Understanding Option Pricing Models

Ramachandran S

July 28, 2024

1 Introduction

In this project, I explored the fundamental principles of option pricing by implementing popular models, specifically the Black-Scholes and Binomial models. The goal was to understand how these models calculate the theoretical prices of options based on various input parameters, and to compare the results against actual market prices to gauge their accuracy and effectiveness.

2 Objectives

- To understand the theoretical foundations of option pricing models.
- To implement the Black-Scholes and Binomial models in Python.
- To analyze and compare the theoretical option prices with market data.

3 Methodology

3.1 Understanding Option Pricing Models

Options are financial derivatives that give the holder the right, but not the obligation, to buy or sell an underlying asset at a predetermined price (strike price) before or on a specific date (expiration date). The two primary models used for pricing options are:

- **Black-Scholes Model**: A closed-form solution used for pricing European options.
- **Binomial Model**: A discrete-time model that can be used for pricing American and European options.

3.2 Model Implementation

1. **Black-Scholes Model**: The model calculates the price of European call and put options based on the current stock price, strike price, time to expiration, risk-free interest rate, and volatility of the underlying asset.

$$
\begin{aligned}
C &= S_0 N(d_1) - K e^{-rt} N(d_2) \\
d_1 &= \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right) t}{\sigma\sqrt{t}} \\
d_2 &= d_1 - \sigma\sqrt{t}
\end{aligned}
$$

2. **Binomial Model**: The model constructs a binomial tree to calculate option prices iteratively. It allows for the pricing of American options, which can be exercised at any time before expiration.

Implementation Code The following Python code snippets were used to implement the models:

3.2.1 Black-Scholes Implementation

```
import numpy as np
import scipy.stats as si

def black_scholes(S, K, T, r, sigma, option_type='call'):
    d1 = (np.log(S / K) + (r + (sigma ** 2) / 2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)

    if option_type == 'call':
        option_price = (S * si.norm.cdf(d1) - K * np.exp(-r * T) * si.norm.cdf(d2))
    else:
        option_price = (K * np.exp(-r * T) * si.norm.cdf(-d2) - S * si.norm.cdf(-d1))

    return option_price
```

3.2.2 Binomial Model Implementation

```
def binomial_option_price(S, K, T, r, sigma, N, option_type='call'):
    dt = T / N
    u = np.exp(sigma * np.sqrt(dt))
    d = 1 / u
    p = (np.exp(r * dt) - d) / (u - d)

    # Initialize asset prices at maturity
    asset_prices = np.asarray([S * (u ** j) * (d ** (N - j)) for j in range(N + 1)])

    # Initialize option values at maturity
```

```
if option_type == 'call':
    option_values = np.maximum(0, asset_prices - K)
else:
    option_values = np.maximum(0, K - asset_prices)

# Backward induction for option pricing
for i in range(N - 1, -1, -1):
    option_values = np.exp(-r * dt) * (p * option_values[1:] + (1 - p) * option_values[

return option_values[0]
```

3.3 Data Analysis

The option prices generated from the models were compared to actual market prices. This involved:

- Collecting historical market prices for specific options.
- Analyzing discrepancies between model outputs and market prices to identify potential improvements in the models or the need for different parameters.

4 Results

The results showed that the Black-Scholes model performed well for European options but had limitations when compared to the actual market prices, particularly in the presence of volatility smiles. The Binomial model provided more flexibility and was able to price American options more accurately due to its iterative nature.

4.1 Findings

- The Black-Scholes model is effective for European options, but adjustments are needed for more volatile markets.
- The Binomial model provides better accuracy for American options, as it can account for the option's early exercise feature.

5 Conclusion

This project deepened my understanding of option pricing theories and the practical implementation of pricing models using Python. The comparison of theoretical prices with market data emphasized the importance of model selection and the parameters involved in accurate option pricing.

6 Future Work

Future work could include:

- Exploring additional option pricing models, such as the Monte Carlo simulation method.
- Enhancing the Binomial model to include more complex scenarios and multiple underlying assets.
- Implementing machine learning techniques to predict volatility and refine pricing accuracy.

7 References

- Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. Journal of Political Economy, 81(3), 637-654.
- Hull, J. C. (2017). Options, Futures, and Other Derivatives. Pearson.
- https://pypi.org/project/scipy/