Advanced Programming 2025

Machine Learning for pricing Apple call options

Final Project Report

Axel Chapignac
axel.chapignac@unil.ch
Student ID: 1287516

January 6, 2026

Abstract

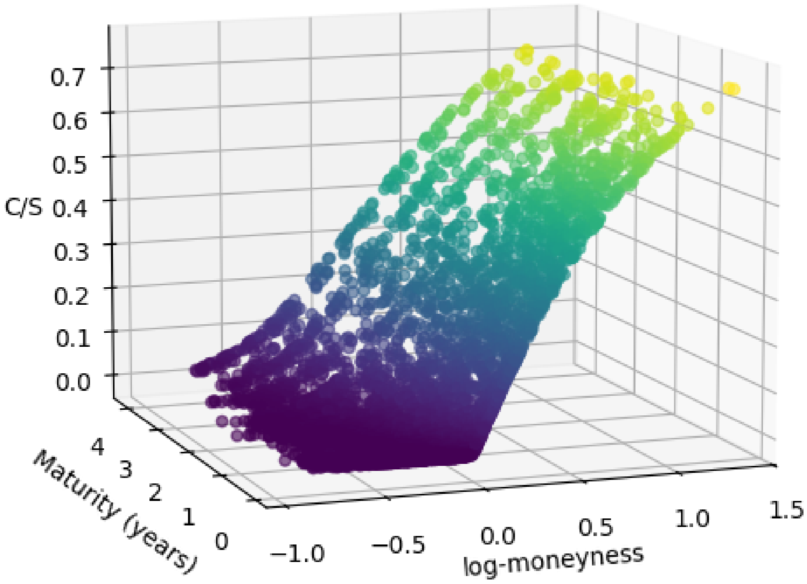
This project investigates the use of machine learning models for pricing European stock options, based on a large dataset of liquid Apple options between 2021 and 2023.

After a phase of cleaning the real market data, a linear model is used as a reference, then non-linear models, Random Forest and Gradient Boosting, are implemented and compared using an identical experimental protocol.

The objective of this study is to compare different pricing models, adopting a progressive approach, moving from simple models to more flexible approaches. Here, we will focus primarily on interpreting performance, strengths and weaknesses, rather than on predictive optimisation.

The results will highlight the practical limitations of these machine learning models for evaluating options. They will therefore pave the way for more specialised methods to better understand the fine structure of price surfaces.

Keywords: Data science, Python, Machine Learning, Option Pricing, Financial Engineering, Random Forest, Gradient Boosting



Contents

1 Introduction

Modern financial markets rely heavily on derivative instruments, whose value depends on underlying assets such as equities, indices or interest rates. Among these instruments, options play a central role in risk management, speculation and investment. Understanding how an option is valued is therefore a fundamental issue in finance, at the crossroads of economics, probability and applied mathematics.

An option is a financial contract that gives its holder the right, but not the obligation, to buy or sell an underlying asset at a predetermined price, called the strike price, on a given future date, called the maturity date. There are two main types of options: calls, which give the right to buy the asset, and puts, which give the right to sell it. In this project, we focus exclusively on European call options, i.e. those that can only be exercised at maturity.

The payoff of a European call option at maturity T is given by the formula:

$$\text{Payoff}_{\text{call}}(T) = \max(S_T - K, 0), \tag{1}$$

- S_T denotes the price of the underlying asset at maturity T,
- K is the strike price.

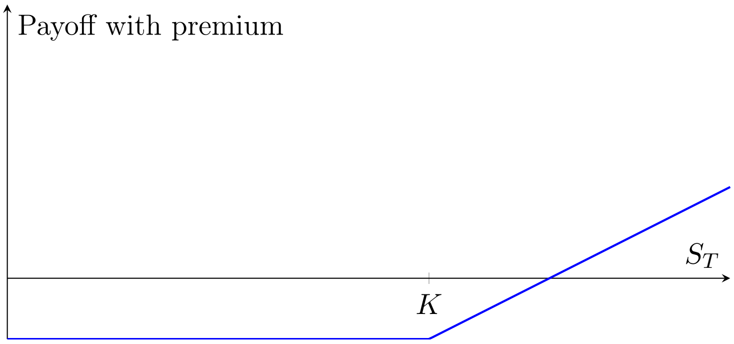


Figure 1: Payoff of a European call option at maturity

This expression immediately illustrates the asymmetric nature of the call: the gain is potentially unlimited when the price of the asset rises, while the maximum loss is limited to the price paid to acquire the option. The difficulty in pricing therefore lies in determining today the 'fair' price of this future right, taking into account the uncertainty surrounding the performance of the underlying asset.

In this project, the underlying asset studied is Apple stock, one of the world's most highly capitalised and liquid companies. This high liquidity translates into a particularly rich options market, with a large number of strikes and maturities available, making it an ideal subject for studying and modelling option prices.

The aim is to explore the extent to which machine learning methods can contribute to better option pricing models, while maintaining a critical and interpretable approach.
We will look at 3 different models of varying complexity to model this problem:

- Linear Regression model.
- Random Forest model.
- Gradient Boosting model.

2 Relevant Literature

2.1 Black-Scholes Model

The Black-Scholes model is the theoretical starting point for the vast majority of modern work on option pricing. Introduced by Fischer Black and Myron Scholes in the 1970s, it provides an analytical framework for determining the price of a European option.

The model assumes that the price of the underlying asset S_t follows a geometric Brownian motion with constant volatility σ and drift r (the risk-free interest rate):

$$dS_t = rS_t dt + \sigma S_t dW_t, \tag{2}$$

- W_t is a standard Brownian motion.

The value of a European call option with maturity and strike is then given by the Black-Scholes closed-form formula:

$$C(S_t, t) = S_t \Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2), \tag{3}$$

2.2 Greeks

The Greeks are sensitivity measures that quantify how the price of an option reacts to small variations in key model parameters. They play a central role in risk management and hedging strategies, as they describe the local behavior of the option price with respect to changes in the underlying asset, volatility, and time.

Delta and Gamma The *Delta* measures the sensitivity of the option price to a change in the underlying asset price, while the *Gamma* captures the curvature of this relationship.

$$\Delta = \frac{\partial C}{\partial S} \qquad \Gamma \ = \frac{\partial^2 C}{\partial S^2} \tag{4}$$

Vega and Theta The *Vega* measures the sensitivity of the option price to changes in volatility, whereas the *Theta* represents the sensitivity of the option value to the passage of time, often referred to as time decay.

$$\text{Vega} = \frac{\partial C}{\partial \sigma} \qquad \Theta \ = \frac{\partial C}{\partial t} \tag{5}$$

2.3 Empirical limits of Black-Scholes

Despite its theoretical elegance and widespread use, the Black-Scholes model has empirical limitations when confronted with actual market data. In practice, the assumptions of constant volatility, log-normal returns and continuous trading are often violated. Market prices typically exhibit volatility curves and asymmetries, which contradict the assumption of constant volatility. As a result, the Black-Scholes model struggles to accurately replicate observed option prices across the entire surface.

These limitations encourage exploration of other approaches capable of capturing the complex, non-linear relationships between option prices. In this context, machine learning models offer a flexible framework for approximating the pricing function directly from data, without imposing restrictive parametric assumptions. By leveraging the wealth of market data that is increasingly available to us, machine learning models are increasingly being used by banks to complement traditional pricing frameworks.

3 Methodology

3.1 Data Description

The dataset used in this project comprises historical data on the Apple stock options market, covering the period from 2021 to 2023. It is publicly available and provides detailed information on European options traded across a wide range of strike prices and maturities. Due to the high liquidity of Apple options, the dataset contains a large number of observations, making it particularly suitable for data-driven modelling approaches.

Size and General Characteristics The dataset contains over 548,000 option contracts, each corresponding to an exercise price and maturity date. The large sample size allows for robust statistical analysis and smooth training of machine learning models. The data covers different market regimes over the period under review, including environments of varying volatility, which is essential for assessing the stability and generalisation capabilities of pricing models.

Features Each observation includes both raw market variables and derived quantities. The main features can be grouped as follows:

- **Contract specifications**: Strike price, expiration date, time to maturity (DTE), and option type (call/put).
- **Market prices**: Last traded option price, bid and ask prices, and trading volume.
- **Underlying asset information**: Spot price of the Apple stock at the observation date.
- **Model-implied quantities**: Implied volatility and option Greeks such as Delta, Gamma, Vega and Theta.

Data Quality and Preprocessing Issues The raw data present several quality issues that require careful pre-processing. These include missing values in certain fields (notably transaction volume and implied volatility).

3.2 Cleaning - NoteBook 1

Prior to model implementation, the raw dataset is restricted to European call options only.

All relevant variables are converted to numerical format. A set of economic consistency filters is then applied to exclude invalid entries, including non-positive prices, strikes, maturities, or underlying values, as well as implausible implied volatility levels Beyond these filters, additional consistency checks are implemented to ensure data integrity and numerical stability. These checks help prevent the propagation of invalid observations into subsequent modeling stages, which is especially important in a machine learning context where such errors can significantly distort training outcomes.

3.3 Construction of financial variables

Following data cleaning, several derived variables are constructed to better capture the economic structure of option prices and to facilitate learning by machine learning models.

$$T_{\text{years}} = \frac{\text{DTE}}{252}, \quad \text{252 is the number of trading days in a year.}$$

$$\text{Moneyness} = \frac{S}{K} \quad \log \text{Moneyness} = \log\left(\frac{S}{K}\right).$$

These constructions are motivated by both financial intuition and modeling considerations. Moneyness provides a normalized measure of how far an option is in-, at-, or out-of-the-money. This representation allows option contracts with different strikes and underlying price levels to be compared on a common scale, which is essential when modeling large cross-sections of options. The logarithmic transformation of moneyness further enhances this representation by centering the at-the-money region around zero and restoring symmetry between in-the-money and out-of-the-money options. In addition, log-moneyness improves numerical stability and helps machine learning models capture non-linear pricing patterns more efficiently, particularly in regions where option prices exhibit strong convexity.

3.4 Definition of the target

Rather than modeling the option price directly, the target variable is defined as the normalized call price, given by the ratio between the option price and the underlying asset price:

$$\text{Target} = \frac{C}{S}.$$

This choice is mainly motivated by statistical and modelling considerations. First, normalising the option price by the price of the underlying asset reduces heteroscedasticity, as price variance generally increases with the level of the underlying asset. This leads to a more homogeneous error structure between observations. Furthermore, from a modelling perspective, this transformation results in more stable coefficients, particularly in linear models. Finally, by limiting the range of the target variable, we facilitate the learning of machine learning algorithms, particularly for linear and tree models, which benefit from improved numerical stability.

3.5 Split training/prediction

Observations from January 2021 to December 2022 are used for training, while data from January 2023 to March 2023 are reserved for testing, corresponding respectively to approximately 87% and 13% of the available observations. This chronological split ensures that the model is always evaluated on future observations relative to the training period, thereby avoiding any form of look-ahead bias. This choice is particularly important in an option pricing context, where market conditions evolve over time. By relying on a temporal split, the evaluation better reflects realistic pricing conditions, in which models are trained on historical data and applied to unseen future market environments. Moreover, this split strategy is applied consistently across all models considered in the project, ensuring fair and comparable performance assessments.

- **Training set**: January 2021 – December 2022 ($\sim 87\%$)
- **Test set**: January 2023 – March 2023 ($\sim 13\%$)

4 Linear Regression - NoteBook 2

4.1 Motivation

The modelling strategy begins with a linear regression model, which provides a clear assessment of the extent to which standard financial variables explain option prices under restrictive assumptions. In particular, it provides a natural benchmark against which more flexible non-linear learning models can be compared at a later stage.

4.2 Implementation

Feature Selection and Information Leakage Option Greeks are deliberately excluded from the linear pricing model in order to prevent information leakage and ensure a fair evaluation framework. This decision is motivated by the following considerations:

- **Circularity risk**: Greeks are typically computed from the option price itself or derived from the same inputs as the pricing model, which would introduce circular dependence in the regression.
- **Artificial explanatory power**: Including Greeks would mechanically increase in-sample performance without providing genuine predictive information.
- **Economic consistency**: The feature set is restricted to variables that are observable prior to pricing and independent from the target variable.

Multicollinearity Assessment Before finalizing the model specification, the presence of multicollinearity among explanatory variables is evaluated using the Variance Inflation Factor (VIF). This diagnostic measures how much the variance of a regression coefficient is inflated due to linear dependence with other regressors. Low VIF values indicate weak correlations among features and ensure that coefficient estimates are stable and interpretable.

Table 1: Interpretation of Variance Inflation Factor (VIF) values

VIF Range	**Interpretation**
VIF $\in [1, 2]$	Low multicollinearity (acceptable)
VIF > 5	Significant multicollinearity
VIF > 10	Severe multicollinearity

Table 2: VIF values for the explanatory variables retained in the linear model

Features	**VIF**
Log-moneyness $\log(S/K)$	1.15
Time to maturity (T)	1.25
Implied volatility (σ_{imp})	1.40

Final Linear Specification Based on these considerations, the linear baseline model is specified as:

$$\frac{C}{S} = \beta_0 + \beta_1 \log\left(\frac{S}{K}\right) + \beta_2 T_{\text{years}} + \beta_3 \sigma_{\text{imp}} + \varepsilon, \tag{6}$$

4.3 Results

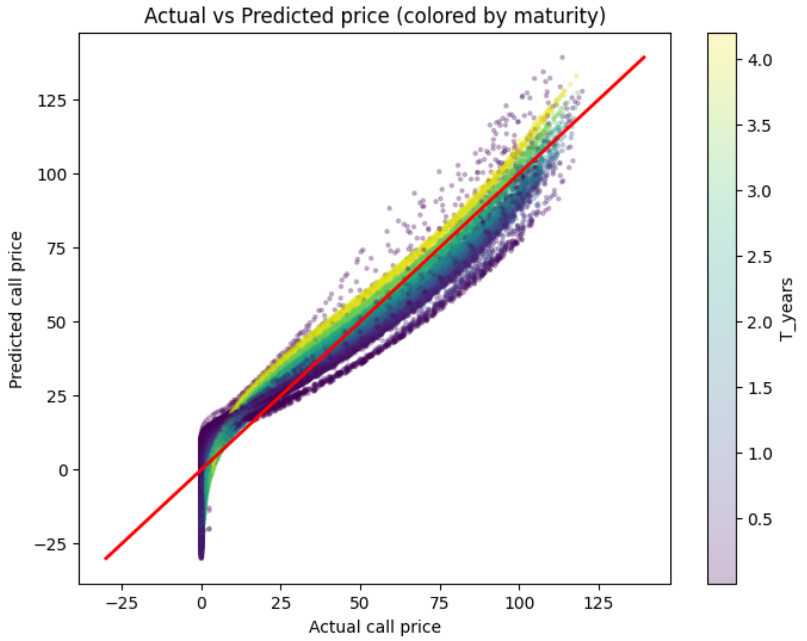
Table 3: Estimated coefficients of the linear regression model

Features	**Estimated Coefficients**
Intercept	0.1810
Log-moneyness $\log(S/K)$	0.1883
Time to maturity (T)	0.0462
Implied volatility (σ_{imp})	0.0230

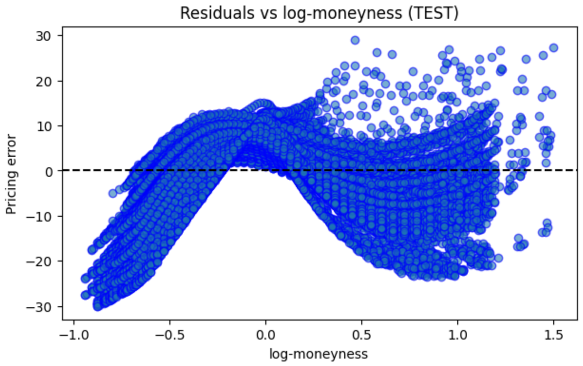
Table 4: Performance metrics of the linear model on training and test sets

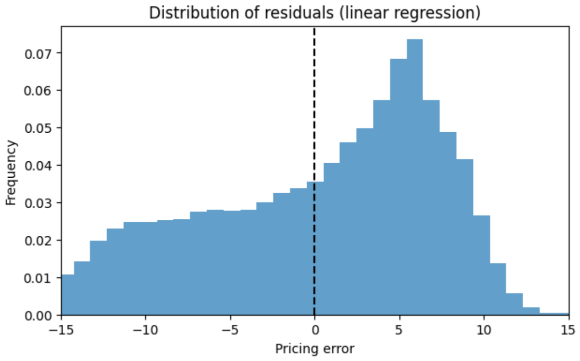
Dataset	**MAE**	**RMSE**	R^2
Training set	0.0423	0.0517	0.9399
Test set	0.0468	0.0584	0.8973

The linear model explains a substantial portion of the variability in the relative call price, with an R^2 close to 0.94 on the training sample and approximately 0.90 on the test sample. The moderate gap between in-sample and out-of-sample performance indicates good generalization ability, with no clear evidence of overfitting. Prediction errors remain well contained, with a root mean squared error below 6% of the spot price, which is notable given the simplicity of the linear specification. The analysis of the estimated coefficients is economically consistent. Log-moneyness emerges as the primary driver of the relative option price, reflecting the strong dependence of call values on the in-the-money and out-of-the-money regimes. Time to maturity and implied volatility exhibit positive but more moderate effects, in line with option pricing theory. However, their contribution remains secondary within a strictly linear framework.



The scatter plot generally follows the diagonal, indicating that the model correctly captures the average price. However, a clear curvature appears for deeply OTM and ITM options, revealing the model's inability to reproduce the convexity of the payoff. Furthermore, the presence of negative predicted prices highlights a structural limitation of the linear model.





The distribution of residuals is relatively broad but remains reasonable overall, with a noticeable shift away from zero indicating the presence of a systematic bias rather than isolated extreme errors. The substantial variance observed in the residuals reflects alternating regions of overpricing and underpricing, suggesting a structural limitation of the model rather than simple random noise. An analysis of residuals as a function of log-moneyness reveals a clear and non-random pattern. Out-of-the-money options exhibit predominantly negative pricing errors, while the at-the-money region is comparatively well explained, where the payoff behaves in an almost linear manner. In contrast, for deeply in-the-money options, the dispersion of residuals increases sharply, reflecting heightened sensitivity to volatility and time to maturity.

This strong dependence of residuals on log-moneyness implies that

$$\mathbb{E}\left[\varepsilon \mid \log\left(\frac{S}{K}\right)\right] \neq 0,$$

which is a clear indication of model bias. The linear specification imposes an affine relationship, whereas the true option pricing function is inherently convex. Consequently, the observed errors do not correspond to random noise but rather reveal the intrinsic inability of the linear model to capture the geometric structure of the payoff, thereby motivating the use of non-linear modeling approaches.

5 Random Forest - NoteBook 3

5.1 Motivation

The linear regression model exhibits structural limitations in capturing the non-linear and convex nature of option prices, as evidenced by systematic patterns in the residuals across moneyness regimes. Random Forest models provide a flexible, non-parametric alternative capable of learning complex, regime-dependent relationships without imposing an explicit functional form. This makes them particularly well suited for addressing the limitations observed in the linear framework.

5.2 Implementation

Table 5: Random Forest configuration

Hyperparameter	**Value**	**Motivation (pricing context)**
`n_estimators`	300	Stabilizes predictions by averaging many trees; reduces variance in a noisy market dataset,compromise robustness/time
`max_depth`	None	High flexibility to capture non-linearities and interactions(ATM convexity).
`min_samples_split`	10	Prevents overly aggressive splits driven by microstructure noise; improves generalization.
`min_samples_leaf`	5	Regularizes the model by enforcing local averaging; avoids memorizing sparse/noisy regions. (ATM very sensible)
`max_features`	`sqrt`	Increases tree diversity and robustness; avoids systematic reliance on moneyness alone.
`oob_score`	True	Provides an internal generalization estimate (out-of-bag) without additional validation split.
`random_state`	42	Ensures reproducibility of results for fair model comparisons.

The out-of-bag (OOB) score close to one confirms the strong generalization ability of the Random Forest on unseen data. The large average number of leaves per tree reflects the high complexity of the learned pricing surface, which is consistent with the strong non-linearities and regime-dependent behaviors observed in option prices.

Table 6: Random Forest diagnostic indicators

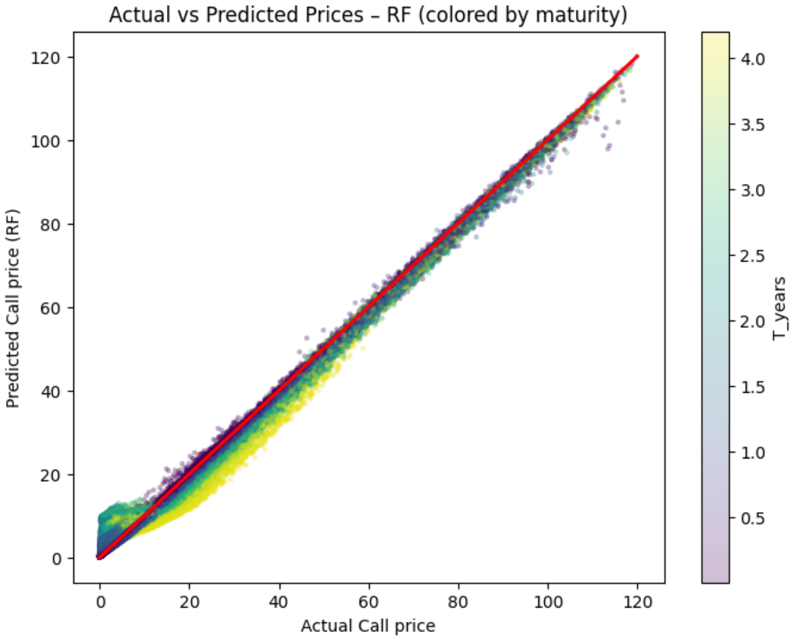
Diagnostic	**Value**	**Interpretation**
Out-of-Bag (OOB) score	0.9995	Indicates good generalization performance, comparable to a built-in cross-validation estimate.
Average number of leaves per tree	41 000	Reflects the flexibility required to approximate a highly non-linear and convex option pricing surface.

5.3 Results - Comparaison with Linear Regression

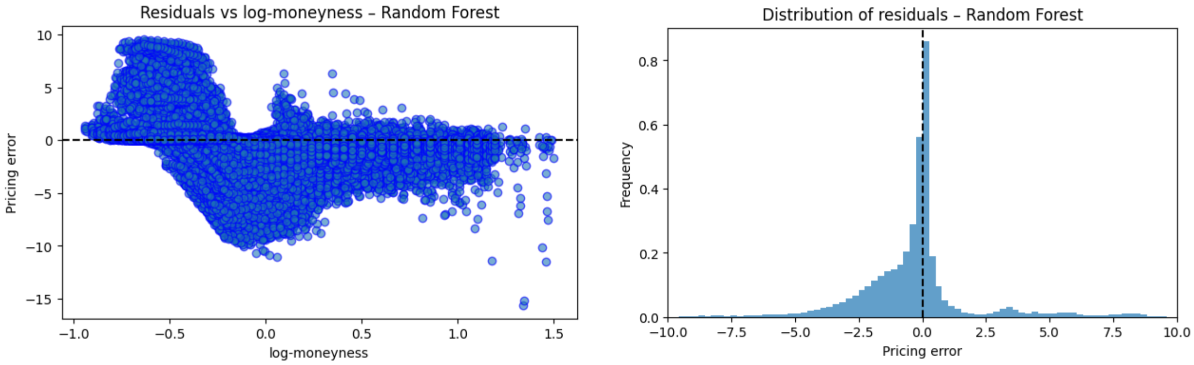
Table 7: Performance metrics of the Random Forest model on training and test sets

Dataset	**MAE**	**RMSE**	R^2
Training set	0.0019	0.0035	0.9997
Test set	0.0091	0.0151	0.9932

Compared to the linear regression model, the random forest shows improvement in terms of both R^2 and RMSE on the training and test sets. The out-of-sample R^2 increases from approximately 0.90 to over 0.99, indicating a stronger ability to explain the variability in option prices. At the same time, the test RMSE decreases sharply from approximately 5.8% to approximately 1.5%, reflecting a significant reduction in pricing errors. This improvement highlights the random forest's superior ability to capture option prices.



The random forest allows for much better overall calibration of option prices, as illustrated by the strong alignment along the identity line. There is virtually no curvature, and the overall bias has disappeared. The colours are mixed for the same level, which shows that the model has been able to learn the interactions (maturity no longer induces bias as in the linear model). However, there is a small cloud for low C/S values (ATM zone, high convexity, difficult structure).



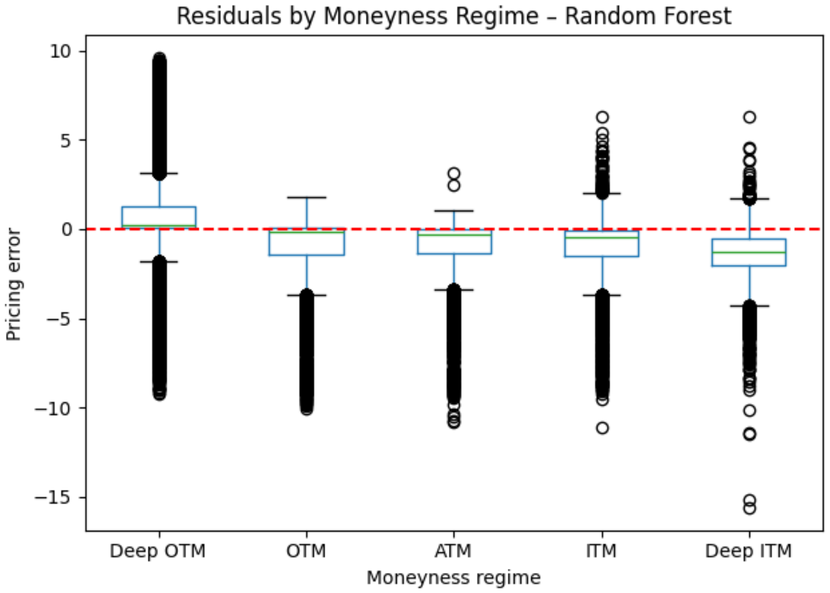
With the random forest model, the smiley face structure is much more attenuated. The residuals-log-moneyness relationship is no longer symmetrical. This reflects the fact that the bias is no longer global but regime-dependent; we manage to capture the overall convexity, and the errors are closer to zero. However, we observe local biases, particularly around the ATM zone, where options tend to be undervalued.

As for the distribution of residuals, it is much better, very centered around zero, and the few false predictions have much smaller absolute errors. We note only a left tail corresponding to the underpricing zone. These two figures convey the same conclusion from complementary perspectives:

- the Random Forest corrects the average pricing error,
- it captures the global non-linearity of the option pricing surface,

But local biases remain in specific regimes,

$$\mathbb{E}\left[\varepsilon \,\middle|\, \log\left(\frac{S}{K}\right)\right] \neq 0.$$



The residuals by regime confirm this.

6 Gradient Boosting - NoteBook 4

6.1 Motivation

The results obtained with Random Forest models motivate the introduction of Gradient Boosting to improve local price modeling, particularly in the ATM zone. Unlike Random Forest, which relies on independent aggregation of trees and prioritizes overall variance reduction, Gradient Boosting proceeds by successively correcting residual errors. This sequential learning approach is particularly relevant when errors are not random but localized in specific regions of the variable space, as is the case around the ATM zone observed with Random Forest.

The objective is therefore not to question the nonlinear capabilities of Random Forest, but to test whether a model capable of focusing its learning efforts on poorly explained areas can improve pricing stability and limit the extreme errors highlighted in previous analyses.

6.2 Implementation

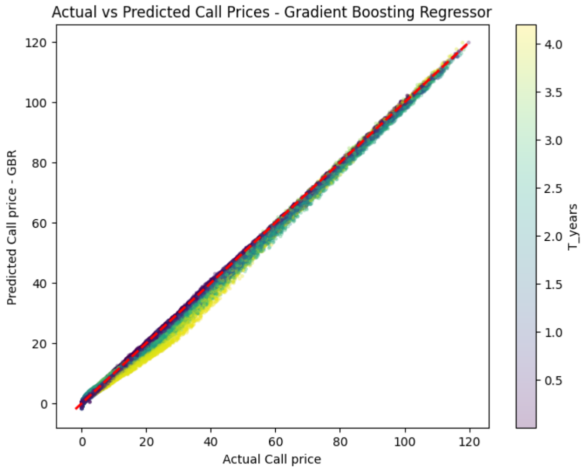
Table 8: Gradient Boosting configuration

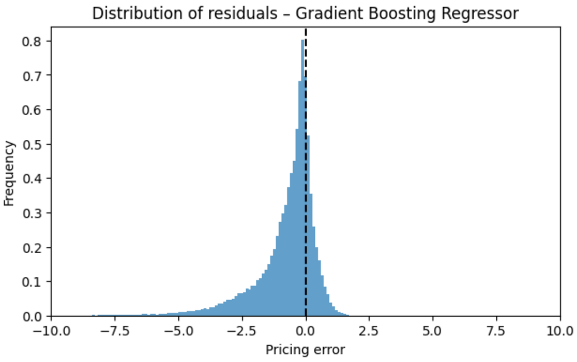
Hyperparameter	**Value**	**Motivation**
`loss`	`Squared error`	The quadratic loss function heavily penalizes large-amplitude errors, which is consistent with pricing where extreme errors, especially around the at-the-money zone, are economically costly.
`Learning rate`	0.05	Progressive correction of residual errors, promoting finer adjustment in highly convex areas without causing overfitting.
`n estimators`	600	A large number of trees compensates for the low learning rate and allows the model to capture complex local structures.
`max depth`	3	A limited depth forces each tree to model simple interactions while allowing the capture of second-order nonlinearities typical of option pricing.
`min sample leaf`	50	Minimum number of observations per sheet, stabilizing predictions in very noisy regions.
`subsample`	0.8	Additional regularization, reducing the variance of the model while retaining its ability to learn dominant pricing patterns.
`random state`	42	Ensures reproducibility of results for fair model comparisons.

6.3 Results - Comparaison with Random Forest

Table 9: Performance metrics of the Random Forest model on training and test sets

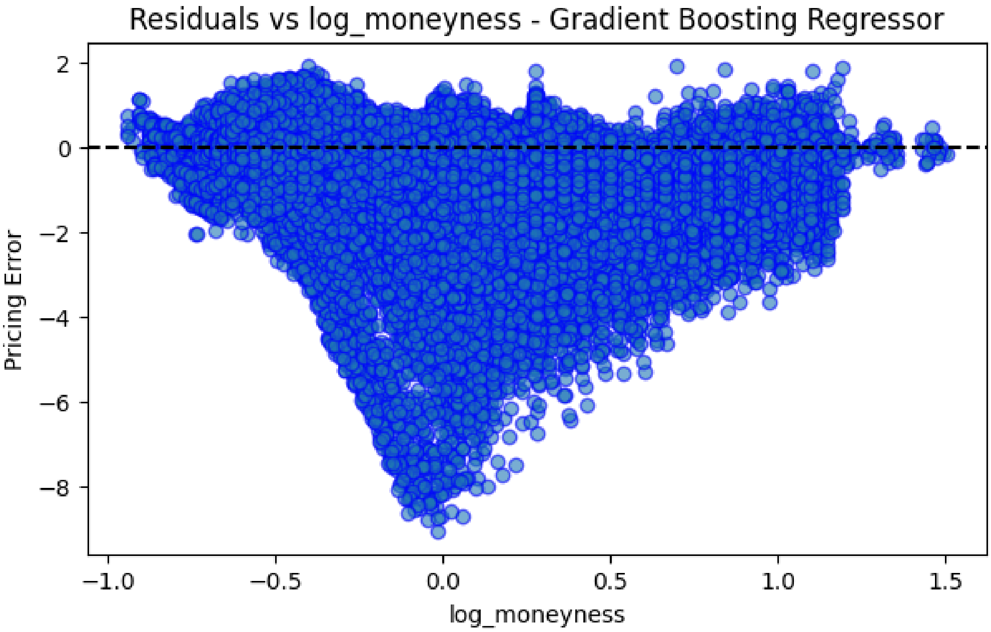
MAE	**RMSE**	R^2
0.0059	0.0094	0.9973





The Actual vs Predicted graphs and the error distribution obtained with Gradient Boosting are very close to those observed with Random Forest and do not reveal any major change in interpretation. Overall performance and the concentration of errors around zero remain comparable.

However, the analysis of residuals as a function of log-moneyness is different; here is the graph obtained:



The residual analysis based on log-moneyness shows that, like Random Forest, Gradient Boosting retains a persistent bias in the most convex region of the price surface. In both nonlinear models, we underprice our ATM options. This tells us that the difficulty is not "the model is too simple" but "the area is inherently unstable and very sensitive."

However, the error dispersion appears more controlled than with Random Forest, with fewer outliers and a smoother structure, suggesting a stability gain related to sequential learning.

Consequently, Gradient Boosting primarily improves variance (stability) rather than correcting the structural bias observed in the ATM area.

7 Conclusion, Perspectives

Conclusion: The linear regression model provided a clear and interpretable baseline, capturing a significant portion of the variability in option prices, but it exhibited strong limitations. In particular, residual analysis revealed systematic patterns across parity regimes, reflecting the inability of a linear specification to reproduce the intrinsic convexity of option prices.

The introduction of nonlinear models significantly improved overall pricing performance. The Random Forest model led to a substantial reduction in pricing errors and successfully captured the overall nonlinear structure of the option price surface. However, diagnostic analyses revealed persistent local biases, particularly in the ATM zone, where pricing errors remained consistently negative.

Gradient boosting was then introduced to assess whether a sequential learning approach could correct these localized residual errors. While gradient boosting further improved the stability of predictions by reducing variance and limiting extreme errors, it failed to eliminate ATM underpricing. This result suggests that the remaining pricing errors are not primarily due to insufficient model flexibility, but rather to intrinsic characteristics of the pricing problem itself, such as strong local convexity, heteroscedasticity, noisy implied volatility data, and the absence of explicit economic constraints

In summary, this study highlights that while machine learning models can greatly improve the empirical approximation of option prices, they do not automatically solve all structural pricing problems. In particular, purely data-driven approaches optimized by global loss functions may struggle to accurately capture economically critical but highly unstable regions of the pricing surface.

Perspectives:

1. Modelling implied volatility instead of prices In this project, the learning task focuses on the normalized option price C/S, which exposes the models to strong convexity and heteroscedasticity effects, particularly in the at-the-money (ATM) region. A natural extension would consist in shifting the modelling target from prices to implied volatility.

$$\sigma_{\text{imp}} = f(\log(S/K), T)$$

Once the implied volatility surface is learned, option prices could be recovered using a parametric pricing formula such as Black–Scholes. This approach presents several advantages:

- implied volatility surfaces are empirically smoother than price surfaces,
- convexity effects are handled analytically by the pricing formula,

2. Hybrid pricing: learning residuals relative to Black–Scholes The results indicate that machine learning models capture the global structure of option prices but struggle in highly sensitive regions. A promising direction is therefore to combine financial theory and machine learning through a residual learning approach.

$$C_{\text{market}} = C_{\text{BS}} + \varepsilon_{\text{ML}}$$

where C_{BS} denotes the Black–Scholes price and ε_{ML} is a data-driven correction learned by the model.

3. Economically weighted loss functions The persistent underpricing observed in the ATM region is partly driven by the use of a uniform loss function across all observations. From an economic perspective, pricing errors are not equally costly across the option surface.

A potential improvement would consist in introducing a weighted loss function of the form:

$$\mathcal{L} = \sum_i w_i \left(\hat{C}_i - C_i \right)^2 ,$$

where the weights w_i reflect the economic sensitivity of each option, for instance:

- higher weights for ATM options,
- weights proportional to the option's gamma or vega,
- regime-dependent weighting schemes.

Such an approach would align the statistical objective of the model with financial risk considerations, allowing the learning process to focus explicitly on the most critical regions of the pricing surface.

References

1. John Hull, Options, Futures, and Other Derivatives, 11th Edition, Pearson, 2021.

2. Victor Panaretos, Linear Models course notes, EPFL, 2025.

3. Perazzi Elena, Probability and stochastic calculus course notes, EPFL, 2025.

4. Dataset Source. AAPL Option Chains - Q1 2016 to Q1 2023. Available at: `https://www.kaggle.com/datasets/kylegraupe/aapl-options-data-2016-2020?select=aapl_2021_2023.csv`

8 Code Repository

GitHub Repository: `https://github.com/chap-pi/Option-Pricing-ML`