Pricing Options

Monte Carlo, Binomial Tree, Black Scholes Modeling

Pratham Rawat, Jerry Chen, Arya Patel, Iram Liu

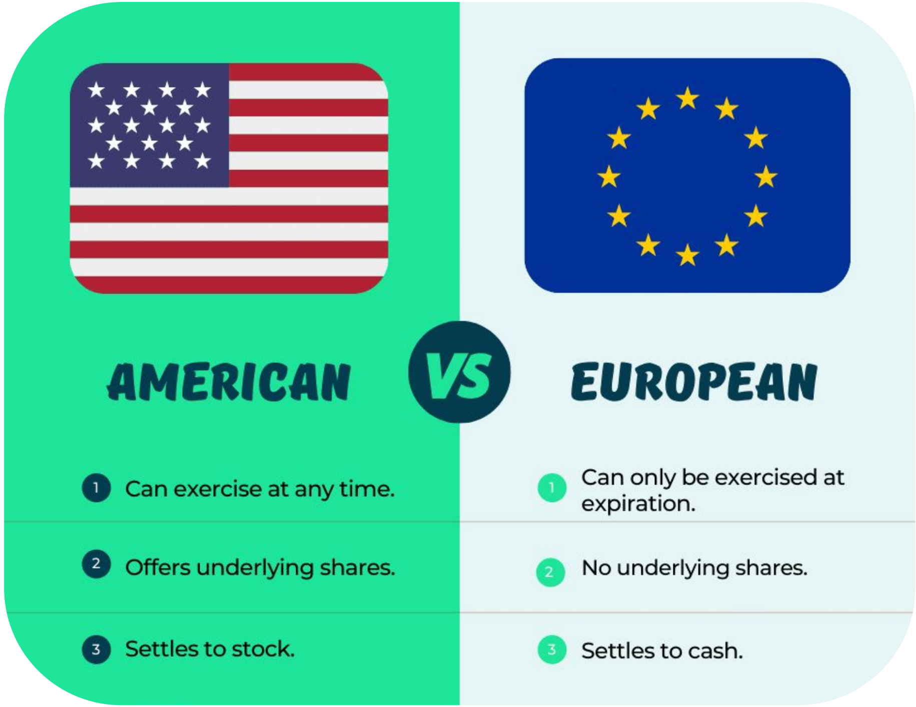
What Are Options?

- Can “place bets” on the price of a stock (or any underlying asset) in the future using derivatives
- Options are one of the four types of derivatives
- Buyer of an option pays a premium to the seller for the right to buy or sell the underlying asset, but is not obligated to do so
- No money is generated, but rather exchanged
- Strike price: the price at which the underlying asset can be bought or sold by the option holder upon exercise of the option contract, determined when the option is issued

Two Types of Options: Calls and Puts

	CALL OPTION	PUT OPTION
Like…	**INSURANCE** on a stock	**DOWN PAYMENT** on a stock
Buyer	Right to **BUY** at the strike price	Right to **SELL** at the strike price
Hope	Stock price **INCREASES**	Stock price **DECREASES**

Context: European vs American Options



Source: Project Finance

Binomial Tree Model

- By assuming a perfectly efficient market, we can be certain about the value of a portfolio consisting of a stock and option
- The return must equal the risk-free interest rate since this portfolio has no risk
- Since we also know that there is only two securities and only two possible outcomes for this portfolio, we can calculate the cost of setting up this portfolio = price of the options

$$p = \frac{e^{rT} - d}{u - d}$$

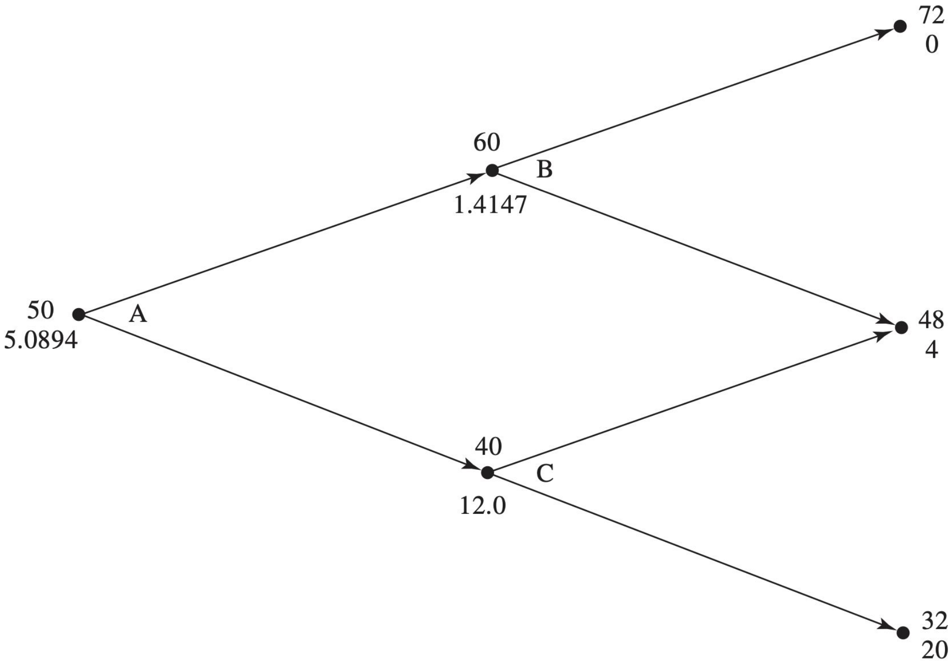
$$f = e^{-rT}[p f_u + (1 - p) f_d]$$

Price of an options contract = cost of setting up risk free portfolio

Probability of upward movement over a single time step
Assuming that investors are risk neutral

Source: Options, Futures, & Other Derivatives 9th Edition J. Hull Ch. 13

Binomial Tree Implementation



Monte Carlo Model

- "Random" simulations of possible options price paths
- Geometric Brownian Motion (GBM) to simulate time passing and model stock price behavior (discrete time model)
- Explaining Formula **21.17**
  - û is the expected % return in a risk neutral world and σ (or sigma) is the volatility.
  - S(T) denotes the value of S at time T where ε is a random sample from a normal distribution with mean zero and standard deviation of 1.0.
  - $\sigma\varepsilon\sqrt{T}$ known as the "stochastic component" of return
- Use formula to construct paths of S
- Iterate thousands to millions of times (more accurate as N iterations increases)

$$S(T) = S(0)\exp\left[\left(\hat{\mu} - \frac{\sigma^2}{2}\right)T + \sigma\epsilon\sqrt{T}\right] \qquad \textbf{(21.17)}$$

This equation can be used to value derivatives that provide a nonstandard payoff at time T. As shown in Business Snapshot 21.2, it can also be used to check the Black–

Source: Options, Futures, & Other Derivatives 9th Edition J. Hull Ch. 21

Monte Carlo (EU) Implementation

Simple model assuming constant volatility & interest rates:

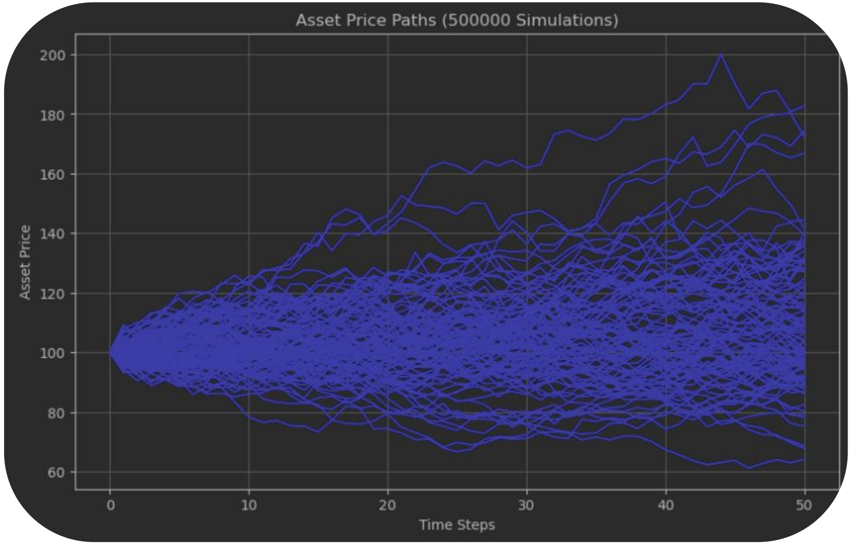
1. **Sample random path S in risk neutral environment**
2. **Calculate payoff depending on call or put**
3. **Repeat 1 & 2: 1 Million Iterations**
4. **Calculate mean of sample payoffs**
5. **Discount at risk free interest rate to evaluate derivative final price**

Source: Options, Futures, & Other Derivatives 9th Edition J. Hull Ch. 21

Monte Carlo (NA) Implementation

- Uses similar approach to EU, except since now we can exercise the option at any time T, we can use something call the Longstaff-Schwartz Method (or **LSM**) to solve a linear regression problem at each time step (from final step and works backwards) in order to calculate in-the money paths

- The regression equation is used to estimate the payoff at each time step, which is compared to the intrinsic value (difference between current market price and strike price) to determine whether the option should be exercised or held to wait for next time t



Note: Learn more about it at this [paper]

Optimizing Run-Time Using C++ & Multithreading

- American Monte Carlo (simple model)
  - Single-threaded runtime: 23515 ms
  - Multi-threaded runtime: 2530 ms
    - ~10x performance increase
    - Median of 3 runs on i7 12650h, RTX 3060 (500K simulations)
- European Monte Carlo
  - Single-threaded runtime: 26944 ms
  - Multithreaded runtime: 2990 ms
    - ~10x performance increase
    - Median of 3 runs on i7 12650h, RTX 3060 (1M simulations)

- Further optimizations possible other than using a vector to store all path calculations from separate threads and lock/unlocking it

Note: Running American pricing models and Binomial Tree is more computationally expensive

Black-Scholes Model

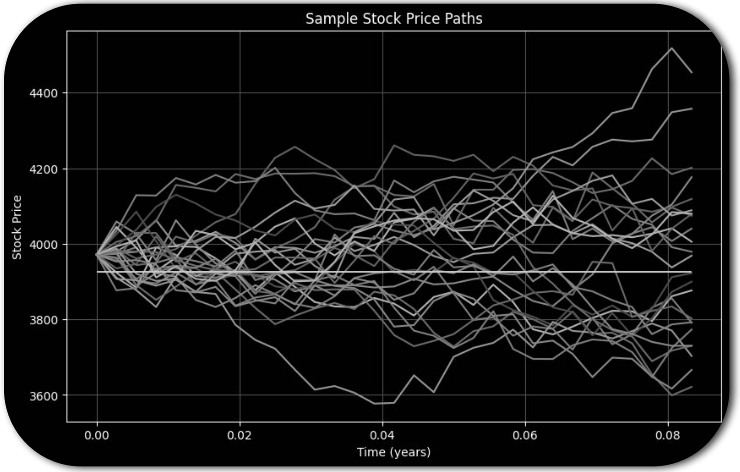
- Most popular methodology for options pricing because of its simplicity, flexibility, accuracy, and widespread adoption
- Closed form solution for an option's price
- Core Assumptions:
  - Options exercised only at expiration date— European-style options only
  - Stock Prices can be modeled as a Stochastic Process
  - Stock prices follow a log-normal distribution so returns are normally distributed
  - Constant volatility
- This allows us to apply Itô's Lemma to the stochastic process formula
- Do a little bit of math and stuff to find a closed form options price formula:

$$S(T) = S(0)\exp\left[\left(\hat{\mu} - \frac{\sigma^2}{2}\right)T + \sigma\epsilon\sqrt{T}\right] \qquad \textbf{(21.17)}$$

This equation can be used to value derivatives that provide a nonstandard payoff at time T. As shown in Business Snapshot 21.2, it can also be used to check the Black–

Black-Scholes (EU) Implementation

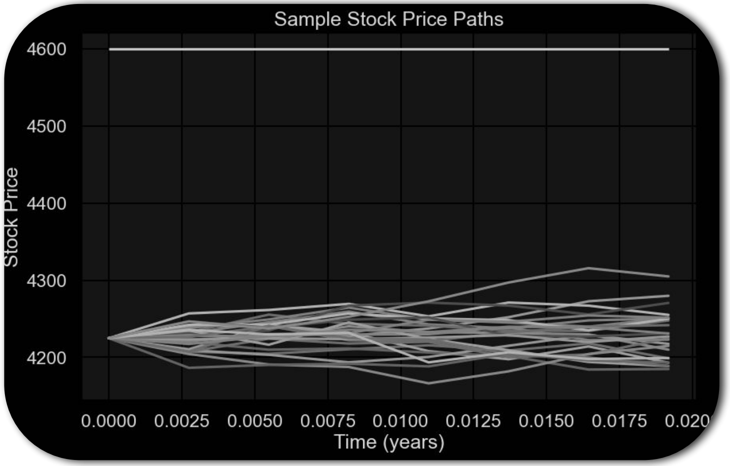
Fig 3. Random simulations for SPX put option
(sigma = 0.2012, T = (1/12), K = 3925.00, S = 3970.99)



Similar estimated price generated as the monte-carlo method

- $65.64 (Black Scholes)
- $65.66 (Monte-Carlo)

Fig 4. Random simulations for SPX put option
(sigma = 0.2012, K = 3925.00, S = 3970.99)

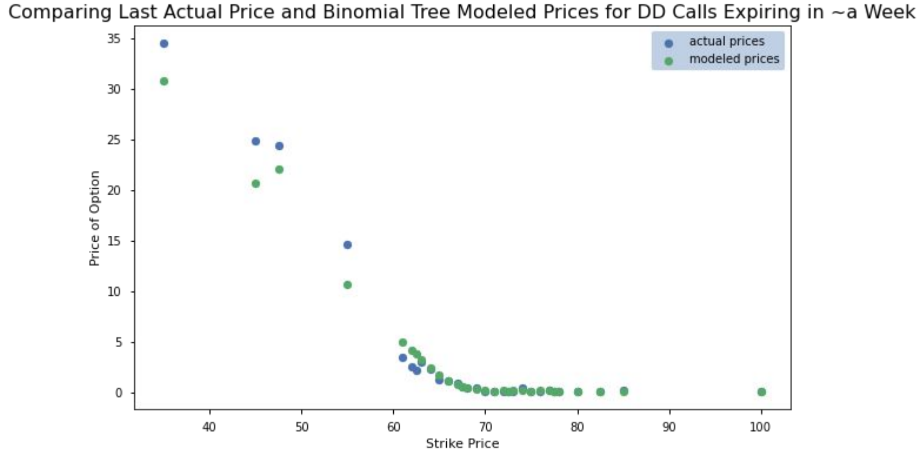


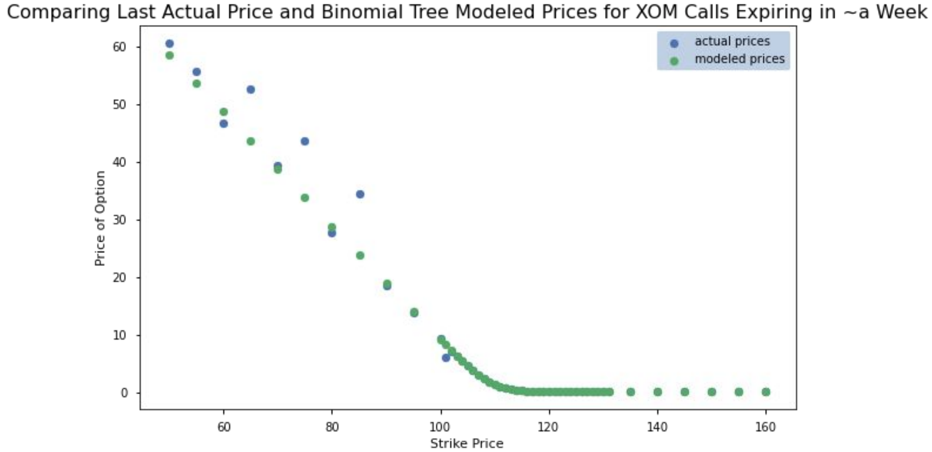
Similar estimated price generated as the monte-carlo method

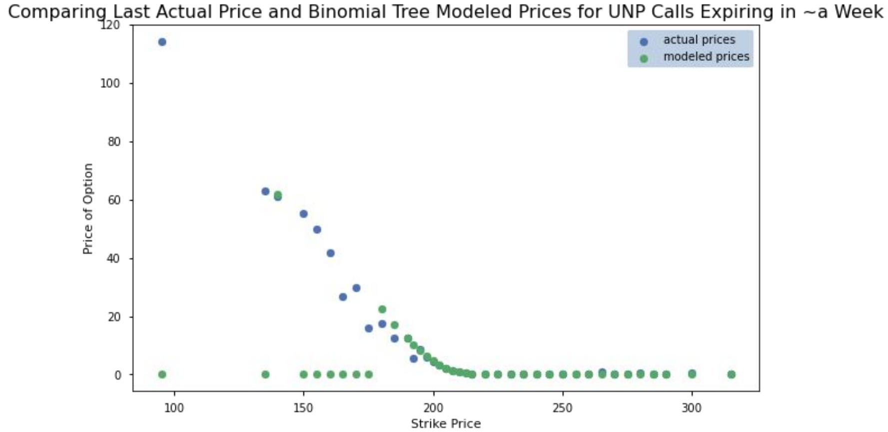
- $375.14 (Black Scholes)
- $375.11 (Monte-Carlo)

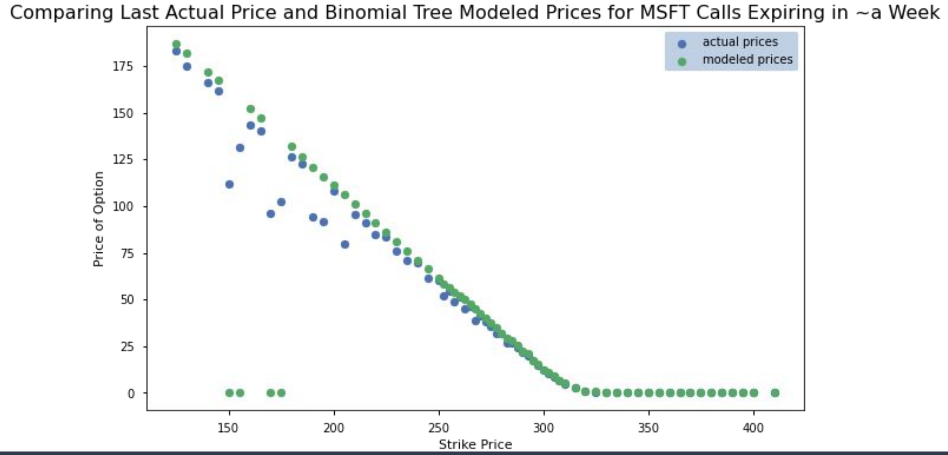
Source: Options, Futures, & Other Derivatives 9th Edition J. Hull Ch. 15

Results

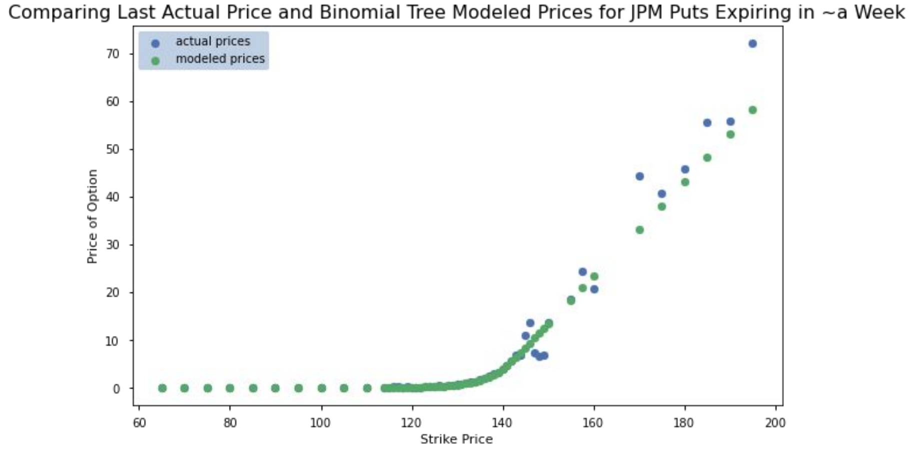


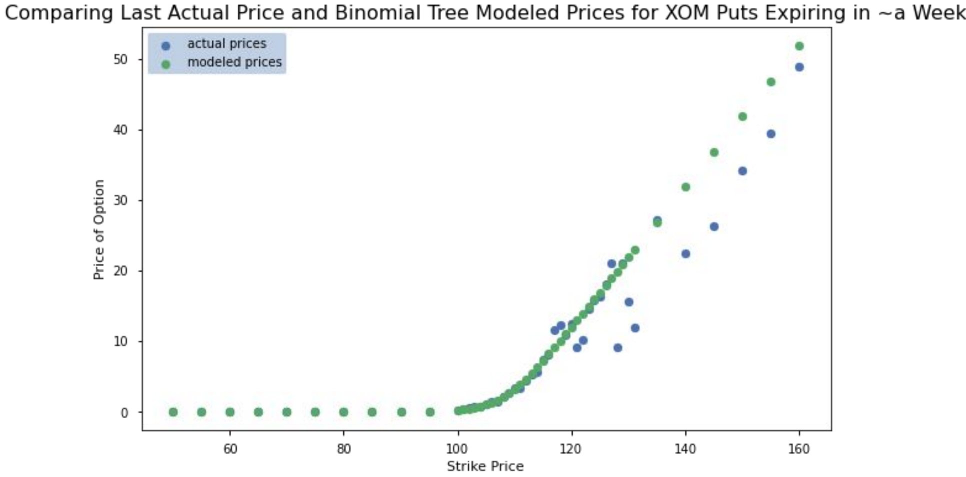


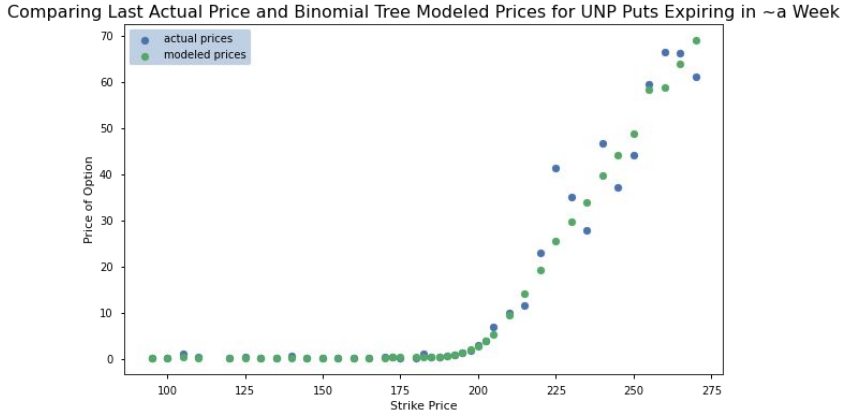


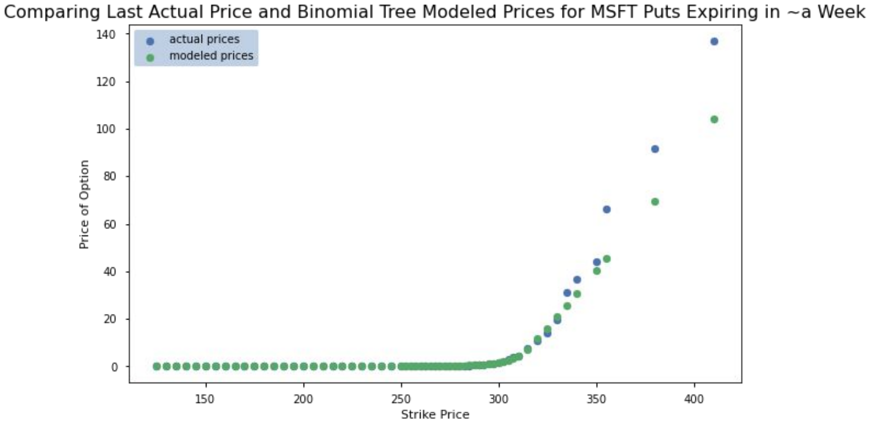


Results & Backtesting: Binomial Tree AM Calls

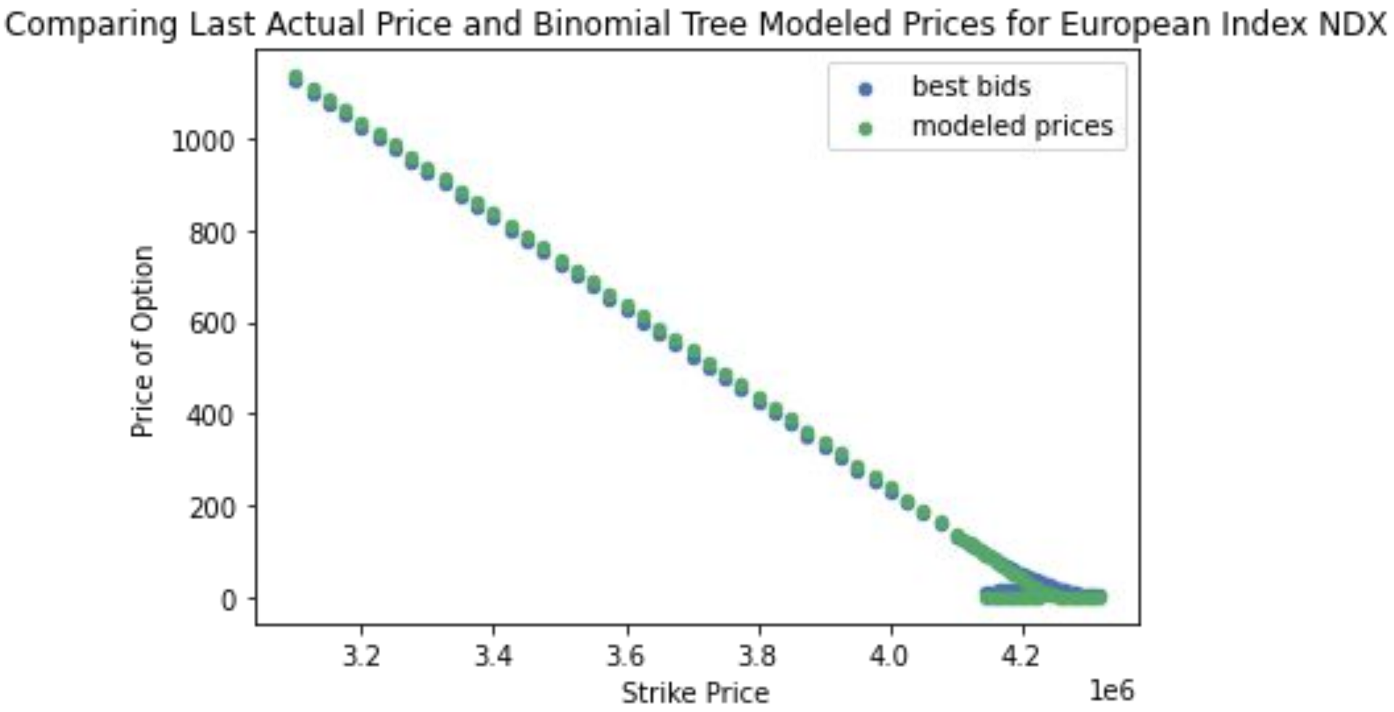




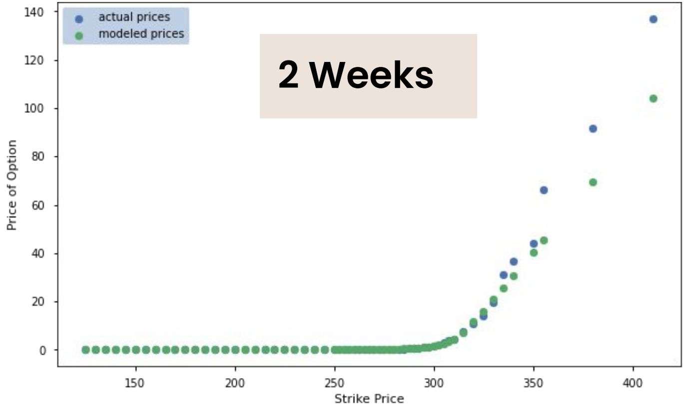


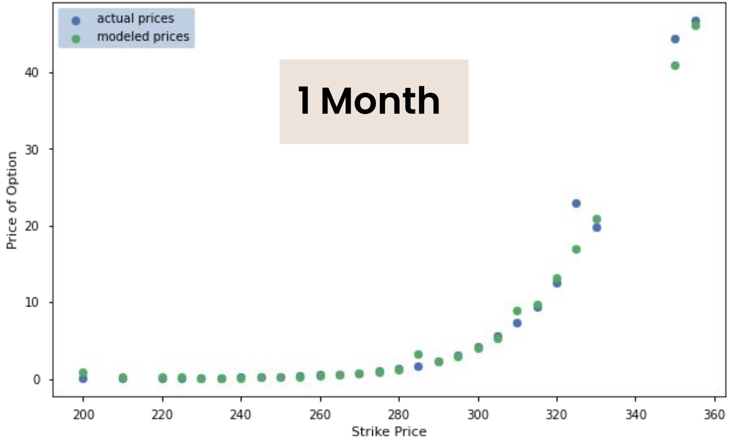


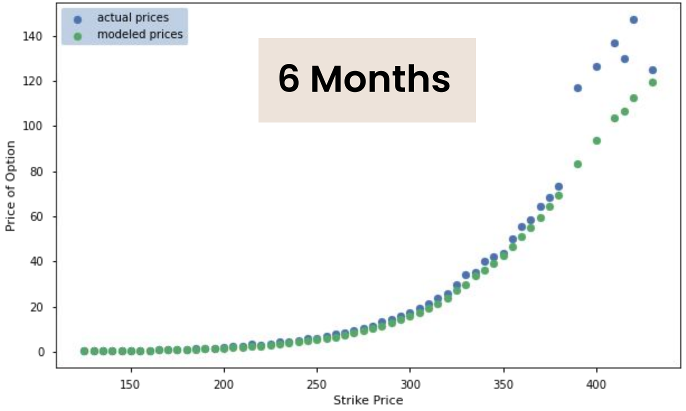
Results & Backtesting: Binomial Tree AM Puts

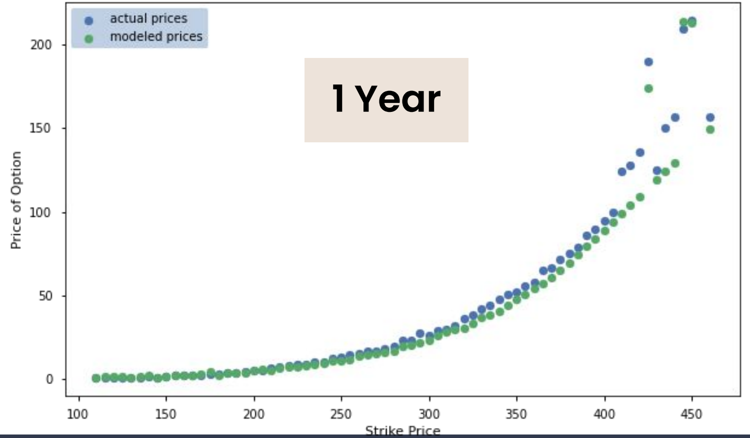


Results & Backtesting: Binomial Tree EU Calls

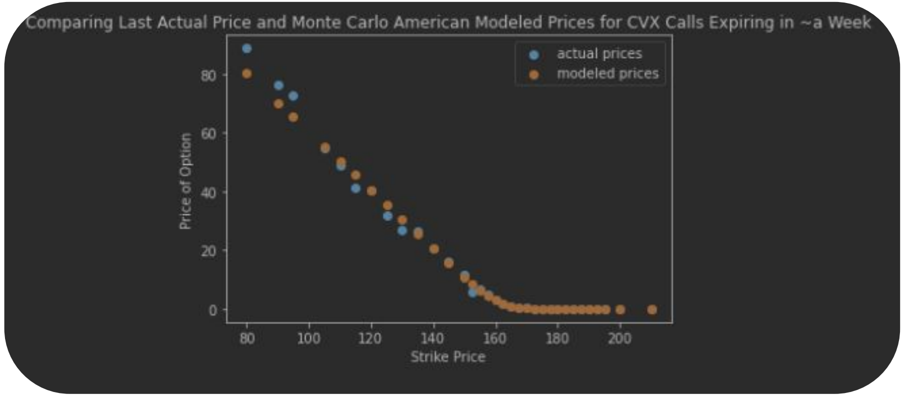


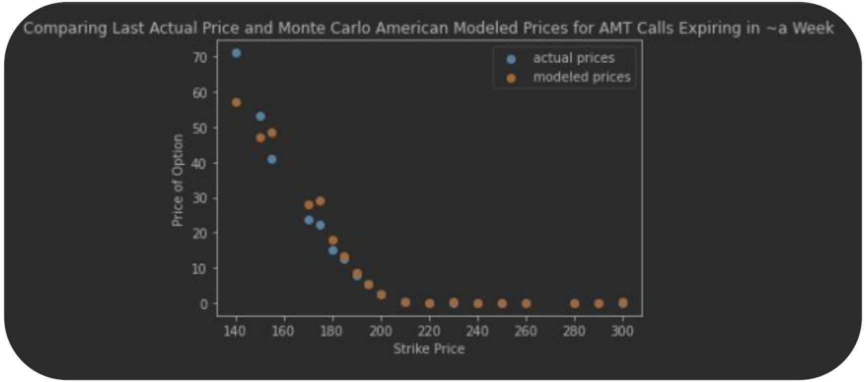




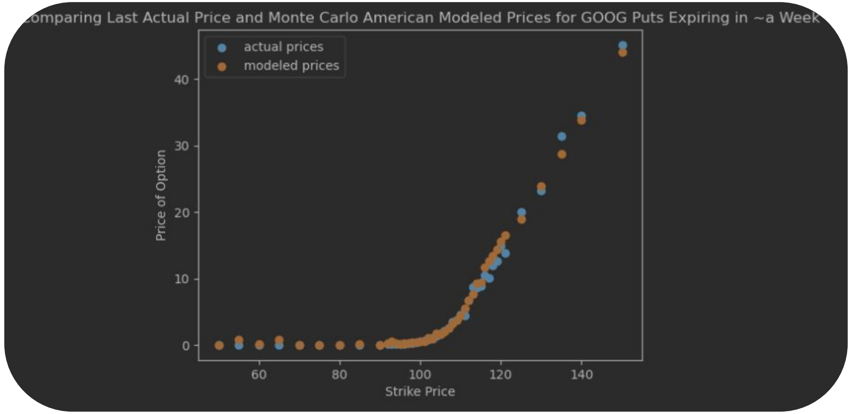


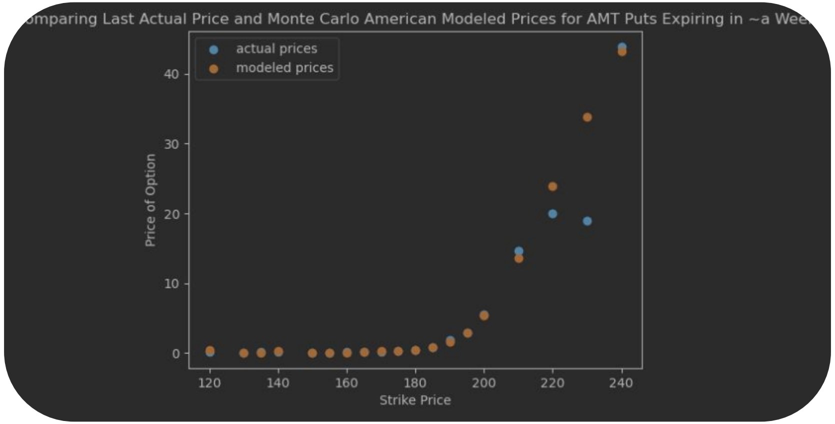
Results & Backtesting: Binomial Tree Key Findings

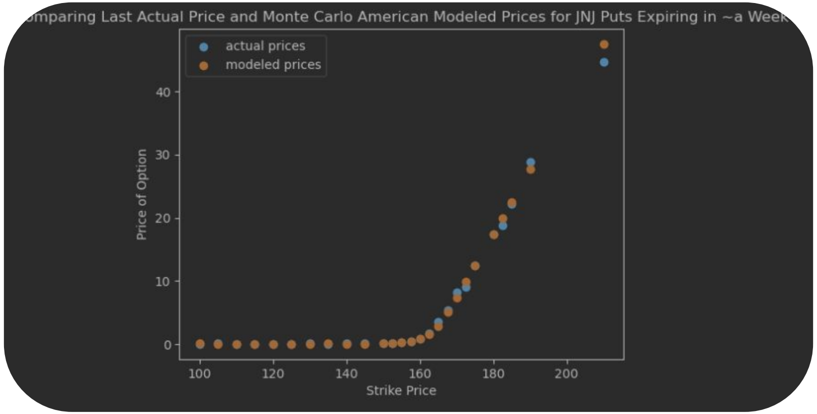


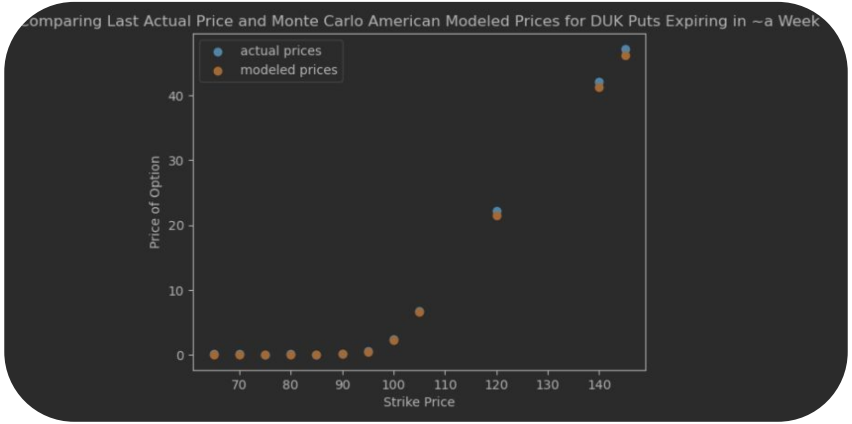


Results & Backtesting: Monte Carlo AM Calls

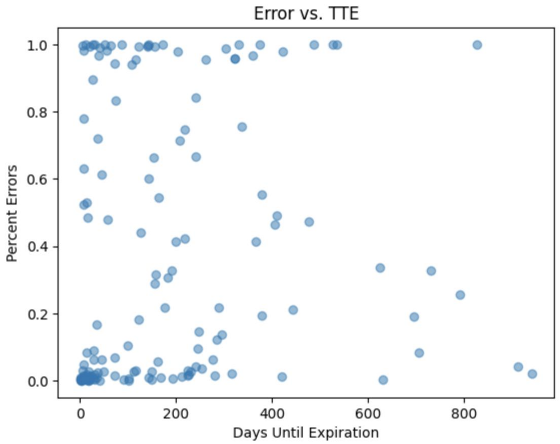


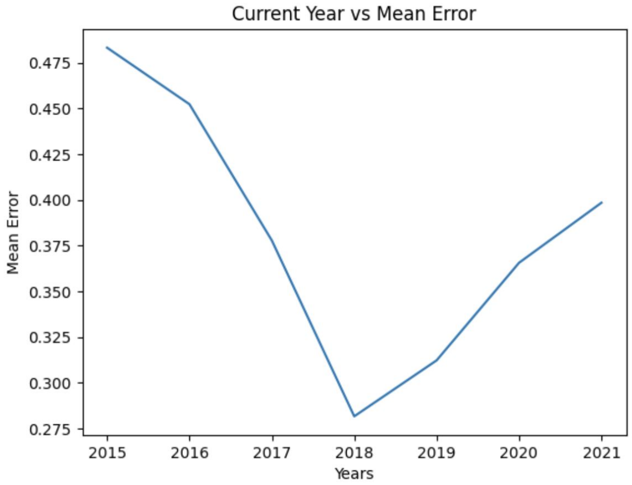


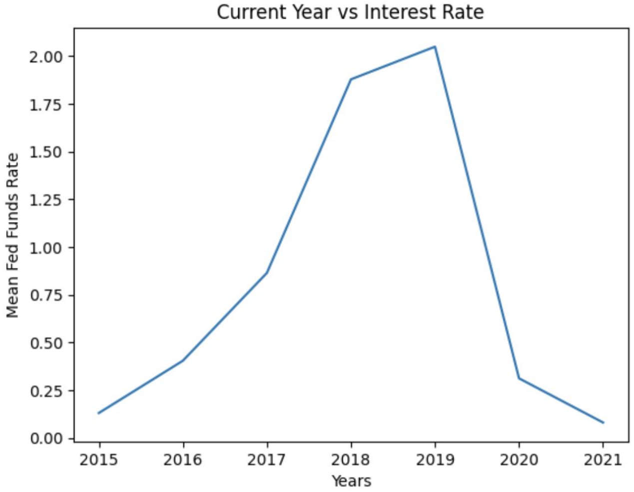


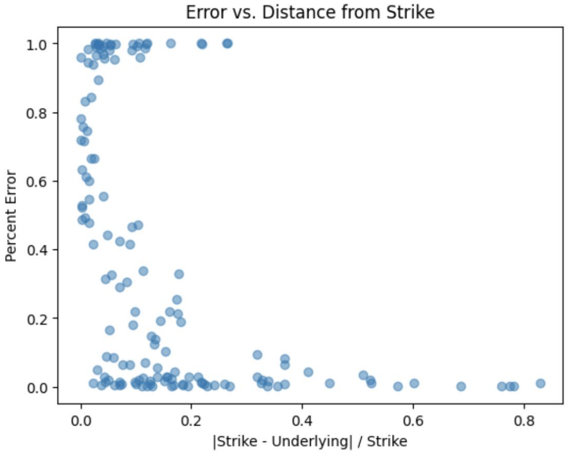


Results & Backtesting: Monte Carlo AM Puts









Interesting Data Points:

- **Mean error heavily skewed: 0.38 vs 0.18**
- **Calls did better than Puts**
  - **Means: 0.32 vs 0.46**
  - **Medians: 0.09 vs 0.33**
  - **Possibly due to differences in distribution**
  - **Model assumes constant volatility over entire lifespan (not what happens in reality**
- **Model seemed to do slightly better when farther from strike**
  - **Short DTE means average return can deviate from average**

Results & Backtesting: Monte Carlo EU

- Binomial model works really well for options with strike price around the price of the stock as well as those with strike prices out of the money
    - ~40% overall error but only 14% error when considering middle of the chain

- Black scholes is the convergence of the binomial tree model as the number of time steps approaches infinity.
    - However, it's more intensive and doesn't give values at each time step

- Monte Carlo had similar results to binomial tree (also about 40% error with 13-14% error with middle splicing)

- Of course, all of these implementations are based on assumptions that the market is efficient so there will be discrepancies between our models and the ones used in the real world. In fact, options are priced with a combination of these models in the real market.

- Next Steps:
    - Test our models on other types of options (exotics, compounding, etc.)
    - Backtest over longer periods (Would need optimizations, better hardware)
    - Learn stochastic modeling and introduce it to potentially increase accuracy

Overall Findings

Questions?