Case Study: A Data-Centric Approach to Core Technical Indicators in Algorithmic Trading

Prepared by: Johny Kumar (AI/ML Engineer & Generative AI Developer)
Date: November 5, 2025
Subject: Mini Project – Trading Strategy Case Study

Page 1: Title Page & Abstract

Title: Decoding the Market's Pulse: A Quantitative Case Study on RSI, MACD, and Moving Averages

Abstract:
Technical indicators are the cornerstone of quantitative and algorithmic trading, transforming raw market data into actionable, statistical signals. This case study provides a deep dive into three foundational indicators: the Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and Moving Averages (MA). We will deconstruct their mathematical formulations, implement them in Python, and critically evaluate their application in modern trading strategies from a data science perspective. The objective is not to present a "holy grail" but to equip the reader with the analytical framework to understand, test, and integrate these tools into a robust, data-driven trading system.

Keywords: *Technical Analysis, RSI, MACD, Moving Averages, Quantitative Finance, Algorithmic Trading, Python, Pandas, Data Visualization, Backtesting.*

Page 2: Introduction & The Role of Technical Analysis

1.1 Introduction
In the realm of financial markets, participants are perpetually in pursuit of an edge. While fundamental analysis assesses a security's intrinsic value, technical analysis operates on a different premise: that all known information is reflected in the price, and that historical price and volume patterns tend to repeat. Technical indicators are the statistical tools derived from this price and volume data, designed to forecast the direction and strength of future price movements.

1.2 A Data Scientist's Perspective
For an AI/ML Engineer, technical indicators are essentially feature engineering for time-series data. They are engineered inputs that can be used to train machine learning models (e.g., LSTM networks, Random Forests) to predict price direction or volatility. Understanding their construction and behavior is crucial for developing sophisticated AI trading agents.

1.3 The Indicators Under Review
This study focuses on three pillars of technical analysis:

1. Moving Averages (MA): A lagging indicator used to smooth price data and identify the underlying trend.
2. Relative Strength Index (RSI): A leading momentum oscillator that measures the speed and change of price movements.
3. Moving Average Convergence Divergence (MACD): A versatile trend-following momentum indicator that shows the relationship between two moving averages.

Page 3: Deep Dive into Moving Averages & RSI

2.1 Moving Averages (MA): The Trend Baseline

Concept: A Moving Average simplifies price data by creating a constantly updated average price over a specific period. This smooths out "market noise" and reveals the underlying trend direction.

Mathematical Formulation:

- Simple Moving Average (SMA): `SMA(n) = (P₁ + P₂ + ... + P ) / n` where `P` is the price and `n` is the period.
- Exponential Moving Average (EMA): Applies more weight to recent prices, making it more responsive to new information.
  `EMA_today = (Price_today * (2 / (n + 1))) + (EMA_yesterday * (1 - (2 / (n + 1))))`

Trading Signals:

- Trend Identification: Price above a rising MA (e.g., 50 or 200-day) suggests an uptrend. Price below a falling MA suggests a downtrend.
- Crossovers: A bullish signal occurs when a short-term MA (e.g., 50-day) crosses above a long-term MA (e.g., 200-day), known as a "Golden Cross." The opposite is a "Death Cross."

Python Implementation Snippet:

```
python
```

```python
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt

# Fetch data
data = yf.download('AAPL', start='2020-01-01', end='2023-10-26')
data['SMA_50'] = data['Close'].rolling(window=50).mean()
data['EMA_20'] = data['Close'].ewm(span=20, adjust=False).mean()

# Plotting
plt.figure(figsize=(12,6))
plt.plot(data['Close'], label='AAPL Close Price', alpha=0.5)
plt.plot(data['SMA_50'], label='50-Day SMA')
plt.plot(data['EMA_20'], label='20-Day EMA')
plt.title('Apple Stock Price with Moving Averages')
plt.legend()

plt.show()
```

2.2 Relative Strength Index (RSI): The Momentum Gauge

Concept: The RSI is an oscillator that measures the magnitude of recent price changes to evaluate overbought or oversold conditions. It ranges from 0 to 100.

Mathematical Formulation:

1. `RS = Average Gain over n periods / Average Loss over n periods`
2. `RSI = 100 - (100 / (1 + RS))`

Trading Signals:

- Overbought/Oversold: Traditionally, RSI above 70 indicates an overbought asset (potential sell), while RSI below 30 indicates an oversold asset (potential buy).
- Divergence: When price makes a new high but RSI fails to do so (bearish divergence), or price makes a new low but RSI fails to confirm (bullish divergence).

Python Implementation Snippet:

```
python
def calculate_rsi(data, window=14):
    delta = data['Close'].diff(1)
    gain = (delta.where(delta > 0, 0)).rolling(window=window).mean()
    loss = (-delta.where(delta < 0, 0)).rolling(window=window).mean()
    rs = gain / loss
    rsi = 100 - (100 / (1 + rs))
    return rsi

data['RSI_14'] = calculate_rsi(data)
plt.figure(figsize=(12,6))
plt.plot(data.index, data['RSI_14'])
plt.axhline(70, linestyle='--', alpha=0.5, color='red', label='Overbought
(70)')
plt.axhline(30, linestyle='--', alpha=0.5, color='green', label='Oversold
(30)')
plt.title('Relative Strength Index (RSI) for AAPL')
plt.legend()

plt.show()
```

Page 4: Deep Dive into MACD & A Combined Strategy

3.1 Moving Average Convergence Divergence (MACD): The Trend-Momentum Hybrid

Concept: The MACD reveals changes in the strength, direction, momentum, and duration of a trend by comparing two EMAs.

Mathematical Formulation:

- MACD Line: `(12-period EMA - 26-period EMA)`
- Signal Line: `9-period EMA of the MACD Line`
- MACD Histogram: `MACD Line - Signal Line`

Trading Signals:

- Crossover: The core signal. A buy when the MACD line crosses above the Signal line. A sell when it crosses below.
- Zero Line Crossover: A buy when the MACD crosses above zero (bullish momentum), and a sell when it crosses below.
- Divergence: Similar to RSI, divergence between MACD and price can signal a potential reversal.

Python Implementation Snippet:

```
python
```

```python
def calculate_macd(data):
    ema_12 = data['Close'].ewm(span=12, adjust=False).mean()
    ema_26 = data['Close'].ewm(span=26, adjust=False).mean()
    data['MACD'] = ema_12 - ema_26
    data['MACD_Signal'] = data['MACD'].ewm(span=9, adjust=False).mean()
    data['MACD_Histogram'] = data['MACD'] - data['MACD_Signal']
    return data

data = calculate_macd(data)
plt.figure(figsize=(12,8))
```

```
plt.subplot(2,1,1)
plt.plot(data['Close'], label='Price')
plt.title('AAPL Price')
plt.legend()

plt.subplot(2,1,2)
plt.plot(data['MACD'], label='MACD', color='blue')
plt.plot(data['MACD_Signal'], label='Signal', color='red')
plt.bar(data.index, data['MACD_Histogram'], label='Histogram', color='grey',
alpha=0.5)
plt.title('MACD')
plt.legend()
plt.tight_layout()

plt.show()
```

3.2 A Simple Multi-Indicator Strategy

A robust strategy often combines indicators to filter out false signals.

- Example Logic:
    1. Trend Filter: Go long only when price > 200-day EMA (uptrend).
    2. Momentum Trigger: Within an uptrend, generate a buy signal when:
        - RSI crosses above 30 (exiting oversold territory).
        - AND the MACD line crosses above its Signal line.

This confluence increases the probability of a successful trade by aligning trend and momentum.

Page 5: Critical Analysis & Limitations

4.1 The Pitfalls: A Quantitative Critique
While powerful, these indicators are not infallible. Key limitations include:

- Lagging Nature: MAs and MACD are inherently lagging, as they are based on past data. Signals often appear after a move has already begun.
- False Signals (Whipsaws): In sideways or choppy markets, indicators can generate multiple buy/sell signals that lead to losses. The 200-day MA filter is a common remedy.
- Over-optimization: It is easy to curve-fit parameters to historical data, creating a strategy that works perfectly in the past but fails in the future.
- Self-Fulfilling Prophecy: Widespread use of the same indicators (e.g., RSI 70/30 levels) can create short-term price reactions as traders act in concert.

4.2 The Role of Machine Learning
Modern quantitative finance addresses these limitations by:

- Using Indicators as Features: Instead of relying on static rules (RSI>70 = sell), ML models (like Gradient Boosting or Neural Networks) can learn complex, non-linear relationships between multiple indicators and future returns.
- Regime Detection: ML models can identify if the market is in a "trending" or "mean-reverting" state and dynamically adjust the strategy's sensitivity to these indicators.
- Backtesting Frameworks: Rigorous backtesting with libraries like `backtrader` or `zipline` is non-negotiable to validate any strategy built on these indicators.

Page 6: Conclusion & References

5.1 Conclusion

The Relative Strength Index, Moving Average Convergence Divergence, and Moving Averages remain fundamental tools in the trader's toolkit. They provide a structured, quantitative framework for interpreting market dynamics. From a data science standpoint, they are essential features for more complex models. However, their effectiveness is maximized not when used in isolation with rigid rules, but when combined thoughtfully, understood critically, and integrated into a broader, disciplined trading system that includes robust risk management and continuous validation through backtesting.

5.2 References

1. Murphy, J. J. (1999). *Technical Analysis of the Financial Markets*. New York Institute of Finance.
2. Pring, M. J. (2002). *Technical Analysis Explained: The Successful Investor's Guide to Spotting Investment Trends and Turning Points*. McGraw-Hill.
3. Investopedia. (2023). *Technical Analysis*. [Online] Available at: https://www.investopedia.com/technical-analysis-4689657
4. pandas Documentation. (2023). *pandas: powerful Python data analysis toolkit*. [Online] Available at: https://pandas.pydata.org/docs/
5. McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.