Stanford | ENGINEERING
Computer Science

Predicting Stock Price Movement Using Social Media Analysis

Derek Tsui | { dtsui }@stanford.edu | CS 229, Autumn 2016

Stanford University

Introduction

Social media platforms provide a wealth of information on real-world patterns and behaviors. This project analyzes aggregated message data from StockTwits, a popular online investor platform, to approach the financial problem of stock price prediction.

Data and Features

We performed analysis on the component stocks of the Dow Jones Industrial Average[1]. Data was collected for the period Dec. 2013 to Dec. 2016, totaling 756 trading days. Two main datasets were used:

- Daily split-adjusted price data, collected via Yahoo finance API.
- StockTwits message data collected and downloaded in raw JSON format, totaling over 540,000 messages.

Preprocessing was necessary to generate “bag-of-words” feature vectors, including removing stop-words and company names, removing posts mentioning multiple stocks, and aggregation by date. Sentiment polarity was also extracted from user-generated “bullish”/”bearish” tags, for which a rolling mean of the ratio was calculated.

70% of the data was used for feature selection, of which a third was reserved for feature selection and cross validation. The remaining 30% was used as the test set and comprised of data from after the dates in the training set to remove any look-ahead bias.

The 3-day future return, calculated as a percentage change, was used as the prediction target to model the short-term correlation with social media activity. A shorter period was not selected to reduce the effects of market noise. Although binary classification was initially the main focus, regression models tended to outperform with respect to accuracy and performance.

Methodology

Several supervised learning methods were considered and tested on the data. Two feature sets were used for the different methods: a pure bag of words model, and a model using a combination of word frequency features and sentiment metrics.

1. Pure Bag of Words Model

After preprocessing, the frequencies of all 6,839 words occurring at least 25 times in the data using the *tf-idf* metric, a statistical metric for word importance in a document that takes the product of term frequency (TF) and inverse document frequency (IDF), and Laplace smoothing.

$$TF(t) = \frac{\text{no. of times } t \text{ occurs in a document}}{\text{total no. of terms in the document}}$$

$$IDF(t) = \log \frac{\text{total no. of documents}}{\text{no. of documents with the term } t \text{ in it}}$$

These metrics are then used as features in our multinomial model.

a. Naïve Bayes Classification

We use the multinomial model for this generative learning algorithm, which assumes feature independence and seeks to maximize:

$$p(y)\prod_{i=1}^{n} p(x_i|y)$$

2. Mixed Features Model

The number of word frequency features was reduced to 1000 first by using uni-variate chi-squared score ranking, and then finally to 506 by using recursive feature elimination with 5-fold cross validation. Other features from the social media data, including message volume change and polarity metrics were also included in this model.

buy	earnings	new	today	price	long
eps	good	short	pt	support	bearish

Figure 1: Sample of words selected by this model. NB: “pt” shorthand for “price target”

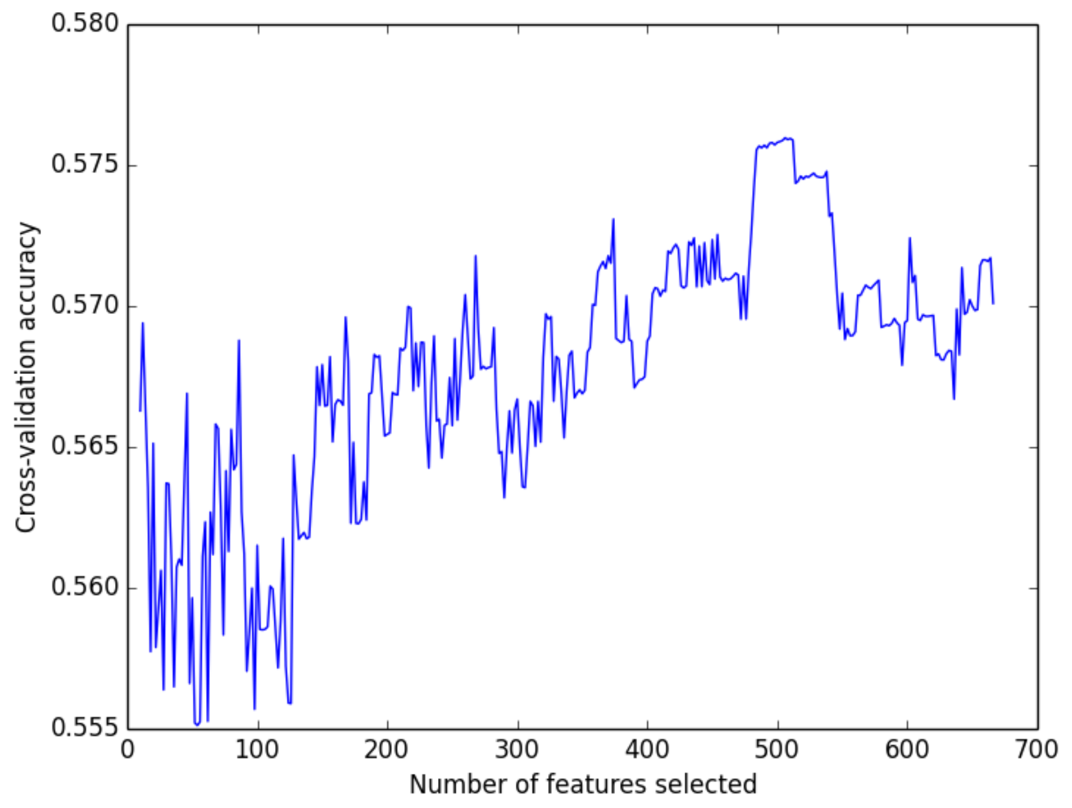


Figure 2: Cross-validation accuracy in recursive feature elimination.

b. Support Vector Regression

We use this good “off-the-shelf” learning algorithm for regression, which, similar to how SVC finds a decision boundary that maximizes the margin, aims to minimize the ε-insensitive loss function created by Vladimir Vapnik:

$$L_\varepsilon(y, f(\mathbf{x}, \omega)) = \begin{cases} 0 & \textit{if } |y - f(\mathbf{x}, \omega)| \leq \varepsilon \\ |y - f(\mathbf{x}, \omega)| - \varepsilon & \textit{otherwise} \end{cases}$$

L2-regularization was used and the SVR ε parameter was optimized with 5-fold cross validation.

c. k-NN Regression

We perform k-nearest neighbors under the same feature set as SVR, using the Euclidean distance metric. The parameter *k* was optimized with 5-fold cross validation.

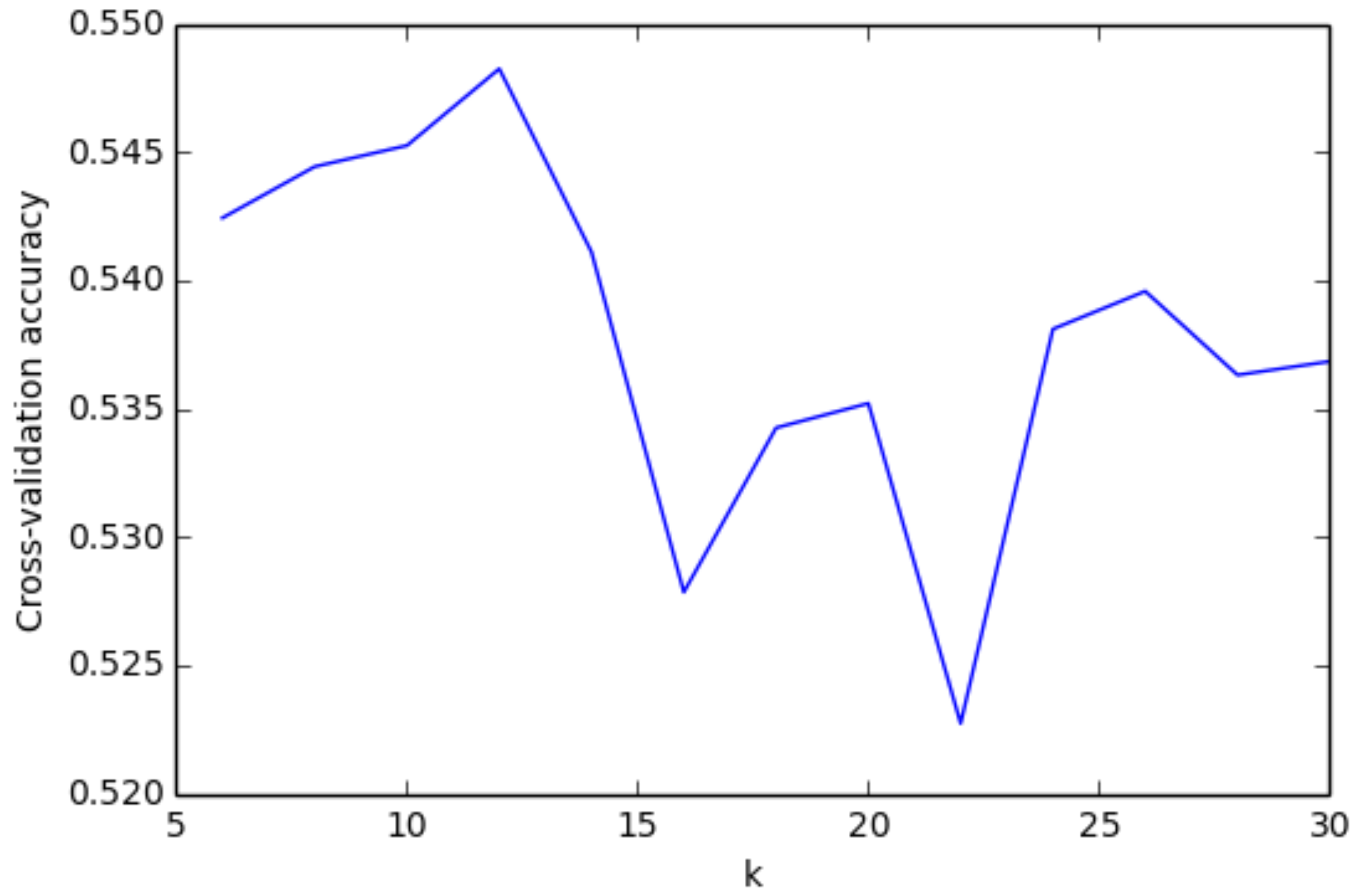


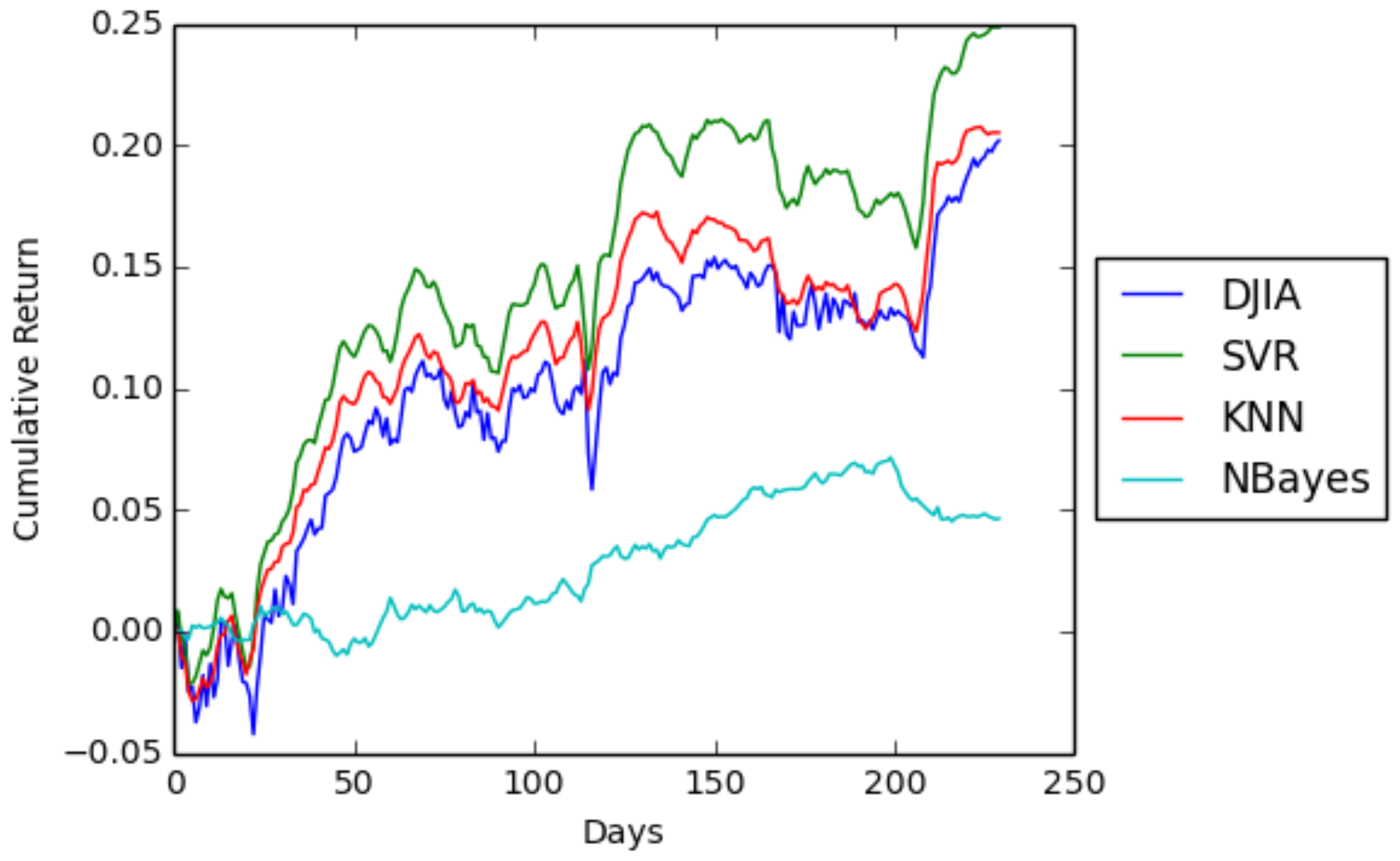
Figure 3: Cross-validation accuracy in k-NN parameter optimization.

Results

The test accuracy for each of the learning models is shown below:

	Test Accuracy
Naïve Bayes	0.5099
SVR	0.5682
k-NN Regression	0.5448

For practical trading purposes, however, the more important metric is profitability. The model predictions for the positive and negative classes were used to generate long/short signals on the test data, and a simulated portfolio[2] allocated 33% equity to daily equal-weighted long/short positions each held for 3 days. The resulting profit-and-loss (PnL) curves are graphed in comparison to the DJIA index performance for the same time period.



This graph shows the results on the test period, from Jan 2016 to Dec 2016. Although Naïve Bayes is consistently profitable, it significantly underperforms the other methods and the broader market. SVR returns a total of 25% within this 11-month period, after simulated commission costs.

Discussion

Overall, the regression models proved to be more accurate and more actionable as trading signals compared to binary classification. One plausible reason for this result is that binary classification (especially the generative model used in Naïve Bayes) will attempt to fit to the noise inherent in stock market price movement, and lumps small, statistically insignificant upward movements indiscriminately with large ones.

The test error rates were below what this project initially aimed to achieve; however, the signal’s positive performance indicates that the selected features are in fact meaningful, and capture some insight into short-term market movements. Notably, the SVR and KNN models are still very correlated to the market, although the SVR model does deviate from the markets in certain timeframes. Future work on this topic could involve testing recurrent neural networks on the data, which would be particularly suitable for time series prediction problems like this one.

[1] AAPL was omitted: the number of posts was exceedingly large.
[2] commission model: $.0075/share