A Modern Approach to Financial Visualization

Richie Fleming, Nick Antoine, Will McCarter

Table of Contents

Chapter 1

The purpose of this project is to develop a data-driven solution that enhances the interpretability and usability of stock price visualizations, particularly for day traders. Traditional tools such as candlestick charts, while widely used, rely solely on price-based attributes and subjective pattern recognition techniques, which limit their predictive capability and accuracy. These charts, originally developed in the 18th century to track rice prices, have remained largely unchanged despite substantial evolution in financial markets and the availability of data.

Candlestick charts are based on the open, high, low, and close prices of a stock over a specified time interval. Traders interpret these patterns using named formations such as *hammers* or *doji*, which lack standardized definitions and introduce ambiguity. For example, terms like "significantly lower" or "close near the opening price" are inherently imprecise, leading to inconsistent interpretations and potential trading errors. Moreover, these patterns consider only historical price behavior, ignoring other significant market variables such as trading volume, sentiment, and macroeconomic indicators.

This project proposes a new visualization framework using machine learning to capture a broader spectrum of market dynamics. The goal is to build a more comprehensive and interpretable representation of asset price movements, improving the decision-making accuracy of traders and providing an upgrade over the heuristics of candlestick strategies.

Research Question: How can a machine-learning-enhanced visualization, integrating wider market data, improve day traders' interpretability and decision-making accuracy relative to traditional candlestick chart representations?

Business Objectives

1. Develop and refine trading strategies to achieve consistent success in dynamic market conditions.
2. Outperform standard candlestick trading strategies, thereby driving higher returns for investors.
3. Construct a network of interacting variables within a chosen industry to explain and model asset price variability.

Data Mining Objectives

1. Analyze historical market data to test standard and custom trading strategies.
2. Identify weaknesses in widely used visualization methods, such as candlestick patterns, and enhance them through data-driven visual and statistical models.
3. Optimize model performance by identifying the conditions under which specific strategies and indicators yield the most reliable outcomes.

Scope and Deliverables The scope of the project includes collecting historical and real-time market data across various sectors, developing models, and constructing a visualization platform. Emphasis is placed on uncovering hidden relationships among market variables, building an interpretable interface, and conducting empirical validation through model testing and user evaluation.

Deliverables:

- A set of predictive and explanatory models for analyzing asset price movements.
- A web-based visualization tool consolidates relevant indicators and model outputs.

- Performance benchmarks comparing the proposed system against traditional candlestick strategies.

Limitations

- Lack of public benchmark data on the accuracy of traditional candlestick strategies necessitates custom performance calculations.
- Ambiguity in candlestick pattern definitions introduces challenges in evaluation and modeling.
- Market dynamics are inherently non-stationary; models may require frequent recalibration to maintain relevance.

This project ultimately seeks to modernize technical analysis by integrating robust modeling techniques with intuitive visualization, thereby offering a more reliable and insightful tool for day traders.

Chapter 2

This chapter outlines the datasets used to support the project's objectives, including their sources, structure, and relevance to our data mining goals. The dataset consists of historical market data retrieved using the Yahoo Finance API (yfinance), covering a wide range of equities, indices, and commodities. The time span selected was from June 1, 2021, to April 1, 2025, providing a rich temporal window for training and testing models against both stable and volatile market conditions.

Data was gathered for 75 financial assets, including stocks from major technology companies (e.g., NVDA, AAPL, AMZN), indexes (e.g., ^VIX, ^TYX), commodities (e.g., GC=F, CL=F), ETFs (e.g., TLT, AGG), and select firms in the transportation and defense sectors. The data spans three distinct market periods:

- **Training Set**: June 1, 2021 – May 1, 2023
- **Testing Set**: May 2, 2023 – April 1, 2025

Each record corresponds to a trading day and includes the following attributes: Open, High, Low, Close, and Volume. These variables are foundational for both traditional candlestick-based strategies and our model-driven visualizations. Daily records were augmented with calculated features: Up (1 if Close > Open), Down (1 if Close < Open), allowing for binary movement labeling.

Initially, the focus was on NVIDIA, enabling rapid iteration and model validation. After confirming model robustness, the scope expanded to include high-cap stocks such as TSLA, MSFT, AAPL, META, AMZN, and GOOG. These companies were selected due to their significant market influence and volatility, ideal for stress-testing trading strategies. This generalization was necessary to meet the data mining objective of identifying conditions under which specific strategies succeed across multiple assets.

The data directly supports the following data mining objectives:

- **Strategy Evaluation**: Binary trend labels (Up/Down) to test traditional candlestick strategies and compare them with model predictions.

- **Weakness Identification**: The rich, multi-sector dataset enables the detection of conditions where candlestick strategies fail, facilitating targeted improvements.
- **Optimization Conditions**: By splitting the data temporally and applying models across sectors, we evaluate how market behavior affects performance, enabling optimization of strategy selection and visualization recommendations.

The dataset assembled for this project is both diverse and time-spanning, enabling robust testing of hypotheses around price movement modeling and strategy development. It provides the necessary breadth for generalization and the temporal depth required for training and evaluation. This structured, multi-phase data approach ensures alignment with the project's technical and business goals, particularly around building context-rich visualizations and outperforming traditional candlestick models.

Chapter 3

This chapter presents the modeling approaches undertaken to support our goal of developing a superior financial visualization tool that enhances trading decisions. Our methodology evolved through several iterations, ultimately leading to the implementation of an FP-Growth-based association rule mining framework. The progression from traditional models to FP-Growth reflects the challenges in predicting asset price movement and the need for interpretable, data-driven strategies aligned with our business objectives.

Initial modeling efforts focused on regression and classification. A gradient boosting regression model was applied using differenced NVIDIA price data as the target variable, with input features consisting of lagged differenced values from an industry network. Despite parameter tuning and robust evaluation using median absolute error ($1.31) and root mean squared error ($1.98), the model failed to yield actionable results due to excessive granularity and limited explanatory power.

A subsequent LogisticRegression model was to predict NVDA's binary movement (up/down). Although this classification model achieved a test accuracy of 52.3%, the correlated structure of financial time series data and the coarse granularity of inputs affected its effectiveness. Given these limitations, we transitioned to an association rule mining approach better suited for capturing movement patterns between financial assets.

The primary model implemented in this project is based on the FP-Growth algorithm, executed using PySpark. FP-Growth was selected over Apriori due to its computational efficiency and scalability, which were necessary given the size and complexity of our dataset.

The model operates on a binary-encoded dataset where each row represents a trading day and each column indicates whether a given asset experienced an "up" or "down" movement. The full dataset includes stocks, ETFs, cryptocurrencies, metal futures, and government bonds. The objective is to discover frequent itemsets and association rules where the antecedent is a collection of asset movements, and the consequent is a one-day-ahead movement in NVDA stock.

We applied filtering criteria to ensure the quality and significance of the generated rules:

- **Minimum Confidence**: 0.8, ensuring that predictions have at least 80% historical reliability.
- **Minimum Support**: 0.2, focusing on frequently occurring patterns.
- **Minimum Lift**: 1.0, filtering out rules that lack predictive advantage.
- **Antecedent Size**: Maximum of four, promoting interpretability and simplifying downstream visualization.

This FP-Growth model produced over 10,000 rules, from which we derived high-confidence subsets to construct investment strategies. These association rules form the backbone of our FP-Growth Bot, which makes trade decisions based on rule activation on a given day.

We benchmarked our FP-Growth-based strategy against two baselines: a Buy-and-Hold strategy and a simple Candlestick-based strategy. The Buy-and-Hold portfolio achieved the highest return ($88,096.08), followed by the FP-Growth bot ($63,268.19), and the Candlestick bot ($35,895.73). While Buy-and-Hold yielded superior results over the test period, the FP-Growth bot substantially outperformed the Candlestick strategy—validating our objective of exceeding traditional strategies by 5–15% in decision-making accuracy and potential returns.

Our modeling efforts strongly align with the project's business objectives:

- The FP-Growth model introduces a robust and interpretable strategy tailored to dynamically shifting market conditions.

- By outperforming standard candlestick-based strategies, the FP-Growth bot meets the objective of improving accuracy and investor returns.
- The association rules model, enhanced by entropy-based rule evaluation, effectively models inter-asset relationships, fulfilling the objective of building an industry network that explains asset price variability.

This modeling framework offers a foundation not just for visualizing asset movement but also for building adaptive, data-driven trading strategies that are transparent, testable, and scalable.

Chapter 4

Initial Models

Before the base model is discussed, an overview of models that were initially developed for this task will be covered. At first, XLK was the target asset. Other assets within the same industry, such as AAPL, and key economic indicators, such as VIX were defined as XLK's industry network. To guard against data leakage, the industry network was lagged by values of 1, 3, 7, 30, and 90 days. To help with apparent data granularity problems, both the industry network and the XLK data were differenced. That is, the value of their close price at timestep *t* became the value of the close price at timestep *t - 1*.

xgboost's *XGBRegressor* was the first model used to directly predict the difference in price for XLK using the lagged and differenced industry network data. *XGBRegressor* was believed to be a smart modeling option for the simple fact that it does not make any distributional assumptions and very little assumptions otherwise. Before the first real meeting with the project supervisor for this project, no grid-search cross-validation was done on the *XGBRegressor*. Thus, its default parameters were used. These can be found in xgboost's documentation of the class.[1] To evaluate it, sklearn's *median_absolute_error* and *root_mean_squared_error* functions were used on test data. The results are given below.

Median Absolute Error	Root Mean Squared Error
$1.31	$1.98

[1] https://xgboost.readthedocs.io/en/stable/python/python_api.html#xgboost.XGBRegressor

sklearn's *LogisticRegression* was the second model used. Instead of directly predicting the difference in price for XLK, it was discretized into up/down (whether the close price was higher/lower than the open price) and the lagged and differenced industry network data was used to predict this binary outcome. Unlike *XGBRegressor*, *LogisticRegression* does make significant assumptions about the data. It assumes that there is no multicollinearity between the predictors and the target variable. Though the data certainly violated this assumption, it was not believed to be too egregious to get meaningful results from the model. Similarly to the implementation of the *XGBRegressor*, no grid-search cross-validation was done on the *LogisticRegression*. It, too, had its default parameters used. These can be found in sklearn's documentation of the class.[2] This model was evaluated using sklearn's *accuracy_score* function. It achieved a score of 0.523, which is just a little better than a model which is randomly classifying. The classification report is given below.

	Predicted Up	Predicted Down
Actually Up	110	120
Actually Down	108	140

At the advice of the project supervisor, there was no future to the use of the *LogisticRegression* or the *XGBRegressor*, and both the input data and the output data had to be discretized.

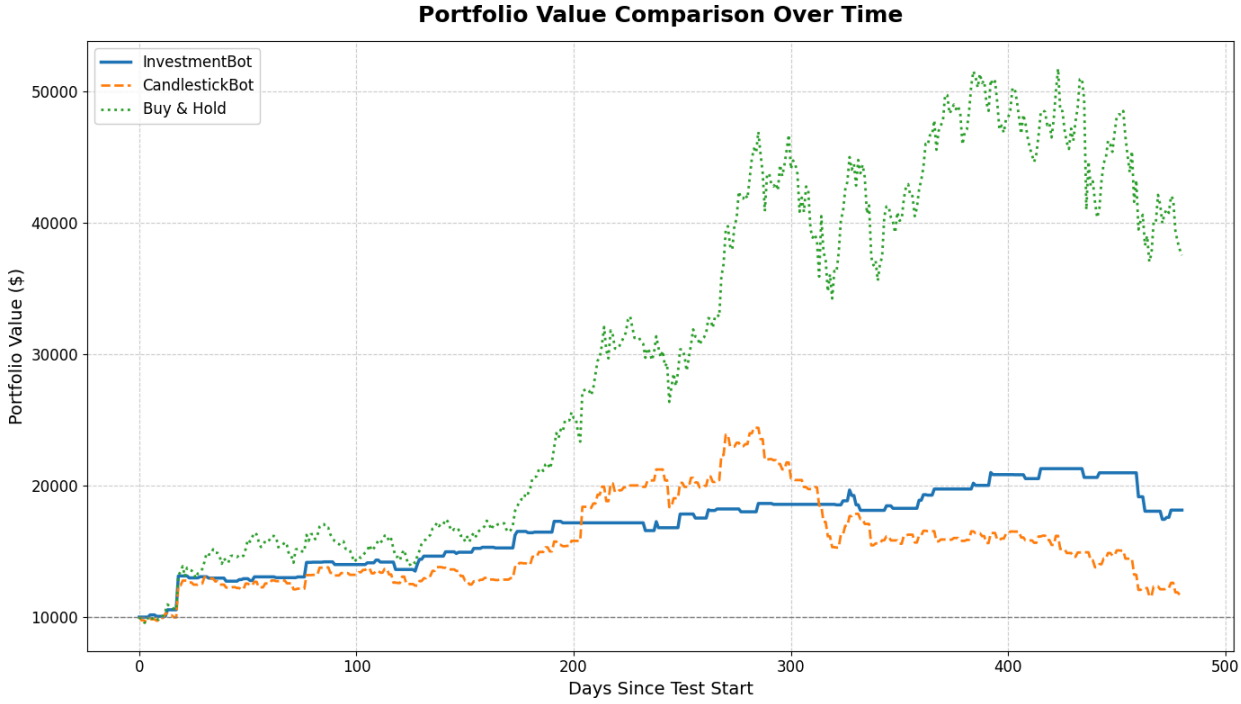
Base Model

Upon the reflection of the team, it was decided that XLK (since it is an index fund containing many assets) was not the best example case for the project. Thus, NVDA became the target asset and the industry network was not changed. The base model used was an Apriori association rule mining model, which (in itself) makes no assumptions about the data. The data was converted into a transactional format (e.g., NVDA_Up and AAPL_Down as binary features for each day), and NVDA was lagged by one day so that the association rules mined would be indicating NVDA's movement the next day. The Apriori model was extraordinarily computationally expensive due to the size of the industry network and, because of this, a relatively high minimum support of 0.3 had to be used. With this, no meaningful results were obtained.

First Improvement

[2] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

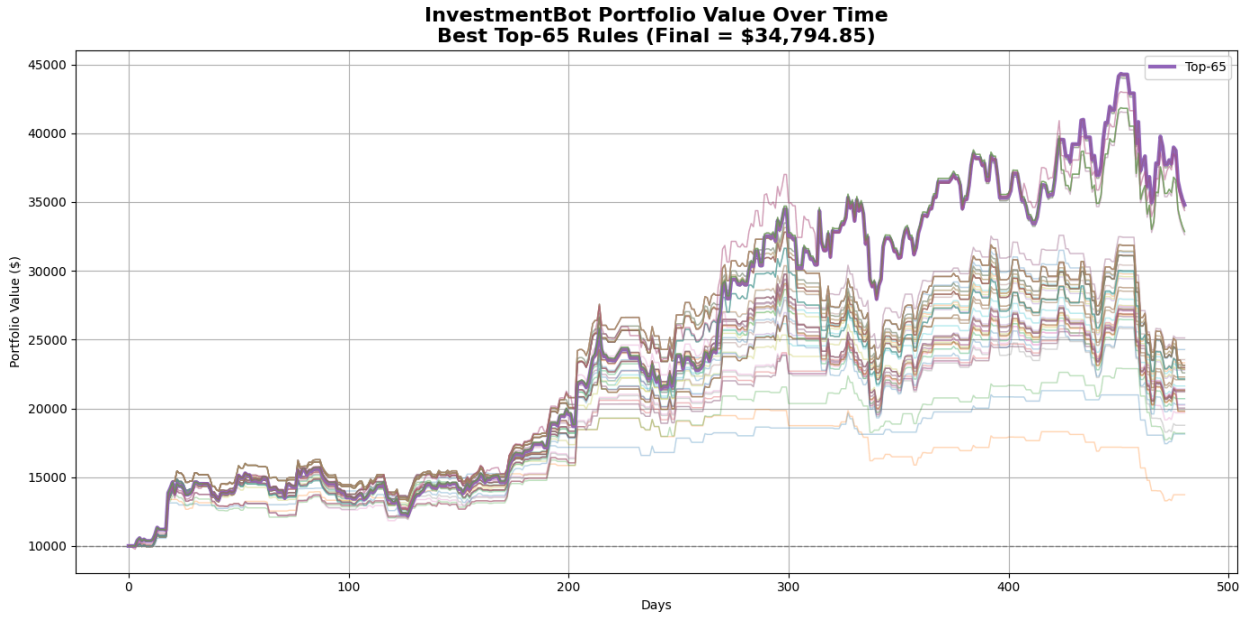
PySpark's *FPGrowth* was used due to its efficiency and speed compared to alternative association rule mining algorithms, like Apriori. It also (in itself) does not make assumptions about the data. To make sure that the association rules obtained were statistically significant, a minimum support of 0.2 and minimum confidence of 0.8 was selected. All rules with a lift under 1 were filtered out. All rules where the antecedent contained more than 4 transactions were filtered out. All rules where the consequent was not NVDA_Up or NVDA_Down were filtered out. Even with these stringent requirements, *FPGrowth* mined over 50,000 buy and sell association rules. This confirmed the relevance of the industry network to NVDA's price movement. An *InvestmentBot* class was manually implemented which is fed association rules generated on train data, and (with a starting cash of $10,000) automatically buys or sells NVDA based on whether the transactions seen in the test data meet the association rule criteria for buying or selling (a buy association rule is triggered or a sell association rule is triggered for that day). It should be noted that the use of this *InvestmentBot* does make an assumption: that the association rules do not become dubious over time. The *InvestmentBot* class was compared to a *CandlestickBot* class which only contained a single rule for buying or selling based on whether the well-defined candlestick pattern of *engulfing* appeared, and a *BuyHoldBot* class which bought NVDA at the first day of the test data and held until the end. The *InvestmentBot* class at first only contained a single association rule. Results are plotted below.



The *InvestmentBot* performs better than the *CandlestickBot* with a single rule, having a final portfolio value of around $18,000 whereas the *CandlestickBot* has a final portfolio value of $11,000. It performs worse than the *BuyHoldBot,* but this is to be expected since day-trading strategies usually never outperform buy & hold strategies.

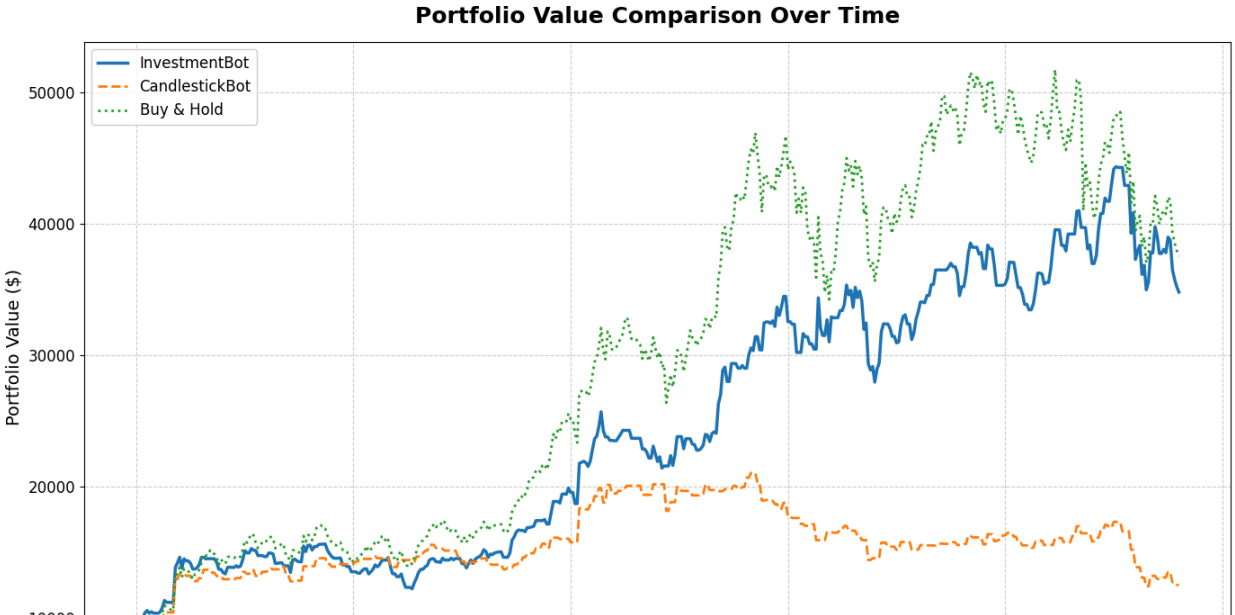
Second/Final Improvement

Since the *InvestmentBot* only takes in one parameter, which is the number of association rules used during the trading process, a grid-search was performed over the number of rules used (from 1 to 75) and the results are plotted below.



The *InvestmentBot* performed best when it was using 65 association rules. More analysis showed that, generally, the more rules used the better since a more well-rounded picture of the industry market is preserved.

This iteration of the *InvestmentBot* was compared to the *CandlestickBot* using 10 different buying and selling rules and the *BuyHoldBot*. Results are plotted below.



The *CandlestickBot* roughly performs the same, with a final portfolio value around $12,000. The *InvestmentBot* has a final portfolio value of nearly $35,000, getting very close to surpassing the *BuyHoldBot*. An interesting result of this graph is that from days 400 to around 430, the *InvestmentBot*'s portfolio value is increasing while the *BuyHoldBot*'s portfolio value is decreasing.

Chapter 5
Restating the Business and Data Mining Objections

In our project proposal, we presented these business objectives:

- Enhance day traders' decision-making accuracy by presenting richer, context-aware price movement visualizations.
- Improve user profitability through integrated insights on underlying variables influencing market volatility.
- Increase trader engagement and retention by delivering a superior alternative to traditional candlestick charts.
- Streamline the trading workflow by consolidating multiple data points into a single, intuitive visualization tool.

And these data mining objectives:

- Identify key underlying factors driving price movements by analyzing historical and real-time market data.
- Model market variables such as volume, news sentiment, and economic indicators to price fluctuations.
- Detect emerging patterns and trends in price movements to anticipate potential market shifts.
- Uncover hidden relationships between multiple data sources to improve predictive accuracy.
- Generate actionable insights for enhanced visualization, enabling traders to make data-driven decisions.

Evaluation of the Model Against the Objectives

Model Summary

An association rule mining model has been developed to achieve the key business and data mining objectives. In order to build a "richer, context-aware price movement visualization" that "improve[d] user profitability" it was decided that an asset's historical data was not the only consideration in whether an investor should buy or sell into that asset, but also the historical price movement of an asset's *industry network* (a set of conceptually related assets). For this

project, the target asset was NVDA and the industry network was composed of assets such as AAPL, TSLA, and GOOG. Once the data had been turned into transactional format, PySpark's *FPGrowth* algorithm was used to mine high-confidence, high-lift association rules between NVDA and its industry network. *FPGrowth* was preferred over the more traditional Apriori algorithm due to its efficiency and scalability in handling large sets of transactional data. An *InvestmentBot* Python class was constructed which, when fed the association rules generated by *FPGrowth*, automatically bought NVDA, sold NVDA, or stayed each day, depending on whether a "buy" signal or "sell" signal was present in the test data. An example of how this would work in practice is given below:

Assume two association rules that have been mined are:

AAPL_Up + GOOG_Down → NVDA_Up
TSLA_Down + AAPL_Down → NVDA_Down

On day *t*,

AAPL has a closing price $5 above its opening price.
GOOG has a closing price $15.50 below its opening price.
TSLA has a closing price $1 above its opening price.

Since AAPL_Up is true and GOOG_Down is true, a "buy" signal has been indicated. Thus, on day *t+1*, the *InvestmentBot* will put all of its available cash into NVDA.

On day *t+1*,

AAPL has a closing price $0.36 below its opening price.
GOOG has a closing price $3.10 above its opening price.
TSLA has a closing price $4 below its opening price.

Since TSLA_Down is true and AAPL_Down is true, a "sell" signal has been indicated. Thus, on day *t+2*, the *InvestmentBot* will sell all of its NVDA for cash.

On day *t+2*,

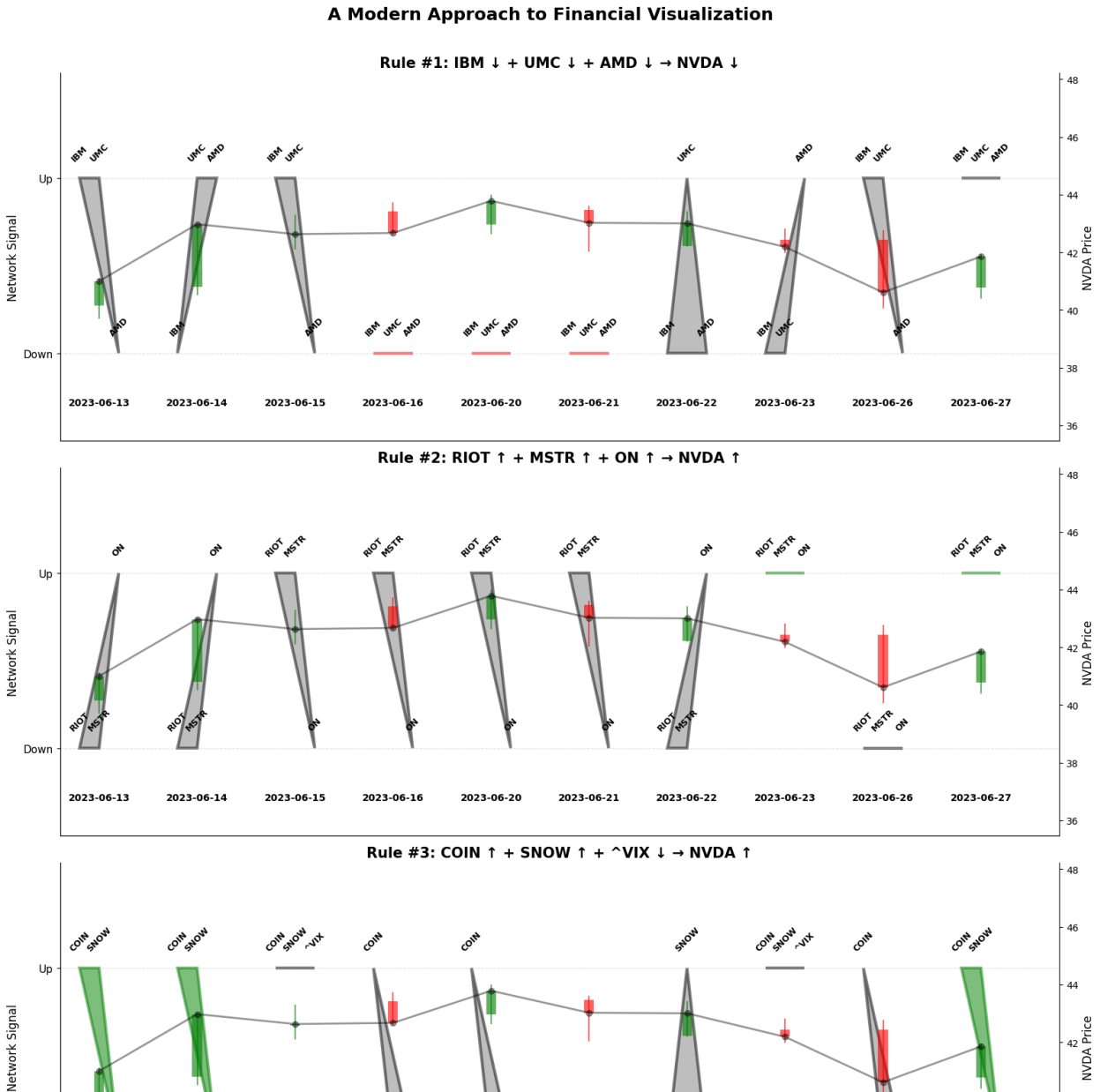
AAPL has a closing price $1 above its opening price.
GOOG has a closing price $0.15 above its opening price.
TSLA has a closing price $0.50 above its opening price.

Neither a “sell” signal or a “buy” signal have been indicated. Thus, on day *t*+*3*, the *InvestmentBot* will not invest anything into NVDA.

Evaluation Against the Business Objectives

- Enhance day traders’ decision-making accuracy by presenting richer, context-aware price movement visualizations.

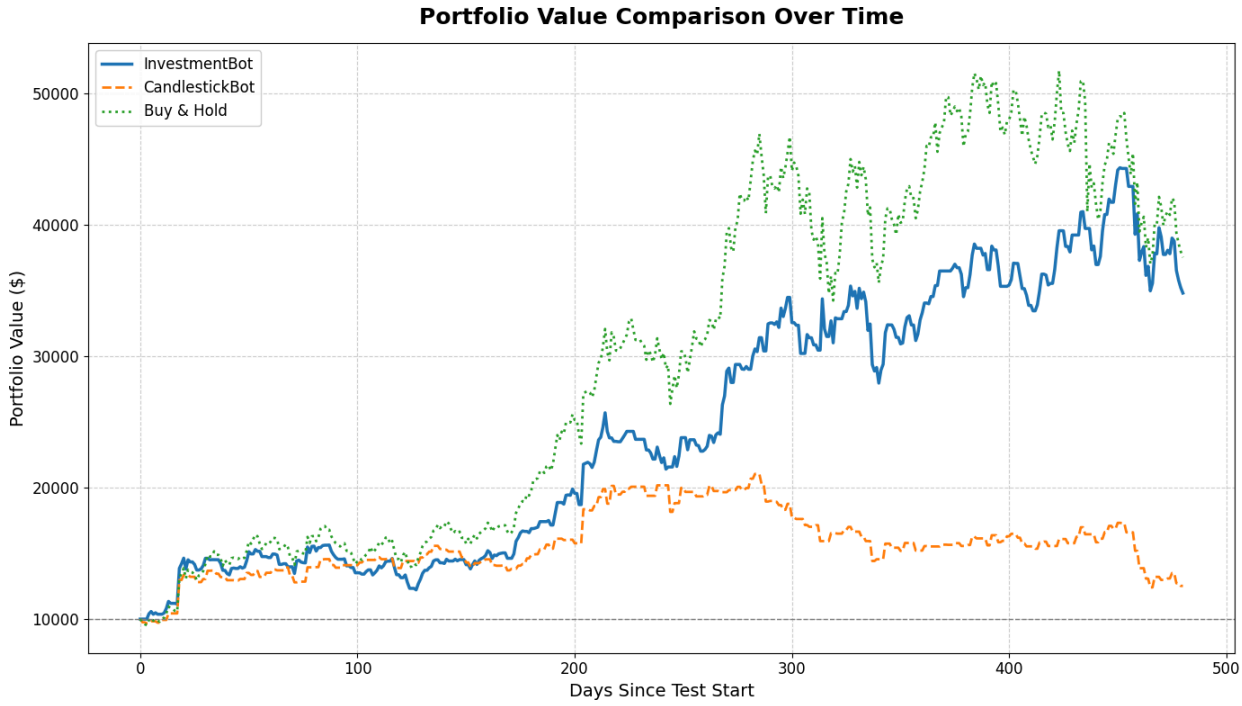
A fitting visualization was derived from considering the latent geometry naturally present in association rules. An example is given below:



For each association rule a user would like to use, a plot is generated over the past *N* days which tracks both NVDA's price movement (as a candlestick chart in the center of each visualization) and the binary price movement of each asset in the association rule. When a "sell" signal is indicated, the polygon is colored red. When a "buy" signal is indicated, the polygon is colored green. When neither is indicated, the polygon is colored grey. This visualization truly is context-aware and better than the anemic picture given by traditional candlestick strategies.

- Improve user profitability through integrated insights on underlying variables influencing market volatility.

This project assumes that most of the market volatility that would influence an asset's price is volatility that is within the asset's industry network. Testing on the *InvestmentBot* showed that this is a reasonable assumption. The *InvestmentBot* was compared to a buy-and-hold trading strategy and a traditional candlestick trading strategy, and the results are given below:



With an initial balance of $10,000, the *InvestmentBot* has a final return of nearly $35,000, the *BuyHoldBot* has a final return of around $38,000, and the *CandlestickBot* has a final return of just about $12,000. A user faithfully investing or selling according to the visualization would far outperform a user faithfully investing based on some of the most common candlestick patterns.

- Increase trader engagement and retention by delivering a superior alternative to traditional candlestick charts.

Traders are more likely to use this visualization if, when faithfully used, it will outperform traditional candlestick charts in terms of returns.

- Streamline the trading workflow by consolidating multiple data points into a single, intuitive visualization tool.

As demonstrated in the example visualization given at the beginning of this section, this visualization does consolidate multiple data sources into a single, intuitive visualization tool.

Evaluation Against the Data Mining Objectives

- Identify key underlying factors driving price movements by analyzing historical and real-time market data.

The model did identify the importance that an industry network's price movement has to the price movement of any asset within it. Whether it is deficient in regard to the use of "real-time" market data is up to interpretation. Everything can be synced to the most recent day of data, but this project does not make use of minute-by-minute or hour-by-hour data, since generating association rules at this level of granularity would be incredibly computationally expensive.

- Model market variables such as volume, news sentiment, and economic indicators to price fluctuations.

The model did not make use of volume or news sentiment since using the industry network was sufficient, and it was not easy to see how to integrate these other things. It did make use of macroeconomic indicators though, since part of the industry network is these macroeconomic indicators.

- Detect emerging patterns and trends in price movements to anticipate potential market shifts.

The model is able to effectively detect emerging patterns and trends by retraining the *FPGrowth* algorithm on the most recent data. Since the association rule mining is done on 1-day lagged NVDA price movement, the model is basically using the association rules to predict whether NVDA will be up or down the next day (short-term market shifts).

- Uncover hidden relationships between multiple data sources to improve predictive accuracy.

This is by definition association rule mining, so the model certainly achieved this data mining objective.

- Generate actionable insights for enhanced visualization, enabling traders to make data-driven decisions.

As was stated earlier, the association rules that were mined were the “actionable insights” that were put into visualization form, allowing traders to make “data-driven decisions” with regard to whether they’d like to buy or sell their assets, like NVDA.

Evaluating Other Findings

Most of the significant findings which emerged during the data mining process were found during exploration into using entropy for determining association rule confidence and asset volatility.

Observations of the Data

When observing the data, it was quickly realized that nonparametric methods were needed when working with probabilistic models (entropy). The variable used for these distributions was the rate of returns (ROR). Upon observation of the RORs it was found that over the course of 252 days (1 business year), RORs followed a normal distribution. However, when observing more granular time periods like quarters, months, and weeks, distributions were often skewed. To make high frequency trading decisions it is important to balance what information is currently relevant to a probability distribution.

Findings from Supervised Learning

When creating an LSTM it was difficult to determine what the input and output should be. The goal was to find the optimal distribution for entropy of an asset. This goal provided a vague criteria for what needed to be found in the data. Ultimately a combination of different time windows provided the LSTM with a “supervised” variable where it could see the performance of multiple different time windows and weigh them accordingly.

Conclusion

In its current state, this project has met all of the initial business objectives but not all of the initial data mining objectives. While a simple, profitable visualization tool (which can be generalized to other assets) has been created, satisfying the business objectives, the model does not make use of more granular data or non-financial features like news sentiment. Admittedly, some of the objectives had been set before the team realized how difficult generating association rules on high granularity data or incorporating news sentiment into the project would be. It does not seem like the team will be able to meet this specific goal. Thankfully, most of the data

mining objectives were completed: the model was able to find the underlying factors influencing an asset's price movement and generate actionable insights for them based on the association rules mined by it.

Future Work

While we were able to generalize our findings to other industries in the stock market, the immediate next step for our project would be to start testing our model on a more diverse set of industries. Additionally, finding an effective way to update our parameters/rules could provide more confident results to users. This could be an extension of the entropy model that we used to update our parameters. A more refined version of this entropy model in conjunction with apriori rule mining would most likely be the direction we would look for rule updating. Use of attention models would also help reduce bias in our findings as well. Instead of manually choosing time windows for backtesting we could use an LSTM or Transformer model to computationally determine a time window. Finding a reliable method for getting more granular results would also provide users with a more diverse range of trading window options. Finally, implementing a live visualization based on the still images we created would be the final step of tying together a product that is ready for users.

Bibliography

Leonardo N. Ferreira, Liang Zhao. *A Time Series Clustering Technique based on Community Detection in Networks*. Procedia Computer Science. Volume 53, 2015, Pages 183-190. ISSN 1877-0509. https://doi.org/10.1016/j.procs.2015.07.293.

Autoregressive Moving Average ARMA(p, q) Models for Time Series Analysis - Part 3 QuantStart. (n.d.). https://www.quantstart.com/articles/Autoregressive-Moving-Average-ARMA-p-q-Models-for-Time-Series-Analysis-Part-3/

Zhang, Yifan. "Stock Price Prediction Method Based on XGboost Algorithm." Proceedings of the 2022 International Conference on Bigdata, Blockchain and Economy Management

(ICBBEM 2022), edited by D. Qiu et al., Atlantis Press, 2023, pp. 595–603. Advances in Humanities and Social Sciences, vol. 5. https://doi.org/10.2991/978-94-6463-030-5_60.

Quant Club, IIT Kharagpur. "Entropy and Its Application in Stock Market." Medium, 13 Mar. 2024, https://medium.com/@quantclubiitkgp/entropy-and-its-application-in-stock-market-accd880ef241.

GunKurnia. "Apriori Algorithm: Unlocking Hidden Patterns and Driving Sales with Product Bundling." Medium, 14 Sept. 2024, https://medium.com/@gunkurnia/apriori-algorithm-unlocking-hidden-patterns-and-driving-sales-with-product-bundling-e94ba7636195.