analysis of the predictor of a volatility surface by machine learning

Valentin Lourme

Sciences de la Décision et Management des Risques

2022-2023

Analyse de la prédiction d'une Nappe de Volatilité par machine Learning

Lourme Valentin

UNIVERSITÉ PARIS 1 PANTHÉON SORBONNE
IAE PARIS
SORBONNE BUSINESS SCHOOL

Arts et Métiers
Sciences et Technologies

Notice Bibliographique

Auteur/e : Lourme Valentin

Matricule : 2509072061R

Année : 2023

Campus de rattachement : Paris

Réf projet : PA-C

Type de document : mémoire de master recherche

Directeur/e de recherche : Dantan Jean Yves

Titre de l'article de recherche : Analyse de la prédiction d'une nappe de volatilité par Machine Learning

Mots clés (5 maximum) : Machine Learning, Volatilité, Nappe, Interpolation, Prédiction

Résumé en français (10 lignes) :

Cette étude vise à comparer deux approches d'évaluation des points d'une nappe de volatilité. La première approche utilisée est l'interpolation par spline cubique, tandis que la seconde approche est un algorithme de machine Learning, le XGBoost. Cette comparaison a pour but de définir le cas d'utilisation ou l'algorithme de machine Learning XGBoost est plus adapté par rapport au spline cubique. La comparaison entre les deux approches est mesurée avec l'erreur entre la volatilité mesurée et la volatilité interpolée ou prédite. L'interpolation par spline cubique nécessite les données de volatilité au jour de l'étude pour que l'interpolation soit réalisée. L'algorithme de Machine Learning XGBoost va s'entrainer sur des données historiques pour prédire la valeur de volatilité au jour de l'étude.

Keywords (5 maximum): Machine Learning, Volatility, Surface, Interpolation, Prediction

Abstract (10 lignes):

The purpose of this study is to compare two approaches to assessing the points of a volatility layer. The first approach used is cubic spline interpolation, while the second approach is a machine learning algorithm, the XGBoost. The purpose of this comparison is to define the use case where the XGBoost Learning machine algorithm is more suitable compared to the cubic spline. The comparison between the two approaches is measured with the error between the measured volatility and the interpolated or predicted volatility. Cubic spline interpolation requires volatility data on the day of the study for interpolation to occur. The XGBoost Machine Learning algorithm will train on historical data to predict the volatility value on the day of the study.

J'atteste que ce travail est personnel, cite en référence toutes les sources utilisées et ne comporte pas de plagiat. Le nom de l'auteur du texte vaut signature.

NOM : Lourme Valentin

Date : 21/06/2023

Remerciements

Pour ce travail, je tiens à remercier mon professeur Jean-Yves Dantan, ainsi que mes collègues Mezouar Nassim, Le Razer Yannick, Brik Réda, Huang Men, Perelman Paul et Garcin Vincent, qui m'ont aidé et guidé dans la réalisation du plan de cette revue et qui m'ont apporté des pistes d'étude pour étudier ce sujet. Je tiens également à remercier l'ensemble du corps enseignant du master qui m'a apporté une initiation au monde de la finance ainsi qu'à l'ensemble des membres de l'équipe IPV Equity de Natixis pour leur temps et toutes les explications sur le monde de la finance de Marché.

Abstract structuré

Chaque rubrique doit comporter au maximum 5 lignes.

Nom de la revue : Analyse de la prédiction d'une nappe de volatilité par Machine Learning

Introduction (thème de l'article) : Réalisation d'une étude qui compare les méthodes d'interpolation traditionnelles avec les algorithmes de Machine Learning pour prédire les données manquantes dans les nappes de volatilité. L'objectif est d'améliorer les estimations de volatilité et les méthodes de valorisation des produits financiers.

Question de recherche : Les Méthodes de Machine Learning sont-elles adaptées à la construction d'une surface de volatilité qui ne soit pas arbitrable et qui permettent une estimation en accord avec les évènements présents sur le marché ?

Méthode : Comparaison entre la prédiction du Machine Learning et l'interpolation par spline Cubique. Récupération des données de nappes de Volatilité sur les mois de Mars, Avril et Mai 2023. Entrainement du modèle de Machine Learning le plus adapté (XGBoost) sur les données historiques et comparaison des deux approches sur des points manquant d'une nappe de volatilité en date du 1er juin 2023

Résultats : Obtention de 4 courbes de comparaison entre le modèle de Machine Learning et l'interpolation par spline cubiques. Ces courbes sont étudiées pour des valeurs de strikes autour de 100%. L'erreur entre chaque modèle et les données de volatilité observées permet de mesurer l'écart de performance entre les deux approches.

Conclusion (synthèse de l'article) : Le modèle de machine learning (XGBoost) est recommandé pour des valeurs de maturités courtes et des strikes proches de 100%, tandis que l'interpolation par spline cubique est préférable pour des maturités plus longues. Cependant, les résultats dépendent des données d'entrée, et le modèle de machine learning peut souffrir de sur-apprentissage.

I. Introduction

L'avènement des nappes de volatilité dans le domaine de la finance de marché est étroitement lié à une avancée révolutionnaire dans la valorisation des produits dérivés : le modèle Black-Scholes. Développé par Fischer Black et Myron Scholes dans les années 1970, ce modèle a introduit une nouvelle approche mathématique pour évaluer les options et a apporté les bases de la gestion moderne des risques (Black et Scholes, 1973).

Le modèle Black-Scholes repose sur l'hypothèse selon laquelle la volatilité du prix d'un actif financier est constante et connue. Cependant, dans la réalité, la volatilité est une variable dynamique et peut varier au fil du temps. La prise en compte de cette réalité a conduit à l'émergence des nappes de volatilité (M. Britten-Jones et A. Neuberger, 2000).

Les nappes de volatilité sont des outils graphiques qui illustrent la relation entre la volatilité implicite, les prix d'exercice et les échéances des options. Elles permettent aux professionnels de la finance d'appréhender la structure et la dynamique de la volatilité sur une gamme d'échéances et de niveaux de prix. Ainsi, les nappes de volatilité fournissent une vision plus complète et plus précise de la volatilité que la simple considération d'une valeur constante.

En reconnaissant l'importance de la volatilité comme facteur clé dans la détermination des prix des produits dérivés et dans la gestion des risques, les acteurs de la finance de marché ont commencé à utiliser les nappes de volatilité pour améliorer leurs modèles d'évaluation et leurs stratégies de négociation. Ces nappes offrent une perspective plus nuancée et plus réaliste de la volatilité, permettant ainsi une meilleure estimation des prix et des risques associés aux options d'achat et de vente (D. Backus et al., 1997).

Les nappes de volatilité peuvent cependant être incomplètes en raison d'un manque d'information. Dans ce cas, les méthodes d'interpolations spline cubique ou LSVi sont généralement utilisées pour prédire les données n'ayant pas pu être récupérées. Une étude doit être menée spécifiquement sur ces nappes qui peuvent être comblées et prédites, en comparant les approches traditionnelles d'interpolation avec les avancées récentes des algorithmes de Machine Learning. En explorant ces approches, nous cherchons à améliorer la précision des estimations de la volatilité.

Cette étude permet de mieux comprendre les avantages et les limites des deux approches, ainsi que leur pertinence dans le contexte financier actuel. En évaluant la capacité des algorithmes de Machine Learning à prédire les données manquantes dans les nappes de volatilité, nous espérons contribuer à l'amélioration des méthodes de valorisation des produits financiers et à la prise de décisions plus éclairées dans le domaine de la finance.

II. Revue de littérature

Un ensemble de techniques de prédiction a été testé sur le jeu de données via une technique de la librairie automl (lazypredict), Les techniques basées sur les arbres de décisions ont eu une meilleure précision de prédiction, de ce fait, uniquement ces techniques sont détaillées dans le chapitre suivant, ainsi que les techniques de références, splines cubique et LSVi

A. Le Machine Learning

Le Machine Learning, ou apprentissage automatique, est un domaine de l'intelligence artificielle qui vise à développer des algorithmes capables d'apprendre à partir de données et d'effectuer des tâches sans être explicitement programmés. En 1986, le chercheur Ross Quinlan développa un modèle basé sur des arbres de décisions pour effectuer une classification supervisée des données. Ce modèle constitue la base de développement des algorithmes de Random Forest (L. Breiman, 2001).

1. Les Arbres de décisions

À l'origine, les arbres de décision ont été conçus pour résoudre des problèmes de classification. Cependant, ils ont depuis été adaptés pour traiter des problèmes de régression. En utilisant un ensemble de données mesurables et normalisées, un algorithme basé sur les arbres de décision cherche à créer des nœuds de décision afin de diviser de manière optimale les données d'entrée et prédire une valeur y. Les arbres de régression cherchent à prédire une valriable y par rapport à un jeu de données d'entrée X. La moyenne $\bar{y}$ est calculée sur l'ensemble des données y. Les données sont classifiées en fonction des nœuds ou critères de décision, puis un calcul de la moyenne pour chaque sous-catégorie est effectué. Ainsi, lorsqu'une nouvelle donnée est étudiée, elle est classifiée en fonction des différents nœuds de décision, et la valeur prédite est la moyenne de la sous-catégorie à laquelle cette donnée appartient (Maimon et al., 2005). À chaque nœud de décision, un indice de pureté appelé MSE (Mean Squared Error) est calculé pour évaluer la précision de la régression pour ce nœud.

$$MSE^2 = \sum_{i \in Noeuds} |\bar{y} - y_i|^2$$

La fiabilité de l'estimation des valeurs y par l'arbre de régression augmente à mesure que l'erreur moyenne de prédiction (MSE) diminue. Une fonction de coût est attribuée à chaque nœud, et en ajustant les conditions de chaque nœud, il est possible de réduire au minimum cette fonction de coût.

Cette fonction notée J a la forme suivante :

Pour un nœud de décision d'indice k donné, en notant $\omega_{True}, \omega_{False},$ les proportions de population présentes respectivement dans la branche ou la condition du nœud k est vérifiée et dans celle ou la condition du nœud k n'est pas vérifiée, et $MSE_{True}, MSE_{False},$ les erreurs moyennes de prédiction respectives des branches ou la condition du nœud k est vérifiée et ou cette condition n'est pas vérifiée, on obtient

$$J(k) = \omega_{True} * MSE_{True} + \omega_{False} * MSE_{False}$$

En ayant plusieurs règles de décision et en calculant le cout de chacune des règles, on cherche celle qui minimise la fonction J. L'avantage majeur des arbres de décisions réside dans leur

simplicité d'utilisation et dans leur facilité d'apprentissage. Cependant, les arbres de décisions sont très sensibles à des petites variations dans le jeu de données d'entrée et ont un problème de sur-apprentissage si trop de branches sont créées. Afin de surmonter ces limites, l'algorithme de la forêt aléatoire est apparu pour combler les lacunes des arbres de décisions.

2. L'algorithme de Random Forest

L'algorithme de Random Forest utilise l'« ensemble Learning » pour effectuer une prédiction. Les méthodes d'« ensemble Learning » utilisent plusieurs algorithmes d'apprentissage et intègrent les résultats de ces modèles pour améliorer les performances prédictives par rapport aux modèles individuels. Les arbres de décision qui composent une forêt aléatoire doivent satisfaire une condition importante : le critère de diversité de la population, c'est-à-dire qu'ils doivent avoir des nœuds de décision différents pour que les prédictions varient d'un arbre à l'autre. L'algorithme de la Random Forest fonctionne en créant plusieurs arbres de décision avec un faible nombre de nœuds qui s'entraînent individuellement sur des partitions aléatoires de l'ensemble de données (M.Belgiu et L. Dragut, 2016). Cette sélection aléatoire vise à maximiser la diversité des données. La prédiction de la forêt aléatoire est obtenue en prenant la moyenne des prédictions des arbres de décision. La limitation majeure de l'algorithme de la Random Forest réside dans sa partition aléatoire des données. Un nouvel algorithme va alors permettre de combler cette lacune : Le gradient Boosting Tree.

3. Le gradient Boosting Tree et le XGBoost

Le XGBoost est une implémentation de l'algorithme de gradient boosting tree. Cet algorithme commence par calculer la moyenne des données notée $\bar{y}$. À partir de cette moyenne, il calcule l'erreur entre la moyenne et chaque valeur y de la série. Ensuite, il cherche à prédire l'erreur (également appelée résidu). En utilisant à la fois la moyenne et la prédiction de l'erreur, il arrive à une nouvelle prédiction de la donnée y (T. Chen et C. Guestrin, 2016). Ce processus est répété jusqu'à ce que la prédiction de l'erreur fournisse une erreur globale acceptable. Le XGBoost présente tous les avantages des arbres de décision et du random forest en rajoutant le fait que le modèle pondère chacun des arbres de décision. Il apparait ainsi comme un modèle robuste de machine Learning que nous nous proposons d'utiliser dans le cadre de l'étude de prédiction des données manquantes sur une nappe de volatilité. L'objectif de cette étude consiste à utiliser l'algorithme XGBoost pour déterminer les données de volatilité manquantes, afin de comparer les prédictions générées par l'apprentissage automatique avec la méthode actuelle d'interpolation par spline cubique.

B. Calibration de la nappe de volatilité par le modèle LSVi

Le modèle 'Local Stochastic Volatility implied' (LSVi) est un modèle qui estime la volatilité d'un actif sous-jacent dans le temps. Dans ce modèle, la volatilité est considérée comme une variable aléatoire qui peut changer au cours du temps. Le modèle 'Local Stochastic Volatility implied' suppose que la volatilité suit un processus stochastique (M. Lorig et al, 2015). Ce modèle nécessite plusieurs paramètres pour être calibrer correctement. Actuellement, les experts utilisent ce modèle pour estimer la volatilité implicite des options par rapport au marché. Lorsque les nappes de volatilité sont calculées par le modèle LSVi, il est alors possible d'interpoler certains points de volatilité par la méthode du spline cubique.

C. L’interpolation par spline cubique

Lorsque les nappes de volatilité ont pu être estimées à partir du modèle LSVi, il est possible d’interpoler les données de volatilité en certains points précis par la méthode du spline cubique. Cette approche consiste à construire une courbe lisse à partir des points de volatilité disponibles, en utilisant des polynômes de degré trois (cubiques) entre les points de données adjacents. L'interpolation par spline cubique offre plusieurs avantages. Tout d'abord, elle permet de combler les lacunes dans les données de volatilité en fournissant des estimations pour les points manquants. Cette méthode est particulièrement utile lorsque les données disponibles sont dispersées ou incomplètes. En utilisant des polynômes cubiques, l'interpolation par spline permet de capturer les variations complexes de la volatilité dans les nappes. Elle produit ainsi une courbe lisse qui s'ajuste mieux aux données disponibles, ce qui conduit à une estimation plus précise de la volatilité.

Lors de la résolution d’un problème d'interpolation, le but est de trouver une fonction S qui passe par n points donnés. Néanmoins, le phénomène de Runge se présente lorsque le nombre de points à interpoler augmente, notamment lors de l'application de l'interpolation de Lagrange (J.P. Demailly, 2006). En effet, dans certains cas les polynômes interpolateurs de Lagrange ne constituent pas une approche d’approximation efficace lorsque le nombre de points n augmente.

Pour éviter le phénomène de Runge, une approche alternative serait d'utiliser plusieurs polynômes de degré inférieur ou égal à trois sur chaque intervalle entre les points d'interpolation, puis de les assembler pour définir une fonction S sur l'intervalle complet. La fonction S doit satisfaire des critères de continuité et de dérivabilité spécifiques. Cela implique que le degré des polynômes interpolateurs augmente. L'objectif est de trouver un compromis optimal, et celui qui est généralement adopté est celui des splines cubiques.

Soient $K_1, \ldots, K_n$, $\sigma_1, \ldots, \sigma_n$ des nombres réels. On appelle Spline cubique associé à la famille (K_i, σ_i) toute fonction S de classe C^2 polynomiale de degré au plus trois sur chacun des intervalles $[K_i, K_{i+1}[$ et telle que pour tout i dans {1, …, n} $S(K_i) = \sigma_i$.

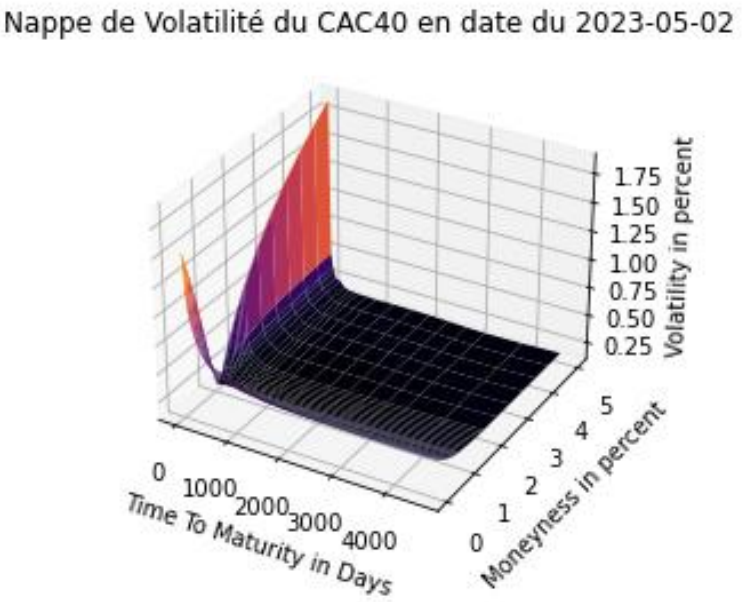
Une telle fonction S nécessite quatre coefficients à déterminer sur chacun des n-1 intervalles ce qui fait un total de 4n-4 coefficients à définir. Afin de déterminer ces coefficients, on applique des conditions aux limites aux points à interpoler sur chaque intervalle, ce qui donne lieu à 2n-2 équations. De plus, nous souhaitons que la dérivée et la dérivée seconde de S soient continues en K_i, pour i allant de 2 à n-1, ce qui ajoute 2n-4 conditions. Pour les deux dernières conditions, nous imposons une contrainte de minimisation d'énergie, à savoir que les dérivées seconde aux points K_1 et K_n soient nulles.

Dans le cadre de cette étude, l'algorithme XGBoost sera appliqué pour estimer les valeurs manquantes de volatilité. En comparant les résultats obtenus par l'algorithme XGBoost avec les estimations produites par l'interpolation par spline cubique, nous serons en mesure d'évaluer la performance et l'exactitude des deux approches.

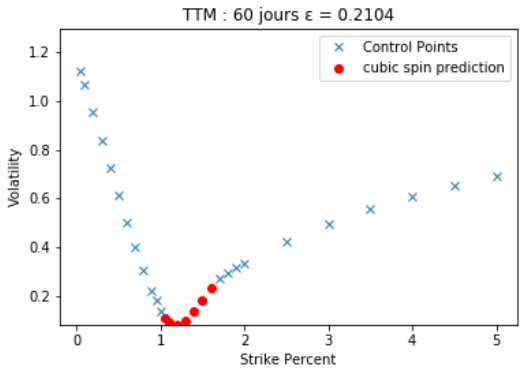
III. Données

Afin de mener à bien cette étude comparative, il est essentiel de définir les données à récupérer ainsi que la plage d'historique spécifiée. J'ai pu collecter quotidiennement les données des nappes de volatilité pour l'indice CAC 40 pendant les mois de mars, avril et mai, correspondant à la volatilité observée sur le marché à la clôture. À titre d'exemple, voici la structure générale d'une nappe de volatilité ainsi que l'affichage de la nappe en date du 2 mai :

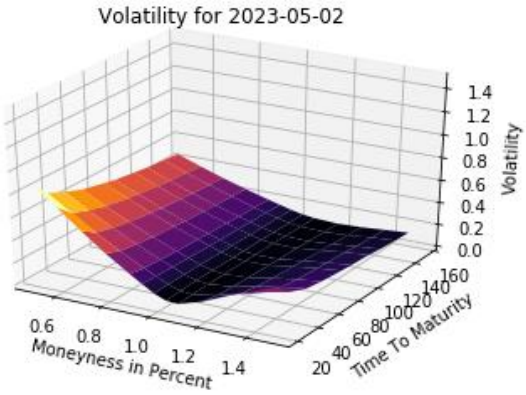
Valuation Date	Maturity	Time_To_Maturity	Forward	Strike_Percent	Volatility
02/05/2023	03/05/2023	1	6999.05263948262	0.05	1.512157902673097
02/05/2023	03/05/2023	1	6999.05263948262	0.1	1.457077305072727
02/05/2023	03/05/2023	1	6999.05263948262	0.2	1.3421485762516172
02/05/2023	03/05/2023	1	6999.05263948262	0.3	1.2188328373909572



Le "Strike Percent" représente la valeur du strike en pourcentage par rapport au forward. Ce strike Percent prend des valeurs comprises dans l'intervalle [0.05, 5]. Le « Time To Maturity » est la maturité des options portant sur le CAC40 mesurée en jours. Elle est calculée par différence entre la « Valuation Date » et la « Maturity ». Le 'Time To Maturity' varie selon les jours d'observations et elle prend souvent ses valeurs dans l'intervalle [1, 7200]. « La valuation Date » correspond au jour d'observation de la nappe et enfin le Forward correspond au prix Forward du contrat. Ces nappes récupérées sont donc assez complètes en termes d'informations fournies. Nous allons donc nous concentrer uniquement sur les intervalles de 'Strike Percent' et de 'Time To Maturity' sur lesquelles l'interpolation par spline cubique s'avère moins performante que sur les autres intervalles.



On remarque que pour les maturités inférieures à 360 jours, pour des valeurs d'options à la monnaie, ou le strike en pourcent est proche de 1, on observe un écart entre les valeurs interpolées et les valeurs réelles plus important que sur le reste de la courbe lorsque sept points successifs de la courbe sont supprimés. Nous décidons donc de focaliser notre étude sur des maturités comprises entre 15 jours et 1an (360 jours) et pour des valeurs de strike en pourcent comprises dans l'intervalle [0.5, 1.5]. Ainsi la nouvelle nappe de notre étude aura la forme suivante :



Le jeu de données de volatilité considéré est celui en date du 1er juin 2023. Ces valeurs mesurent l'écart de performance entre les prédictions de l'algorithme de Machine Learning XGBoost et l'interpolation du spline cubique.

Pour filtrer les valeurs aberrantes dans les données récupérées, j'ai vérifié que les données historiques récupérées sur les mois de mars, Avril et Mai étaient non-arbitrables. Ainsi dans le cadre de cette étude, l'objectif était de vérifier si les valeurs de volatilité des options hors de la monnaie étaient plus élevées que celles des options à la monnaie. Si ce n'est pas le cas, une stratégie d'arbitrage peut s'appliquer, la stratégie Call-Spread. Nous avons aussi cherché à vérifier si, pour les options à la monnaie, la volatilité augmentait avec la maturité. Dans le cas inverse, une stratégie d'arbitrage peut s'appliquer, il s'agit du Calendar-Spread. Mes données récupérées vérifiaient les deux conditions de non-arbitrage et donc ne contenaient pas de valeurs aberrantes pour l'entrainement du modèle de machine Learning. Ainsi dans la suite de cette étude, nous allons supposer que les données d'étude sont non-arbitrables.

IV. Méthodologie de comparaison

Les données de marché sont considérées comme non-arbitrables et adaptées à l'entraînement d'un modèle de Machine Learning. Pour la comparaison entre le modèle de Machine Learning et l'interpolation par spline cubique, ce sont les données en date du 1er juin 2023 qui serviront de référence pour évaluer les performances des deux méthodes.

A. Limitation de l'interpolation par spline cubique

La méthode d'interpolation par spline cubique est utilisée pour compléter les données manquantes dans les nappes de volatilité. Cependant, certaines limitations sont constatées lorsque l'interpolation par spline cubique tente d'interpoler plus de trois points consécutifs en fonction du strike pour une valeur de maturité fixée. De plus, l'erreur entre les points d'interpolation et les points de volatilité réellement observés est plus prononcée pour les options à la monnaie avec des maturités inférieures à 1 an. Cette observation peut être expliquée par la forme spécifique des courbes $\sigma(K \mid T = T_j)$ avec T_j la maturité comprise entre une journée et une année. L'Hypothèse principale formulée est alors la suivante :

L'interpolation par spline cubique n'est pas remise en question lorsque moins de deux points consécutifs sont manquants à la surface de volatilité.

Il reste alors à définir un intervalle de strike, un intervalle de maturité et un nombre de points consécutifs pour lesquels la méthode d'interpolation par spline cubique pourrait éventuellement être remplacée par un algorithme de Machine Learning. Si cet algorithme est correctement entraîné, il pourrait capturer la dynamique de la surface de volatilité en ces points spécifiques.

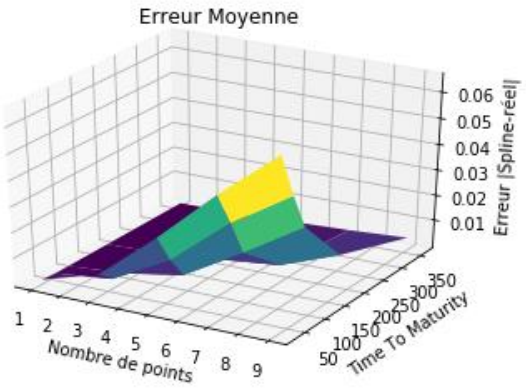
B. Domaine de définition de l'étude

D'après les analyses de prédictions par spline cubique, ce sont les valeurs de volatilité pour les options à la monnaie qui sont les plus écartées vis-à-vis des valeurs réelles. L'analyse se focalise donc pour des strike en pourcent, proche de l'ATM. Pour le choix de la maturité et du nombre de points à prendre pour l'étude, l'erreur moyenne agit comme indicateur de la prédiction par spline cubique en fonction du nombre de points et de la maturité (en jours) :

$$Err(n, TTM) = \frac{1}{n} \sum_{k=1}^{n} |y_{k,T=TTM} - y_{\widehat{k,T=TTM}}|$$

Avec n qui représente le nombre de points enlevés autour du pourcentage de strike 100% (option à la monnaie), TTM correspond à la maturité de l'option, $y_{k,T=TTM}$ représente la valeur réelle pour la valeur de strike K et de maturité TTM et $y_{\widehat{k,T=TTM}}$ représente la valeur prédite par le spline cubique pour un strike K et une maturité TTM (en jours)

Le graphique suivant montre l'évolution de l'erreur en fonction du nombre de points enlevés et du 'Time To Maturity'.



Une augmentation de l'erreur moyenne à mesure que le nombre de points supprimés augmente est observée, tandis qu'elle diminue lorsque la maturité augmente. Nous choisissons de nous concentrer sur le cas spécifique où sept points sont supprimés et nous fixons la "Time To Maturity" à 60 jours (deux mois), car l'erreur moyenne atteint 0,02 points de volatilité, ce qui semble être une valeur critique pouvant être réduite.

C. Calibration du Machine Learning XGBoost

Au regard de l'interpolation par spline cubique, Les performances de l'interpolation deviennent moins bonnes pour des valeurs de volatilité proche de l'ATM et pour des maturés autour de 60 jours, pour un nombre de sept points consécutifs supprimés. Le XGBoost est l'algorithme de Machine Learning défini pour approximer les points de volatilité. Les résultats du Machine Learning sont ensuite comparé avec le spline cubique. La mesure utilisée pour comparer les écarts entre l'interpolation du spline cubique et la prédiction par machine learning est l'erreur moyenne (Mean Absolute Error) par rapport aux données de marché observées.

Pour pouvoir entrainer le modèle XGBoost, il est nécessaire de lui fournir des données d'entrée qui soient mesurables pour qu'elles puissent être analysées. Voici une partie de la structure générale des données d'entrées communiqués au modèle XGBoost :

Strike_Percent	Maturity_In_Days	min_volatility_last_7days	min_volatility_last_21days	min_volatility_last_42days	max_volatility_last_7days
0,05	1	1,475637	0,721102	0,721102	1,7967
0,05	3		1,400097	1,400097	
0,05	4		1,262067	1,262067	

Ce sont les données qui vont composer le vecteur d'entrée X. Elles contiennent les colonnes 'Strike Percent', 'Maturity In Days', ainsi que les valeurs minimales, maximales, la moyenne et l'écart-type sur les sept derniers jours, sur les 21 derniers jours ainsi que sur les 42 derniers jours. Les sept derniers jours permettent d'avoir une information à court terme des valeurs de volatilité. Les 21 derniers jours permettent d'avoir cette information à moyen terme et les 42 derniers jours nous donnent ces informations à long terme. Enfin, nous fournissons également au modèle les valeurs de volatilité en date du 1er juin 2023 qui ont permis la calibration la méthode des splines cubique. Ainsi, de cette manière, les deux modèles ont en entrée des informations similaires. Le XGBoost est un modèle de régression qui va fournir un vecteur y contenant les valeurs prédites par ce modèle.

D. Comparaison des deux modèles sur le domaine de définition de l'étude

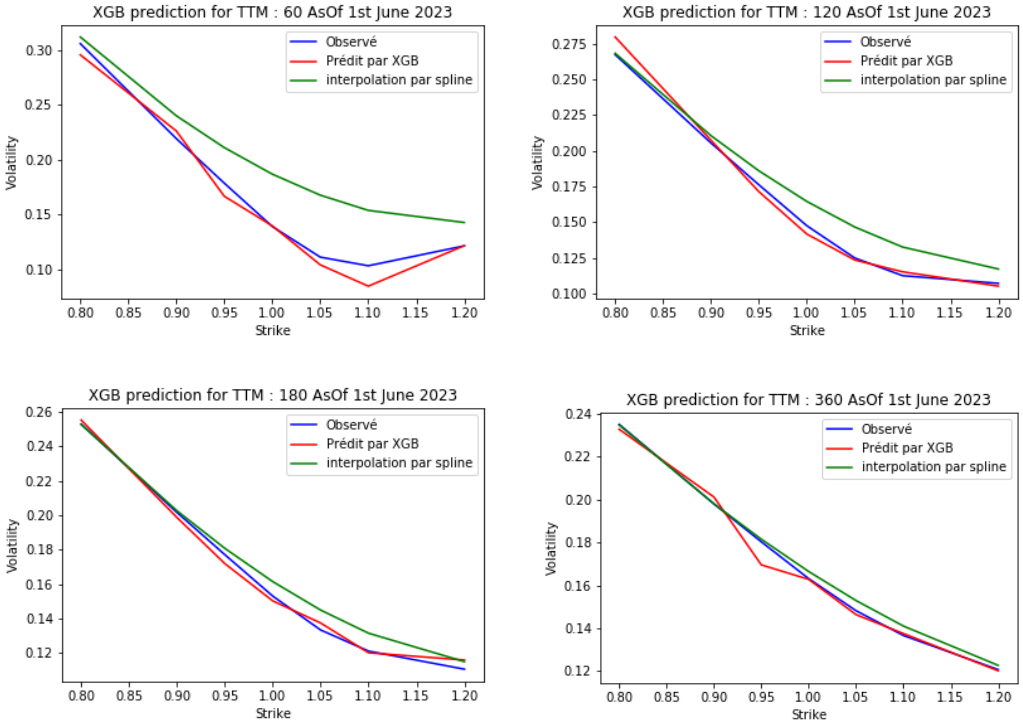
A partir des résultats de prédiction de l'algorithme « XGBoost regression » et des résultats de l'interpolation par spline cubique sur des données de volatilité basé sur des options vanilles européennes de maturité 60 jours (2 mois) pour des valeurs de strike comprises entre 80% et 120% (à la monnaie), pour sept points supprimes à estimer, nous pouvons calculer l'erreur de l'estimation de chacun de ces deux modèles par rapport à la volatilité observée pour en déduire la performance de chacun de ces deux modèle.

L'erreur entre le modèle XGBoost et les données de volatilité observée est calculée selon la formule suivante :

$$Err(n, TTM) = \frac{1}{n} \sum_{k=1}^{n} |\sigma_{k,T=TTM} - \sigma_{\widehat{k,T=TTM}}|$$

Avec n=7, le strike K prenant ses valeurs dans l'ensemble {80%, 90%, 95%, 100%, 105%, 110%, 120%}. $\sigma_{k,T=TTM}$ est la valeur de volatilité observée sur le marché et $\sigma_{\widehat{k,T=TTM}}$ est la valeur estimée par le modèle XGBoost. La même formule s'applique pour l'interpolation par spline cubique en modifiant la valeur $\sigma_{\widehat{k,T=TTM}}$ par celle estimée par l'interpolation par spline cubique.

V. Résultats

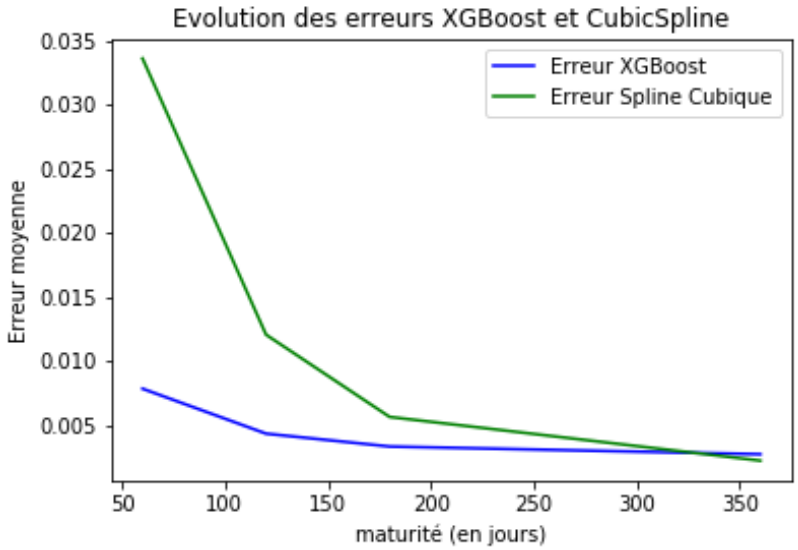


Sur les figures suivantes, nous avons tracé la volatilité observée sur le marché, la volatilité prédite par le modèle XGBoost ainsi que la volatilité interpolée par spline cubique en fonction du strike pour quatre maturités différentes. Les maturités sont comprises dans l'ensemble {60 jours, 120 jours, 180 jours, 360 jours}. Les erreurs entre les modèles et les valeurs observées sont référencées dans le tableau suivant :

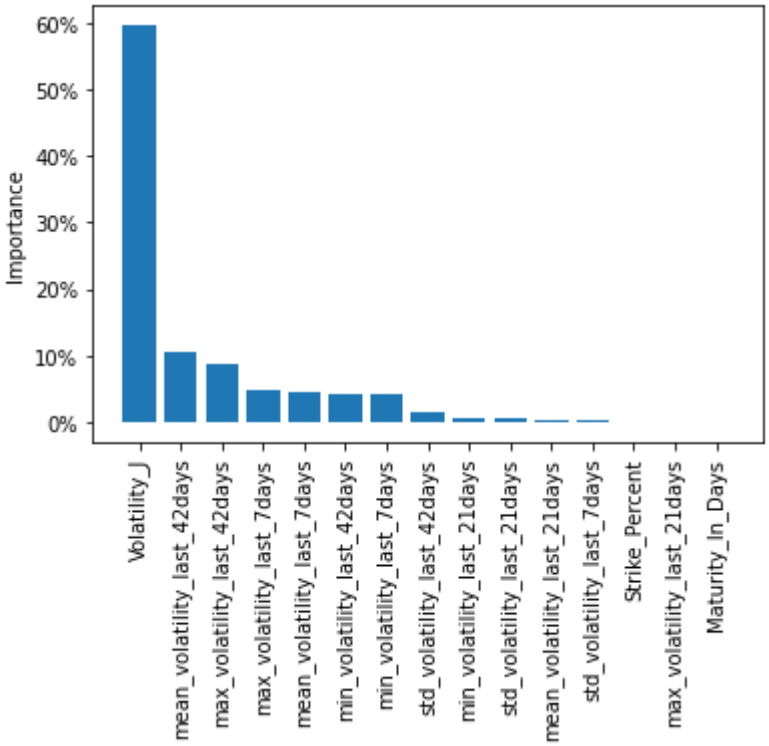
maturité (en jours)	60	120	180	360
Erreur XGBoost / Observée	0,0079	0,0044	0,0034	0,0028
Erreur Spline Cubique / Observée	0,0336	0,0121	0,0057	0,0023

Nous pouvons observer sur la première ligne du tableau la maturité en jours ainsi que les quatre maturités caractéristiques de notre étude {60, 120, 180, 360}. Sur la deuxième ligne apparait l'erreur entre le modèle XGBoost et les valeurs de volatilité observées sur le marché pour chacune des maturités. Enfin, sur la troisième ligne figure l'erreur entre l'interpolation par spline cubique et les valeurs de volatilité du marché. Nous pouvons observer directement le modèle qui approche le mieux les données de marché avec la nuance de couleur pour chaque maturité. Chacune des valeurs de ce tableau représente un écart de volatilité moyen.

L’évolution de l’erreur pour chacun des deux modèles est obtenue à l’aide du graphique suivant :



Par ailleurs, l’histogramme suivant donne la répartition en pourcentage de l’importance de chacune des caractéristiques. Nous observons que la volatilité au jour d’étude occupe une place prépondérante pour la prédiction



VI. Discussions

A partir des quatre courbes que nous observons, nous remarquons qu'à mesure que le temps à maturité augmente, l'erreur entre les données de volatilité observées sur le marché et l'interpolation par spline cubique diminue. Il en est de même pour l'erreur entre les prédictions XGBoost et les données observées. Nous remarquons que pour les maturités courtes {60 jours, 120 jours, 180 jours} l'algorithme de machine Learning XGBoost est plus performant comparé à l'interpolation par spline cubique. En revanche, lorsque la maturité dépasse 360 jours, l'interpolation par spline cubique a une valeur de MAE plus faible. L'algorithme XGBoost arrive à capturer les tendances du marché pour toutes les courbes de maturités différentes, en particulier celles ou les strikes sont proches de 100%. Dans cette zone, la volatilité ne suit pas de tendance particulière et c'est pourquoi le spline cubique rencontre des difficultés pour interpoler les données en ces points particulier. Er revanche, pour des maturités longues (supérieures à 1 an) la volatilité décrit une courbe hyperbolique que le spline cubique arrive à bien capturer, c'est pourquoi sur ces valeurs de maturités, l'erreur MAE du spline cubique est plus faible que l'erreur MAE du XGBoost.

Par ailleurs, l'histogramme représentant l'importance des caractéristiques fournies en entrées pour l'entrainement du XGBoost montre que la volatilité au jour J représente près de 60% d'importance pour la prédiction des données manquantes, ensuite, la moyenne et la valeur maximale à long terme (42 jours) de volatilité représentent chacune 10% d'importance pour la calibration du modèle.

En ce qui concerne l'algorithme de XGBoost, ce dernier réalise des prédictions avec une erreur plutôt faible, cependant, étant donné que l'erreur mesuré sur le jeu d'entrainement est négligeable devant celle mesurée sur le jeu de test, nous pouvons considérer que le modèle XGBoost rencontre un problème de sur-apprentissage.

VII. Conclusion

Pour conclure, Le modèle de machine Learning est à privilégier pour des valeurs de maturités faibles, inférieures à 1 an, pour des valeurs de strike autour de 100% tandis que l'interpolation par spline cubique est plus avantageuse pour des maturités élevées qui sont supérieures à un an en raison de l'allure hyperbolique de la courbe de volatilité. Cependant, ces résultats dépendent des données fournies en entrée. En effet, Le modèle s'est entrainé sur des données des trois mois de mars, Avril et Mai. Cet entrainement sur ces données spécifiques est à remettre en question au regard des évènements financiers majeurs qu'il y a pu y avoir sur le marché (faillite de la Silicon Valley Bank) pendant cette période. De plus, les caractéristiques que nous avons fournies au modèle à savoir la moyenne, l'écart-type, la valeur minimale et la valeur maximale sont critiquables car d'autres caractéristiques descriptives auraient pu être prises à la place, comme la tendance sur 7 jours. De plus le modèle de machine Learning est sujet à une problématique de sur-apprentissage.

VIII. Références

David Backus, Silverio Foresti, Kai Li et Liuren Wu « Accounting for biases in Black-Scholes ». *CRIF Working Paper series* (25 novembre 1997)

M. Britten-Jones et A. Neuberger « Option prices, implied price processes, and stochastic volatility ». The journal of Finance (Avril 2000): Volume 55 Issue 2: 839-866

F. Black et M. Scholes « The pricing of options and corporate liabilities ». *Journal of political economy* (1973): 637–654

J. Ross Quinlan, *Machine Learning*, 1986, « Induction of decision trees », p. 81-106

Leo Breiman, « Random Forests », *Machine Learning*, vol. 45, nº 1, 2001, p. 5–32

Maimon, Oded; Rokach, Lior. *Data Mining and Knowledge Discovery Handbook Decision Trees* (2005).

Mariana Belgiu, Lucian Dragut, Random *Forest in remote sensing: A review of applications and future directions* (2016)

Tianqi Chen, Carlos Guestrin, *XGBoost: A Scalable Tree Boosting System* (2016)

JP Demailly : *Analyse numérique et équations différentielles* (2006)

Matthiew Lorig, Stefano Pagliarani, Andrea Pascucci, *Explicit implied volatilities for multifactor local stochastic volatility models,* Mathematical Finance (2015) 1-35