# K-NN Based Prior for Option Pricing Strategy

Ayan Goswami

## 1 Abstract

We present a non-parametric framework for modeling and trading short-term equity options using K-Nearest Neighbors (KNN) trying to estimate the prior distribution of option price. By predicting standardized option prices from key Greeks and volatility measures, we identify residuals with predictive power for future returns. A directional hypothesis test confirms that under- and over-predicted options exhibit statistically significant forward price behavior. A simulated trading strategy using residual-based signals outperforms randomized baselines with strong significance. This work demonstrates a pragmatic intersection of machine learning and financial theory, providing evidence for short-term inefficiencies in option markets.

## 2 Introduction

This document summarizes the findings reported in `strategy_book.pdf`. The original analysis uses option data to evaluate pricing efficiency and develop a trading approach. Short-dated options have become increasingly popular in modern markets, yet pricing them efficiently remains a challenge. Traditional models such as Black-Scholes assume constant volatility and smooth dynamics, often invalid in practice. We adopt a K-Nearest Neighbors regression model to estimate standardized option prices, bypassing strong parametric assumptions.

## 3 Methodology

Our approach has three key components: (i) exploratory analysis of option features, (ii) KNN-based prediction of standardized prices using Greeks and moneyness, and (iii) residual-based hypothesis testing to detect mispricings that yield directional signals for trading.

### 3.1 Data Loading and Preparation

Option data was loaded from a socket stream leveraging Alpaca's IEX market data (see ./socket), updated every 10 seconds. For the purposes of this investigation, we focused on options with 0 days to expiry (0dte), as they have

the largest trading volume and are hence assumed to self-correct quicker than options with a later expiry. All options with strikes between 99% to 101% of the current price were stored in a csv, along with all the corresponding greeks and implied volatility. To read more about how this was calculated see the Appendix. When this investigation was conducted, SPY hovered around 620$, hence for a given timestamp, 14 or 15 calls and puts were stored. This was done to get an even spread of in the money (ITM), at the money (ATM) and out the money (OTM) options.

Moneyness, defined as the relative position of the underlying asset price to the strike price, exhibits a strong relationship with the option premium. For call options, those that are in the money (ITM), i.e., with strike prices lower than the spot price ($K < S$), tend to have higher intrinsic value and thus higher premiums. Conversely, out of the money (OTM) options ($K > S$) carry mostly time value and are priced lower. This nonlinear relationship is especially pronounced in short-dated options, where the time decay is steep and the implied volatility surface varies across moneyness levels.

There is usually a very steep drop-off in 0dte OTM options' prices, mainly due to the fact that they will expire worthless by the end of the day. To enable fair comparison of option prices across different timestamps and mitigate scale differences between call and put options, we applied a logarithmic normalization procedure to standardize the prices. The transformation is defined as:

$$\text{standardized\_price} = \frac{\log(p+1) - \log(p_{\min}+1)}{\log(p_{\max}+1) - \log(p_{\min}+1)}$$

where $p$ is the latest trade price of the option, and $p_{\min}, p_{\max}$ are the minimum and maximum trade prices for a given option type (call or put) within a specific timestamp. For example, at the first timestamp of the data set: 2025-07-08 08:45:38 (UTC-04:00), with $p_{\max} = 6.31$ and $p_{\min} = 0.04$, a call option with strike = 619 and price $p = 2.71$ would have a standardized price of:

$$\begin{aligned}
\text{standardized_price} &= \frac{\log(2.71+1) - \log(0.04+1)}{\log(6.31+1) - \log(0.04+1)} \\
&= \frac{\log(3.71) - \log(1.04)}{\log(7.31) - \log(1.04)} \\
&= \frac{1.312 - 0.039}{1.989 - 0.039} = \frac{1.273}{1.950} = 0.653
\end{aligned} \tag{1}$$

This way, all options at a given timestamp are contextualized, as the options must be analyzed in context of others to determine over/under valued prices. This transformation is applied separately for call and put options, and performed independently at each timestamp. The rationale is twofold: (1) call and put options have inherently different pricing distributions, and (2) the market conditions change across timestamps, requiring local normalization to preserve intra-timestamp price structure.

Logarithmic scaling was chosen to compress the skewness in price distribution, especially in deep ITM or OTM options where price differences can be exponential. Adding 1 inside the logarithm avoids issues with near-zero prices. The result is a normalized feature `standardized_price` bounded in $[0, 1]$, suitable for downstream tasks such as KNN modeling and residual analysis. The following is a random timestamp slice:

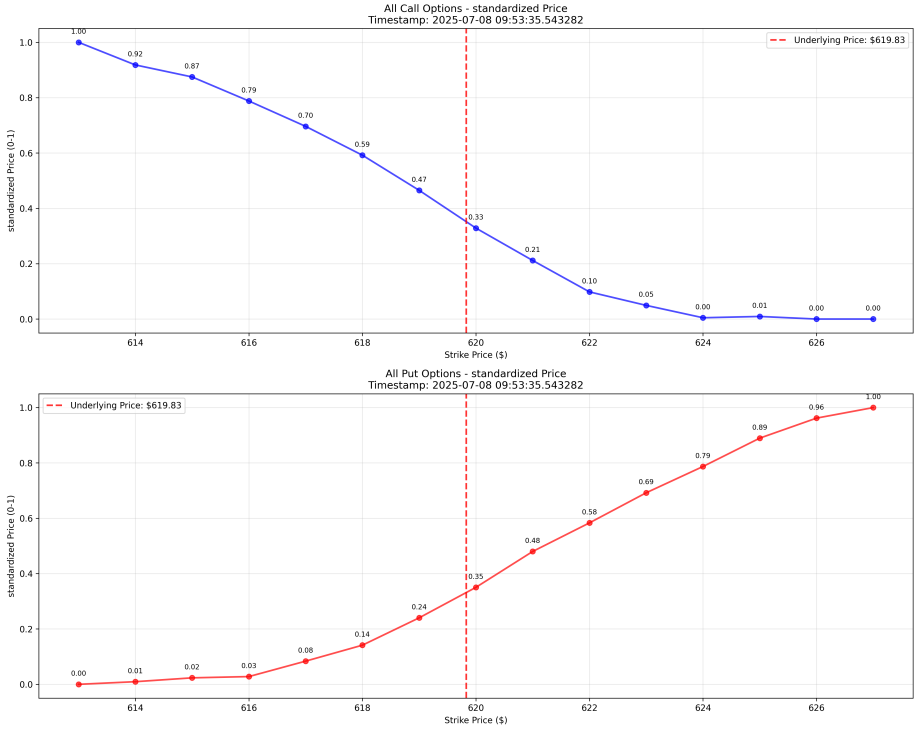


Figure 1: Standardized option prices for a given strike

Option data were loaded from several processed files and split into training and validation sets using a 56%/14%/30% split for training, validation and testing. Since the $\theta$ decay is so high for 0dte options, we focused on trade entry and exit windows between 30 seconds to 5 minutes. This way the $\alpha$ can be measured reliably without accounting for price decay. Hence, a column was appended to our dataset called `price_diff`, which measured:

$$\text{price\_diff}_x = \frac{\frac{1}{x}\sum_{i=t}^{t+x-1} \text{price}_i}{\text{price}_t}$$

where x is the look-forward period in tens of seconds, and t is the current time period. This will serve as a metric of success and efficacy in our paper. The top and bottom 5% were trimmed in this data set to mitigate the risk

of outlier-driven distortion in both the residual distribution and downstream performance metrics, such as the estimated significance of prediction errors.

## 3.2 KNN Modeling

K-Nearest Neighbors (KNN) is a non-parametric regression algorithm that predicts the output of a data point based on the average of its $k$ nearest neighbors in the feature space [3]. Unlike parametric models, KNN does not assume a functional form for the relationship between input variables and the target. Instead, it leverages local structure in the data to make predictions. This makes KNN particularly useful in financial contexts where relationships between variables may be complex, noisy, or nonlinear [1]. Our model predicts standardized option prices based on option greeks, volatility and moneyness (see Appendix). A wrapper function scales predictors and returns both root mean squared error (RMSE) and $R^2$ values. Exhaustive subset selection determined that `delta`, `gamma`, `moneyness`, and `rho` gave the best performance with RMSE 0.0269 and $R^2 = 0.9946$. The $k$ value used in the analysis was $\sqrt{n}$ where $n$ is the number of rows in the training set. This is standard is most analyses We define the KNN estimator:

$$\hat{y}_t = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x_t)} y_i$$

where $\mathcal{N}_k(x_t)$ is the set of $k$ nearest neighbors of feature vector $x_t$.

Optimal $k$ was found by evaluating a range of values and occurred at $k = 3$. Such a low $k$ indicates a complex space, and from our analysis it is possible that the optimal $k$ is 1. $k = 1$ would require discussion of the bias-variance trade-off; $k = 3$ is our compromise in this regard. The following is the performance of the model on the validation data.

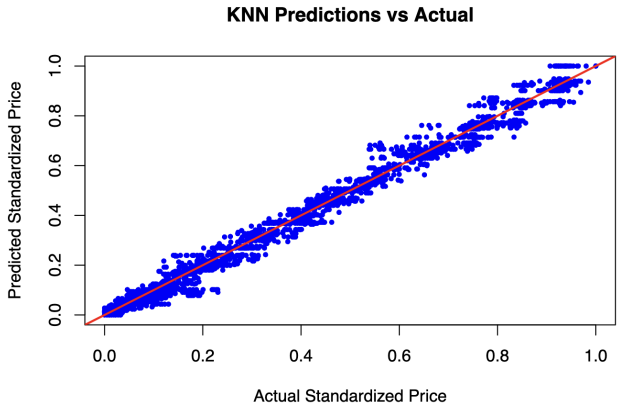


Figure 2: Predicted vs Actual Standardized prices

### 3.3 Residual Analysis and Hypothesis Testing

Residuals are defined as:

$$r_t = \hat{y}_t - y_t$$

We categorize residuals exceeding the 95th percentile of the empirical distribution as significant. Predictions greater than this threshold $r_t > \bar{r}$ , are interpreted as undervalued options, where the model overestimates the price relative to observed values. Conversely, predictions with residuals below the negative threshold, $r_t < -\bar{r}$, are considered overvalued, where the model underestimates the price. These cases form the basis for directional trading signals. A Shapiro-Wilk normality test revealed a very high confidence that the residuals were normally distributed, hence we were justified in using the same threshold in both cases. The key hypothesis tested is:

$$H_1 : \mu_{\text{GEQ}} > \mu_{\text{All}} > \mu_{\text{LEQ}}$$

where $\mu_{\text{GEQ}}$, $\mu_{\text{LEQ}}$, and $\mu_{\text{All}}$ are the mean future price differences ($price\_diff$) for positive, negative, and all residuals, respectively. Directional t-tests were conducted to confirm significance for the time windows.

The null hypothesis was rejected for the six-period window with p-values below 0.05. The alternative hypothesis was partially supported in several horizons.

| Period (10s) | Mean (Under) | Mean (Over) | Mean (Overall) | $P(\mu_{\text{GEQ}} = \mu_{\text{All}})$ | $P(\mu_{\text{All}} = \mu_{\text{LEQ}})$ |
|---|---|---|---|---|---|
| 3 | 1.0111 | 0.9932 | 1.0019 | 0.0742 | 0* |
| 6 | 1.0125 | 0.9915 | 0.9991 | 0.0265* | 0.0006* |
| 12 | 1.0114 | 0.9889 | 0.9989 | 0.0504 | 0.0003* |
| 15 | 1.0086 | 0.9897 | 0.9975 | 0.0818 | 0.0052* |
| 30 | 1.0028 | 0.9889 | 0.9961 | 0.2437 | 0.0405* |

Table 1: Mean future price ratios by residual group and associated $p$-values. * indicates significance at $p = 0.05$.

## 4 Trading Simulation

### 4.1 Signal Generation and Trade Execution

Trading signals were generated on the test dataset using KNN model predictions. Long positions were initiated when residuals exceeded the 95th percentile threshold $r_t > \bar{r}$, indicating undervalued options. Short positions were initiated when residuals fell below the negative threshold $r_t < -\bar{r}$, indicating overvalued options.

For each trade, the profit or loss was calculated using the future price ratio $price\_diff_x$:

$$\alpha_{\text{long}} = (price_t \cdot price\_diff_x) - price_t,$$
$$\alpha_{\text{short}} = price_t - (price_t \cdot price\_diff_x).$$

### 4.2 Performance Evaluation

Based on the hypothesis testing results, we implemented the trading strategy using the 6-period horizon, where the null hypothesis was fully rejected. The strategy executed 5,268 trades and generated $1,685 in cumulative profit.

### 4.3 Statistical Significance Testing

To assess the statistical significance of our results, we conducted a randomized baseline comparison. We test the null hypothesis:

$$H_0 : \alpha_{\text{strategy}} = \alpha_{\text{random}}$$

where $\alpha_{\text{strategy}}$ is the cumulative return from our residual-based trading signals and $\alpha_{\text{random}}$ is the expected return from random trading.

For each of 100 Monte Carlo simulations, we randomly selected $N = 5,268$ trades from the test dataset and assigned random long/short positions with equal probability:

$$\alpha_{\text{random},i} = \begin{cases} (price_{t,i} \cdot price\_diff_{6,i}) - price_{t,i} & \text{with probability 0.5} \\ price_{t,i} - (price_{t,i} \cdot price\_diff_{6,i}) & \text{with probability 0.5} \end{cases}$$

The cumulative return for each simulation was calculated as $\sum_{i=1}^{N} \alpha_{\text{random},i}$. The empirical null distribution yielded a 95th percentile threshold of $445.64. Since our observed return of $1,685 exceeds this threshold, we reject $H_0$ and conclude that the strategy performs significantly better than random selection at the 5% significance level.

## 5 Results

Figure 3 displays the average future price changes stratified by residual group across different forecast horizons. Options with high residuals (where the model overestimates actual prices) consistently exhibit higher mean price ratios compared to options with low residuals (where the model underestimates actual prices). This pattern supports the directional hypothesis that KNN model residuals contain predictive information for short-term price movements.

## 6 Discussion

The residual-based signals produced consistent return differences in favor of positive and negative residuals. The overvalued short trades yielded significantly lower p-values than the undervalued long trades in Table 1, suggesting that the model's ability to identify overpriced options may be more reliable than its capacity to detect underpriced ones.

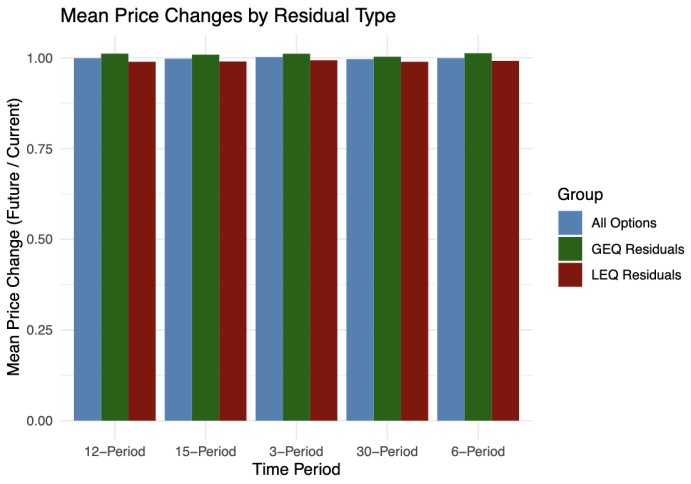


Figure 3: Mean future price ratios by residual group

The cross validation in selecting the optimal $k$ used a sequencing function to generate a list of numbers as a function of $n$ where $n$ is the number of rows. However, this was done arbitrarily, to reduce computational overhead. The optimal $k$ could fall outside of this range, and may in fact be 1. The figure below illustrates this fact, showing the residual mean squared error decrease greatly when approaching 1.

It is possible that Nearest-neighbor interpolation may be the best model fit, however it will lead to a large amount of variance in this model. While nearest-neighbor interpolation ($k = 1$) may offer the best local fit in terms of minimizing bias, it typically results in high variance and overfitting, particularly in noisy or high-dimensional datasets [2]. For this reason, choosing a $k$ greater than 1 is strongly recommended in non-spatial applications such as financial modeling, where data is not naturally embedded in low-dimensional geometric space.

From a Bayesian perspective, the KNN model can be interpreted as providing a data-driven prior distribution for option prices conditional on the selected Greeks. The residuals $r_t = \hat{y}_t - y_t$ represent deviations from this prior, effectively capturing information not encoded in the feature set. When these residuals exhibit predictive power for future price movements, they suggest that the market has not fully incorporated all available information into current prices, creating exploitable inefficiencies.

The posterior belief about an option's true value can be viewed as a combination of the KNN prior estimate and the residual information. Options with large positive residuals (where $\hat{y}_t > y_t$) indicate that the model's prior suggests higher value than the market price, potentially signaling undervaluation. Conversely, large negative residuals suggest overvaluation relative to the model's

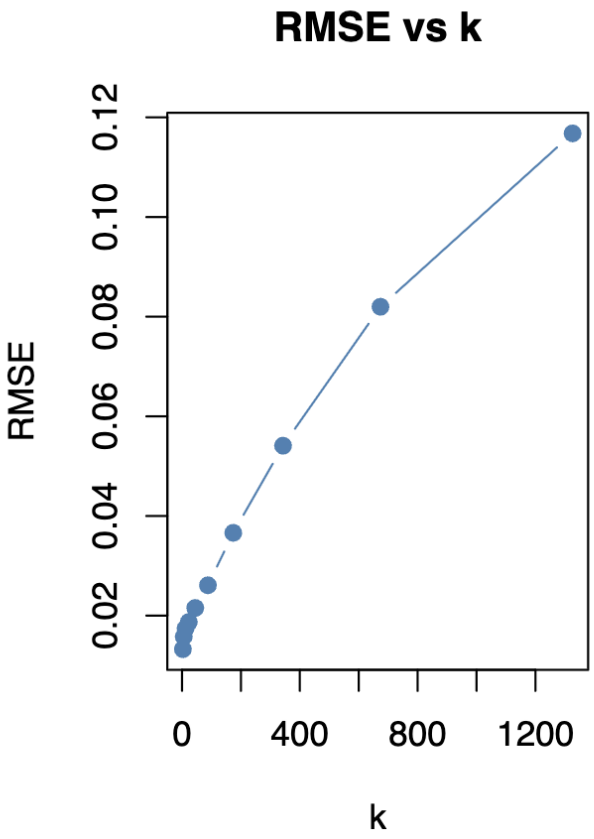


Figure 4: KNN RMSE by k value

expectations.

While the profits observed in backtesting were modest, the approach performed significantly better than random trading according to the null threshold. However, we assume zero execution friction; incorporating slippage and fees is a promising direction for future refinement. Though for this analysis, KNN served as an effective non-parametric estimator for the conditional distribution of standardized option prices given `delta`, `gamma`, `moneyness`, and `rho`, more sophisticated models such as random forests and boosted trees could potentially capture additional nonlinear relationships [2].

To better understand the drivers of trade success, we conducted K-means clustering on executed trades using key predictive features: `delta`, `gamma`, `moneyness`, and `rho`.

A Chi-squared test was conducted to determine if $P(\text{Profit}|\text{Cluster}) \neq P(\text{Profit})$ at $p = 0.05$. This analysis revealed structurally distinct trade clusters with sig-

nificantly different profit probabilities, particularly among short-sell trades. The following figure shows the 10 clusters created, by profitability.

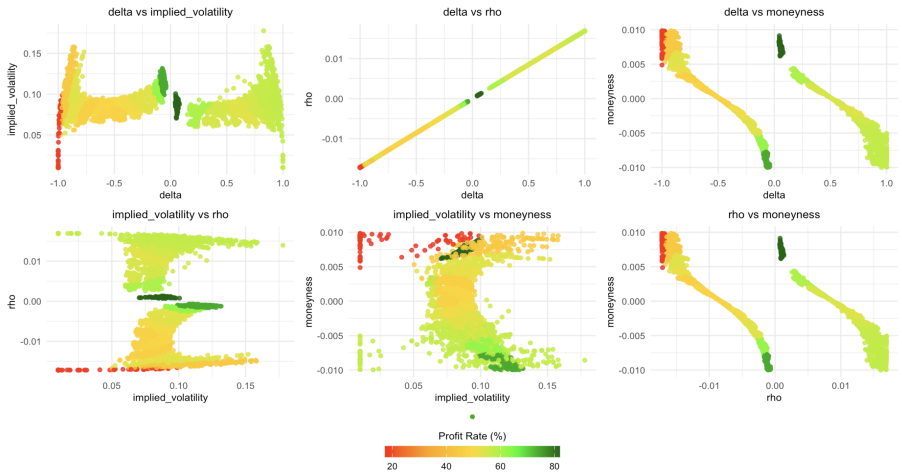


Figure 5: Clusters by profitability

This figure shows us the trends in our data, such as OTM puts being particularly profitable. These clusters can help us identify areas of weakness, and add an additional layer of robustness to our trade execution. This can be done with the help of a rule engine, which only allows trades within the most profitable clusters.

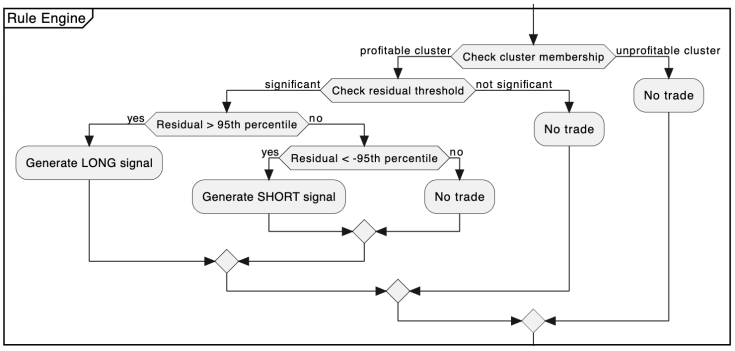


Figure 6: Rules to execute trade

# 7 Conclusion

This paper introduces a residual-driven K-Nearest Neighbors framework for detecting and exploiting short-term mispricings in equity options. By predicting

standardized option prices using a carefully selected subset of option Greeks, we demonstrate that the residuals from the KNN model carry directional information about future price movements. Through hypothesis testing , we identify statistically significant deviations in residuals that serve as actionable trading signals.

A trading strategy built on these residual-based signals achieves performance that significantly exceeds a randomized baseline, generating over $1600 in profit across thousands of trades and surpassing the empirical 95th percentile of the Monte Carlo simulated outcomes. This finding underscores the presence of persistent inefficiencies in the short-term options market that can be captured using non-parametric techniques with minimal assumptions.

Beyond its practical implications, our framework showcases how interpretable machine learning models can be fused with financial domain knowledge to produce both profitable and statistically sound trading strategies. Future work may extend this framework by incorporating Bayesian uncertainty estimation, dynamic $k$ selection, or hybrid ensemble models. Additionally, analyzing how residual signals evolve over intraday time segments could further enhance signal quality and timing.

Conclusively, the approach outlined here represents a potentially scalable, data-driven methodology that bridges predictive modeling with hypothesis-driven market interpretation—offering both theoretical insight and empirical edge.

# A Black-Scholes Formulas

## A.1 Option Pricing Formulas

The Black-Scholes model for European option pricing is given by the following formulas:

### A.1.1 Call Option Price

$$C(S, K, T, r, \sigma) = S \cdot N(d_1) - Ke^{-rT} \cdot N(d_2) \tag{2}$$

### A.1.2 Put Option Price

$$P(S, K, T, r, \sigma) = Ke^{-rT} \cdot N(-d_2) - S \cdot N(-d_1) \tag{3}$$

### A.1.3 Parameters $d_1$ and $d_2$

$$d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \tag{4}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{5}$$

where:

- $S$ = Current stock price

- $K$ = Strike price
- $T$ = Time to expiration (in years)
- $r$ = Risk-free interest rate
- $\sigma$ = Volatility of the underlying asset
- $N(\cdot)$ = Cumulative distribution function of the standard normal distribution

## A.2 Option Greeks

The Greeks measure the sensitivity of option prices to various factors:

### A.2.1 Delta

Measures the rate of change of option price with respect to changes in the underlying asset's price.

**Call Option Delta:**

$$\Delta_{call} = N(d_1) \tag{6}$$

**Put Option Delta:**

$$\Delta_{put} = N(d_1) - 1 \tag{7}$$

### A.2.2 Gamma

Measures the rate of change of delta with respect to changes in the underlying price.

$$\Gamma = \frac{N'(d_1)}{S\sigma\sqrt{T}} = \frac{e^{-\frac{d_1^2}{2}}}{S\sigma\sqrt{2\pi T}} \tag{8}$$

Gamma is the same for both call and put options.

### A.2.3 Theta

Measures the rate of change of option price with respect to the passage of time (time decay).

**Call Option Theta:**

$$\Theta_{call} = -\frac{SN'(d_1)\sigma}{2\sqrt{T}} - rKe^{-rT}N(d_2) \tag{9}$$

**Put Option Theta:**

$$\Theta_{put} = -\frac{SN'(d_1)\sigma}{2\sqrt{T}} + rKe^{-rT}N(-d_2) \tag{10}$$

Theta is typically expressed in value per day, dividing by 365.

### A.2.4 Vega

Measures the rate of change of option price with respect to changes in volatility.

$$Vega = S\sqrt{T}N'(d_1) \tag{11}$$

Vega is the same for both call and put options and is typically expressed as change per 1% change in volatility.

### A.2.5 Rho

Measures the rate of change of option price with respect to changes in the risk-free interest rate.

**Call Option Rho:**

$$\rho_{call} = KTe^{-rT}N(d_2) \tag{12}$$

**Put Option Rho:**

$$\rho_{put} = -KTe^{-rT}N(-d_2) \tag{13}$$

Rho is typically expressed as change per 1% change in interest rate.

## A.3 Implied Volatility

Implied volatility is the volatility value that, when input into the Black-Scholes formula, yields a theoretical option price equal to the market price. It is typically solved using numerical methods such as the Newton-Raphson method:

$$\sigma_{n+1} = \sigma_n - \frac{BS(S, K, T, r, \sigma_n) - Market\_Price}{Vega} \tag{14}$$

where $BS(\cdot)$ is the Black-Scholes pricing function and iterations continue until convergence.

## A.4 Put-Call Parity

For European options on non-dividend paying stocks:

$$C + Ke^{-rT} = P + S \tag{15}$$

This relationship can be used to derive the price of a put option from a call option with the same strike and expiration, or vice versa.

# B References

# References

[1] Naomi S Altman. *An introduction to kernel and nearest-neighbor nonparametric regression.* The American Statistician, 1992.

[2] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[3] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.