Time Series Analysis on Financial Data from the NYSE

Sanchit Singhal, Gaurav Lalwani

Introduction

The modeling of stocks traded on the NYSE into binary classes of gainers, shares that the model expects to appreciate, and losers, shares that the model predicts will lose value, over a time period is a challenge that can be extremely rewarding. The motivation behind this work is to help investors identify financial opportunities and, given a certain risk threshold (probability of price movement), be able to invest in the stock market with more confidence based both on fundamentals of a company as well as the share price history itself. Other desired benefits, apart from predicting price fluctuations for individual securities include the fact that the model should be able to aggregate data for all stocks, given it calculates the probability for each stock anyway, and provide insight into what the overall market should look like across various indexes and sectors (i.e. SP 500, the tech sector, etc), thus further aiding investors to park their money more intelligently through index/mutual funds if a more diversified portfolio is desired.

There are a plethora of existing solutions for the purpose of predicting share prices. Since at least the 1950s, companies have been building models to predict the future value of the stock market. Historically, these analysis could be categorized into fundamental analysis, which looks at the underlying company and its performance, and technical analysis, which focuses on patterns and trends of the past prices. The existing methodology(Porshnev, Redkin, and Shevchenko 2013) mostly refrains from combining the two approaches. Further, the literature we found on the topic(Porshnev, Redkin, and Shevchenko 2013; Patel et al. 2015; Devi, Sundar, and Alli 2013) seems to solely focus on the prediction of share prices in the short term - intraday trading or at most daily charts. The team believes there exists a gap in the literature of modelling a longer time frame.

The project aims to build models that help classify stocks into two classes - gainers and losers. Although this itself is nothing new, the team will aim to predict these fluctuations daily - as opposed to intraday probabilities - thereby enabling a new approach to solve an existing challenge. The team also plans to augment historical price trends with fundamentals of the company to achieve a more holistic view of the security. Given the project parameters, the team believes this model will be useful for investors who want to invest for the longer term, rather than benefit from rapid, day-trading. Most current technologies use automation software to buy/sell at a faster pace, but in contrast, our project will aim to identify investment opportunities that will appreciate in the slightly longer term.

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Related Works

Regression Techniques in Financial Models

- Machine learning in prediction of stock market indicators based on historical data and data from Twitter sentiment analysis. (Porshnev, Redkin, and Shevchenko 2013)
- Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques (Patel et al. 2015)
- Predicting stock market index using fusion of machine learning techniques (Patel et al. 2015)
- Evolving Least Squares Support Vector Machines for Stock Market Trend Mining (Yu et al. 2009)

All the above mentioned literary articles had SVM as the common regression technique to predict stock prices. The articles didnt focus or display information regarding other methods such as Decision tree and KNN - even though it was just vaguely mentioned in the second reference. We propose to use all the regression techniques and display accuracy metrics to select the best possible technique. None of the articles worked on enhancing techniques such as dimension reduction or subset applications to reduce the variance and thus keeping minimal bias. We propose to work on smaller data sets that keep only the essential subsets of the data. Also the articles focused more on prediction of stock market prices and none of the papers classified the data into groups to predict the possibility of the market opening up or down for a given day. We propose to work on classification of data so as to predict the open price to be below or above the close price at the end of the previous day.

Ensemble Learning in Financial Modeling

- Chinese Stock Index Futures Price Fluctuation Analysis and Prediction Based on Complementary Ensemble Empirical Mode Decomposition (Chen and Pan 2016)

- Comparison of individual, ensemble and integrated ensemble machine learning methods to predict Chinas SME credit risk in supply chain finance (Zhu et al. 2017)
- Stock market prediction with multiple classifiers (Qian and Rasheed 2007)

Ensemble learning is a popular technique to gain prediction accuracy. These works discuss model architecture that have been applied and some strengths and weaknesses that were found. These analysis have used a combination of artificial neural networks, decision trees, and k-nearest neighbors and through appropriate collaboration achieved 65 percent correct results. From the first reference, it is also important to note that we must consider the type of market when selecting the modelling tools - markets in an emergent state, such as China, will use different analysis than mature ones such as the NYSE. That being said, some learnings from these works can be transferred over to our project: random subspace boosting seemed to provide a substantial increase in accuracy. Our work is different from these works because we plan on building various models, creating ensembles from them over various combinations, and then optimizing our model to select optimal parameters. None of these papers discuss building the ensembles and the effect of different inputs on the results.

Deep Learning Techniques for Financial Models

- A deep learning framework for financial time series using stacked auto-encoders and long-short term memory (Bao, Yue, and Rao 2017)
- Stock prediction using deep learning (Singh and Srivastava 2017)
- A new hybrid constructive neural network method for impacting and its application on tungsten price prediction (Muzhou et al. 2017)
- A review of unsupervised feature learning and deep learning for time-series modeling (Längkvist, Karlsson, and Loutfi 2014)
- Stock price prediction using LSTM, RNN and CNN-sliding window model (Selvin et al. 2017)

Apart from the techniques mentioned in the above articles, we might also be using recurrent neural network to view the result outcome. Also apart from the mentioned convolution methods we aim to change the pooling layer size to observe the accuracy for different sizes so as to select the best possible method. Although the team is unsure whether we will utilize deep learning in the models, it was important to note some related work that has already been done in the space. If we do find out that certain types of neural network, such as RNN, are useful - the team will explore this area further.

Time Series Analysis for Financial Data

- An Effective Time Series Analysis for Stock Trend Prediction Using ARIMA Model for Nifty Midcap-50 (Devi, Sundar, and Alli 2013)
- Financial time series forecasting using support vector machines (Kim 2003)
- Twitter mood predicts the stock market (Bollen, Mao, and Zeng 2011)

Most previous works focus on using artificial neural networks (ANNS) for the challenging task of financial time-series predictions. However, the third reference points out that these models can have limitations in that learning patterns is difficult when there is a lot of noise and complex dimensionality. Our work is different from these works as we plan to apply Recurrent Neural Network with Long Short-Term Memory so that the model is better able to retain information from previous nodes.

Methods

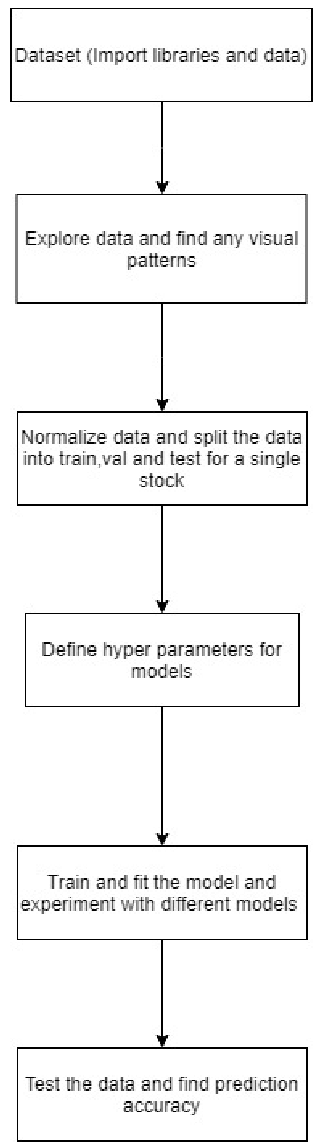

Figure 1: Workflow representing the entire process of the experiment

The procedure aims at extracting data from the database and building a machine learning model that involves 4 major steps:

Loading :
The data from the csv file is loaded into the database using Python. The data is loaded using the python package pandas in the form of a Dataframe. The packages required for the analyses are imported in the beginning. The values of hyperparameters were defined prior to the analyses. Data exploration was performed to get a better grasp of the dataset.

Filtering :
The Dataframe was then filtered to remove unused columns(volume and stock symbol) which wont be used in the analyses. The value of train, validation and test sets were defined. The sets were then scaled and normalized to be used later while fitting the model.

Visual Analysis :
Plots of volume v/s time and price v/s time were drawn to find any visual patterns and relations within the data. For this purpose one specific stock was chosen manually and the analyses involved this particular stock. The model was built on the entire data but the analyses was carried out just to depict the relation w.r.t a single stock.

Build Machine Learning Model :
First the model was trained using the training dataset. Then the model was fitted using the validation dataset. Finally it was tested for accuracy using the test dataset. The model was optimized using different hyper-parameters using the test classifications predictions as further experiments to determine a final model.

The features used in the model are the open, close, low and high. The algorithm used in the model is RNN.

Experimental Design

:

The main purpose of the experiments was to determine which of the hyper-parameters yield the highest prediction accuracy. The motive is to find the best possible combination that yield highest accuracy without over fitting the data. The following three experiments were conducted. The dataset is the real world data related to the new york stock exchange from 2010 till 2016. The data was available on a site kaggle.com. The dataset contains the open, close, high and low prices of different stock prices during each day for each of the seven years. We conducted our experiments on the same dataset each time.

Optimal Number of Neurons :
The purpose of this experiment was to test the model predictive power on various values of number of neurons to determine one that would yield the best test classification rate. Initially, the model RNN with LSTM was chosen with 200 neurons but the team wanted to further optimize the neural network. This hyper parameter will be chosen based on the evaluation metric of correct classification which we have denoted in our model as the open price of a stock minus the close price from the previous day.

Optimal Number of Layers :
The purpose of this experiment was to test the model predictive power on various values of number of layers to determine one that would yield the best test classification rate. Initially, the model RNN with LSTM was chosen with 2 layers but the team wanted to further optimize the neural network. This hyper-parameter will be chosen based on the evaluation metric of correct classification which we have denoted in our model as the open price of a stock minus the close price from the previous day.

Optimal Batch Size :
The purpose of this experiment was to test the model predictive power on various values of batch size to determine one that would yield the best test classification rate. Initially, the model RNN with LSTM was chosen with a batch size of 50 layers but the team wanted to further optimize the neural network. This hyper-parameter will be chosen based on the evaluation metric of correct classification which we have denoted in our model as the open price of a stock minus the close price from the previous day.

All selected hyper-parameters were chosen via the test rate because it gives us the best performing model. Although, the team could have chosen to pick through an indirect method but this seemed impractical as we had the true values of the predicted dates.

Experimental Results

Our main findings from the results were that a recurrent neural network (RNN) was the best at modeling time series for financial data. Further, a gated RNN with Long Short-Term Memory (LSTM) provides significant predictive power gain to the model. Although the hyper parameters number of neurons, number of layers, and batch size were optimized, we do not expect any significant difference as the initial runs were set with fairly standard parameters. The optimal number of neurons for our model was X. The optimal number of layers was Y. Finally, the best batch size to train on was Z.

A few lines of research that would be interesting to follow this analysis up would be the addition of ensemble learning. During the team's literature review, ensemble learning (in relation to Random Forests) were highly recommended. It would be interesting to see the results when our RNN model could be improved upon by combining several network or by adding convolution layers. Another area our experiments fail to cover is the inclusion of the volume feature. I would expect the network to learn the impact of volume levels on share prices but we were not able to consider this path due to time constraints.

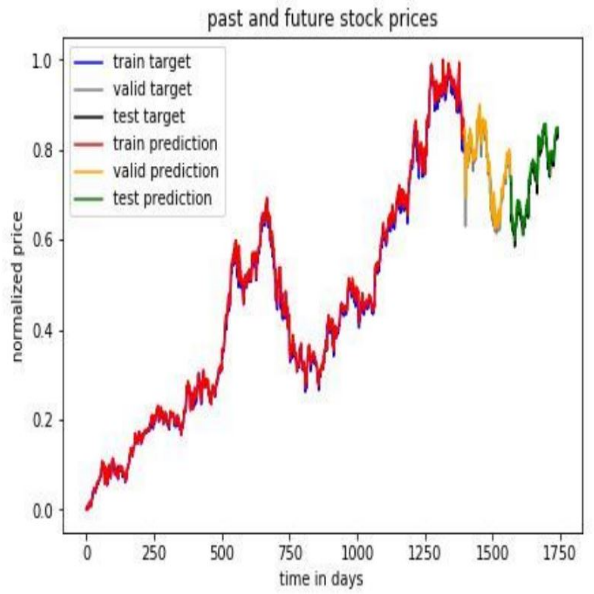

Figure 2: Normalized Price Predictions for Training/Validation/Test sets

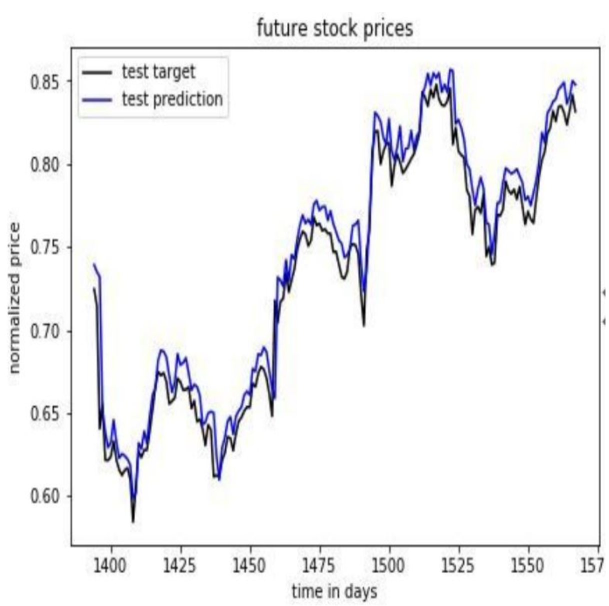

Figure 3: Normalized Price Predictions for Test Set vs True Values

Conclusion

Our analysis show that a recurrent neural network work strikingly well for time series analysis. We were able to gain a relatively high percentage of correct classifications of gainers and losers for each share daily. Adding a gate to our RNN, via LSTM, was tremendously useful. The team believes this aids the model weights to remember history and learn not just from neighboring neurons but more similar to a biological brain, in the sense that it is learning through experience.

References

Bao, W.; Yue, J.; and Rao, Y. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one* 12(7):e0180944.

Bollen, J.; Mao, H.; and Zeng, X. 2011. Twitter mood predicts the stock market. *Journal of computational science* 2(1):1–8.

Chen, R., and Pan, B. 2016. Chinese stock index futures price fluctuation analysis and prediction based on complementary ensemble empirical mode decomposition. *Mathematical Problems in Engineering* 2016.

Devi, B. U.; Sundar, D.; and Alli, P. 2013. An effective time series analysis for stock trend prediction using arima model for nifty midcap-50. *International Journal of Data Mining & Knowledge Management Process* 3(1):65.

Kim, K.-j. 2003. Financial time series forecasting using support vector machines. *Neurocomputing* 55(1-2):307–319.

Längkvist, M.; Karlsson, L.; and Loutfi, A. 2014. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters* 42:11–24.

Muzhou, H.; Taohua, L.; Yunlei, Y.; Hao, Z.; Hongjuan, L.; Xiugui, Y.; and Xinge, L. 2017. A new hybrid constructive neural network method for impacting and its application on tungsten price prediction. *Applied Intelligence* 47(1):28–43.

Patel, J.; Shah, S.; Thakkar, P.; and Kotecha, K. 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications* 42(1):259–268.

Porshnev, A.; Redkin, I.; and Shevchenko, A. 2013. Machine learning in prediction of stock market indicators based on historical data and data from twitter sentiment analysis. In *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*, 440–444. IEEE.

Qian, B., and Rasheed, K. 2007. Stock market prediction with multiple classifiers. *Applied Intelligence* 26(1):25–33.

Selvin, S.; Vinayakumar, R.; Gopalakrishnan, E.; Menon, V. K.; and Soman, K. 2017. Stock price prediction using lstm, rnn and cnn-sliding window model. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, 1643–1647. IEEE.

Singh, R., and Srivastava, S. 2017. Stock prediction using deep learning. *Multimedia Tools and Applications* 76(18):18569–18584.

Yu, L.; Chen, H.; Wang, S.; and Lai, K. K. 2009. Evolving least squares support vector machines for stock market trend mining. *IEEE Transactions on evolutionary computation* 13(1):87–102.

Zhu, Y.; Xie, C.; Wang, G.-J.; and Yan, X.-G. 2017. Comparison of individual, ensemble and integrated ensemble machine learning methods to predict chinas sme credit risk in supply chain finance. *Neural Computing and Applications* 28(1):41–50.