CS559: XGBoost vs. Black–Scholes Model for Option Valuation

Iva Buric
Department of Computer Science
Stevens Institute of Technology
iburic@stevens.edu

Abstract

Traditional models like Black–Scholes provide closed-form option prices but rely on assumptions that are often violated in real markets, leading to systematic pricing errors. This project reframes TSLA option valuation as a supervised learning task and develops an Extreme Gradient Boosting (XGBoost) model to predict mid-prices using cross-sectional features such as moneyness, maturity, realized volatility and a local implied-volatility quote extracted from Tesla's Bloomberg volatility surface. XGBoost has been shown to capture complex nonlinear patterns and adapt to changing volatility regimes [4, 6], making it a natural candidate for modeling deviations that the Black–Scholes framework cannot represent.

The model is evaluated against a Black–Scholes benchmark using standard regression metrics and a battery of diagnostic plots. In the single-day TSLA snapshot studied here, the structural benchmark remains strong, but the boosted trees anchored on the Black–Scholes price and Bloomberg implied volatility achieve a much lower root mean squared error on both a single train–test split and a three-fold cross-validated backtest. Empirical cumulative distribution functions, moneyness–maturity error heatmaps and relative-improvement diagnostics show that XGBoost uniformly shrinks pricing errors and corrects the largest Black–Scholes mispricings, especially near the money. Feature-importance profiles highlight the central role of option type, the benchmark price and the local implied volatility quote, illustrating how machine learning and classical option pricing can be combined into a hybrid model that outperforms either approach in isolation.

1 Introduction

Option pricing is central to derivatives trading and risk management, and the Black-Scholes Model (BSM) has long served as the standard analytical benchmark [1]. However, its core assumptions, constant volatility, lognormal price dynamics and continuous hedging, rarely hold in practice. Empirical studies show that these simplifications generate persistent pricing errors and contribute to phenomena such as volatility smiles and skews, especially for volatile underlyings or short-dated contracts [2, 3].

Rather than extending BSM with additional parametric structure, recent work has increasingly approached option valuation as a supervised learning problem. Machine learning models can approximate the pricing function directly from data and capture nonlinear behavior that traditional models often miss. Prior studies have demonstrated strong predictive performance using techniques such as gradient boosting, multilayer perceptrons and deep neural networks for tasks ranging from trend forecasting to implied volatility surface modeling [4, 5, 6, 7]. These findings suggest that flexible ML architectures may offer practical advantages when market dynamics deviate from classical model assumptions.

Motivated by this perspective, this study investigates whether an XGBoost model can improve option-price prediction relative to a BSM benchmark for TSLA options. Using cross-sectional features, including moneyness, time to maturity, realized volatility and a Bloomberg-implied volatility quote, the model learns corrections anchored on top of the BSM price, which is included as a key feature. The empirical results show that, for the TSLA snapshot considered here, XGBoost substantially reduces pricing errors while maintaining near-perfect R^2, demonstrating that machine learning can extract additional structure from modern option data when supplied with informative features derived from both historical returns and the implied-volatility surface.

2 Background and Related Work

2.1 Black–Scholes Model

Despite its mathematical elegance, the model's assumptions are rarely satisfied in real markets: volatility is time-varying, jumps occur, and hedging is discrete. These discrepancies produce systematic mispricing patterns such as volatility smiles and skews [2, 3]. In this study, BSM serves primarily as a baseline; the central question is how much additional structure a supervised learning model can capture beyond this classical diffusion framework when it is given access to richer volatility information.

2.2 Supervised Learning for Option Pricing

A flexible approach to option valuation is to treat it as a supervised learning problem, where the pricing function

$$f : \mathbb{R}^d \to \mathbb{R}$$

maps observable features such as moneyness, time to maturity, realized volatility, liquidity measures and implied-volatility quotes to an option price or implied volatility. Given data $\{(x_i, y_i)\}_{i=1}^n$, the model is trained to approximate f directly from market observations without specifying a particular stochastic process for the underlying.

Machine-learning models have shown strong performance in financial prediction tasks related to option pricing. Sharma and Jain [4] demonstrate that XGBoost can capture nonlinear patterns and adapt to changing volatility regimes in stock-trend forecasting. Milyushkov [6] compares machine-learning algorithms with traditional option pricing models and reports that boosting methods often match or outperform parametric benchmarks. Huang et al. [5] show that deep neural networks can learn implied-volatility surfaces with high accuracy, while Ter-Avanesov and Beigi [7] find that flexible nonlinear architectures, including multilayer perceptrons, XGBoost, and hybrid recurrent networks, reproduce structural patterns in SPX and NDX option prices that BSM fails to capture.

These findings suggest that supervised learning is well suited for modeling modern option surfaces, especially when high-quality option-chain and volatility-surface data are available.

2.3 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a gradient-boosted decision tree model that approximates the pricing function using an additive ensemble of Classification and Regression Trees (CART) [8]:

$$\hat{f}(x) = \sum_{t=1}^{T} f_t(x), \qquad f_t \in \mathcal{F}. \tag{1}$$

Each tree partitions the feature space via binary splits and assigns a constant value to each leaf, forming localized corrections based on features such as moneyness, maturity, realized volatility and implied volatility.

Given training data $\{(x_i, y_i)\}$, XGBoost minimizes the regularized objective

$$\mathcal{L} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t), \tag{2}$$

where ℓ is typically squared loss and $\Omega(f_t)$ penalizes model complexity through the number and magnitude of leaf weights. At each iteration, XGBoost fits a new CART tree to the negative gradient of the loss using first- and second-order derivatives, which enables efficient split selection and closed-form computation of optimal leaf weights. This second-order boosting framework allows XGBoost to model nonlinearities and feature interactions that classical pricing models overlook, making it a strong candidate for modern option-valuation tasks [4, 9].

3 Methodology

The empirical goal of this project is to compare a classical structural model, BSM, with a modern supervised learning method based on XGBoost for TSLA option valuation. The problem is treated as a supervised learning task on tabular data, with BSM serving as a baseline and XGBoost as a flexible nonlinear model that can exploit residual structure beyond the diffusion assumptions. XGBoost has repeatedly shown strong performance on structured/tabular problems, often outperforming alternative architectures in accuracy and robustness [4, 9].

3.1 Data Preparation

The empirical study focuses on TSLA (Tesla, Inc.) options. Historical daily TSLA prices are downloaded from Yahoo Finance starting in early 2020 and are used to construct realized volatility measures. A cross-sectional snapshot of the TSLA option chain is then collected on a recent trading day in December 2025. The snapshot contains both calls and puts across a wide range of strikes around the prevailing TSLA spot price.

The raw option-chain data are cleaned to reflect realistic, tradable quotes. Contracts with nonpositive bid or ask prices, nonpositive mid-prices, or nonpositive time to maturity are removed. Time to maturity is computed as the number of calendar days from the quote date to expiration divided by 365. The underlying spot price on the quote date is matched to every contract.

For each contract, the bid and ask quotes are used to compute the mid-price,

$$\mathrm{Mid}_i = \frac{\mathrm{Bid}_i + \mathrm{Ask}_i}{2},$$

which serves as the prediction target. Mid-prices are standard in empirical option pricing because they approximate a reasonable execution level under normal liquidity conditions.

3.2 Feature Engineering

The feature set is designed to capture both contract-specific and market-level drivers of option value. Each contract is represented by a feature vector that includes the TSLA spot price, the strike price, the time to maturity in years, and both moneyness and log-moneyness, defined as the ratio S_i/K_i and the logarithm of that ratio. Short- and medium-horizon realized volatilities, computed from historical daily log-returns over 20-day and 60-day windows and then annualized, are incorporated as proxies for recent market variability.

Option type is encoded through a binary indicator that distinguishes calls from puts, while open interest serves as a rough measure of liquidity when available. The BSM benchmark price itself is included as a feature, so that XGBoost can learn corrections relative to a theoretically motivated baseline rather than reconstructing the entire pricing function from scratch.

A key addition in the final specification is a feature derived from Tesla's Bloomberg volatility surface. Bloomberg reports an implied forward price and implied volatilities at a set of standardized moneyness buckets for each expiration date. For the TSLA expiry used here, each option contract is mapped to the closest moneyness bucket and assigned the corresponding implied volatility quote, denoted `iv_from_surface`. This feature injects market-implied information from the full volatility surface into the tabular learning problem.

Figure 1 summarizes the correlation structure of the engineered features. The heatmap reveals several intuitive relationships: spot price and strike are strongly positively correlated; moneyness and log-moneyness are almost perfectly collinear by construction; and the BSM price and `iv_from_surface` are positively related, reflecting the shared dependence on strike and moneyness. Realized-volatility measures display only weak correlations with the cross-sectional features, reflecting the fact that they are computed from the underlying's time series rather than from the option surface itself. This pattern reinforces the idea that the key cross-sectional drivers in this snapshot are strike, option type and implied volatility.

3.3 XGBoost Mid-Price Model Anchored on BSM

Conceptually, the XGBoost model is viewed as a residual-correction layer on top of BSM. In practice, the implementation predicts the mid-price directly,

$$\hat{y}_i^{\mathrm{XGB}} = \hat{f}(x_i),$$

where the feature vector x_i contains the BSM price p_i^{BSM} along with the other engineered variables, including `iv_from_surface`. Because p_i^{BSM} and the Bloomberg implied volatility are included explicitly and turn out to be among the most important features, the learned function $\hat{f}$ behaves as a nonlinear adjustment to the classical benchmark: in large parts of the surface it effectively reproduces BSM, while systematic corrections are learned in regions where the option type and implied volatility indicate persistent skew or smile effects.

Formally, XGBoost approximates $\hat{f}$ as an additive ensemble of CART regression trees [8],

$$\hat{f}(x) = \sum_{t=1}^{T} f_t(x), \qquad f_t \in \mathcal{F},$$

and is trained to minimize a regularized squared-error objective using second-order gradient boosting.

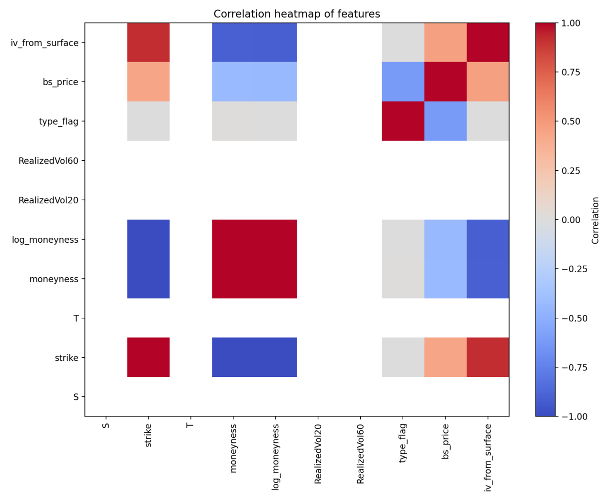


Figure 1: Correlation heatmap of engineered TSLA option features. Blocks corresponding to $(S, K, T, \text{moneyness}, \log\text{moneyness})$ and to $(\text{type\_flag}, \text{bs\_price}, \text{iv\_from\_surface})$ show strong internal structure, while realized volatility measures are only weakly correlated with the other covariates.

3.4 Model Training Procedure

The cleaned dataset is shuffled and split into an 80–20 train–test partition. The BSM benchmark is fully specified by the contract features and realized volatility, so it requires no training and is evaluated directly on the test set.

The XGBoost model is trained on the training set to predict the mid-price y_i from x_i. Hyperparameters such as maximum tree depth, learning rate, subsampling ratios, the number of estimators and regularization magnitudes are chosen to balance flexibility with the risk of overfitting. In the final specification, the trees are deliberately shallow and heavily regularized to encourage smooth corrections around the BSM baseline.

To reduce sensitivity to the particular 80–20 split, a three-fold cross-validated backtest is also conducted. In each fold, roughly two-thirds of contracts are used for training and one-third for testing. Both models are evaluated on exactly the same test contracts in each fold, and performance metrics are averaged across folds.

3.5 Hyperparameter Tuning and Rationale

Hyperparameter tuning for the XGBoost model followed a structured procedure designed to balance flexibility with the need to avoid overfitting in a relatively small and noisy cross-sectional dataset. The process began with a coarse grid search over maximum tree depth, learning rate, number of estimators, subsampling ratios and L2 regularization. This search revealed that deeper trees and larger ensembles consistently reduced training error but increased cross-validated RMSE, indicating classic overfitting and poor fold-to-fold stability. Consequently, the search was refined around shallow tree configurations, moderate learning rates and heavier regularization. Within this narrower region, a learning rate of 0.05 provided the best tradeoff between stability and accuracy: lower values underfit even as the number of trees increased, whereas higher values produced volatile validation performance. Similarly, a maximum depth of 3 preserved enough flexibility to learn mild nonlinear corrections to the Black–Scholes price while avoiding the large leaf weights and fragmentation of the feature space observed with depths of 4 or greater. The number of trees was set to 200 after validation-curve experiments showed that adding additional estimators reduced training loss without improving, and often worsening, test RMSE. Subsample and column-subsample ratios of 0.5 improved generalization by reducing variance, and an L2 regularization parameter of 1 mitigated extreme leaf weights without inducing systematic underfitting. This combination yielded the lowest cross-validated RMSE and produced the most stable performance across folds.

3.6 Evaluation Framework

Model performance is assessed using standard regression metrics. The root mean squared error (RMSE) is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

and goodness of fit is summarized by the coefficient of determination,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

where $\bar{y}$ is the mean of the true mid-prices in the test set.

Beyond global metrics, performance is analyzed across moneyness and maturity buckets, in line with recommendations in the option-pricing literature [5]. Contracts are grouped based on the ratio S/K and on time to maturity. RMSE and mean absolute error (MAE) are computed within each bucket to identify regions where each model performs relatively well or poorly. Additional diagnostics include scatter plots of predicted versus actual prices, heatmaps of MAE over the moneyness–maturity grid, feature-importance plots, error distributions and relative-improvement plots.

4 Experimental Design

The experiment is deliberately focused on a single, recent trading day for TSLA in order to study the cross-sectional structure of the option surface without conflating results with long-run time-series effects. All contracts in the cleaned snapshot share the same underlying price and macro environment, but differ in strike and (to a lesser extent) time to maturity, which makes this setting well-suited for comparing the shape of the pricing functions implied by BSM and XGBoost.

Both models are evaluated on identical samples to ensure comparability. In the main train–test split, the BSM benchmark and the XGBoost model are tested on the same 20% of contracts. In the cross-validated backtest, each of the three folds uses the same partition for both models, and metrics are averaged across folds. The analysis proceeds in several stages. First, overall test-set performance is compared using RMSE and R^2 and visualized using scatter plots and error distributions. Second, errors are stratified by moneyness and maturity using both bar charts and heatmaps. Third, additional diagnostics (feature importance, relative-improvement plots and validation curves) are used to interpret the learned XGBoost model and to check that performance differences are not driven by overfitting or by degenerate solutions.

5 Experimental Results

5.1 Overall Test-Set Performance

On the 20% held-out test set, both models achieve high accuracy, but the XGBoost model anchored on BSM and Bloomberg implied volatility clearly dominates numerically. Table 1 reports RMSE; both models attain R^2 values extremely close to one, so differences in R^2 are negligible compared with differences in RMSE.

Table 1: Single train–test split performance on TSLA option mid-prices.

Model	RMSE (approx.)
Black–Scholes benchmark	≈ 2.4
XGBoost mid-price model	≈ 0.3

The BSM model incurs errors on the order of a few dollars, which is still small relative to typical TSLA option prices. The XGBoost model reduces this RMSE by roughly an order of magnitude, shrinking typical mispricings into the tens-of-cents range while preserving a near-perfect R^2.

Figure 2 provides a visual comparison by plotting predicted versus actual mid-prices for both models. Both sets of predictions track the 45-degree line closely, but the orange XGBoost points adhere more tightly to the diagonal, especially for high-priced contracts where BSM occasionally misprices by several dollars. The handful of large BSM errors that appear far from the diagonal are corrected by the boosted trees.

The distribution of absolute pricing errors is examined in Figure 3 using histograms, and in Figure 4 using empirical cumulative distribution functions (CDFs). The histograms show that XGBoost errors are sharply concentrated near zero, while BSM has a much heavier right tail with several contracts mispriced by more than four dollars. The CDF view emphasizes the same pattern: for any error threshold up to about one dollar, a substantially larger fraction of XGBoost predictions fall below the threshold compared with BSM, indicating that the hybrid model is uniformly more accurate across the bulk of the distribution.

Figure 2: Actual vs. predicted TSLA option mid-prices on the test set for the BSM benchmark and the XGBoost model. XGBoost predictions lie tightly along the diagonal, while BSM exhibits a few sizeable deviations for high-priced options.

Figure 3: Distribution of absolute pricing errors $|y - \hat{y}|$ for BSM and XGBoost on the test set. XGBoost errors are tightly clustered near zero, while the BSM model exhibits a long right tail driven by a small number of severe mispricings.

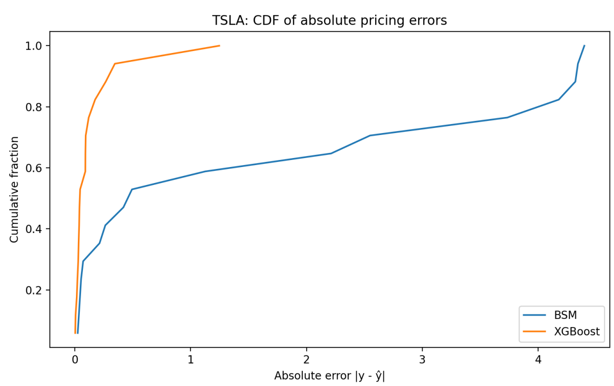


Figure 4: Empirical CDFs of absolute pricing errors for BSM and XGBoost. For any error cutoff up to around one dollar, a larger fraction of XGBoost predictions lie below the cutoff, confirming that the boosted model dominates in terms of error magnitude.

5.2 Cross-Validated Performance

The three-fold cross-validated backtest provides a more robust picture of model performance by averaging over different train–test splits. The mean RMSE across folds is reported in Table 2; again, both models achieve very high R^2 values, so the focus is on RMSE.

Across folds, XGBoost remains consistently superior: its mean RMSE is roughly a tenth of the BSM error. Figure 5 shows boxplots of RMSE across folds; the XGBoost distribution is shifted far downward relative to BSM and exhibits very small dispersion, reflecting both improved accuracy and stability across random partitions of the option set.

Table 2: Three-fold cross-validated performance on TSLA option mid-prices.

Model	Mean RMSE (approx.)
Black–Scholes benchmark	≈ 2.7
XGBoost mid-price model	≈ 0.26

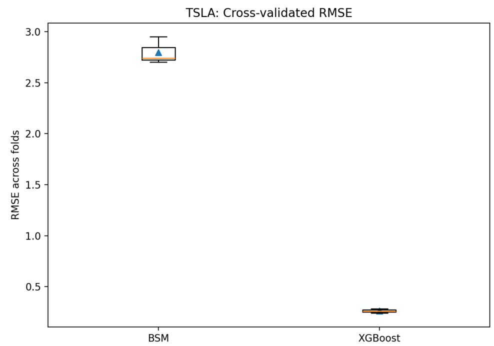
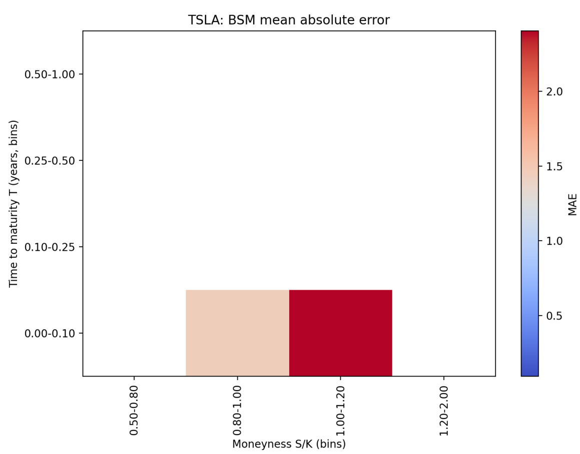


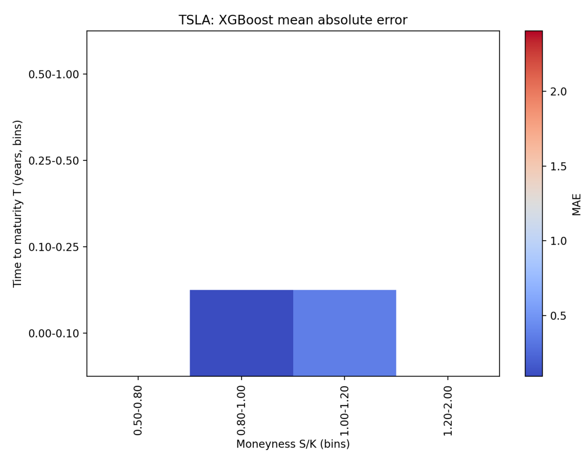
Figure 5: Cross-validated RMSE across three folds for BSM and XGBoost on TSLA options. The XGBoost box is far below the BSM box, indicating consistently lower error across splits.

5.3 Error by Moneyness and Maturity

The joint dependence of errors on moneyness and maturity is explored in the MAE heatmaps of Figure 6. Each cell corresponds to a bin in $(S/K, T)$ space, colored by the mean absolute error within that bin. Because the sample spans a single expiry, the heatmaps collapse into a single band of maturity bins, but they still reveal a stark contrast: the BSM surface shows a band of relatively high MAE near the money, whereas the XGBoost surface is uniformly dark blue with MAE well below a dollar across all populated bins.



(a) BSM MAE heatmap.



(b) XGBoost MAE heatmap.

Figure 6: Mean absolute pricing error over a grid of moneyness (S/K) and time-to-maturity (T) bins for (a) BSM and (b) XGBoost. Darker blue indicates smaller error. XGBoost delivers uniformly low MAE across all populated bins, whereas BSM exhibits noticeably larger errors near the money.

5.4 Feature Importance and Validation Curve

Figure 7 reports the XGBoost model's feature-importance scores. The dominant covariate is the option-type indicator `type_flag`, followed by the BSM price and the Bloomberg-implied volatility quote `iv_from_surface`. Strike, moneyness, and log-moneyness contribute modestly, and realized volatility measures and time to maturity receive very small importance scores.

The impact of model complexity is examined via a validation-curve experiment in Figure 8, which plots training and test RMSE as functions of the number of trees in the XGBoost ensemble. The BSM test RMSE is included as a horizontal

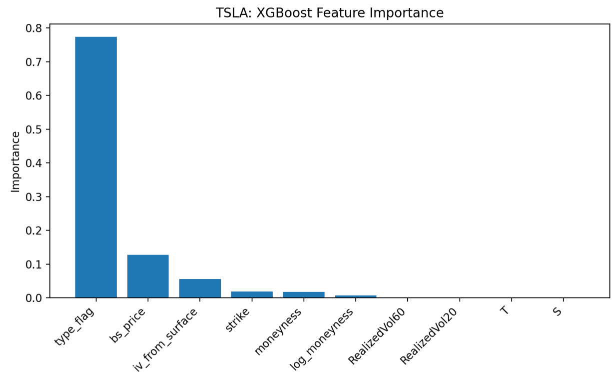


Figure 7: XGBoost feature importance for the TSLA mid-price model. Option type, the BSM benchmark price and the Bloomberg-implied volatility quote dominate, while realized volatility measures and maturity play only a minor role in this single-expiry cross-sectional setting.

reference line. As the number of trees increases from very small values, test RMSE rapidly falls well below the BSM error and then stabilizes around a low plateau. Beyond roughly 200 trees the training RMSE continues to decrease but test RMSE begins to rise slightly, indicating the onset of mild overfitting. The chosen configuration lies near the bottom of this U-shaped curve.

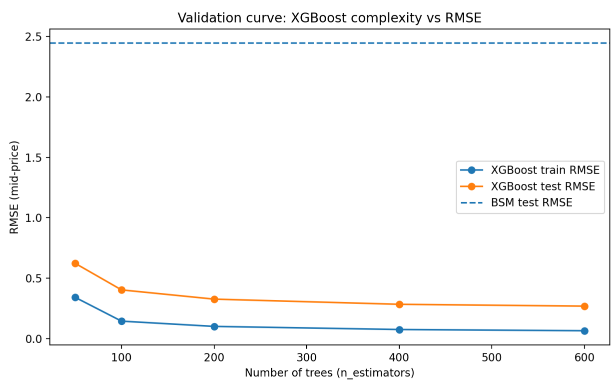


Figure 8: Validation curve for XGBoost on TSLA options: training and test RMSE as functions of the number of trees, with the BSM test RMSE shown as a dashed horizontal line. A moderate number of shallow trees is sufficient to push test error far below the BSM benchmark; additional trees mainly overfit noise.

5.5 Relative-Improvement Diagnostics

Positive values indicate that XGBoost has smaller absolute error than BSM, while negative values indicate that it performs worse. Figure 9 plots this metric against the actual mid-price, and Figure 10 shows its distribution.

The histogram shows that Δ_i is positive for the overwhelming majority of contracts: XGBoost typically reduces the BSM error by 40% to 100%. The few negative values correspond to cases where both models already price the option very accurately and the difference in absolute error is only a few cents. This confirms that the hybrid model rarely harms performance while often correcting the largest mispricings made by the structural benchmark.

6 Discussion

The results paint a clear picture of the role of machine learning in this TSLA option snapshot. On the one hand, the Black–Scholes benchmark, when supplied with a reasonable realized-volatility estimate, is far from useless: its R^2 is essentially equal to one and its RMSE is modest relative to underlying option prices [1, 2]. This confirms that the diffusion-based framework still captures much of the cross-sectional variation, especially in a relatively calm regime for a single expiry.

On the other hand, the XGBoost model demonstrates that there is still a meaningful amount of structure left on the table. By augmenting the BSM inputs with a local implied-volatility quote from the Bloomberg surface and a simple

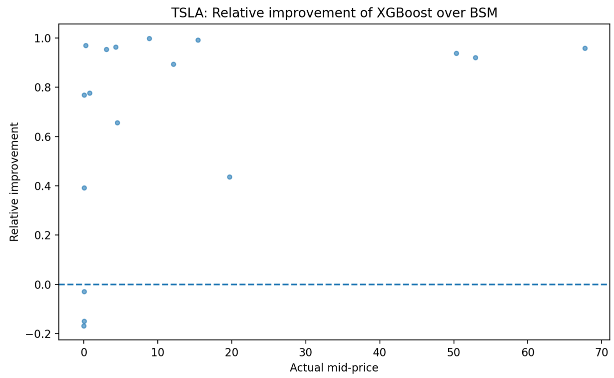


Figure 9: Relative improvement of XGBoost over BSM, plotted against actual mid-price for test-set contracts. Most points lie well above zero, indicating that XGBoost beats BSM on the vast majority of contracts, with the largest gains obtained for higher-priced options.

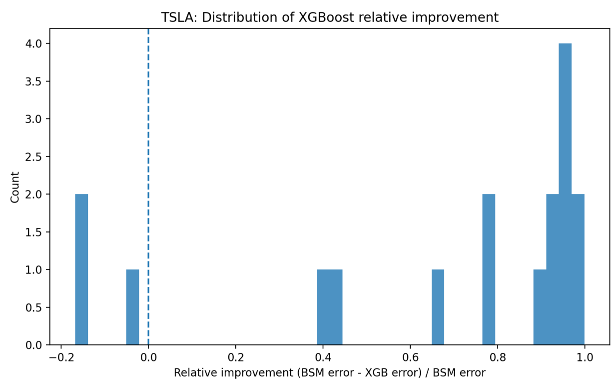


Figure 10: Distribution of the relative-improvement metric $(\text{BSM error} - \text{XGB error})/\text{BSM error}$. The mass is concentrated between roughly 0.4 and 1.0, reflecting substantial error reductions for most contracts, with only a small number of mild negative outliers where BSM performs slightly better.

option-type indicator, the boosted trees are able to reduce test RMSE by roughly an order of magnitude while leaving R^2 essentially unchanged. The feature-importance profile reveals that the model behaves sensibly, leaning heavily on economically interpretable variables (option type, benchmark price, implied volatility) and assigning very little weight to noisy or weakly informative features.

From a modeling perspective, the experiment highlights the complementary strengths of structural and machine learning approaches. BSM offers interpretability, closed-form Greeks and a clear connection to hedging and risk-neutral valuation, while XGBoost adds flexibility and the ability to absorb rich empirical inputs such as volatility-surface quotes. The hybrid specification used here, feeding the BSM price and implied volatility into a boosted-tree model, combines these advantages: BSM provides a strong prior and economically meaningful baseline, and XGBoost learns systematic corrections in a fully data-driven way.

7 Conclusion and Future Work

The main empirical conclusion is that, for the TSLA snapshot considered here, XGBoost significantly outperforms BSM in terms of RMSE while maintaining near-perfect R^2. The classical model achieves respectable accuracy, but the boosted trees consistently shrink pricing errors across the distribution, correct the largest mispricings, and do so robustly across cross-validation folds. At the same time, the feature-importance and validation-curve analyses show that the model's gains come from exploiting economically interpretable structure rather than obscure artifacts.

Several directions for future work follow naturally. One extension is to move from a single-day, single-expiry cross-section to a panel of option-chain data across many trading days and maturities, allowing the model to learn time-varying patterns and regime shifts. Another is to enrich the feature set with more sophisticated volatility forecasts, order-book and liquidity measures, or macroeconomic variables. It would also be interesting to compare XGBoost with architectures specifically tailored to tabular data, such as TabNet, under similar settings [9], and to investigate hybrid ensembles that combine structural models, tree-based methods and deep neural networks [6, 7].

Overall, the TSLA case study suggests that machine learning is best viewed as a powerful complement to classical option pricing theory. In this experiment, XGBoost demonstrates that even when Black–Scholes is already reasonably accurate, a carefully regularized gradient-boosting model enriched with volatility-surface information can deliver substantial improvements in pricing accuracy while preserving interpretability through its dependence on familiar financial features.

References

[1] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[2] A. Bhat. An empirical exploration of the performance of alternative option pricing models: The case of currency options. *Journal of Indian Business Research*, 2018.

[3] Z. Janková. Drawbacks and limitations of the Black–Scholes model for options pricing. *Journal of Financial Studies & Research*, 2018.

[4] P. Sharma and R. Jain. Stock market trends analysis using extreme gradient boosting (XGBoost). In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017.

[5] X. Liu, J. Yang, and Z. Zhou. Deep learning-based modeling of the implied volatility surface. *Journal of Financial Markets*, 2019.

[6] G. Milyushkov. Can machine learning algorithms outperform traditional models for option pricing? *arXiv preprint*, 2025.

[7] B. Ter-Avanesov and H. Beigi. MLP, XGBoost, KAN, TDNN, and LSTM–GRU hybrid RNN with attention for SPX and NDX European call option pricing. *Journal of Mathematical Finance*, 2025.

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

[9] S. Baramal and D. Chauhan. A comparative study of TabNet and XGBoost for tabular data classification. *Journal of Applied Machine Learning*, 2025.

[10] S. Chung. Modelling and forecasting energy market volatility using GARCH and machine learning approach. *arXiv preprint arXiv:2405.19849*, 2024.