A

SYNOPSIS

of

MINOR PROJECT

on

ALGORITHMIC TRADING USING PYTHON

Submitted by

LOKESH KUMAR MONANI

ROLL NO. 21EGICA021

Project Guide
Mr. Paras Kothari

Head of Department
Dr. Mayank Patel

Geetanjali Institute of Technical Studies, Dabok , Udaipur (Raj.)
Department of Computer Science and Engineering
July,2024

Problem Statement

Develop a robust algorithmic trading system using Python that leverages price action analysis to generate buy and sell signals for Bitcoin (BTCUSD) in a live trading environment.

Brief Description

This project implements a trading strategy using the Dual Thrust method for the BTCUSD market. The algorithm calculates buy and sell triggers based on historical price data and generates trading signals using these triggers. The project involves setting up the algorithm environment, subscribing to BTCUSD data, and executing trades based on the calculated triggers.

Objective and Scope

Objective:

The primary objective is to create an algorithmic trading system using Python that:

- Calculates buy and sell triggers using historical price data.
- Generates and logs trading signals based on these triggers.
- Executes trades in a simulated or live trading environment.

Scope:

The scope includes:

- Developing a price action-based Alpha model to calculate triggers.
- Integrating the Alpha model with the QCAlgorithm framework.
- Implementing portfolio construction and execution models.

Methodology

The methodology outlines the step-by-step process followed to achieve the project's objectives:

1. **Alpha Model Development**: Implement the `PriceActionAlpha` model to calculate buy and sell triggers using the Dual Thrust method based on historical price data.

2. **Algorithm Setup**: Initialize the trading algorithm, and set the backtesting period, initial cash, and brokerage model.

3. **Data Subscription**: Subscribe to BTCUSD data at a minute resolution.

4. **Trigger Calculation**: Use historical data to calculate the Dual Thrust signal range and set buy and sell triggers.

5. **Signal Generation**: Generate buy signals when the current price crosses the buy trigger and cancel insights when the price falls below the sell trigger.

6. **Execution and Portfolio Management**: Use the Immediate Execution Model to execute trades and an Equal Weighting Portfolio Construction Model to manage the portfolio.

Hardware and Software Requirements

Hardware: A computer with internet access.

Software:

- Python 3. x
- QuantConnect Lean Algorithm Framework
- IDE or text editor (e.g., VSCode, PyCharm)

Technologies

- **Programming Language:** Python
- **Algorithm Framework:** QuantConnect Lean
- **Trading Strategy:** Dual Thrust Method
- **Data Handling:** Pandas for data manipulation

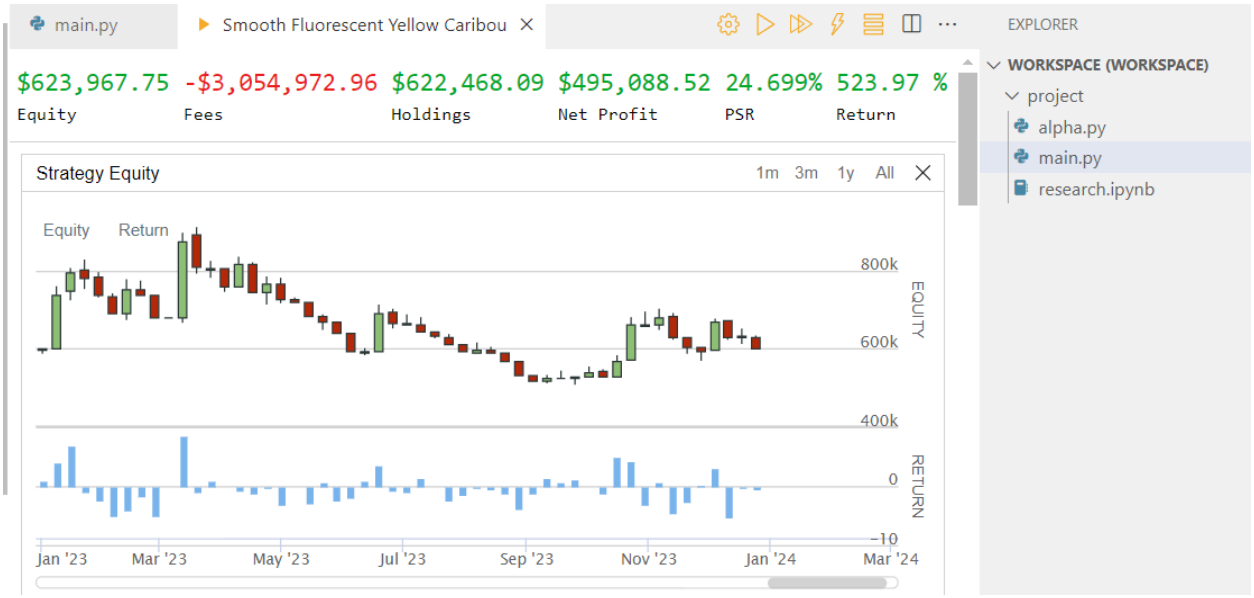
Testing Techniques

- **Backtesting**: Run the algorithm on historical data from 2017 to 2024 to evaluate its performance and make necessary adjustments.
- **Unit Testing**: Test individual components of the Alpha model to ensure accurate calculation of triggers.
- **Simulation Testing**: Simulate the algorithm in a paper trading environment to verify its behavior in real market conditions without financial risk.

Project Contribution

- **Educational:** Demonstrates the implementation of a price action-based trading strategy using QuantConnect's Lean Algorithm Framework.
- **Practical:** Provides a functional trading algorithm that can be adapted and extended for other markets or strategies.
- **Community:** Contributes to the algorithmic trading community by providing a well-documented example of a Dual Thrust trading strategy.

Project Screenshots



Start Equity	100000.0
End Equity	623967.75
Net Profit	523.968%
Sortino Ratio	1.138
Loss Rate	65%
Win Rate	35%
Profit-Loss Ratio	2.74