Using AI to Make Predictions and Decisions on Stock Market

Jack Jin; Alice Zheng

Project Introduction

The problem we want to tackle in the project is to create an automated tool for managing investments in the stock market with a limited amount of selected stocks. Our model comprises of two separate parts. The first part takes in the stock prices in the past and makes a prediction of the future prices, which is then fed into the second part to be used for making decisions on which stocks to buy/sell, and when to buy/sell the shares we hold.

Predictions based on Time Series Data

We started with a simplified problem: predicting whether the prices will increase or decrease in the next n days, using the stock prices and volumes in the past m days. For each stock, we used Alpha Vantage API to access the **daily open price, daily high price, daily low price, daily close price** and the **daily volume** from the beginning of 2000.

(m, n)	(20, 5)	(20, 3)	(10, 3)	(10, 1)	(5, 1)
Logistic Regression	51.26%	51.04%	52.37%	47.97%	48.08%
Bayesian Network	50.84%	50.97%	48.52%	47.09%	46.87%
Simple Neural Network	47.06%	45.93%	44.66%	42.14%	42.83%
SVM with rbf kernel	46.22%	44.79%	43.38%	41.33%	41.01%

Table 1. Test error rates for different models on Time Series Data

The high error rate suggests that the underlying process is not well represented by the models we chose, or our choice of predictors are not good.

Making decisions

Our decision part is defined as a search problem with the future stock prices known, and we implemented it as a Markov decision process with the actions only comprising of selling all the stocks we currently hold and investing all the money on one specific stock for each timestep. For each timestep, we look ahead at the next 20 timesteps and find the best policy to achieve the best result in 20 timesteps, and we use the action of the acquired policy in the first timestep as the action of our current timestep. This theoretically should provide a nearly optimal solution if we are sure that the predicted future prices are exact.

Predictions based on Technical Indicators

Technical indicators are models of processing time series data. After reading several investment education websites, we picked the following **11 most popular technical indicators** to work with.

Moving Average Convergence Divergence (**MACD**)	$\begin{aligned} shortema &= 0.15\times price + 0.85 * shortema_{[-1]} \\ longema &= 0.075\times price + 0.925 * longema_{[-1]} \\ MACD &= shortema - longema \end{aligned}$
Stochastic oscillator (**STOCH**)	$\begin{aligned} \%K &= 100\times\frac{close - LowestLow_{[last\ n\ periods]}}{HighestHigh_{[last\ n\ periods]} - LowestLow_{[last\ n\ periods]}} \\ \%D &= MovingAverage(\%K) \end{aligned}$
Relative strength index (**RSI**) values	$\begin{aligned} &\text{If } close > close_{[-1]} \text{ then} \\ &\quad up = close - close_{[-1]} \\ &\quad dn = 0 \\ &\text{else} \\ &\quad up = 0 \\ &\quad dn = close_{[-1]} - close \\ &upavg = \frac{upavg\times(n-1)+up}{n} \\ &dnavg = \frac{dnavg\times(n-1)+dn}{n} \\ &RMI = 100\times\frac{upavg}{upavg+dnavg} \end{aligned}$
Average directional movement index (**ADX**) values.	$ADX = \frac{ADX_{[-1]}\times(n-1)+DX}{n}$

APO-SMA Absolute price oscillator values with SMA	$PO = \text{SlowMA}(price) - \text{FastMA}(price)$
APO-EMA	
CCI Commodity channel index values	$\begin{aligned} &CCI = \frac{TP - ATP}{0.015\times MD} \\ &TP = \frac{high_n + low_n + close}{3} \\ &\text{TP} = \text{Typical Price} \\ &\text{high}_n = \text{Highest high in the last n time periods} \\ &\text{low}_n = \text{Lowest low in the last n time periods} \\ &ATV = \text{SimpleMovingAverage}(TV) \\ &MDTV = \text{MeanDeviation}(TV) \end{aligned}$
AROON	$\begin{aligned} AroonUp &= 100\times\left(\frac{n - PeriodsSinceHighestHigh}{n}\right) \\ AroonDown &= 100\times\left(\frac{n - PeriodsSinceLowestLow}{n}\right) \end{aligned}$
Bollinger bands (**BBANDS**) values	$\begin{aligned} TP &= \frac{high+low+close}{3} \\ MidBand &= \text{SimpleMovingAverage}(TP) \\ UpperBand &= MidBand + F\times\sigma(TP) \\ LowerBand &= MidBand - F\times\sigma(TP) \end{aligned}$
Chaikin A/D line (**AD**) values.	$\begin{aligned} CLV &= \left(\frac{(close-low)-(high-close)}{(high-low)}\right) \\ AD &= AD_{[-1]} + CLV\times volume \end{aligned}$
On balance volume (**OBV**) values.	$\begin{aligned} &\text{If } close > close_{[-1]} \text{ then} \\ &\quad OBV = OBV_{[-1]} + volume \\ &\text{elseIf } close < close_{[-1]} \text{ then} \\ &\quad OBV = OBV_{[-1]} - volume \\ &\text{else} \\ &\quad OBV = OBV_{[-1]} \end{aligned}$

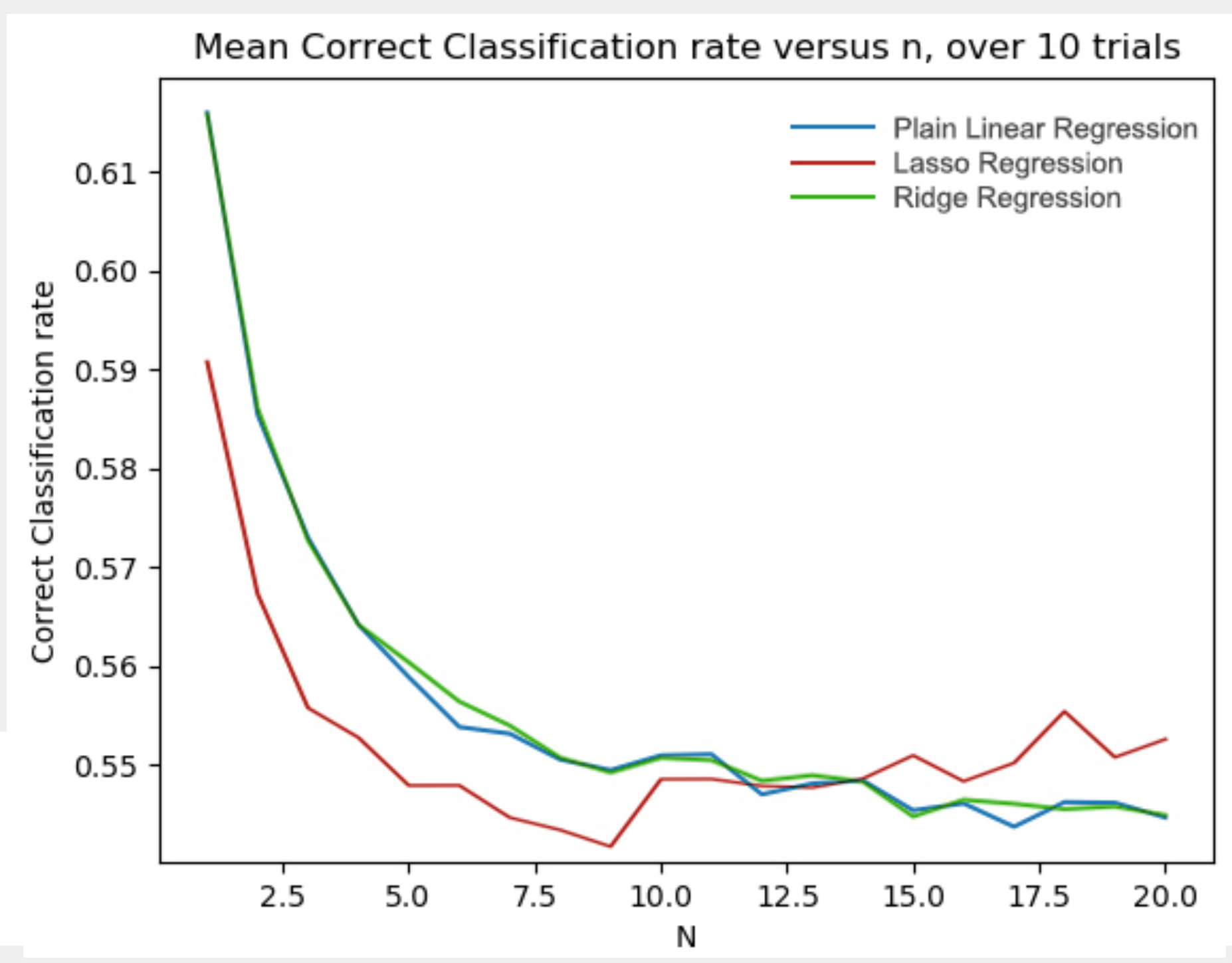
Prediction Model and Results

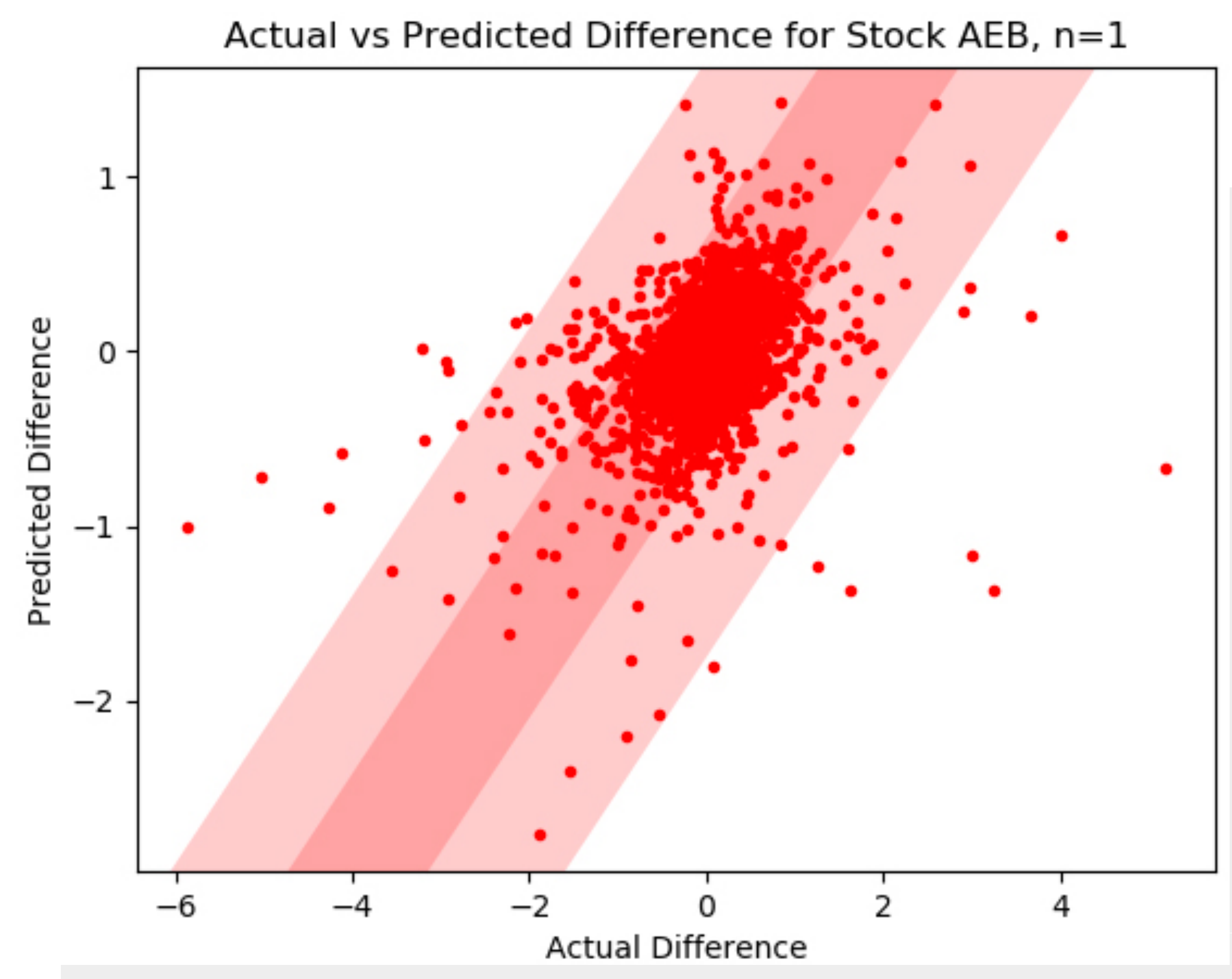
In order to run our decision machine, we need a prediction of the specific price value, so we switch to a regression problem. The response variable we target is **the difference between the prices over a n-day period**, and our predictors include **the stock price at the end of the day at the beginning of the period, the volume of the same day, and the 11 indicators that we chose in the previous section.**

We collected a total of 260k data samples from 82 stocks (time series data) we partition the data into **training set, dev set, and test set in a ratio of 7:2:1.** We used two metrics for evaluating different regression models: the first one is the **classification** of the price trend (increase/decrease), and the second metric is the **mean squared error.**

After evaluating different models on the data set, we see that support vector regression gives the best result. However, it is also extremely costly to compute, so we only train the model on a subset of all our data. For processing more data, we see linear regression is the best tool we have. By comparing the training MSE and the test MSE, we see that there is **almost no over-fitting** in the model.

To the right is a figure of the correct classification rate versus n for linear regression, plain and with 2 types of regularization (ridge and lasso). We see that as n increases, the correct rate drops, suggesting that time series data do not provide us the required information to make good predictions for over a very long time.





To the left is a figure of the actual difference versus the predicted difference for one of the stocks we reserved as part of the test set. Even though we see that our best prediction correct rate is lower than 70%, we see that we tend to have a much higher prediction correct rate for prices changes that are "bigger". For example, we can achieve a 92.3%correct rate for the trend when the price increases for more than $2, and a correct rate of 72.9% when decreases by more than $2

NOTE: When we ran support vector regression with rbf kernel on 10% of the training data, we were actually able to achieve a correct rate of 68% on test data. However, the computation is quadratic to the sample size when the sample is small, and it becomes near exponential when the cache is incapable of holding all required data, we hope to find more powerful computation resources and finish SVR before the final report.

References

[1] Vantage, Alpha., Alpha Vantage - Free Apis For Realtime And Historical Financial Data, Technical Analysis, Charting, And More!. Alphavantage.co. N.p., 2017. Web. 19 Nov. 2017.
[2] Pedregosa et al., Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830, 2011.
[3] Kyoung-jae Kim, Ingoo Han. "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index". Expert Systems with Applications, Volume 19, Issue 2, 2000, Pages 125-132, ISSN 0957-4174.
[4] Technical Indicator Reference.. Fmlabs.com. N.p., 2017. Web. 19 Nov. 2017.
[5] Investopedia. Investopedia. N.p., 2017. Web. 19 Nov. 2017.