EQUITY EDUCATION - TRACK 2 PROTOTYPE

Joey Squillaci

ABSTRACT

Many beginner traders are exposed to large amounts of financial data and technical indicators, but often lack the experience needed to interpret these signals in a meaningful way. This project presents an educational equity analysis prototype that combines machine learning-based forecasting with an interactive GUI to help users better understand stock market behavior. The system uses historical equity data, engineered technical indicators, and a configurable long short-term memory (LSTM) model to generate predictions of future price movement. These predictions are presented through a Python-based GUI that also visualizes historical price trends, moving averages, support and resistance levels, and training performance information. In addition to supporting forecasting, the prototype is designed to promote financial education by connecting model outputs to easy to read, interpretable context. Initial evaluation is performed on examples with different volatility characteristics, including SPY and AAPL, in order to study directional prediction behavior across multiple trading days. The results and prediction content suggest that the prototype can serve as a useful educational decision-support tool for novice traders, while also highlighting the challenges of applying machine learning methods to financial markets. Project code can be found on Github: https://github.com/joeysquillaci/ECE-57000-Artificial-Intelligence-Final-Project.

1 INTRODUCTION

Financial markets generate large volumes of time-series data that are difficult for novice traders to interpret. While many trading platforms provide technical indicators such as relative strength index (RSI), moving averages, and volatility metrics, these signals often require significant experience to translate into meaningful insights. As a result, beginners frequently struggle to connect raw indicators with actionable understanding of market behavior.

To address this challenge, we develop a prototype educational forecasting system that combines machine learning predictions with visual explanations. The system integrates a sequential prediction model with an interactive GUI that allows users to explore predictions and supporting indicators.

The main contributions of this work include:

- A configurable LSTM-based forecasting model for short, medium, and long term equity movement prediction.
- An interactive educational GUI that visualizes predictions alongside technical indicators.
- An integrated training and evaluation process designed to support experimentation and data interpretation.

2 PROBLEM DEFINITION AND USER REQUIREMENTS

Many beginner traders rely on simplified tools that expose raw market indicators without explaining their relationship to market behavior. This creates a barrier to understanding, as users must interpret multiple technical metrics simultaneously without guidance on how they relate to future price movement. The problem addressed in this work is the lack of accessible tools that translate financial indicators and predictive models into interpretable insights for novice users.

From the perspective of the target user, the system must satisfy several key requirements:

- Provide interpretable predictions that help to provide context to model outputs with supporting indicators.

- Allow for configurable training parameters so users can experiment with different model settings.

- Present results through clear visualizations that connect historical price behavior with predictions.

- Offer a simple interface that enables a level of machine learning experimentation without requiring advanced technical knowledge.

3 Related Work and Technology Selection

Many approaches have been used to try to predict movements in financial markets. Traditional methods often rely on statistical models such as ARIMA or simple regression techniques that analyze historical price trends. In recent years, machine learning and deep learning models have also been applied to financial forecasting tasks. These models can learn patterns from historical data and can capture relationships that are difficult to model with traditional statistical techniques.

Some approaches include recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, which are commonly used for time-series data because of how they are designed to process sequential inputs. Since stock prices naturally form a sequence over time, LSTM models are a practical choice for exploring price prediction using time-based data.

A particularly relevant example is the work of Fischer and Krauss, who applied LSTM networks to predict directional movements for S&P 500-based stocks over a long historical period (1). Their study showed that LSTMs can be effectively used for large-scale financial market prediction and provided evidence that sequence-based deep learning models can outperform several memory-free baseline models. This is important for the project prototype because it supports the use of LSTMs as a reasonable modeling choice for equity movement prediction, especially when the objective is to capture patterns that may not be well represented by simpler models.

For this project, an LSTM-based model was selected as the core forecasting method. In addition, a python GUI was developed to make the system easier to use and interpret. The goal of the interface is not only to display predictions, but also to help users understand how technical indicators and historical price behavior relate to the model's predictions. In that sense, this project differs from prior work such as Fischer and Krauss by focusing not only on the predictive model itself, but also on the educational presentation of the model's outputs to novice users.

4 System Overview and Architecture

The system consists of a process that transforms historical financial data into easy to understand, actionable insights for the user. The workflow begins by initiating the model training by gathering market data via the Yahoo Finance API, which is then followed by feature and sequence construction for model training. The trained LSTM model generates predictions that are visualized through the GUI, while the Education tab supports further user exploration and learning.

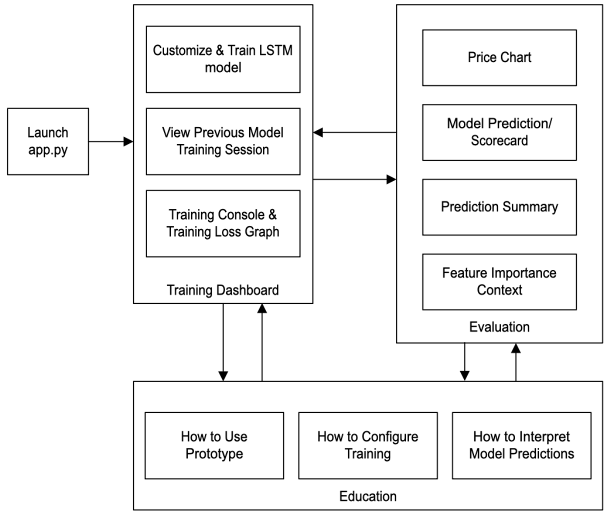

Figure 1: UI/UX System Layout.

5 Data and Feature Engineering

Historical market data is obtained through the Yahoo Finance API that provides open, high, low, close, and volume (OHLCV) values for each day. Users can select different tickers and time ranges, allowing the system to construct datasets tailored to different forecasting horizons.

To capture meaningful aspects of market behavior, several categories of engineered features are derived from the raw data. Momentum indicators such as RSI and MACD capture directional trends in price movement. Volatility-based features describe short-term fluctuations in returns, while activity metrics such as volume ratios provide insight into market participation. Together, these features provide a deeper representation of market conditions for the forecasting model.

6 Model and Training Methodology

The forecasting component of the system uses a long short-term memory (LSTM) network to model temporal patterns in financial data. The input features outlined in the "Data and Feature Engineering" section are organized into sequential windows that allow the model to observe a rolling history of market conditions before making a prediction. This sequential representation enables the model to capture dependencies between historical market behavior and future price movement.

Training is performed using configurable hyperparameters such as learning rate, number of epochs, training split, loss type, and sequence lookback length. Regularization techniques include dropout and early stopping, which are used to reduce overfitting and stabilize the training process. These safeguards help to ensure that the model remains robust while allowing users to experiment with different training configurations.

7 Prototype Implementation and User Workflow

The prediction system is implemented as a python-based prototype with a tkinter GUI that integrates model training, evaluation, and visualization. Through the interface, users can configure training parameters, select a financial instrument, and initiate the training process without interacting directly with the underlying code.

The user workflow begins by selecting configuration parameters; this can be either the default (short, medium, or long term) horizons or a custom configuration, and then initiating model training. Once training is complete, the system presents evaluation metrics and visualizations of historical prices alongside predicted market movement. This workflow allows users to explore how different training configurations affect model performance while maintaining a clear connection between predictions and historical market context.

The user experience is organized around three main views: Training Dashboard, Evaluation, and Education. In the Training Dashboard, the user selects either a preconfigured short, medium, or long-term model setup, or customizes individual hyperparameters before starting training. During this process, the interface displays a real-time training console and loss graph so that users can monitor model progress and review the most recent training session.

After training is completed, the prototype transitions to the Evaluation view, where the user is presented with the current price, price change from the previous close, predicted move, and directional accuracy. Historical price behavior is also visualized through an interactive graph that can display different time ranges, moving averages, and support/resistance levels. This allows users to connect the model's output to recent market context rather than viewing the prediction in isolation.

The Education view is intended to reinforce the prototype's instructional purpose. In this tab, users can explore supporting explanations about technical indicators, forecasting context, and general information related to the tool's operation. Together, these three views provide a complete workflow in which the user can configure a model, observe the training process, evaluate the resulting prediction, and better understand the reasoning context surrounding the output.

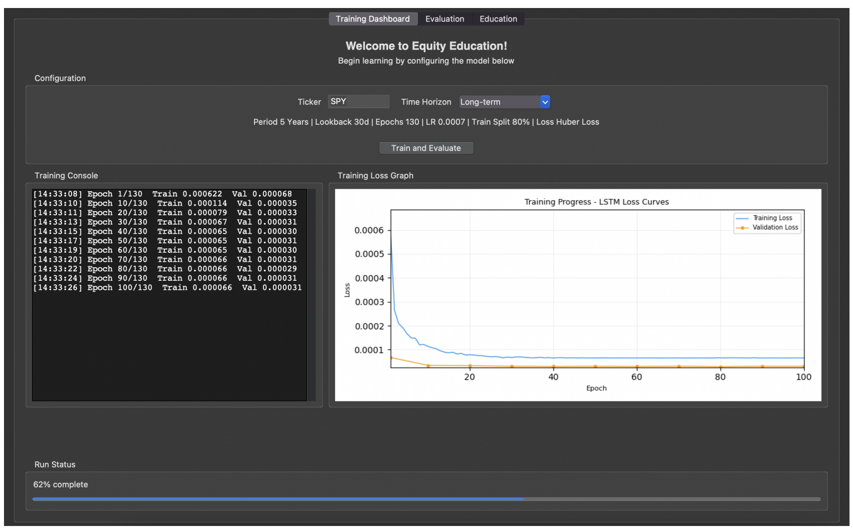

Figure 2: Example of training the LSTM model on a long term configuration for SPY.

8 Evaluation and Results

To evaluate the functionality of the prototype, experiments were conducted each day over a one week time frame for a selected set of equities (SPY, AAPL) over a long-term training configuration (period: 5 years, lookback: 30 days, epochs: 130). The tables below detail the close prices of the equities over the five days the market was open for the experiment (4/6/26 - 4/10/26).

SPY is an ETF that tracks the S&P 500. Because it is diversified, the price tends to move more steadily and is generally less volatile. AAPL (Apple) is a single-company stock, so its price depends only on that company's performance and market expectations. Because of this, Apple's stock may experience larger and faster price swings, making it more volatile than SPY. The evaluation of these two equities gives insight into how the model reacts to more or less volatile price trends.

	Predicted Move	Actual Move	Predicated Directional Accuracy	Direction Correct?
Day 1	+0.1%	+0.4%	52.9%	Y
Day 2	+0.116%	+2.55%	53.7%	Y
Day 3	+0.08%	0.58%	53.7%	Y
Day 4	+0.22%	-0.07%	54.5%	N
Day 5	+0.08%	+0.98%	53.7%	Y

Table 1: One Week Evaluation of Model, Tracking SPY

	Predicted Move	Actual Move	Predicted Directional Accuracy	Direction Correct?
Day 1	+0.2%	-2.07%	52.9%	N
Day 2	+0.2294%	+2.13%	52.9%	Y
Day 3	+0.12%	+0.61%	52.9%	Y
Day 4	+0.18%	0.00%	52.9%	Y
Day 5	+0.15%	-0.49%	52.1%	N

Table 2: One Week Evaluation of Model, Tracking AAPL

Evaluation focuses on directional accuracy, which measures whether the model correctly predicts the direction of price movement. Across the one-week window, the model produced consistently small positive forecasts for both SPY and AAPL, while the true market moves varied much more widely. This suggests that the prototype learned a relatively stable upward bias from the historical data, but it was less successful at capturing the magnitude of short-term daily swings.

The SPY results are more consistent with the intended behavior of the system. Although the predicted daily moves are much smaller than the realized returns, the model generally tracks SPY as a more stable instrument and maintains directional accuracy values in the low-to-mid 50% range. In contrast, the AAPL results illustrate the difficulty of applying the same model structure to a more reactive single-company stock. The predictions remain narrowly positive even when actual movement becomes negative or flat, showing that the system has more difficulty adapting to rapid day-to-day changes in price behavior.

Overall, these results suggest that the prototype is more effective as an interpretive and exploratory tool than as a high-confidence forecasting system. Even when predictions do not align closely with realized returns, the GUI remains useful because it allows users to compare model output against market behavior and better understand where a data-driven forecasting approach succeeds or fails.

9 Design Decisions and Tradeoffs

Several design decisions influenced the development of the system. An LSTM architecture was selected because it can capture sequential patterns in financial time-series data, though this also increases model complexity. A tkinter GUI was chosen to provide a simple environment where users can experiment with forecasting models without requiring complex infrastructure. This allowed the frontend and backend to run smoothly in python, though it also meant that the interface was less responsive and visually polished than alternatives such as Javascript-based frameworks.

While more complex architectures may improve predictive performance, they can also reduce interpretability. For that reason, the system prioritizes transparency and educational value over maximizing predictive accuracy. The prototype is intended to help users understand the context of an underlying's price movement without requiring them to interpret raw technical metrics themselves.

10 Limitations and Future Work

Financial markets are highly dynamic and influenced by factors that are difficult to capture in a single predictive model. As a result, the prototype should be interpreted as an educational tool rather than a definitive trading system. Its purpose is not to produce winning trades, but to give users more context for understanding market behavior.

The evaluation results reinforce this limitation. In both SPY and AAPL, the LSTM generated smooth and interpretable outputs, but it did not fully reflect the magnitude or abrupt direction changes of the real market. This suggests that historical price and indicator data contain useful structure, but do not fully describe the broader market environment.

Future improvements should move beyond relying on one model trained only on past price behavior. A more robust system would likely combine multiple model types, broader feature sources, and stronger validation procedures. Possible extensions include market-wide indicators, news or sentiment inputs, ensemble methods, regime-sensitive modeling, and improved interpretability tools.

11 Conclusion

This work presents a prototyped tool that integrates LSTM-based machine learning with an educational visualization interface for stock market and equity analysis. By combining LSTM techniques with an easy to use GUI, the system enables users to explore financial modeling predictions while gaining a deeper understanding of how their equity of interest has been moving relative to a selected time frame.

The evaluation shows that the prototype can generate stable and interpretable prediction outputs, particularly for a lower-volatility asset such as SPY, but it also highlights the limitations of applying a single historical-sequence model to more unpredictable market behavior. In particular, the contrast between SPY and AAPL demonstrates that real market movement can vary much more sharply than the model's forecasts, especially for individual stocks that are more sensitive to fast-changing external conditions.

The prototype therefore succeeds most clearly as an educational and decision-support tool rather than as a standalone predictive trading engine. Its strongest contribution is the combination of forecasting, visualization, and user-facing explanation in a single accessible workflow. At the same time, the observed gaps between predicted and actual movement make clear that meaningful financial forecasting requires a deeper and more comprehensive system than one LSTM trained only on historical data. This project provides a useful prototype foundation for that broader direction while also helping novice users engage with market data in a more informed and structured way.

References

[1] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

A Appendix

A.1 LLM Acknowledgement

LLMs were utilized during the development of the prototype with the following explicit purposes: understanding and modification of GUI frameworks, reformatting/cleaning up of code (unused and cluttered code), generalized commenting of code, generation assistance of documentation, and lastly to use for creative refinement of the project. LLMs were not relied upon for the core LSTM model used in the prototype. The LSTM model was developed by the author and customized to fit the format of the project and the GUI. Below is an itemized list of contributions/acknowledgements from LLMs:

- Developing GUI framework (utilizing and customizing Tkinter windows and widgets)
- Use of creative recommendations for refinement of UI/UX design
- Debugging support, formatting/cleaning up code blocks and comments
- Documentation; converting report from docx to latex, assistance with README.md