# Automated Trading System with ARIMA Forecasting and Optimized MACD Analysis

Yusup Orazov

## Description

This Python script implements an automated trading system leveraging ARIMA (AutoRegressive Integrated Moving Average) forecasting and MACD (Moving Average Convergence Divergence) analysis. The system utilizes historical price data obtained from the Robinhood trading platform to make predictions about future price movements of a cryptocurrency (Bitcoin).

## Dependencies

- `json`: Used for reading and writing JSON data.
- `time`: Utilized for adding delays to the execution of the script.
- `numpy`: Required for mathematical calculations and array operations.
- `robin_stocks.robinhood`: A Python library for interacting with the Robinhood API.
- `matplotlib.pyplot`: Utilized for creating visualizations such as line plots.
- `statsmodels.tsa.arima.model.ARIMA`: Used for implementing ARIMA modeling.
- `pmdarima.arima.auto_arima`: Utilized for automatic ARIMA model selection.
- `pandas`: Required for data manipulation and analysis.

## Functionality

1. **Robinhood Login:** The script logs into the Robinhood trading platform using user credentials.
2. **Initialization:** Various variables and lists are initialized to store price data, moving averages, MACD values, and predicted prices.
3. **Optimized Moving Averages and MACD Calculation:** The script calculates moving averages (12 and 100 periods) and MACD values based on the historical price data. Additionally, the script employs a method to detect sideways movements by computing the mean of MACD values and subtracting it from the current MACD value after multiplying by a specific threshold. This approach enables the script to identify sideways movement and filter out false signals.
4. **Optimal $p$, $d$, $q$ Selection:** The script determines the optimal values of $p$, $d$, and $q$ for the ARIMA model based on the most recent price data and updates after specific amount of price changes to adapt to market changes more effectively.
5. **Price Prediction:** After being updated with most optimal p, d, q values, ARIMA modeling is used to predict future price movements. Predicted prices are logged and stored for further analysis.
6. **Visualization:** The script generates line plots to visualize historical prices, predicted prices, MACD, and signal line values.
7. **Analysis:** The script performs analysis on recent predicted prices to identify potential trade opportunities. Trade-related data is logged for further review and decision-making.
8. **Continual Execution:** The script continuously runs, periodically fetching new price data and updating predictions and visualizations.

## Backtesting Statistics

- **Accuracy Assessment:** Backtesting is performed using historical price data to evaluate the accuracy and effectiveness of the trading strategy implemented by the script. Another Python script I wrote subtracts real values and predicted values to measure prediction accuracy. It calculates mean absolute error, mean error, minimum error, and maximum error. The error depends on how far into the future the prediction is made. On average, the data for BTC predictions is as follows (BTC Price current @ 42399.95-ish):
  - Mean Absolute Error: 9.059038461538666 = 0.021 percent inaccuracy
  - Mean Error: -3.4961538461539723 = 0.008 percent inaccuracy
  - Minimum Error: -25.19999999999709 = 0.059 percent inaccuracy
  - Maximum Error: 15.459999999999127 = 0.0365 percent inaccuracy

  The largest negative differences are:
  - 27: 42370.75 (Real) - 42345.55 (Predicted) = -25.19999999999709
  - 80: 42387.22 (Real) - 42364.02 (Predicted) = -23.200000000004366
  - 63: 42364.19 (Real) - 42341.28 (Predicted) = -22.910000000003492
  - 83: 42386.36 (Real) - 42364.06 (Predicted) = -22.30000000000291
  - 82: 42386.9 (Real) - 42364.77 (Predicted) = -22.130000000004657

  The largest positive differences are:
  - 47: 42357.71 (Real) - 42371.76 (Predicted) = 14.05000000000291
  - 163: 42384.06 (Real) - 42398.32 (Predicted) = 14.260000000002037
  - 45: 42354.53 (Real) - 42368.93 (Predicted) = 14.400000000001455
  - 44: 42353.68 (Real) - 42368.77 (Predicted) = 15.089999999996508
  - 43: 42353.65 (Real) - 42369.11 (Predicted) = 15.459999999999127
- **Performance Metrics:** Key performance indicators such as profit/loss ratio, win rate, and drawdown are calculated to assess the overall performance of the trading system.
- **Risk Management:** The script incorporates risk management techniques to minimize potential losses, stop-loss and stop-gain thresholds, and optimize returns during back-testing and live trading.

## Note

- The script relies on the availability and accuracy of price data from the Robinhood platform.
- Trading decisions based on the predictions and analyses provided by this script should be made with caution and informed judgment.

## Author

The script was authored by Yusup Orazov and is provided for educational and informational purposes only.