Stock Market Analysis with CNN

DATA 6150 Individual Project

Ramy Qranfal
School of Data Science
Wentworth Institute of Technology
Boston, Massachusetts, USA
rqranfal@gmail.com

ABSTRACT

Financial markets generate a large volume of historical data that can reveal meaningful insights in terms of price trends and index movement. Understanding these patterns and the relationship between market indices is important when creating predictive models. This project studies five major U.S. stock indices and examines their correlation over time. Feature importance analysis using XGBoost identifies key indicators and market variables that contribute to price movement. A convolutional neural network (CNN) is used to classify S&P 500 next day direction using historical data and associated features. Together, these methods provide meaningful analysis into understanding index patterns and the potential use of machine learning in market prediction.

KEYWORDS

Stock Prediction, Historical Data, Machine Learning, CNN, Market Indices

1 Introduction

Stock market indices are important factors of economic performance, and predicting their movements is a challenging problem in finance. This project analyzes historical trends in major U.S. stock indices and examines how they move together. It also investigates which features from past price data are most important for predicting closing prices using an XGBoost model. Additionally, a CNN model is used to classify next day's movement as up or down, as prior research has shown strong results in stock prediction tasks [1].

2 Data

2.1 Source of dataset

The dataset used in this project is the CNNpred Stock Market Dataset, downloaded from the UCI Machine Learning Repository, available at, UCI Machine Learning Repository. UCI is a credible source and is used by many as an academic source for machine learning research. The datasets contain daily historical data for major U.S. stock indices from 2010 to 2017 and was generated by collecting available market information such as closing prices, technical indicators, commodity prices, and exchange rates. Data was cleaned and gathered by the dataset creators to support stock market research.

2.2 Characters of the datasets

The dataset consists of five CSV files; each file corresponds with a major U.S. stock index containing daily data from 2010-2017. All files share the same structure, with over 80 features per day and around 1984 rows per index. Each row represents a single trading day with numerous features such as price, technical indicators, commodities, currency exchange rates, global market indices, and futures data. The table below provides an example of some features in a grouped summary, containing the feature types and sources.

Feature Group	Example Features	Type	Source
Price and Technical Indicators	Close, Volume, MOM-1	Primitive, Technical Indicators	TA-Lib, Yahoo Finance
Commodities	Oil, Gold, Silver	Commodity Prices	FRED, Investing.com
Currency Exchange Rates	USD-JPY, USD-EUR, USD-CAD	Exchange Rates	Yahoo Finance, Investing.com
Global Market Indices	RUSSELL, NYSE, HIS, DJI	World Indices	Yahoo Finance
Futures Data	S&P-F, NASDAQ-F	Future Contracts	Investing.com

Data was prepared by converting date column into datetime format and creating a binary up/down target variable for CNN classification on the criteria that the variable is 1 if next day price higher than today, otherwise 0. Numerical features were normalized before training with the CNN model, and all 5 datasets were merged for analysis.

3 Methodology

3.1 XGBoost

XGBoost is a gradient boosted decision tree model that is used to evaluate which features are most important for predicting stock index closing prices. It assumes past numerical features can help predict future outcomes. The advantages of this model are fast training, capability of handling many features, and providing a clear feature importance score. The disadvantage is that the model does not look at the data in order, meaning it cannot naturally understand how yesterday affects today. I chose the XGBoost model because of its fast training and to identify which variables in the dataset are most important towards predicting next day's index prices. This method was implemented in Python using the xgboost.XGBRegressor module, with default hyper parameters.

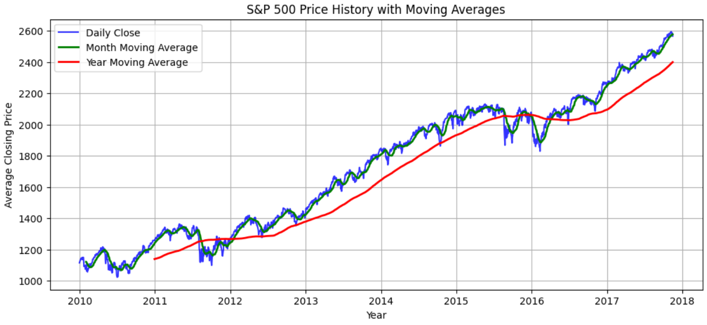
3.2 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) works by sliding filters across sequences of past data to detect short term patterns that can help predict next day movement. CNNs assume that recent historical features and indicators can help predict future direction of price. The model advantage is that it automatically learns these patterns in the training phase; however, the disadvantages are that they require more tuning via hyperparameters and are harder to interpret compared to other models. I chose CNN because previous research showed a strong performance in the stock movement classification task. The model is built using python, using the PyTorch library, with normalized inputs and a sliding window structure to improve the neural network ability to predict patterns.

4 Results

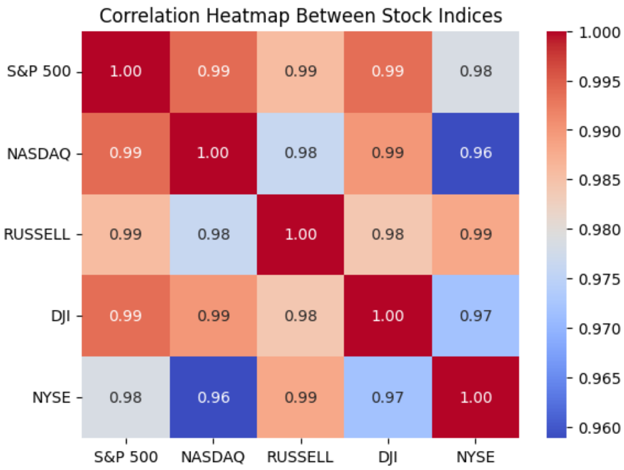
4.1 Historical Price Trends of S&P 500

A line chart displaying daily closing prices with monthly and yearly moving averages was created to understand S&P 500 long-term patterns. The chart reveals a consistent upward trend from 2010 to 2017, with prices rising from around 1100 to 2600. When the daily prices is above the yearly moving average, it indicates a bull pattern, and if below indicates a bear. The early 2016 dip, where prices fell below the yearly moving average, indicated a short bearish period before recovery. The moving averages smooth out daily prices going up and down, making long term patterns easier to identify.



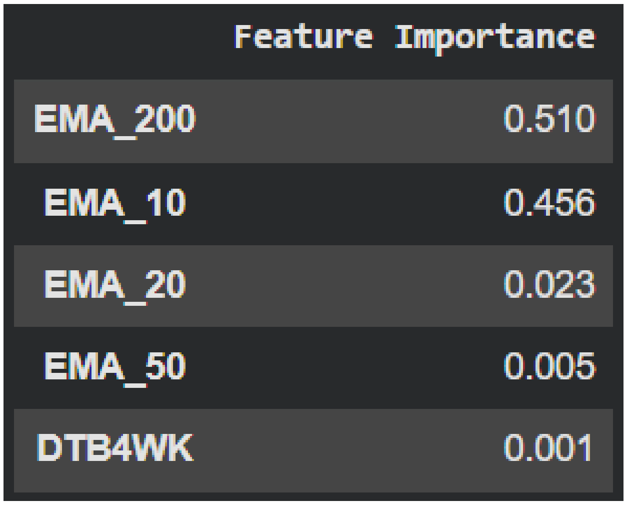
4.2 Stock Indices Closing Price Correlation

A correlation matrix was created to show how closely stock indices closing prices move together. The heatmap contains all five stock indices of S&P 500, NASDAQ, RUSSELL, DJI, and NYSE, and the variable being closing price. All stock indices have a correlation value greater than or equal to 0.96, indicating these major U.S. stock indices move very closely together.



4.3 Most Important Features For Predicting Future Prices

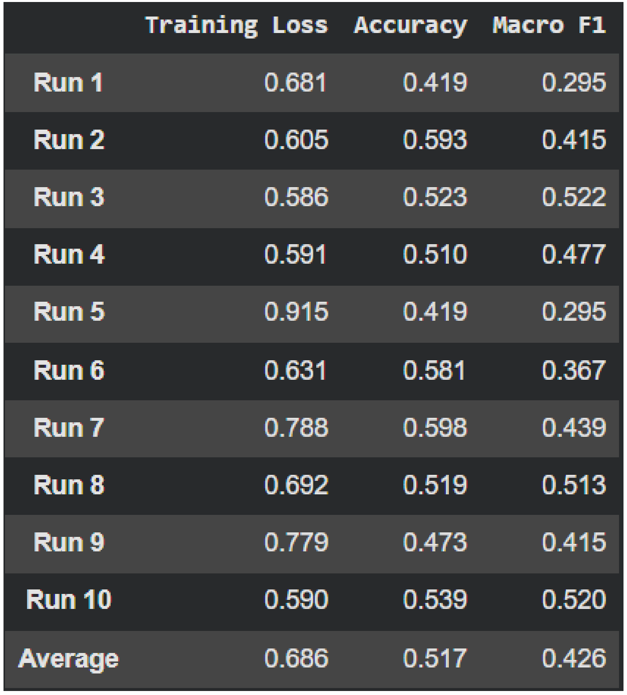
A XGBoost model was trained to predict future prices, using only numerical input data. The training data consisted of daily data in the S&P 500 index from 2010-2016, and the testing data for the year 2017. The model was trained with 100 decision trees with max depth of 5, and resulted in a MSE of 193. The top five features include four technical indicators and one economic variable, with their feature importances shown below.



	Feature Importance
EMA_200	0.510
EMA_10	0.456
EMA_20	0.023
EMA_50	0.005
DTB4WK	0.001

4.4 Next Day Movement Classification with CNN

A CNN model was trained to classify the next day's movement as up or down using the past 60 days' data. The training data was split into training windows, where the training windows are the past 60 days inputs features, and the output is the label 0 for down or 1 for up for the next day. The model consists of a structure with two convolutional layers with batch normalizations and ReLU activation, adaptive average pooling, and two fully connected layers with ReLU and dropout. The table below summarizes the results 10 runs, each trained for 50 epochs with different random seeds. The model achieved an average training loss of 0.686, accuracy of 0.517, and macro F1 of 0.426, performing slightly better than randomly guessing whether a stock goes up or down the next day.



	Training Loss	Accuracy	Macro F1
Run 1	0.681	0.419	0.295
Run 2	0.605	0.593	0.415
Run 3	0.586	0.523	0.522
Run 4	0.591	0.510	0.477
Run 5	0.915	0.419	0.295
Run 6	0.631	0.581	0.367
Run 7	0.788	0.598	0.439
Run 8	0.692	0.519	0.513
Run 9	0.779	0.473	0.415
Run 10	0.590	0.539	0.520
Average	0.686	0.517	0.426

5 Discussion

One limitation of my project is that many features showed low feature importance with the XGBoost model, suggesting that the dataset contained noise or redundant variables that may have not been meaningful towards price prediction. Along with that, the limitation of the CNN model is that it was unstable depending on different weight initialization, as 3 out of the 10 runs ended up predicting one single class. The average macro f1 score of 0.426 shows difficulty in predicting the next day's movement. Future works could use more hyperparameters tuning and different transformers architecture.

6 Conclusion

This project explored machine learning methods for predicting stock market behavior using CNN for daily movement classification and XGBoost for price prediction. Data analysis reveals high correlations (0.96-0.99) between all five major U.S. stock indices, showing they move together and suggesting that predicting one index might generalize to the other. The moving average analysis showed a constant upward trend in prices from 2010-2017, showing a bull market after the 2008 financial crisis.

The CNN model achieved an average macro of 0.426 across 10 runs on 50 epochs each, while XGBoost achieved an MSE of 193. The CNN model performed slightly better than random guessing, confirming that short-term market direction is still a challenging problem. Feature importance analysis from XGBoost revealed that many indicators have minimal impact on predictive value, suggesting simpler models with fewer features to perform better or equally well.

For the real-world application, these results suggest that focusing solely on technical indicators and historical patterns is not enough for making market decisions. Investors and traders should use such predictive models as a tool rather than a final decision maker.

ACKNOWLEDGMENTS

I would like to thank my professor for the guidance in this project.

REFERENCES

[1] Ehsan Hoseinzade and Saman Haratizadeh. 2018. CNNPred: CNN-based stock market prediction using several data sources. arXiv preprint arXiv:1810.08923. https://doi.org/10.48550/arXiv.1810.08923

[2] CNNpred: CNN-based stock market prediction using a diverse set of variables [Dataset]. 2019. UCI Machine Learning Repository. https://doi.org/10.24432/C55P70