Machine Learning for Nifty Trading

Aashutosh Bhattad, Shubham Kotal, Gaurav Kumar
aashutosh.bhattad@gmail.com, shubhkotal@gmail.com,
purushottam.adarshi@gmail.com

Abstract

Building a framework to trade derivatives in NSE: Nifty 50 index using Machine Learning and Deep Learning framework. The framework is designed and backed up by Daily data (OHLC) of Nifty and market sentiment data synthesized using NLP and NLU models. An efficient risk management framework is incorporated by hedging active positions.

Keywords: *Machine Learning, Feature Selection, Deep Learning, Nifty*

1. Introduction

The NSE: Nifty50 Index is a synthetic index formulated by taking a market-cap wise weightage of top 50 Indian companies (based on market cap). This Index is tradeable using derivatives (futures and Options) through the National Stock Exchange, India. We would use Machine Learning and deep learning methodologies to predict the daily direction of this index and place trades in derivates to derive monetary benefit from the predicted direction.

2. Objectives

Below are the key objectives that we would like to accomplish during this capstone project:

- Predict the future prices and price movement of Nifty 50 (Synthetic Index)
- Use NLP pre trained model (BERT) to strengthen and improvise the accuracy of the predicted direction of the index.
- Incorporate market sentiment into model to add context to the prediction.
- Develop framework for trade automation using python
- Hedging of taken positions to reduce risk

3. Methodology

The basic framework of the workflow is to use the OHLC (open-high-low-close) data and create multiple useful parameters i.e., feature engineering that can help to predict the next day's direction of the market based on the historical data.

The possible feature would be as last 7 days high, last 7 days low, last day high, last day low, last week high, last week low, last 7 days range, and yesterday range. On the feature, we will be using statistical methods to check the relevance of the data set.

Methods such as correlation check, checking the correlation score of the independent variable with respect to the dependent variable. Also, the multicollinearity between the independent variables, checking the VIF (variance inflation factor) which measures the covariance. Normalization of the data set, bringing the range of features to the same scale if required. This helps to reduce the overfitting and underfitting of the model while training. The imputation method will be used in case of missing values. Based on the best features we will try to create a model with all possible combinations and check the importance of the feature suggested by the model. To evaluate the model performance will check the accuracy of the model i.e., RMSE (Root mean square error). The model which shows the lowest RMSE score will be chosen as the ideal model. To cross-check the model performance on a different set of validation data, we can perform k-cross-validation as well.

For extra confirmation to predict the next day's direction of the market, we will be using the NLP models (Bert) to check the sentiment of the market based on recent trending news of the market. The sentiment prediction will be giving an extra edge for understanding the market strength and direction. Combining the above two decisions will be predicting the next day's direction.

3. Ongoing work

Part 1: (Code 1)

The basic framework of the workflow is to use the OHLC (open-high-low-close) data and create multiple useful parameters i.e., feature engineering that can help to predict the next day's direction of the market based on the historical data.

After Importing required libraries.

1. Data collection: Downloading data from NSEPY for the symbol NIFTY. Where start date is (2010,1,1) and end date is (2022,8,27)

2. Feature Extraction & Engineering

   Created many required features from the OHLC data. Such as nifty_close_next_day, nifty_open_next_day, nifty_day_gain, nifty_day_chng

3. Data cleaning: Dropping few redundant columns and formatting dates as per required.

4. A/B Testing (Train & Test Split) : Creating data sets to train and validate the model results

5. Model Building: Creating two leading models in machine learning today, one based on bagging method and one on boosting Method.

   a. Random Forest Model
   b. XG Boost Model

   The models been created with added variable refinement process.

6. Model Results: Checking the model result. Using RMSE Score to evaluate model performance. The Current RMSE score is close to 1.

Part 2: (Code 2)

For extra confirmation to predict the next day's direction of the market, we will be using the NLP models (Bert) to check the sentiment of the market based on recent trending news of the market. The sentiment prediction will be giving an extra edge for understanding the market strength and direction. Combining the above two decisions will be predicting the next day's direction.

Using a pre trained model FinBERT for analyzing the sentiments.

FinBERT is a pre-trained natural language processing (NLP) model that analyses the sentiment of financial writing. It is created by fine-tuning the BERT language model for financial sentiment categorization by further training it in the finance domain using a large financial corpus.

1. Importing required libraries and downloading required pre trained model files.4
   Link to download pre trained model: FINBERT

2. Defining function required for cleaning the text data and further adding extra characters used in the operation of pre trained model operation.
3. Giving labels to the classes
4. Calling the pre-trained model
5. Sample testing
6. Results:

Example:

1. Reliance trades will go high

Prediction: Neutral
Score: 14%

2. Reliance planning to expand jio networks.

Prediction: Positive
Score: 84%

Based on the above example, the testing results shows clever predictions been done.
The pre trained model is been trained well and is able to classify the matter well.
More try and checks were been done, before completely accepting the model.

Conclusion:

As per the above explanation we were able to get the satisfied results. More refinements will be added going forward.

Project Github Link: https://github.com/shubhamkotal/WQU-Capstone/

References

1. K.K Suresh Kumar and Dr N.M Elango, “Performance analysis of stock market prediction using Artificial Neural Network”, Global Journal of Computer science and Technology, Volume 12 Issue 1 Version 1.0 January 2012.

2. Prakash Ramani and Dr.P. D Murarka, “Stock market prediction using Artificial Neural Network”, International journal of advance research in computer science and software engineering, Volume 3 Issue 4, April 2014.
3. Takashi Kimoto and Kazoo Asakawa, Morio Yoda and Masakazu Takeoka, “Stock market prediction system with modular neural network”
4. Kumar, Gourav & Sharma, Vinod. (2016). Stock Market Index Forecasting of Nifty 50 Using Machine Learning Techniques with ANN Approach. International Journal of Modern Computer Science. 4. 22 -27.
5. Rasekhschaffe, Keywan and Jones, Robert, Machine Learning for Stock Selection (February 8, 2019). Financial Analysts Journal, vol. 75, no. 3 (Third Quarter 2019), Available at SSRN: https://ssrn.com/abstract=3330946
6. Ritter, Gordon, Machine Learning for Trading (August 8, 2017). Available at SSRN: https://ssrn.com/abstract=3015609 or http://dx.doi.org/10.2139/ssrn.3 015609
7. Koshiyama, Adriano and Firoozye, Nick and Treleaven, Philip, Algorithms in Future Capital Markets (January 29, 2020). Available at SSRN: https://ssrn.com/abstract=3527511 or http://dx.doi.org/10.2139/ssrn.3 527511Time-Series and Cross – Sectional Stock Returns Forecasting: New Machine Learning Methods
8. Chaudhuri, Koushiki Dasgupta. “Time Series Forecasting with Extre me Learning Machines -.” Analytics Vidhya, 13 Dec. 2021, www.analyticsvidhya.com/blog/2021/12/time-series-forecasting-with-extreme-learning-machines.
9. Shirosaidev. “GitHub - Shirosaidev/Stocksight: Stock Market Analyzer and Predictor Using Elasticsearch, Twitter, News Headlines and Python Natural
   Language Processing and Sentiment Analysis.” GitHub, 27 Mar. 2022, github.com/shirosaidev/stocksight.
10. López de Prado, Marcos and López de Prado, Marcos, The 10 Reasons Most Machine Learning Funds Fail (January 27, 2018). Journal of Portfolio Management, Forthcoming, Available at SSRN: https://ssrn.com/abstract=3104816 or http://dx.doi.org/10.2139/ssrn.3 104816

11. Rai, Akhilesh. “Trading Strategy with Predictive ML Model of Nifty | by Akhlesh Rai | Medium.” Medium, 1 July 2019, medium.com/@akhlesh.rai/trading-strategy-with-predictive-ml-model-of-nifty-bf71ee16861.