

Using neural network techniques to correlate past and future large-cap stock returns

KENNETH ALLEN, Armstrong State University, United States of America

JEFFREY YOUNG, Armstrong State University, United States of America

Quantitative analysis of the stock market to predict future behavior has long been a very active area of study with clear applicability, both for profit-seeking and informing macroeconomic policy. We apply support neural network techniques to show that simple daily total return data can be shown to be correlated with future performance. Although our method is not immediately applicable for predicting beyond the current moment in time, it demonstrates that the potential for such models does exist.

Additional Key Words and Phrases: Machine learning, Neural networks, Neuroph, Stock prediction, S&P 500

ACM Reference Format:

Kenneth Allen and Jeffrey Young. 2017. Using neural network techniques to correlate past and future large-cap stock returns. 1, 1 (December 2017), 5 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 BACKGROUND

Predicting movement of the stock is a longtime attractive topic. The purpose of this research venture was to explore the usefulness of S&P 500 data in relation to the use of machine learning methodology as a predictive model. The S&P 500 Index is a free float-adjusted market capitalization-weighted stock market index in the United States. It is used to record and monitor daily changes of the largest companies of the American stock market and is the main indicator of the overall market performance in the United States. Historically, many methods of S&P 500 analysis have been implemented with varying results but, overall, three analysis types have been explored most often: Naïve Bayesian, Support Vector Machines, and Neural Networks. Table 1 gives some values for the success of historical efforts.

1.1 Feature Selection

Feature analysis for all three common types of S&P 500 predicates have been similar. The most common features extracted from S&P 500 data have been daily returns, NASDAQ daily return, S&P 500 momentum, and S&P 500 rate-of-change. For this project, we used daily total returns.

Define $c(t)$ as the weighted average closing price of all stocks in the index for day t .

- S&P 500 daily return: $\frac{c(t)-c(t-1)}{c(t-1)}$.
- NASDAQ daily return: as above, but with weights and stocks from the NASDAQ market.
- S&P 500 momentum ($n = 4$): $\frac{c(t)-c(t-4)}{c(t-4)}$.
- S&P 500 rate-of-change ($n = 4$): $\frac{c(t)}{c(t-4)}$.

Authors' addresses: Kenneth Allen, Armstrong State University, Department of Computer Science & Information Technology, Savannah, GA, 31419, United States of America, ka3878@stu.armstrong.edu; Jeffrey Young, Armstrong State University, Department of Computer Science & Information Technology, Savannah, GA, 31419, United States of America, jy8672@stu.armstrong.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2017 Copyright held by the owner/author(s).

XXXX-XXXX/2017/12-ART

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Table 1. Historical S&P predictor accuracy.

Naïve Bayesian	60.38%
Linear SVM	59.79%
RBF SVM	62.51%
Polynomial SVM	59.43%
Neural Network	65.08%

- S&P 500 momentum ($n = 3$): $\frac{c(t)-c(t-3)}{c(t-3)}$.
- S&P 500 rate-of-change ($n = 3$): $\frac{c(t)}{c(t-3)}$.

2 DATA PROCESSING

2.1 Data

The economics department of Armstrong State University has a dataset spanning over 16 years of daily total returns for each stock listed in the Standard & Poor's 500 Index, which consists of 505 stocks from 500 of the largest companies in the world, measured by total market capitalization. We were given limited access to this data to perform the experiments for this project. The data takes the form of dated percentage daily total returns and occupies about 23.8 mibibytes (25 million bytes).

For each test, 10% of the data was randomly removed and used as a test set while the remaining 90% was used as a training set.

2.2 Training Labels

Labels for this undertaking were calculated from the excluded data from the test period, which was chosen as the final year of the data: March 22, 2016, to March 21, 2017. Each stock was labeled with whether its total return exceeded that of the S&P 500 index during that period.

2.3 Classifier

Our system uses the Neuroph library to implement neural network training and classification. It interacts with its programmatic Java API to add features like statistics tracking and training models with different sensitivities. We tested different numbers of layers and layer sizes, and achieved our best results with a single hidden layer of 30 neurons, connected to an input layer of 4075 features and an output layer of a single neuron. Parameters like maximum error, maximum optimization passes, and starting weight randomization were taken from values recommended by the library authors.

2.4 Prediction

Prediction quality was measured with a bevy of statistics. Accuracy is simple to calculate ($\frac{\text{correct classifications}}{\text{tested elements}}$).

For binary criteria, more informative values can be generated. Use TP , FP , FN , and TN , to represent true positives, false positives, false negatives, and true negatives, respectively. Especially important are true positive rate ($TPR = \frac{TP}{TP+FN}$), true negative rate ($TNR = \frac{TN}{FP+TN}$), positive predictive value ($PPV = \frac{TP}{TP+FP}$), and negative predictive value ($NPV = \frac{TN}{FN+TN}$).

The F_1 score is calculated as the harmonic mean of TPR and PPV :

$$F_1 = \frac{2}{\frac{1}{TPR} + \frac{1}{PPV}}$$

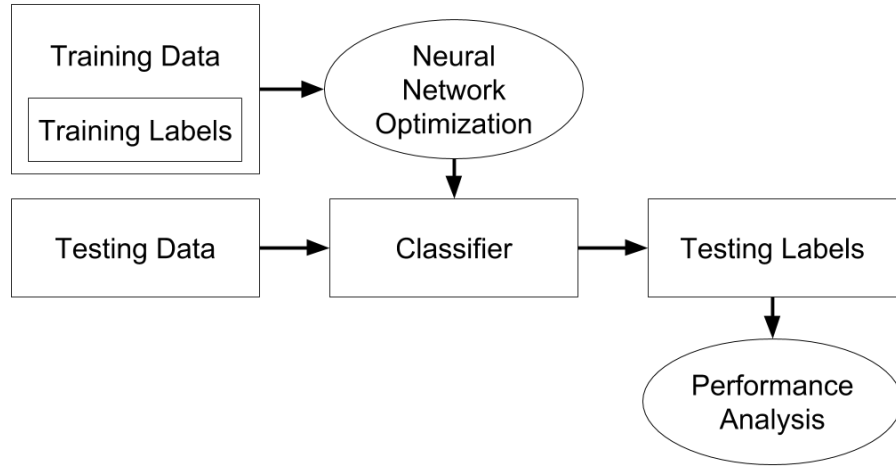


Fig. 1. Block diagram of classifier system.

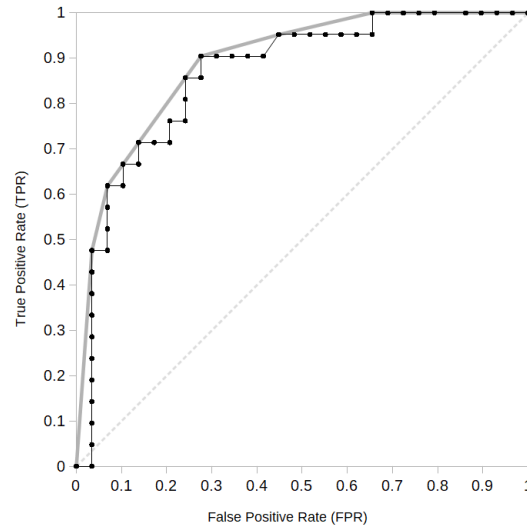


Fig. 2. Receiver Operating Characteristic with interpolated convex hull.

2.5 Receiver Operating Characteristic

By tuning our sensitivity parameter, we can adjust our tradeoff between excluding false negatives and including false positives. Plotted on a graph, we see our Receiver Operating Characteristic, or ROC. The area under these points creates our “AUC” statistic, or area-under-the-curve (the ‘curve’ being the black line in Fig. 2). 0.5 is the AUC of a totally random classifier (like the dashed line in Fig. 2), and 1 is only achievable with no mistakes. AUC

Table 2. Classifier results.

(sensitivity = 0.5) Accuracy	0.800
(sensitivity = 0.5) PPV	0.704
(sensitivity = 0.5) NPV	0.913
(sensitivity = 0.5) F_1 score (positive condition)	0.792
(sensitivity = 0.5) F_1 score (negative condition)	0.808
(sensitivity = 0.5) simulated return	35.1%
ROC AUC	0.865
ROC AUC (interpolated)	0.892

is more representative than simple accuracy because it takes into account the spectrum of configurations for the classifier's sensitivity.

We can trivially interpolate between these points by imagining a series of classifiers that randomly choose with a certain probability to defer to one of two of our classifiers, creating a 'line' of these between those two points. If we choose to apply this technique to the convex hull of our points, we get an improved curve that interpolates between our best results and represents the best the classifier is capable of (the solid gray line on Fig. 2). We calculate the area under this curve as our "interpolated AUC".

3 RESULTS AND PERFORMANCE

Table 2 gives statistics for one run of the classifier, with sensitivity set to 0.5. The classifier shows a clear bias for being better at rejecting underperforming stocks than determining overperforming ones. We simulate a return value for the test period by determining what the return would have been for an investment split equally between the classifier's chosen stocks, and it significantly outperforms a simple index fund over the same period.

Our AUC values are very encouraging, suggesting that this classifier could be the basis of a powerful tool to evaluate stock likenesses for a variety of desired sensitivities.

Table 3 gives the specific results from the network configured with a sensitivity of 0.5. The stocks are sorted by increasing return during the test period, and those that were predicted to outperform the S&P 500 index have their tickers bolded. The false positives (GE, APA, FLR, DNB, JNRP, AKAM, GWW, and FISV) and false negatives (KHC and BDX) are marked with asterisks. The index itself is italicized in approximately the middle of the list, where one would expect it.

4 DISCUSSION

4.1 Applications

Although we used 'foreknowledge' of the test period to train our model, these results do demonstrate that stocks that will outperform the market in the future could have similarities in their past performance that machine learning techniques can discuss. One application of this would be for someone with expert knowledge or insider information to label a training set of stocks. Neural network optimization would create a model that could evaluate whether other stocks are 'like' the expert's picks. Another application could be to time-shift data and see if there are recognizable patterns in daily returns that repeat over intervals, which could potentially be projected into the future.

4.2 Conclusion

Machine learning techniques can be used to distinguish effectively between large-cap stocks that will and will not outperform the market over a specific period of time. Neural networks are well-suited to this task because

Table 3. Example classification of test set (sensitivity = 0.5).

JWN	-25.3%	VIAB	9.7%	BDX*	21.7%
CVS	-21.7%	STZ	10.6%	CTXS	30.9%
AN	-10.1%	AKAM*	12.0%	JEC	31.1%
COG	-7.1%	GWV*	12.6%	UNP	31.8%
MOS	-1.8%	ADS	12.8%	AAPL	34.7%
ORLY	-1.5%	TMO	12.8%	KLAC	36.1%
O	0.1%	TDC	15.0%	DGX	43.6%
GE*	0.3%	AET	16.3%	ROK	45.1%
K	0.8%	DTE	16.6%	CMI	47.3%
APA*	1.5%	PNW	17.3%	TXN	47.8%
FLR*	2.4%	FISV*	17.3%	ULTA	48.2%
SO	3.2%	CVX	17.5%	FITB	53.5%
DNB*	6.7%	<i>S&P 500</i>	17.5%	WDC	59.9%
NWSA	7.4%	WAT	17.9%	PFG	63.5%
DPS	7.5%	WFC	19.7%	CSX	79.8%
CL	8.8%	HOLX	20.5%	RF	81.1%
JNPR*	9.0%	KHC*	20.7%	URI	94.0%

of how they are able to detect subtle correlations even among thousands of variables. Although our model was constructed using foreknowledge of stocks' performance, other, less absolute methods could be used to label training data for the same or similar purposes.

5 WORKS CITED

- *Forecasting S&P 500 Stock Index Using Statistical Learning Models*. Available from: <https://www.researchgate.net/publication/31150> [accessed Dec 04 2017].
- Lin, H. (n.d.). *Feature Investigation for Stock market Prediction*. Retrieved from <http://cs229.stanford.edu/proj2013/Lin-Feature%20Investigation%20for%20Stock%20Market%20Prediction.pdf>
- Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). Deep learning for stock prediction using numerical and textual information. *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. doi:10.1109/icis.2016.7550882