{i=n} K\left(\frac{x - x_i}{h}\right) \tag{2.9}$$

Ainsi, la fonction de densité estimée finale est la moyenne de la somme des densités gaussiennes de chaque point. En pratique, il est possible d'utiliser un autre noyau que lafonction gaussienne et il est nécessaire d'optimiser la valeur du paramètre $h$ par validation croisée.

A l'aide de la librairie plotnine, tracez le graphique des densités conditionnelles. Quel seuil de décision vous semble le plus adapté au problème ?

```
# Courbes de densités conditionnelles
from plotnine import *
```

```
D = Data1.copy()
D.Y = D["Y"].astype("category")
p = (ggplot(D,aes(x="proba",color="Y"))+geom_density(size=1)+
      labs(x="Probabilités prédites",y="Densité",
           title="Training Set's Predicted Score"))
print(p)
```

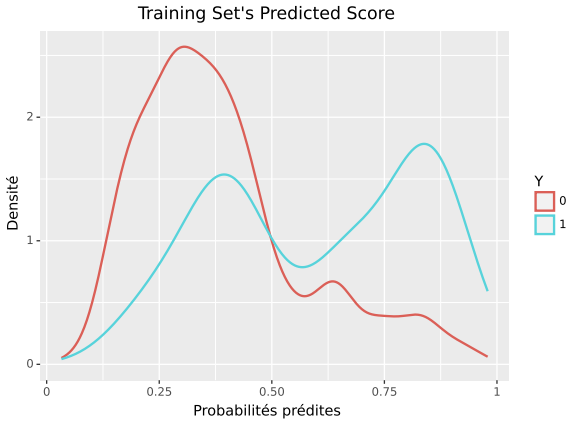


**Figure 2.1 –** Courbes de densités conditionnelles

### 2.2.6 Construction de la courbe ROC

#### 2.2.6.1 Courbe ROC

Nous allons jouer sur le seuil de décision afin de voir son impact sur deux métriques : la sensitivity et la specificity. Pour différentes valeurs de seuil choisies dans l'intervalle $[0, 1]$ avec un pas de $0.1$ calculez la sensitivity et la specificity et stockez les valeurs dans des vecteurs.

```
# Sensitivité et Specificité
s = np.array([x / 100 for x in range(0, 101)])
conf_matrix_vec = list(map(lambda x : confusion_matrix(Data1.Y,Data1.proba,x),s))
TP_vec = np.array([recall(x) for x in conf_matrix_vec])
FP_vec = np.array([specificity(x) for x in conf_matrix_vec])
```

#### 2.2.6.2 A partir des données précédentes, tracez le graphique de la sensitivité (axe Y) versus 1-spécificité (axe X).

```
# Courbe ROC
roc_df = pd.DataFrame({"FP" : 1 - FP_vec, "TP" : TP_vec})
p = (ggplot(roc_df,aes(x="FP",y="TP"))+geom_line(color="blue",size=1)+
      geom_abline(slope=1,intercept=0,color="black")+
      labs(x="False Positive Rate",y="True Positive Rate",
           title="ROC Curve"))
print(p)
```

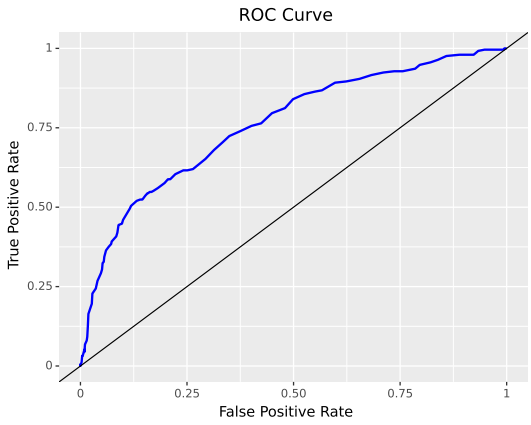


**Figure 2.2 –** ROC Curve

#### 2.2.6.3 Approximation de l'AUC : à partir de la méthode des trapèzes, calculez l'aire sous la courbe ROC

**Aide**

Vous devriez utilisez la formule ci - dessous :

$$\int_a^b f(x)\mathrm{d}x \approx (b-a) \times \frac{f(a)+f(b)}{2} \tag{2.10}$$

```
# Compute AUC
def compute_auc(fpr,tpr):
  fpr, tpr = np.array(fpr), np.array(tpr)
  n =len(fpr)
  auc = 0
  for i in range(1,n):
    auc = auc +((fpr[i]-fpr[i-1])*(tpr[i]+tpr[i-1]))
  auc = auc/2
  auc = np.sum(auc)
  return auc
# Application
auc = compute_auc(FP_vec, TP_vec)
print("AUC = %.3f"%(auc))
```

```
## AUC = 0.760
```

### 2.2.7 Contrôlez les résultats à l'aide du package sklearn

```
# ROC Curve with Sklearn
from sklearn import metrics as mt
fpr, tpr, _ = mt.roc_curve(Data1.Y,Data1.proba)
auc2 = mt.roc_auc_score(Data1.Y,Data1.proba)
# Représentation graphique
roc2_df = pd.DataFrame({"FP" : fpr, "TP" : tpr})
```

```
p = (ggplot(roc2_df,aes(x="FP",y="TP"))+geom_line(color="blue",size=1)+
      geom_abline(slope=1,intercept=0,color="black")+
      labs(x="False Positive Rate",y="True Positive Rate",
           title="ROC Curve (AUC = "+str(round(auc2,3))+")"))
print(p)
```

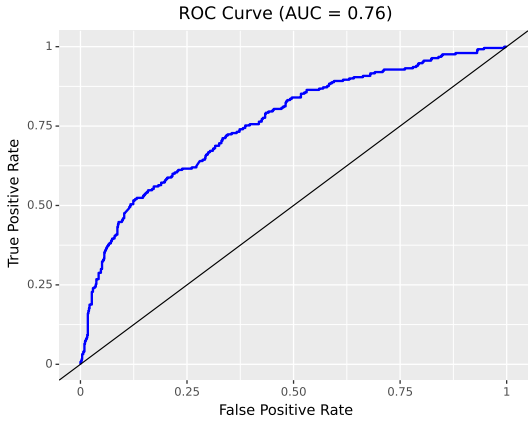


**Figure 2.3 –** ROC Curve

## 2.3 Généralisation des métriques d'évaluation dans le cas multiclasses

Dans cet section, nous étendons le cas binaire au cas multi - classes (avec C le nombre de classes). Les probabilités prédites sont alors données par la fonction softmax :

$$\text{softmax}(Z_j) = \frac{e^{Z_j}}{\sum_{c=1}^{c=C} e^{Z_c}} \tag{2.11}$$

Nous disposions d'un fichier de donnée (`Data_TD1_partie2.csv`) contenant : la variable « Y » qui est la variable cible. Cette variable peut prendre comme valeur $1$, $2$ ou $3$ correspond au vecteur de probabilités prédites par le modèle pour la classe $C1$ (resp. $C2$, $C3$) tel que $C1 + C2 + C3 = 1$.

L'objectif est d'évaluer ce classifieur à l'aide des données qui vous sont fournies

Pour cela, vous allez utiliser des mesures introduites dans l'exercice 1 et des mesures spécifiques au cas de la classification muiltiple.

### 2.3.1 Chargez les données "data_TD1_partie2"

En utilisant les outputs du modèle, assignez une classe à chaque observations. La règle d'attribution est la suivante : la classe ayant la probabilité la plus élevée est la classe prédite pour l'observation.

**Aide**

Utiliser la fonction « argmax » de la librairie numpy..

```
# Chargement de la base
Data2 = pd.read_csv("./donnee/data_TD1_partie2.csv",sep=";",decimal=',')
Data2.info()
```

```
## <class 'pandas.core.frame.DataFrame'>
## RangeIndex: 1000 entries, 0 to 999
## Data columns (total 4 columns):
##  #   Column  Non-Null Count  Dtype
## ---  ------  --------------  -----
##  0   Y       1000 non-null   int64
##  1   C1      1000 non-null   float64
##  2   C2      1000 non-null   float64
##  3   C3      1000 non-null   float64
## dtypes: float64(3), int64(1)
## memory usage: 31.4 KB
```

```
# Récupération des données
Data2["Class"] = np.argmax(Data2.iloc[:,1:].values,axis=1)+1
Data2.info()
```

```
## <class 'pandas.core.frame.DataFrame'>
## RangeIndex: 1000 entries, 0 to 999
## Data columns (total 5 columns):
##  #   Column  Non-Null Count  Dtype
## ---  ------  --------------  -----
##  0   Y       1000 non-null   int64
##  1   C1      1000 non-null   float64
##  2   C2      1000 non-null   float64
##  3   C3      1000 non-null   float64
##  4   Class   1000 non-null   int64
## dtypes: float64(3), int64(2)
## memory usage: 39.2 KB
```

### 2.3.2 Définissez une fonction "multi_confusion_matrix"

Cette fonction calcule le nombre de vrai positifs, vrais négatifs, faux positifs, faux négatifs. Elle prend en arguments la variable réponse, le vecteur des classes prédites et la classe de référence ($C = 1, 2$ ou $3$).

```
# Fonction multi confusion matrix
def multi_confusion_matrix(y_true, y_pred, classe):
  y_true, ypred = np.array(y_true), np.array(y_pred)
  FP = np.sum((y_pred == classe)*(y_true != classe))
  TP = np.sum((y_pred == classe)*(y_true == classe))
  FN = np.sum((y_pred != classe)*(y_true == classe))
  TN = np.sum((y_pred != classe)*(y_true != classe))
  return {"TP" : TP,"TN" : TN, "FP" : FP,"FN" : FN}
```

### 2.3.3 A partir de la fonction définie précédemment et des fonctions créées dans la section 2, calculez les métriques de performance du classifieur pour chaque classes.

```
#création de la matrice de confusion pour la classe 1
cm_c1 = multi_confusion_matrix(Data2.Y,Data2.Class,1)
#création de la matrice de confusion pour la classe 2
cm_c2 = multi_confusion_matrix(Data2.Y,Data2.Class,2)
#création de la matrice de confusion pour la classe 3
cm_c3 = multi_confusion_matrix(Data2.Y,Data2.Class,3)
multi_cm_df = pd.concat([confusion_matrix_df(cm_c1),confusion_matrix_df(cm_c2),
                         confusion_matrix_df(cm_c3)],axis=0).round(3)
multi_cm_df.index = ["C"+str(x+1) for x in range(3)]
```

**Table 2.3 –** Valeurs des métriques multiclasses

| | Accuracy | Misclassification rate | Specificity | Recall | Precision | F1 |
|---|---|---|---|---|---|---|
| C1 | 0.909 | 0.091 | 0.931 | 0.866 | 0.864 | 0.865 |
| C2 | 0.907 | 0.093 | 0.939 | 0.840 | 0.870 | 0.855 |
| C3 | 0.918 | 0.082 | 0.931 | 0.893 | 0.867 | 0.880 |

### 2.3.4 Evaluation globale du modèle

En utilisant les formules ci - dessous, calculer le micro - F1, macro - F1 et le weighted - F1.

#### 2.3.4.1 Calcul Micro - F1

Ce calcul s'effectue en trois étapes :

— **Etape 1 :** Calcul du micro - recall

$$\text{recall}_{\text{micro}} = \frac{\sum_{i=1}^{i=C} TP_i}{\sum_{i=1}^{i=C} (TP_i + FN_i)} \tag{2.12}$$

```
# Micro - recall
def micro_recall(y_true,y_pred,nb_classes):
  TP, FN = np.ones(nb_classes), np.ones(nb_classes)
  for i in range(1,nb_classes+1):
    cm = multi_confusion_matrix(y_true,y_pred,i)
    TP[i-1] = cm["TP"]
    FN[i-1] = cm["FN"]
  return(np.sum(TP)/(np.sum(TP)+np.sum(FN)))
```

— **Etape 2 :** Calcul de la micro - précision

$$\text{precision}_{\text{micro}} = \frac{\sum_{i=1}^{i=C} TP_i}{\sum_{i=1}^{i=C} (TP_i + FP_i)} \tag{2.13}$$

```
# Micro - précision
def micro_precision(y_true,y_pred,nb_classes):
  TP, FP = np.ones(nb_classes),np.ones(nb_classes)
  for i in range(1,nb_classes+1):
    cm = multi_confusion_matrix(y_true,y_pred,i)
    TP[i-1] = cm["TP"]
    FP[i-1] = cm["FP"]
  return(np.sum(TP)/(np.sum(TP)+np.sum(FP)))
```

— **Etape 3 :** Calcul du micro - F1

$$F1_{\text{micro}} = 2 \times \frac{\text{precision}_{\text{micro}} \times \text{recall}_{\text{micro}}}{\text{precision}_{\text{micro}} + \text{recall}_{\text{micro}}} \tag{2.14}$$

```
# Micro - F1
def compute_micro_f1(y_true, y_pred, nb_classes):
  micro_r = micro_recall(y_true, y_pred, nb_classes)
  micro_p = micro_precision(y_true, y_pred, nb_classes)
  return(2*(micro_p*micro_r)/(micro_p+micro_r))
# Application
micro_f1 = compute_micro_f1(Data2.Y,Data2.Class,3)
print("Micro - F1 = %.3f"%(micro_f1))
```

```
## Micro - F1 = 0.867
```

#### 2.3.4.2 Calcul du Macro - F1

$$F1_{\text{macro}} = \frac{1}{C} \sum_{i=1}^{i=C} F1_i \tag{2.15}$$

```
# Macro - f1
def compute_macro_f1(y_true, y_pred, nb_classes):
  f_1 = np.ones(nb_classes)
  for i in range(1,nb_classes+1):
    cm = multi_confusion_matrix(y_true,y_pred,i)
    f_1[i-1] = f1(cm)
  return(np.sum(f_1)/nb_classes)
# Application
macro_f1 = compute_macro_f1(Data2.Y,Data2.Class,3)
print("Macro - F1 = %.3f"%(macro_f1))
```

```
## Macro - F1 = 0.867
```

#### 2.3.4.3 Calcul du weighted - F1

$$F1_{\text{weighted}} = \frac{\sum_{i=1}^{i=C} n_i \times F1_i}{\sum_{i=1}^{i=C} n_i} \tag{2.16}$$

```
# Weighted - F1
def compute_weighted_f1(y_true, y_pred,nb_classes):
  f_1, temp = np.ones(nb_classes), np.ones(nb_classes)
  for i in range(1,nb_classes+1):
    cm = multi_confusion_matrix(y_true,y_pred,i)
    f_1[i-1] = f1(cm)
    temp[i-1] = np.sum(y_true==i)
  return np.sum(f_1*temp)/np.sum(temp)
# Application
weighted_f1 = compute_weighted_f1(Data2.Y,Data2.Class,3)
print("Weighted - F1 = %.3f"%(weighted_f1))
```

```
## Weighted - F1 = 0.867
```

#### 2.3.4.4 Que remarquez - vous sur les résultats donnés par ces trois métriques ?

### 2.3.5 Courbe ROC dans le cas multi - classes

Il est également possible de construire une courbe ROC dans le cas multi - classes. Pour cela, on commence par construire un vecteur de probabilités pour chaque classes puis on utilise la même méthodologie que dans la partie 1 pour l'affichage de la courbe ROC.

#### 2.3.5.1 Conversion en problème binaire : construisez une variable indicatrice pour chaque classes à partir de la variable de réponse et récupérez le vecteur de probabilités associé

```
# Récupération des variables indicatrices pour chaque classe
label_c1 = (Data2.Y==1)*1
label_c2 = (Data2.Y==2)*1
label_c3 = (Data2.Y==3)*1
# Récupération des scores de chaque classes
y_proba_c1 = Data2.C1
y_proba_c2 = Data2.C2
y_proba_c3 = Data2.C3
```

#### 2.3.5.2 Construction de la courbe ROC pour chaque classes : utilisez les fonctions construites dans la partie 1 pour construire la courbe ROC

```
# Sensitivité et Specificité pour la classe 1
conf_matrix_c1 = list(map(lambda x : confusion_matrix(label_c1,y_proba_c1,x),s))
TP_vec_c1 = np.array([recall(x) for x in conf_matrix_c1])
FP_vec_c1 = np.array([specificity(x) for x in conf_matrix_c1])

# Sensitivité et Specificité pour la classe 2
conf_matrix_c2 = list(map(lambda x : confusion_matrix(label_c2,y_proba_c2,x),s))
TP_vec_c2 = np.array([recall(x) for x in conf_matrix_c2])
FP_vec_c2 = np.array([specificity(x) for x in conf_matrix_c2])

# Sensitivité et Specificité pour la classe 3
conf_matrix_c3 = list(map(lambda x : confusion_matrix(label_c3,y_proba_c3,x),s))
TP_vec_c3 = np.array([recall(x) for x in conf_matrix_c3])
FP_vec_c3 = np.array([specificity(x) for x in conf_matrix_c3])


# Courbe ROC
D = pd.DataFrame({"FP1" : 1-FP_vec_c1,"TP1" : TP_vec_c1,
                  "FP2" : 1-FP_vec_c2,"TP2" : TP_vec_c2,
                  "FP3" : 1-FP_vec_c3,"TP3" : TP_vec_c3})
# Représentation graphique
p = (ggplot(D)+
      geom_line(aes(x="FP1",y="TP1"),color="blue",size=1)+
      geom_line(aes(x="FP2",y="TP2"),color="red",size=1)+
      geom_line(aes(x="FP3",y="TP3"),color="green",size=1)+
      geom_abline(slope=1,intercept=0,color="black")+
      labs(x="False Positive Rate",y="True Positive Rate",
           title="ROC Curve"))
print(p)
```

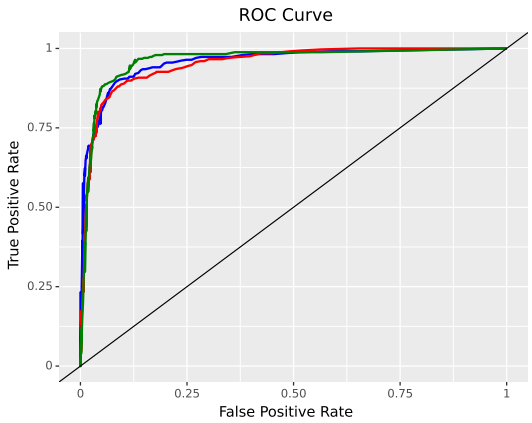


**Figure 2.4 –** ROC Curve

#### 2.3.5.3 Calculez l'AUC pour chaque classe

```
# AUC
multi_auc_df = pd.DataFrame({
  "C1" : compute_auc(FP_vec_c1, TP_vec_c1),
  "C2" : compute_auc(FP_vec_c2, TP_vec_c2),
  "C3" : compute_auc(FP_vec_c3, TP_vec_c3),
},index =["auc"]).round(3)
```

**Table 2.4 –** AUC pour chaque classe

| | C1 | C2 | C3 |
|---|---|---|---|
| auc | 0.959 | 0.953 | 0.963 |

#### 2.3.5.4 Contrôle des résultats à l'aide du package Sklearn

```
# ROC AUC Score
from sklearn.metrics import roc_auc_score
multi_auc = roc_auc_score(Data2.Y,Data2.iloc[:,1:4],average=None,
                          multi_class='ovr',labels =[1,2,3])
multi_auc


## array([0.95889559, 0.95304109, 0.96233289])

# Classification report
report = mt.classification_report(Data2.Y,Data2.Class,
                                  target_names=["C1","C2","C3"],
                                  output_dict=True)
report = pd.DataFrame(report).transpose()
```

**Table 2.5 –** Classification report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| C1 | 0.8639053 | 0.8664688 | 0.8651852 | 337.000 |
| C2 | 0.8698413 | 0.8404908 | 0.8549142 | 326.000 |
| C3 | 0.8674352 | 0.8931751 | 0.8801170 | 337.000 |
| accuracy | 0.8670000 | 0.8670000 | 0.8670000 | 0.867 |
| macro avg | 0.8670606 | 0.8667116 | 0.8667388 | 1000.000 |
| weighted avg | 0.8670300 | 0.8670000 | 0.8668689 | 1000.000 |

### 2.3.6 A partir des éléments précédents, concluez sur la qualité du classifieur

D'après les métriques calculées aux questions précédentes, le classifieur est capable de prédire les trois classes avec une performance similaire qui est très élevée. En pratique, il est très rare qu'on ait de si bon résultats (notamment en credit scoring).