Predicting Stock Price Changes Using Past Prices and News Articles

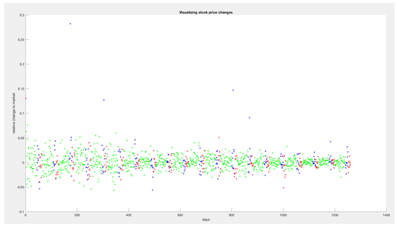
Ang Li (al171@stanford.edu)

Objective

The goal of the project is to predict price changes in the future for a given stock. Information that might be leveraged to make these predictions more accurate include prices from previous days, and financial news articles related to the company of interest.

To further reduce volatility in the results, predictions are only made on days that are far from a company's Earnings Report dates (excluding 10 days before and after each Earnings Report).

The model outputs classification (1 or 0) of the sign of the next day's price change. The model achieved 59% accuracy and an annualized return of ~15%.



Data

Nasdaq → Preprocess

xignite The Market Data Cloud → Preprocess

Symbol	Change Relative to NDXT (11/14)	Change Relative to NDXT (11/15)	More Dates ...
FB	2.3%	-0.5%	...
NDXT	0.0%	0.0%	...

Date	News Headline
11/14	"Facebook Stock Could Reach $200 But Amazon Like Growth Unlikely"
11/15	"Class Action Suit Against Facebook Accuses Company Of Hiding Concerns About Growth Forecasts"
More Dates ...	More Headlines ...

Market data in the last 5 years were obtained from nasdaq.com and preprocessed to extract non-earnings days and relative price changes to the technology industry using NDXT as the technology sector index.

News data were pulled from xignite.com and preprocessed to remove duplications and have exact date of news assigned to each headline.

Naive Bayes for News Articles

- Naive Bayes using the multinomial event model with Laplace smoothing was trained on 400 days, and tested on 200 days of data. Output of the model is the probability of a positive relative price change.
- Numbers, percentages, money amounts were substituted with fixed tokens that consolidate these words together.

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = 1\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} n_i + |V|}$$

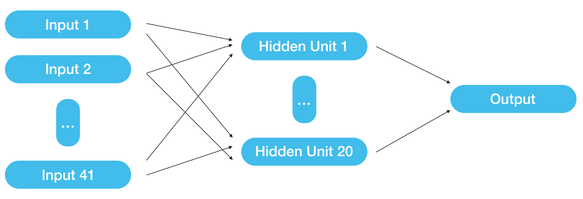
$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = 0\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} n_i + |V|}.$$

$$\phi_y = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}.$$

Features for NN Model

- 41 features in total to predict today's relative price change
  - Relative price changes for the same stock from last 20 trading days.
  - News predictors (between 0 and 1) for the last 20 trading days.
  - News predictor (between 0 and 1) for today.

Neural Network



$$z^{[1]} = W^{[1]}x^{(i)} + b^{[1]}$$
$$a^{[1]} = g(z^{[1]})$$
$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$
$$a^{[2]} = g(z^{[2]})$$
$$W^{[\ell]} = W^{[\ell]} - \alpha\frac{\partial\mathcal{L}}{\partial W^{[\ell]}}$$
$$b^{[\ell]} = b^{[\ell]} - \alpha\frac{\partial\mathcal{L}}{\partial b^{[\ell]}}$$

- The neural network was structured to have 1 hidden layer with 20 nodes. The input layer has 41 nodes, and the output layer has 1 node.
- The activation functions used in the hidden and output layer are sigmoids.
- Mini-batch gradient descend (with batch size 20) with forward and backward propagation was applied to learn parameters in the neural network.
- L2 Regularization with lambda = 0.0001 was also applied to reduce overfitting to training data.

Results

- One of the most important evaluation metric in this project is percentage gain if trading according to the predictions. E.g., a value of 0.01% daily gain means the expected gain over 100 days is 1%.

Model	Training Error (probability of wrong prediction)	Test Error (probability of wrong prediction)	Test Daily % Gain
Naive Bayes (News only)	0.0248 (403 samples)	0.445 (100 samples)	N/A
Linear Regression	0.4601 (589 samples)	0.4365 (126 samples)	0.0311
Logistic Regression	0.4584 (589 samples)	0.4762 (126 samples)	0.0018
Neural Network	0.0084 (403 samples)	0.4100 (100 samples)	0.0423

- Naive Bayes news article positiveness predictor achieved an accuracy of 55.5% on the test set.
- Linear regression with standard least squares error and the previous price changes features (no news) resulted in 56.35% accurate predictions.
- Logistic regression with the previous price changes features (no news) resulted in 52.38% accurate predictions.
- Neural network as structured in previous section with all features (including news) produced the best accuracy (59%) and daily gain (0.0423%).

Discussion & Future Work

- Overall, there was a lot of overfitting to training data despite the regularization techniques used, due to the fact that the stock market is way to complex to be captured in merely price changes and news.
- Higher quality news data would have helped with prediction accuracy. The news data pulled from xignite.com appear to have mediocre quality.
- Naive Bayes predictions could be improved to increase accuracy, which would in turn increase with the prediction accuracy of subsequent models. Ways to improve Naive Bayes predictions include removing insignificant words, reducing stemmed words into base forms, and so on.
- Neural networks with other activation functions and different architecture (e.g. using ReLU or a different number of hidden units) could be explored.
- Augment the neural network by adding more input features, such as trends in other industries, political influences, competitor trends, and many more, to capture the full scale of complexities of a stock market.