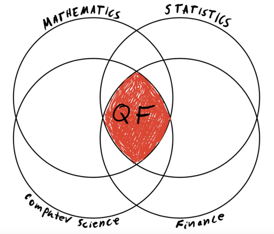
Soutenance de projet TX : Découverte du Machine Learning pour la finance

Ambroise THIBAULT

Encadré par Mme. Ghislaine GAYRAUD et M. Cyprien GILET

Lundi 17 juin 2024

Introduction et objectifs



De nombreux domaines en finance quantitative : Marché des actions, taux d'intérêt, produits dérivés, gestion de portefeuille, risques, régulations...
Notre objectif :

- Comprendre les produits dérivés de base (Forwards, Options)
- Implémenter les méthodes classiques de pricing (Black-Scholes)
- Comparer les résultats avec les méthodes de Machine learning, sur des données réelles.

Sommaire

Sommaire

Approche mathématiques et modèle de BSM

On considère le modèle avec 2 types actifs :

- Actif risqué : une action noté S (stock):
- Actif sans risque : une obligation, un bond du trésor... noté B (bond), avec un taux sans risque r.

Le modèle de Black-Scholes-Merton (BSM) est composé de 2 actifs ayant les dynamiques suivantes :

$$dB_t = rB_t dt$$

$$dS_t = \mu S_t dt + \sigma S_t dW_t^{\mathbb{P}}$$

Avec :

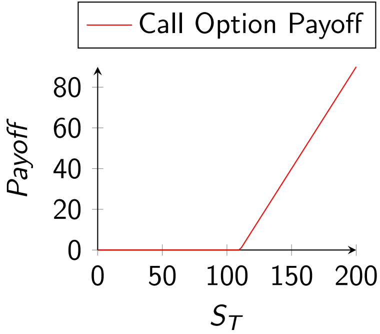
- $\mu \in \mathbb{R}$ l'espérance de rentabilité de l'action
- $\sigma \in \mathbb{R}$ la volatilité de l'action
- $W^{\mathbb{P}}$ un mouvement brownien sous la probabilité $\mathbb{P}$

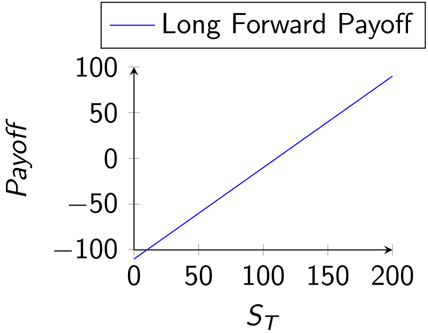
Produits dérivés, payoff et prix

Un produit dérivé, ou contigent claim, $\mathcal{X}$ est un actif financier dont la valeur dépend de la valeur d'un autre actif financier (sous-jacent).
On notera $\mathcal{X}$: $\Phi(S(T))$

Prix actuel $S_0 = 100$€ Prix d'exercice $K = 110$€ (obligation d'achat dans T = 6 mois)	Forward $\longrightarrow$	Dans 6 mois Gagner ou perdre la différence en- tre $S(T)$ et K
Droit d'achat (Option d'achat dans T = 6 mois)	Option $\longrightarrow$	Payer pour l'option Π_0 Perte maximale fixe Gain non limité

Payoff : Ce que fait perdre ou gagner l'exercice du contrat.





$$\Phi(S(T))^{\text{Call}} = \begin{cases} S_T - K & \text{si} \quad S_T > K \\ 0 & \text{sinon} \end{cases} \qquad \Phi(S(T))^{\text{Long Forward}} = S_T - K$$

Formule de B-S pour un option call

On considère une option call de strike K et de maturité T.
On a le payoff suivant :

$$\Phi(S(T)) = \begin{cases} S_T - K & \text{si} \quad S_T > K \\ 0 & \text{sinon} \end{cases}$$

En appliquant la formule d'évaluation risque neutre à ce payoff, on obtient :

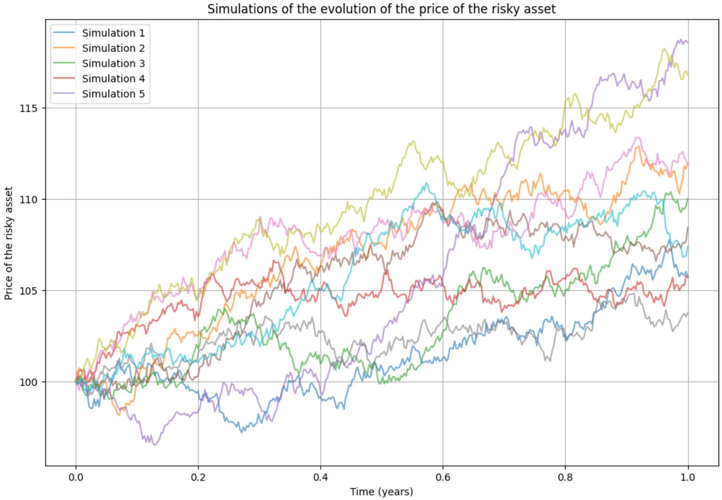
$$F_c(t, S_0) = S_0 \mathcal{N}(d_1) - Ke^{-r(T-t)} \mathcal{N}(d_2)$$

Avec :

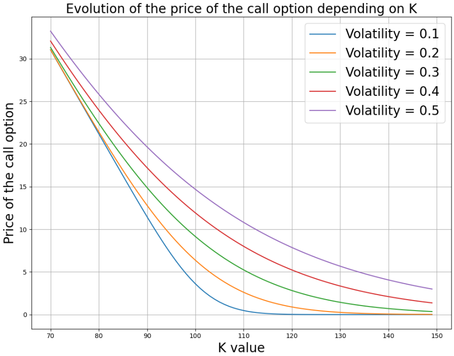
$$d_1 = \frac{1}{\sigma\sqrt{T-t}} \left[\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t) \right]$$

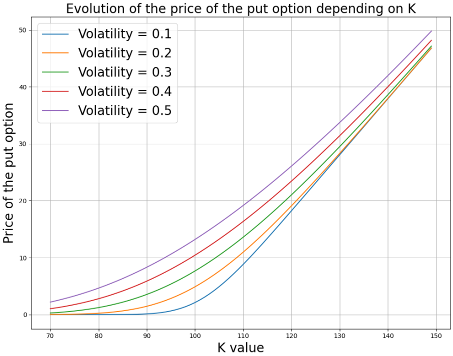
$$d_2 = d_1 - \sigma\sqrt{T-t}$$

Simulation de l'évolution de l'actif risqué sous BS

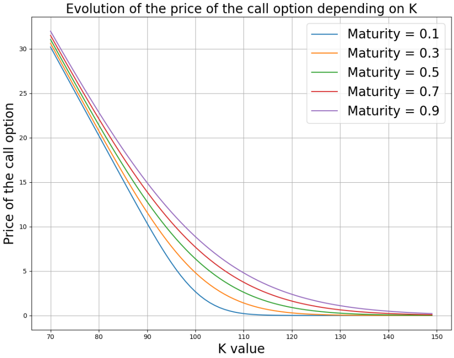


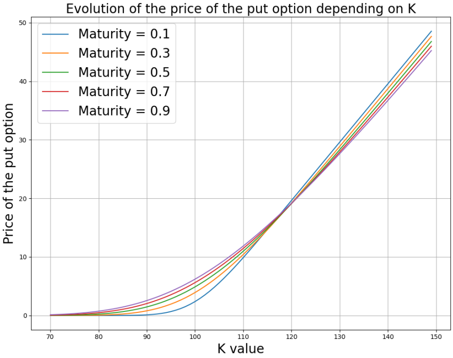
Prix d'options en fonction de la volatilité σ et du prix d'exercice K



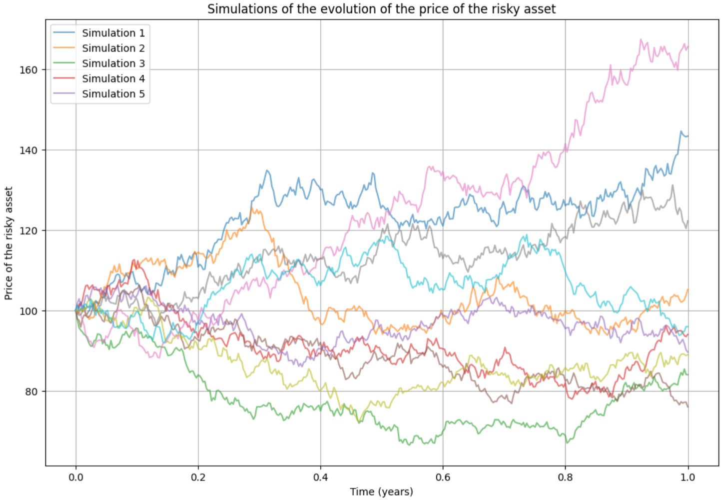


Prix d'options en fonction de la maturité T et du prix d'exercice K





Simulation de l'évolution de l'actif correspondant



Sommaire

Les 3 sources de données

Une base de donnée Dolthub contenant des données de marché d'options

- Bid et Ask
- Strike K
- date d'expiration et de création
- Type d'option (Call ou Put)
- Grecks (Delta, Gamma, Vega, Theta, Rho)
- Volatilité
- Symbol de l'action sous jacente

Une base de donnée Dolthub contenant des données de marché d'actions

- Prix de l'action
- Date
- Symbol de l'action

Le site du trésoir américain pour les taux d'intérêt sans risque r

Construction du jeu de données

Finalement après pre-traitement, on obtient un jeu de données contenant sous la forme d'un fichier csv avec

- 928 514 individus
- L'ensemble des variables nécessaires pour la formule de Black-Scholes
- Label (prix de l'option) est la moyenne du bid (demande) et de l'ask (offre)
- Les grecques : sensibilité du prix de l'option à des variations de paramètres

Sommaire

Modèle de Régression Non Paramétrique avec Processus Gaussien

- Régression par Processus Gaussien (GPR)
  - Modélise une fonction f à partir d'un ensemble de données $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$.
  - Suppose que les valeurs de f sont des variables aléatoires gaussiennes, idéal pour des données bruitées.
- Processus Gaussien (GP)
  - Ensemble de variables aléatoires dont tout sous-ensemble fini a une distribution gaussienne conjointe.
  - La fonction noyau (fonction de covariance) mesure la similarité entre les points.
  - Les paramètres du noyau sont appris en maximisant la vraisemblance des données d'entraînement.

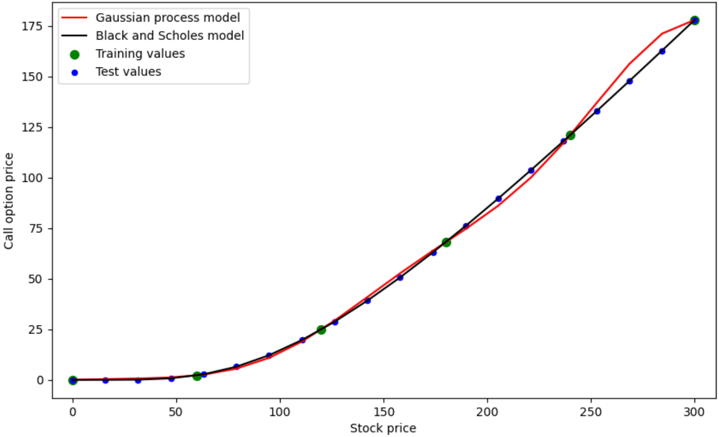
Formalisation du Modèle

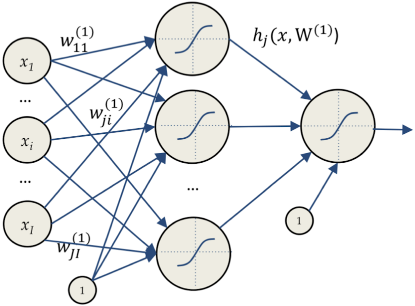
- Données Fournies
  - X : Matrice $n \times d$ des entrées d'entraînement.
  - y : Vecteur $n \times 1$ des sorties d'entraînement.
- Définition du Processus Gaussien
  - Pour tout ensemble de points $X = \{x_1, \dots, x_n\}$, $(f(x_1), \dots, f(x_n)) \sim \mathcal{N}(\mu, K_{X,X})$.
- Fonction de Covariance $k(x_i, x_j)$
  - Choisie pour capturer les similarités entre les points de données.
  - Paramètres estimés à partir des données pour ajuster la structure de covariance.

Distribution Prédictive

- Prédiction au Point de Test x_*
  - $f_* | x_*, X, y \sim \mathcal{N}(\mathbb{E}[f_* | x_*, X, y], \mathrm{var}(f_* | x_*, X, y))$
- Espérance et Variance de la Distribution Prédictive
  - $\mathbb{E}[f_* | x_*, X, y] = K_*(K + \sigma^2 I)^{-1} y$
  - $\mathrm{var}(f_* | x_*, X, y) = k(x_*, x_*) - K_*(K + \sigma^2 I)^{-1} K_*^T$

Application à des données générés par un modèle de Black-Scholes





$$g_1(x_n|w) = g_1\left(\sum_{j=0}^{J} w_{1j}^{(2)} \cdot h_j\left(\sum_{i=0}^{I} w_{ji}^{(1)} \cdot x_{i,n}\right)\right)$$

On cherche à minimiser : $E(\omega) = \sum_{n=1}^{N}(t_n - g_1(x_n, w))^2$
On utilise principe de back propagation pour minimiser cette fonction
Chain rule pour calculer les dérivées partielles de $E(\omega)$ par rapport à ω puis on applique descente de gradient.

Fonctions d'activations

Fonctions utilisées :

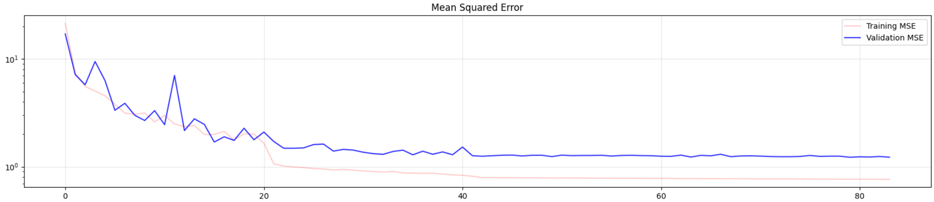
- Sigmoïde : $\sigma(x) = \frac{1}{1+e^{-x}}$
- Tanh : $\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
- ReLU : $f(x) = \max(0, x)$
- Leaky ReLU : $f(x) = \max(0.01x, x)$
- Softmax : $\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$

On utilisera pour les couches cachées Leaky ReLU et pour la couche de sortie ReLU

Architecture du modèle

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 10)	0
dense_1 (Dense)	(None, 128)	1,408
relu_1 (Activation)	(None, 128)	0
dense_2 (Dense)	(None, 128)	16,512
relu_2 (Activation)	(None, 128)	0
output (Dense)	(None, 1)	129
output_activation (Activation)	(None, 1)	0

Apprentissage du modèle

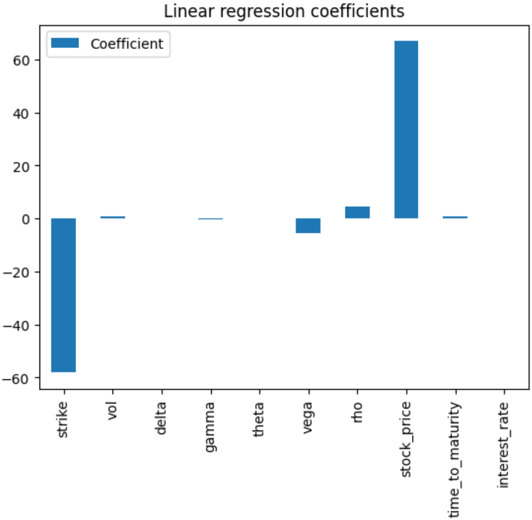


Régression linéaire multiple

standardize	fit_intercept	MSE
False	True	8.747
True	True	8.209
False	False	8.576
True	False	38.449

Variable	std_F_intercept_T	std_T_intercept_T	std_F_intercept_F	std_T_intercept_F
strike	-0.363373	-58.916325	-0.363689	-58.916325
vol	3.756853	0.721962	3.606665	0.721962
delta	0.335665	0.114061	0.255977	0.114061
gamma	-3.869552	-0.387780	-3.998690	-0.387780
theta	0.935594	0.111969	0.922205	0.111969
vega	-36.960191	-5.399981	-36.864209	-5.399981
rho	54.976466	4.846585	55.246246	4.846585
stock_price	0.414179	66.938573	0.414222	66.938573
time_to_maturity	11.772136	0.554776	11.046448	0.554776
interest_rate	4.394877	0.006967	-89.811216	0.006967
intercept	-1.674259	5.471931	0.000000	0.000000

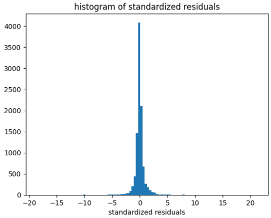
Coefficients de la régression linéaire multiple

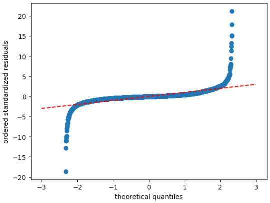


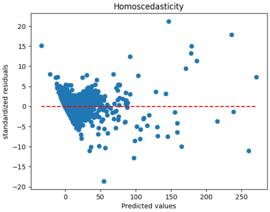
Résultats de la régression

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.943
Model:                            OLS   Adj. R-squared:                  0.943
Method:                 Least Squares   F-statistic:                 1.660e+04
Date:                Fri, 14 Jun 2024   Prob (F-statistic):               0.00
Time:                        13:22:16   Log-Likelihood:                -24196.
No. Observations:               10000   AIC:                         4.841e+04
Df Residuals:                    9989   BIC:                         4.849e+04
Df Model:                          10
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
Intercept           -1.1779      0.281     -4.187      0.000      -1.729      -0.626
strike              -0.3235      0.003   -108.430      0.000      -0.329      -0.318
vol                  3.4654      0.145     23.842      0.000       3.181       3.750
delta                0.6758      0.098      6.914      0.000       0.484       0.867
gamma               -4.2191      0.286    -14.762      0.000      -4.779      -3.659
theta               -0.2269      0.348     -0.653      0.514      -0.908       0.454
vega               -39.7103      0.426    -93.157      0.000     -40.546     -38.875
rho                 66.0698      0.847     77.992      0.000      64.409      67.730
stock_price          0.3701      0.003    117.378      0.000       0.364       0.376
time_to_maturity     9.2353      0.660     13.996      0.000       7.942      10.529
interest_rate       -7.6503     16.722     -0.457      0.647     -40.429      25.129
==============================================================================
Omnibus:                     5021.997   Durbin-Watson:                   1.995
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1332294.862
Skew:                           1.222   Prob(JB):                         0.00
Kurtosis:                      59.494   Cond. No.                     1.69e+05
```

Analyse des résidus

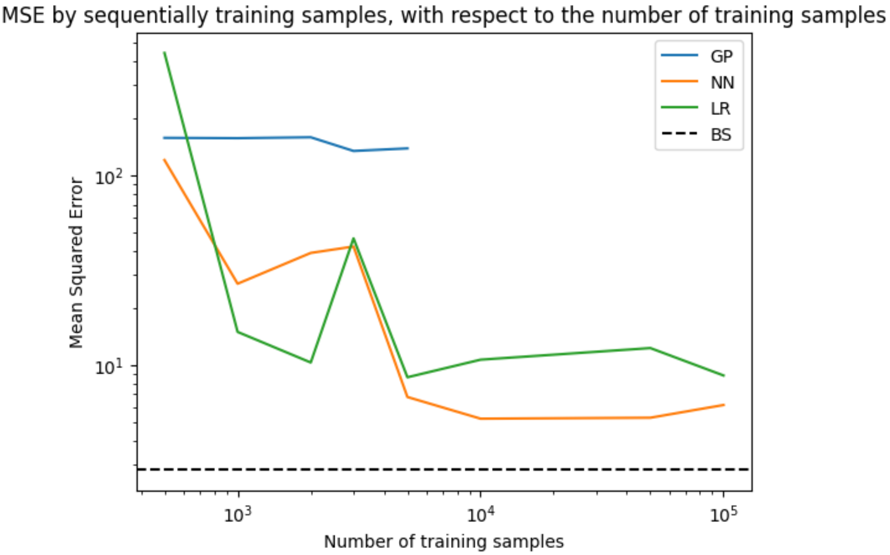




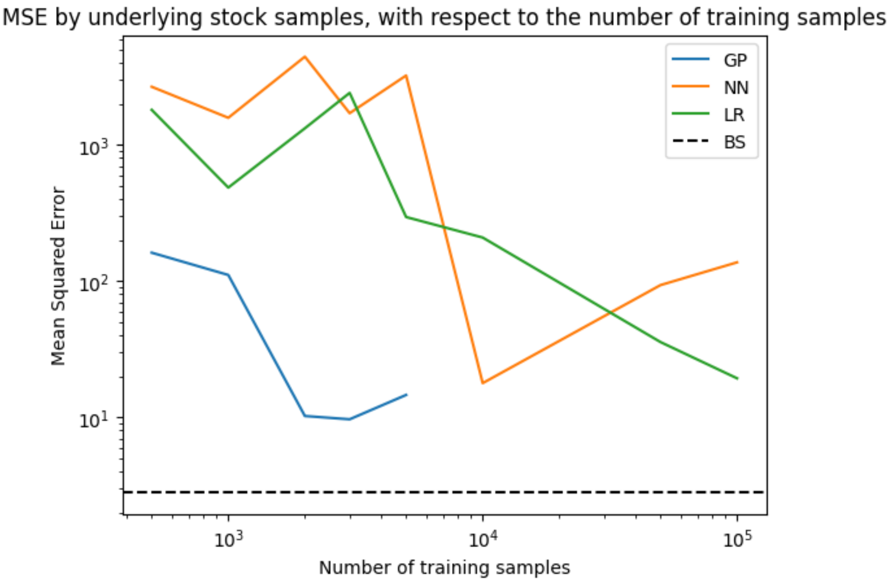


Sommaire

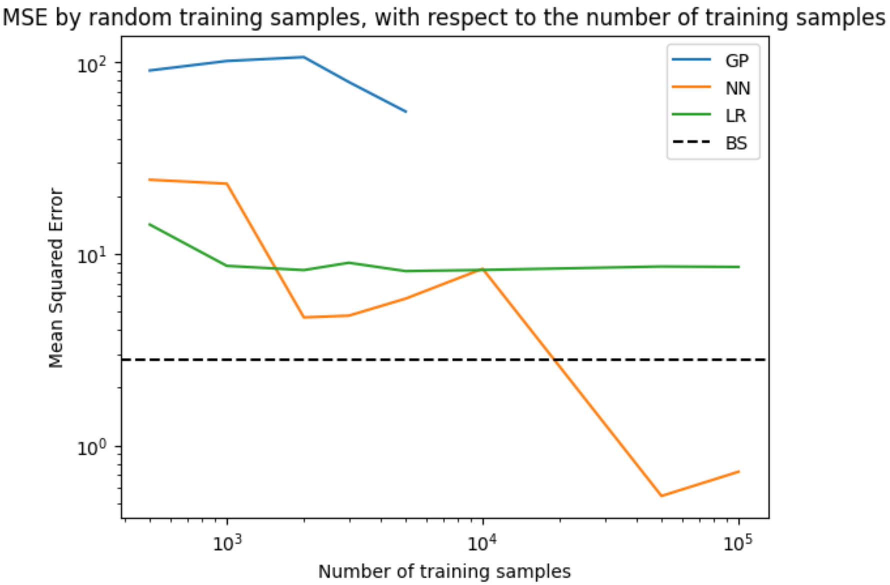
Résultats, données par index : séquentiel



Résultats, données par sous-jacent



Résultats, données Aléatoires



Comparaison des Méthodes

Méthode	Avantages	Inconvénients
Black-Scholes	Très précis et rapide en raison de son aspect déterministe	Impossible d'améliorer le modèle. Les hypothèses sont loin de la réalité. Nombre de caractéristiques fixe
Processus Gaussien	Peut capturer des motifs complexes avec une petite quantité de données	Très lent à entraîner, résultats pas très bons avec cette architecture
Réseaux de Neurones	Très bons résultats, peut capturer des motifs complexes. Il existe des améliorations majeures pour cette architecture *naïve*	Faible niveau d'interprétation : c'est une boîte noire. Nécessite beaucoup de données pour être efficace
Régression Linéaire Multiple	Bons résultats, très rapide à entraîner, haute interprétabilité	Difficile d'améliorer les résultats

Définition d'un processus stochastique

Un processus stochastique est une fonction à deux variables :

$$X : \Omega \times \mathbb{R}_+ \to (E, \mathcal{E})$$

On note généralement X, X_t ou $X_t(\omega)$
On dit que le processus est *adapté* à la filtration $(\mathcal{F}_t)_t$ si X_t est $\mathcal{F}_t$-mesurable pour tout t.
En générale, X représente un actif financier, un contrat...

Mouvement Brownien

$B = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, (B_t)_{t\geq 0}, \mathbb{P})$ est un mouvement Brownien si :

- $B_0 = 0$ p.s.
- $B_t - B_s \sim \mathcal{N}(0, t-s) \quad \forall s \leq t$
- $B_t - B_s \perp F_s \quad \forall s \leq t$

Remarques :

- Le mouvement est un processus stochastique continu, i.e. les trajectoires (ω fixé) sont continues.
- $B_t \sim \mathcal{N}(0, t)$
- $\forall t_1 < t_2 < ... < t_n$, les incréments $B_{t_1}, B_{t_2} - B_{t_1}, B_{t_3} - B_{t_2}, ..., B_{t_n} - B_{t_{n-1}}$ sont indépendants

Formule d'Itô

Ce résultat central du calcul stochastique donne la différentielle d'une fonction de processus d'Itô.

Soit :

- X un processus d'Itô tel que $dX_t = \mu_t dt + \sigma_t dB_t$
- $f : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}$, $\mathcal{C}^1$ en t et $\mathcal{C}^2$ en x

Alors $Y_t = f(t, X_t)$ est un processus d'Itô et sa différentielle est donnée par :

$$dY_t = \left(\frac{\partial f(t, X_t)}{\partial t} + \frac{\partial f(t, X_t)}{\partial x} \mu_t + \frac{1}{2} \frac{\partial^2 f(t, X_t)}{\partial x^2} \sigma_t^2 \right) dt + \frac{\partial f(t, X_t)}{\partial x} \sigma_t dB_t$$

EDP ou Équation de B-S-M

On se place dans le modèle de B-S-M
Notre objectif est de trouver le prix d'un produit dérivé $\mathcal{X} = \Phi(S(T))$ à l'instant t via la fonction $F(t, S_t) = \Pi_t$
En utilisant la formule d'Itô, on a l'équation de B-S-M :

$$\frac{\partial F(t,s)}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 F(t,s)}{\partial s^2} + rs\frac{\partial F(t,s)}{\partial s} - rF(t,s) = 0$$

L'évaluation risque-neutre

Objectif : trouver une formule explicite pour le prix, trouver $F(t, S_0)$
Équation de B-S-M est indépendante de μ.
On peut donc supposer que les investisseurs sont neutres face au risque.
On introduit un nouvel actif X :

$$dX_t = rX_t dt + \sigma X_t dW_t^{\mathbb{Q}}$$

En appliquant la formule d'Itô à $e^{-rt}F(t, S_t)$, on obtient la formule d'évaluation risque neutre :

$$F(t, S_0) = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}[\Phi(S(T))]$$

Les grecques

Les grecques représentent la sensibilité du prix d'une option par rapport à différentes variables (le sous-jacent (S), la volatilité (σ), taux sans risque (r), durée de vie de l'option).

- Delta Δ : sensibilité du prix de l'option par rapport à S

$$\Delta_c = \frac{\partial C}{\partial s} = \mathcal{N}(d_1)$$

- Gamma Γ : sensibilité de Δ par rapport à S

$$\Gamma_c = \frac{\partial^2 C}{\partial^2 s} = \frac{\mathcal{N}'(d_1)}{\sigma S_t \sqrt{T-t}}$$

- Véga ν : sensibilité par rapport à la volatilité σ

$$\nu_c = s\mathcal{N}'(d_1)\sqrt{T-t}$$

Références

John Hull.
Options, Futures, et Autres Actifs Dérivés, 11ème édition.
Pearson, 2018.

P. Baldi.
Stochastic Calculus.
Springer, 2017.

T. Bjork.
Arbitrage Theory in Continuous Time.
Oxford University Press, 2009.

Matthew F. Dixon, Igor Halperin, Paul Bilokon.
Machine Learning in finance.
Springer, 2020.

Références

Codruț-Florin Ivașcu.
Option pricing using Machine Learning.

Alexander Ke, Andrew Yang
Option Pricing with Deep Learning.