A comparison of option pricing models: Black Scholes vs Deep Learning

Haveesh Singirikonda

July 14, 2025

1 Introduction

Options are powerful financial instruments that give traders the right, but not the obligation, to buy or sell an asset at a predetermined price before or at a specified expiration date. These derivatives are widely used for hedging, speculation, and strategic portfolio management, making their pricing an area of significant interest in both academic finance and industry.

There are two primary types of options: call options and put options. A call option gives the holder the right to buy the underlying asset at the strike price, while a put option allows the holder to sell it. Whether an option is profitable or not depends on its relationship to the current market price of the underlying asset. If exercising the option is immediately beneficial, the option is said to be in the money (ITM). Conversely, if exercising would lead to a loss, the option is out of the money (OTM).

Accurately pricing options is crucial for fair market functioning and risk management. Classical models like the Black-Scholes model offer closed-form solutions under simplifying assumptions, but real market data often exhibits behaviours that such models struggle to capture. This motivates the exploration of data-driven methods, such as deep learning, which can model more complex relationships between option features and market prices.

In this project, I compile a dataset of options across several major U.S. stocks (e.g., AAPL, NVDA, TSLA) and use it to compare traditional pricing models like Black-Scholes with machine learning approaches, particularly neural networks. The project includes preprocessing real-world options data (e.g., implied volatility, risk-free rate, time to maturity), assessing model accuracy, and identifying scenarios where data-driven models outperform traditional theory.

2 Data and Methods

The data for equity options is freely available from Yahoo Finance, and can be conveniently accessed programmatically using the Yahoo Finance API. In this project, I utilised the Python package `yfinance` to retrieve the full option-chain data, including both call and put options, for a selection of popular U.S. stocks. As an illustrative example, an option chain for Nvidia expiring on 20 March 2026 can be viewed here.

For this project, I collected option-chain data for the eight major U.S. stocks listed in Table 1, resulting in a dataset comprising approximately 15,000 call options and 14,000 put options. The dataset is visualised in Figure 1, where the strike prices are plotted against the option prices for each stock. The data points are colour-coded according to whether the option is a call or a put, and whether it is in-the-money (ITM) or out-of-the-money (OTM).

Stock Name	Stock Ticker (yfinance)
Apple	AAPL
Nvidia	NVDA
Tesla	TSLA
AMD	AMD
Amazon	AMZN
Microsoft	MSFT
Meta	META
S&P 500	SPY

Table 1: Stocks

The plot reveals a remarkably structured separation into four distinct regions, based on option type and moneyness, which is quite insightful. Notably, the behaviour of Nvidia (NVDA) options stands out: while the options for other stocks follow consistent and expected trends, the Nvidia options display significant deviations. This anomaly likely reflects the unusually high volatility of Nvidia's stock at the time of data collection. Nvidia had recently made headlines by becoming the first company to surpass a 4 trillion USD market capitalisation, which may have triggered an atypically high volume of speculative trading activity. As such, the Nvidia options exhibit characteristics that differ markedly from the more stable patterns seen in other stocks.

For the analysis using the Black-Scholes model, it is essential to incorporate the risk-free interest rates corresponding to the trade dates. These rates were retrieved from the U.S. Treasury's daily yield curve data, available here.

After data collection and preprocessing, the following features were selected as inputs to the neural network models for predicting option prices:

- **strike**: The strike price of the option, i.e., the fixed price at which the underlying asset can be bought (call) or sold (put) upon exercise.
- **stockClosePrice**: The closing price of the underlying stock on the day the option was last traded.
- **deltaT_years**: The time remaining until option expiry, expressed in years. This feature is a core input in analytical models such as Black-Scholes.
- **riskfree_rate**: The risk-free interest rate applicable on the trade date, derived from the U.S. Treasury yield curve.
- **isCall**: A binary indicator denoting whether the option is a call (1) or a put (0).
- **volume**: The total number of contracts traded for the option on the trade date.
- **openInterest**: The total number of outstanding contracts that have not yet been exercised, closed, or expired.
- **impliedVolatility**: The market-implied volatility of the underlying asset, extracted from current option prices.
- **inTheMoney**: A binary feature indicating whether the option is currently in-the-money (1) or not (0), i.e., whether exercising the option would be profitable.
- **isNVDA**: A binary flag indicating whether the option belongs to the NVIDIA (NVDA) stock. This stock-specific indicator was included to help the model account for the unique behaviour observed in NVDA options. The rationale for including this feature is discussed in the following section.

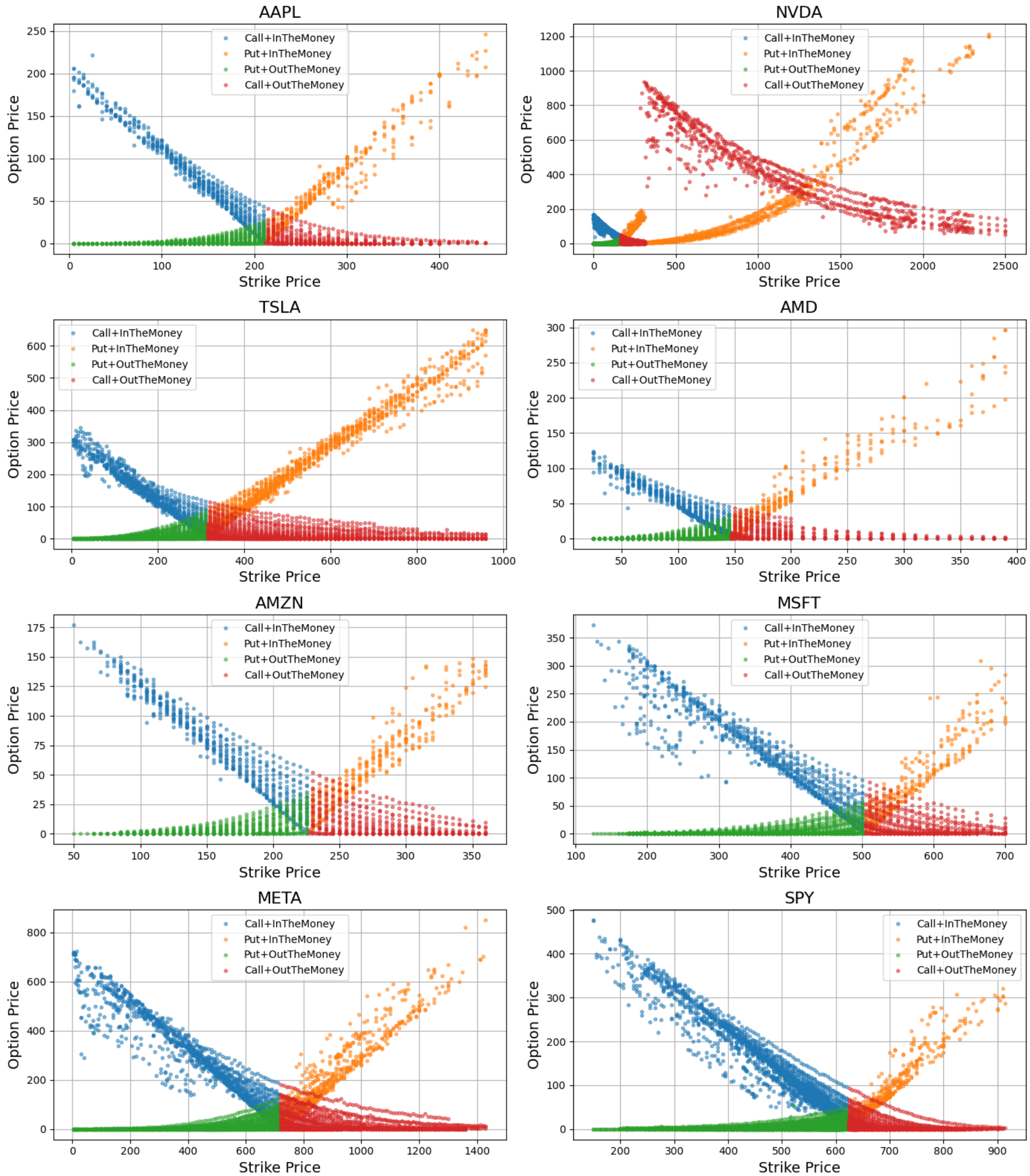

Figure 1: Plot of the strike prices against the price of the option for each stock ticker, and colour coded based on whether the option is a Call or Put option, or whether it is InTheMoney or OutTheMoney.

3 Analysis with the Black Scholes Model

3.1 Black Scholes Model

The Black-Scholes model provides a theoretical estimate for the price of European-style options. For a European call option, the price C is given by:

$$C = S_0\,\Phi(d_1) - Ke^{-rT}\,\Phi(d_2) \tag{1}$$

For a European put option, the price P is:

$$P = Ke^{-rT}\,\Phi(-d_2) - S_0\,\Phi(-d_1) \tag{2}$$

where:

$$d_1 = \frac{\ln(S_0/K) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} \tag{3}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{4}$$

The variables are defined as follows:

- S_0: Current price of the underlying asset
- K: Strike price of the option
- r: Risk-free interest rate (continuously compounded)
- T: Time to maturity (in years)
- σ: Volatility of the underlying asset (annualized)
- $\Phi(\cdot)$: Cumulative distribution function of the standard normal distribution

3.2 Comparison with data

For comparison with the data obtained from `yfinance`, we use the price of the option from its last trade and compare it with the prediction from the Black Scholes model. These results are visualised in Figure 2.

The calculated price of the option from Black Scholes is plotted against the price of the option from its last trade (market price) for each stock ticker, and colour coded based on whether the option is a Call or Put option, or whether it is InTheMoney or OutTheMoney. The plot also shows what a perfect fit would look like ($x = y$ line for the actual price of the option).

From the plots in Figure 2, we can see that the Black-Scholes calculates the price close to the actual price, although there are outliers. But again as we saw in Figure 1, the options for Nvidia stocks are outliers in this dataset, for which there are clear regions where Black Scholes clearly gets the value wrong. The predictions or calculated prices are fine when the market price of the option is low, but for higher prices of the options, Black Scholes gets it completely wrong. This weird behaviour is most probably attributed to the current state of Nvidia stocks. In the figure for Nvidia, we can clearly see that the calculated prices are really higher than the actual price for put options which are in the money (ITM). And the calculated prices are much lower than the actual or market prices for call options which are out the money (OTM). For all the other stocks, regardless of the option type or whether it is in or out the money, most of the points lie close to the perfect fit, with outliers being there, of course.

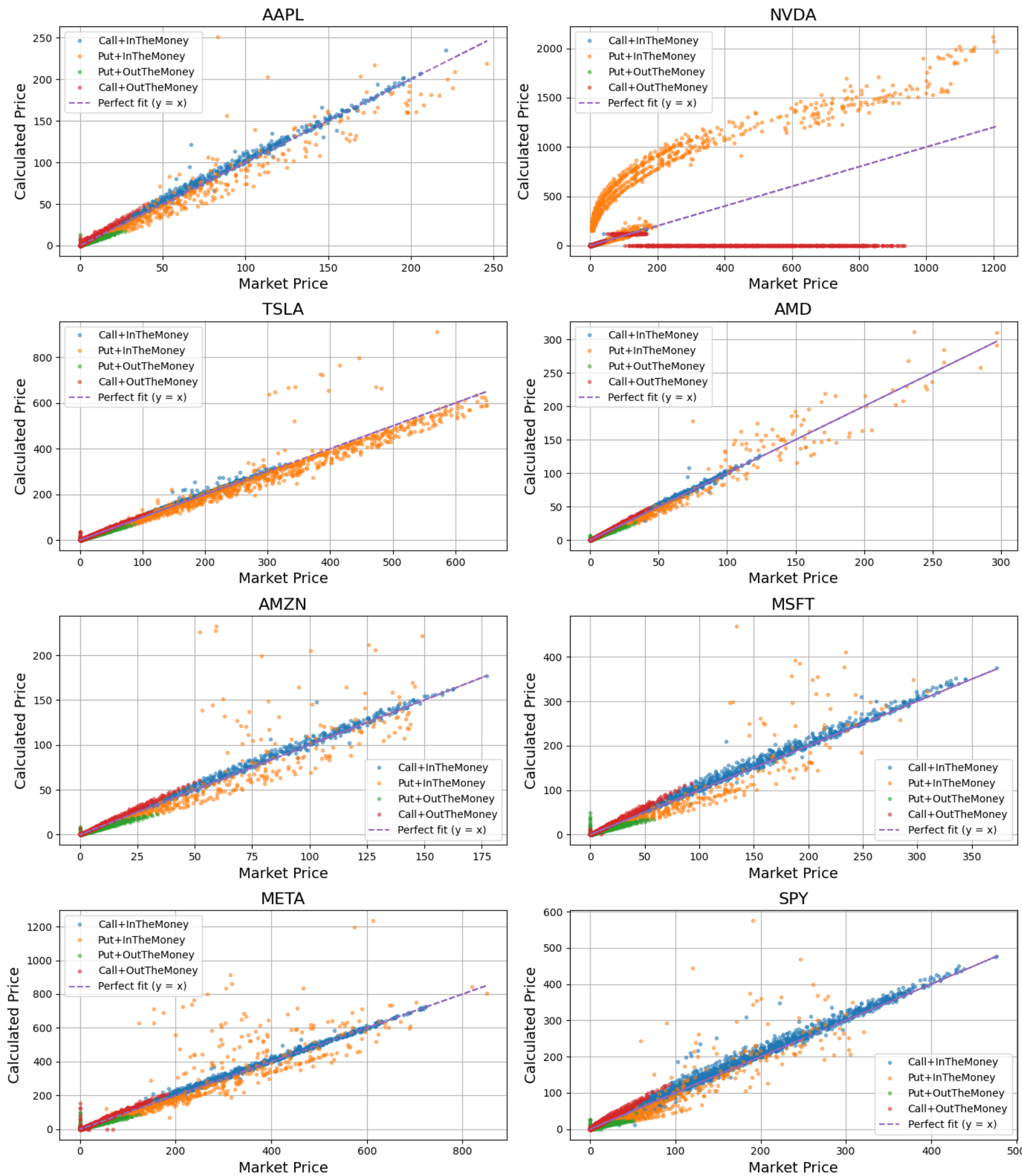

Figure 2: Plot of the calculated price of the option from Black Scholes against the price of the option from its last trade (market price) for each stock ticker, and colour coded based on whether the option is a Call or Put option, or whether it is InTheMoney or OutTheMoney.

Hyperparameters	Model 1	Model 2
Number of layers	3	3
Units in layer 1	64	32
Units in layer 2	64	128
Units in layer 3	64	64
Activation function between layers	ReLU	ReLU
Optimizer	Adam	Adam
Learning rate	0.0005	0.00093
Input feature standardisation	True	False
Epochs	50	50
Batch size	32	32

Table 2: Hyperparameters for the two neural network models used

To quantify the "fit" of Black Scholes model, we can compute the Mean Absolute Error (MAE), defined as:

$$\text{MAE} = \text{mean}\left(|Y_{\text{predicted}} - Y_{\text{actual}}|\right) \tag{5}$$

where Y is the target variable, the option price here. Black Scholes model yields an MAE of 31.4. Given that the mean of the option prices in the dataset is around 80 and the standard deviation is around 120, this MAE is not too bad. But it is not too good either, especially when we see Figure 2, where there are many outliers. Therefore, it is important to see if we can improve upon these fits. In the next section, I will try to improve on this fit using a neural network. To also capture the strange behaviour of the Nvidia stocks, I also include a column "isNVDA" as an input feature for the neural network.

4 Analysis with a Neural Network

Neural Networks need no introduction, and these are one of the most advanced machine learning techniques present at the moment. They use layers of nodes with lots of parameters to successfully learn many kinds of patterns, and I will use them to improve upon Black Scholes and build a better model for option pricing.

4.1 Models

In this project, two fully connected neural network models were developed and compared for the task of option price prediction. The hyperparameters for both models are summarised in Table 2. Both models consist of three hidden layers with ReLU activation functions and were trained using the Adam optimizer for 50 epochs with a batch size of 32.

Model 1 was designed manually with a consistent architecture of 64 units in each hidden layer. The input features were standardised (i.e., normalised to zero mean and unit variance), and a learning rate of 0.0005 was used.

Model 2 differs primarily in its architecture and preprocessing strategy. It was developed using the Keras Tuner, which performs automated hyperparameter optimisation. The resulting configuration uses a smaller first layer (32 units), a larger second layer (128 units), and a final hidden layer with 64 units. Notably, no standardisation was applied to the input features in this case. The optimal learning rate was found to be 0.00093, as selected by the tuner.

Both models share the same input features and output (option price), and serve to highlight the impact of manual vs. automated hyperparameter selection, as well as the importance of feature scaling.

	Black Scholes	Model 1	Model 2
MAE	31.42	4.13	7.90

Table 3: Mean Absolute Error (MAE) values for predictions from Black Scholes and the two neural network models

4.2 Results

The accuracy of the Black-Scholes formula and the two neural network models was evaluated by comparing their predicted option prices against actual market prices on the test dataset. Figure 3 presents scatter plots of predicted vs. actual prices for each model, with the ideal fit indicated by the dashed $y = x$ line.

From visual inspection, the predictions from the Black-Scholes model show many deviations from the market prices, particularly from the Nvidia stocks as seen in Figures 1 and 2. In contrast, the neural network models (Model 1 and Model 2) do not have the same outliers which troubled the Black Scholes model. So the neural networks were able to capture the strange behaviour of the Nvidia stock well. In particular, Model 1 demonstrates a much tighter clustering around the ideal line, indicating a more accurate fit. Model 2 also performs well but appears slightly less precise than Model 1.

These qualitative observations are supported quantitatively by the Mean Absolute Error (MAE) values presented in Table 3. The Black-Scholes model yields a high MAE of 31.42, while Model 1 significantly outperforms it with an MAE of just 4.13. Model 2, though more accurate than Black-Scholes, has a slightly higher MAE of 7.90 compared to Model 1.

Overall, this comparison highlights the potential of neural networks, particularly Model 1, in improving option price predictions over traditional analytical approaches like the Black-Scholes formula.

5 Conclusion and Future Outlook

In this project, we investigated option pricing using both the classical Black-Scholes model and data-driven approaches based on neural networks. We constructed a dataset of over 29,000 options across multiple major U.S. stocks, incorporating features such as strike price, time to maturity, implied volatility, and risk-free rates. The Black-Scholes model, though foundational in financial theory, exhibited significant prediction errors—particularly for American-style options where its assumptions do not strictly hold.

From visual inspection, the predictions from the Black-Scholes model show many deviations from the market prices, particularly for Nvidia stock, as seen in Figures 1 and 2. In contrast, the neural network models (Model 1 and Model 2) do not display the same outliers that challenged the Black-Scholes model. The neural networks were able to capture the unusual pricing behaviour of Nvidia options more effectively. Notably, Model 1 demonstrates a much tighter clustering around the ideal line, indicating a more accurate fit. Model 2 also performs well but appears slightly less precise than Model 1.

Overall, the results show that while the Black-Scholes model provides a useful theoretical benchmark, data-driven methods offer clear advantages in capturing complex market behaviours, particularly for stocks with atypical option pricing dynamics.

There are several directions for improving and extending this work:

- Modelling American Options More Accurately: The Black-Scholes model is designed for European options. Future implementations could replace it with more appropriate methods such as

Figure 3: Comparison of predicted option prices vs. actual market prices for the Black-Scholes model and two neural network models. The dashed line represents the ideal prediction line ($y = x$), where predicted prices perfectly match market prices. Model 1 shows the closest alignment with the actual prices, followed by Model 2, while the Black-Scholes predictions exhibit the largest deviation.

binomial or trinomial trees, finite difference methods, or even analytical approximations for early exercise features.

- Advanced Neural Network Architectures: While our models were simple multilayer perceptrons, more powerful architectures like recurrent neural networks (RNNs), convolutional neural networks (CNNs), or attention-based models could be explored, especially for capturing temporal patterns or hierarchical feature interactions.
- Feature Engineering & Data Enrichment: Including more detailed market microstructure data (e.g. bid-ask spreads over time, volatility surfaces), macroeconomic indicators, or sentiment scores from financial news could enhance model performance.
- Generalisation and Robustness Testing: Evaluating model performance on out-of-sample tickers or different market conditions (e.g. high-volatility periods) would be essential for assessing generalisability.

Overall, this work shows the promise of data-driven models in financial pricing problems, particularly where traditional assumptions may fall short. Combining domain knowledge with machine learning has the potential to significantly improve predictive accuracy in option pricing and beyond.