STOCK PREDICTION USING TWITTER SENTIMENTAL ANALYSIS

AHMED ZUBAIR	RIZWAN SHAFIQUE	HAMZA ANJUM
16L-4259	16L-4299	16L-4179
CS-E	CS-E	CS-D

1. Introduction:

Stock market prediction has been an active area of research for a long time. The Efficient Market Hypothesis (EMH)states that stock market prices are largely driven by new information and follow a random walk pattern. Though this hypothesis is widely accepted by the research community as a central paradigm governing the markets in general, several people have attempted to extract patterns in the way stock markets behave and respond to external stimuli. In this paper, we test a hypothesis based on the premise of behavioral economics, that the emotions and moods of individuals affect their decision-making process, thus, leading to a direct correlation between "public sentiment" and "market sentiment". We perform sentiment analysis on publicly available Twitter data to find the public mood and divide it into three categories - Positive, Negative and Neutral. We use these moods and previous days' closing price of stock to predict future stock movements and then use the predicted values in our portfolio management strategy.

Related Work:

Our work takes inspiration from Bollen et al's work which received widespread media coverage recently. They also attempted to predict the behavior of the stock market by measuring the mood of people on Twitter. The authors considered the tweet data of all twitter users in 2008and used the Opinion Finder and Google Profile of Mood States (GPOMS) algorithm to classify public sentiment into 6

categories, namely, Calm, Alert, Sure, Vital, Kind and Happy. They cross validated the resulting mood time series by comparing its ability to detect the public's response to the presidential elections and Thanksgiving Day in 2008. They also used causality analysis to investigate the hypothesis that public mood states, as measured by the Opinion Finder and GPOMS mood time series, are predictive of changes in DJIA closing values. The authors used Self Organizing Fuzzy Neural Networks to predict DJIA values using previous values. Their results show a remarkable accuracy of nearly 87% in predicting the up and down changes in the closing values of Dow Jones Industrial Index (DJIA). It is regarded as the state of art work in predicting stock prices using tweet data.

Another inspiration comes from Anshul Mittal and Arpit Goel's work which itself is based on Bollen's work. They followed similar path and achieved acurracy of 75.56%.

Both of these studies predict the DJIA value for the next day but we predict whether the stock price will g up or down the next day.

2. Dataset:

The data set we used is StockNet dataset courtesy of Yumo Xu which can be obtained from Github. The dataset includes two-year price movements from 01/01/2014 to 01/01/2016 of 88 stocks are selected to target, coming from all the 8 stocks in the Conglomerates sector and the top 10 stocks in capital size in each of the other 8 sectors. The full list of 88 stocks and their companies selected from 9 sectors is available in StockTable. It also contains tweet data of all these companies for two-year period separated by each day.

Preprocessing:

The stock data is present for working days and do not contain weekend price. We predicted this price by calculating average of previous and next present day. For n days we used the formula $(x + y)/2$ where x is previous present day and y is next present day for n days. We have also stored only closing price. Tweet data present is raw data generated from twitter. We have gathered tweet text, favorite count and retweet count. We have used TextBlob library to calculate sentiment of tweet. Favorite count and Retweet count are added together as weight. There are three types of sentiment - Positive, Negative, Neutral. We calculate the maximum sentiment value and assign sentiment to that day. We only used data of Apple for

prediction. It contains sentiment value and closing price of stock for each day from 01/01/2014 to 31/03/2016.

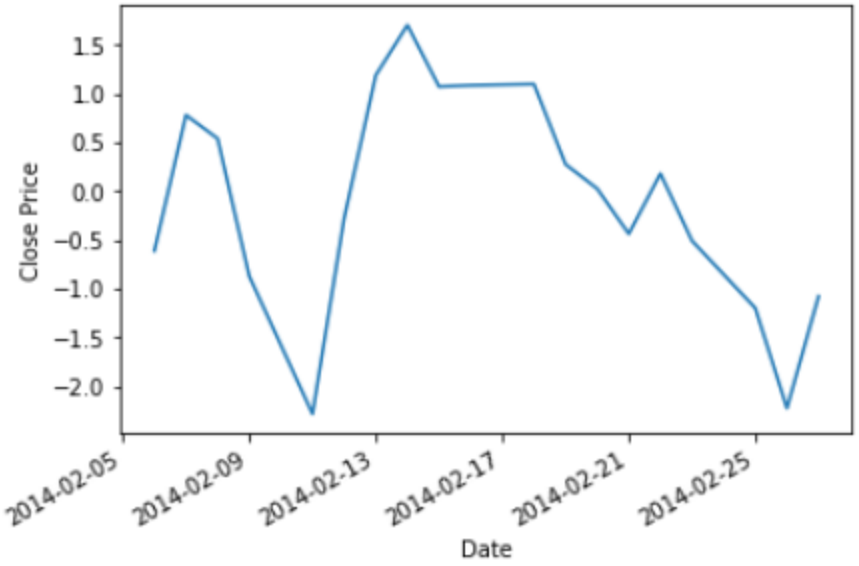

Fig 1. Plot of closing stock price of Apple for one month time period.

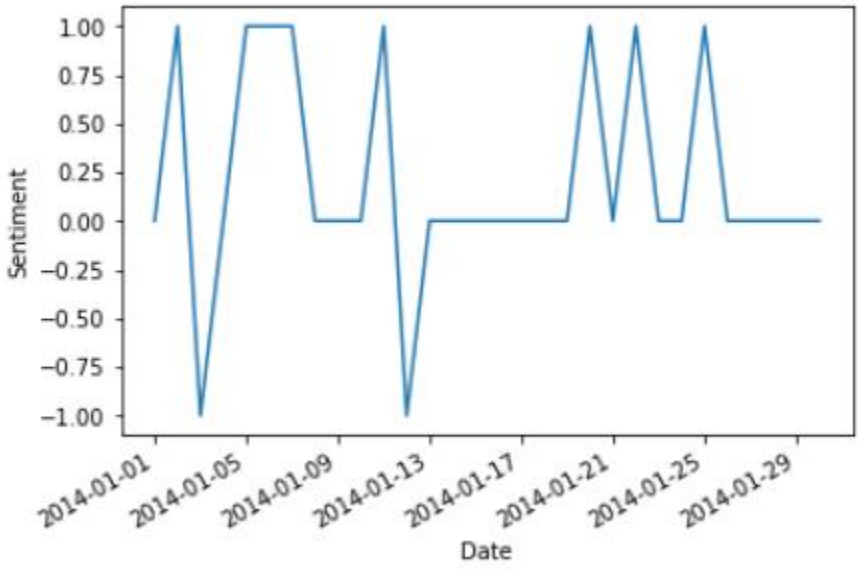

Fig 2. Plot of sentimental values for each day of a month for Apple

3. Methodology:

Data is prepared by using a 30 day window and sliding it over the complete time period. For sentiment, neutral is assigned value of 0, positive 1 and negative -1. At

start of each window a sentiment score is set to zero and value of each day is added to it. Score corresponding to each is added day is set as sentiment value of that day. The closing price of that day is also stored. Normal values of price and sentiment score is calculated and used. It is one instance of training data. The label for this instance is closing price of 31st day. Random 5% data is separated from training data as test data.

Data Preparation:

In case of time series, our data is just 1D (the plot we usually see on the graph) and the role of channels play different values, close prices and tweet sentiment. You can also think about it from other point of view, on any time stamp our time series is represented not with a single value, but with a vector (price, sentiment). We don't need to predict some exact value, so expected value and variance of the future isn't very interesting for us, we just need to predict the movement up or down. That's why we will risk and normalize our 30-days windows only by their mean and variance (z-score normalization) $((x-\mu)/\sigma)$, supposing that just during single time window they don't change much and not touching information from the future. But we are going to normalize every dimension of time window independently. But as we want to forecast movement of a price up or down next day, we need to consider the change of a single dimension. So, the data we will train on, are time windows of 30 days, but on every day we will consider whole price and sentiment data correctly normalized to predict the direction of close price movement.

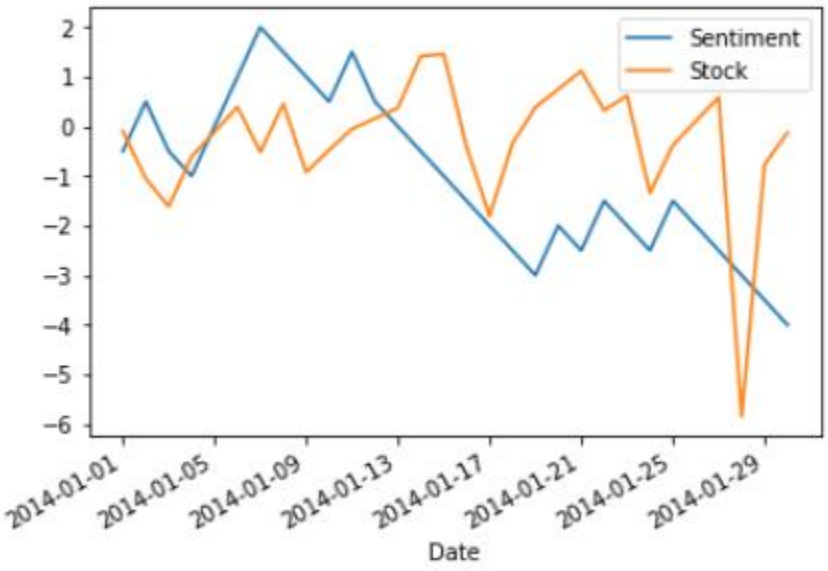

Fig 3. Stock price and Twitter sentiment time series for one-month period of Apple. Subtle correlation can be observed.

Model:

For the purpose of prediction we have used CNN model. Convolutional neural networks are deep artificial neural networks that are used primarily to classify images (e.g. name what they see), cluster them by similarity (photo search), and perform object recognition within scenes. They are algorithms that can identify faces, individuals, street signs, tumors, platypuses and many other aspects of visual data. CNN has been successful in various text classification tasks. In KimY's report, Convolutional Neural Networks for Sentence Classification, the author showed that a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks – improving upon the state of the art on 4 out of 7 tasks. A 1D CNN is very effective when you expect to derive interesting features from shorter (fixed-length) segments of the overall data set and where the location of the feature within the segment is not of high relevance. This applies well to the analysis of time sequences of sensor data (such as gyroscope or accelerometer data). It also applies to the analysis of any kind of signal data over a fixed-length period (such as audio signals). Another application is NLP (although here LSTM networks are more promising since the proximity of words might not always be a good indicator for a trainable pattern). Mainly we choose CNN because of flexibility and interpretability of hyperparameters (convolutional kernel etc.) and performance similar to RNNs, better than MLP with much faster training. The architecture used has two 1d convolution layers with activation set to Leaky Relu and dropout set to 0.5. It is followed y a fuly connected layer with 64 neurons and activation set to Leaky Relu. Last layer is a Softmax layer. We have used Nadam as the optimizer with learning rate set to 0.002. Batch size of 128 is used for 50 epochs.

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 30, 16)	144
batch_normalization_1 (Batch	(None, 30, 16)	64
activation_1 (Activation)	(None, 30, 16)	0
dropout_1 (Dropout)	(None, 30, 16)	0
conv1d_2 (Conv1D)	(None, 30, 8)	520
batch_normalization_2 (Batch	(None, 30, 8)	32
activation_2 (Activation)	(None, 30, 8)	0
dropout_2 (Dropout)	(None, 30, 8)	0
flatten_1 (Flatten)	(None, 240)	0
dense_1 (Dense)	(None, 64)	15424
batch_normalization_3 (Batch	(None, 64)	256
activation_3 (Activation)	(None, 64)	0
dense_2 (Dense)	(None, 2)	130
activation_4 (Activation)	(None, 2)	0

Total params: 16,570
Trainable params: 16,394
Non-trainable params: 176

Fig 4. Model Architecture

4. Results:

Maximum accuracy computed over the period of 50 epochs is **70%**. The model was trained for 50 epochs and the weights with least loss were used for making the final predictions. The price movement is a 0-1 classification problem and hence categorical cross entropy, which reduces to binary cross entropy in this case is used a measure of loss. The results of minimizing the loss over the course of 50 epochs are as shown in Figure 5. Initially the training loss is very high and test

loss is moderate. After around 8 epoch test loss achieves a minimum value and the model is well fitted at this stage.

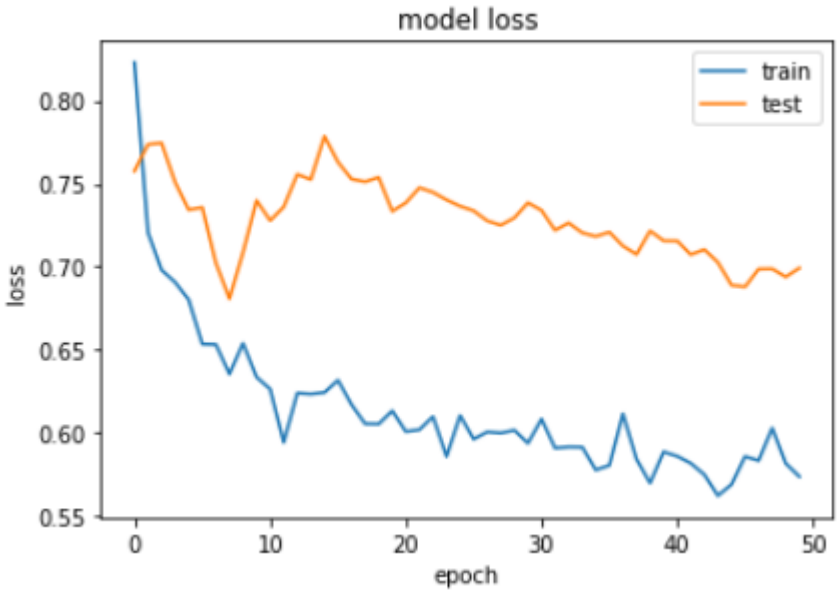

Fig 5. Train and Test Loss over 50 epochs

Train and test accuracy are shown in figure 6. It shows that test accuracy fluctuates over time but peaks at around 8th epoch and maximum accuracy of 0.699999988079071 or roughly 70% is achieved at that stage.

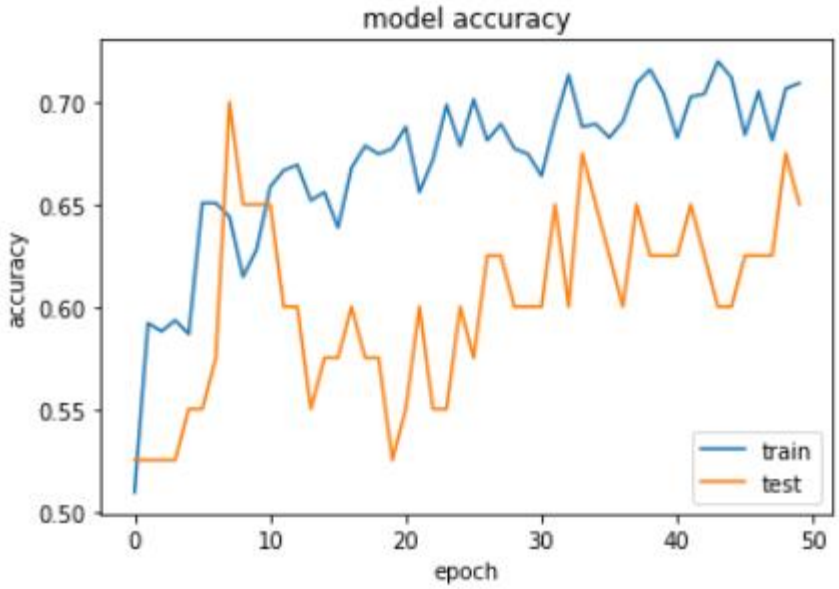

Fig 6. Train and Test Accuracy over 50 epochs

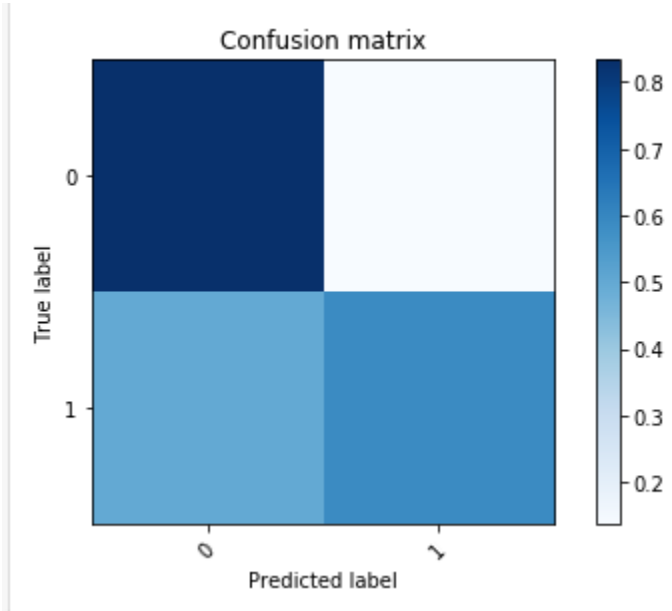

Fig 7. Confusion Matrix

5. Conclusions And Future Work:

Multivariate Time Series Deep Learning holds great potential for stock price predictions and has been shown to achieve better results than techniques relying solely on price data by supplementing it with Twitter Sentiment data. An accuracy of 70% for predicting price change direction has been obtained beating comparable models that relies on just price data. Our results are in conjucture with past experimental works that Twitter data can be used to predict movement of stock prices.

Finally, its worth mentioning that our analysis doesn't take into account many factors. Firstly, our dataset doesn't really map the real public sentiment, it only considers the twitter using, English speaking people. It's possible to obtain a higher correlation if the actual mood is studied. It maybe hypothesized that people's mood indeed affect their investment decisions, hence the correlation. But in that case, there's no direct correlation between the people who invest in stocks and who use twitter more frequently, though there certainly is an indirect correlation - investment decisions of people may be affected by the moods of people around them, ie. the general public sentiment. All these remain as areas of future research.

It is worth noting that the model uses Twitter information for stock market as a whole instead of targeted recognized sources to make trading decisions for individual stocks which restricts its performance and at the same time makes the implementation simplistic. Also, sentiment can be divided into more categories of mood to better predict correlation between public sentiment and stock price. News data can also incorporated with tweet data making prediction more accurate. Also, the high frequency price data and live news feed can be used to make a real time High Frequency Trading (HFT) system using the architecture of proposed model.

The test accuracy can be improved by using more sophisticated models such as LSTM in the future.

To sum it up, this work provides an exposure into the application of Multivariate Time Series learning for financial and tweet time series data. The positive results obtained validate the credibility of model for more complex tasks.

6. References:

1. https://github.com/yumoxu/stocknet-dataset
2. http://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf
3. https://medium.com/@alexrachnog/neural-networks-for-algorithmic-trading-2-1-multivariate-time-series-ab016ce70f57