EMA Crossover Strategy Profitability Prediction

Problem Statement:

In financial markets, technical analysis strategies are often used to generate buy/sell signals, in this project we'll be looking at "Golden Crosses" which are generated when the EMA50 signal line crosses over the EMA200 signal line. However, not every EMA crossover results in a profitable trade. This project addresses the real world challenge of enhancing the traditional EMA crossover by building a machine learning model that predicts the success probability of a crossover event.

Investors and traders currently lack a systematic, predictive way to evaluate the quality of a crossover before acting on it. Our model aims to fill this gap by providing a success probability at each crossover event, allowing for better informed trading decisions and reduced risk.

Financial Terms:

Prior to getting into the coding process, it's key to understand several financial terms that will be used, I will provide an overview of each term below.

For stock trading:

- **Ticker:** A unique symbol used to identify a specific stock or security on a stock exchange.
- **Open:** The price at which a stock begins trading at the start of a trading session.
- **High:** The highest price a stock reaches during a specific trading period, indicating the peak value.
- **Low:** The lowest price a stock drops to during a specific trading period, showing the minimum value.
- **Close:** The final price at which a stock trades at the end of a trading session or period.
- **Volume:** The total number of shares or contracts traded for a stock during a specific period, reflecting the level of market activity or interest.

Technical Indicators:

- Commodity Channel Index (CCI):
  - Measures how far a price deviates from its average (typical price) over a set period (usually 20 days).
  - Good at identifying mean-reversion points and overbought/oversold conditions.
  - CCI compares the current typical price (average of high, low, and close) to its moving average, adjusting for volatility. A CCI value above +100 often signals overbought conditions, while a value below -100 indicates oversold conditions.

    $$CCI = \frac{Typical\ Price - MA\ of\ Typical\ Price}{0.015\ \times \text{Mean Absolute Deviation}}$$

- MACD Line and MACD Signal Line:
  - Measures momentum changes via moving averages (trend + momentum).
  - Tracks changes in the strength, direction, and duration of a stock's trend by analyzing the relationship between two exponential moving averages.
  - The MACD Line represents the difference between a short-term EMA (typically 12 day) and a longer-term EMA (typically 26 day), while the Signal Line is a 9-day EMA of the MACD Line.

    $$MACD\ Line = EMA_{12} - EMA_{26}$$
    $$Signal\ Line = EMA_{9}\ of\ MACD\ Line$$
    $$Histogram = MACD\ Line - Signal\ Line$$

- Relative Strength Index (RSI):
  - Momentum oscillator that measures the speed and change of price movements to identify overbought/oversold conditions (14-day typical period).
  - Ranging from 0 to 100, with values above 70 indicating overbought conditions (sell signal) and below 30 suggesting oversold conditions (buy signal).

    $$RSI = 100 - \frac{100}{1 + RS}$$

- Exponential Moving Average (EMA):
  - Type of moving average that gives more weight to recent prices, making it more responsive to new information compared to a simple moving average

- Used to determine the direction and strength of a trend, smoothing out price fluctuations to help traders identify potential support/resistance levels or trend reversals.
  $EMA_t = \alpha \times Price_t + (1 - \alpha) \times EMA_{t-1}$

- Average True Range (ATR):
  - Measures market volatility by calculating the average range of price movements over a specified period (typically 14 days).
  - ATR considers the true range (the greatest of the current high minus low, absolute value of the current high minus the previous close, or absolute value of the current low minus the previous close) and averages it over time. Higher ATR values indicate greater volatility, while lower values suggest a calmer market.
    $ATR_t = \frac{(Previous\ ATR \times (n-1) + True\ Range_t)}{n}$

- Hull Moving Average (HMA):
  - Is a fast and smooth moving average designed to reduce lag while maintaining trend direction accuracy; useful in detecting trend changes
  - HMA uses weighted moving averages of different periods, applying a square root adjustment to the period length to minimize lag.
    $HMA_n = WMA\left(2 \times WMA_n(Price) - WMA_n(Price), \sqrt{n}\right)$
    Where: $WMA_n = Weighted\ Moving\ Average\ over\ n\ periods$

- Bollinger Bands:
  - Consist of a middle band (typically a 20-day SMA) and two outer bands (standard deviations above and below the middle band, usually 2 standard deviations).
  - Used to measure volatility and identify overbought or oversold conditions, as well as potential breakout points.
    $Middle\ Band = SMA_n(Close)$
    $Upper\ Band = Middle\ Band + (k \times Standard\ Deviation_n(Close))$
    $Lower\ Band = Middle\ Band - (k \times Standard\ Deviation_n(Close))$

Dataset:

The data was sourced by first scraping S&P 500 ticker symbols from *https://www.slickcharts.com/sp500* using `*requests*` and `*BeautifulSoup*`, with tickers adjusted for compatibility. Historical price data for these tickers, spanning January 1, 2010, to April 22, 2025, was then downloaded from Yahoo Finance via the `*yfinance*` library, with adjustments for splits and dividends, resulting in a multi-indexed data frame for analysis containing:

- Date, Ticker, Open, High, Low, Close, Volume

Dataset Preprocessing: Simply transformed the multi-indexed data frame into a singular column data frame.

Feature Engineering:

Engineered 21 technical features using common indicators

- EMA10: The 10- day Exponential Moving Average (EMA10)
- EMA20: The 20-day Exponential Moving Average (EMA20)
- EMA50: The 50-day Exponential Moving Average (EMA50)
- EMA200: The 200-day Exponential Moving Average (EMA200)
- momentum_5: The 5-day momentum measures the percentage change in the closing price over the past five days
- ema_distance: The EMA distance represents the difference between the 50-day and 200-day EMAs
- ema50_slope: The 5-day slope of the EMA50 calculates the change in the 50-day EMA over five days
- price_ema200_dist: The price to EMA200 distance measures the difference between the current closing price and the 200day EMA, indicating how far the price deviates from its long-term average
- EMA50_prev: The previous day's 50-day EMA, aids in detecting crossovers
- EMA200_prev: The previous day's 200-day EMA, aids in detecting crossovers
- RSI_14: The 14-day Relative Strength Index evaluates momentum by comparing the magnitude of recent gains to losses, identifying overbought/oversold conditions
- MACD: The Moving Average Convergence Divergence line tracks momentum by calculating the difference between the 12-day and 26-day EMAs
- MACD_signal: The MACD signal line is a 9-day EMA of the MACD

- macd_diff: The MACD difference measures the gap between the MACD and its signal line
- ATR_14: The 14-day Average True Range quantifies market volatility by averaging the true range of price movements over 14 days
- CCI_20: The 20-day Commodity Channel Index assesses how far the price deviates from its average
- BB_MID: The Bollinger Bands middle line is a 20-day simple moving average of the closing price
- BB_STD: The 20-day standard deviation measures the volatility of the closing price
- BB_UPPER: The upper Bollinger Band is calculated as the middle band plus two standard deviations
- BB_LOWER: The lower Bollinger Band is the middle band minus two standard deviations
- HMA_20: The 20-day Hull Moving Average smooths the closing price with reduced lag

Labeling:

- crossover: Identifies the exact day when the EMA50 lines crosses above EMA200 (this is where 'EMA50_prev' and 'EMA200_prev' are used)
- forward return: Measures the stock's 10-day forward return to determine how much the stock price increases or decreases shortly after the crossover
- label: Creates a binary classification target labeling each crossover as profitable (1) if the forward 10-day return exceeds 3%, or unprofitable (0) otherwise

Modelling:

We'll be trying out different models for training, the goal will be to train the best supervised model that can accurately identify "good" crossover events (golden crosses that yield a >3% return within 10 days).

There will be a focus on a variety of tree-based ensemble models, that are well suited for tabular financial data and robust against missing values, which are common in time series datasets, therefore classifiers SVM and Logistic Regression will be excluded. Below are our results for the untuned classifiers:

Untuned Classifiers Classification Report

Metric	***LightGBM***		***Decision Tree***		***Random Forest***		***XGBoost***		***CatBoost***	
	Class 0	*Class 1*	*Class 0*	*Class 1*	*Class 0*	*Class 1*	*Class 0*	*Class 1*	*Class 0*	*Class 1*
Precision	0.74	0.94	0.73	0.96	0.73	0.96	0.74	0.89	0.73	0.97
Recall	0.93	0.78	0.96	0.76	0.95	0.77	0.86	0.80	0.97	0.76
F1-Score	0.82	0.85	0.83	0.85	0.83	0.85	0.79	0.84	0.83	0.86
Support	683	1025	683	1025	683	1025	683	1025	683	1025
Accuracy	0.84		0.84		0.84		0.82		0.85	
Macro Avg Precision	0.84		0.84		0.84		0.82		0.85	
Macro Avg Recall	0.85		0.86		0.86		0.83		0.87	
Macro Avg F1-Score	0.84		0.84		0.84		0.82		0.85	
Weighted Avg Precision	0.86		0.87		0.87		0.83		0.88	
Weighted Avg Recall	0.84		0.84		0.84		0.82		0.85	
Weighted Avg F1-Score	0.84		0.84		0.84		0.82		0.85	

Analysis: When building a model on stock analysis, there is a higher value set on recall for class 1 (“good” crossovers) because missing profitable opportunities (false negatives) means forgoing potential gains which is critical in maximizing returns. LightGBM leads with the highest recall for class 1 at 0.78, ensuring it misses fewer profitable trades. While still focusing on recall, CatBoost and Decision Tree achieve recalls of 0.76 for class 1 and XGBoost slightly edges out at 0.80 but for the higher recall it compensates for a hit on precision at 0.89. Therefore for the untuned classifiers LightGBM is our current champion model.

Now we will look at the hyperparameter tuned classifiers using `RandomizedSearchCV` with the scoring attribute set to ‘f1’ as it encapsulates the balance between precision and recall.

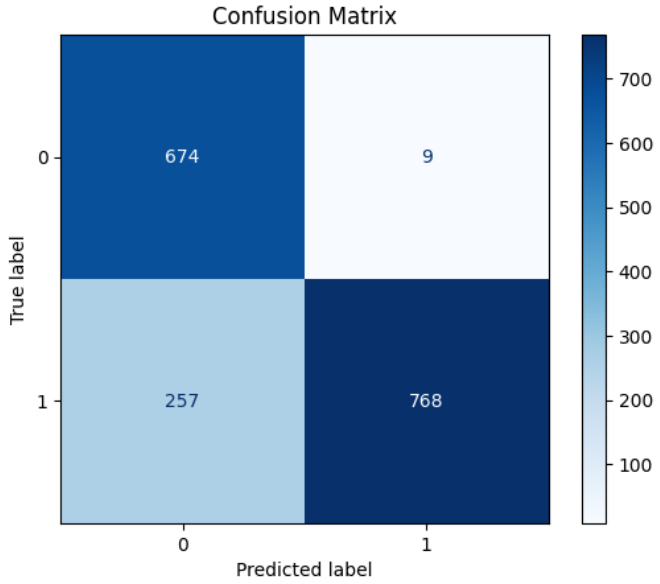
Hyperparameter Tuned Classifiers Classification Report

Metric	***LightGBM***		***Decision Tree***		***Random Forest***		***XGBoost***		***CatBoost***	
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
Precision	0.72	0.99	0.72	1.00	0.72	1.00	0.72	0.99	0.73	0.99
Recall	0.99	0.75	1.00	0.75	1.00	0.75	0.99	0.75	0.99	0.75
F1-Score	0.84	0.85	0.84	0.85	0.84	0.85	0.84	0.85	0.84	0.85
Support	683	1025	683	1025	683	1025	683	1025	683	1025
Accuracy	0.84		0.85		0.85		0.85		0.85	
Macro Avg Precision	0.86		0.86		0.86		0.86		0.86	
Macro Avg Recall	0.87		0.87		0.87		0.87		0.87	
Macro Avg F1-Score	0.84		0.85		0.85		0.85		0.84	
Weighted Avg Precision	0.88		0.89		0.89		0.89		0.88	
Weighted Avg Recall	0.84		0.85		0.85		0.85		0.85	
Weighted Avg F1-Score	0.85		0.85		0.85		0.85		0.85	

Analysis: All tuned models achieve a recall of 0.75 for class 1, with accuracies of 0.85–0.86 and high precision of 0.99–1.00 for class 1. However, the untuned LightGBM model remains the champion with a higher recall for class 1 at 0.78, ensuring it misses fewer profitable opportunities, which is why we're proceeding with it despite the tuned models' improvements.

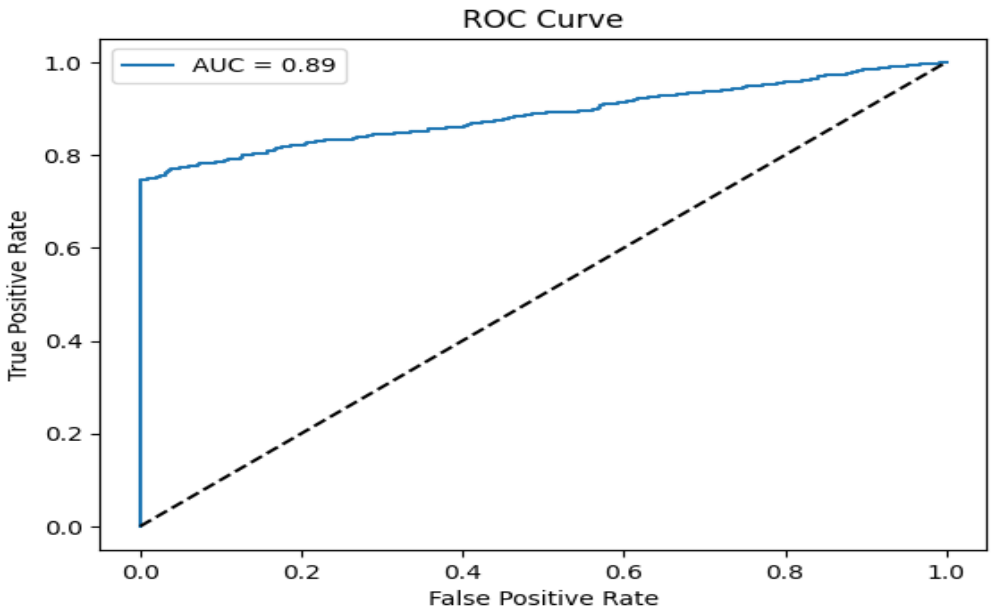
Let's further evaluate our chosen LightGBM model:

Confusion Matrix:



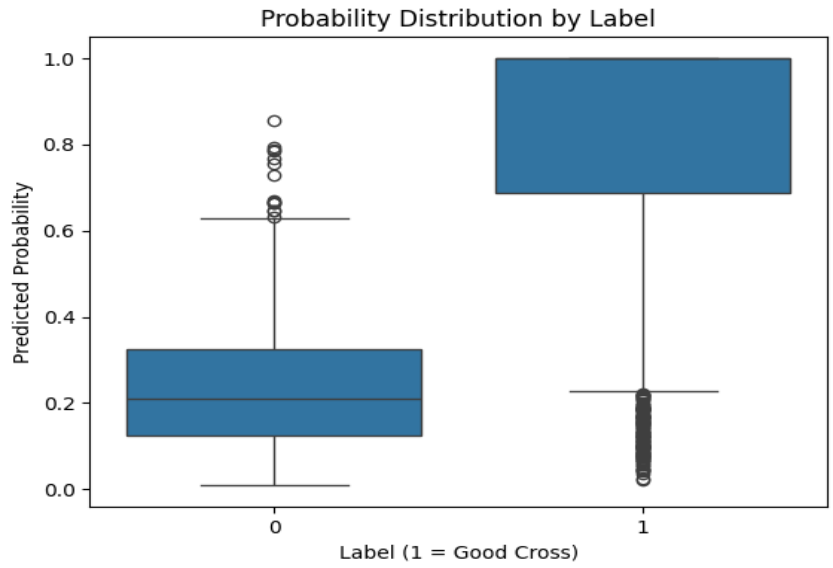
This matrix shows that the model correctly classified 674 of the class 0 (underperforming) cases and 768 of the class 1 (outperforming) cases, with a moderate number of false positives 9 and false negatives 257

ROC Curve:



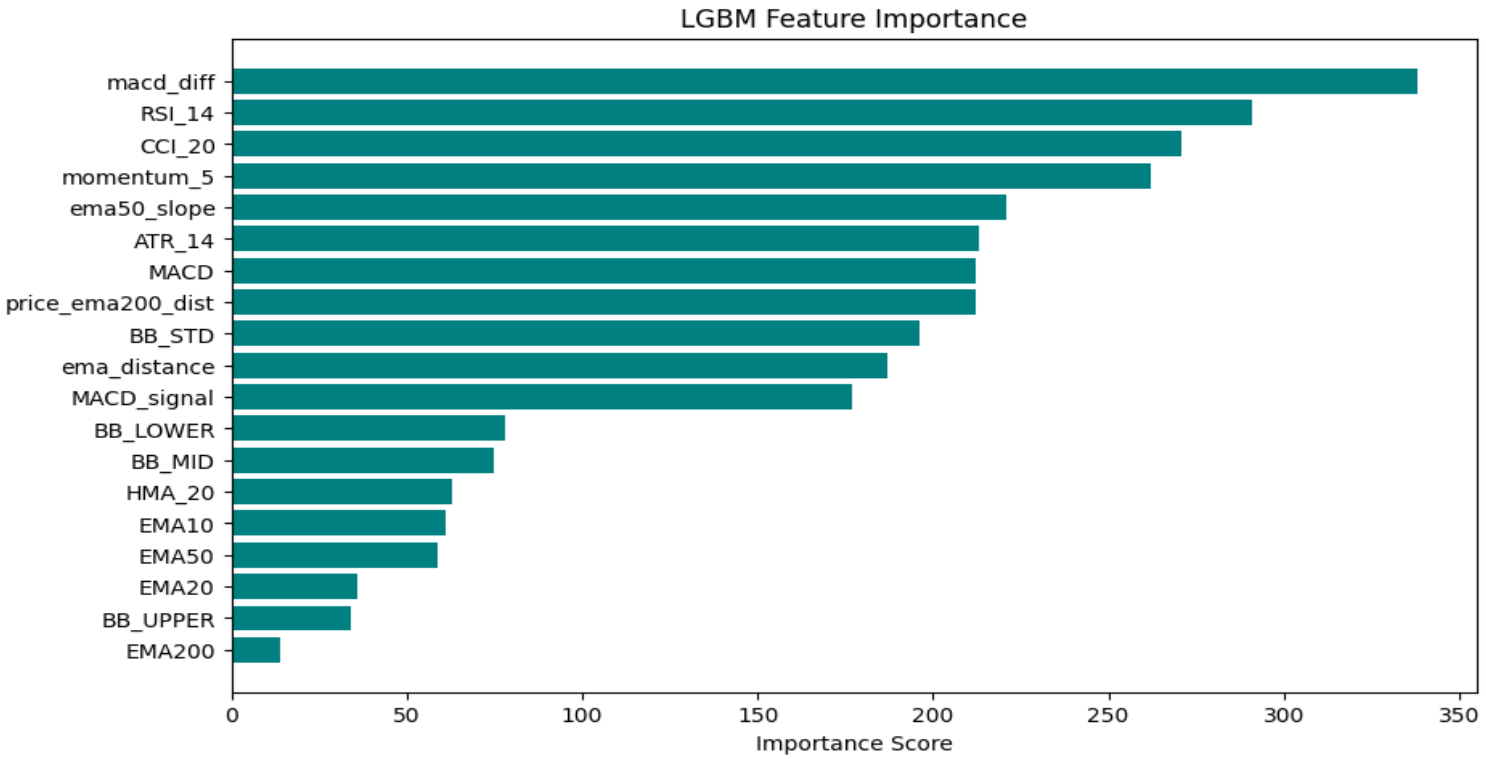
An AUC score of 0.89 indicates strong separability between the two classes, showing that the model is capable of distinguishing good trading signals from poor ones.

Probability Distribution:



This boxplot shows class 1 examples have significantly higher predicted probabilities compared to class 0, confirming that the model is correctly assigning higher confidence to truly positive cases while still maintaining a reasonable spread.

Feature Importance:



This feature importance shows indicators like `macd_diff`, `RSI_14`, and `CCI_20` were the most influential in determining the class label, highlighting which technical indicators had the strongest predictive value.

Simulated Backtest:

With our selected model we now want to test our predictions in a simulated backtest, here's how it will flow:

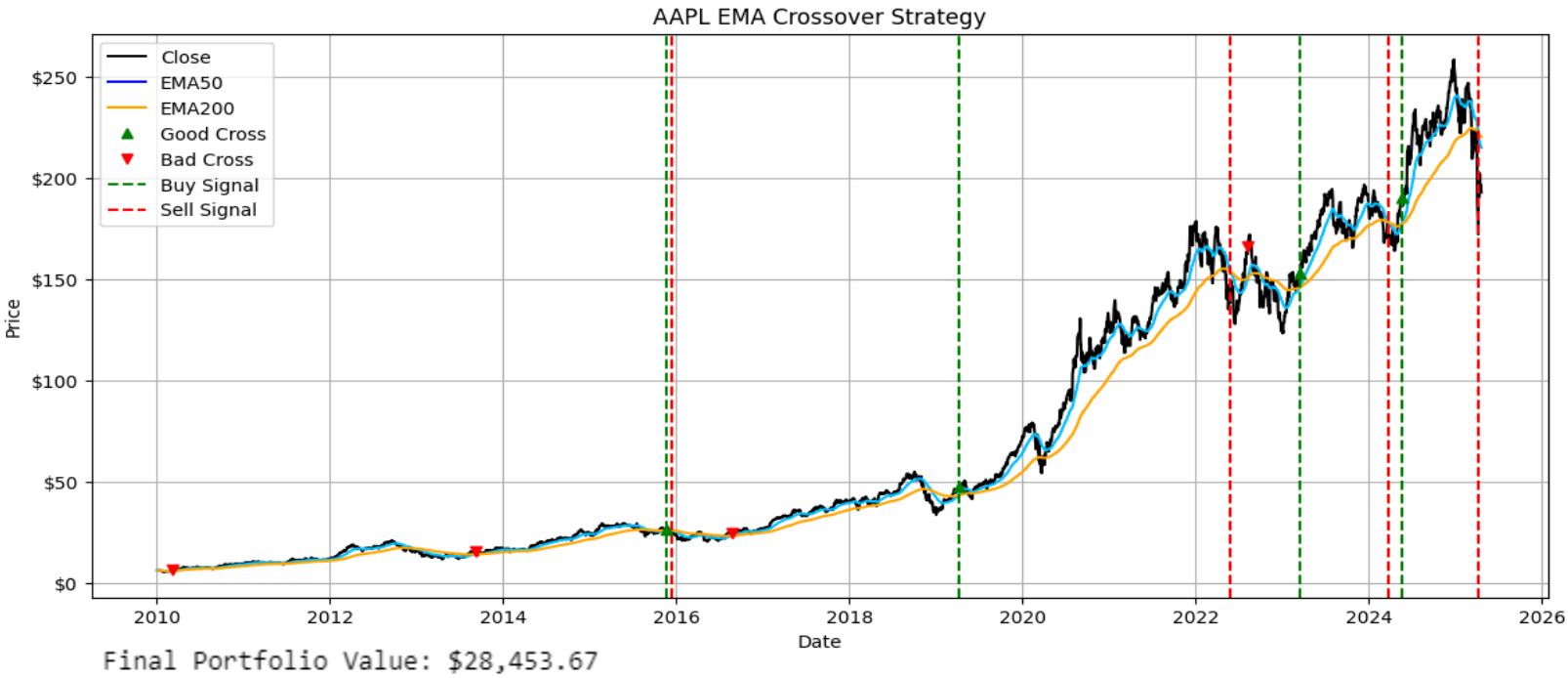
- The backtest will be run on one stock, it will not be a portfolio backtest
- Initial cash is set to $10,000
- A buy signal will be generated depending if the LGBM model predicted a good cross (class 1)
- A sell signal will be generated once EMA50 goes equal or below EMA200 (EMA50 <= EMA200)
- For every buy signal, purchase as much stock with the cash available (all in)
- For every sell signal, sell all stock

To emphasize, the goal of this backtest isn't to aim for a high return percentage, it's to see if the model is able to learn the patterns that make a profitable EMA crossover. Although, this strategy can be implemented with other strategies that would equate to a higher cumulative return.

A table of predicted probabilities for each crossover as well as the trade log showing precise buy and sell statistics is available in the jupyter notebook for deeper analysis.

We will be looking at the performance of AAPL, TSLA, and META stock:

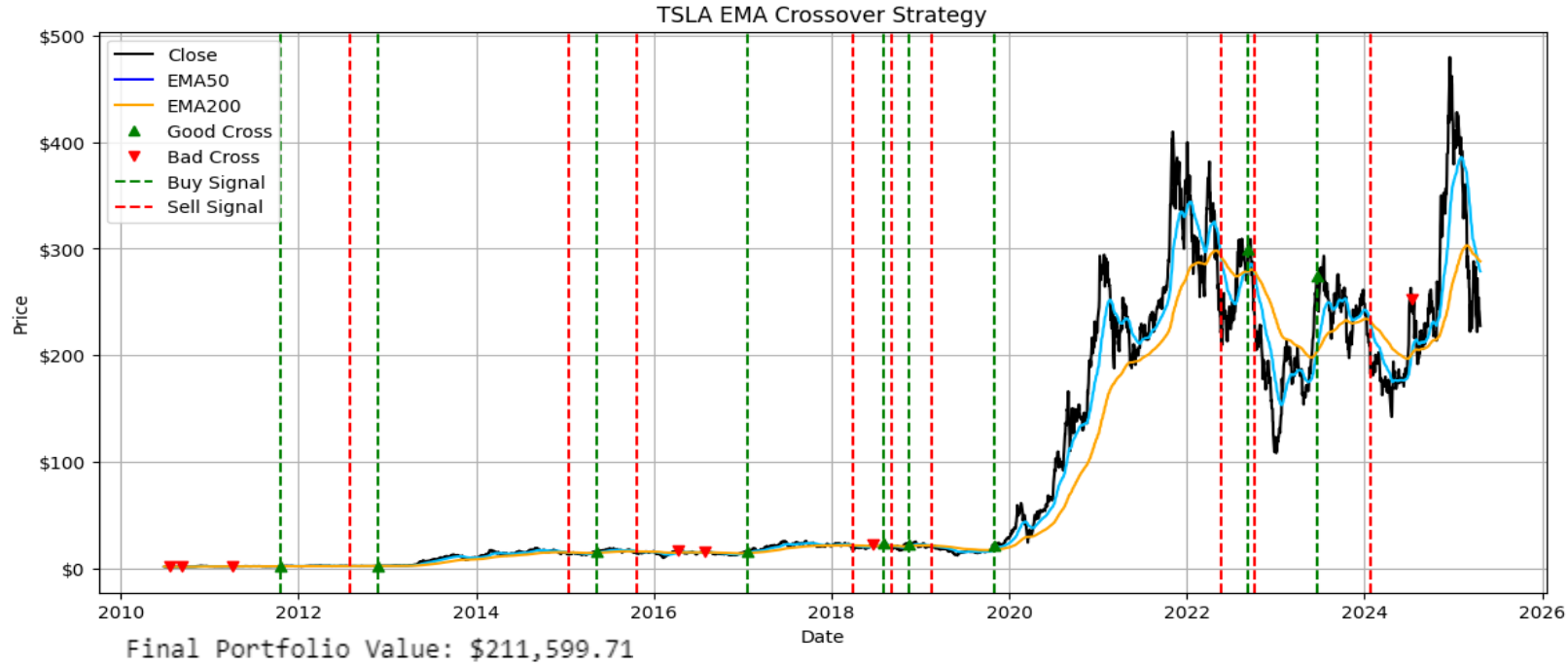
AAPL:




```
Final Portfolio Value: $28,453.67
Cumulative Return: 184.54%
```

For AAPL the model performed fairly well accurately predicting the huge trends in 2019, 2023, and 2024 but making a small error in around late 2016. The cumulative return percentage would definitely be higher if our sell signal wasn't as lagging.

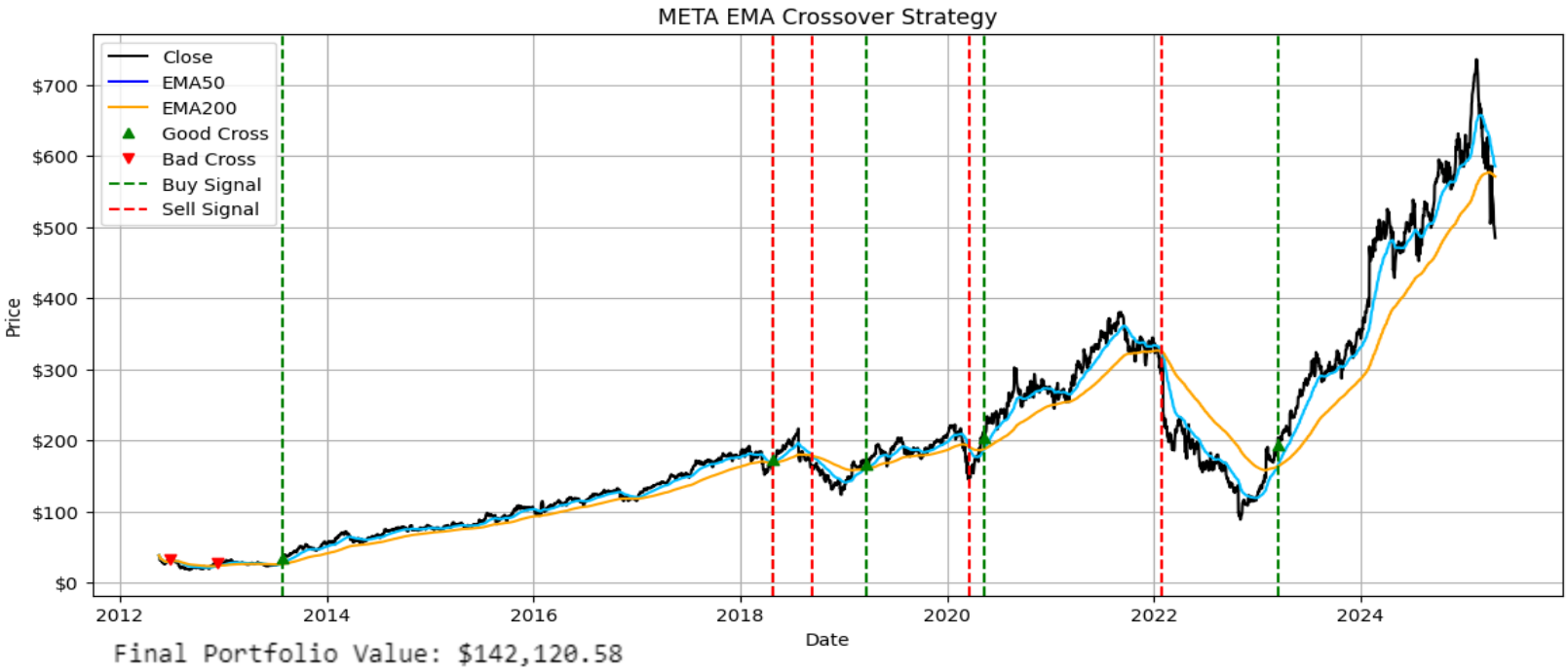
TSLA:




```
Final Portfolio Value: $211,599.71
Cumulative Return: 2016.00%
```

For TSLA our model made more mistakes but accurately predicted the huge leap in late 2019. The big mistake that stands out is the bad cross in 2024, although this could be due to the stocks high volatility which the model seems not to perform well in.

META:



Final Portfolio Value: $142,120.58
Cumulative Return: 1321.21%

For META the detection of good crosses appears to be as intended accurately capturing profitable crosses. In 2018 we see a good cross and then a sell signal almost immediately afterwards this is probably due to EMA50 and EMA200 being extremely close to one another – more in-depth research would have to be done to be certain.

Insights & Opportunities

- It's visible that the model struggles identifying "good" and "bad" crossovers when the stock price is in consolidation, in other words when the price is moving sideways trying to figure out which direction to go. It's understandable why the model struggles in this situation, the model possibly needs access to more technical indicators or a hard coded logic setup that helps deal with this type of situation.

- Further setup to the backtest can be done focusing on the sell signal as it is quite lagging behind. Additionally, the buy signal is also lagging as by the time the "good"

cross has formed there is quite a gap from the most recent lowest price point. If we were able to adjust these signals accordingly it would account for more profitable trades.

- The buy and sell signal can be further improved when dealing with a portfolio, so you can weigh the number of shares to allocate for each stock. Additional macroeconomics data and/or industry based data could prove to be helpful.

- A negative aspect on this strategy is that these crossovers don't happen often so your forced to only think long term. A combination of some short-term strategy with this long-term approach would support a much higher cumulative return.

- This framework can be used in live trading platforms.

- This setup can also be engineered into a stock screener where a system scans and predicts the success probability of any crossovers that occur between the EMA50 and the EMA200 for all live trading stocks.

Summary

This project explored a data driven approach to identifying profitable EMA crossover events using machine learning. By engineering features from common technical indicators and training a LightGBM classifier, we were able to isolate high probability "good" crossovers across S&P 500 stocks.

The model achieved an accuracy of 84%, with a fair balance between precision and recall for good crossover signals. Feature importance analysis confirmed that MACD divergence, RSI(14), and CCI(20) were key drivers in identifying profitable trades.

To test this model in a realistic trading scenario, we created a simulated backtest where trades were initiated at model predicted "good" crossovers and exited when EMA50 dropped below or on par to EMA200. Most of the "good" crossovers became quite profitable and the "bad" crosses were on average unsuccessful achieving little profit or loss.

To conclude, this notebook serves as a strong foundation for building machine learning enhanced technical trading systems, transforming a basic EMA crossover into a more robust and informed decision engine.