Compte Rendu Gestion des Risques

2023 - 2024

Axel Grille

Introduction

Ce document éclaircira les projets 1 et 3 de gestion des risques. La première partie sera donc dédiée au code de l'approche RaRoC et la seconde sur les pricers d'options vanilla, napoleon, Himalaya et tunnel. Pour chacune des parties, nous allons expliciter les fondements mathématiques et les formules suivies puis nous expliquerons la structure du code.

I/ L'approche RaRoC

---

1. Les fondements mathématiques

Dans un premier temps, nous allons voir comment nous allons calculer notre RaRoC en fonction des données que nous avons dans la feuille 'Params' de l'excel fournit.

La formule que nous utiliserons est la suivante:

$$RaRoC = \frac{Risque\ ajuste\ au\ capital + Capitaleconomique \times TSR - taxes}{Capital\ economique}$$

Nous calculerons donc dans un premier temps le Produit Net Bancaire (PNB), les coûts de liquidité et de crédit, l'expected loss (EL), l'unexpected loss (UL) et le taux sans risque (TSR)
Nous prendrons également en compte les probabilités de défauts liées aux pays, les taux de transfert, les garants et les garanties.

Nous calculerons le PNB:

$$PNB = customer\ autorization \times margin$$

L'Expected Loss:

$$UL = EAD \times LGD \times (f \times \beta \times \sqrt{\rho \times PD \times (1 - PD)})$$

Avec: f : Diversification du portefeuille
β : Facteur de correction gaussien
ρ : Corrélation moyenne du portefeuille

Puis nous aurons :

$$Credit = PNB - cout\ credit - cout\ liquidite$$

Qui nous permettra de calculer le risque ajusté:

$$Risque\ ajuste = credit - EL$$

Le capital économique est calculé de la façon suivante:

$$Capital\ economique = (f \times \rho \times PDcond \times (1 - PDcond) \times LGD \times (EAD - garanties)) \times TSR$$

---

2. Organisation du code

Nous utiliserons les bibliothèques suivantes:

- Pandas pour extraire et traiter les données du tableur excel
- La librairie maths pour les calculs
- Tkinter pour les formulaires

Le premier bloc de code permet d'extraire les probabilités de défaut et d'en faire un dictionnaire pour faciliter le traitement. Nous ajoutons également quelques notations de pays et des garanties. Ces deux derniers nous permettent de calculer le RaRoC de manière plus réaliste en combinant tous les facteurs.

1. La classe RarocDefault

Cette classe nous sert à initialiser les paramètres par défaut, elle n'a rien de particulier. La docstring énumère tous les paramètres initialisés.

2. La classe CustomerMaturity

Cette classe permet d'initialiser les garanties du client et de prendre en compte les garants. Ces données sont utilisées dans le calcul du RaRoC.

3. La classe CustomerData

Cette classe modélise le client qui veut contracter un crédit, il s'agit d'initialiser les champs décrits dans la docstring.

4. La méthode creditWarranties

Cette méthode calcule la valeur des garanties du client en prenant en compte les données de CustomerMaturity.

5. La méthode countryData

Cette méthode va chercher le LGD associé au pays, la note du pays et le taux de transfert dans les données pays du premier bloc de code

6. La méthode rarocCalcul

Enfin, cette méthode prend en compte toutes les données précédentes pour calculer le RaRoC en pourcentage

7. La classe RarocCalculatorGUI

Cette classe permet simplement de créer un formulaire avec des champs a remplir et un bouton calcul pour afficher le résultat du RaRoC.

II/ Les pricers d'options

Pour ce second projet, nous allons voir comment réaliser des pricers d'options pour les options:

- Vanilla
- Napoleon
- Tunnel
- Himalaya

Tous les pricers sont structurés de la même manière, la principale difficulté résidant dans le calcul du payoff de chaque option.
Nous allons donc expliciter le calcul de chaque payoff et présenter une seule fois la structure du code.

1. Modèle de diffusion

Nous considérons le modèle de diffusion suivant:

$$S_t = S_{t-1} \exp(\mu - \frac{\sigma^2}{2} dt \ + \ \sigma \sqrt{dt} W)$$

Où W suit une loi normale centrée réduite

2. Payoff Vanilla

Le payoff d'une option vanilla s'écrit de la sorte:

$$payoff = \max((Moyenne\ 60\ derniers\ jours \div Moyenne\ 60\ premiers\ jours) - K, 0)$$

Nous ne considérons dans notre cas que les 2 premiers mois et les 2 derniers mois, d'où la création des vecteurs temporels.

3. Payoff Napoleon

Le payoff d'une option napoleon est:

$$Payoff_i = \sum_{i=1}^{n} max(floor, C \ + min_j(R_{i,j}))$$

Ce payoff est distribué annuellement, notre vecteur temporel s'étendra donc sur un an.

4. Payoff Tunnel

Nous avons une valeur inférieure K1 et une valeur supérieure K2 qui forment un tunnel, nous regarderons le prix de l'action par rapport a ce tunnel tous les 3 mois. Le coupon stocké par le détenteur de l'option sera alors de :

- $\frac{S_t}{S_0} \times \beta \; si \; S_t > K_2$

- $\frac{S_t}{S_0} \times \alpha \; si \; K_2 > S_t > K_1$

- L'acheteur perd tous ses coupons si $K_1 > S_t$

Le payoff d'une option tunnel est:

$$Payoff = Coupon \; + \; \max(\frac{S_n}{S_0} - K_2, 0)$$

Où n est l'année de maturité.

5. Payoff Himalaya

Dans ce contexte, nous avons un ensemble d'actions de prix $Y_1(t), Y_2(t), \ldots, Y_d(t)$ au temps t.
Notons $R_{i,j} = \frac{S_i(t_j)}{S_0(t_j)}$ la performance de l'action i au temps t_j.
D'où $R_{k(i)} = max_{j \in 1,n \setminus \bigcup_{l=1}^{i-1}} R_{l,i}$

Et notre payoff sera donc:

$$Payoff = max(\sum_{i=1}^{n} R_{k(i),i}, i, 0)$$

Le payoff est calculé annuellement.

6.Structure du code

Notre code est structuré en plusieurs méthodes:
__init__ : Cette méthode sert à initialiser les variables de chaque pricer.

generate_t : Cette méthode crée des vecteurs temporels en fonction des caractéristiques de chaque option.

simulate_gbm: Cette méthode va simuler le mouvement brownien du prix en fonction du modèle de diffusion.

payoff : Calcule le payoff de l'option

simulate_monte_carlo : Cette méthode va générer un nombre d'expériences déterminé par l'utilisateur.

converence_mc : Cette méthode va calculer les points de la méthode de Monte-Carlo en fonction d'un intervalle de confiance donné.

generate_convergence_curve: Cette méthode trace les graphiques de convergence calculés avec la méthode précédente.

La classe OptionPricerGUI crée les formulaires d'utilisation des pricers, permettant d'entrer les paramètres et d'afficher les graphiques de convergence.

IV/ Sources

Toutes les ressources sont tirées des documents donnés en cours, de sites internets et de chat GPT pour la resolution de bugs et informations supplémentaires.

J'ai également utilisé le cours de calcul stochastique de Mme Halconruy pour mieux comprendre certaines notions.