



PROJET PYTHON

Note de procédure et méthodologique

Conception d'une plateforme pour la gestion des options financières.

Réalisé par :

BENYAMINA Lina

SOMAVO Modeste Emmanuel

TOUAK Enzo

Sous la supervision de :

FARIA Thomas

2024-2025

Sommaire

1. Synthèse du document	3
1.1 Objectif.....	3
1.2 Contexte	3
2. Conception et organisation des données	4
2.1 Objectif.....	4
2.2 Conception de la base de données.....	4
2.3 Données en Temps Réels	5
3. Présentation et portée de l'application.....	6
3.1 Actions sélectionnées : représentativité et pertinence.....	6
3.2 Fonctionnalités principales	7
4. Architecture du projet.....	8
4.1 Structure du dépôt GitHub	8
4.2 Gestion des données et architecture globale	8
5. Méthodologie de calcul	9
5.1 Pricing des options : Méthode de Longstaff-Schwartz (2001).....	9
5.2 Méthodologie pour le calcul des Greeks.....	10
6. Avantages de l'application OptionLab et pistes d'amélioration.....	11

1. Synthèse du document

1.1 Objectif

Ce document vise à présenter le développement et l'implémentation de l'application Streamlit réalisée dans le cadre du projet Python.

1.2 Contexte

Les acteurs des marchés financiers, qu'ils soient acheteurs, vendeurs ou analystes, doivent souvent faire face à des calculs complexes (Greeks, valorisation) et à un manque d'outils intuitifs pour comparer et comprendre les options disponibles sur un ensemble d'actions stratégiques comme **Apple**, **Microsoft**, **Tesla**, et bien d'autres.

Cette application a été conçue pour répondre à cette problématique en simplifiant l'expérience utilisateur. Elle offre des outils d'aide à la décision basés sur des méthodes de calcul fiables, des visualisations claires et des recommandations adaptées, rendant l'analyse des options sur ces actions clés plus accessible et efficace.

2. Conception et organisation des données

Cette section décrit les objectifs principaux du projet ainsi que la structure de la base de données utilisée pour alimenter l'application, y compris les données stockées et les données consultées en temps réel.

2.1 Objectif

L'objectif de l'outil OptionLab est de :

- Faciliter la prise de décision pour les acheteurs et vendeurs d'options
- Offrir une visualisation interactive et en temps réel des cours des actions
- Simplifier la consultation et l'analyse des options disponibles sur des actions clés

2.2 Conception de la base de données

La base de données utilisée dans l'application a été conçue pour centraliser efficacement les informations relatives aux options sur les actions. Sa conception a suivi les étapes suivantes :

i) Identification des entités clés

Deux entités principales ont été identifiées :

- Ticker, représentant les symboles des actions
- Options, contenant les informations sur les options

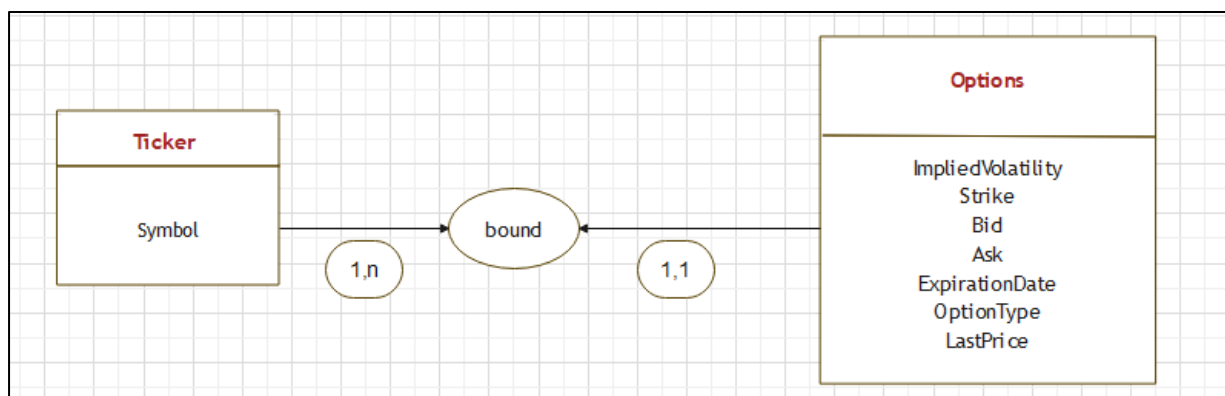
ii) Création du dictionnaire de données

Un dictionnaire de données a été élaboré pour définir de manière précise les attributs de chaque objet à conserver.

iii) Mise en place du modèle conceptuel de données (MCD)

Chaque Ticker est lié à une ou plusieurs Options par une relation "un-à-plusieurs", ce qui permet de structurer les données de manière cohérente et intuitive.

Figure 1 : Schéma du Modèle Conceptuel de Données



iv) **Organisation et stockage des données**

Initialement basé sur un modèle conceptuel relationnel, le projet a finalement adopté une structure centralisée où les données sont regroupées dans un grand tableau unique et stockées au format **Parquet** sur le cloud **SSP**. Cette décision a été motivée par plusieurs avantages :

- Une simplification de la structure.
- Une flexibilité accrue pour l'accès, la manipulation et l'intégration des données dans l'application.

2.3 Données en Temps Réels

En complément de la base de données, l'application intègre une fonctionnalité permettant de récupérer des données financières en temps réel, sans stockage. Cette fonctionnalité offre la possibilité de consulter les cours actuels des actions majeures, d'analyser des indicateurs techniques tels que les moyennes mobiles, et de visualiser de manière interactive les tendances du marché à l'aide de graphiques dynamiques.

3. Présentation et portée de l'application

Dans cette section, nous présentons la portée du projet, notamment les actions sélectionnées pour son développement, ainsi que les principales fonctionnalités offertes par l'application.

3.1 Actions sélectionnées : représentativité et pertinence

i) Sélection des actions

Le choix des actions incluses dans OptionLab garantit leur représentativité et pertinence pour les investisseurs : **AAPL (Apple), MSFT (Microsoft), TSLA (Tesla), JNJ (Johnson & Johnson), JPM (JPMorgan Chase & Co), XOM (Exxon Mobil), PG (Procter & Gamble), NVDA (Nvidia).**

Ces actions couvrent des secteurs variés : technologie, énergie, finance, biens de consommation et santé. Cette diversification répond à différents profils d'investisseurs.

ii) Critères de sélection

➤ Volatilité implicite :

- Les actions à forte volatilité (> 50%) (TSLA, NVDA) offrent des opportunités pour des stratégies dynamiques telles que les straddles.
- Les actions à faible volatilité (< 20%) (JNJ, PG) garantissent une stabilité, idéales pour des stratégies prudentes telles que la vente d'options couvertes.
- Les profils équilibrés (AAPL, MSFT) offrent une flexibilité stratégique avec des volatilités modérées.

➤ Volume et liquidité :

- AAPL et MSFT offrent une liquidité élevée pour des exécutions rapides.

➤ Diversification sectorielle :

- XOM (énergie), JPM (finance), complètent la représentation économique.

➤ Stabilité et dividendes :

- PG et JNJ attirent les investisseurs axés sur les revenus passifs.

➤ Potentiel d'innovation :

- AAPL, MSFT, NVDA illustrent des opportunités technologiques solides.

Ces choix permettent à OptionLab d'offrir des analyses adaptées aux besoins variés des utilisateurs.

3.2 Fonctionnalités principales

L'application propose plusieurs fonctionnalités pour répondre aux besoins des utilisateurs :

➤ **Consultation des options disponibles**

Les utilisateurs peuvent explorer les options disponibles sur les actions sélectionnées. Des filtres permettent de spécifier la date d'expiration, le prix d'exercice et le type d'option (Call/Put).

➤ **Visualisation en temps réel des cours des actions**

Cette fonctionnalité permet de suivre les cours des actions en direct et de bénéficier d'une visualisation claire des tendances.

➤ **Analyse des sensibilités des options (Greeks)**

Les sensibilités des options, telles que le Delta, Gamma, Vega, Theta, et Rho, sont calculées pour guider les décisions d'investissement.

➤ **Calcul et recommandations personnalisées**

Cette section offre des outils adaptés aux **vendeurs** et **acheteurs** d'options :

- Pour les vendeurs : L'application calcule la valeur théorique des options grâce au modèle LSMC (Least-Square Monte Carlo for American Options) et recommande de vendre si le prix de marché est supérieur. Si ce n'est pas le cas, il est conseillé d'attendre une hausse du prix.
- Pour les acheteurs : L'application identifie les options sous-évaluées en comparant leur valeur théorique et leur prix de marché. Les utilisateurs peuvent ainsi repérer les meilleures opportunités d'achat.

Les recommandations s'appuient sur des simulations Monte Carlo, des données actualisées et une interface interactive pour faciliter la prise de décision.

4. Architecture du projet

Cette section décrit l'organisation technique du projet, couvrant la structure du dépôt GitHub, la gestion des données et le développement de l'application.

4.1 Structure du dépôt GitHub

Le projet est organisé en plusieurs répertoires et fichiers clés pour une gestion efficace et une collaboration simplifiée. Voici la structure principale du dépôt :

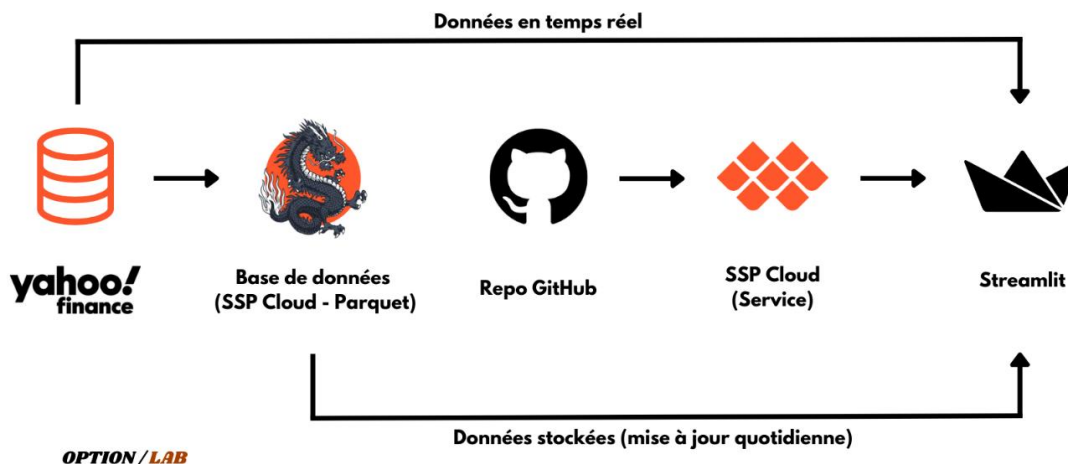
- **OptionLab.py** : Fichier principal contenant le code de l'application Streamlit.
- **importation.py** : Script dédié à l'importation et à la mise à jour des données depuis des sources externes (par exemple, l'API Yahoo Finance).
- **media** : Contient les fichiers multimédias, notamment la vidéo de présentation utilisée dans l'application et les ressources graphiques.
- **requirements.txt** : Liste des dépendances Python nécessaires pour exécuter l'application.
- **README.md** : Documentation de base pour présenter le projet, son installation et son utilisation.
- **.github/** : Répertoire contenant les workflows GitHub Actions, permettant d'automatiser certaines tâches (ex. : mise à jour des données via importation.py).
- **policy.json** : Fichier de configuration définissant les règles ou paramètres globaux.

4.2 Gestion des données et architecture globale

L'application utilise des données de l'**API Yahoo Finance**, via yfinance, réparties en deux catégories:

- **Données stockées** : Centralisées dans un fichier **Parquet** sur SSP Cloud, elles sont mises à jour quotidiennement (6h) ou lors des modifications sur GitHub.
- **Données en temps réel** : Directement récupérées via l'API pour des analyses et visualisations sans stockage.

Figure 2 : Architecture technique du projet



5. Méthodologie de calcul

Cette section détaille les approches techniques utilisées dans l'application pour les simulations et le calcul des sensibilités.

5.1 Pricing des options : Méthode de Longstaff-Schwartz (2001)

La valorisation des options américaines est rendue complexe par la possibilité d'exercer l'option à tout moment avant l'échéance. La méthode de Longstaff-Schwartz propose une approche par simulation pour résoudre ce problème, basée sur des régressions successives afin d'estimer la valeur de continuation.

➤ Etape 1 : Simulation des trajectoires de prix

On génère N trajectoires simulées de l'évolution des prix de l'actif sous-jacent sur M dates à l'aide d'un processus stochastique (mouvement brownien géométrique).

$$S_{t+1} = S_t \cdot e^{\left(r - \frac{\sigma^2}{2}\right)\Delta t + \sigma\sqrt{\Delta t}Z} \quad \text{Où } Z \sim N(0,1) \text{ et } \Delta t = T/M$$

Pour garantir la reproductibilité des résultats, un random seed est fixé au moment de la génération des trajectoires.

➤ Etape 2 : Calcul des payoffs

À chaque date t , le payoff immédiat est calculé pour toutes les trajectoires ($\max(S_t - K, 0)$ pour un call et $\max(K - S_t, 0)$ pour un put).

➤ Etape 3 : Estimation des valeurs de continuation par régression

À chaque date t , les valeurs de continuation sont estimées pour les trajectoires où l'option est "in-the-money" ($\text{Payoff}_t > 0$) :

Les valeurs futures actualisées $\text{Payoff}_{t+1} \cdot e^{-r\Delta t}$ sont régressées sur les prix S_t des trajectoires sélectionnées. Une régression polynomiale (par exemple, de degré 5) est utilisée pour modéliser la valeur de continuation $C(S_t)$ en fonction des prix.

➤ Etape 4 : Estimation des valeurs de continuation par régression

Pour chaque trajectoire, à chaque date t , on compare le payoff immédiat et la valeur de continuation. Si $Payoff_t > C(S_t)$, l'option est exercée. Sinon, elle est conservée.

➤ **Etape 5 : Retour en arrière**

À la date T , on fixe les cashflows comme étant égaux au payoff immédiat pour toutes les trajectoires simulées. Le processus est itératif : on recule dans le temps, depuis l'échéance (T) jusqu'à la première date ($t=1$), en mettant à jour les valeurs actualisées des trajectoires restantes.

➤ **Etape 6 : Estimation finale du prix de l'option**

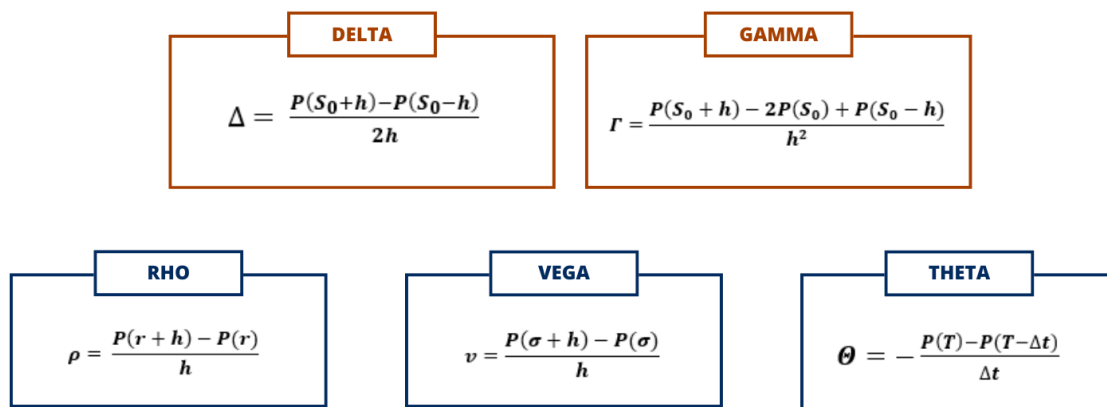
Le prix de l'option est obtenu en moyennant les valeurs actualisées pour toutes les trajectoires.

Pour les calculs, le taux sans risque utilisé correspond au rendement du **Bon du Trésor américain à 5 ans**, considéré comme représentatif. L'implémentation de cette méthode a été réalisée dans la classe Python `AmericanOptionsLSMC`, avec des méthodes telles que `_generate_price_paths`, `_calculate_payoffs`, `_backward_induction`, et `price`.

5.2 Méthodologie pour le calcul des Greeks

Les sensibilités des options, appelées Greeks, sont au nombre de cinq principales : Delta, Gamma, Vega, Theta, et Rho. Les formules d'approximation utilisées pour estimer ces sensibilités sont présentées dans le graphique ci-dessous :

Figure 3 : Formules d'estimation des greeks



Pour la mise en œuvre des approximations nous considérons une variation $h = S_0 \cdot 0,01$ et dans chaque cas, les prix nécessaires au calcul des Greeks sont obtenus à l'aide de la méthode de Least-Squares Monte Carlo (LSMC), implémentée dans la classe AmericanOptionsLSMC. Cette approche permet une estimation précise des valeurs grâce aux simulations réalisées pour les options américaines.

6. Avantages de l'application OptionLab et pistes d'amélioration

OptionLab centralise efficacement les données et offre une analyse avancée des options grâce à des outils interactifs et des calculs précis basés sur la méthode LSMC. L'interface intuitive permet une exploration facile, avec des visualisations claires et des sensibilités calculées en temps réel.

Pour aller plus loin, l'application pourrait élargir sa couverture d'actifs financiers et enrichir les options de personnalisation pour mieux répondre aux besoins des utilisateurs.