Mining Twitter for Financial Event Prediction

Ayush Agnihotri - 2010EE10147

Ankit Rao - 2010EE10653

Can you predict financial markets?

- It's quite fascinating to even think of doing so. There have been efforts in the past with varying success.
- Things like chart patterns, time series analysis, machine learning n stuff are being used for the same.
- Can we add another dimension to it ? Say the prevailing public sentiments. Would that help ?

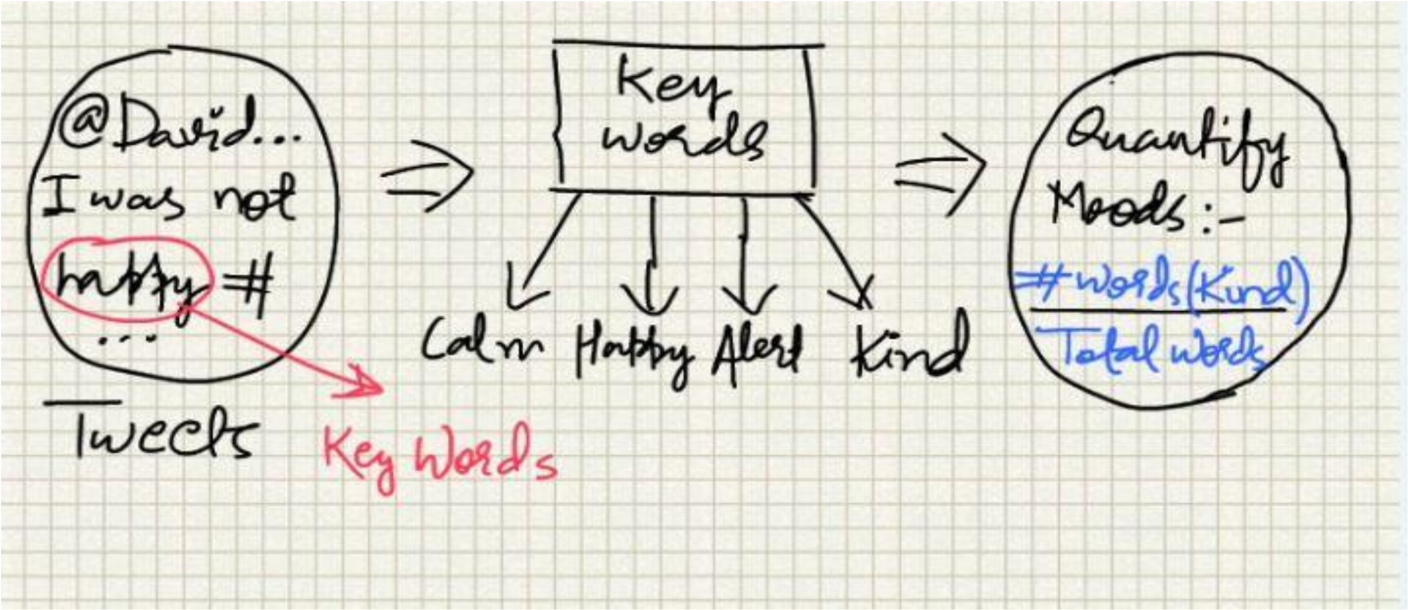
Using Twitter

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com

- Twitter is an exciting platform wherein public sentiments can be captured.
- By tracing key words in tweets, one can gauge the general public sentiments.
- These sentiments in turn govern how people behave in real life, say while executing orders on a trading floor.

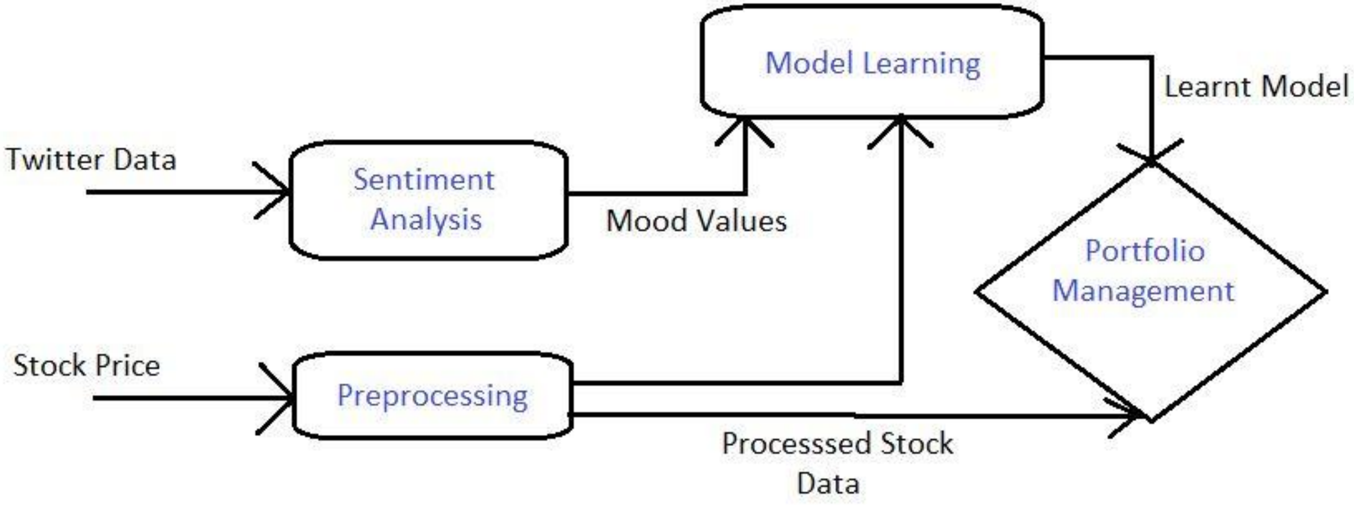
How to quantify sentiments ?

- People express feelings through tweets, hence by looking for key words we can quantify various moods relatively :



Correlating Mood Scores With Stock Series Trends

- Now we try to learn in a supervised setting how does mood scores affect price movements, say correlating last month's market with corresponding tweets.



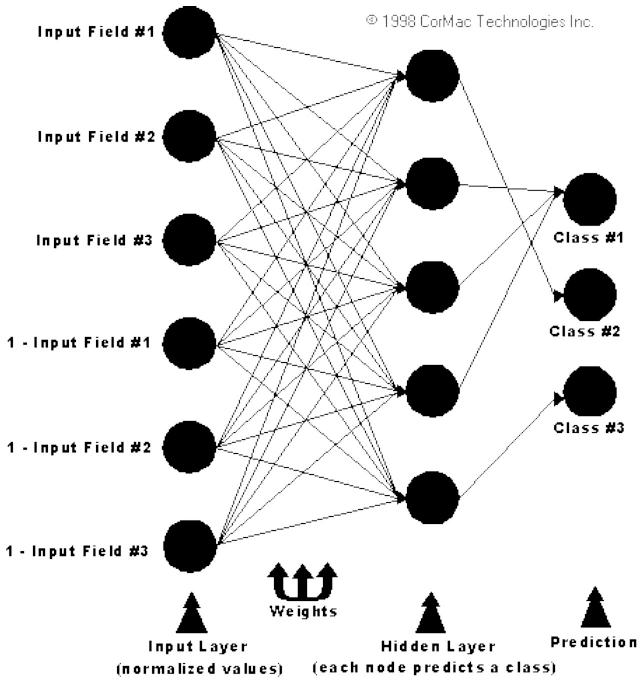
Data ?

- How did we mine the twitter data?

  We made use of the developer tools provided by twitter and along with it we used the twitteR package in R to extract the feeds from twitter, we were able to extract daily tweets for a particular hashtag.
- To get the corresponding stock prices we used the data provided by Yahoo! Finance.

The SFANn

- Stands for Simplified Fuzzy ARTWORK Neural Network.(Ref Kasuba.T, 1993)
- The inputs need to be normalized.
- Single hidden layer of fuzzy neurons.
- Number of neuron in the hidden layer variable



The methodology

- We feed the following parameters (and their compliments, by the definition of SFANn):-

1. Normalized EoD stock prices for 3 previous days
2. Normalized sentiment vector values
3. Labels, +1 for the stock going up that day and -1 otherwise.

- And now we train the SFANn with the given data for a number of epochs and update the weights and the no. of neuron accordingly

- For each neuron, the weight update and the output activation and weight update are as follows.

$$T_j(I) = \frac{|\mathbf{I} \wedge \mathbf{W}_j|}{\alpha + |\mathbf{W}_j|}$$

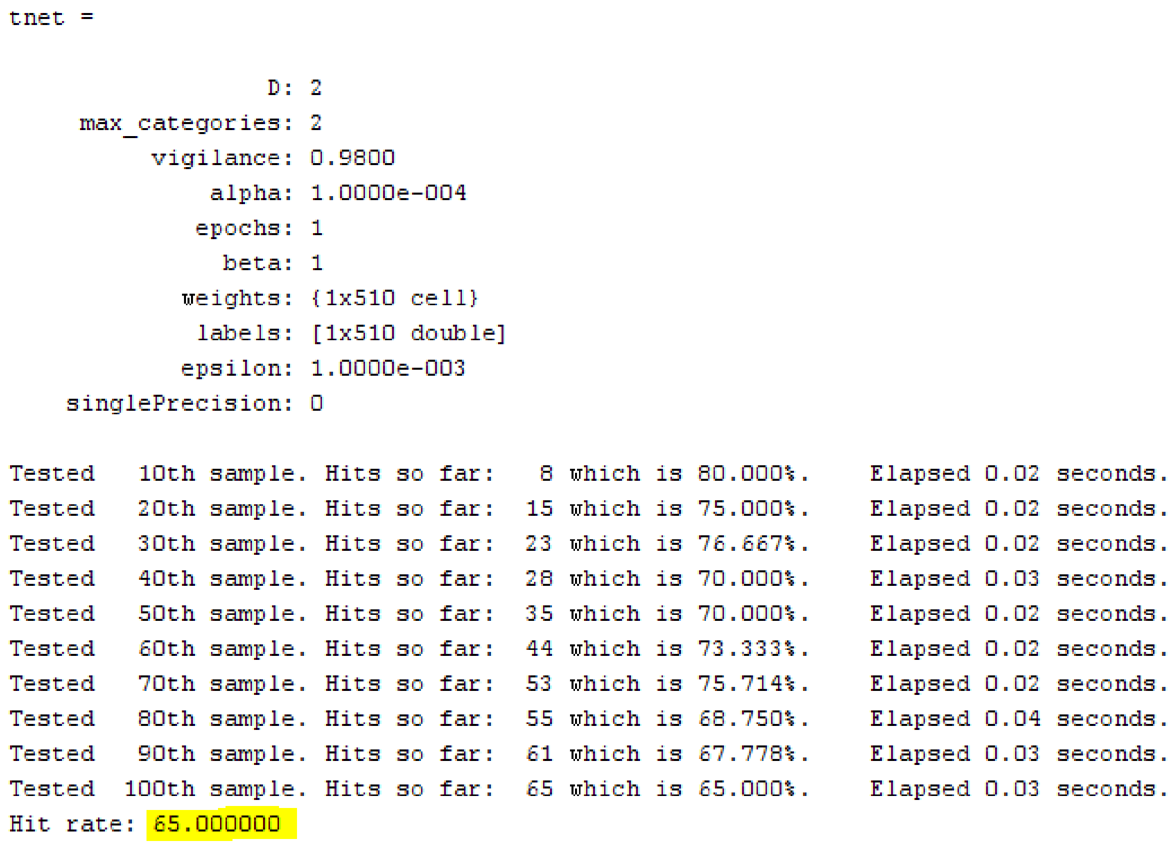
Activation

$$\mathbf{W}_j^{\text{new}} = \beta(I \wedge \mathbf{W}_j^{\text{old}}) + (1 - \beta)\mathbf{W}_j^{\text{old}}$$

Weight update equation

- Once the network is trained on the training data, we apply it on the test data and check the results against actual movements.
- We worked on the stock data for Boeing, and the network gave us an accuracy of 65.0% on a test data of size 100

Results



```
tnet =

                 D: 2
    max_categories: 2
         vigilance: 0.9800
             alpha: 1.0000e-004
            epochs: 1
              beta: 1
           weights: {1x510 cell}
            labels: [1x510 double]
           epsilon: 1.0000e-003
   singlePrecision: 0

Tested   10th sample. Hits so far:   8 which is 80.000%.    Elapsed 0.02 seconds.
Tested   20th sample. Hits so far:  15 which is 75.000%.    Elapsed 0.02 seconds.
Tested   30th sample. Hits so far:  23 which is 76.667%.    Elapsed 0.02 seconds.
Tested   40th sample. Hits so far:  28 which is 70.000%.    Elapsed 0.03 seconds.
Tested   50th sample. Hits so far:  35 which is 70.000%.    Elapsed 0.02 seconds.
Tested   60th sample. Hits so far:  44 which is 73.333%.    Elapsed 0.02 seconds.
Tested   70th sample. Hits so far:  53 which is 75.714%.    Elapsed 0.02 seconds.
Tested   80th sample. Hits so far:  55 which is 68.750%.    Elapsed 0.04 seconds.
Tested   90th sample. Hits so far:  61 which is 67.778%.    Elapsed 0.03 seconds.
Tested  100th sample. Hits so far:  65 which is 65.000%.    Elapsed 0.03 seconds.
Hit rate: 65.000000
```

•Though we do accept that our results are very modest, but we hope to fine tune the parameters to do a better job.