Stock Price Prediction

Runyu Qian
University of California Los Angeles
Los Angeles, CA
rqian@g.ucla.edu

Hao Wang
University of California Los Angeles
Los Angeles, CA
haowanglud@g.ucla.edu

Xuening Wang
University of California Los Angeles
Los Angeles, CA
sherry9788@g.ucla.edu

Shuoyi Wei
University of California Los Angeles
Los Angeles, CA
shuoyi@g.ucla.edu

Pei Zhou
University of California Los Angeles
Los Angeles, CA
zpcs03@g.ucla.edu

ABSTRACT

This paper investigates the ability of Twitter posts (tweets) in predicting stock price. With crawled tweets on a specific company as the data set, we employ word2vec and Global Vectors to convert words in tweets to vectors. We use manually labels tweets to train sentiment classifiers, including Logistic Regression, Random Forest Algorithm and CNN, and use these models to generate all the labels for each of the tweets. We categorizes tweets with labels: Positive, Negative and Neutral. Using the results of classification, we analyze the relationship between mood of tweets and stock market movements.

ACM Reference Format:
Runyu Qian, Hao Wang, Xuening Wang, Shuoyi Wei, and Pei Zhou. 2018. Stock Price Prediction. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

According to Efficient Market Hypothesis (EMH) developed by Professor Eugene Fama, it is impossible to "beat the market" as the stock price follows a random walk pattern and has at most 50% accuracy in predictions [2]. For years researchers have been looking for efficient stock price predictors that are theoretically supported by behavioral economics, and have already concluded that public sentiment is a valuable indicator of stock market movements.

Twitter is a social media platform where millions of users share their real-time stories/thoughts/opinions with "tweets" that are limited to 140 characters. Twitter has been growing popularity over the latest 10 year since its launch in 2006, and has attracted researchers who wish to learn from behavioral economics. Some early works used tweets as indicators of public sentiment and have found strong correlation between the proportion of emotional tweets significantly correlated with the overall stock market movements (Dow Jones, NASDAQ and S&P 500, etc.) [11]. [1] classified sentiments to 6 different levels to deeply investigate the mechanism of Behavioral Economics. More about our problem definition and formalization can be found in Section 2.

In this paper, our goal is to construct a significant correlation between public sentiment of a specific industry/company (Apple Inc. in this paper) reflected in tweets and the real-time fluctuation of this industry/company's stock price. We will filter data from Twitter by key words and categorize each tweet to one of the three levels: Positive, Negative and Neutral. We will also apply additional information beside tweets, such as number of retweets, to adjust the weights accordingly in our classifiers to conclude and further predict the public sentiment on a specific target that we believe will influence the stock price substantially. In our research, we choose "iPhone X/8" as the key word. The introduction of iPhone X/8 has been the headlines recently, and has triggered a wide discussion among people. We believe that the discussion is an interesting reflection and effective indicator of public sentiment and will potentially influence the stock price of Apple.

The rest of our paper is organized as follows. Section 2 specifies our problem and provides formalization. Section 3 describes the details in data preparation and processing. Section 4 shows the details of the methods we use to do the experiments, including word2vec that we use to convert text data to numeric vector data. Besides, Logistic Regression, Random Forest Algorithm and CNN are analyzers we plan to use in categorization of tweets, also talked about in Section 4. Experiments design and Evaluations on different methods can be found in Section 5. Section 6 is about related works for this paper. And finally Section 7 presents our conclusion.

2 PROBLEM DEFINITION AND FORMALIZATION

Our input is Twitter data and output is the prediction of rise or fall of the stock price of Apple. Since we consider public moods to be the major fact influencing the stock price, now the problem becomes mining sentiments from tweets (sentiment analysis of tweets).

Furthermore, to do the sentiment analysis on Twitter corpus, we consider it as a supervised classification task in numerical data. So we manually label all the crawled tweets with 3 being positive, 2 being neutral and 1 being negative. After training with the manually labeled tweets, our classifier models (specified in later sections) can predict the label given the tweet.

Then, we have to consider the problem of converting the text data from Twitter to numerical representations for classification

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

task. We use word2vec and GloVe (specifically for CNN classifier) to solve this problem. These neural language models help convert each word to a vector. By combining each word embedding in a tweet, we can get a vector representation of a tweet. Also, we need to preprocess the tweets for word2vec training since they contain lots of noises. Note that the above formalization is in a reverse format of how we conduct our experiments.

In summary, the eventual problem we are going to solve is to predict the movement of the stock of a company by the mood of public on its production and recent market strategies. And we separate that problem into several sub-problems:

- Preprocess and label Twitter corpus
- Convert text data from Twitter to numerical vectors with labels of sentiment
- Build classifiers to classify sentiments of tweets
- Predict stock price rise or fall from sentiments by methods like majority voting.

3 DATA PREPARATION DESCRIPTION AND PREPROCESSING

3.1 Tweets

We stream a collection of public tweets related to target company/industry. Tweets data are crawled through the Twitter REST API, accessed through Python package tweepy. We only take into account tweets wrote in English in order to perform sentiment analysis. The metadata of each tweet, most notably includes the following features, are collected:

- Text content of the Tweet (at most 140 characters)
- Date-time of the submission (GMT+0)
- Number of favorites
- Number of followers

Text are parsed into words in order to be used as training set. Date-time are rounded into dates to match daily quotes of stock price.

3.2 Stock Price

Historical quotes for stock price for Apple Inc.(AAPL), including Open, High, Low, Close and Volume are downloaded from TradeStation. All historical data are daily.

3.3 Label Tweets

Different tweets contain different mood. We use numerical number 1,2,3 to represent three different type of mood: including negative,neutral,and positive. Firstly, we manually label 1,2,or 3 to 900 tweets, and then using these 900 tweets as training set to train the model, including Random Forest Tree, Logistic Regression and Convolutional Neural Network. The final step is to use these three models to label all the tweets. There are still some space need to prove in this labeling part. But to the limitation of time of number of teammates, we don't label enough tweets, and moreover there are lots of tweets are labeled neutral, which is 2. Later, we can increase the percentage of posituve and negative tweets to make the model more sensitive to moods.

3.4 Preprocessing Tweets

Corpus from Twitter tends to be informal and unclean, since lots of noises like emojis, slangs, and informal usages of languages exist, and they can potentially affect the results of experiments. To address this issue, we use several NLP tools to preprocess the data crawled from Twitter.

First, we keep only words that are in English, i.e. words that are composed of a-z and A-Z characters. Then we converted all characters to lowercase English letters. For the next step, we started with removing the stop words that are in the NLTK package. We later found, however, that some stop words can actually carry important semantic meanings for our sentiment analysis and price predictions. Words like "don't", "against" contain negative semantic meanings towards the topic in the tweet, and they will contribute to negative sentiments as well as fall in stock prices. So we keep the stop words in the tweets in that step (the removing of stop words step is still in the preprocessing.py file for future usage purposes).

Then we proceed the preprocessing work by using a python preprocessor for Twitter. The preprocessor currently supports cleaning, tokenizing and parsing of following items:

- URLs
- Hashtags
- Mentions
- Reserved words (RT, FAV)
- Emojis
- Smileys

After using the preprocessor, our preprocessing tweets part is complete. In summary, the text preprocessing part keeps only English words and convert them to lowercase, removes some specific noises for Twitter data like hashtags, emojis, etc. Our preprocessed data is a lot cleaner as a result (see next page for examples).

4 METHODS DESCRIPTION

In this section, we describe major steps and methods used to predict stock prices from Twitter data.

4.1 Neural Language Model

To convert input of text data with labels to numeric values for later analysis, we used the open-source tool word2vec [7] to learn word embeddings in specified dimension space. The training objective of word2vec is to learn word vector representations that are good at predicting the nearby words. In the model, $v(w) \in R^d$ is the vector representation of the word $w \in W$, where W is the vocabulary and d is the embedding dimensionality. Given a pair of words (w^t, c), the probability that the word c is observed in the context of word w^t is given by,

$$P(D = 1|v(w_t), v(c)) = \frac{1}{1 + e^{-v(w^t)^T v(c)}} \tag{1}$$

The probability of not observing word c in the context of w^t is given by,

$$P(D = 0|v(w_t), v(c)) = 1 - P(D = 1|v(w_t), v(c)) \tag{2}$$

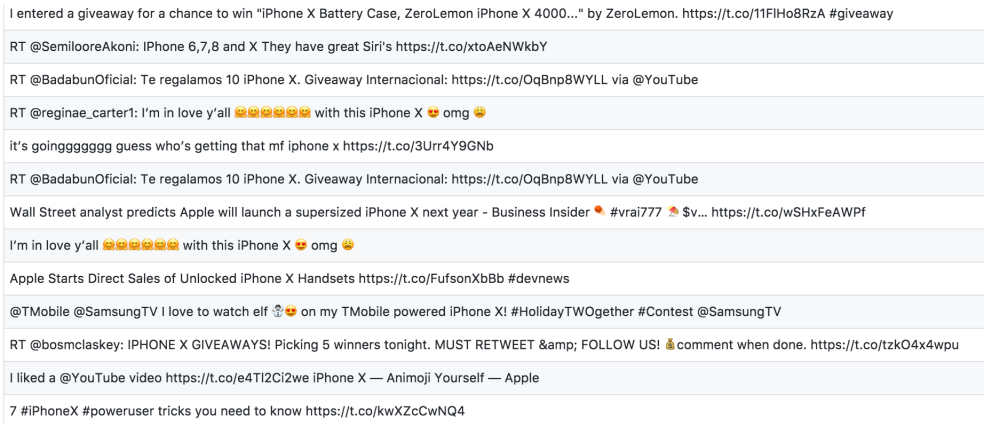

Figure 1: Not cleaned twitter data

```
i entered a giveaway for a chance to win iphone x battery case zerolemon iphone x by zerolemon
rt iphone and x they have great siri s
te regalamos iphone x giveaway internacional via
i m in love y all with this iphone x omg
it s goingggggggg guess who s getting that mf iphone x
te regalamos iphone x giveaway internacional via
wall street analyst predicts apple will launch a supersized iphone x next year business insider v
i m in love y all with this iphone x omg
apple starts direct sales of unlocked iphone x handsets
i love to watch elf on my tmobile powered iphone x
iphone x giveaways picking winners tonight must retweet amp follow us comment when done
i liked a video iphone x animoji yourself apple
tricks you need to know
```

Figure 2: Preprocessed twitter data

Given a training set containing the sequence of word types $w_1, w_2, ..., w_T$, the word embeddings are learned by maximizing the following objective function:

$$J(\theta) = \sum_{(w_t, c_t) \in D^+} \sum_{c \in c_t} \log P(D = 1 | v(w_t), v(c)) + \sum_{(w_t, c'_t) \in D^-} \sum_{c' \in c'_t} \log P\left(D = 0 | v(w_t), v\left(c'\right)\right)$$

where w_t is the t^{th} word in the training set, c_t is the set of observed context words of word w_t and c'_t is the set of randomly sampled, noisy context words of the word w_t. D^+ consists of the set of all observed word-context pairs (w_t, c_t) $(t = 1, 2, ..., T)$. D^- consists of pairs (w_t, c'_t) $(t = 1, 2, ..., T)$ where c'_t is the set of randomly sampled, noisy context words for the word w_t.

Once we trained this neural language model on our dataset, we can obtain a unique vector representation of each word that appears in the tweets collected. And after we combine all representations of words in a tweet simply by summing up all vector representations of words in the tweet, we obtain an embedding for the tweet.

Note that for Convolutioanl Neural Network classifier that will be covered later, we used a similar kind of embedding: GloVe [8] developed by Stanford University. This is because that for CNN, the mechanism of this model considers the frequency of each element in a context by doing convolution on a window shifting on the text input by stride size. GloVe, compared to Word2vec, improves the embedding of word2vec because it considers the global frequency of a word in a corpus, which corresponds to the mechanism of CNN.

4.2 Sentiment Classifier

After we feed the tweets to the neural language model specified above and get vector representations of tweets, we use these embeddings to train several classifier models for sentiment detection task since we have already labeled the tweets. The following describes some classifiers.

4.2.1 Logistic Regression. Logistic regression is a model for classifying categorical variables, in our case, labels of sentiments. It is first developed by statistician David Cox in 1958 [10].

Since we defined 3 classes of sentiments in this research, our first step is to binarize our labels. For example, a tweet that is labeled as 1 has its label binarized to an array $[1, 0, 0]$. In this way, we transform our multi-label classification to binary, which fits the nature of logistic regression. However, unlike the general multilabel classification where each data point belongs to multiple classes, we assign one data to only one of the labels as the labels are naturally mutually exclusive. In the rest of the training and testing, we use One-vs-the-rest strategy to train over each class.

One-vs-the-rest (OvR) is a well-known multinomial classification strategy. With this strategy, we fit one classifier per class. The most significant advantage of this approach is its interpretability. Since each class is represented by one and one classifier only, it is possible

to gain knowledge about the class by inspecting its corresponding classifier. This is the most commonly used strategy for multiclass classification and is a fair default choice . This is achieved with **OneVsRestClassifier** in sklearn.multiclass. This strategy is also applied in Random Forest model that will be discussed next.

Logistic regression utilizes Maximum Likelihood Estimation and logistic function to classify each data point into a class based on probabilities.

$$L = \sum_i y_i \mathbf{x}_i^T \beta - log(1 + exp\mathbf{x}_i^T \beta)$$

4.2.2 Random Forest Algorithm. Random Forest Algorithm is The first algorithm for random decision forests was created by Tin Kam Ho [3]. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. We chose Random Forest over Decision Tree because it corrects for decision trees' habit of overfitting to their training set.

With limited number of input data (as we have to manually label crawled tweets), overfitting becomes the potential problem that will affect our training model badly. Overfitting is caused by training the parameters of a prediction function and testing it on the same data. To avoid it, it is common practice when performing a (supervised) machine learning experiment to hold out part of the available data as a test set X_test, y_test.

Here our solution to overfitting is called cross-validation. Cross-Validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model. In typical cross-validation, the training and validation sets must cross-over in successive rounds such that each data point has a chance of being validated against.

The basic approach, called k-fold CV, has the following procedures:

(1) Splits the training set into k smaller segments or folds.
(2) for each k = 1, 2, . . . K, fit the model with parameter λ to the other K-1 parts.

Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k-1 folds are used for learning. [9]

In both models, we utilized the **train_test_split** method provided in sklearn.model_selection for cross-validation.

4.2.3 Convolutionary Neural Network. We used CNN [5], a Neural Network model that utilizes layers with convolving filters that are applied to local features [6]. CNN has shown great performance on task such as Computer Vision and Natural Language Processing, and we use it in the paper specifically for sentence classification task [4].

In this project, we used the tweets crawled from twitter from November 29th to December 10th, UTC (detailed descriptions see section 3), and vectorized them with GloVe embedding [8]. GloVe stands for "Global Vectors for Word Representation". It is an unsupervised learning algorithm for obtaining vector representations of words, and is a further improvement made on Word2Vec embedding model by researchers at Stanford University, measuring not only the symmentic distance between words, but also the word-word co-occurrence statistics from the corpus. Since training the vector representation of each word is not the focus of this paper, and it would require a huge amount of data to obtain a comprehensive and accurate word vector embedding, we utilized the pretrained GloVe model on a 2014 dump of English Wikipedia, with 400k words and 100 dimensions for each word.

We treated each tweet as a single data tuple we needed to feed in to the model, and first filtered out all uncommon words within the global data corpus by setting a threshold number of top frequent words. The threshold was chosen to be 1600 for our training data set by empirical experiments. By doing this, we filtered each tweet and kept only the top frequent words in it, which helped reduce infrequent words' influence (without doing this, words appeared only once or twice in the whole corpus might affect the prediction similar to the way null values affect lift and χ^2), and also the negative affect of misspelled words by twitter users.

After filtering out these infrequent words, each tweet was left with only frequent words and we represented the whole tweet with a vector of the indexes of the words left in sequential order within this tweet. To make sure each tweet is of same dimension, we padded the vector with 0 at the end to the length of 140 words. Tuples with length longer than 140 words were also truncated to 140. The reason we use 140 as the threshold length was that tweets were supposed to have a maximum length of 140 words. Until now, we had a representation of a tweet that was ready to be filled in to the CNN network. An example is given in the following: suppose we have a tweet "My iPhone ks on fire", with 'my', 'iPhone', 'on', 'fire' to be the top 1600 frequent words. The misspelled word 'ks' (which should be 'is') will be filtered out. Suppose that the indexes for 'my' is 4, 'iPhone' is 7, 'on' is 1090, 'fire' is 81, the representation for this tweet will be [4, 7, 1090, 81, 0, 0, ..., 0], with 136 0 paddings at the end.

With the vector encoding of each tweet ready, we fed them into a convolution network with two convolution layers and two max-pooling layers, and the index for each word in each tweet will be converted to its embedding in GloVe through another map from index to word embedding. The convolution network then adjust its weights with the loss function as categorical cross entrophy, which is common for categorical classification problem. The convolution layer is set with 128 filters and 5 as the kernel size, and rectified linear unit as the activation function, which was a common setting for text prediction. We trained the model with 10 epoches randomly shuffled at each run and a batch size of 16 limited by our training set size as we only have three classes to predict and also manually labeling large amount of tweets possess challenges to current condition. The evaluation was done with 10 percent of the whole data set.

4.3 Stock Price Prediction

4.3.1 Assumptions. We make the following assumptions when developing our algorithm for forecasting the stock price of Apple Inc.

(1) The short term stock price of a given company is primarily driven by the sales of its new release product.

(2) The sales of a new release product is crucially affected by its reception and reputation.
(3) Consumers tend to tweet their honest review and attitude of a new product before the market can react to similar information communicated through news and financial reports.

As a consequence, mining the consumer sentiment of a new release product through twitter can predict the movement of stock price of that company.

4.3.2 Algorithm. After finishing the classification tasks on sentiments from tweets, we combine the sentiments for each day and calculate the aggregated sentiment using equal weight majority voting. We then use the aggregated sentiment to predict the directional movement of the stock price on that day. Then a directional forecast is issued base on that sentiment. If the overall sentiment toward the new iPhone product is positive, then the price of Apple Inc. will rise, and vise versa.

In this case each tweet is assumed to contribute equally to the overall sentiment. Since retweet are crawled as separate tweets, and favorites are rarely used by twitter users, weighting on the basis of retweets or favorites are not optimal.

5 EXPERIMENTS DESIGN AND EVALUATION

In this section, we first show our experiment designs, and then we study the effectiveness of vector represenation produced by the word2vec algorithm and the performance of the three classifier algorithms, namely Logistic Regression, Random Forest, and CNN. For evalutation word2vec, We will be following the evaluation method of Tomas Mikolov [7] with semantic and syntactic questions designed specifically for this model. For the classifier algorithms, we will be using real data sets of Twitter from 11.29 TO 12.10 where there is a significant change in the stock price of Apple.Inc to test on the precision and recall of each classifier.

5.1 Experiment Design

We continuously crawled data for 12 days (from 11-29-2017 to 12-10-2017). We expected to obtain 100 tweets including the key word "iPhone X/8" every day (as of the Twitter API limit). However, due to the encoding issue, for some of the days the crawler stops after getting less than 100 tweets. We manually labeled 878 tweets from the first 9 days as the training set (from which we apply cross-validation) to train our three classifiers. The manually added labels are our ground truth and conduct a supervised learning. We evaluate our models by comparing the predicted labels produced from the three classifiers and the manually added labels.

The inevitable problem we face is the accuracy of these manually added labels. We only have three levels to rank the sentiment of a tweet. In the process of labeling, we realize that most of the tweets can not be arbitrarily assigned to simply "positive" or "negative". For example, there are a large number of "ads" that claims "retweet to win an iPhone X". There can be a positive signal in this tweet as iPhone X has to be good to be chased as a gift. However, it is not directly related to the product or the company, and we decided to label it as "neutral". Also, different people would read different message in the tweets, so the labeling process may not be absolutely accurate or consistent even.

The limited number of data points is also an obstacle. As we know, the classifier models would perform better when there is a large training data set. This may result in overfitting.

5.2 Word2vec Evaluation

To evaluate the effectiveness of word2vec model, word2vec be using a comprehensive test set that contains five types of semantic questions and nine types of syntactic questions. An example of each question type is provided in the table below.

Type of Relation	Word Pair 1	
Common Capital	Beijing	China
All Capital Cities	Astana	Kazakhstan
Currency	RMB	China
City-in-state	Chicago	Illinois
Man-woman	brother	sister
Adjective to Adverb	happy	happily
Opposite	happy	unhappy
Comparative	good	better
Superlative	good	best
Present Participle	walk	walking
Nationality Adjective	Chinese	China
Past Tense	eat	ate
Plural Form	cat	cats
Plural verbs	eat	eats
Common Capital	Tokyo	Japan
All Capital Cities	Harare	Zimbabwe
Currency	Yen	Japan
City-in-state	Sacramento	California
Man-woman	queen	king
Adjective to Adverb	sad	sadly
Opposite	breakable	unbreakable
Comparative	bad	worse
Superlative	bad	worst
Present Participle	sleep	sleeping
Nationality Adjective	Japanese	Japan
Past Tense	sleep	slept
Plural Form	index	indexes
Plural verbs	die	dies

First, a list of word pairs of the above types are created manually, and then a larger list of questions with only the first part of the word pair will be used to test the model. Only when the model answer's the second part of the example, we say the model makes a correct prediction. In other words, only when the closest word of the given question in the latent space produced by word2vec is exactly the same as the second part of the word pair, we consider this as a correct prediction. No sysnonums are thus considered to be correct. One thing to note is that we only consider single token words (thus words like the United States, Los Angeles are not allowed).

The accuary of word2vec is evaluated as the overall accuracy of the whole model as well as the accuracy of each type of questions.

Table 1: Logistic Regression Evaluation

	precision	recall	f1-score	support
negative	0.48	0.32	0.38	47
neutral	0.77	0.76	0.77	178
positive	0.52	0.48	0.50	65
avg/total	0.67	0.63	0.64	290

Table 2: Random Forest Evaluation

	precision	recall	f1-score	support
negative	0.76	0.35	0.48	54
neutral	0.78	0.76	0.77	165
positive	0.56	0.28	0.37	71
avg/total	0.72	0.57	0.62	290

5.3 Classifer Evaluation

To evaluate the three classifier methods, we will be using both Receiver Operating Characteristics curves and Precision-Recall method.

For the ROC curves, we will plot the truth positive rate and false positive rate and compare the area below the curve for each of the three classifiers. Truth positive rate and false positive rate's formula is given as following:

$$TruthPositiveRate = TruthPositive/Positive \quad (3)$$

$$FalsePositiveRate = FalsePositive/Negative \quad (4)$$

For the Precision-Recall method, we will be calculating the precision recall value for each of the three classifiers, where precision and recall is calculated as following:

$$Precision = TruthPositive/(TruthPositive + FalsePositive) \quad (5)$$

$$Recall = TruthPositive/(TruthPositive + FalseNegative) \quad (6)$$

Logistic Regression and Random Forest both utilized cross-validation, and the two tables illustrate the evaluation on one testing data. From these tables, we can see that Random Forest performs better on negative labels. "Neutral" has the highest accuracy as it appears most frequent in the labels.

Reading from the ROC curves of Logistic Regression model and Random Forest model (see appendix Figure 3 and Figure 4), we see that the area under the curve is larger for Random Forest model. This indicates that Random Forest model performs better overall.

*The statistics in the tables above may change slightly after each trial as cross-validation randomly split the dataset into training and testing.

5.4 Prediction Evaluation

We used the trading days between 11/28/2017 and 12/07/2017 to back-test our directional forecasts. The aggregated sentiment obtained by the three classifiers on these dates are placed alongside the stock price to directly test their accuracy. In figure 5, 6 and 7, the blue bar represents aggregate sentiment, and the red dotted line represents the close price for AAPL. Positive value aggregate sentiment represent an positive overall sentiment over the new iPhone model, the negative value represent a negative overall sentiment.

Table 3: CNN Evaluation

	loss	accuracy	val loss	val accuracy
epoch1	0.9986	0.5588	0.8549	0.7011
epoch2	0.9419	0.5601	0.8194	0.7011
epoch3	0.8977	0.5803	0.8490	0.7011
epoch4	0.8551	0.6056	0.8197	0.6782
epoch5	0.8212	0.6226	0.8490	0.7126
epoch6	0.8035	0.6359	0.8544	0.6897
epoch7	0.7816	0.6448	0.8347	0.7126
epoch8	0.7777	0.6485	0.8544	0.7011
epoch9	0.76	0.6536	0.8929	0.7126
epoch10	0.78	0.6485	1.0624	0.7011

The magnitude of the sentiment represents the intensity of that sentiment, which can be interpreted as the likelihood of the stock price moving accordingly

As we can see directly from the graphs, our the sentiments show strong predictive power. The accuracy for directional forecast are listed below. All of the models significantly outperform the random walk model, and significantly differs from 50% accuracy.

Algorithm	**Back-testing accuracy**
Logistic Regression	71%
Random Forest	86%
Convolutionary Neural Network	71%

Here the Random Forest model shows the highest accuracy. The three models give the following prediction on the stock price on Dec. 11th, 2017, so we predict a fall in AAPL price on that day.

Algorithm	**Prediction of AAPL on 12/11/2017**
Logistic Regression	FALL
Random Forest	FALL
Convolutionary Neural Network	UNCHANGED

6 RELATED WORK

6.1 Convolutional Neural Network for Sentence Classification

The Convolution Neural Network methods have achieve remarkable results in computer vision and natural language processing. However,the method of using the Convolutional Neural Network to train the word vectors for sentence-level classification tasks has existed for years, and this method has achieved excellent result on multiple benchmarks. Build on the top of Word2vec, the neural network achieve a series of good experiment results.

6.2 Random Forests in the Structured Language Model

Random Forests, which were originally developed as classifiers, are a combination of decision tree classifiers. Each tree is grown based on random training data sampled independently and with the same distribution for all trees in the forest, and a random selection of possible questions at each node of the decision tree. This paper extends the original idea of RFs to deal with the data sparseness problem encountered in language modeling. RFs have been studied in the context of n-gram language modeling and have been shown

to generalize well to unseen data. This paper shows that RFs using syntactic information can also achieve better performance in both perplexity (PPL) and word error rate (WER) in a large vocabulary speech recognition system, compared to a baseline that uses Kneser-Ney smoothing.

6.3 Facebook's daily sentiment and International stock market

Faceboook, which is the word's largest social network site, has a substantial amount users likely to invest. This work measures Gross National Happiness Index and calculates its relation to the stock price. The Happiness Index also shows the mood as we did, and it has the similar ideas as our labels. However, this experiment did not use the machine learning algorithm to generate labels.

6.4 Comparative Analysis of Hedge Funds in Financial Markets using Machine Learning Models

The term hedge fund in financial markets is defined as an investment fund that aggregates the capital from independent investors, firms and institutions and thereby further invests in variety of assets using risk profile analysis and portfolio selection techniques. The comparative analysis of hedge funds is important from investment point of view.This paper layout machine learning models for comparative analysis of hedge funds in financial markets for investments. The machine learning and deep learning neural network models have been effective in exploiting the exogenous and complex data interactions in financial domain datasets and producing useful insights. The author in this paper discusses models such as semi-supervised learning, decision tree learning and hybrid time series classification along with experimental results to juxtapose the hedge funds and hence producing useful results for investments. The analysis shows that the discussed models in the paper can be used for comparative analysis of funds and the hybrid time series classification model is more effective rather than using the semi-supervised and decision tree models individually.

7 CONCLUSIONS

In this paper, we did find a strong correlation between the public sentiment from Twitter data with the movement of the Apple stock price.

From our pipeline of work, we first label the sentiments: positive, neutral and negative of the crawled tweets, preprocess them to reduce noises, use word2vec to get word embeddings for words in each tweet, and add each vector representation of a word in a tweet to get tweet embedding.

Then we design three classifiers (note that for CNN we used GloVe) to classify the labels of the tweet based on the vector representation of each tweet. From the predicted sentiments, we use statistic methods to find the correlation between sentiments and the rise or fall of stock price of Apple Inc.

This paper develops a complete pipeline from tweets to predictions of a specific stock, and evaluates several classifier methods.

Potential future work could extend the model to find specific price changes of stock prices.

REFERENCES

[1] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

[2] Eugene F Fama. The behavior of stock-market prices. *The journal of Business*, 38 (1):34–105, 1965.

[3] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[4] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[9] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer, 2009.

[10] Strother H. Walker and David B. Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1/2):167–179, 1967. ISSN 00063444. URL http://www.jstor.org/stable/2333860.

[11] Xue Zhang, Hauke Fuehres, and Peter A Gloor. Predicting stock market indicators through twitter âĂIJi hope it is not as bad as i fearâĂİ. *Procedia-Social and Behavioral Sciences*, 26:55–62, 2011.

8 APPENDIX

Table 4: Task Distribution

Task	People
1. Collecting Twitter Data and Labeling Data	Shuoyi Wei, Xuening Wang, and Runyu Qian
2. Preprocess Data and Get Word Embeddings using Word2vec	Pei Zhou
3. Implement Logistic Regression and Random Forest Algorithms	Xuening Wang
4. Implement Convolutional Neural Network Algorithm	Hao Wang
5. From Sentiment to Stock Price Prediction	Shuoyi Wei
6. Write Report	Pei Zhou, Hao Wang, Xuening Wang, Shuoyi Wei, and Runyu Qian

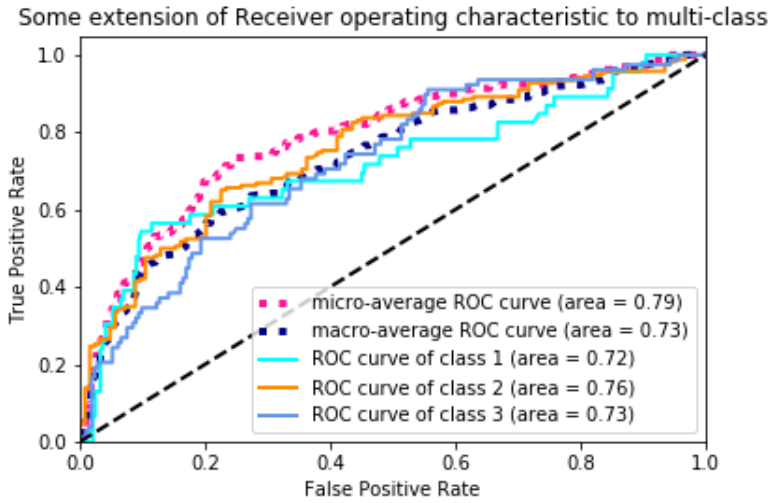

Figure 3: ROC Curve for Logistic Regression

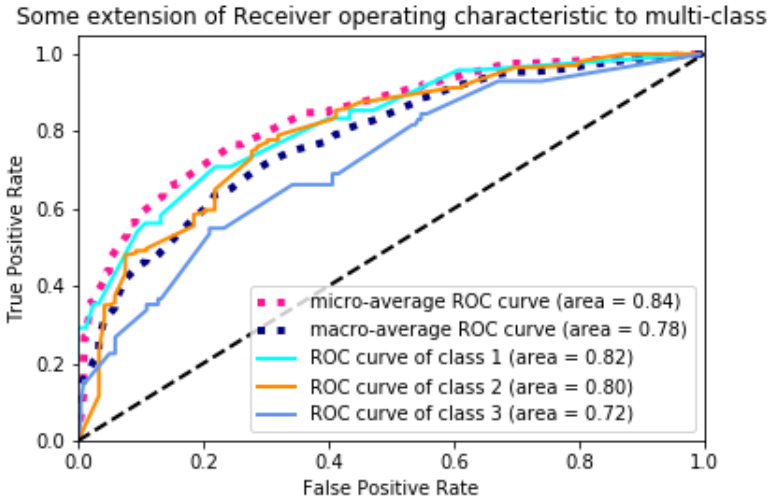

Figure 4: ROC Curve for Random Forest

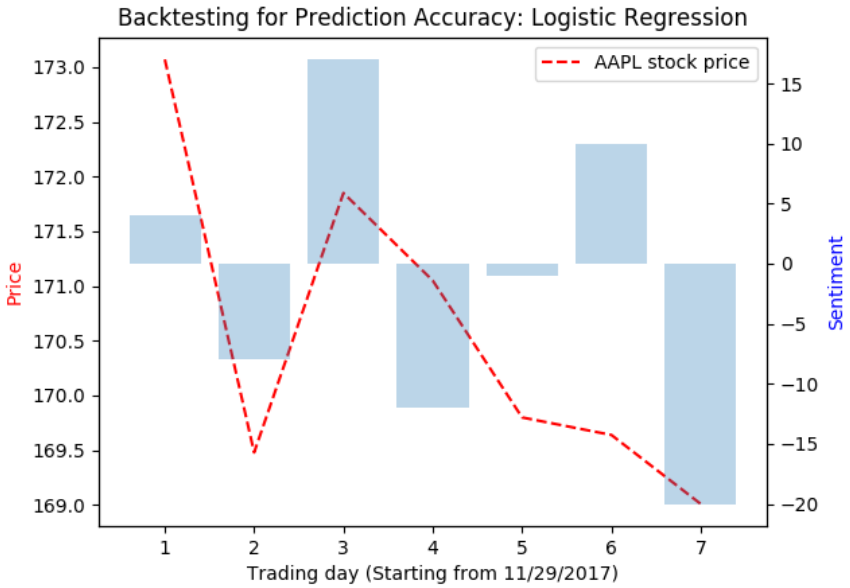

Figure 5: Prediction using Logistic Regression

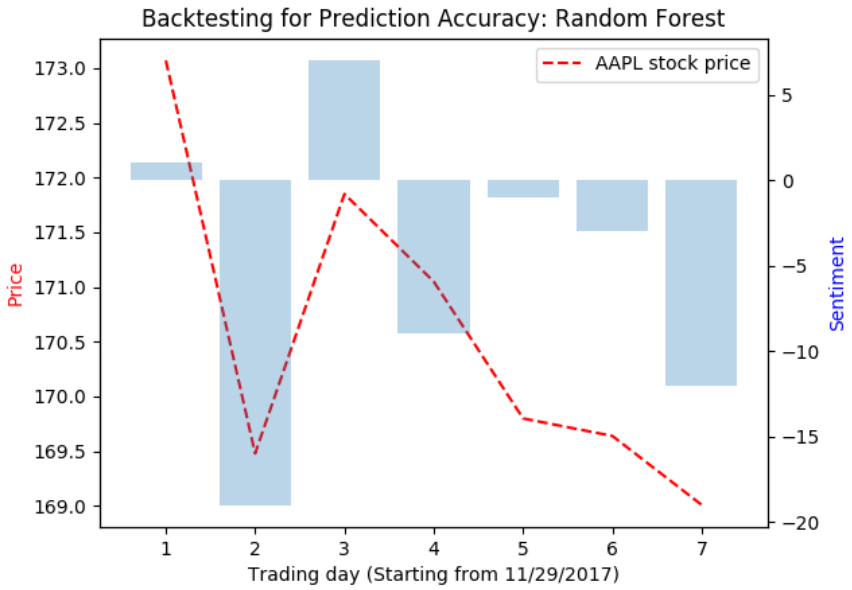

Figure 6: Prediction using Random Forest

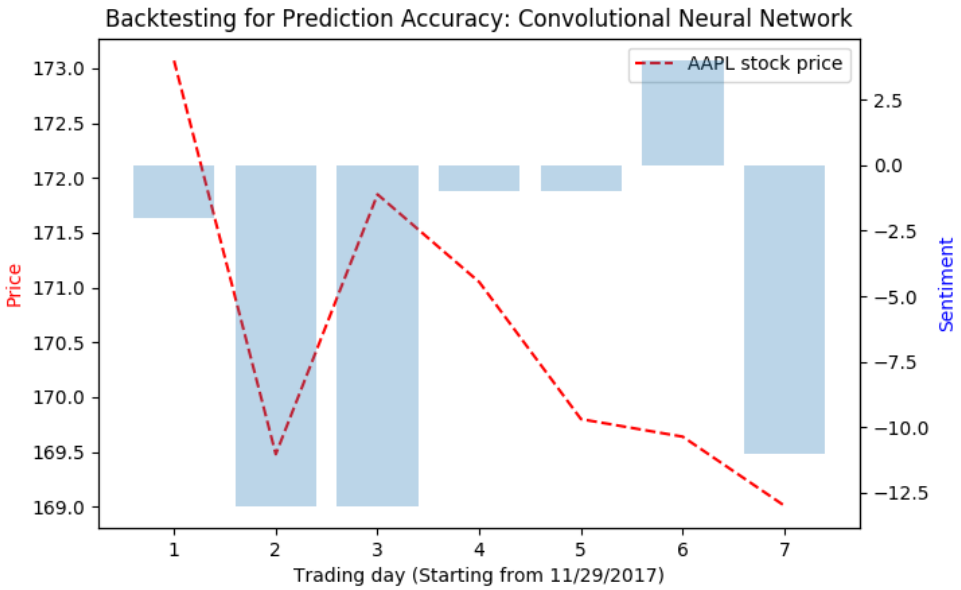

Figure 7: Prediction using Convolutional Neural Network