Sentiment-Enhanced Trading with Optimal Policy Trees

15.095 - Machine Learning Under a Modern Optimization Lens

Fiona Aga (faga@mit.edu),
Pranav Girish (pranav7@mit.edu)

Executive Summary

Problem Description: This project aims to develop an intelligent trading agent capable of making real-time trading decisions informed by both historical market data and the sentiment derived from financial news and related articles. By utilizing a combination of sentiment analysis and Optimal Policy Trees (OPTs), we aim to to capture the sentiment-driven fluctuations in the stock market alongside historical stock behaviour to optimize investment returns.

Sentiment Analysis: The proposed trading agent utilizes market sentiment to improve decision-making. In order to quantify this sentiment in an effective manner, we developed a sentiment analysis model. This model utilizes the TextBlob Python library, chosen for its efficiency in assigning sentiment scores. The model was trained and validated on a dataset of labelled financial news articles and then run on an extensive dataset of news related to a chosen set of stocks from Yahoo Finance.

Prescriptive Analysis using OPTs: The main aim of the study is to generate intelligent trading decisions. This is achieved through the use of an optimal policy tree (OPT). The tree was trained on both historical stock price data and corresponding sentiment scores. Once trained and validated, the developed model was benchmarked against rule-based trading decisions. The model was seen to greatly outperform common rule-based investing methods, and thus cements the importance of using both elements of data for a holistic representation of market conditions.

Conclusion: This is an especially useful application of such machine learning models, as when combined with sentiment analysis, the obtained strategies can change based on sentiment-driven market dynamics. The project stands as a testament to the innovative applications of machine learning in finance, showing potential for the future of algorithmic trading where data-driven insights and advanced analytics can enable individuals to make more informed and strategic trading decisions.

1 Background and Motivation

The financial markets are complex ecosystems influenced by a variety of factors, including economic indicators, geopolitical events, and the individual feelings and sentiments of investors. Traditional approaches to stock trading often rely on historical price trends and analysis of other market conditions. However, in recent years, there has been a growing recognition of the impact of sentiment analysis on market dynamics. Investor sentiments, as reflected in financial news and articles, can significantly affect stock prices and market movements. Motivated by the need for more sophisticated and adaptive trading strategies, this project aims to bridge the gap between historical market data and real-time sentiment analysis. The motivation stems from the realization that a comprehensive understanding of market behavior requires a nuanced analysis that incorporates both quantitative historical data and the qualitative aspects of investor sentiment.

The recent growth of natural language processing (NLP) and machine learning techniques has opened new possibilities for extracting valuable insights from textual data, making it feasible to assess the sentiment expressed in financial news articles. Integrating sentiment analysis into trading models presents an opportunity to capture the emotional and psychological factors that drive market fluctuations, providing a more holistic view of market conditions. The utilization of Optimal Policy Trees (OPTs) adds an additional layer of depth to the project. By combining historical market data with sentiment scores, the trading agent aims to find optimal trading policies that dynamically adapt to changing market conditions. The motivation behind this approach is to develop an intelligent trading system capable of outperforming traditional rule-based strategies. To do this, we leverage the power of machine learning to navigate the complexities of sentiment-driven markets. In summary, the project is motivated by the desire to enhance traditional trading strategies by incorporating sentiment analysis and OPTs. This paves the way for a more adaptive, data-driven, and strategically informed approach to algorithmic trading. The combination of using historical data, sentiment analysis, and machine learning techniques to guide our trading decisions allows for better investment returns in the changing landscape of financial markets.

2 Overall Methodology

The methodology used in this project followed a systematic approach to develop and evaluate an intelligent trading agent. Initial data collection involved sourcing comprehensive historical stock price data from reliable financial databases and gathering a diverse dataset of financial news articles related to selected stocks from reputable sources. After much exploration, we settled on Yahoo Finance as the source for both elements of data required for the study. This provided the foundation for subsequent analysis.

A sentiment analysis model was constructed using the TextBlob Python library. This model underwent comprehensive training and validation using a labeled dataset of financial news articles, allowing it to effectively assess sentiment in real-time stock market contexts. The goal was to incorporate qualitative aspects, such as investor sentiment, into the analysis as well as quantitative historical market data.

Feature engineering played a crucial role in integrating historical stock price data and sentiment scores into a carefully curated feature set. This involved capturing various aspects, including historical price trends, volume indicators, and sentiment dynamics. The objective was to create a comprehensive feature set that could represent both the quantitative market behavior and the nuanced sentiment dynamics influencing stock prices.

The core of the methodology involved the application of Optimal Policy Trees (OPTs) to generate an intelligent trading policy. The OPT was trained using the engineered feature set, incorporating both historical data and sentiment scores. The training process aimed to optimize decision nodes within the tree, aligning them with optimal trading policies that could dynamically adapt to the evolving market conditions.

Subsequently, the results were interpreted through a performance assessment, benchmarking the OPT against rule-based trading decisions. Key performance metrics, including return on investment, were thoroughly analyzed to evaluate the effectiveness of the proposed approach. Interpretation involved a comprehensive examination of the trading strategies generated by the OPT, shedding light on its ability to capture sentiment-driven market dynamics and outperform conventional rule-based approaches.

3 Data Acquisition

The data collection process for the project was carefully designed to ensure a robust and comprehensive dataset, which was vital to the the success of our sentiment analysis and trading decision models.

Selection of Time Frame and Stocks: A six-month period was selected for data collection, achieving a balance between obtaining a dataset that was extensive yet manageable. This duration was sufficient for providing actionable insights without overgeneralizing results to a specific market phase. The analysis spanned across 20 different stocks to ensure diversity in the data, preventing the over-reliance on trends from a single stock or market sector.

Sentiment Analysis Training and Validation: The first step of the project was to develop a trained sentiment analyzer. For this, the FinancialPhraseBank dataset from ResearchGate was utilized, which consists of sentences from financial news articles tagged with sentiment scores (positive, negative, neutral). This dataset served as a benchmark to train and evaluate the sentiment analysis models.

Initial Data Collection Attempt: In order to apply the sentiment analyzer, investor sentiment and market news data was required over the observed period. Initial efforts involved using the Alphavantage website for data scraping. However, there were glaring limitations with the Alphavantage API, which restricted our ability to gather data over an extended time frame. This limitation prompted us to seek an alternative data source that could offer more extensive coverage for both historical economic trends and sentiments.

Financial News Data: The pitfalls of the Alphavantage platform led to a pivot to using Yahoo Finance, a more reputable and reliable source for financial data. This platform enabled access to scrape a more substantial amount of data, not only in terms of the time frame but also in the variety and number of stocks covered. The Yahoo Finance dataset provides with a wealth of financial articles, enabling a more in-depth and varied analysis. Advanced web scraping techniques were utilized to systematically extract financial articles related to chosen stocks from Yahoo Finance. This process was automated to efficiently compile a dataset that included a diverse array of articles, capturing various market sentiments and opinions.

Temporal Stock Price Data: Yahoo Finance also provides public access to temporal stock price data. This includes the daily open, close, high, and low prices of a variety of stocks along with their corresponding volume. The data also accounts for auxiliary fees present by providing an adjusted close price value. This dataset was the basis of crucial feature engineering to develop a proper report of economic and quantitative trends.

Data Integrity and Reliability: The shift to Yahoo Finance significantly enhanced the quality and reliability of the dataset. As a widely recognized source in the financial world, Yahoo Finance offered a dataset that could be trusted for accuracy and relevance, which was crucial for the subsequent analysis and prescription.

This comprehensive approach to data collection laid a solid foundation for the project, ensuring that the subsequent stages of sentiment analysis and OPT implementation were built upon a dataset that was not only extensive but also diverse and reliable.

4 Sentiment Analysis

The sentiment analysis phase was a crucial component of our project, aimed at understanding the emotional undertones in financial texts and their potential impact on market behavior.

4.1 Development

Data Preprocessing: The initial step involved loading and preprocessing the FinancialPhraseBank data. This process was essential to prepare the text for analysis, involving cleaning, tokenization, and normalization to make it suitable for machine learning models.

Model Exploration and Selection: During the exploratory phase, an extensive variety of sentiment analysis models were benchmarked. We began with a baseline logistic regression model, followed by an enhanced version with hyperparameter tuning. Our exploration also included random forest models, both with and without hyperparameter tuning, and a neural network approach. However, due to the complexity of our problem and the need for high accuracy, we eventually settled on a Language Learning Model (LLM) approach using BERT (Bidirectional Encoder Representations from Transformers). This choice was driven by the model's superior performance in terms of loss, accuracy, validation loss, and validation accuracy.

Initial Approach - Zero-Shot Classification: Our first approach to sentiment analysis was the implementation of a baseline NLP model using zero-shot classification. This method was initially chosen for its ability to classify sentiment without the need for training on a specific dataset, offering a quick and broad assessment of sentiment in financial articles.

Consideration of Model Fine-Tuning: After applying the zero-shot classification technique, we contemplated fine-tuning this NLP model with specific training to enhance its accuracy and adaptability to financial texts. This consideration aimed to refine the model's ability to capture more nuanced sentiments prevalent in complex financial narratives.

Shift to TextBlob for Efficiency: Upon evaluating the initial approach, we recognized the need for a more direct and efficient method. This realization led us to pivot to using the TextBlob Python library. TextBlob provided a streamlined, effective way to analyze sentiment without the extensive training and fine-tuning process. Its simplicity and direct approach made it a preferable choice for our sentiment analysis needs.

4.2 Results

Before finalizing the BERT model, we conducted a thorough comparison of all explored models. Key metrics for this comparison included accuracy, precision, recall, and the F1-score. We also considered factors like model complexity, scalability, and robustness, ensuring that our final choice was well-rounded and fit for our project's needs. Initially, the LLM model was applied to sentiment predictions on data collected from AlphaVantage. However, with the expansion of our dataset through web scraping from Yahoo Finance, we discovered the TextBlob Python library. This library offered an efficient means to generate sentiment scores for the broader range of articles obtained from Yahoo Finance. This sentiment analysis process not only provided a deep understanding of market sentiments but also laid the groundwork for the subsequent application of Optimal Policy Trees, where these insights were critical in making informed trading decisions.

Model Selected	Evaluation	Inference Time	Training Time	Computational Cost
Logistic Regression	F1 Score: 0.73	Fast	Short	Low
Fine-tuned Logistic Regression	F1 Score: 0.77	Fast	Short	Low
Random Forest	F1 Score: 0.71	Moderate	Longer	Medium
Fine-tuned Random Forest	F1 Score: 0.77	Moderate	Longer	Medium
Neural Network	Test Accuracy: 0.813	Variable	Long	High
Zero-shot LLM	Val Accuracy: 0.85	Slow	None (pre-trained)	High

Table 1: Sentiment Analysis Model Comparison

5 Optimal Policy Tree (OPT)

OPT was chosen for their ability to provide clear, interpretable decision rules and their suitability for time-series data like stock prices. These trees were trained to determine the optimal trading actions (whether or not to buy a certain stock on a certain day) at each decision point, based on the current market conditions and sentiment analysis results.

5.1 Development

Data Collection: The prescriptive modelling efforts started by aggregating quantified sentiments from an article-wise level to a daily level for each stock. Once this was done, the resulting set of scores was merged with the temporal stock price data scraped from Yahoo Finance. This constituted the initial dataset for further analysis.

Feature Engineering: The stock price data available included the open, close, high, and low prices of a stock for each day in the chosen period. Utilizing this data, additional features were engineered to better capture economic trends prevalent in the market. Lagged features gives the model indication of historical prices of the stock for improved training. Further, the Simple Moving Average of prices was taken over multiple horizons (5 and 15 days), as this is a very prominent feature commonly utilized in financial analysis to understand slow moving and fast moving market trends. The limitations of the data available limited us to a horizon of 1 week for reward estimation, wherein the stock prices one week ahead is compared to the current stock price.

Model Training: To train an OPT, we require 2 sets of values; the auxiliary data X, the rewards data y. The auxiliary data in our case is the dataset of financial features and sentiment values for the given set of stocks for the given time period. The reward is defined as the returns of a given stock over a week, which is the percentage increase of price of the stock one week after the current date. This serves as a proxy of the reward obtained by making the decision to buy the stock on the current date, while the reward of not buying the stock is defined similarly taking the opportunity cost or benefit of not making the corresponding investment. Splitting the available data based on date, the model was trained using the developed data and estimated rewards to prescribe whether or not a certain stock should be bought on a certain date.

5.2 Results

Model Version	Weekly profit (%)	Reward Estimation Accuracy (%)
OPT with Sentiment	4.3	80.9
OPT without Sentiment	1.6	65.5
Rule-based strategy #1	0.9	52.9 ⟵ Based on previous returns
Rule-based strategy #2	1.1	59.7 ⟵ Based on SMA Ratio

Table 2: Prescription Results

The developed and trained OPT model was tested on the following month of data. The results are promising, as the prescriptive model makes decisions that result in greater overall profits compared to both a similar formulation without considering sentiment data and rule-based strategies. The rule-based strategies utilized for the purpose of comparison include a strategy wherein all stocks that see an uptick in price from the previous day are bought and a strategy wherein all stocks with a simple moving average ratio greater than 1 are purchased.

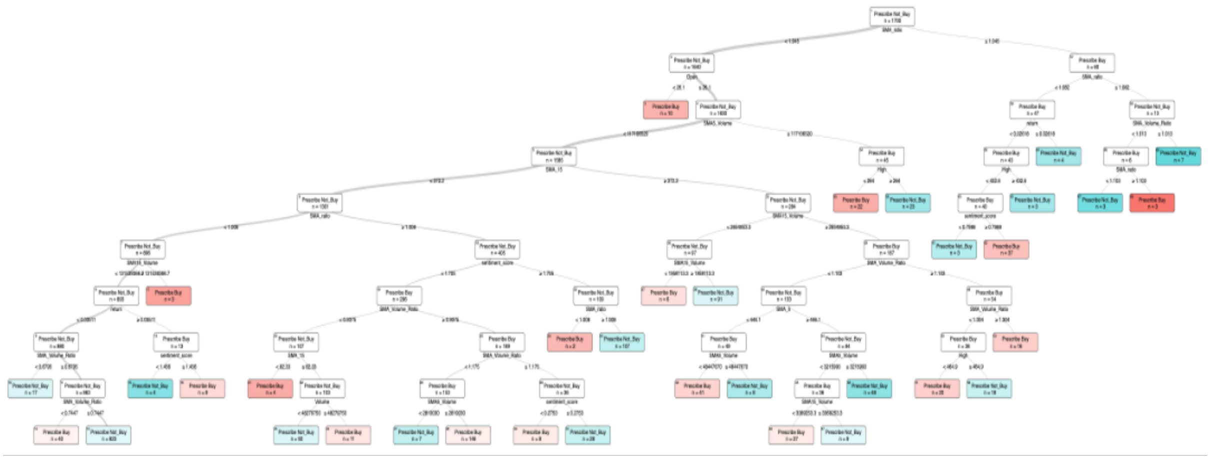

Figure 1: Optimal Policy Tree with sentiment data

6 Conclusion

In conclusion, this project has demonstrated the potential of combining historical market data with sentiment analysis and Optimal Policy Trees (OPTs) to create an intelligent trading agent. The incorporation of sentiment analysis allowed for a more nuanced understanding of market dynamics, capturing the impact of investor sentiment on stock prices. The Optimal Policy Trees, trained on this integrated dataset, showcased a superior ability to generate adaptive trading strategies compared to traditional rule-based approaches. The project's success highlights the importance of leveraging machine learning techniques for a holistic representation of market conditions. The integration of quantitative and qualitative data has allowed for more informed and strategic trading decisions. The results indicate that such an approach could offer a competitive edge in the dynamic and sentiment-driven landscape of financial markets. This projects outlines the importance of machine learning methods in finance, showing us what the future of algorithmic trading could entail. By utilizing data-driven insights and advanced analytics, the project highlights the potential for evolving trading strategies that adapt to real-time market sentiment, ultimately enhancing investment returns.

Possible Implementation Improvements: While this project has made significant progress in developing an intelligent trading agent, there are several avenues for future exploration and refinement. Further improvements in sentiment analysis models could be explored to better capture the intricacies of financial news and social media sentiments. This could involve utilizing more sophisticated natural language processing (NLP) techniques and incorporating a broader range of textual data sources. Experimentation with more advanced machine learning models beyond Optimal Policy Trees may also yield additional insights. Exploring deep learning approaches, reinforcement learning, or ensemble models could enhance the agent's ability to adapt to complex market conditions.

Future Steps: Moving from a retrospective analysis to real-time implementation is a crucial step. Integrating the intelligent trading agent into live trading environments and assessing its performance in real-time conditions would be essential for practical deployment. Future work could focus on incorporating sophisticated risk management strategies to ensure the robustness and stability of the trading agent in various market scenarios. Further expanding the scope of the project to include a broader range of financial instruments, such as commodities or cryptocurrencies, would provide a more comprehensive understanding of the trading agent's applicability across different markets. In pursuing these future steps, the goal is to continually refine and enhance the intelligent trading agent, making it more adaptive, robust, and capable of navigating the changing landscape of financial markets.

Individual Contributions

In order to make sure both of us equally contributed to the project, we split it up in such a way that each person prioritizies and manages a different component of the project. Our project can be broken down into 4 main categories, which are as follows: data processing and generation, sentiment analysis, feature engineering, and optimal policy trees. Fiona focused on the data processing and generation, and sentiment analysis components. Pranav focused on the feature engineering and optimal policy trees components. While each of us primarily led the team efforts in their assigned areas, we both assisted each other at various stages to ensure that both of us had a solid understanding of all areas of the project. For instance, each section can be broken down into further subcomponents, so we made sure that when one of us was leading a task, the other team member was available to assist for debugging, additional challenges that may have arised in the middle, or simply doing a code review. In addition to the main components of building our final product, we had to draft the project proposal, write the final report, and create the slides. For these three tasks, we divided the work equally, working on it synchronously and making changes as we gathered feedback from one another. Overall, we feel that the work was split up in a way such that both team members have a solid understanding of all concepts involved and made sure to contribute to each other's primary tasks and assist whenever required.