Course: MSc Finance

Institution: Dublin City University (DCU Business School)

Module: FBA1020 Financial Markets & Derivatives

Prepared by:

1. **Megha Sirohi - A00010701**
2. **Purva Dilip Choudhari - A00012455**
3. **Sai krishna Sithakari - A00045640**
4. **Shreya Mohanan Nair - A00043528**
5. **Udit Ganglani - A00045535**
6. **Vikas Gupta - A00047802**
7. **Vipul Baliyan - A00046346**

Table Of Contents

Introduction

Finding the correct price for derivatives stands as a basic backbone of risk management, investment strategy, and the general efficiency of the market in a much-volatile backdrop like the financial markets. Derivatives, for instance, the option, are financial transactions whose value depends upon the performance of an underlying asset-whether in stocks, commodities, currencies, or indices. The instruments provide much utility to the investor, as they can hedge possible losses, speculate on future prices, or combine investments into complex portfolios fit for certain risk-return profiles. The ability to derive the correct prices for these derivatives is by no means an academic exercise, but rather an indispensable skill that market players need to acquire, from the single trader to the largest institutional investor.

The things related to computers and programming nowadays concerning the finance domain have become more versatile, especially the derivatives pricing field. Among the wide variety of programming languages available today, more functions can be found in Python, as it has most libraries and simple syntax, making it the most utilized tool for financial modelling and analysis in recent times. Using it, one can easily implement very complicated algorithms, generate market scenarios, and analyse huge datasets. Then this project uses python to study option pricing specifically into the basics of derivative pricing, i.e., European call and put options, which represent an essential foundation in the whole derivatives arena.

It is the Binomial option pricing model at the core of this project and forms a critical pillar in financial theory. It has an uncomplicated yet strong framework to explain how option-prices are determined. This model involves a discretization of time and a modeling of possible price movements of the underlying asset as a series of up or down steps so that one can follow the process in a very lucid and intuitive understanding of the reasons as to why an option would attain a value. Thus, the simple model makes this an ideal start for those new to derivatives pricing and an adaptable foundation upon which a more detailed, real-world extension could be created.

The stock which is chosen as the major underlying asset of the project is that of Advanced Micro Devices (AMD), a devices-producing company that specializes in the production of high-performance computing and graphics technologies. The stock of AMD provides a fascinating case for investigation due to its volatility in market conditions and its occupation of industries that are undergoing fast advances in terms of technology, as well as stiff competition. Therefore, studying the different derivatives that relate to AMD's stock gives the reader a first-hand opportunity to understand how the views of the industry, accompanied by the trends in the company's processes and events, significantly influence the overall pricing of options.

This project extends beyond mere price calculation; it aims to instill a deeper understanding of financial principles, foster collaborative learning, and enhance practical skills. Through hands-on coding, participants will not only compute option prices but also verify crucial concepts such as call-put parity, a relationship that ensures consistency between call and put option values. Furthermore, the project emphasizes the importance of teamwork, knowledge sharing, and continuous improvement, empowering participants to become proficient in both financial theory and computational practice. By engaging with AMD's stock data, participants will have the chance to explore the impacts of real-world variables on options prices. The project is structured to enable group members to become proficient by leveraging collaborative learning and skill development through shared knowledge and insights throughout the project, with the main goal of enriching everyone's learning journey.

Objectives

- Have a thorough understanding of the Binomial Model - Study the mathematics behind the Binomial Model in-depth, including risk-neutral probabilities, time steps, and option valuation at nodes in the pricing tree.
- Apply Python in Financial Modelling – Do financial modelling with Python, using some of the most important libraries such as yahoo finance, NumPy, and math for quick computation of option prices.
- Integration of Market Data in Real-time- Fetch stock market data from Yahoo Finance in real- time for the accurate and relevant calculations that reflect current market conditions.
- Implement European Option Pricing - Derive the prices of an European call and put option via the Binomial Model, then observe their changes with different market scenarios.
- Call-Put Parity Verification – The next step is to verify the theoretical aspect of our arbitrage-
- free pricing by establishing the adherence of the computed call and put option prices to the call- put parity, validating our pricing method.
- Interpret Option Prices in 'Financial Terms'- This involves analysing the impact of various input parameters like volatility, interest rates, and time to maturity on the outcome of option prices.
- Enhance Financial and Technical Skills- Better understanding derivatives pricing, computational finance and programming in Python, which are passed to the domain of quantitative finance and risk management.
- Furnish Hands-on Insights to Traders and Investors- Evaluate how option pricing models prove to be useful for traders and investors in making decisions regarding, hedging, speculation, and risk management in the financial market.

Computation of the price of a European call or put option using the Binomial model

Step 1: Importing Libraries

```
import yfinance as yf
import pandas as pd
import numpy as np
import math
from scipy import stats
import matplotlib.pyplot as plt
```

We are importing libraries, to do the coding and necessary action.

Step 2: Fetching stock data of AMD

```
# Fetch AMD stock data
amd = yf.Ticker("AMD")
data = amd.history(period="5y")

# Check the columns in the data
print(data.columns)  # This will help you confirm the available columns

# Function to fetch the latest AMD stock price
def get_amd_stock_price():
    amd = yf.Ticker("AMD")
    data = amd.history(period="1d")  # Get the latest stock price
    return data["Close"].iloc[-1]  # Return the latest closing price
```

Now we are fetching 5 years of AMD stock-data using yahoo finance and will use it to analyse AMD's historical performance over the time period.

Step 3: Binomial Model for European Option Pricing

```
def binomial_option_price(S, K, T, r, g_u, g_d, option_type, n=100):
    """
    Computes the price of a European option using the Binomial Model.

    Parameters:
    - S: Current stock price.
    - K: Strike price.
    - T: Time to maturity (years).
    - r: Risk-free interest rate (continuously compounded).
    - g_u: Growth rate during an up move.
    - g_d: Growth rate during a down move.
    - option_type: "call" or "put".
    - n: Number of steps in the binomial model (default 100).

    Returns:
    - Computed price of the option.
    """
    dt = T / n  # Time step size
    u = 1 + g_u  # Upward factor
    d = 1 + g_d  # Downward factor
    p = (math.exp(r * dt) - d) / (u - d)  # Risk-neutral probability

    # Option value at maturity
    option_values = np.zeros(n + 1)
    for i in range(n + 1):
        stock_price = S * (u ** (n - i)) * (d ** i)
        if option_type == "call":
            option_values[i] = max(0, stock_price - K)
        elif option_type == "put":
            option_values[i] = max(0, K - stock_price)

    # Backward induction
    for j in range(n - 1, -1, -1):
        for i in range(j + 1):
            option_values[i] = math.exp(-r * dt) * (p * option_values[i] + (1 - p) * option_values[i + 1])
```

The function binomial_option_price uses the binomial model to determine the pricing of European call or put options. This begins with calculations of time step size (dt), upward and downward movement factors (u and d, respectively), and risk-neutral probability (p) based on the risk-free rate (r). At maturity, this option's value is determined by intrinsic value (which is simply the difference between stock price and strike K). Then backward induction is applied whereby the option values at maturity are recursively discounted back to present time, yielding the fair value of the option. Being risk-neutral, the value at each step is simply an average of the possible outcomes.

Step 4: Verifying Call-Put Parity for AMD Options using a Binomial Model

```
# Verify Call-Put Parity: Call Price - Put Price = Stock Price - PV(Strike Price)
def verify_call_put_parity(call_price, put_price, S, K, r, T):
    lhs = call_price - put_price  # Left-hand side of parity equation
    rhs = S - K * math.exp(-r * T)  # Right-hand side of parity equation
    return math.isclose(lhs, rhs, rel_tol=1e-4)

# Main Execution
if __name__ == "__main__":
    # Given Inputs
    S = get_amd_stock_price()  # Fetch latest AMD stock price
    K = 100  # Strike price
    T = 2  # Time to maturity in years
    r = 0.04  # 4% risk-free rate (continuous)
    g_u = upswing_rate / 100  # Convert upswing rate to decimal
    g_d = downturn_rate / 100  # Convert downturn rate to decimal

    # Compute Call and Put Prices
    call_price = binomial_option_price(S, K, T, r, g_u, g_d, "call")
    put_price = binomial_option_price(S, K, T, r, g_u, g_d, "put")

    # Verify Call-Put Parity
    parity_holds = verify_call_put_parity(call_price, put_price, S, K, r, T)

    # Print Results
    print(f"Latest AMD Stock Price: ${S:.2f}")
    print(f"European Call Option Price: ${call_price:.2f}")
    print(f"European Put Option Price: ${put_price:.2f}")
    print(f"Call-Put Parity Holds: {parity_holds}")


    # Create a DataFrame to store results
    df = pd.DataFrame({
        "Option Type": ["Call", "Put"],
        "Option Price": [call_price, put_price]
    })
```

Verifying of Call-Put Parity with regards to European options. Call-Put Parity states that the difference between the price of a European call option and that of a European put option should be equal to the stock price minus the present value of the strike price. First, there is price retrieval for stocks in which the script fetches the latest stock price of AMD using get_amd_stock_price().

Next is an option pricing calculation where the script calculates call and put option prices using a binomial model taking into account upswing rate (g_u) and downturn rate (g_d). Then there is Call-Put Parity Verification which checks whether the calculated option prices satisfy the Call-Put Parity equation using verify_call_put_parity().

Step 5: Printing the result

```
    # Print the results in tabular form
    print("\nOption Prices:")
    print(df)


# Print the calculated upswing and downturn growth rates

print(f"Upswing Growth Rate: {upswing_rate:.2f}%")
print(f"Downturn Growth Rate: {downturn_rate:.2f}%")
```

Output:

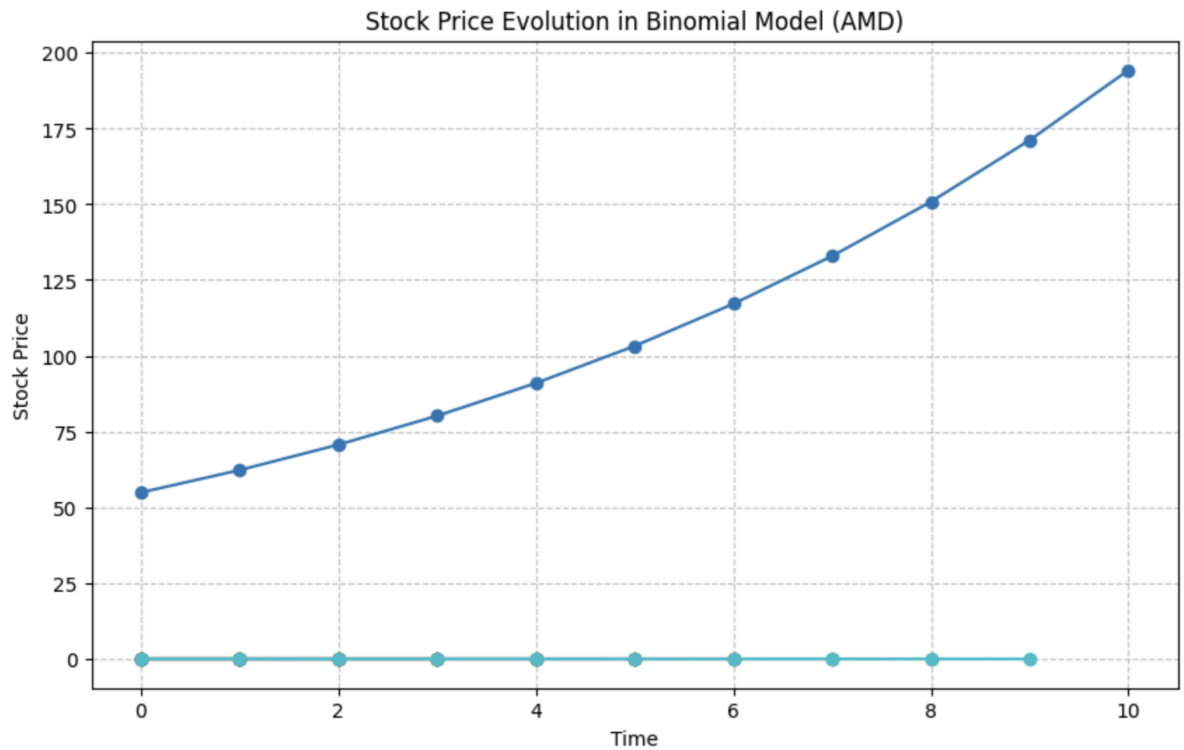
```
Index(['Open', 'High', 'Low', 'Close', 'Volume', 'Dividends', 'Stock Splits'], dtype='object')
Latest AMD Stock Price: $103.22
European Call Option Price: $99.87
European Put Option Price: $88.96
Call-Put Parity Holds: True

Option Prices:
  Option Type  Option Price
0        Call     99.870684
1         Put     88.962317
Upswing Growth Rate: 74.42%
Downturn Growth Rate: -27.65%
```

Outputs and Data Representation, where the stock price, computed option prices and parity validation result are displayed using pandas Dataframe. Displayed growth rates which were upswing and downturn growth rates.

Step 6: Visual representation of Stock Price Evolution in Binomial Model of AMD

```
def plot_stock_tree(stock_tree, N):
    plt.figure(figsize=(10, 6))
    for i in range(N+1):
        plt.plot(range(i+1), stock_tree[:i+1, i], marker='o', linestyle='-')
    plt.title('Stock Price Evolution in Binomial Model (AMD)')
    plt.xlabel('Time')
    plt.ylabel('Stock Price')
    plt.grid(True, linestyle='--', alpha=0.7)
    plt.show()

def main():
    S0 = 103.22  # Latest AMD stock price
    K = 100      # Strike price
    T = 1        # Time to maturity (in years)
    r = 0.05     # Risk-free rate
    sigma = 0.2  # Volatility
    N = 10       # Number of time steps

    dt = T / N
    u = np.exp(sigma * np.sqrt(dt))
    d = 1 / u

    stock_tree = np.zeros((N+1, N+1))
    for j in range(N+1):
        stock_tree[j, N] = S0 * (u ** j) * (d ** (N - j))

    plot_stock_tree(stock_tree, N)

if __name__ == "__main__":
    main()
```

Output:



This visualized how the price of the AMD stock evolved using the binomial tree method. The first thing that is done herein is to specify some key financial parameters, including and not limited to the initial stock price S_0 = 103.22 dollars, strike price K = 100 dollars, time to maturity T = 1 year, risk-free rate of return r = 5%, and volatility of stock prices σ = 20%. The time horizon is divided as N = 10 discrete steps. The above script calculates movement factors upward (u) and downward (d) according to volatility and time step (dt = T/N).

The code generates a binomial stock price tree, with each node within its corresponding price outcome at a particular time period. It then uses a nested loop to actually get stock prices at distinct times according to the binomial model method. Once calculated, all this information on stock price evolution is plotted using matplotlib, with which a marker is present at each step to represent possible stock price paths over time. The chart will contain a grid with a number of dashed lines and have its x-axis denote time and y-axis price. Thus, we get a clear visualization of stock prices as they would evolve in time under the binomial model.

Step 7: Calculating the Annual Returns Statistics

```
# Get the upswing and downturn growth rates
upswing_rate = annual_returns[annual_returns > 0].mean()  # Mean of positive returns (Upswing)
downturn_rate = annual_returns[annual_returns < 0].mean()  # Mean of negative returns (Downturn)

# Calculate mean, median, mode, standard deviation, and variance of returns
mean_return = annual_returns.mean()  # Mean
median_return = annual_returns.median()  # Median

# Calculate the mode
mode_result = stats.mode(annual_returns)

# If there's more than one mode, mode_result.mode is an array
# If it's a scalar, we simply get the first mode
mode_return = mode_result.mode[0] if isinstance(mode_result.mode, np.ndarray) else mode_result.mode

std_dev_return = annual_returns.std()  # Standard Deviation
variance_return = annual_returns.var()  # Variance

# Print statistical results
print(f"\nAnnual Returns Statistics:")
print(f"Mean: {mean_return:.2f}%")
print(f"Median: {median_return:.2f}%")
print(f"Mode: {mode_return if mode_return is not None else 'No distinct mode'}")
print(f"Standard Deviation: {std_dev_return:.2f}%")
print(f"Variance: {variance_return:.2f}%")
```

Output:

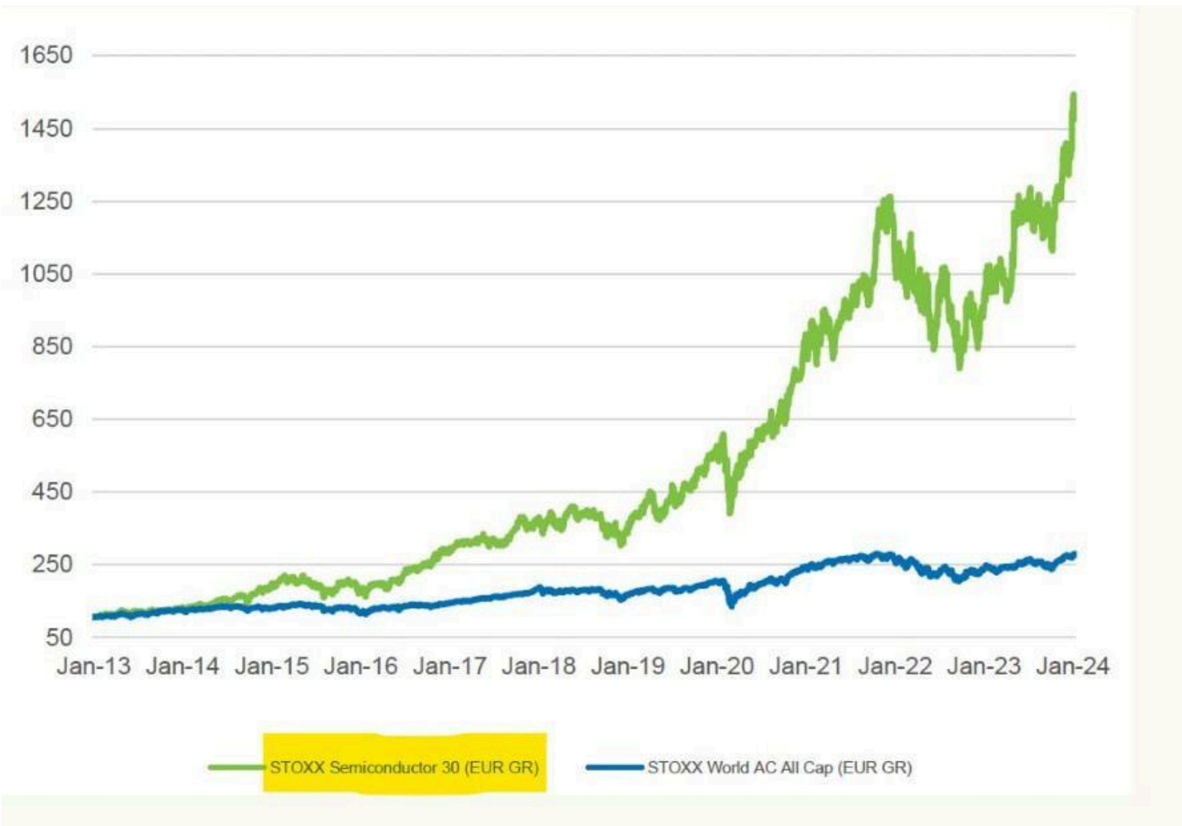
```
Annual Returns Statistics:
Mean: 23.39%
Median: 22.81%
Mode: -61.119021011757226
Standard Deviation: 60.14%
Variance: 3617.11%
```

The script analyses yearly stock returns calculating the upswing growth rate and the downturn growth rate, representing average positive returns and negative returns, respectively. The basic statistics such as the mean, median, mode, standard deviation and variance, interpretation of return distribution and volatility is included. Insights from the results are market trends, risk evaluation, and investment decision-making, highlighting the average performance, consistency of returns, and variations in stock returns.

Key Industrial Trends

The derivatives market for the semiconductor industry is transforming extremely fast because of some decent, high-impact factors like advancements in technology, positive or negative investor sentiment, liquidity shifts, and increasing regulatory changes across the market. AMD increases its market share in CPUs and GPUs, especially in applications cantered on AI and data centres, traders and investors are increasingly looking at derivatives, like hedge funds, as a more innovative and newer approach towards risk management and capturing opportunities in semiconductor research. Some emerging industry trends that influence the derivatives market are as follows:

1. Rising Liquidity Requirements and Regulatory



Liquidity is still the key factor in the derivatives market where investors prefer very liquid options and futures that feature a very thin spread and lower cost of trading. At the same time, changes in the regulation concerning clearing, margin requirements, and transparency are causing a redesign in trading strategies. Force this changes have put into the firms to invention in terrible instances by creating new structured products and trade mechanisms that provide a cost-effective solution.

2. Advanced Semiconductor Packaging:

The movement towards advanced packaging technologies, such as chip-on-wafer-on-substrate (CoWoS) and fan-out panel-level packaging (FOPLP), the semiconductor industry has a history of such technological advancements that are meant to overcome advanced chip design limits. The move by AMD in considering these developments is good for product acceptance and investor sentiment and derivative pricing. With growing competition in AI and high-performance computing, innovation in chip design will prove to be extremely critical to the market valuation of AMD.

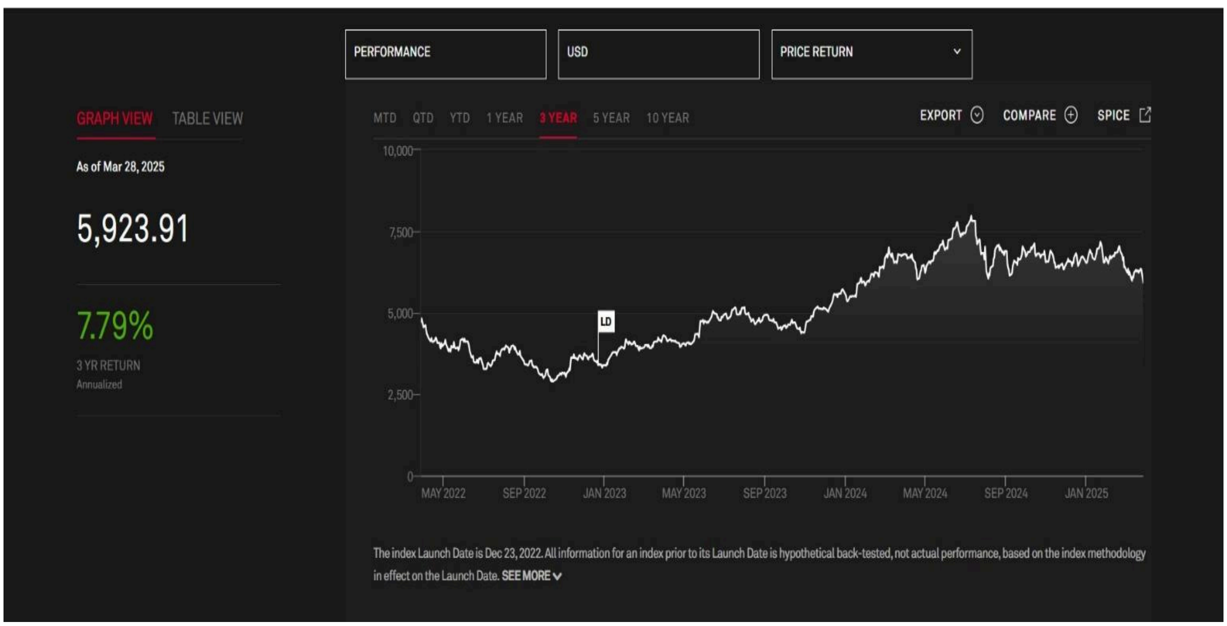
3. Brokerage CFDs and More Leverage Trading:

Contracts for Difference (CFDs) are growing in popularity among traders speculating in AI-related stocks without actually owning the stock. CFDs provide leverage for investors to control larger positions with a small initial investment. On the downside, because of the high volatility of the semiconductor stocks, trading with leverage comes with high levels of risks. Stock prices fluctuate based on market sentiment, policies, regulatory framework, and supply chain. Thus, CFDs are an opportunity for some traders and a challenge for others.

4. Global Semiconductor Market will dominate even double digits in growth rates:

It is expected that the semiconductor industry will grow by 15% in 2025, certain areas, like high-bandwidth memory (HBM), will witness even higher growths. This expansion is sky-high, mainly attributed to AI Chips, high-performance processors, and 5G infrastructure.

S&P 500 Semiconductor ETF as of March 2025



From various opportunities for AMD's growth, while the company competes with Nvidia and Intel in the AI and data centre market, the firm would most likely follow the stock trend of the general semiconductor market. That would reflect and impose the pricing and demand for other derivatives such as options and futures contracts. Furthermore, while growth expectations are bright, geopolitical conflicts and tensions between China and Taiwan can also bring about some risks to the industry, including supply chain disruptions and regulatory challenges. These uncertainties must be put into consideration by whatever derivative investor when thinking of potential trading strategies.

To sum up, we see rapid evolution in the derivatives market, fuelled by liquidity transitions, technology-driven shifts, and growing investor appetite for semiconductor stocks. Major points of consideration include:

- **Liquidity and regulation** are changing how derivatives are traded.
- **Index-based products and ETFs** are gaining more popularity as an investment tool.
- **The blend of exchange-traded and OTC markets** is enhancing price efficiently.
- **AI-driven semiconductor demand** is creating volatility and opportunities in the derivatives market.
- **Advanced packaging technologies** as discussed above are reshaping AMD's competitive position.
- **Global semiconductor growth** is driving investor interest but comes with geopolitical risk challenges.
- **CFD instruments provide good leverage to investors but also increase exposure to risk.**

As the semiconductor industry continues to progress, AMD remains in the middle of major market shifts. Whether through direct investment in stocks, options trading, or futures contracts, investors will need to stay informed about these trends to steer the dynamic derivatives market effectively.

Conclusion

To carry out a fuller evaluation of the Advanced Micro Devices (AMD) with respect to its stock performance and derivative nature, some financial models like the Binomial Model for pricing European options and the Call-Put Parity checks have also been utilized.

- The present study indicates certain industrial trends in respect to AMD's historical stock price trajectory.
- Computes annual return statistics in terms of increasing liquidity factors and regulatory regimes for derivatives trading, increasing reliance on index-based derivatives and ETFs for risk hedging.
- Market volatility thanks to the upswing in AI-backed semiconductor demand, giving further impetus to trading opportunities.
- In high-performance computing, AMD's strategic positioning matters in derivatives price modelling because of the forthcoming role the company will play in the burgeoning data centre and AI chip markets.

Feedback

The findings also address the subject of risk against reward with regard to leverage-traded products, such as CFDs, besides a matter-of-factly reading that forecasts the semiconductor industry to show double-digit growth in an era besmirched by political risks and supply chain vulnerabilities. The proper structuring of the framework, typesetting of models, good graphics of the data, and well-researched insights into the industry make the study sustainable, while numerous opportunities exist to go further, including real-life case introductions for validation of theoretical models, sensitivity analyses to see the impacts of various risk factors, and a peer performance index against NVIDIA or Intel to provide context to AMD's performance. AI-enabled supply chain solutions with the entire new facet of strategic partnerships will play an even more significant role if these partnerships withstand the test of demand volatility and help the semiconductor player remain more competitive.

Individual Report

Megha Sirohi

What I Learned:

- Collecting and preprocessing stock market data with Python.
- Techniques of statistical analysis over historical stocks.
- The role of risk-free interest rate and stock volatility in the option pricing .
- Estimation of options′ fair value with the Binomial Model.
- Implementation of numerical methods using ‘numpy’ and ‘pandas’.

What I Taught My Peers:

- Developed data preprocessing steps with Vikas.
- Explained trends of stock price movement to Purva and Shreya.
- Discussed the significance of historical data in financial modeling with Vipul.
- Assisted Sai Krishna in trend analysis for drawing some conclusions.
- Provided insights to Udit on how structural changes in industries affect stock price trends.

Purva Dilip Choudhari

What I Learned:

- Implementation of Binomial Option Pricing Model using Python.
- How option prices depend on input parameters.
- Logic behind Call-Put Parity and its verification by coding
- Visual representation of financial models using Python.
- Debugging and improving Python scripts for better accuracy.

What I Taught My Peers:

- Data pre-processing steps are made better for Megha and Vikas.
- Helped Vipul understand putting the theoretical frame into Python.
- Explained the script structure and logic to Shreya.
- Discussed techniques of interpretation of result with Udit.
- Worked with Sai Krishna on summarizing the role of Python in pricing derivatives.

Sai Krishna Sithakari

What I Learned:

- The application of financial models in trading.
- The importance of risk assessment while trading in derivatives.
- The validity of a pricing model by Call-Put Parity.
- How industry trends affect financial strategies.
- The usefulness of peer collaboration in financial research.

What I Taught My Peers:

- Presented the highlights to the entire group.
- Discussed the risks with Udit.
- Explained to Vipul how the different parts of the project correlated.

- Helped Purva and Shreya improve on their discussion of results.
- Provided a structured summary of their findings in data analysis to Vikas and Megha.

Shreya Mohanan Nair

What I Learned:

The importance of binomial tree models in derivative pricing.
- Using Python for validating Call-Put Parity.
- The mathematical reasoning behind risk-neutral probabilities.
- Importance of multiple test cases in verifying results.
- Improving the efficiency and readability of a script.

What I Taught My Peers:
- Worked with Purva on implementing the binomial pricing model.
- Explained the programming logic behind the model to Vipul.
- Assisted Megha and Vikas in structuring data inputs for the script.
- Helped Udit understand how financial models are coded.
- Worked with Sai Krishna in finalising the output presentation.

Udit Ganglani

What I Learned:
- Pricing of derivatives and economic trends.
- Liquidity and regulatory policies in financial markets.
- Informing how stock market movements will have their effect on the pricing strategy for options.
- The growing importance of index-based derivatives and exchange-traded funds.
- How financial modeling is related to actual industry applications.

What I Taught My Peers:
- Shared insights to Vipul and Megha industry trends.
- Discussed the impact of economic changes on stock prices with Vikas.
- Explained market shifts and their relevance to financial models with Purva and Shreya
- Provided real-world examples to Sai Krishna for his conclusion.

Vikas Gupta

What I Learned:
- The importance of historical stock data in financial decisions.
- Implementation and interpretation of the binomial pricing model.
- Effect of risk-free rate and stock price changes on option prices.
- Practical applications of quantitative finance to real worldl markets.
- Techniques for verifying Call-Put Parity using Python.

What I Taught My Peers:
- Worked with Megha to collect and analyze the financial data.
- Explained stock market trends to Purva and Shreya.
- Assisted Vipul in connecting theory with results derived from data.
- Discussed financial relevance of binomial model findings with Udit.
- Helped Sai Krishna in interpreting statistical results towards the conclusion.

Vipul Baliyan
What I Learned:
- Underlining the fundamentals involved in derivatives pricing and their importance in financial markets.
- The role of Python in implementing quantitative financial models.
- The Binomial Model for European option pricing and its foundations.
- The role of Call-Put Parity in validating option pricing results.
- How industry professionals use computational models to get a fair estimate of option prices.

What I Taught My Peers:
- Clear articulation of the objectives of the project and its relevance to the team.
- Provided Megha and Vikas insights on why data analysis is an important aspect of financial modelling.
- Made Purva and Shreya understand the theoretical aspects of the Binomial Model.
- Helped Udit relate industry trends to real-world practice in derivatives pricing.
- Shared the important learnings with Sai Krishna for his conclusion part.

Final Note: This project has been a very good learning experience, where the group interacted well and had practical exposure in derivative pricing using Python. Each of the group members contributed highly to the accuracy and the organization of a script, thereby enabling the building of their own knowledge in finance and programming.

References

Talking Points: Navigating the Evolving Derivatives Market, Trends, Innovation and the Rise of Index-Based Solutions, 26th February 2025. https://www.spglobal.com/spdji/en/education/article/talkingpoints-navigating-the-evolving-derivatives-market-trends-innovation-and-the-rise-of-index-based-solutions

Eurex introduces futures on STOXX Semiconductor 30 index, 18th March 2025. https://stoxx.com/eurex-introduces-futures-on-stoxx-semiconductor-30-index/

Is AMD Stock A Buy Before Chipmaker's First-Quarter Report?, 26th March 2025. https://www.investors.com/news/technology/amd-stock-buy-now/

The Future of Semiconductors in 2025: Technology Trends and Industry Outlook, 1st January 2025. http://randtech.com/the-future-of-semiconductors-in-2025-technology-trends-and-industry-outlook/?utm

Trading Semiconductor Stocks: All You Need to Know, 19th Feb 2025. https://www.plus500.com/en-dk/newsandmarketinsights/trading-semiconductor-stocks-all-you-need-to-know

IDC: Global Semiconductor Market to Grow by 15% in 2025, Driven by AI, 2024. https://www.idc.com/getdoc.jsp?containerId=prAP52837624&utm

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

Cox, J.C., Ross, S.A. and Rubinstein, M., 1979. Option pricing: A simplified approach. Journal of Financial Economics, 7(3), pp.229–263.

Hull, J.C., 2018. Options, Futures, and Other Derivatives. 10th ed. Harlow: Pearson Education.

Yahoo Finance, n.d. Advanced Micro Devices, Inc. (AMD) stock data. [online] Available at: https://finance.yahoo.com/quote/AMD/

Yfinance Developers, 2023. yfinance: Yahoo! Finance market data downloader. [online] Available at: https://github.com/ranaroussi/yfinance

Google Finance (n.d.) Advanced Micro Devices, Inc. (AMD) stock overview. Available at: https://g.co/finance/AMD:NASDAQ

Google Finance (n.d.) Advanced Micro Devices, Inc. (AMD) 5-year stock performance. Available at: https://g.co/finance/AMD:NASDAQ?window=5Y