TradeCraft: Building Profitable Strategies with Zipline and PyFolio

Sri Karthik A
School of Computer Science and Engineering,
Vellore Institute of Technology,
Chennai, India
srikarthik.a2020@vitstudent.ac.in

Nithin Kodipyaka
School of Computer Science and Engineering,
Vellore Institute of Technology,
Chennai, India
kodipyaka.nithin2020@vitstudent.ac.in

Vinayaka Srinivas R
School of Computer Science and Engineering,
Vellore Institute of Technology,
Chennai, India
vinayakasrinivas.r2020@vitstudent.ac.in

Akila Bala
School of Computer Science and Engineering,
Vellore Institute of Technology,
Chennai, India
akila.b2020@vitstudent.ac.in

Simran Bohra
School of Computer Science and Engineering,
Vellore Institute of Technology,
Chennai, India
simran.bohra2020@vitstudent.ac.in

Shreyas V
School of Computer Science and Engineering,
Vellore Institute of Technology,
Chennai, India
shreyas.v2020@vitstudent.ac.in

***Abstract*- This project explores the evolution and advancements in risk analysis and measurement in the contemporary financial landscape. It underscores the transition of these concepts from being niche to becoming integral parts of portfolio management and trading strategies. The project highlights the practical challenges in risk assessment and analysis, such as the selection of suitable risk models and avoidance of portfolio optimization pitfalls. It also addresses complex issues like horizon mismatches and the significance of out-of-sample testing. The project aims to offer a comprehensive overview of recent developments in risk analysis and modelling, with a focus on their practical applications for portfolio managers and traders. It seeks to illustrate how these tools can provide critical insights into portfolio risk dynamics. TradeCraft presents a comprehensive framework for implementing profitable trading strategies using Zipline, a powerful backtesting library, and PyFolio, a performance analysis tool. This paper outlines the steps involved in developing and evaluating trading algorithms, emphasizing machine learning techniques for predictive modeling. By leveraging Zipline's capabilities for historical data analysis and portfolio management, coupled with PyFolio's robust performance metrics, traders can effectively design, backtest, and optimize strategies for real-world deployment.**

Keywords—Algorithmic Trading, Backtesting Frameworks, Finance, Portfolio Management, Quantitative Finance, Machine Learning Models, Zipline, PyFolio

I. INTRODUCTION

In today's dynamic financial landscape, risk analysis and measurement have evolved from niche concepts to essential components of effective portfolio management and trading strategies. However, mastering the accurate assessment and analysis of risk poses numerous practical challenges, ranging from selecting appropriate risk models to avoiding pitfalls in portfolio optimization. Moreover, issues such as horizon mismatches and the importance of out-of-sample testing further complicate the process.

TradeCraft aims to provide traders and algorithmic developers with a structured approach to building profitable trading strategies. The integration of Zipline and PyFolio offers a seamless workflow from strategy development to performance evaluation, facilitating data-driven decision-making and strategy refinement.

This project aims to provide a comprehensive overview of recent advancements in risk analysis and modelling, emphasizing their practical applications for both portfolio managers and traders. By delving into these developments, we seek to demonstrate how these tools can offer invaluable insights into portfolio risk dynamics. In the ever-evolving financial landscape of today, the concepts of risk analysis and measurement have undergone a significant transformation. Once considered niche, these concepts have now become indispensable components of effective portfolio management and trading strategies. This shift is a testament to the increasing complexity and dynamism of financial markets. This project aims to provide a comprehensive overview of the recent advancements in risk analysis and modelling. It emphasizes their practical applications for both portfolio managers and traders, showcasing how these tools can offer invaluable insights into portfolio risk dynamics. By delving into these developments, the project seeks to demonstrate the potential of these advanced tools in enhancing risk management practices. It explores various risk models, portfolio optimization techniques, and testing methodologies, providing a deep dive into their workings and applications.

In conclusion, the project underscores the importance of continuous learning and adaptation in the field of risk analysis. As the financial landscape continues to evolve, so too must our approaches to understanding and managing risk. By staying abreast of the latest developments and embracing a holistic approach to risk management, portfolio managers and traders can navigate the complexities of the financial markets with greater confidence and efficacy.

II. LITERATURE REVIEW

[1] Financial Risk Measurement for Financial Risk Management

Current practice primarily employs constrained methodologies to market risk measurement, such as historical simulation or Risk Metrics. In contrast, we suggest adaptable approaches that make use of recent advances in financial econometrics and are anticipated to generate more accurate risk evaluations, addressing both portfolio-level and asset-level analysis. Asset-level analysis is complicated since the needs of real-world risk management in financial institutions, particularly real-time risk tracking in very high-dimensional circumstances, place stringent constraints on model complexity. As a result, they emphasize robust but efficient models that are simple to estimate. Furthermore, they emphasize the importance of gaining a better knowledge of the relationships between market risk and macroeconomic fundamentals, with a particular emphasis on the relationships between stock return volatility, real growth, and real growth volatility. Throughout, we endeavours not just to expand our scientific understanding.

[2] Strategic financial risk management and operations research

Risk management has become an important concern for financial organizations since the 1990s. Asset/liability management systems are valuable for managing a company's financial risks. They manage these risks by dynamically balancing the firm's assets and liabilities to meet its goals. We explore how renowned worldwide organizations such as Towers Perrin, Frank Russell, and Falcon Asset Management use asset/liability management to effectively manage risk over long periods. Three aspects of asset/liability management are discussed: 1) A multi-stage stochastic program for coordinating asset/liability decisions; 2) A scenario generating technique for modelling the stochastic parameters; and 3) Solution algorithms for the ensuing large-scale optimization issue.

[3] Financial Risk Management by Insurers: An Analysis of the Process

On-site inspections of financial service organizations were carried out to analyse and evaluate their risk management systems. This evaluation included renowned life/health and property-liability insurers from the United States and around the world. The information gained addressed both the philosophy and practice of financial risk management. This article presents the findings of this inquiry. It describes the current state of risk management approaches in the industry. It reports the standard of practice and assesses how and why it is carried out in the specific manner chosen. In addition, critiques are provided where applicable. We cover the difficulties that the sector finds most difficult to handle, the flaws of the present risk-analysis methodology, and the aspects that are missing in the current methods.

[4] The Present and Future of Financial Risk Management

Current research on financial risk management applications of econometrics focuses on accurately assessing individual market and credit risks, with little theoretical or applied econometric research on other types of risk, aggregation risk, data incompleteness, and optimal risk management. We argue that taking into account the model risk caused by crude aggregation rules and insufficient data may result in a new class of reduced-form Bayesian risk assessment models. Logically, these models should be placed within a common factor framework, allowing correct risk aggregating procedures to be constructed. We demonstrate how such a system could also provide critical linkages between risk control, risk assessments, and efficient resource allocation.

[5] Portfolio Performance Evaluation

This study examines the methods used to measure portfolio performance as well as the evidence supporting professionally managed investment portfolios. Traditional performance measurements, heavily influenced by Sharpe's Capital Asset Pricing Model (1964), were created before 1990. We describe some of the properties and significant issues related to these metrics. We then look at more modern Conditional Performance Evaluation approaches, which are designed to account for projected rewards and hazards that may change over time, addressing a significant deficiency of traditional measurements. We also talk about weight-based performance metrics and the stochastic discount factor technique. We examine the evidence that these newer measures have provided regarding selectivity and market timing skills for professionally managed investment funds. The evidence includes equity.

[6] A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem

Financial portfolio management is the process of constantly redistributing a fund across various financial products. This research introduces a financial-model-free Reinforcement Learning framework that provides a deep machine learning solution to the portfolio management problem. The framework includes the Ensemble of Identical Independent Evaluators (EIIE) topology, Portfolio-Vector Memory (PVM), Online Stochastic Batch Learning (OSBL), and a fully exploiting and explicit reward mechanism. This paradigm is realised in three stages in this study using a Convolutional Neural Network (CNN), a basic Recurrent Neural Network (RNN), and a Long Short-Term Memory (LSTM). They, along with a variety of recently reviewed or published portfolio-selection strategies, are tested in three back-test trials in a cryptocurrency market over a 30-minute trading period. Cryptocurrencies are electronic and decentralized alternatives to government-issued money, with Bitcoin as the best-known example of a cryptocurrency. All three instances of the framework monopolize the top three positions in all experiments, outdistancing other compared trading algorithms. Although with a high commission rate of 0.25% in the backtests, the framework can achieve at least 4-fold returns in 50 days.

[7] Financial modelling: Where to go? With an illustration for portfolio management

The EURO working group on financial modelling defines financial modelling as "the development and implementation of tools supporting firms, investors, intermediaries, governments, and others in their financial-economic decision making, including the validation of the premises behind these tools and the measurement of the effectiveness of their use." Clearly, in this formulation, the decision and its resolution are central. In contrast to our notion of financial modelling, the theory of finance is more concerned with the influence of numerous individuals' decisions and actions on the creation of prices in financial markets. It is not surprising that the assumptions underlying financial theory, which at best explain 'average individuals' and 'average choice scenarios', are not suited to describe specific individual decision problems. In our view it is the role of financial modelling to support individual decision making, taking account of the peculiarities of the actual case, where possible taking benefit from the results of the financial theory. This philosophy towards financial modelling is illustrated by a framework for portfolio management.

[8] Financial portfolio management through the goal programming model: Current state-of-the-art

Since Markowitz (1952) introduced the portfolio selection problem, several scholars have created models that aggregate multiple contradictory variables such as return on investment, risk, and liquidity. The portfolio manager generally seeks the finest combination of stocks/assets to satisfy his or her investing goals. The goal programming (GP) concept is frequently used in finance and portfolio management. The purpose of this study is to present the various versions of the GP model that have been used to the financial portfolio selection problem from the 1970s to the present.

[9] Project portfolio management in practice and in context

Companies struggle with project suboptimization and changes, despite the introduction of different normative directives and best practices for project portfolio management. This study focuses on the importance of understanding project portfolio management in practice and context. The goal is to present a review of recent empirical research literature on project portfolio management, to highlight the limitations of viewing portfolio management as a rational decision process, and to propose new avenues for research on project portfolio management in practice and context. As a result, this article demonstrates that, in addition to rational decision procedures, project portfolio management may be understood as negotiation and bargaining, as well as structural reconfiguration, in order to respond to business uncertainties and complexities.

[10] Financial Portfolio management: Overview and Decision Making in investment Process

Portfolio management is critical in the globalisation era when it comes to security investments. Portfolio management involves both art and science. It entails much more than a financial adviser selecting securities from a catalogue or a security analyst applying a formula to a set of financial data inputs. It is a dynamic decision-making process that is both continuous and systematic, but also necessitates a high level of managerial acumen regarding the securities markets and the individual for whom the portfolio is managed. Portfolio management is a critical aspect for the good performance of new product development and compliance with corporate objectives because it defines not only new product projects, but also revisions, updates, and even decisions regarding the discontinuation of products that are produced and commercialized. This article proposes a framework with the specific objective of presenting an approach that could be useful to portfolio management. The framework proposed in this article presents a holistic perspective of portfolio management, suggesting the use of a set of formal management methods for not only evaluating product projects but also extending to organizational aspects and including them in strategic planning and portfolio reviews.

[11] Financial distress and corporate risk management: Theory and evidence

This research expands on existing theoretical models of business risk management in the presence of financial distress costs and evaluates the model's predictions with a large dataset. I demonstrate that shareholders optimally engage in ex-post (i.e., after debt issuance) risk-management actions, even when there is no prior commitment to do so. The model predicts a positive (negative) relationship between leverage and hedging for moderately (highly) leveraged companies. Consistent with the idea, I observe a non-monotonic relationship between leverage and hedging. Furthermore, the impact of debt on hedging is greater for companies in highly concentrated industries.

[12] Corporate risk management and investment decisions

This paper delves into the critical realm of corporate risk management, particularly focusing on how managers grapple with investment allocation decisions amidst multiple projects. It addresses these challenges through the lens of target-beating in capital budgeting, offering insights that extend to financial management, particularly in the realm of venture capital finance. Employing value-at-risk, a key downside risk measure deemed suitable for economic agents, the analysis leverages probability theory and optimal control methodologies to derive analytical solutions. The findings unveil an optimal corporate investment allocation strategy centered on maximizing the probability of surpassing a predefined target, accompanied by insights into the associated probability and expected earliest time of success. While the study primarily focuses on value-at-risk, future research avenues could explore diverse utility functions of economic agents and alternative dynamic downside risk measures

[13] The Theory and Practice of Corporate Risk Management

The 2008 financial crisis and subsequent recession caught many businesses off guard, serving as a stark reminder of the significance of proper risk management. While academic theory has long emphasised the benefits of risk management, corporations' approaches to putting theory into reality vary widely. Based on a global study of over 300 CFOs from non-financial organisations, the authors indicate that, while most CFOs believe their risk management programmes provide considerable benefits, the risk management function in general need more focus. A considerable number of the finance professionals surveyed agreed that the most significant organisational risks reach far beyond the CFO's direct reporting, and that risk-based thinking is not incorporated into regular business processes.

[14] Corporate Risk Management as a Lever for Shareholder Value Creation

Financial risks, such as unanticipated fluctuations in foreign exchange rates, interest rates, and commodity prices, have a wide-ranging impact on firm value, both directly and indirectly. However, the fact that a large number of firms are allocating resources to risk management operations is merely one indicator of corporate risk management's potential to boost firm value. This report provides a detailed evaluation of positive theories and empirical evidence on the impact of business risk management on shareholder value. It is stated that due of true capital market inefficiencies, such as agency costs, transaction costs, taxes, and increasing costs of external financing, risk management at the business level (rather than risk management by stock owners) represents a strategy to boost firm value.

[15] Portfolio Analysis of Investments in Risk Management

In many practical investment circumstances, the quantity of available memory for stock data is enormous. As a result, many investors are drawn to making judgements based on information that is "currently available in their minds" (see [1, 2]. In this study, numerous risk measuring methods used in investment management are discussed. First, we introduce the concept of the mean variance efficient frontier and Markowitz's approach for identifying all efficient portfolios that maximise expected returns while minimising risk. Markovian risk measures are also discussed. Some measures of portfolio analysis based on the entropy mean-variance frontier are investigated. The risk aversion index and the Pareto-optimal risk distribution are explained. In light of these facts, it is quite interesting to analyse how an investor should make investments.

[16] Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading

The paper discusses the challenges and advancements in stock prediction for quantitative trading strategies. It highlights the limitations of traditional statistical methods and the emergence of deep learning techniques, particularly LSTM networks, for predicting stock movements using historical data. The authors propose using LSTM for prediction and various portfolio optimization techniques such as EQ, MCS, and MVO for constructing efficient portfolios. They report that their LSTM model achieved high accuracy in stock prediction, outperforming linear regression and support vector machine models. Furthermore, the constructed portfolios, employing optimization techniques, demonstrated superior performance compared to the S&P 500 index in terms of returns and Sharpe ratios. Overall, the study suggests the effectiveness of LSTM-based predictions and optimization strategies in quantitative trading.

[17] Portfolio Risk Assessment: A Tool For Dam Safety Risk Management

Dam owners, engineers, and regulators are accountable for the safety of dam groups and should prioritise dam safety reviews or financing for structural and non-structural risk reduction measures. Traditionally, "downstream hazard" assessments, weighting algorithms, and judgement were utilised for this purpose. However, these contain substantial flaws and cannot be used to build an effective and efficient programme for managing and lowering dam safety concerns. This study discusses the Portfolio Risk Assessment (PRA) approach, which the authors created and used to a number of dams. The technique is a tool for prioritising risk-reduction initiatives and doing additional assessments. Furthermore, it adds value to many business processes and helps to strengthen other areas of a dam safety programme.

[18] Risk Assessment and Management of Portfolio Optimization for Power Plants

In electric power markets, power plants become one of the main players and they are facing different choices of supply markets. Different markets have unique fluctuating characters of market prices and revenue rate. The power producer will decide the proportional electricity supply in different markets for a bidding strategy of the annual total electricity supply. The power portfolio strategy is to maximize the annual profit under the lowest level of bidding risks and under the restriction of exceeding the minimum expected annual profit. Risks measurement and management techniques, Value at-Risk (VaR) and Conditional Value-at-Risk (CVaR), proposed in financial research field recently are introduced in this paper to build a portfolio optimization model with CVaR risk minimization for power producers in electricity markets. The mathematical model is discussed, the solution is given out and then the model is applied into a simulation scenario to verify its efficiency. Simulation results indicate that the proposed CVaR based risk measurement and decision model for generators can be efficiently adopted to give out the optimal allocation of bidding generation in different markets.

[19] Managing a portfolio of risks Managing a portfolio of risks

Managers and stakeholders concerned about all sorts of entities often face a complex mixture of threats originating from multiple sources. Traditional management approaches that deal with multiple decisions, such as project portfolio management and risk assessment, are focused on single sources of risk or integrate risks using simplified assumptions (e.g., simple addition). Current challenges – including emergency response to natural disasters, homeland security threats, emerging materials, and correlated events – and an uncertain future with respect to societal concerns – for example climate change, economic development, and social/religious instabilities – require integrated application of these methods. This chapter describes a range of risk approaches that enable a flexible approach to risk portfolio management. Decision quality serves as an integrating framework for applying a mix of analytic methods to facilitate successful risk portfolio management. Methods including portfolio decision analysis and risk management as well as other tools from risk analysis, utility theory, and modern portfolio theory from finance help to address complexities of current risk portfolio management challenges.

[20] A Portfolio Approach to Supply Chain Risk Management

Modern supply networks are important infrastructure for creating organisational value. This infrastructure is frequently exposed to high levels of risk due to a variety of variables, and supply chain risk management is an ongoing study topic. While existing literature has identified a number of risk variables and ways to risk management in supply chains, a more comprehensive approach is required. Prior research has identified some associations between risk management practices and risk. However, there is still need for further study, knowledge, and integration of the linkages between hazards, risk management techniques, and supply chain value. This article examines the various linkages between supply chain risks, risk management solutions, and supply chain value in a comprehensive manner. It proposed a metric of supply chain value based on finance theory and analyzes its relationship with risks and risk management strategies. It develops an analytical model to understand the above-mentioned relationships, proposes a measure of synergy associated with a portfolio of risk management strategies, and develops a more general simulation model to analyze the value of such a portfolio. The simulation model can be used to understand the value of individual supply chain risk management portfolio components, as well as to compare alternate portfolios of risk management strategies.

III. Dataset

The Yahoo Finance dataset serves as a fundamental resource for financial analysis, providing a diverse array of historical data on stocks, indices, currencies, commodities, and economic indicators. This report offers an overview of the key features and utility of the Yahoo Finance dataset for quantitative finance research and analysis.

S&P 500 Index: The benchmark data is retrieved from the Federal Reserve Economic Data (FRED) database using the *pandas_datareader* library. Specifically, it downloads the daily percentage change in the S&P 500 index from the 'SP500' series from 2014 to 2018.

Key Features:

- Historical Stock Prices: The dataset includes daily, weekly, or monthly historical price data for individual stocks, encompassing essential metrics such as open, high, low, and close (OHLC) prices, along with trading volume information.
- Company Information: Comprehensive fundamental data is available, including company profiles, financial statements, earnings reports, and corporate actions such as stock splits and dividends. This data enables in-depth analysis of individual companies and their financial performance over time.
- Market Indices: Historical data for major market indices such as the S&P 500, Dow Jones Industrial Average, and NASDAQ Composite Index are provided. This facilitates analysis of broader market trends and benchmarking of investment portfolios.
- Economic Indicators: The dataset includes historical data for key economic indicators such as unemployment rates, inflation rates, GDP growth, and interest rates. These indicators play a crucial role in shaping macroeconomic trends and influencing stock market movements.
- Currency Exchange Rates: Historical exchange rate data for major currency pairs are available, enabling analysis of currency market dynamics and their impact on international trade and investments.
- Commodities Data: Historical prices for commodities such as gold, oil, natural gas, and agricultural products are included in the dataset. This allows for analysis of commodity market trends and correlations with other asset classes.

IV. Methodology

Fig. 1. Explains the paper outline and a formal methodology/workflow for backtesting a trading algorithm that incorporates a forecasting model. The approach leverages readily available tools and data sources, aiming to provide a practical framework for researchers and practitioners as shown in Fig. 2.

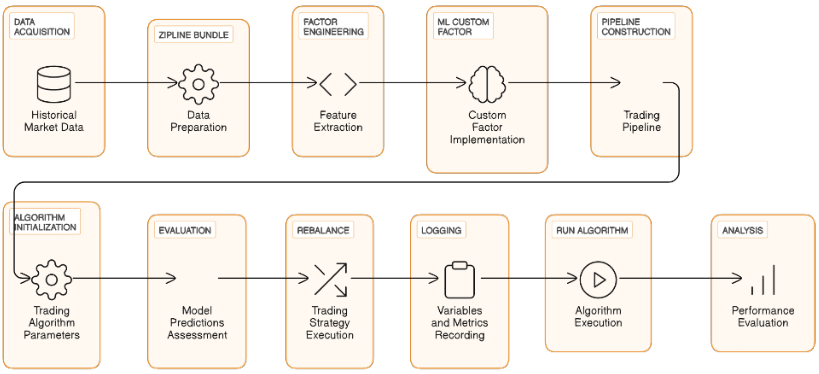

Fig. 1. Workflow of TradeCraft.

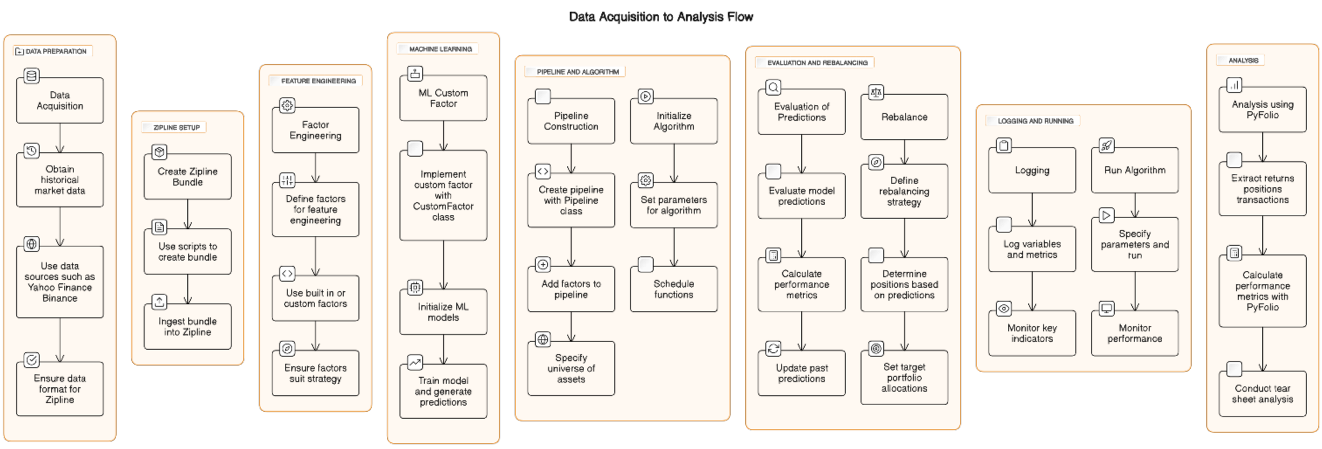

Fig. 2. Module Decomposition of TradeCraft.

A. *Data Acquisition and Preprocessing:*

Data acquisition and preprocessing play a crucial role in developing accurate and robust trading strategies. In the TradeCraft framework, historical market data is obtained from various sources and undergoes preprocessing to extract relevant features and ensure data quality.

Data Acquisition:

Historical market data is typically sourced from financial data providers, including but not limited to:

1. *CSV Files:* Historical price and volume data can be obtained from CSV files downloaded from financial databases or market data repositories.
2. *Online Data Providers:* APIs provided by platforms such as Yahoo Finance, Binance, or Poloniex can be used to fetch historical market data directly into the Zipline environment.
3. *Custom Data Bundles:* Users can create custom data bundles using data from alternative sources or proprietary datasets. These bundles can include additional features such as sentiment scores, economic indicators, or sector-specific data.

Data Preprocessing:

Once the historical market data is acquired, it undergoes preprocessing to prepare it for analysis and model training. The preprocessing steps may include:

1. *Cleaning:* The data is cleaned to handle missing values, outliers, and erroneous data points. Missing values may be imputed using interpolation or forward/backward filling techniques.
2. *Feature Engineering:* Relevant features are extracted from the raw data to capture market dynamics and trading signals. Commonly used features include:
   - Price Momentum: Calculated as the rate of change in asset prices over a specified period.
   - Volatility: Measures of asset price variability over time, such as standard deviation or average true range.
   - Mean Reversion: Deviations from the historical mean or moving averages of asset prices.
   - Money Flow Volume: Measures of buying and selling pressure based on price and volume data.
   - Technical Indicators: Indicators such as Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), or Stochastic Oscillator.
3. *Normalization and Scaling:* Feature scaling techniques such as Min-Max scaling or Standardization (z-score normalization) are applied to ensure that all features are on a similar scale. This prevents certain features from dominating the model training process and helps improve model convergence.
4. *Train-Test Split:* The dataset is divided into training and testing sets to evaluate model performance. Typically, the training set contains historical data up to a certain date, while the testing set contains data beyond that date for out-of-sample evaluation.
5. *Model-specific Preprocessing:* Certain machine learning models may require specific preprocessing steps, such as encoding categorical variables, handling time series data, or performing dimensionality reduction techniques.

B. *Forecasting Model Selection:*

In the TradeCraft framework, various forecasting models are evaluated and selected based on their performance metrics and suitability for the task at hand. The implementation follows a systematic process aimed at identifying the most effective model for predicting asset returns. It demonstrates the implementation of a linear regression model using stochastic gradient descent (SGD) as the forecasting model. The model is trained using historical market data to predict future asset returns. Performance metrics such as IC, RMSE, and MAE are calculated to evaluate the model's predictive accuracy and effectiveness in generating trading signals.

Model Evaluation Metrics:

Before selecting a forecasting model, it is essential to define evaluation metrics that measure the model's performance accurately. Commonly used evaluation metrics include:

- *Information Coefficient (IC):* Measures the correlation between predicted returns and actual returns, indicating the model's predictive power.
- *Root Mean Squared Error (RMSE):* Measures the average magnitude of the errors between predicted and actual returns, providing insights into the model's accuracy.
- *Mean Absolute Error (MAE):* Measures the average absolute difference between predicted and actual returns, providing a robust measure of prediction accuracy.
- *Sharpe Ratio*: Measures the risk-adjusted return of the trading strategy, considering both returns and volatility.

Model Selection Process:

The model selection process involves the following steps:

- *Model Exploration:* Various forecasting models, including linear regression, support vector machines, decision trees, or neural networks, are explored and implemented using suitable libraries (e.g., scikit-learn, TensorFlow, PyTorch).
- *Model Training:* Each forecasting model is trained using historical market data, and the model parameters are optimized to maximize performance metrics such as IC, RMSE, MAE, or Sharpe Ratio.
- *Cross-Validation:* Cross-validation techniques such as k-fold cross-validation or time-series cross-validation are used to assess the model's generalization performance and prevent overfitting.
- *Performance Evaluation:* The trained models are evaluated using predefined evaluation metrics on a

separate testing dataset to assess their predictive accuracy and robustness.
- *Model Comparison:* The performance of each forecasting model is compared based on evaluation metrics, and the model with the best performance is selected for further analysis and deployment.

C. *Backtesting Algorithm Development:*

The backtesting algorithm implemented in the TradeCraft framework leverages the functionalities provided by the Zipline library. We demonstrates the integration of historical market data, custom factors, and predictive models to backtest the trading strategy over a specified time period. Performance metrics such as the Information Coefficient (IC), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) are calculated to evaluate the model's predictive accuracy and effectiveness in generating trading signals. The algorithm utilizes historical market data and simulates the execution of trading signals to assess the strategy's performance over time.

Components of Backtesting Algorithm Development:
- *Strategy Definition:* The backtesting algorithm begins with the definition of a trading strategy. The strategy involves the selection of long and short positions based on predictive signals generated by the forecasting model. The strategy also includes rules for position sizing, risk management, and trade execution.
- *Historical Data Acquisition:* Historical market data is acquired from the specified data sources, such as Yahoo Finance, Binance, or Poloniex. The data includes price, volume, and other relevant features for the selected assets. The backtesting algorithm utilizes this data to simulate trading activity over the historical time period.
- *Pipeline Construction:* A pipeline is constructed using the Zipline library to facilitate the backtesting process. The pipeline defines the universe of tradable assets, filters out assets based on specified criteria, and computes custom factors used for generating trading signals. The pipeline incorporates factors such as price momentum, volatility, and mean reversion.
- *Model Integration:* The backtesting algorithm integrates machine learning models or statistical techniques to generate predictive signals for trading. A linear regression model trained on historical market data is used to predict future asset returns. These predictions serve as inputs for generating buy/sell signals in the trading strategy.
- *Execution Logic:* The execution logic of the backtesting algorithm determines how trades are executed based on the generated signals. The algorithm rebalances the portfolio periodically to maintain target long and short positions. Position sizing and risk management strategies are implemented to control the allocation of capital and manage portfolio risk.
- *Performance Evaluation:* The performance of the trading strategy is evaluated using various performance metrics, including returns, volatility, drawdowns, and risk-adjusted measures such as the Sharpe ratio. These metrics provide insights into the strategy's profitability, risk characteristics, and overall effectiveness in generating alpha.

D. *Backtesting Execution and Analysis:*

The backtesting execution and analysis are integrated into the TradeCraft framework using the Zipline library. The algorithm initializes parameters, fetches pipeline output, generates predictions, executes trades based on rebalancing logic, and records performance variables. Performance metrics are calculated and evaluated to assess the strategy's effectiveness in generating trading signals and generating alpha.

Performance metrics recorded during backtesting are analyzed and visualized to gain insights into the strategy's performance. Visualizations include plots of IC, returns spread, and other performance metrics over time. Additionally, rolling metrics such as rolling returns and rolling Sharpe ratio may be plotted to assess the strategy's consistency and risk-adjusted returns.

Components of Backtesting Execution and Analysis:
- *Algorithm Initialization*: The backtesting algorithm is initialized with the necessary parameters and configurations to define the trading strategy, select assets, and set up performance evaluation metrics. Initialization includes setting up slippage, commission, and scheduling functions for rebalancing and performance recording.
- *Before Trading Start:* Before the start of each trading day, the algorithm executes certain tasks such as fetching pipeline output, generating predictions, and evaluating past predictions' performance. This step ensures that the algorithm is prepared to make trading decisions based on the latest available information.
- *Rebalancing Logic:* The rebalancing logic determines how trades are executed based on the generated signals. The algorithm rebalances the portfolio by buying or selling assets to maintain target long and short positions. Positions are adjusted based on the predicted returns and current portfolio composition.
- *Performance Evaluation:* After executing trades, the performance of the trading strategy is evaluated using various performance metrics. These metrics include the Information Coefficient (IC), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and returns spread in basis points (bps). These metrics provide insights into the strategy's predictive accuracy, risk-adjusted returns, and overall profitability.
- *Recording Variables:* Throughout the backtesting process, relevant variables such as leverage, IC, RMSE, MAE, and returns spread are recorded for further analysis. These variables are stored and used to generate performance reports and visualizations.

V. IMPLEMENTATION

A. *Zipline:*

Zipline bundles serve as essential components in the data infrastructure of algorithmic trading using the Zipline framework. These bundles encapsulate historical market data for various assets, facilitating backtesting and analysis within the Zipline environment. The creation and management of Zipline bundles involve several key steps.

- Backtesting Framework: Zipline provides a comprehensive framework for backtesting trading algorithms using historical market data.
- Data Ingestion and Management: Zipline allows users to ingest and manage historical market data through bundles, which are collections of data for specific assets.
- Pipeline API: Zipline's Pipeline API enables the creation of complex trading strategies by defining factors, filters, and screens to select and manipulate data.
- Integration with Machine Learning: Zipline seamlessly integrates machine learning models into trading algorithms through custom factors, allowing for predictive analytics and data-driven decision-making.
- Execution Simulation: Zipline simulates the execution of trades based on specified parameters such as slippage, commission, and capital base, providing insights into the performance of trading strategies.

Algorithm: Simple Moving Average Crossover Strategy using Zipline

---

Initialize:
- Select the asset to trade (e.g., AAPL)
- Define the window for the simple moving average (e.g., 20 days)

Handle Data:
- For each trading day:
  - Get historical price data for the selected asset over the specified window
  - Calculate the simple moving average (SMA) of the historical prices
  - If the current price is below the SMA:
    - Buy 100 shares of the asset
  - If the current price is above the SMA:
    - Sell 100 shares of the asset

Record:
- Record the current price and SMA for visualization and analysis

---

Firstly, historical market data for assets of interest is collected from different sources such as Yahoo Finance, Binance, or other data providers. This data is then preprocessed and formatted according to Zipline's requirements, ensuring compatibility and consistency across assets. Next, the data is organized into bundles, which are essentially collections of assets with associated pricing and metadata. Each bundle typically represents a specific exchange or data source, containing data for a range of assets traded on that exchange. For example, a bundle may include historical price data for stocks, cryptocurrencies, or futures contracts traded on a particular exchange. The process of creating a Zipline bundle involves packaging the historical data into a format that Zipline can ingest and use for backtesting. This includes converting data into the appropriate format (e.g., OHLCV - Open, High, Low, Close, Volume), handling adjustments such as dividends or stock splits, and ensuring accurate alignment of timestamps. Once created, Zipline bundles can be ingested into the Zipline environment using the 'zipline ingest' command. This command loads the bundle into the Zipline data repository, making the historical data accessible for backtesting and analysis.

B. *Factor Engineering and ML Custom Factor:*

Factor Engineering: Factor engineering involves the creation and selection of factors that capture relevant market behaviors and patterns. Factor engineering is implemented through the definition of custom factors using Zipline's pipeline framework. These factors extract features from historical market data, which are then used as inputs for predictive modeling.

Key factors defined in the code include:

- Momentum: Captures the rate of change in asset prices over a specified period.
- Volatility: Measures the degree of variation in asset prices over time, calculated as the standard deviation of returns.
- Mean Reversion: Indicates the extent to which asset prices deviate from their historical mean, helping identify potential reversal points.
- Money Flow Volume: Reflects the net inflow or outflow of funds into an asset based on price and volume data.
- Price Trend: Estimates the linear price trend using linear regression on historical price data.
- Price Oscillator: Measures the percentage difference between short-term and long-term moving averages of asset prices.

These factors, along with others like MACD Signal and True Range, provide valuable insights into different aspects of market dynamics, forming the basis for predictive modeling.

ML Custom Factor: The ML Custom Factor extends factor engineering by integrating machine learning models into the factor pipeline. The ML Custom Factor, named LinearModel, is implemented to generate predictions using machine learning algorithms.

Key features of the ML Custom Factor include:

- Training Model: The ML Custom Factor incorporates a machine learning model (SGDRegressor) trained on historical data to predict future returns.
- Predictions: On each trading day, the ML Custom Factor computes predictions based on the trained model and current feature inputs.

- Model Training Frequency: The model is trained periodically (e.g., three days a week) to adapt to changing market conditions and ensure model accuracy.

By integrating machine learning into the factor pipeline, the ML Custom Factor enhances predictive capabilities and enables the algorithm to adapt to evolving market dynamics, ultimately improving trading performance.

C. *Pipeline Construction and Algorithm Initialisation:*

Pipeline Construction: Pipeline construction involves assembling a series of factors and computations to generate signals for trading decisions. Pipeline construction is facilitated using Zipline's Pipeline class, which allows for the creation of a structured data processing pipeline as shown in Fig. 3.

- Defining Factors: Custom factors such as Momentum, Volatility, and Mean Reversion are defined to extract relevant features from historical market data.
- Integration of ML Custom Factor: The ML Custom Factor, named LinearModel, is incorporated into the pipeline to generate predictions using machine learning models trained on historical data.
- Pipeline Columns: The pipeline columns consist of various factors and computations, organized to facilitate data processing and signal generation.
- Screen: The screen filters the universe of assets based on specific criteria, such as average dollar volume, to focus on tradable assets.

By constructing a pipeline with appropriate factors and computations, traders can generate signals for trading decisions based on historical market data and predictive modeling.

Algorithm Initialisation: Algorithm implementation involves defining trading rules and executing trades based on signals generated by the pipeline. The trading algorithm is implemented using the initialize, before_trading_start, and rebalance functions, which are called at specific points in the trading cycle.

- Initialization: The initialize function sets up the trading algorithm by specifying parameters such as slippage, commission, and capital base. It also attaches the pipeline to the algorithm for signal generation.
- Data Processing: The before_trading_start function is called every day before market open to generate signals using the pipeline and evaluate model predictions against actual returns.
- Rebalancing: The rebalance function executes trading decisions based on signals generated by the pipeline, adjusting portfolio allocations and executing buy and sell orders as needed.

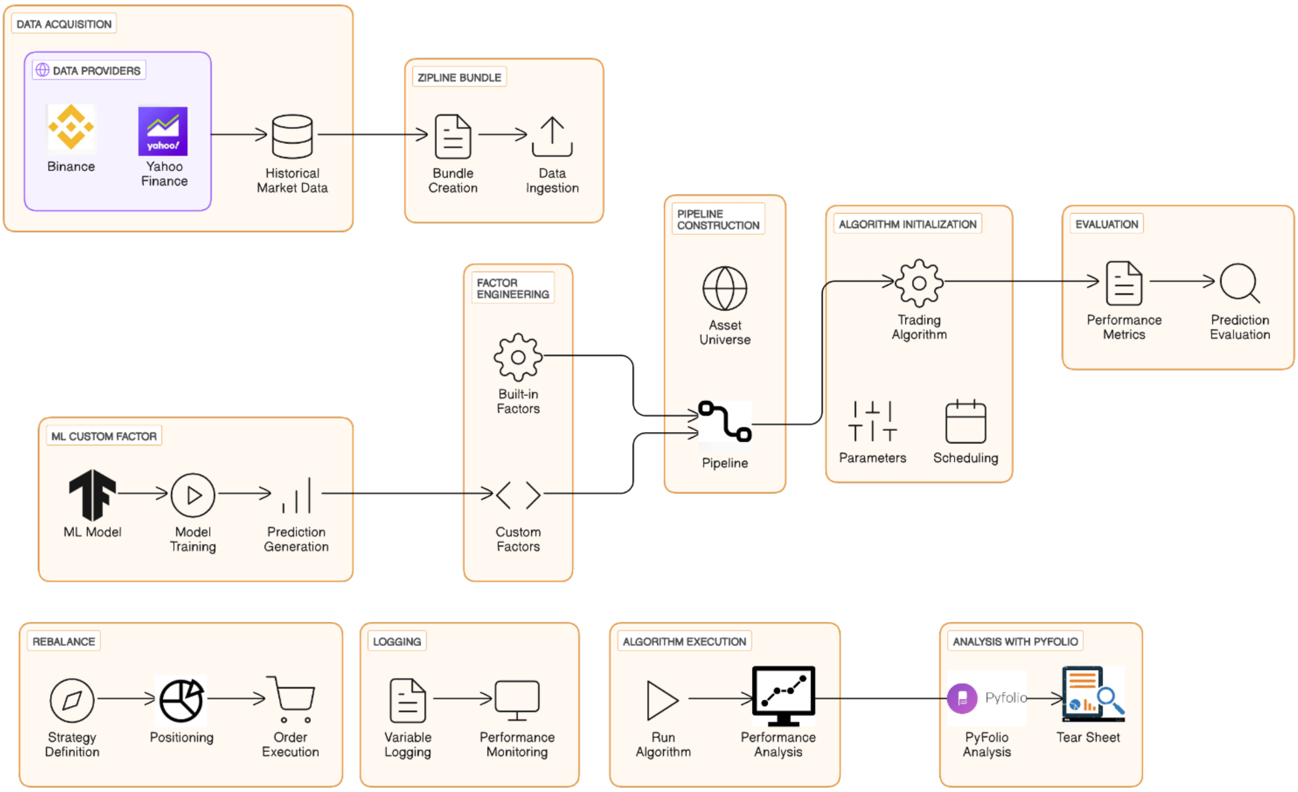

Fig. 3. Integrating ML into Trading Strategies with Zipline.

By implementing a trading algorithm that leverages signals generated by the pipeline, traders can automate trading decisions and execute trades based on predefined rules and strategies.

D. Rebalancing and Logging:

Rebalancing: The rebalancing process is implemented within the *rebalance* function, which is called according to the specified schedule during the trading cycle. The primary objective of rebalancing is to adjust the portfolio's composition based on signals generated by the pipeline, ensuring that the portfolio remains aligned with the desired trading strategy.

Key steps in the rebalancing process include:

- Portfolio Position Adjustment: The function starts by adjusting the existing portfolio positions based on the signals generated by the pipeline. It identifies assets for long and short positions by sorting the predictions and selecting the top N longs and shorts.
- Order Execution: Once the desired positions are determined, buy and sell orders are executed to adjust the portfolio allocations accordingly. Orders are placed using the *order_target_percent* function, targeting the desired percentage allocation for each asset.
- Minimum Position Threshold: To prevent excessive trading and ensure portfolio stability, a minimum position threshold is enforced. If the number of longs or shorts falls below this threshold, the function avoids rebalancing and maintains the existing positions.
- Portfolio Rebalancing: If the number of longs and shorts exceeds the minimum threshold, the function rebalances the portfolio by adjusting positions for each asset accordingly. Long positions are set to target a percentage of the portfolio's value, while short positions are set to target a negative percentage.

By implementing a robust rebalancing process, the algorithm can adapt to changing market conditions and maintain optimal portfolio allocations in line with the trading strategy.

Logging: Logging plays a crucial role in monitoring the algorithm's performance and tracking key metrics throughout the trading process. Logging is implemented using the *record_vars* function, which is called at the end of each trading day to record relevant variables and metrics for analysis.

Key components of the logging process include:

- Variable Recording: The *record_vars* function records various performance metrics, including leverage, information coefficient (IC), root mean squared error (RMSE), mean absolute error (MAE), and returns spread in basis points (bps). These metrics provide insights into the algorithm's performance, risk exposure, and predictive accuracy.
- Performance Monitoring: By logging performance metrics regularly, traders can monitor the algorithm's performance over time, identify areas for improvement, and make informed decisions to optimize trading strategies.
- Customized Logging: The logging function can be customized to record additional variables and metrics based on specific requirements and objectives. This flexibility allows traders to tailor the logging process to their unique trading strategies and goals.

E. Analysis using PyFolio:

Analysis using PyFolio involves leveraging PyFolio's capabilities to generate comprehensive performance metrics, plots, and tear sheets for evaluating the algorithm's performance against benchmarks. Let's delve into how PyFolio is utilized and the functions involved:

Functions Used:

- *extract_rets_pos_txn_from_zipline:* This function extracts returns, positions, and transactions data from the Zipline results, providing the necessary inputs for PyFolio analysis. Returns represent the daily portfolio returns, positions indicate the portfolio's holdings over time, and transactions capture the executed trades.
- *create_full_tear_sheet:* This function generates a comprehensive tear sheet analysis using PyFolio, incorporating various performance metrics, plots, and visualizations. It takes inputs such as returns, positions, transactions, benchmark returns, and a live start date to produce the tear sheet.

PyFolio Process:

- *Data Extraction:* The *extract_rets_pos_txn_from_zipline* function extracts returns, positions, and transactions data from the Zipline results obtained during algorithm execution. These data provide insights into the algorithm's performance, portfolio composition, and trading activity.
- *Tear Sheet Generation:* Using the extracted data, the *create_full_tear_sheet* function generates a comprehensive tear sheet analysis using PyFolio. The tear sheet includes various sections, such as performance summary, returns analysis, position analysis, transaction analysis, and risk analysis.
- *Performance Metrics:* PyFolio calculates and presents key performance metrics, including cumulative returns, annualized returns, volatility, Sharpe ratio, drawdowns, and other risk-adjusted measures. These metrics provide insights into the algorithm's profitability, risk exposure, and overall performance.
- *Plots and Visualizations:* PyFolio generates plots and visualizations to illustrate performance trends, portfolio composition, trading activity, and risk metrics. These visualizations include cumulative returns plots, rolling returns plots, position allocation plots, transaction plots, and drawdown plots.
- *Benchmark Comparison:* PyFolio compares the algorithm's performance against a benchmark, such as a market index or a custom benchmark portfolio. This comparison helps assess the algorithm's relative performance and benchmark outperformance.
- *Customization:* Users can customize the tear sheet analysis by specifying parameters such as risk-free rate, risk-free asset ticker, and benchmark ticker.

This customization allows traders to tailor the analysis to their specific requirements and objectives.

VI. Results and Analysis

PyFolio performs multiple analyses on the trading results generated by the algorithm. Let's break down the analysis based on the different visualizations and functions used:

A. *Plotting Model Performance Metrics:*

The first part of the analysis plots the rolling averages of two key performance metrics: Information Coefficient (IC) and Returns Spread (basis points, bps). These metrics in Fig. 4. provide insights into the algorithm's predictive accuracy and trading profitability, respectively. The rolling averages are calculated over a 21-day period to smooth out fluctuations and highlight long-term trends.

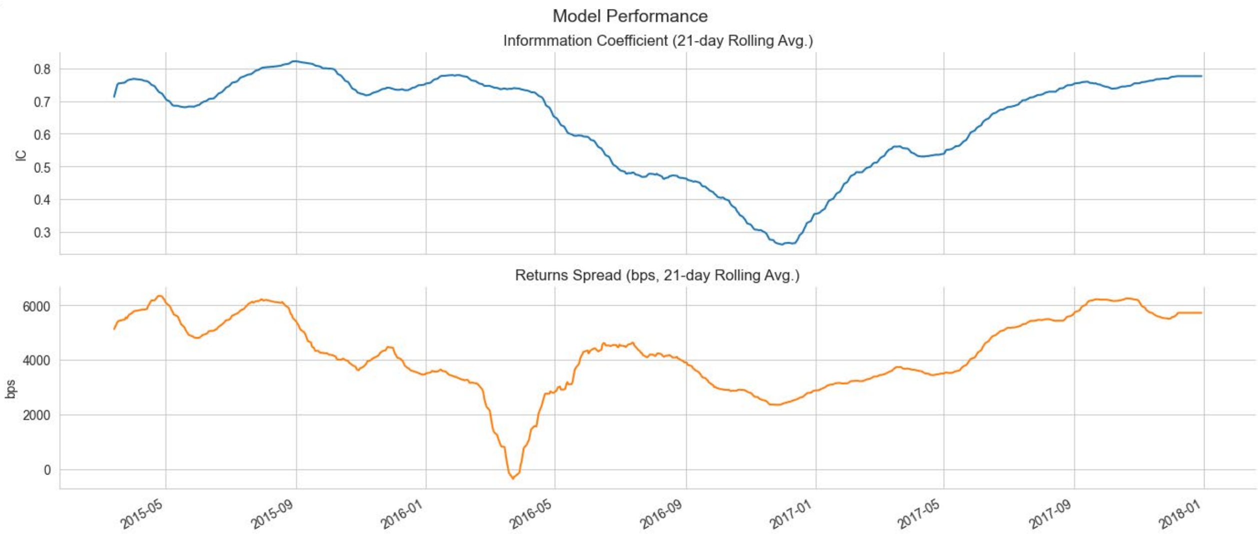

Fig. 4. Model Performance of TradeCraft.

B. *Calculating and Visualizing Cumulative Returns:*

We calculate the cumulative returns of the trading strategy and plots them as shown in Fig. 5 alongside the benchmark (S&P 500) returns. This comparison allows for an assessment of the algorithm's performance relative to the broader market. Additionally, the plot distinguishes between in-sample and out-of-sample returns, providing insights into the strategy's performance during different time periods.

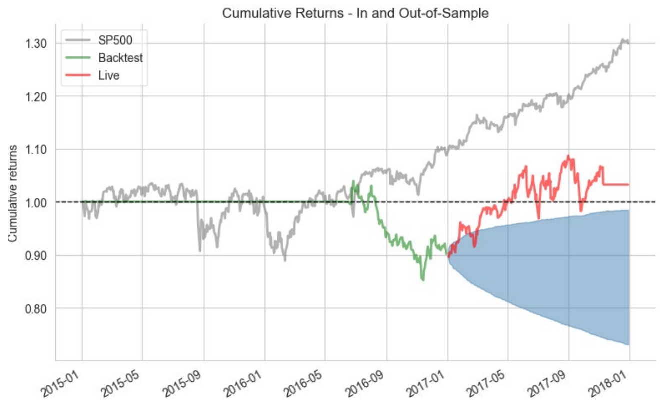

Fig. 5. Cumulative Returns – In and Out-of-Sample.

C. *Assessing Rolling Sharpe Ratio:*

Another plot visualizes the rolling Sharpe ratio in Fig. 6, which measures the risk-adjusted returns of the trading strategy. By analyzing the Sharpe ratio over time, traders can evaluate the consistency of risk-adjusted performance and identify periods of relative strength or weakness.

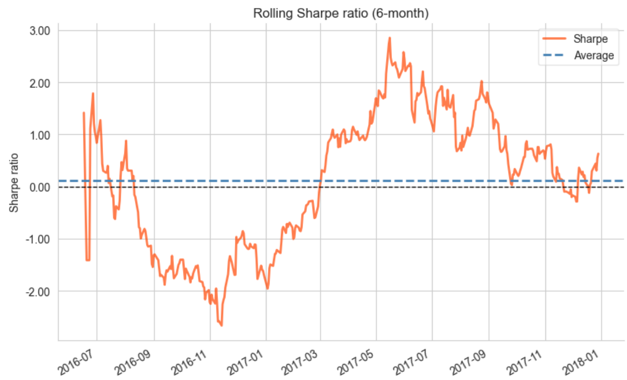

Fig. 6. Rolling Sharpe ratio (6-month).

D. *Generating Full Tear Sheet:*

Finally, we generate a comprehensive tear sheet as shown in Fig. 7 and Fig. 8 using PyFolio's *create_full_tear_sheet* function. This tear sheet includes a wide range of performance metrics, plots, and analyses, such as cumulative returns, drawdowns, rolling metrics, transaction analysis, and risk attribution. By presenting these metrics in a standardized format, the tear sheet facilitates a thorough evaluation of the algorithm's performance and aids in identifying areas for improvement.

Start date			2015-01-02
End date			2017-12-29
In-sample months			24
Out-of-sample months			11
	In-sample	Out-of-sample	All
Annual return	-5.047%	14.546%	1.063%
Cumulative returns	-9.839%	14.484%	3.22%
Annual volatility	7.36%	14.808%	10.449%
Sharpe ratio	-0.67	0.99	0.15
Calmar ratio	-0.28	1.51	0.06
Stability	0.41	0.60	0.00
Max drawdown	-18.042%	-9.612%	-18.042%
Omega ratio	0.80	1.20	1.04
Sortino ratio	-0.94	1.34	0.21
Skew	0.95	-0.76	-0.38
Kurtosis	18.79	2.73	7.98
Tail ratio	0.64	0.74	0.86
Daily value at risk	-0.947%	-1.807%	-1.31%
Gross leverage	2.00	2.00	2.00
Daily turnover	37.261%	33.322%	34.73%
Alpha	-0.05	0.06	0.01
Beta	-0.03	0.47	0.02

Stress Events	mean	min	max
Fall2015	0.00%	0.00%	0.00%
New Normal	0.01%	-3.72%	4.07%

Top 10 long positions of all time	max
sid	
SBAC	8.42%
MAR	8.08%
RCL	8.05%
CAT	8.00%
NVDA	5.19%
SRPT	4.82%
X	4.78%
ADSK	4.59%
NFLX	4.49%
ULTA	4.46%

Summary stats	All trades	Short trades	Long trades
Total number of round_trips	8704.00	4728.00	3976.00
Percent profitable	0.55	0.46	0.66
Winning round_trips	4798.00	2193.00	2605.00
Losing round_trips	3881.00	2518.00	1363.00
Even round_trips	25.00	17.00	8.00

PnL stats	All trades	Short trades	Long trades
Total profit	$2624.99	$-199252.90	$201877.89
Gross profit	$1460860.52	$752889.72	$707970.80
Gross loss	$-1458235.53	$-952142.62	$-506092.91
Profit factor	$1.00	$0.79	$1.40
Avg. trade net profit	$0.30	$-42.14	$50.77
Avg. winning trade	$304.47	$343.31	$271.77
Avg. losing trade	$-375.74	$-378.13	$-371.31
Ratio Avg. Win:Avg. Loss	$0.81	$0.91	$0.73
Largest winning trade	$14587.13	$13203.82	$14587.13
Largest losing trade	$-16378.17	$-15669.29	$-16378.17

Symbol stats	AA	AAL	AAOI	AAP	AAPL	ABBV	ABC	ABT	ADBE	ADI
Avg returns all round_trips	-0.01%	0.09%	-0.06%	0.01%	-0.01%	0.00%	0.01%	-0.02%	0.01%	-0.02%
Avg returns winning	NaN	0.12%	0.05%	0.04%	0.02%	0.03%	0.06%	0.04%	0.02%	0.01%
Avg returns losing	-0.01%	-0.10%	-0.14%	-0.01%	-0.05%	-0.05%	-0.04%	-0.05%	-0.02%	-0.05%
Median returns all round_trips	-0.01%	0.03%	-0.00%	-0.00%	0.00%	0.00%	0.00%	-0.00%	0.00%	0.00%
Median returns winning	NaN	0.05%	0.02%	0.01%	0.00%	0.00%	0.02%	0.01%	0.00%	0.01%
Median returns losing	-0.01%	-0.10%	-0.01%	-0.00%	-0.03%	-0.00%	-0.00%	-0.00%	-0.00%	-0.02%
Largest winning trade	-0.01%	0.47%	0.56%	0.20%	0.18%	0.13%	0.52%	0.22%	0.20%	0.03%
Largest losing trade	-0.01%	-0.10%	-1.58%	-0.16%	-0.26%	-0.19%	-0.23%	-0.33%	-0.11%	-0.16%

Duration stats	All trades	Short trades	Long trades
Avg duration	24 days 00:29:04.163602941	17 days 22:59:38.680203045	31 days 05:29:31.931589537
Median duration	7 days 00:30:00	7 days 00:00:00	9 days 00:00:00
Longest duration	277 days 00:00:00	145 days 23:00:00	277 days 00:00:00
Shortest duration	1 days 00:00:00	1 days 00:00:00	1 days 00:00:00

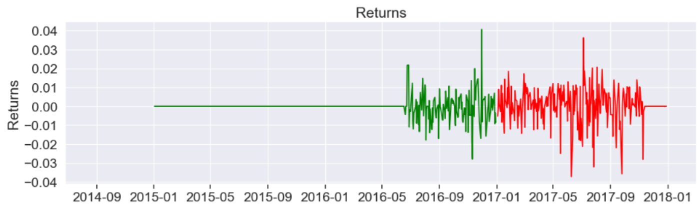

Fig. 7. PyFolio's Tearsheet according to the S&P 500 benchmark.

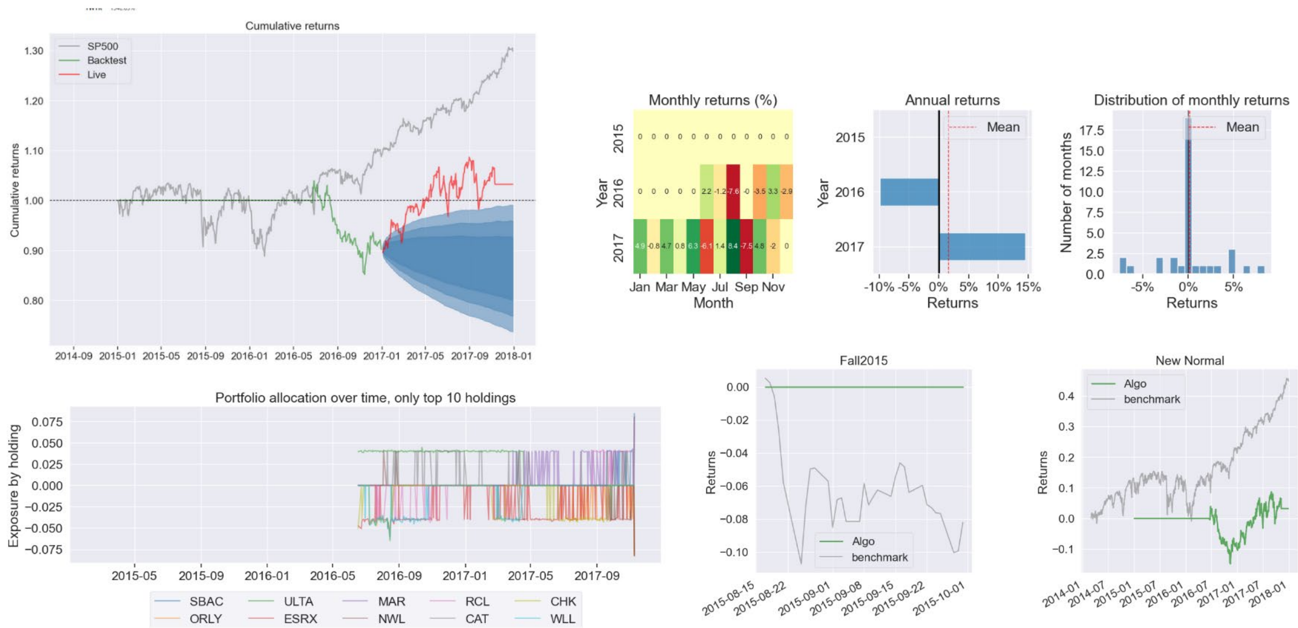

Fig. 8. Full Tearsheet including plots offering valuable insights into its predictive accuracy, profitability, risk management efficacy, and overall effectiveness in navigating dynamic market conditions.

VII. Conclusion

The conclusion drawn from the comprehensive analysis of the trading algorithm's performance offers valuable insights into its effectiveness, robustness, and potential areas for improvement. Through a systematic evaluation of key performance metrics and visualizations, stakeholders can derive actionable conclusions to guide future decision-making processes. One of the primary observations from the analysis pertains to the predictive accuracy of the algorithm, as measured by the Information Coefficient (IC) and Returns Spread. The rolling averages of these metrics provide valuable insights into the algorithm's ability to generate accurate predictions and capitalize on profitable trading opportunities. A consistent upward trend in the IC and Returns Spread indicates a positive correlation between the algorithm's predictions and actual market outcomes, underscoring its efficacy in generating alpha. Furthermore, the analysis of cumulative returns offers crucial insights into the algorithm's overall profitability and performance relative to the benchmark (S&P 500 index). By comparing the cumulative returns of the trading strategy against the benchmark, stakeholders can assess its ability to outperform the broader market and generate alpha consistently over time. Additionally, the segmentation of returns into in-sample and out-of-sample periods enables a nuanced evaluation of the strategy's performance across different market conditions and timeframes. The examination of the rolling Sharpe ratio provides further insights into the risk-adjusted returns generated by the trading strategy. A stable and positive rolling Sharpe ratio indicates a consistent ability to generate risk-adjusted returns, highlighting the algorithm's effectiveness in managing risk and maximizing returns relative to the level of risk undertaken. Conversely, fluctuations or declines in the rolling Sharpe ratio may signal periods of increased risk or potential areas for risk management improvement. Moreover, the generation of a comprehensive tear sheet using PyFolio's toolkit offers a standardized framework for evaluating the algorithm's performance across multiple dimensions. By analyzing a diverse array of performance metrics, including cumulative returns, drawdowns, rolling metrics, transaction analysis, and risk attribution, stakeholders can gain a holistic understanding of the algorithm's strengths and weaknesses. This enables informed decision-making and facilitates iterative improvements to the trading strategy.

In conclusion, the analysis of the trading algorithm's performance provides valuable insights into its predictive accuracy, profitability, risk management efficacy, and overall effectiveness in navigating dynamic market conditions. While the algorithm demonstrates promising results across key performance metrics, there is always room for refinement and optimization. By leveraging the insights gleaned from the analysis, stakeholders can iteratively enhance the algorithm's performance, mitigate risks, and capitalize on emerging market opportunities, ultimately maximizing returns and achieving long-term investment objectives.

REFERENCES

[1] Andersen, T. G., Bollerslev, T., Christoffersen, P. F., & Diebold, F. X. (2013). Financial risk measurement for financial risk management. In Handbook of the Economics of Finance (Vol. 2, pp. 1127-1220). Elsevier.

[2] Mulvey, J. M., Rosenbaum, D. P., & Shetty, B. (1997). Strategic financial risk management and operations research. European Journal of Operational Research, 97(1), 1-16I.

[3] Santomero, A. M., & Babbel, D. F. (1997). Financial risk management by insurers: An analysis of the process. Journal of risk and insurance, 231-270.

[4] Alexander, C. (2005). The present and future of financial risk management. Journal of Financial Econometrics, 3(1), 3-25

[5] Aragon, G. O., & Ferson, W. E. (2007). Portfolio performance evaluation. Foundations and Trends® in Finance, 2(2), 83-190.

[6] Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. arXiv preprint arXiv:1706.10059.

[7] Spronk, J., & Hallerbach, W. (1997). Financial modelling: Where to go? With an illustration for portfolio management. european Journal of operational research, 99(1), 113-125.

[8] Aouni, B., Colapinto, C., & La Torre, D. (2014). Financial portfolio management through the goal programming model: Current state-of-the-art. European Journal of Operational Research, 234(2), 536-545.

[9] Martinsuo, M. (2013). Project portfolio management in practice and context. International journal of project management, 31(6), 794-803.

[10] Kapoor, N. (2014). Financial portfolio management: Overview and decision making in the investment process. International Journal of Research (IJR), 1(10), 1362-1369

[11] Purnanandam, A. (2008). Financial distress and corporate risk management: Theory and evidence. Journal of Financial Economics, 87(3), 706-739.

[12] Li, X., & Wu, Z. (2009). Corporate risk management and investment decisions. The Journal of Risk Finance, 10(2), 155-168.

[13] Servaes, H., Tamayo, A., & Tufano, P. (2009). The theory and practice of corporate risk management. Journal of applied corporate finance, 21(4), 60-78.

[14] Bartram, S. M. (2000). Corporate risk management as a lever for shareholder value creation. Financial Markets, Institutions & Instruments, 9(5), 279-324.

[15] Hooda, D. S., & Stehlik, M. (2011). Portfolio analysis of investments in risk management. The Open Statistics and Probability Journal, 3(1).

[16] Ta, V. D., Liu, C. M., & Tadesse, D. A. (2020). Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. Applied Sciences, 10(2), 437.

[17] Bowles, D. S., Anderson, L. R., Glover, T. F., & Chauhan, S. S. (1998). Portfolio risk assessment: A tool for dam safety risk management. Proceedings of USCOLD 1998 Annual Lecture. Buffalo, New York.

[18] Wang, R., Shang, J. C., Zhou, X. Y., & Zhang, Y. C. (2005, August). Risk assessment and management of portfolio optimization for power plants. In 2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific (pp. 1-5). IEEE.

[19] Keisler, J., & Linkov, I. (2011). Managing a portfolio of risks.

[20] Kumar, R. L., & Park, S. (2019). A portfolio approach to supply chain risk management. Decision Sciences, 50(2), 210-244.