ML Stock Predictor (Proposal)

Rishi Patel, Baturalp Unlu, Dhruv Chaudhari, Michael Tang, Adrian Kalisz
Georgia Institute of Technology

GitHub Repo

Introduction and Background

The stock market reflects a country's economic health and provides valuable insights into its future. While it's impossible to predict a stock's exact movement with absolute certainty, we can reduce risk and forecast potential stock movements by analyzing key indicators such as market sentiment, GDP, and earnings reports.

Due to the time-consuming process of analyzing data, algorithmic trading has emerged as an effective measure to analyze the market more safely and efficiently. For this project, We will use FRED economic data, which contains U.S. macroeconomic data, Yahoo Finance, which contains useful APIs for daily market coverage, and FINRA's datasets to develop a profitable investing strategy that helps us predict the direction of the stock market.

Yahoo Finance Stock Market Dataset

FRED Dataset

SEC Data

Problem Definition

Predicting the direction of a stock is challenging due to its volatility. Analyzing data from various key indicators for a stock is also time-consuming. To streamline this process and improve market predictions, we will use machine learning models to forecast stock trends and maximize profit. By leveraging historical data and advanced algorithms, these models can identify patterns that may be difficult for humans to detect manually.

Methods

We anticipate that the data may have imperfections, which we plan to address using the following techniques:

- For columns with missing data, we will use K-Nearest Neighbors (KNN) to analyze how similar data behaves and substitute the missing values accordingly.
- To reduce the number of features and simplify the model, we will apply Principal Component Analysis (PCA), allowing us to drop features while retaining most of the variance [1].
- We will remove outliers to prevent the model from being skewed by unrepresentative data. Specifically, we aim to remove data points falling below and above 2x the interquartile range (IQR).

For our stock market predictor, we will use models such as Support Vector Machine (SVM), Convolutional Neural Networks (CNN), and Linear regression. However, the performance of the project depends largely on how well we select and prepare the data.

- SVM is effective for predicting rapid price changes in complex stock markets, as it excels in handling high-dimensional data and classifying intricate patterns. It is also commonly used to predict price movements [2].
- CNN, traditionally used for image and pattern recognition, can be adapted for stock market prediction by treating past price data as a two-dimensional system. CNN is effective in revealing subtle patterns in stock data through its many technical indicators. Unlike SVM, CNN provides visual insights into non-linear relationships [3].

- Linear Regression is one of the simplest models used in stock market prediction. It works best when the relationship between input features (e.g., past prices, volume) and output features (e.g., expected future prices) is linear [4]. While it allows for quick interpretation of simpler, linear data, more complex datasets may benefit from using SVM or CNN.

Midterm Checkpoint Methods Implementation

We have implemented Ridge Regression as our supervised learning model, using historical stock data from Yahoo Finance via the yfinance library in Python. This model captures trends over time by establishing a linear relationship between input features (e.g., past prices, volume) and output features (e.g., expected future prices). Ridge Regression's regularization penalizes large coefficients, helping to reduce overfitting. Additionally, we standardize the input features to have a mean of 0 and a standard deviation of 1.

To enhance the model, we have expanded the dimensionality of the data by raising features to different powers, ranging from no expansion to powers up to 4 (i.e., x, x^2, x^3, x^4). We also standardize these expanded features to maintain the significance of the regularization term and to prevent the matrix $(X^TX + \lambda(I))$ from becoming ill-conditioned. Missing data is handled with linear interpolation, using the two closest points in the data frame to estimate missing values under the assumption of a linear relationship.

Final Report Methods Implementation

Initially, we planned to implement SVM for stock price classification. However, as our project evolved, we determined that predicting continuous stock prices would benefit more from SVR. SVR builds upon SVM by adapting its principles to regression tasks and allows to predict precise outcomes instead of categorical classifications. SVR was implemented to handle the high-dimensional nature of stock market data and provide precise predictions for rapid price changes. By leveraging its ability to classify non-linear relationships, the model performs exceptionally and achieves an accuracy up to 98% when applied to historical stock data for the ticker $AAPL. The exceptional accuracy of the model reflects on its ability to handle high-dimensional

data in predicting rapid price changes. Furthermore, the extension to higher dimensions with data standardization was also implemented in this model. It did not yield as much of an improvement as it did with the Ridge Regression, but an improvement was still found.

Traditionally used for image recognition, CNN was adapted for stock market prediction by treating historical stock data as a two-dimensional representation. This allowed the model to identify non-linear relationships in the data and enhance the prediction accuracy. CNNs are uniquely suited for capturing complex patterns that may not be apparent with traditional methods such as SVM or Ridge Regression. The implemented CNN model consists of 1D convolution followed by two Dense layers, which proved sufficient for this task.

Potential Results & Discussion

To evaluate our model, we will use metrics from scikit-learn, including Mean-Squared-Error (MSE), which measures the average squared difference between predicted and actual prices, Mean-Absolute-Error (MAE), which measures the average error in prediction, and the R^2 Score, which measures how well the model is likely to predict unseen samples. Our project goal is to develop a stock prediction model that accurately forecasts prices using real-time data, aiming for a success rate of 80%. Specifically, we are targeting an MSE of less than $5, an MAE of less than $3, and an R^2 score greater than 0.8.

Midterm Checkpoint Progress

Currently, the team has run the model on Amazon and Tesla stock data from the past 20 years. For Tesla, the model achieved approximately 72% accuracy, likely due to Tesla's rapid growth in more recent years, which makes older data less predictive. In contrast, the Amazon model nearly reached the 90% accuracy benchmark. Expanding into higher dimensions had the most notable impact; initially, both ridge and linear regression models performed similarly, with accuracy between 60% and 67%. By increasing the polynomial degree, we improved accuracy, with the optimal performance typically occurring around degrees 7 to 8, as

higher dimensions tended to lead to overfitting. We evaluated accuracy using the R^2 scores on the test data.

While it is challenging to visualize metrics from higher-dimensional spaces in two dimensions, visualizations still illustrate how the shape changes with each polynomial degree, roughly aligning with the stock value distribution over time and explaining the improvements in R^2 scores. The model's performance was limited by the lack of additional data pre-processing techniques and by the model's simplicity. Although extending features to higher dimensions significantly improved accuracy, scaling for proper ridge regression regularization may have constrained further score improvements.

Potential solutions include:

- Testing linear regression with higher-degree features without standardizing the inputs
- Exploring alternative algorithms, as outlined in the methods section

Quantitative Metrics

The following statistics are for stock ticker $TSLA, using the highest achieved average R^2 score.

1. Support Vector Regression (SVR)
    - MSE: 65.503
    - MAE: 4.847
    - R^2 Score: 0.984
2. Convolutional Neural Network (CNN)
    - MSE: 571.693
    - MAE: 11.629
    - R^2 Score: 0.948
3. Ridge Regression
    - MSE: 2818.539

- MAE: 39.245
- R^2 Score: 0.730

Analysis and Comparison of the 3 Algorithms

Based on the metrics, the SVR model appeared to explain 98.4% of the variance (R^2 = 0.984) in the stock price data. The relatively low MSE and MAE indicate that the model's predictions are close to the actual values. This is a realistic outcome, especially with the data being preprocessed well and relatively smooth (e.g., without significant outliers or large volatility).

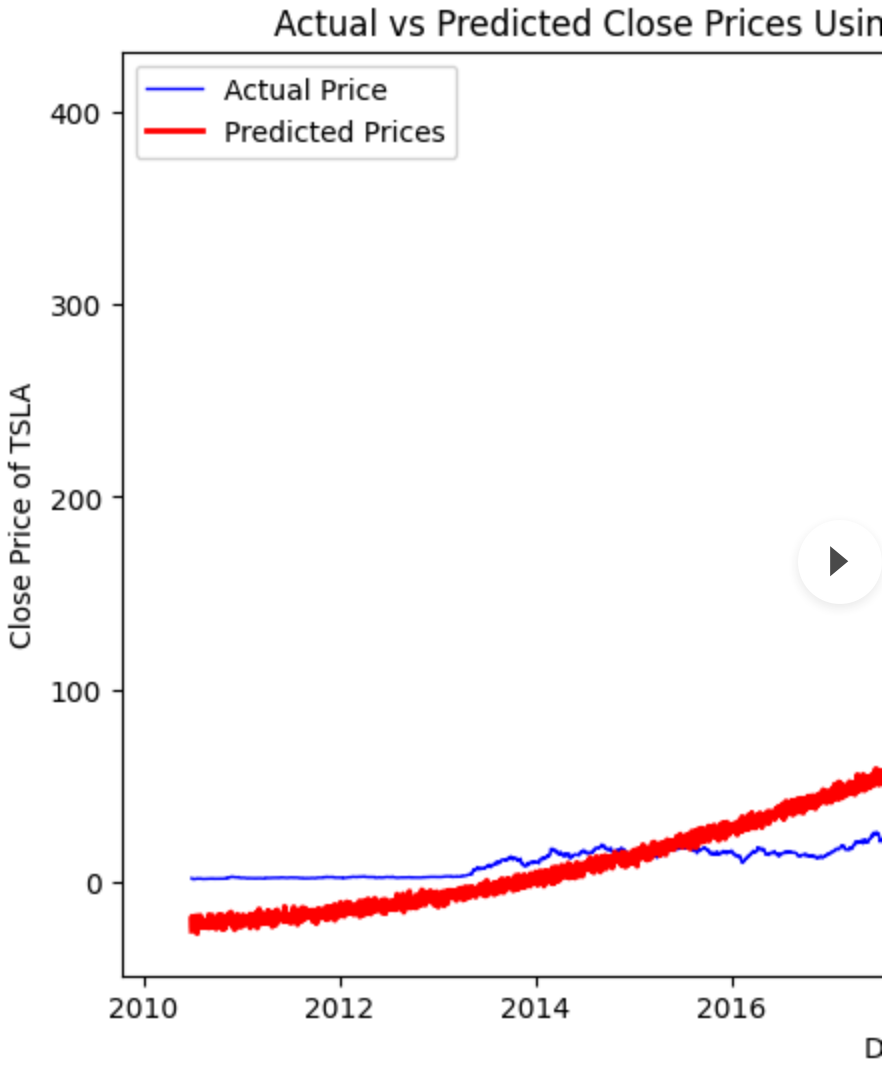
While CNN's R^2 score is slightly lower than SVR, the model still performs well overall. The higher MSE and MAE suggest that it is less accurate than SVR on average, but not by a significant margin. Since this is a standalone CNN, that doesn't utilize time-series preprocessing or hybrid architectures (e.g., CNN-LSTMs), the performance is as expected.

Ridge Regression struggles compared to the other models. The much higher MSE and MAE, combined with a lower R^2 score, suggest that it cannot capture the complex relationships in the data as effectively as SVR or CNN. This is expected because Ridge Regression is a linear model, and stock price movements are rarely linear. However, it can still serve as a good baseline for comparison.

Next Steps

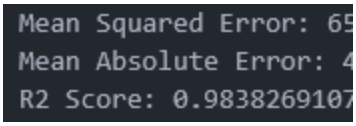
We can refine the data preprocessing methods by normalizing or scaling the features and potentially using returns or log-returns as the target variable instead of raw prices, which can mitigate issues with wide dynamic ranges. For the SVR model, fine-tuning key hyperparameters, such as the kernel type and regularization parameter, may further improve performance. For the CNN model, we can consider hybrid models such as CNN-LSTMs if the data is time-series-based, or the inclusion of temporal features like moving averages, lagged variables, or Fourier-transformed components. We can also revisit Ridge Regression

with advanced feature engineering to better capture any linear relationships present in the data. These steps collectively aim to improve our models' overall predictive capabilities.



Stock price prediction for Te

Quantitative metrics for C



Quantitative metrics for S

References

[1] M. Ghorbani and E. K. P. Chong, "Stock price prediction using principal components," PLoS ONE, vol. 15, no. 3, p. e0230124, 2020, doi: 10.1371/journal.pone.0230124.

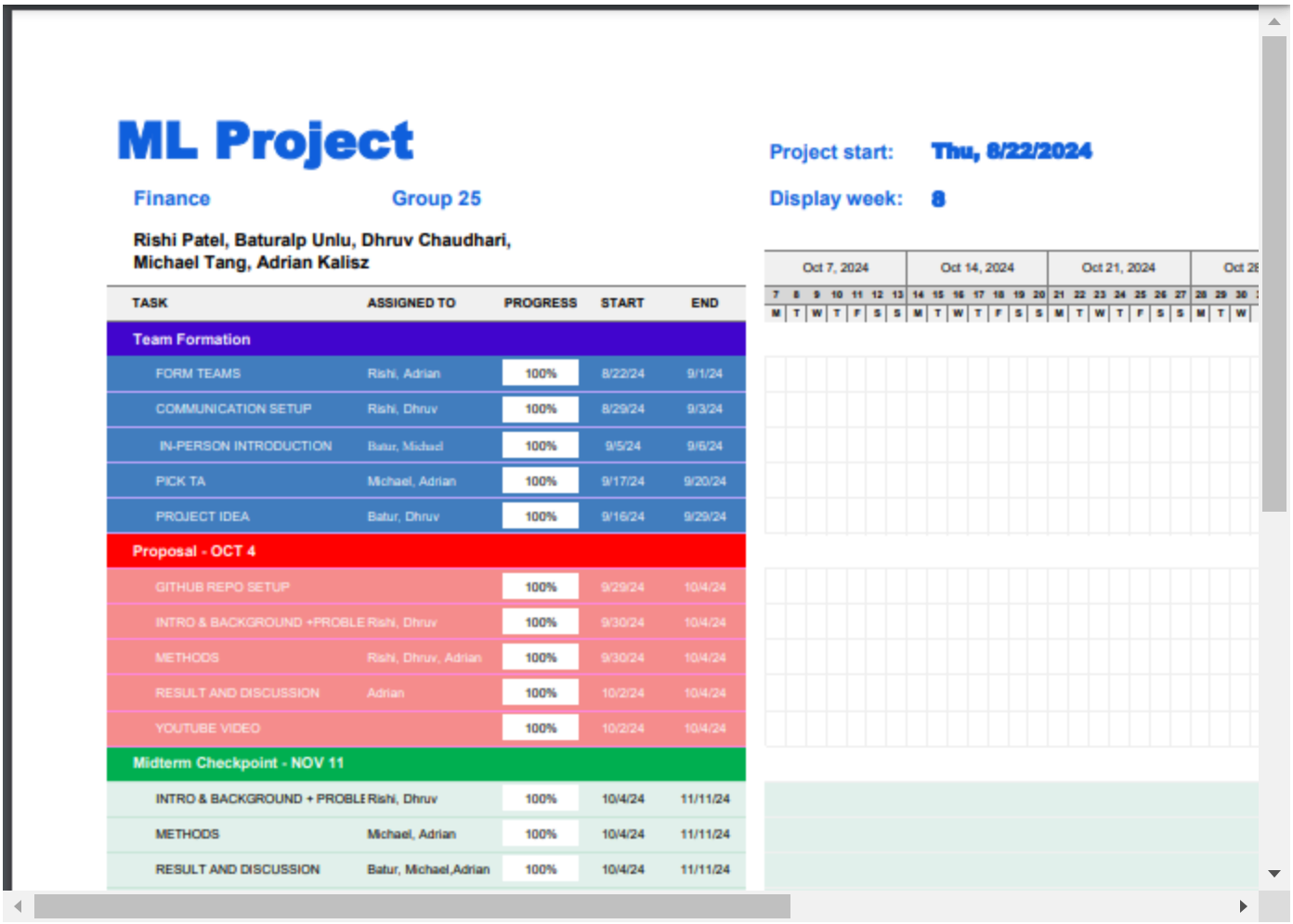
[2] S. Madge and S. Bhatt, "Predicting stock price direction using support vector machines," 2015.

[3] S. Chen and H. He, "Stock Prediction Using Convolutional Neural Network," IOP Conf. Ser.: Mater. Sci. Eng. 435 012026, 2018, doi: 10.1088/1757-899X/435/1/012026

[4] T. Rishi, "Stock market analysis using linear regression," Proceedings of the Jepson Undergraduate Conference on International Economics, vol. 4, article 4, 2022. Available at: https://scholarworks.uni.edu/jucie/vol4/iss1/4.

Gantt Chart

Gantt.pdf 1 / 1 100%



ML Project

Finance Group 25

Rishi Patel, Baturalp Unlu, Dhruv Chaudhari, Michael Tang, Adrian Kalisz

Project start: Thu, 8/22/2024

Display week: 8

Oct 7, 2024 | Oct 14, 2024 | Oct 21, 2024

TASK	ASSIGNED TO	PROGRESS	START	END
Team Formation				
FORM TEAMS	Rishi, Adrian	100%	8/22/24	9/1/24
COMMUNICATION SETUP	Rishi, Dhruv	100%	8/29/24	9/3/24
IN-PERSON INTRODUCTION	Batur, Michael	100%	9/5/24	9/6/24
PICK TA	Michael, Adrian	100%	9/17/24	9/20/24
PROJECT IDEA	Batur, Dhruv	100%	9/16/24	9/29/24
Proposal - OCT 4				
GITHUB REPO SETUP		100%	9/29/24	10/4/24
INTRO & BACKGROUND +PROBLE	Rishi, Dhruv	100%	9/30/24	10/4/24
METHODS	Rishi, Dhruv, Adrian	100%	9/30/24	10/4/24
RESULT AND DISCUSSION	Adrian	100%	10/2/24	10/4/24
YOUTUBE VIDEO		100%	10/2/24	10/4/24
Midterm Checkpoint - NOV 11				
INTRO & BACKGROUND + PROBLE	Rishi, Dhruv	100%	10/4/24	11/11/24
METHODS	Michael, Adrian	100%	10/4/24	11/11/24
RESULT AND DISCUSSION	Batur, Michael,Adrian	100%	10/4/24	11/11/24

Contribution Table

Name	Final Report Contributions
Rishi Patel	SVR Implementation, Presentation, Gantt Chart

Baturalp Unlu	SVR Implementation, CNN Implementation
Dhruv Chaudhari	Methods Write-up, Presentation, Presentation Video
Michael Tang	CNN Implementation, Quantitative Metrics, Analysis & Comparison, Next Steps, Visualization, GitHub Pages & Repo Deployment
Adrian Kalisz	SVR Implementation, CNN Implementation, Methods Write-up