Improving Stock Prediction Based on News Data

Once Upon a Time - yzhan417, yzhan420, qwang52, ftong

Goal

Stock market prediction is difficult and in this information age, prediction based on quantitative data is not enough. Our goal is to use supervised learning algorithms with news data to predict stock movement with a past dataset to analyze how the news data affects the result.

Data

We selected data from 44 companies (each with over 200 days of news data in a year) from early 2016 to late 2017. For each company and each day we collected:

- **Price data** (clean but raw): originally downloaded from Yahoo Finance. Based on the closing price, we calculated the increase in price both in number and ratio.
- **News data** (pretty dirty): headlines scraped from MarketWatch and Reuters, queried using the symbol or name of the company as the key word and were collected for articles with the same dates as the stock data. Headlines were then lowercased, filtered for stopwords, and lemmatized.
- **Sentiment Score**: we used VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analysis, a dictionary and rule-based sentiment analysis tool that specifically targets sentiments expressed in social media. In our code we used the vaderSentiment package in Python.

Methodology

Overview

We trained k-nearest neighbors (KNN) and support vector machine (SVM) models and found the SVM to be more accurate (figure 1), so using the SVM we compared the model's accuracy with three different situations (figure 2):

- inputting only the baseline (quantitative data)
- inputting both the baseline and news data
- inputting the baselines, news data, and sentiment score

Steps

1. Input preprocessing: For each company and each day, we turned the news data into 100 dimensional vectors with doc2vec. By combining the stock data, news data and sentiment score, the input for each company each day would be a corresponding (6+100+1=) 107-dimensional vector.

2. Training/testing: We split the dataset into a train set (80%) and test set (20%). With each 107-dimensional vector in the training dataset, we trained our model. In the case where the company had insufficient news, the vector would be filled with the news from the previous day.

3. Output: If the model predicted that the price would be flat in the next day, output would be 0; if fall, output would be -1; and if raise, output would be 1. Then we tested on the test dataset.

4. Evaluation: This is a kind of classification problem, so we chose the classic four evaluation indexes to measure results: accuracy, recall, F1 score, and precision of the test set.

Results

Claim 1: Our SVM outperforms our KNN classifier
Support: Both models are trained with all features, including baseline (price data), news data, and sentiment score.

	KNN	SVM
Accuracy of test set	0.6473	0.7305
F1 of test set	0.3482	0.5792
recall of test set	0.3801	0.6243
precision of test set	0.3829	0.6290

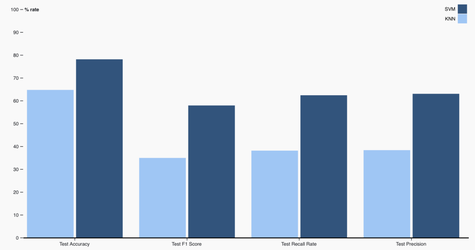

Figure 1: Test scores for our KNN and SVM models for accuracy comparison.

Claim 2: The classifier trained using all features outperforms baseline models by a significant margin.
Support: After figuring out that SVM performs better in this case, we trained it with different kinds of input.

	baseline	baseline + news data	baseline + news data + sentiment score
Accuracy of test set	0.5147	0.7194	0.7305
F1 of test set	0.2448	0.5386	0.5792
recall of test set	0.3337	0.5636	0.6243
precision of test set	0.3348	0.5982	0.6290

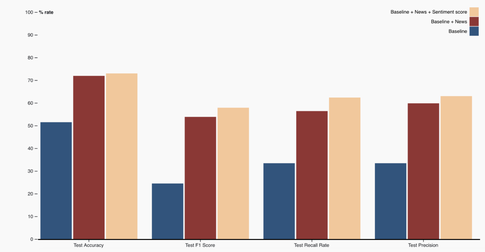

Figure 2: Test scores for different input combinations using the SVM model.

Claim 3: The model performs badly with sudden jumps or falls.
Support: Prediction accuracy are only 49.5328% for the days with more than ±5% increasing ratio of baseline, and can reach up to 79.0000% for those with minor changes (changing percentage less than ±5%).

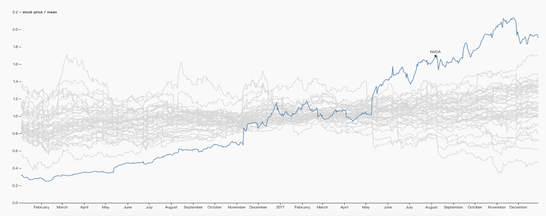

Figure 3: Plot using our raw data showing stock price (using close price / mean close price) change from Jan 04, 2016 to Dec 29, 2017.

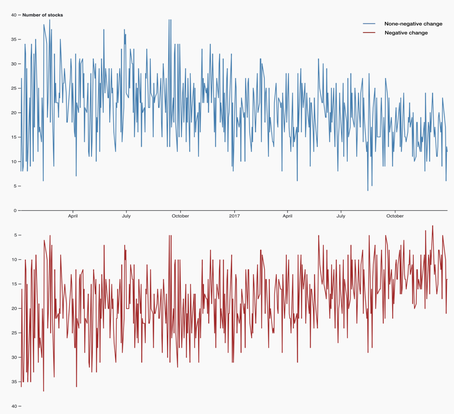

Figure 4: Plot using our raw data showing the number of stocks for each day with non-negative change (blue) and negative change(red) in close price

Figure 5: Typical keywords for stock price going up using news data for days and stock with over 5% rise in closing price.

True value vs model prediction: Using data points from 50 days from the test set, the following figures show the true value and corresponding prediction made by our model. The horizontal axis counts the days and arranges them in a virtual "two-month" period. If the close price rises or drops on a certain day, then the total price is increased or reduced by one respectively (i.e. the plots demonstrate the aggregated result of all the rises (+1) and falls (-1)). We can see that although the true and predict results differ at a few points, the overall rise-and-fall trend of them is similar.

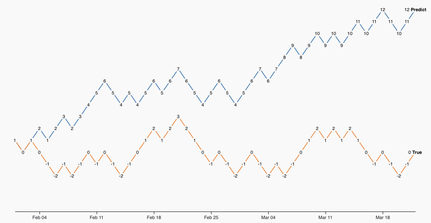

Figure 6: true value vs. model predict for MasterCard(MA)

Figure 7: true value vs. model predict for Nvidia (NVDA)

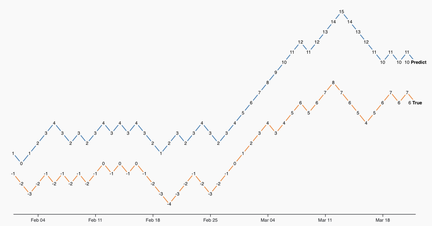

Figure 8: true value vs. model predict for Intel (INTC)

Conclusion

Because we only used historical data to predict stock movement, our model currently can't handle sudden jumps or falls. If there is not enough news data to input, accuracy of the final prediction will also be affected. Since there may be potential relationships among some companies, considering this in future iterations of our model may improve its performance.

Regardless, because we factored in news data and sentiment scores as features, our SVM model was able to both accurately predict stock movement and outperform baseline models by a significant margin. Since our model is really profitable and effective, individuals, institutions and even the government can hire people to build and improve this model so as to make money, monitor or even manipulate the stock market.