CS 6140: Final Project - Group 5

Stock Price Predictor using Machine Learning Algorithms

1st Arjun Patel
Khoury College
Seattle, WA, USA
patel.arjun1@northeastern.edu

2nd Ching Yin Jenny Tang
Khoury College
Seattle, WA, USA
tang.ching@northeastern.edu

3rd Hao Sheng Ning
Khoury College
Portland, ME, USA
ning.ha@northeastern.edu

4th Yuhao Hua
Khoury College
Vancouver, BC, Canada
hua.yuha@northeastern.edu

***Abstract*—The interest in using machine learning and computational intelligence methods to predict stock market behaviours is growing, aiming to facilitate the earning of higher risk-adjusted returns. A large number of market participants use historical data and other pertinent information to forecast future stock prices [1]. Based on [1] and [2], we will fine tune a Machine Learning model using Logistic Regression, Decision Tree, Gaussian Naive Bayes, SVM, and MLP algorithms to predict future stock prices. The features from the references will be selected to yield the best result. Features other than those stated in the references might also be used to further improve model's accuracy.**

I. Basic Idea

The dynamic nature of financial markets has made predicting stock prices an intriguing and challenging task. Traders, investors, and analysts alike have long sought ways to unlock the secrets hidden within market data to gain a competitive edge. In recent years, machine learning has emerged as a powerful tool in the financial realm, offering the potential to uncover valuable insights and make informed predictions. In this study, we will explore various machine learning algorithms, including but not limited to decision trees, regression, and support vector machines (SVMs) to develop robust predictive models. The research will also delve into feature engineering, model evaluation, and optimization techniques to ensure the models perform effectively. The results and insights derived from this research can provide valuable guidance to market participants, contribute to academic advancements in the field of financial forecasting, and serve as a foundation for future developments in the intersection of machine learning and finance.

II. Objectives

The main objective of this project is to develop an application that predicts the future price of the S&P 500 as many more people get into the world of trading. We want to analyze the S&P 500 and utilize other economic data to predict the future price of the S&P 500. We will take data from yahoo finance and Federal Reserve to create our database and train our model using the data to predict future prices. The timeframe we want to analyze is day to day. We will make our model learn from day to day intervals to predict whether the price will be up or down in the next day.

III. Data Collection and Pre-processing

In our data collection, we wanted to gather the following information:

1) Price movement
2) Technical analysis
3) Economic data
4) Commodities data
5) Fundamental data

By focusing our data collection on the following, our model will be able to learn what majority of stock traders use to trade stocks. In order to collect this information, we used reliable and trusted data sources to ensure our model was using correct information. We used the following data sources using API's:

1) Yahoo finance
2) Federal Reserve database
3) Alpha Vantage

When organizing our data, it was important to make sure the target, which is predicting whether a day is up or down, was calibrated correctly. After ensuring our target data was calibrated correctly, we needed to use Yahoo finance to calculate the price movement and technical analysis. Many day traders use price movement and technical analysis to determine when to buy and sell stocks. They determine this based on known concepts such as if a stock is below the simple moving average (SMA), which means that the stock price is less than what it usually trades at.

Once we calculated the price movement, we moved onto gathering data from the Federal Reserve. We joined the data from the Federal Reserve to Yahoo Finance's data. The Federal Reserve reports on a daily, weekly, and monthly basis and posts about the health of the economy. An example of this could be reporting unemployment or measurements they create such as the Weekly Economic Index. By including this into our database, any reports that come out will be accounted for when looking at the stock price of SPY.

The last larger portion to our data collection was calculating out the average of all SP500's EPS estimated and reported. We did this by using a pivot table and joining the data together to report for average estimated and reported EPS on a daily basis. This means any time a company reported earnings, it would update the EPS. After we collected data on the reported and average EPS, we collected data on the income statement. The reason why we picked the income statement over the balance sheet and cash flow statement was because the income statement gives a snapshot of the companies overall profitability in the short term. This was ideal for predicting daily movement.

After we joined all the data together, we needed to figure out how to fill in missing values. For our data, we focused on forward filling because we did not want our model to look at large swings within our data due to values being 0. Our team also recognized that traders take the last value known to them and trade based on the last value seen, which we implemented into our model.

IV. Modeling

Before training the models, our group hand-picked several features and perform parameter tuning using RandomizedSearchCV. It includes scaling, feature selection using SelectKBest, and hyperparameter tuning for different models to yield a better parameter for each of the models.

We perform preprocessing techniques by using standardization before training machine learning models. It involves transforming the features of our dataset so that they have a mean of zero and a standard deviation of one. This process is particularly important when dealing with algorithms and greatly improved our accuracy score of the models. Each feature needs to be carefully chosen to avoid overshooting or slow convergence. Standardized features can simplify this process.

For data splitting, We used 0.9 so that our model could get more data to be trained on the last 2-3 years because the last 3 years were volatile. 2020-2022 was volatile but 2022-2023 was less volatile. Testing our model on a less volatile period would lead us to higher accuracy.

A. Gaussian Naive Bayes

The Gaussian Naive Bayes that was implemented for this project was from sklearn library. Applying standardization can lead to better generalization of the model on new, unseen data, as the model is less sensitive to the scale of input features. The features were chosen by best parameters and incorporated with features that are independent of each other. The data was split into the training and test data for different groups to find the best accuracy score and the corresponding classification report.

B. MLP

We selected MLP for this project and modified the model we used in class. The data was split into a 9:1 ratio for training and testing. The last ten percent of the data was the testing data that we tested our model on. After we split the data, we find the data using RandomizedSearchCV. This will help us to find the best parameters and estimators. A single-layer model is used and is easier to understand and interpret compared to multi-layers. In the model, hidden layer size in MLP defines how many nodes there are within MLP. Each node is calculating its own function and the ones with the highest accuracy are the ones that will be used. By having more nodes, we can improve the accuracy. Adding extra layers, however, did not help our model.

C. Logistic Regression

The model shows the relationship between the independent variables (features) and the probability of the binary outcome using the logistic function, also known as the sigmoid function. Our logistic function maps any real-valued number to a value between 0 and 1, representing the up and down of the stock market. We utilized and optimized the coefficients using a maximum likelihood to minimize the log loss.

D. Random Forest

Random Forest is known for its high accuracy and it provides good reliability. It can handle a large input size for our features and address our classification problem. The prediction in the Random Forest is determined by aggregating the predictions of all the individual trees.

E. SVM

We used SVM in the project to handle large data input. C parameter is used for the regularization, controlling the trade-off between achieving a wider margin and allowing some misclassifications. Our group handled constant features warning by identify and removing the constant values and trained the SVM model using the best combination of features. Utilized the SVC class from Scikit-learn library for the SVM mode. However, it has higher computation complexity than any other models when deal without large datasets.

V. Cross-Validation & Evaluation

In the evaluation of our machine learning models, our primary objective is to thoroughly assess the performance, robustness, and reliability of five distinct algorithms: Support Vector Machine (SVM), Logistic Regression, Random Forest Decision Tree, Gaussian Naive Bayes, and Multilayer Perceptron (MLP). Our evaluation strategy is carefully crafted to address both overfitting and underfitting through the application of k-fold cross-validation. This ensures that the models are evaluated on different subsets of the dataset, providing a more unbiased estimate of their generalization abilities. Furthermore, an extensive ablation study will be conducted to investigate the impact of various configurations, including feature selection, hyperparameter tuning, and model architecture. Alongside these evaluations, we will also delve into an analysis of extreme errors, exploring the circumstances under which our models might fail or excel. The final assessment will encompass a range of metrics, including precision, recall, F1-score, accuracy, confusion matrix, and the ROC-AUC curve, offering a comprehensive view of model performance across different dimensions.

A. SVM

The Support Vector Machine (SVM) was tuned with a hyperparameter space that encompassed various kernel types and a range of values for regularization strength C. The best

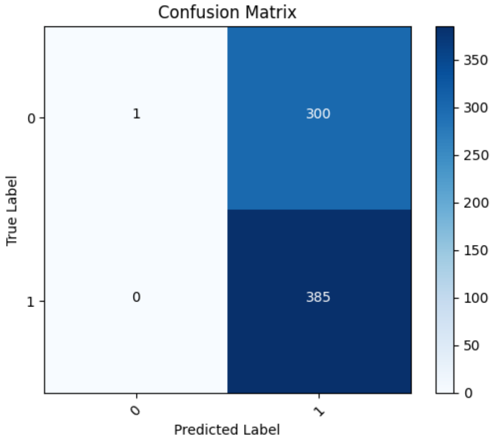

Fig. 1.

parameters identified were a linear kernel with $C = 1$, along with the selection of 17 most relevant features.

The 5-fold cross-validation revealed a mean accuracy of approximately 0.5369, with a very low variance of 1.224×10^{-5}, indicating stability in the model's performance across different subsets of the data. The accuracy scores for each fold are presented, displaying consistent performance. The k-fold cross-validation used in the evaluation strategy helps to address the concerns of over-fitting and under-fitting by providing a more unbiased estimate of the model's performance on unseen data. However, the stark contrast in precision and recall for the two classes could be indicative of an underlying issue, possibly related to class imbalance or feature selection, that may lead to suboptimal generalization.

A detailed look at the classification report shows a significant discrepancy between the precision and recall for the two classes, with the model exhibiting a tendency to favor the positive class. The precision for class 0 was perfect, but the recall was extremely low, reflecting an imbalance in the classification. This is further depicted in Figure 1, the confusion matrix, where a considerable number of false negatives for class 0 is observed.

The ROC-AUC curve, illustrated in Figure 2, has an area of 0.5, which suggests that the model's ability to discriminate between classes is no better than random guessing. The underwhelming performance on the ROC-AUC metric is a critical observation, warranting further investigation and potential adjustments to the model or preprocessing steps.

The analysis of extreme errors is challenging in this context due to the model's inclination to favor one class over the other. The large number of false negatives for class 0 suggests that the model might be missing key patterns or characteristics that define this class. An ablation study, focusing on the selection and impact of the features, could provide more insights into these errors, potentially paving the way for refinement in the model's ability to discern between the classes.

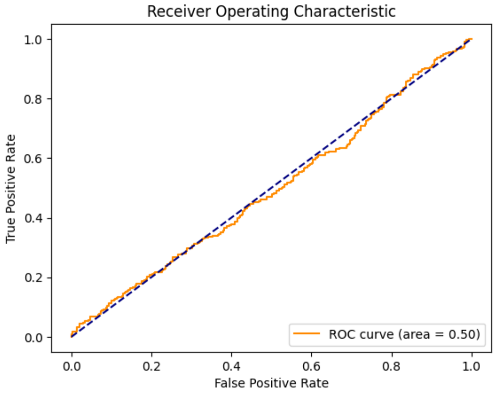

Fig. 2.

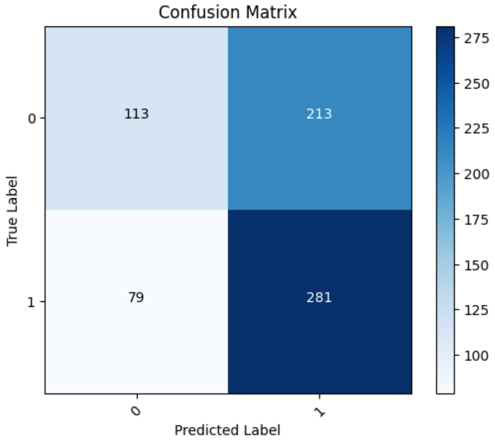

Fig. 3.

B. Logistic Regression Model

The Logistic Regression model was trained using the 10 most relevant features including 'Close', 'Volume', 'SMA', 'VIX', 'UNEMPLOYMENT', 'Crude_Oil_WTI', 'Gold', 'Annual Mean', 'Annual Weekly Mean', and 'Annual Quarterly Mean'.

A 5-fold cross-validation was performed, yielding a mean accuracy of approximately 0.5846, with a low variance of 1.156×10^{-5}. This consistency across different folds signifies a stable performance of the model, addressing concerns of over-fitting and under-fitting.

The precision and recall values for the two classes were more balanced compared to the SVM model, although some discrepancy still exists. Class 0 had a precision of 0.59 and a recall of 0.35, while class 1 had a precision of 0.57 and a recall of 0.78. This reflects a higher propensity of the model to classify instances into class 1. Figure 3, the confusion matrix, provides a visual representation of the classification results, with a notable number of false negatives and false positives. The ROC-AUC curve, depicted in Figure 4, shows an area of 0.61, an improvement over the SVM model, but still indicative of a need for further refinement in the model's ability to

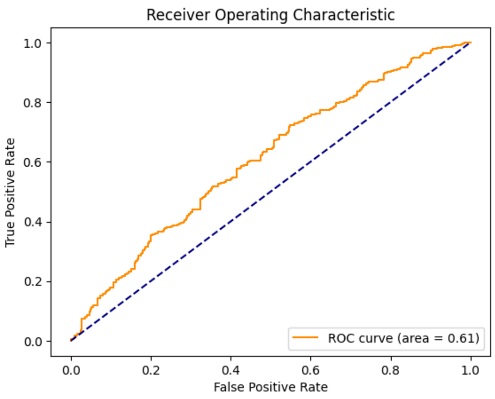

Fig. 4.

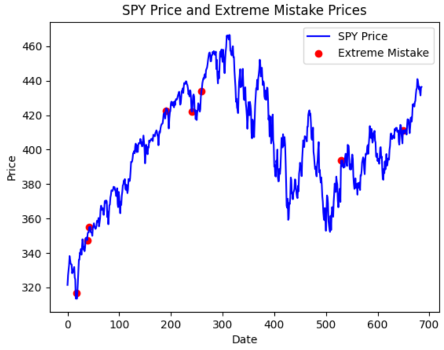

Fig. 5.

differentiate between the classes.

In analyzing the extreme errors, the presence of false negatives and false positives may be tied to the features selected or the nature of the data itself. An ablation study investigating the influence of these 10 features could reveal insights into the behavior of the model and identify opportunities for enhancing its predictive capabilities.

We also ran a separate ablation study to look at overfitting and underfitting a regression model. Using sklearn, running a C (regularization parameter) of 10000 was optimal in addressing overfitting and underfitting. However, despite fine tuning our model, this does not necessarily mean our models even with higher probabilities of 70%, will get each prediction correctly. We found in certain periods, even with high probabilities, we found extreme errors. These errors tend to cluster in certain areas (fig. 5). This might suggest certain periods are harder to predict due to volatility and uncertainty as seen in the second cluster when SPY was around $430.

C. Random Forest Decision Tree

The Random Forest Decision Tree model was trained using a broad set of hyperparameters, including the number of estimators, the criterion for splitting, maximum depth, minimum samples split and leaf, maximum features, and bootstrap options. The best parameters identified were 58 features, 250 estimators, a minimum sample split of 12, a minimum sample leaf of 2, a maximum depth of 70, with ’log2’ as the option for maximum features, ’entropy’ as the criterion, and no bootstrapping.

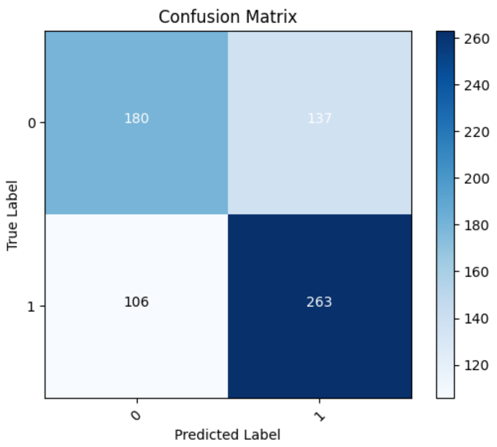

Fig. 6.

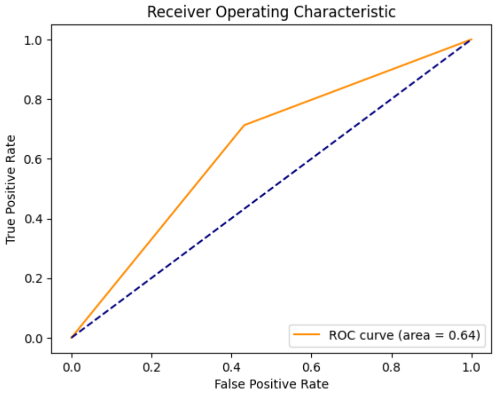

Fig. 7.

The mean accuracy score of approximately 0.6075 with a variance of 0.0001057 shows that the model is relatively stable across the 5-fold cross-validation, indicating that there is no sign of overfitting or underfitting. The balance between precision and recall, for both classes, as shown in the classification report, reflects a well-performing model in identifying both positive and negative instances.

The confusion matrix, illustrated in Figure 5, reveals a more balanced distribution of correct and incorrect predictions compared to previous models. However, it does show areas where the model may struggle, specifically with false negatives for class 0 and false positives for class 1. The ROC-AUC curve, depicted in Figure 6, has an area of 0.64, indicating a reasonable discrimination ability between classes. This, coupled with the consistent precision, recall, and f1-score, signifies a model with strong baseline performance. The analysis of extreme errors does not reveal significant imbalances or favouring of one class over the other. However, the identified false

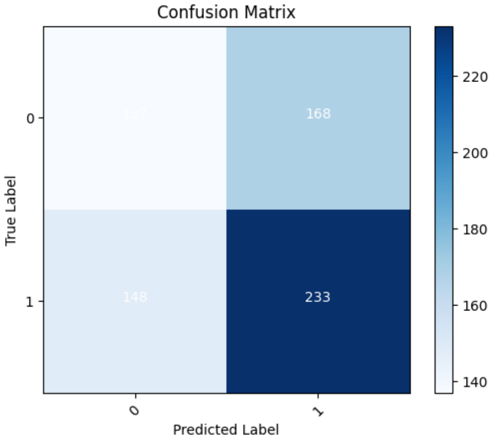

Fig. 8.

positives and negatives might be explored further to understand the underlying reasons for these misclassifications. Careful investigation of feature importance and potential interactions between features could lead to an optimized model, reducing these errors and enhancing overall performance.

D. Gaussian Naive Bayes Model

The Gaussian Naive Bayes model was tuned with a range of values for the number of selected features and a logarithmic scale for the variance smoothing parameter. Different prior probabilities for the classes were also explored. The best parameters were identified as a selection of 42 features, a variance smoothing of 0.8111, and equal priors of 0.5 for both classes.

The 5-fold cross-validation resulted in a mean accuracy of approximately 0.5143 with a variance of 0.0002941. The individual fold accuracy scores, ranging from 0.4891 to 0.5348, indicate that the model's performance is somewhat inconsistent across different subsets of the data.

The classification report unveils a relatively balanced precision and recall for both classes, but the scores are not particularly high. The accuracy of 0.54 reflects the model's modest ability to classify the instances correctly.

The confusion matrix, shown in Figure 7, further illuminates the model's difficulty in making accurate predictions, with a considerable number of false negatives for class 0 and false positives for class 1. The ROC-AUC curve, depicted in Figure 8, has an area of 0.54, confirming that the model's ability to discriminate between classes is barely above random guessing. This is a crucial observation that suggests the need for potential adjustments to the model or preprocessing steps. The Gaussian Naive Bayes model's overall performance appears to be suboptimal, with the results revealing a number of areas where improvements might be sought. Further investigation into the feature selection and the influence of the variance smoothing parameter may reveal opportunities for refining the model.

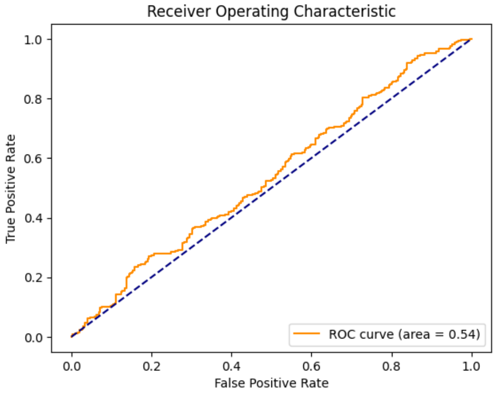

Fig. 9.

E. MLP

The Multi-Layer Perceptron (MLP) was tuned with a hyperparameter space that encompassed a range of values for the number of selected features, hidden layer sizes, alpha regularization strength, and activation functions. The best parameters were identified as 42 features, 750 hidden layer sizes, $\alpha = 0.001$, and a ReLU activation function.

The 5-fold cross-validation revealed a mean accuracy of approximately 0.6738 with a variance of 5.51×10^{-5}, suggesting a fairly stable performance across different subsets of the data. The individual fold accuracy scores are all within the range of 0.6645 to 0.6775, highlighting the model's consistent performance.

The classification report shows an encouragingly balanced precision and recall for both classes, with an accuracy of 0.71 and F1-scores for classes 0 and 1 at 0.68 and 0.74, respectively. This indicates a relatively strong ability to classify instances correctly.

The confusion matrix, represented in Figure 9, offers a detailed view of the classification, with a considerable number of correct predictions for both classes. The model's ability to minimize both false negatives and false positives is noteworthy. The ROC-AUC curve, as depicted in Figure 10, has an area of 0.71, further confirming that the model's ability to discriminate between classes is substantially above random guessing. This performance represents a significant achievement for the MLP model. The overall results for the MLP model are quite promising. Its balanced performance across various metrics suggests that it could be a robust choice for this particular classification task. The exploration of different architectures and tuning of hyperparameters may further optimize the model's performance, but the current configuration has already exhibited strong capabilities.

VI. Case Analysis

Our model is nevertheless not without any misclassification, few examples are as follows:

1) **False Negative Case #1: February 13th, 2008:** Stocks rallied Wednesday after a surprisingly strong

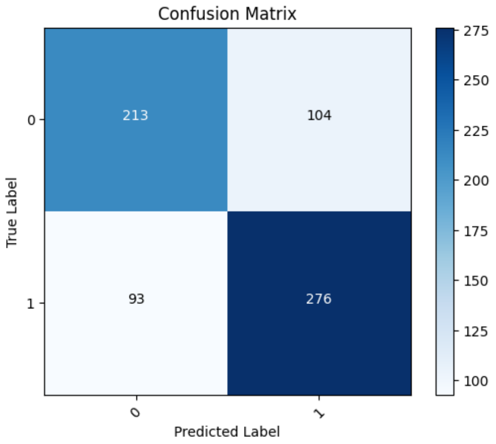

Fig. 10.

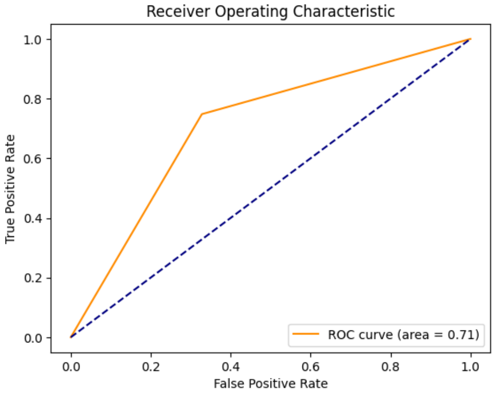

Fig. 11.

January retail sales report helped counter worries that a weakening consumer could send the already struggling economy into a recession. According to early tallies, the Dow Jones industrial average (INDU) gained roughly 1.4%, while the broader Standard & Poor's 500 (SPX) index also added around 1.4%. Both the Dow and S&P 500 ended higher for the third session in a row.

2) **False Negative Case #2: October 13th, 2008:** Britain, Germany, France, Italy and other European governments all announced rescue packages worth hundreds of billions of dollars designed to stave off a global financial crisis that has threatened to spin out of control. In the United States, Treasury Secretary Henry Paulson said Washington was developing plans to buy equity in financial institutions to halt the prolonged market turmoil.
3) **False Positive Case #1: September 29th, 2008:** The news of the bankruptcies of Wall Street brokerage firm Lehman Brothers, Savings and Loan bank Washington Mutual, as well as the Fed's announcement that it would provide an $85 billion bailout for insurance provider American International Group (better known as AIG) to keep it from going under all heavily undermined investors' confidence in the growth of the stock market.
4) **False Positive Case #2: March 18th, 2020:** The declaration of Covid as a global pandemic and subsequent strict lockdowns hampered investors' confidence in the capital market.
5) **False Positive Case #3: May 5th, 2022:** The breakout of the war in Ukraine and subsequent sharp rises of commodities including grains and crude oil all add uncertainty to global economic recovery following Covid.

All 5 mis-classifications can be attributed to unexpected geopolitical events and corresponding government reactions. Different unforseen events can arouse different sentiments towards the future of the stock markets within each individual investors and financial analysts. Fear, greed, and herd mentality can lead to irrational behavior, causing stock prices to deviate from their intrinsic values. These emotional reactions can lead to sudden and unpredictable price swings. In some cases, market participants may attempt to manipulate stock prices to further their benefit. This could be through spreading false information (rumors) or engaging in illegal practices. Market manipulation can make it even more challenging to accurately predict price movements, because they are completely arbitrary and market unrelated.

Potential methods of improvements:

1) Develop a ML powered applications that automatically summarizes and categorizes announcements and news into different sentiments, for example (optimistic, worrisome, pessimistic, indifferent)
2) Add features that takes consideration of announcements and global events. The difficulty with this is that it can be hard to classify events into discrete categories, and doing so can incur tremendous amount of time, effort and domain expert knowledge.
3) Another improvement could be to increase the number of possible categories from 2 to more to capture the magnitude of price decreases or increases. For example, as opposed to just up and down categories, we can have -5% to -10% or 10% to 15%

VII. Conclusion

The proposed solution is an innovative approach to give investors and financial institutions a reliable suggestion to future market trends.

Among the models evaluated, the Multi-Layer Perceptron (MLP) exhibited the most promising performance, achieving a mean cross-validation accuracy score of 0.6738 and a balanced precision and recall across classes. The Random Forest Decision Tree also showed strong capabilities, with a mean accuracy score of 0.6075, suggesting a robust ability to classify instances correctly. Both models leveraged their respective strengths: MLP in complex function approximation and Random Forest in ensemble learning, capitalizing on decision trees.

In contrast, the Gaussian Naive Bayes model's performance was less encouraging, with the lowest mean accuracy of 0.5143 among the models. Its simplifying assumption of feature independence likely hindered its performance in this context. The SVM model's low ROC-AUC area of 0.5 was another critical observation, indicating potential issues in discrimination between classes. This project successfully applied various machine learning models to a complex classification task, highlighting the diversity of approaches that can be used to tackle such problems. The rigorous cross-validation and hyperparameter tuning employed ensured that the models were tested thoroughly, providing insights into their potential real-world applicability.

References

[1] M. M. Kumbure, C. Lohrmann, P. Luukka, J. Porras, ”Predicting and Visualizing Financial Time Series using Machine Learning Techniques,” *Expert Systems with Applications*, vol. 197, no. 2, pp. 24-36, 1 July 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422001452

[2] S. Shen, H. Jiang, T. Zhang, ”Stock Market Forecasting Using Machine Learning Algorithms,” [Online]. Available: http://cs229.stanford.edu/proj2012/ShenJiangZhang-StockMarketForecastingusingMachineLearningAlgorithms.pdf

[3] S. R. Sneha, D. Sneha, E. Anitha, ”Predicting and Visualizing Financial Time Series using Machine Learning Techniques,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 8, no. 7, pp. 86-92, July 2020. [Online]. Available: https://www.ijraset.com/