Financial Data Analyst Entry Task

Hermes Capital Quant Chapter

Fall 2023

Introduction

Welcome to your journey towards becoming a part of Hermes Capital's dynamic team. This entry task is a critical step designed not only to evaluate your aptitude and proficiency in financial data analysis but also to offer a glimpse into the stimulating and innovative environment that defines our firm. Through a series of carefully structured tasks, you will delve into the realms of high-frequency financial data, engaging in complex analyses that mirror real-world challenges in our sector. This exercise is an opportunity for you to demonstrate your analytical prowess, problem-solving skills, and adaptability. Simultaneously, it allows you to acquaint yourself with the intricacies and excitement of a role at Hermes Capital. We encourage you to embrace this task as a meaningful and enriching experience, showcasing your potential to contribute to and thrive in our team. Good luck, and we look forward to your insightful contributions.

1 Task 1: Data Collection, Processing, and Resampling

In this task, you will focus on acquiring and refining high-frequency OHLCV candlestick data from key cryptocurrency exchanges. This involves creating efficient Python functions for data retrieval, transforming raw data into structured formats suitable for analysis, and resampling it to various time scales. The goal is to ensure accuracy, handle anomalies effectively, and prepare the dataset for in-depth time series analysis.

1.1 Data Collection: Fetching OHLCV Candlestick Data

In this segment, you are tasked with the retrieval and initial processing of high-frequency OHLCV (Open, High, Low, Close, Volume) candlestick data. The data should be accurate, consistent, and cover the specified period with the required granularity. Focus on the following cryptocurrency pairs from their respective exchanges:

- **USDT-TMN** from Nobitex (API Documentation).
- **USDT-TMN** from Tabdeal (API Documentation).
- **BTC-TMN** from Tabdeal (API Documentation).
- **BTC-USDT** from either KuCoin (API Documentation) or Binance (API Documentation).

Ensure that your data collection covers the period from **December 1, 2022, to December 1, 2023**, with each record representing a **one-minute** interval. The key tasks are:

1. **Modular Python Functions for API Interactions:** Develop well-structured Python functions that encapsulate the functionality for API interactions with exchanges. Your code should efficiently handle data retrieval, with robust error-handling mechanisms to manage potential issues such as API downtimes, rate limits, or data inconsistencies. Emphasize clean, readable, and reusable code practices to ensure maintainability and scalability of the data fetching process.

2. **Processing OHLCV Data into a Representative Price Series:** Analyze and process the raw OHLCV data to construct a representative price series for each cryptocurrency pair. You are required to determine the most appropriate method for condensing OHLCV data into a single price point per timestamp. Options might include selecting the 'Close' price, calculating the average, or a combination of OHLC values. Critically justify your chosen approach, explaining how it accurately represents market dynamics and supports subsequent analytical tasks.

3. **Extraction of Implied USDT-TMN Price Series:** Utilize the collected BTC-TMN and BTC-USDT data to derive the implied USDT-TMN exchange rate from these two prices as our third USDT-TMN price. In the end, we want to analyze three separate minutely price series for USDT-TMN.

1.2 Resampling

In this segment of the task, you will address the challenge of transforming minutely data into various broader time scales, suitable for both short-term and long-term analytical perspectives. This approach to resampling is critical as it lays the foundation for all subsequent analyses.

1. **Selection of Time Scales:** Resample the minutely data into time frames of **5**, **20**, **60**, and **1440** minutes. These time scales have been chosen to provide insights into intra-day, daily, and longer-term market trends.

2. **Methodological Approach:** You are required to select and implement an appropriate resampling method for each time scale. Possible methods include Time-Weighted Average Price (TWAP), Volume-Weighted Average Price (VWAP), or simply using the last recorded price within each time frame. It is imperative that you provide a detailed justification for your choice of method, focusing on its ability to preserve the essential characteristics of the financial data.

1.3 Handling Market Anomalies

Effective management of market anomalies is essential for the integrity of your analysis. Your task includes identifying and addressing missing data and outliers in a straightforward yet effective manner.

1. **Missing Data:** Implement a basic strategy for filling gaps in the dataset. Options include carrying forward the last known value or using a simple average of adjacent data points. Your approach should be appropriate for the time scales involved and should maintain the continuity of the data series.

2. **Outlier Detection and Correction:** Identify significant outliers using basic statistical methods (e.g., z-scores or box plot derived outliers). Address these outliers (if any) by either adjusting them to a more representative value or excluding them from the dataset, depending on their impact on overall data integrity.

3. **Data Integrity Assurance:** Clearly describe how your methods for handling anomalies preserve the overall quality and reliability of the data. Ensure that the measures taken are simple yet effective in maintaining the true representation of market dynamics.

Key Concepts:

- **OHLCV Data**: Understand the significance of Open, High, Low, Close, and Volume data in financial markets.

- **API Interaction**: Grasp the basics of Application Programming Interfaces (APIs) for data retrieval, focusing on Python usage.

- **Data Processing Techniques**: Learn methods for cleaning, and transforming raw data into a structured and usable format.

- **Time Series Data**: Comprehend the characteristics and analysis of time-stamped financial data points.
- **Resampling Methods**: Explore techniques for changing the frequency of time series data, such as converting minute-level data to hourly data.
- **Anomaly Handling**: Acquire skills for identifying and managing missing data, outliers, and irregularities in datasets.

2 Task 2: Exploratory Data Analysis (EDA)

In Task 2, you'll conduct an in-depth Exploratory Data Analysis (EDA) focusing on key aspects of financial data. This task encompasses the calculation and assessment of log returns and volatility, including a critical examination of their normality, as well as a detailed analysis of autocorrelation and stationarity within financial time series. Additionally, you'll engage in an inter-market analysis to uncover and understand the cross-correlation of log returns across different exchanges, both synchronously and with lagged effects. These analyses are fundamental for developing robust financial models and informing strategic trading decisions.

Important Notes:

- **Price Series Analysis**: You will work with three distinct USDT-TMN price series. Analyze each series individually, and also perform pairwise analyses for all three possible pair combinations of these price series.
- **Multiple Time Scales**: Recall that the data has been resampled into three different time scales. Ensure that each analysis, both single and pairwise, is conducted on each of these time scales, allowing for a comprehensive understanding of market behavior across different time frames.
- **Strategic Use of Visualizations**: The effective integration of visualizations is vital for this analysis. It is your responsibility to discern and judiciously determine where visualizations are most impactful. Employ Plotly to create insightful, relevant, compact, and visually compelling charts, carefully selecting the appropriate chart types for each aspect of your analysis. Visualizations should not be used indiscriminately but should be strategically placed to enhance understanding and to communicate your findings in a clear and effective manner.

2.1 Log Returns, Volatility, and Normality Assessment

In this section, you will delve into the quantitative analysis of financial data by calculating log returns and assessing volatility using the EWMA model. This includes analyzing volatility clustering and performing comprehensive statistical analyses to understand distributional characteristics. A key part of this assessment is evaluating the normality of log returns through graphical and statistical methods, discussing their significance in the context of financial modeling and the challenges posed by non-normal distributions.

1. **Log Returns Computation**: Calculate log returns to normalize price movements, facilitating comparability across time and instruments.
2. **Volatility Estimation and Clustering Analysis with EWMA**: Utilize the Exponentially Weighted Moving Average model to gauge market volatility. In conjunction with this, visually explore volatility clustering to discern distinct market regimes and discuss how these patterns inform risk management practices and strategic decision-making in trading.
3. **Statistical Summaries**: Generate descriptive statistics for log returns and volatility, detailing mean, standard deviation, skewness, and kurtosis to outline their distributional traits.
4. **Graphical Normality Tests**: Use Quantile-Quantile plots to visually inspect the distribution of log returns against a normal distribution.
5. **Quantitative Normality Tests**: Conduct statistical tests such as the Shapiro-Wilk to quantify the normality of log returns.

6. **Importance of Normality**: Discuss the significance of a normal distribution in financial modeling and risk assessment, and consider how non-normality could challenge standard model assumptions, potentially leading to inaccurate risk evaluations and strategy decisions.

2.2 Autocorrelation and Stationarity Analysis

In this analysis, you will explore the fundamental aspects of time series data — autocorrelation and stationarity. By generating ACF and PACF plots, you will investigate the underlying temporal dependencies in price, log returns, and volatility. Complementing this, stationarity tests will assess the stability of these time series. The goal is to understand how autocorrelation and non-stationarity interact, shaping the approach to model building and forecasting in financial markets.

1. **ACF and PACF Plots**: Generate and analyze Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots for price series, log returns, and volatility measures. Use these plots to examine the temporal dependencies and the degree of persistence in market trends.

2. **Stationarity Testing**: Conduct stationarity tests, like the Augmented Dickey-Fuller (ADF) test, on price series, log returns, and volatility to confirm the constancy of statistical properties over time.

3. **Interplay of Non-Stationarity and Autocorrelation**: Examine how non-stationarity and autocorrelation in financial time series are interrelated and critical for effective modeling. Discuss the implications of these properties for predictive modeling and strategy development, particularly how non-stationary data, often exhibiting autocorrelation, may necessitate transformations to capture and accurately forecast market behaviors.

2.3 Inter-Market Analysis

Perform a detailed pairwise cross-correlation analysis of log returns between different USDT-TMN prices. This pairwise approach provides a nuanced view of market interdependencies and is crucial for comprehensive market analysis. This analysis should be conducted at two levels:

1. **Synchronous Correlations**: Assess the immediate correlation of log returns across pairs of USDT-TMN prices **(lag 0)** using a **rolling window of one month**. This will highlight concurrent market movements and potential synchronicity in market reactions.

2. **Lagged Correlations**: Investigate correlations where one set of returns is lagged by one time interval **(lag 1)** using a **rolling window of one month**. This analysis reveals potential lead-lag relationships, indicating predictive or reactionary patterns between markets.

3. **Application to Strategy**: Utilize insights from both synchronous and lagged correlations to inform trading strategies and risk management. The findings can be instrumental in identifying arbitrage opportunities and understanding market dynamics for strategic decision-making.

Key Concepts:

- **Log Returns**: Learn to calculate and interpret log returns from financial price data.
- **Volatility Measurement**: Understand concepts of market volatility and its measurement, including Exponentially Weighted Moving Average (EWMA).
- **Normality Testing**: Familiarize with tests for checking the normal distribution of data, such as the Quantile-Quantile plot and Shapiro-Wilk test.
- **Autocorrelation Concepts**: Study the principles of autocorrelation in time series and methods to measure it.
- **Stationarity vs. Non-Stationarity**: Distinguish between stationary and non-stationary time series and understand their implications.
- **Cross-Correlation Analysis**: Learn to analyze the relationship and potential lead-lag relationships between different time series datasets.

3 Task 3: Cointegration Analysis

In this task, you are invited to explore the intricate dynamics of financial time series through cointegration analysis. Your objective is to assess long-term equilibrium relationships between price series, using advanced econometric techniques. Remember to refer to the **Important Notes** outlined at the beginning of Task 2, as they apply here as well. The methodology (excluding the statistical tests) should be implemented by yourself.

1. **Cointegration Testing Methodology**: Execute cointegration tests by selecting either raw prices or log-transformed prices, justifying your selection based on the inherent characteristics of the dataset, such as non-stationarity. Consider the statistical properties and practical implications of each approach. Utilize standard cointegration tests like the Augmented Dickey-Fuller (ADF) test. Detail the criteria used for test selection and the interpretation of test results in the context of financial market analysis.

2. **Dynamic Analysis of Cointegration Parameters**: Split the dataset into **monthly segments** and perform cointegration tests on each segment separately. This approach allows you to assess the stability and evolution of cointegration relationships over time. Focus on analyzing changes in the cointegration **regression coefficients** α, β and the stationarity **test statistic**. Discuss how these parameters fluctuate and what implications these fluctuations have for understanding market dynamics and potential arbitrage opportunities.

Key Concepts:

- **Cointegration Fundamentals**: Study the concept of cointegration in time series, understanding its significance in financial data analysis.
- **Cointegration Testing Methods**: Get acquainted with cointegration testing techniques such as the ADF test and the Engle-Granger method.
- **Regression in Time Series**: Understand the application of regression analysis in the context of cointegration.

4 Task 4: Error Correction Model (ECM)

Task 4 focuses on the practical application of an Error Correction Model (ECM) to your previously identified cointegrated pairs from Task 3. Your challenge is to not only develop an ECM from the ground up but also to conduct a thorough analysis of its reversion dynamics. This task will test your ability to implement efficient, bespoke solutions without defaulting to standard libraries, and your insights into the speed and significance of market corrections. Evaluating the ECM across monthly data segments, you will gain a deeper understanding of short-term market adjustments and their potential for informing strategic trading decisions. Remember to refer to the **Important Notes** outlined at the beginning of Task 2, as they apply here as well.

1. **ECM Development**: Construct an ECM for the identified cointegrated pairs from Task 3 interchangeably (2 analysis per pair). Emphasize efficiency in your implementation and avoid relying on pre-built libraries. Your code should demonstrate an understanding of the underlying econometric principles and be tailored to the specific dynamics of the financial time series data. Provide a clear rationale for your modeling choices and the steps involved in the ECM construction.

2. **Analysis of Reversion Dynamics**: Perform a detailed analysis of the reversion mechanisms within the ECM, including the **significance** and the **speed** at which deviations from equilibrium are corrected. Split the dataset into **monthly segments** and apply the ECM to each segment to observe how these dynamics evolve over time. Discuss the implications of your findings, particularly in terms of short-term market inefficiencies and potential trading strategies that could exploit these inefficiencies.

Key Concepts:

- **Error Correction Model (ECM)**: Gain an understanding of ECMs, their development, and their role in econometric analysis.
- **Reversion Mechanism Analysis**: Learn to analyze the speed and significance of how deviations from equilibrium are adjusted in ECMs.

5 Technical Requirements

As you undertake these tasks, it is crucial to maintain a high standard in your coding practices. Your code must be not only functional but also optimized for performance and readability. Key aspects to consider include:

- **Modularity**: Structure your code into distinct, reusable modules or functions. This approach enhances maintainability and facilitates easier debugging and testing.
- **Clean Coding Practices**: Write clear, understandable code. Follow Python's PEP 8 style guide to ensure consistency and readability.
- **Efficiency**: Pay special attention to the computational efficiency of your operations. Optimize data processing and analysis steps to handle large datasets without unnecessary resource consumption.
- **Inline Comments**: Use inline comments judiciously to explain the purpose and functionality of complex or non-obvious code segments.
- **Plotly for Visualization**: All plots and visual representations should be created using Plotly. Ensure that your visualizations are not only informative but also aesthetically appealing and interactive, leveraging Plotly's advanced capabilities.

Adhering to these technical requirements will ensure that your work meets the high standards of quality and professionalism expected in this task.

6 Submission Guidelines

To submit your work for this task, please adhere to the following guidelines:

1. **Submission Format**: Complete all tasks in a Jupyter Notebook. Your submission should include comprehensive code along with Plotly visualizations where necessary.
2. **Deadline Determination**: You have 48 hours from receiving this task to set your submission deadline. This deadline is final and cannot be extended under any circumstances.
3. **Communicating Your Deadline**: Email your chosen deadline to quant.hermescapital@gmail.com with the subject line "Entry Task Deadline".
4. **Final Submission**: Send your completed Jupyter Notebook (.ipynb file) to the mentioned email. Use the subject line "Entry Task Submission" for your email.
5. **Timeliness**: Ensure that your submission is sent by the deadline you have determined. Late submissions will not be considered.

By following these guidelines, you ensure that your submission is considered in a timely and organized manner.

7 Evaluation Criteria

Your submission for this task will be assessed on multiple fronts to gauge your potential fit for the position. Key evaluation criteria include:

- **Methodological Correctness**: The accuracy and appropriateness of the methods and techniques applied in your analyses.
- **Analytical Depth**: The thoroughness and insightfulness of your data analysis, including the ability to extract meaningful conclusions from the data.
- **Visualization Quality**: The effectiveness, clarity, and aesthetic appeal of the visualizations created using Plotly.
- **Coding Standards Adherence**: Compliance with best coding practices, code readability, modularity, and efficiency.

- **Task Selection and Completion**: While you have the flexibility to choose which sections to complete based on your strengths and time constraints, your performance in the completed tasks will be a significant indicator of your proficiency for the position.
- **Comprehensive Assessment**: We recognize the diverse backgrounds of our applicants. Your task performance, combined with your background and resume, will be considered holistically. We encourage you not to underestimate your capabilities and to select task sections that best showcase your skills.

We appreciate the varied expertise applicants bring and aim to evaluate each submission fairly, considering both the task performance and individual backgrounds.