## **Introduction**

In today's rapidly evolving financial markets, leveraging advanced computational tools can significantly enhance trading efficiency and accuracy. This project is dedicated to developing, optimizing, and testing two distinct investment strategies using Python. The first strategy is based on a traditional technical analysis indicator, the Exponential Moving Average (EMA) crossover (EMA12 and EMA26), which signals buying opportunities when the short-term EMA crosses above the long-term EMA and selling points when it crosses below. The second strategy employs sophisticated machine learning techniques, specifically Random Forest and XGBoost models, utilizing a set of technical indicators including SMA10, SMA60, EMA10, Momentum, and RSI to predict stock price movements.

Our analysis focuses on NVIDIA's daily stock prices sourced from Yahoo Finance, spanning from January 2018 to January 2024. The dataset will be split into 20% for in-sample testing and 80% for out-of-sample validation to ensure the robustness and effectiveness of our trading signals. This project not only aims to validate the effectiveness of these strategies through backtesting but also seeks to optimize them to achieve the highest risk-adjusted returns possible.

By combining traditional trading strategies with modern machine learning techniques, this project explores the potential to create a more dynamic and profitable trading system that adapts to market changes and complexities.

## Dataset Description

The dataset used in this project is sourced from Yahoo Finance and consists of daily stock prices for NVIDIA Corporation (Ticker: NVDA), covering a comprehensive period from January 2018 to January 2024. This dataset includes the following key financial metrics:

* **Open:** The price at which the stock first traded upon the opening of an exchange on a trading day.
* **Close:** The price at which the stock last traded upon the closing of an exchange on a trading day.
* **High:** The highest price at which the stock traded during the trading day.
* **Low:** The lowest price at which the stock traded during the trading day.
* **Volume:** The number of shares that changed hands during a given day.

**Time Span and Data Split:**

* The data spans over six years, providing a robust timeframe to analyze the performance of the trading strategies under various market conditions.
* The dataset is divided into two segments: 20% for in-sample data used for initial testing and training, and 80% designated as out-of-sample data used for validating the trading strategies. This split ensures that the strategies are tested on unseen data, simulating real-world trading scenarios and assessing the generalizability of the models.

**Data Integrity and Preprocessing:**

* Prior to analysis, the data underwent preprocessing to ensure its quality and reliability. This included checking for missing values, anomalies, or inconsistent entries, and making necessary adjustments to prepare the data for effective analysis.
* Given the importance of accurate and timely data in trading, special attention was given to aligning date stamps and ensuring that all data points were correctly ordered and synchronized.

**Utilization in Strategies:**

* The dataset's various metrics are utilized differently in the two trading strategies developed in this project. For the EMA crossover strategy, the 'Close' prices are particularly crucial as they are used to calculate the EMAs which form the basis of the trading signals.
* In the machine learning models, multiple features derived from the raw data, such as moving averages (SMA and EMA), momentum indicators, and relative strength index (RSI), are used to feed into the Random Forest and XGBoost algorithms for predicting future price movements.

## Methodology

The project employs two distinct investment strategies to analyze and predict the stock price movements of NVIDIA, utilizing Python for implementation. The following subsections outline the development, optimization, and evaluation methodologies for each strategy.

**1. EMA12 and EMA26 Crossover Strategy**

**Technical Setup:**

* This strategy uses two exponential moving averages (EMAs): a short-term EMA of 12 days (EMA12) and a long-term EMA of 26 days (EMA26).
* **Buy Signal:** Generated when EMA12 crosses above EMA26, indicating a potential upward price momentum.
* **Sell Signal:** Generated when EMA12 crosses below EMA26, suggesting a potential downward price trend.

**Implementation Details:**

* The EMAs are calculated using the closing prices of NVIDIA stocks. The formula for an EMA is:
* Python’s pandas library is utilized to handle data manipulation and calculation of EMAs. The crossover points are identified through logical conditions applied to the EMA time series.

**Optimization and Testing:**

* The strategy is first tested on in-sample data (20% of the dataset) to adjust parameters such as the length of EMAs if necessary.
* Comprehensive backtesting is then conducted on the out-of-sample data (80%) to validate the strategy’s effectiveness across different market phases.

**2. Machine Learning-Based Trading Using Random Forest and XGBoost**

**Feature Selection:**

* Multiple features are extracted from the daily stock data, including simple moving averages (SMA10, SMA60), additional EMAs (EMA10), momentum indicators, and the Relative Strength Index (RSI).
* These features are chosen based on their historical performance in capturing market trends and their potential to provide predictive insights into price movements.

**Model Development:**

* Two machine learning models are deployed:
  + **Random Forest:** A robust ensemble learning method that uses a multitude of decision trees to make predictions and improve accuracy.
  + **XGBoost:** An implementation of gradient-boosted decision trees designed for speed and performance.
* Both models are implemented using Python’s sklearn and xgboost libraries.

**Training and Validation:**

* The models are trained using the in-sample data with a focus on minimizing overfitting through techniques such as cross-validation and hyperparameter tuning using grid search.
* Predictions are then made on the out-of-sample data to assess the models' generalizability and effectiveness in real-world scenarios.

**Performance Evaluation:**

* The strategies are evaluated based on standard financial metrics such as the Sharpe Ratio and Total Returns, calculated for both the training and testing datasets.
* Comparisons are drawn between the two strategies to determine their respective strengths and suitability under varying market conditions.

**Optimization:**

* Both strategies undergo a phase of optimization based on their performance metrics. This includes tuning the lengths of EMAs in Strategy 1 and adjusting the hyperparameters in the machine learning models for Strategy 2.