Analysis of macroeconomic metrics and prices of top twenty markets in NASDAQ 100 to predict behavior and returns using machine learning

Mauricio Ruiz
National College of Ireland
Final Project
Dublin, Ireland
x24146102@student.ncirl.ie

Abstract: The main purpose of this project is to use supervised machine learning algorithms combining with historical financial data from the top 20 markets in NASDAQ-100 along with macroeconomic indicators to predict NASDAQ 100's returns or losses as well as the direction it will move (bullish or bearish). The data was gathered from October 1st, 2015, until November 29th, 2024, and includes stock prices, volume and some other relevant features from the top 20 constituent stocks in NASDAQ 100 as well as data related to macroeconomic indicators such as Federal Funds Rate, CPI, M2 Money Supply, and Treasury Yields. By understanding how these factors interact and affect NASDAQ's index returns it is possible to make better informed decisions when investing in the market.

The algorithm used is Extreme Gradient Boost (XGBoost) for the two models, which is known to perform accurately in time series for continuous variables as well as categorical. Two models were trained, one for the market movement direction (bullish, bearish, or stable) and one for returns. The training is based on splitting the data as follows: 80% training and 20% testing chronologically. The accuracy for each model was measured using different metrics. The classification model resulted with an accuracy of 74% and a F1-Score of 0.78 predicting bullish movements. For the regression model, the metrics used were R^2 with a score of 0.4609 and Root Mean Squared Error (RMSE) of 0.0081 which reflects predictions close to the actual returns in the index.

Keywords: supervised machine learning, NASDAQ-100, macroeconomic analysis, stock prices prediction, financial modelling.

I. Introduction

The usage of algorithms is something that has been applied in software development to have computers performing tasks ever since their invention. Over the last years algorithms have been used to create what is now known as machine learning (ML). Machine learning is the science of programming computers to perform tasks based on data given to train a model and using statistics to obtain the most accurate outcome possible from the data fed to the model. [1]

National Association Of Securities Dealers Automated Quotations (NASDAQ) is a system used to provide quotations to brokers and traders and uses networks of phones and computers that are not centralized for all the trading. Nasdaq is home to thousands of different markets including big technological companies such as Microsoft, Apple, Amazon just to mention a few.[2] Nasdaq 100 is an index that tracks the 100 largest companies listed on NASDAQ. This index includes corporations such as Amazon, Apple, Microsoft and Nvidia.

Nowadays, markets' behaviors are hard to identify due to different aspects being capable of affecting the indices' performance. This makes it complex to know how the markets will behave one day compared to the previous one and make a well-informed decision when it comes down to choosing the right time to invest. Taking this into consideration, it can be possible to use some of those factors to predict the return of the companies tracked by NASDAQ 100 by using a machine learning model.

On the one hand, currently there are models used to predict market prices using deep learning techniques. These models based on neural networks require a lot of computational power and sometimes can be hard to interpret as they are composed of many layers. The purpose of this project is to create a robust predictive model using Extreme Gradient Boosting (XGBoost) which will not require so much CPU power and that can be reliable and easy to implement.

On the other hand, some predictive models are built based only on stock prices and do not include other possible variables that could affect the market's performance. In this research the model will be used to classify the market's behavior by adding macroeconomic metrics. This extra layer can help investors to understand when markets will go up or down based on the economic situation.

Including different variables will allow to have a much clearer overview of the market and how its behavior over the last decade is key to understand the future performance. These different features can help to build a very robust and reliable predictive model that can produce a much higher return of investment.

II. Related Work

A. ML is stock price prediction

ML has been studied previously to understand how it can be used to predict stock prices and help investors to make better decisions. Statistics have always been core to understanding markets' behaviors however, ML offers a faster and easier way to implement these calculations. Besides this, ML also has been used to identify more complex patterns in the data that elevates the stock market prices' predictions to a higher level.

Using ML algorithms and historical data to train a system can allow automatic investments as well as to companies or investors to stay on top of the matter itself. [3]

B. ML Models used in stock price prediction

The usage of different machine learning algorithms is something that has been done already in different research. These studies have used different approaches, different data with different features and different models that specifically fit the project's purpose.

Kumar et al. (2022) used different models such as Linear Regression (LR), Support Vector Machines (SVM), Random Forests (RF), Artificial Neural Networks (ANN) and Long Short-Term Memory networks (LSTM) on their research. It was found that ANN and LTSM models were the top performers. However, it was concluded that some other features should be considered and included to improve the models' accuracy such as market's directional movements and other indices prices as these factors are commonly correlated. [4]

Another approach has been proposed by Vardhan and Jaffino (2022) using Automated Time Series (Autots) which is a python's package. This uses different calculations and uses genetic algorithms to choose the most useful model for predictions. Using Autots demonstrated that this option is very accurate in the predictions of stock prices and does not require hyperparameter adjustments which eliminates the need of human intervention. [5]

Taking these publications as a starting point, it is noticeable that many different ML models can be used to predict stock prices but not all of them perform with the accuracy expected when they are only trained on stock price data and other factors are ignored. Adding other relevant features could potentially add valuable information to the model increasing the accuracy of its predictions.

C. Contribution

Using only stock prices to train a ML model will not produce very accurate predictions. Stock prices are often affected by other factors such as political and economic situation. Economics and geopolitics cause impacts in markets, causing difficulties in predicting the prices behaviors' which is why the usage of advanced algorithms can help to predict trends and identify risks. [6]

By adding extra features to the training data such as macroeconomic indicators, the model could learn more complex patterns and be more robust.

On the other hand, previous studies have implemented different ML models such as Convolutional Neural Networks (CNN) which have proven to be expensive to train and demand a lot of computational resources. [7]

This project is focused on using extreme boosting gradient model, which will require less CPU resources and will be easier to implement.

In summary, the aim of this research is to fill gaps from previous studies by adding macroeconomic indicators to the training data to produce more accurate results in stock price prediction while using a more efficient and interpretable ML model (XGBoost).

III. METHODOLOGY

A. Research Design

This study was designed based on CRISP – DM (Cross Industry Standard Process for Data Mining) model. [8]

This methodology is based on the following steps:

- **Business Understanding**: Used to explain the needs and reason for this study. Predict Nasdaq-100 index prices to help investors make more accurate decisions.
- **Data Understanding:** Includes data collection from Yahoo Finance and FRED and understanding the usability of the data in this study.
- **Data Preparation:** Including data cleaning, preparation and creating engineered features.
- **Modeling**: Training a model (XGBoost) for two different purposes, classification and regression.
- **Evaluation:** Analyze the model's accuracy by using different metrics.
- **Deployment:** Not currently to be implemented but this model could be useful if predictions are accurate.

B. Data Collection

Data was collected from two different sources, Yahoo Finance and Federal Reserve Economic Data (FRED). The time frame used to select the data was from 1st of October 2015 to 31st of December 2024.

1) *Macroeconomic Metrics:*

- Source: FRED
- Dimensions: 7 features and 113 observations.
- Indicators: Federal Funds Rate (FedFunds) - Consumer Price Index (CPI) - Unemployment Rate - Gross Domestic Product (GDP) - 10-Year Treasury Yield (TenYear) - M2 Money Supply (M2)
- Frequency: Monthly

2) *Market's metrics:*

- Source: Yahoo Finance
- Dimensions: 103 features and 2368 observations.
- Markets selected: Apple Inc. - Microsoft Corporation - NVIDIA Corporation - Amazon.com Inc. - Alphabet Inc. (Google) - Meta Platforms Inc. (formerly Facebook) - Tesla Inc. - Broadcom Inc. - Adobe Inc. - PepsiCo Inc. - Costco Wholesale Corporation - Advanced Micro Devices Inc. - Netflix Inc. - Intel Corporation - Cisco Systems Inc. - Texas Instruments Inc. - Qualcomm Incorporated - Applied Materials Inc. - PayPal Holdings Inc. -

Starbucks Corporation – NASDAQ (supervised featured).

- Features: Date, Open, High, Low, Close, Adjusted Close, Volume.
- Frequency: Daily.

C. Exploratory Data Analysis (EDA)

The EDA was done as follows:

1) *Identifying Missing Values:*

- Nasdaq-100 did not have any missing value so there was no need to implement any processing.
- FRED data contains missing values in the GDP variable; a total of 76 missing values were identified (Figure 1). This is due to GDP being reported quarterly. There are 37 quarters in the period selected for this study. As daily data is used in stock prices, forward filling was used to fill the gaps as there are no changes in GPD until reported again.

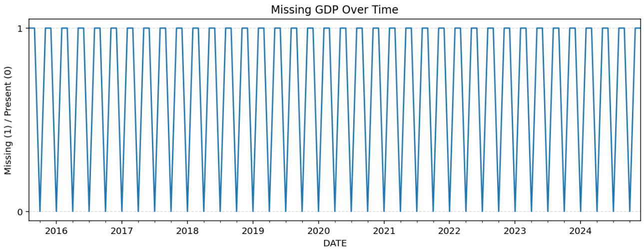

Fig 1. Missing values in GDP across time

2) *Date alignment:*

- As stock prices are reported daily and macroeconomic indicators typically monthly or quarterly, forward and backward filling was used as an interpolation method. This was done to extend the macroeconomic dataset to match the daily reports on stock prices and have consistency. Forward and backward filling was implemented to keep consistency in the FRED data as macroeconomic indicators tend to be constant until new reports are published.

3) *Scaling features:*

- Considering the macroeconomic indicators are reported in different units, it is necessary to normalize the features to have consistency with stock prices scale. The method used was Z-Score normalization as it keeps original distribution of the data making it more suitable for modeling. This process was applied to the features engineered.

D. Feature Engineering

To improve the model's accuracy, it was necessary to engineer some features that would capture trends in the macroeconomic indicators as well as in the market's behaviour. This was done in two different sets as each model requires a different feature engineered.

1) *Regression and Classification Models:* A series of engineered features were created to be used in the two models*:*

a) Stock Market Returns: Calculated based on the percentage change compared to previous day for the 20 stock markets used in this study. This helps the model as it reflects the most change in the prices and momentum.
Featured Engineered: All markets features (e.g., AAPL_Return).

b) Lagged Variables: Calculated for 6 macroeconomic indicators and the 20 stock market returns. These features assign the previous date values (date – 1) to the following date so the model can capture how was the indicators' behaviours prior to stock price changes.
Features Engineered: FedFunds_lag1, CPI_lag1, Unemployment_lag1, GDP_lag1, TenYear_lag1, M2_lag1 and the 20 stock markets returns (e.g., AAPL_Return_lag1).

c) GPD Flag: Used as a binary to indicate the model when the GDP was reported so it can interpret the impact on the market's behaviour.
Feature Engineered: GDP_Updated.

d) Index Return Rolling Statistics: Used the target variable to calculate trends and volatility (standard deviation and average) in 3 days timeframe. This would help the model understand patterns when the market is bullish or bearish.

Feature Engineered: NDX_Return_rolling_mean3, NDX_Return_volatility3.

2) *Regression Model:*

a) Rolling Averages: These were calculated based on macroeconomic indicators to avoid noise in the data and to capture trends in a more accurate manner. They were calculated based on 3 days averages.
Features Engineered: FedFunds_ma3, Unemployment_ma3.

b) Rolling Volatility: This was calculated using standard deviation based on 3 days and used to identify volatile A periods and how they can affect the market's behaviour.
Features Engineered: CPI_rolling_std3

3) *Classification Model:*

a) Macro indicators change (%): Calculated to reflect the changes in percentage compared to previous report. Features Engineered: FedFunds_pct_change, CPI_pct_change, Unemployment_pct_change, GDP_pct_change, TenYear_pct_change, M2_pct_change.

b) Lagged Nasdaq Close: This feature is based on the previous day closing price and shifted by one day.
Features Engineered: NDX_Close_lag1

c) Binary Target: Feature to identify the market's next day direction. Using 1 as possible bullish and 0 as bearish or stable.
Features Engineered: NDX_Direction_Tomorrow

After creating the new engineered features raw data from stock prices and macroeconomic indicators were dropped. This was done to reduce redundancy in features and to train the model with the only necessary and relevant features.

E. Features Analysis

To explore and understand better the interaction between the macroeconomic indicators and the returns in NASDAQ-100 a series of line plots were created. Each of these plots were used to compare the behaviour of all economic indicators in the data (normalised) and NASDAQ returns across time. The indicators used for the plots were: Federal Funds Rate, Consumer Price Index (CPI), Unemployment Rate, Gross Domestic Product (GDP), 10-Year Treasury Yield, and M2 Money Supply. The plots present on the Y left axis and blue line the macro indicators (normalised) and on the Y right axis and orange line the NASDAQ-100 returns.

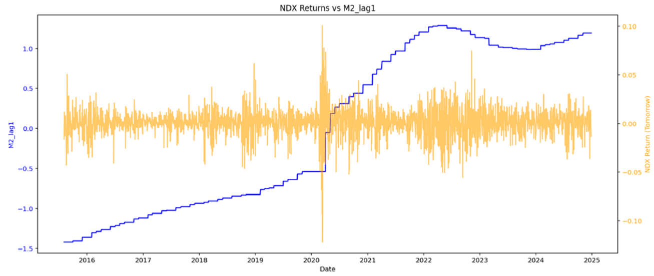

Fig 2. Lagged Money supply vs Nasdaq returns.

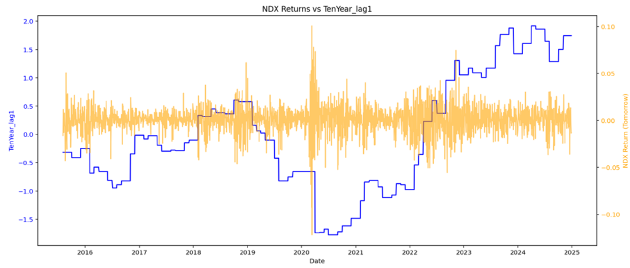

Fig 3. Lagged Ten-year treasury vs Nasdaq returns.

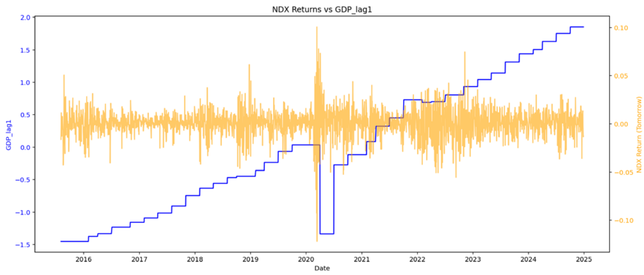

Fig 4. Lagged gross domestic product vs Nasdaq returns.

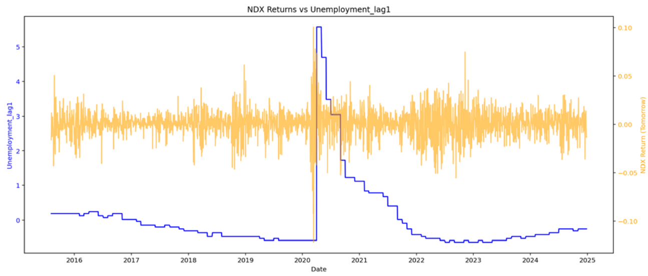

Fig 5. Lagged unemployment vs Nasdaq returns.

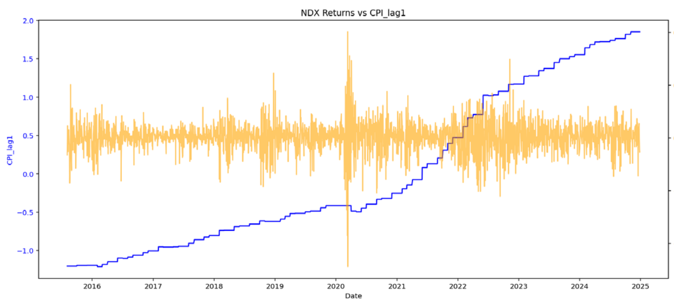

Fig 6. Lagged consumer price index vs Nasdaq returns.

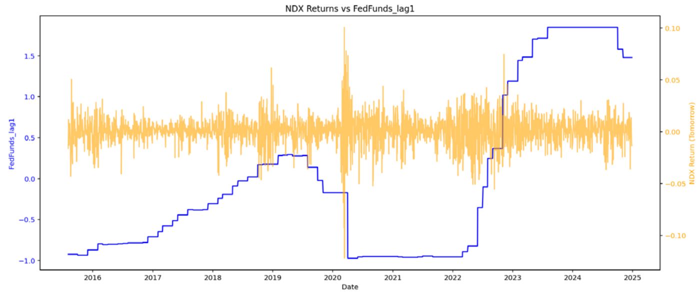

Fig 7. Lagged federal funds rate vs Nasdaq returns.

After analyzing all the plots (figures 2-7), it is noticeable that trends in macroeconomic indicators do not create drastic anomalies in the market's daily returns. The behaviour of the index seems to be consistent and independent from the economic indicators. The only time frame in which it seems to be connections between the indicators and NASDAQ returns was during the Covid period when all markets and economies suffered changes.

IV. Model

For this research Extreme Gradient Boost (XGBoost) was the ML model selected. This project applied supervised learning from structured data in which XGBoost has proven to be excellent to predict continuous and discrete values (regression and classification). [9]

XGBoost uses gradient boosting which creates decision trees that improve the decision of previous ones to create more accurate predictions. It is a model that has been used for financial predictions, and it is robust for time series prediction whether they are classifications or numeric as it uses different regularization techniques L1 (Lasso) and L2 (Ridge) to avoid having an overfitting model.

XGBoost was used for the two models proposed in this research – classification and regression.

1) *Classification Model:* The classification model was used to predict the NASDAQ-100 directional movement based on the top 20 stock markets and financial macro indicators. This model used as a target variable a binary feature engineered (NDX_Direction_Tomorrow) which establishes if the market was bearish or bullish. Predictions were made by using XGBoost classifier module.
Performance evaluation metrics used: Accuracy, Precision and Recall, F1-Score and Confusion Matrix.

2) *Regression Model:* The regression model was used to predict NASDAQ-100 next day returns on the top 20 stock markets and financial macro indicators. This model used as a target a continuous engineered feature (NDX_Return_Tomorrow) which shows the variation of the market closing prices compared to the previous day. Predictions were made by using XGBoost regressor module. Early stopping was applied to avoid overfitting and get the most accurate results when validation is not improving anymore.
Performance evaluation metrics used: R^2 Score, Root Mean Squared Error (RMSE).

3) *Training and testing:* The data used to train the two models was split as follows:
Train the model: 80%
Test the model: 20%

4) *Tuning:* The hyperparameters tuned to improve the models' performance were: n_estimators, max_depth, learning_rate, subsample, colsample_bytree, reg_alpha, (L1) and reg_lambda (L2) regularization, random_state.

A. Tools

The model was implemented using python using Spyder as IDE and a series of libraries such as:

- Pandas
- Yfinance (used to get NASDAQ 100 data and stock prices)
- pandas_datareader (used to get macro indicators data)
- Numpy
- sklearn.preprocessing
- Scikit-learn
- XGBoost
- Scikit-learn's GridSearchCV
- Matplotlib
- Seaborn

V. Results

A. Regression Model:

1) *Performance:* The model was trained using 80/20 partition based on time (chronologically). The model's performance was optimized after experimenting with the hyperparameters (Table 1) and after three rounds tuning them the model produced the most accurate results (Table 2). Overfitting was prevented by using early stopping in each of the three rounds. Accuracy was measured using Root Mean Squared Error (RSME) and R^2 score. The final most accurate results were:
- RMSE: 0.008153
- R^2 Score: 0.4609
- Best iteration: 997

The model required 997 iterations (figure 8) to find the most accurate results. The RSME was 0.0081, which indicates that on average the prediction error is 0.81%. The R^2 was 0.46, indicating that the features used to train the model explain 46% of the variance in the NASDAQ-100 next day returns.

Hyperparameter	1st Round	3rd Round
n_estimators	300	1000
max_depth	5	3
learning_rate	0.05	0.01
subsample	0.8	0.8
colsample_bytree	0.8	0.8
reg_alpha	0.1	0.1
reg_lambda	1.0	1.0
random_state	42	42
early_stopping_rounds	20	40

Table 1. Hyperparameter tuning rounds for regression model

Tuning Round	RMSE	R2	Best Iteration
1st round	0.008258	0.4468	183
3rd round	0.008153	0.4609	997

Table 2. Accuracy results for regression model

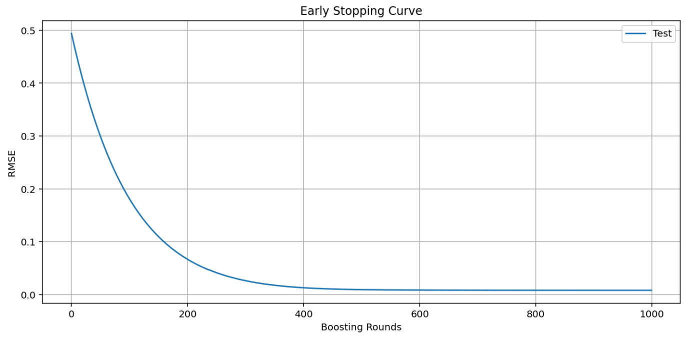

Fig 8: XGBoost early stopping cure (iterations needed)

2) *Features:* The importance of the features were extracted from the XGBoost model. The top 10 (figure 9) and bottom 10 (figure 10) were plotted to identify which ones contributed the most and the least to the predictions.

The NASDAQ-100 return average rolling feature calculated was the one that contributed the most to the model followed by the lagged returns from tech companies.

The macro indicators did not contribute much to the predictions. Macro indicators might did not explain much about the market's behaviors in short term predictions. These metrics might be more insightful in the long-term return predictions as they don't vary daily.

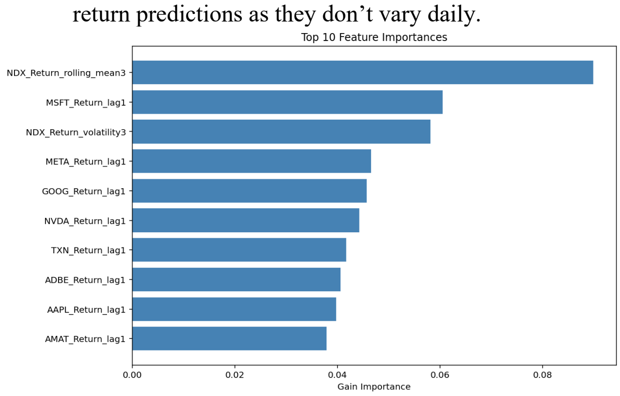

Fig 9. Top 10 features contributing to the model

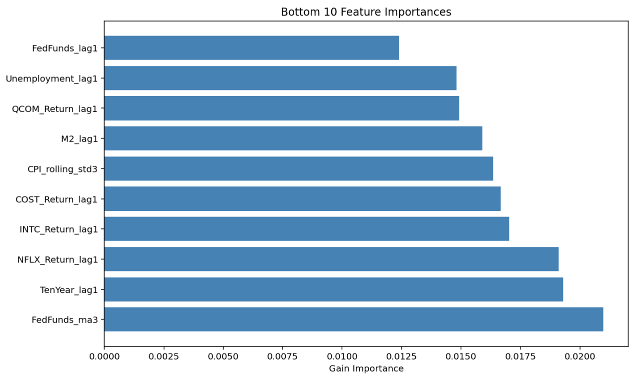

Fig 10. Bottom 10 features contributing to the model

3) *Predictions results:* The model performs well overall. From figure 11 it can be seen predictions tend to be accurate when the market returns are not volatile. This can be seen during periods with flatter curves where predictions are close to the actual results. On the other hand, periods where sudden peaks or drops take place, the model seems to underestimate being more cautious in its predictions.

From figure 12 it can be confirmed that most of the predictions are close to NASDAQ-100 actual returns. The points are concentrated close to the diagonal line which is more noticeable when the market's behavior is less volatile.

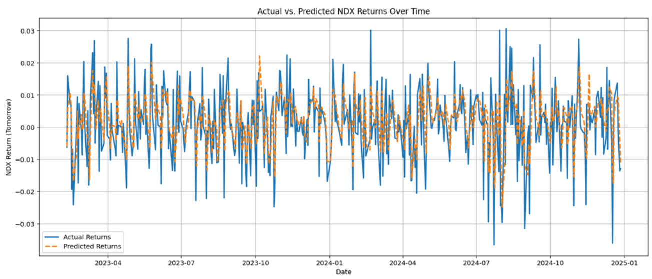

Fig 11. Trend line presenting model's prediction vs actual returns

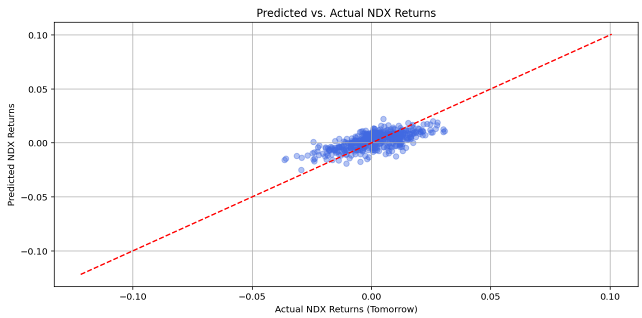

Fig 12. Scatter plot of predicted vs. actual NASDAQ-100 next-day returns.

B. *Classification Model:*

1) *Performance:* The model was trained using 80/20 partition based on time. The algorithms' hyperparameters were tuned three times (Table 3) to get the most accurate results along with early stopping to avoid overfitting. The algorithm stopped after 688 iterations providing the most optimal results (Figure 13). The model's performance was assessed using standard classification metrics such as precision, recall, F1-score, and accuracy. The final most accurate results are reflected in table 4. The model's predictions were accurate 74% of the times and the predictions in class 1 (bullish) were accurate 78% compared to a 71% in class 0 (bearish) based on F-1 score. All metrics present similar results which means the model performs consistently without being skewed towards a specific class.

Hyperparameter	1st Round	3rd Round
n_estimators	300	1000
max_depth	5	3
learning_rate	0.05	0.01
subsample	0.8	0.8
colsample_bytree	0.8	0.8
reg_alpha	0.1	0.1
reg_lambda	1.0	1.0
random_state	42	42
early_stopping_rounds	20	40

Table 3. Hyperparameter tuning rounds for classification model

Classification	Precision	Recall	F-1 Score	Support
0 (Stable/Bearish)	0.70	0.71	0.70	202
1 (Bullish)	0.78	0.77	0.78	272
Accuracy			0.74	474
Macro Average	0.74	0.74	0.74	474
Weighted Average	0.74	0.74	0.75	474

Table 4. Classification Performance Metrics by Class

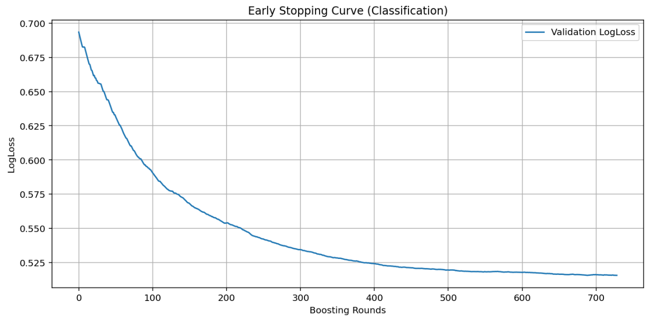

Fig 13: XGBoost early stopping cure (iterations needed)

2) *Features*: The importance of the features were extracted from the XGBoost trained model. The top 10 (figure 14) and bottom 10 (figure 15) were plotted to identify which contributed the most and least to the predictions.
The model relied mostly in the NASDAQ-100 3-day rolling average and the lagged returns from different companies. On the opposite side, macro indicators had low importance scores. These metrics might be more insightful in the long-term return predictive models as they don't vary daily unlike constituent stock returns.

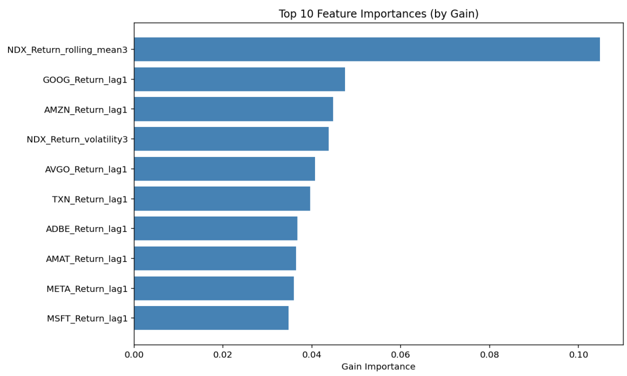

Fig 14. Top 10 features contributing to the model

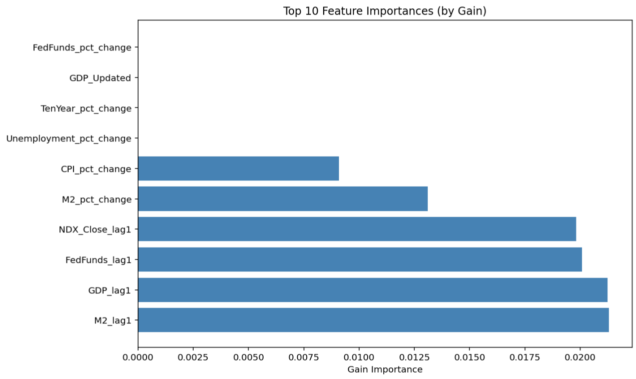

Fig 15. Bottom 10 features contributing to the model

3) *Prediction Results:* Using a confusion matrix (Figure 16), it is possible to identify the total of correct results – True positives and negatives – and the total of incorrect results – False positives and negatives. This supports the F1-Score results that the model is more accurate predicting correctly the bullish trends than bearish ones. The misclassification is even for both classes however the model still missed rallies 63 times and predicted 58 times rallies when the market went down.

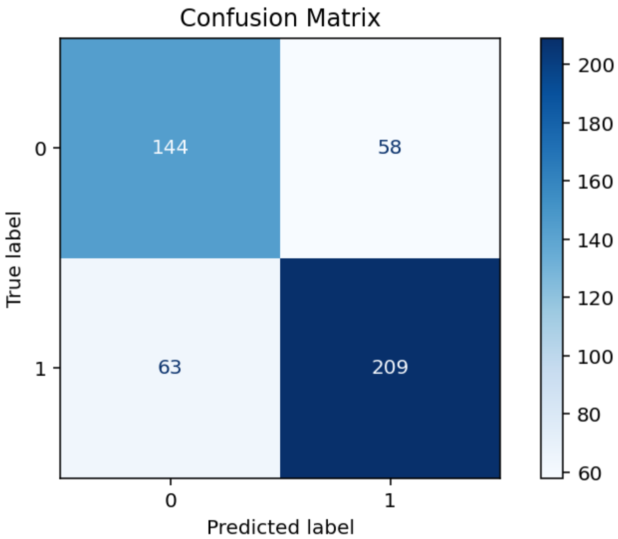

Fig 16. Confusion matrix displays True positives/false positives

The confidence vs correctness histogram (Figure 17) allows to understand how the model behaved when making decisions predicting class 1 (bullish). The model was very confident and accurate at predicting results when the probabilities were above 0.75 and below 0.2 however, the model struggled at predicting accurately between 0.4 – 0.7 suggesting the model is uncertain in this rage of confidence and the decisions are less reliable.

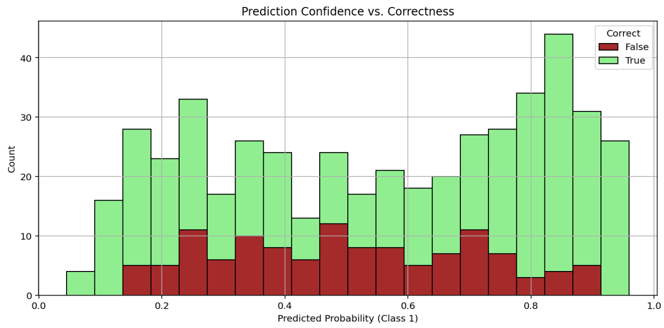

Fig 17. Prediction Confidence vs Correctness for class 1 predictions (bullish)

The main objective of this project, building a ML model to predict NASDAQ-100 daily returns and directional movement by using historical data from the top 20 markets in the index as well as macroeconomic indicators, was achieved by using Extreme Gradient Boosting (XGBoost).
Two ML models were trained and tested independently; a classification model to predict the index's movement that achieved a 74% accuracy, and a regression model to predict the index's variation on daily basis which achieved a R^2 of 46.09 and Root Mean Squared Error of 0.0081 proving moderate predictive power.

VI. Conclusions

A. *Conclusions*

This research has proven the feasibility of XGBoost ML to predict the returns and directional movement of NASDAQ-100 from historical market's data and macroeconomic indicators.
The classification model achieved results of 74% accuracy and proved to be more precise in predicting uptrends in the markets with a 0.78 F-1 Score for bullish movements.
The regression model performed well in predicting returns when the market did not have major dips or rallies. The model did not overfit, however in the periods where market was volatile it tended to be more reserved in the predictions. The daily returns predictions were in average off by 0.81 percentage points as measured by RMSE.
The macroeconomic indicators were not highly influential in the two models' predictions. In both models, the macroeconomic indicators were in the bottom features importance and did not provide important information to improve the models' accuracy.

B. *Limitations*

Using a single algorithm (XGBoost) does not allow to cross check results. By using other algorithms, it could be possible to compare which one captures the patterns better and provides more accurate predictions.
Macro indicators are one possible cause of the market's returns variation. To appropriately capture the market's movements more economic drivers need to be considered as they could help the model to forecast returns more accurately.

C. Future work

Model diversification: Implementing different algorithms to train different models would allow to compare results and understand if the data used is enough to predict more accurate results and which model would provide the best forecast.

Expanded dataset: Including all the markets in NASDAQ-100 could help to capture higher variance from the data and the models to predict more accurately.
Periodicity: When using macroeconomic indicators, using different prediction periods can be more insightful rather than daily considering that these indicators are often released on a monthly or quarterly basis.

VII. References

[1] E. Alpaydin, Introduction to Machine Learning, MIT Press, 2020.

[2] D. Oglesby, "Nasdaq (National Association of Securities Dealers Automated Quotations)," 2007. [Online]. Available: https://research.ebsco.com/linkprocessor/plink?id=d0646bc0-6154-33f4-98ee-dae5698f0d30. [Accessed 15 07 2025].

[3] J. Wang, "Analysis on the Application of Machine Learning Stock Selection Algorithm in the Financial Field," in *The 2021 International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy*, 2021.

[4] S. K. P. L. K. M. R. K. G. P. D. V. N. D. &. A. R. P. Bharath Raj Anand Kumar, "Evaluating the Performance of Diverse Machine Learning Approaches in Stock Market Forecasting," in *International Conference on Multi-disciplinary Trends in Artificial Intelligence*, 2023.

[5] M. V. Vardhan and G. Jaffino, "Stock Price Prediction using Machine Learning," in *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)*, Vellore, 2024.

[6] Y. A. Reddy, "Predictive Modeling of Financial Market Trends Using Advanced," *International Journal of Research Publication and Reviews,* vol. 5, no. 1, p. 2571, 2024.

[7] A. M. Aleksandr Medvedev, "Forecasting financial markets using advanced," in *E3S Web of Conferences*, 2023.

[8] IBM, "Introduction to CRISP-DM," IBM, 17 08 2021. [Online]. Available: https://www.ibm.com/docs/en/spss-modeler/saas?topic=guide-introduction-crisp-dm. [Accessed 16 07 2025].

[9] V. Rahangdale, XGBoost Explained: A beginner's Journey into Machine Learning, 2025.