

Market Intelligence Analysis: Stock Prediction using StockTwits Dataset

Jia Ji
Columbia University, New York
Email: jj2860@columbia.edu
UNI: jj2860

Tianrui Peng
Columbia University, New York
Email: tp2522@columbia.edu
UNI: tp2522

I. INTRODUCTION

Stock market prediction has been a really important part of the market analysis. There has been a lot of research done related to stock market prediction. There are various ways to predict stock market such as analyzing new reports, analyzing social media data, and analyzing economic models. Some researchers believe that the price of stock market can be influenced by the moods and attitudes of general public. We usually call the moods of people toward a certain stock public sentiment. Based on a hypothesis of behavioral economics, many researchers believe that there is a direct correlation between public sentiment and stock market prices.

In this paper, we use sentiment analysis of social media data to extract the public sentiment. And then we analysis the extracted sentiment to get a popularity score and a sentiment score for each stock. People can use these scores as a reference for predicting the future stock market movement. There are many related researches, but almost all of them used the publicly available Twitter dataset [1], [2]. Even though Twitter dataset has been widely used and accepted, in this paper, we create and analyze a different dataset called StockTwits dataset. StockTwits.com is a website that is similar to Twitter, but it is stock-focused [3]. People only post messages about information related to stock or their stock exchange record. StockTwits has thousands of experienced traders that posting and sharing their investment ideas and suggestions. We believe that through analyzing the StockTwit data, we can get useful information about the public sentiment toward a stock, and use it to predict the general movement of the stock. Moreover, since we believe this a valuable dataset for stock prediction, we parsed the real-time StockTwits data and combined with its past data to build a database for StockTwits data. For this research, we design and build a StockTwits dataset, and we provide an easy to use API to query and use the information of this dataset. Future researchers can utilize our database and our API to analyze the correlation between public sentiment and stock market movement. They can also combine our data with the Twitter data to get a better prediction. We believe our database can be really useful for future researches related to this field.

In all the previous works, there is one research that performed big data analysis of StockTwits to predict sentiments

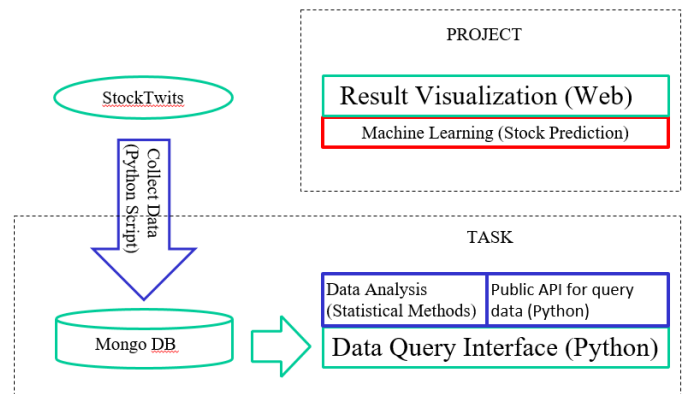


Fig. 1. General Flow of our Progress

in the Stock Market. This paper proved that they can use a machine learning model to predict the public sentiment of StockTwits. In their conclusion section, they mentioned that they believe their predicted public sentiment can be used as a reference for predicting the stock market movement. However, they did not show how to use these extracted sentiments for predicting stock market movement, and they did not provide any experiment result related to this. In our research, we not only extract public sentiment from StockTwits data, and also analysis the public sentiment to predict future stock market movement. We also evaluate our method using past data to get the accuracies.

There are four main parts of our research project. The first part is design and build a database for StockTwits data. We query real-time data through StockTwits API, and then analysis and extract information related to sentiment analysis and stock market prediction. Then we save the analyzed data into our database. We also create a convenient Python API for people to query various information from our database. Other researchers can easily use our API and database for building various machine learning applications to predict stock market. The second part is analyzing the information stored in our database. For each stock, we calculate a daily popularity score and a daily sentiment score. The popularity score is calculated based on various information such as message counts and

subscribers count. And the sentiment score is calculated based on sentiment analysis of the posts. For each hour, our program would automatically analysis the scores based on our current data, and then store these scores in our database too. The third part of our research is creating a standard user interface for people to check our prediction. We will use flask to create a website which provides daily updated stock market analysis. For each day, we will provide the 10 most popular stocks, 10 most bullish stocks, and 10 most bearish stocks. These results are created based on our analyzed popularity scores and sentiment scores. Users can use these information as a reference for predicting the movement of stock market. Figure 1 shows the general flow of our progress. For our final project, we used the information we collected in our database and the historical data to create a machine learning model for stock prediction. We also created a dynamic learning system which can automatically extract new data from StockTwits everyday, retrain the model itself, and use the new model to make prediction. This system is designed to continuously learning new data and information, and use these information to improve its accuracy.

II. DATASET

In this paper, we collect and analyze a dataset called StockTwits dataset. StockTwits.com is a website that is similar to Twitter, but all the posts are related to stocks [3]. People only post messages about information related to stock or their stock exchange record. StockTwits has thousands of experienced traders that posting and sharing their real-time investment ideas and suggestions. We believe that through analyzing the StockTwit data, we can get useful information about the public sentiment toward a stock, and can use it to predict the general movement of the stock.

III. DESIGN AND BUILD STOCKTWITS DATABASE

For this research, we design and build a StockTwits dataset, and we provide an easy to use API to query and use the information of this dataset. StockTwits dataset is a valuable dataset for stock prediction research, we parsed the real-time StockTwits data and combined with its past data to build a database. Future researchers can utilize our database and our API to analyze the correlation between public sentiment and stock market movement. They can also combine our data with the Twitter data to train their machine learning models for stock prediction.

A. Database Design

Our database is consist of these few collections: messages, stocks, and daily rank. Daily rank is our analysis result, which we saved to our database, and update every hour. We also query data from Yahoo Finance API to add price information of each stock into our dataset.

Figure 2 is the general flow of our database.

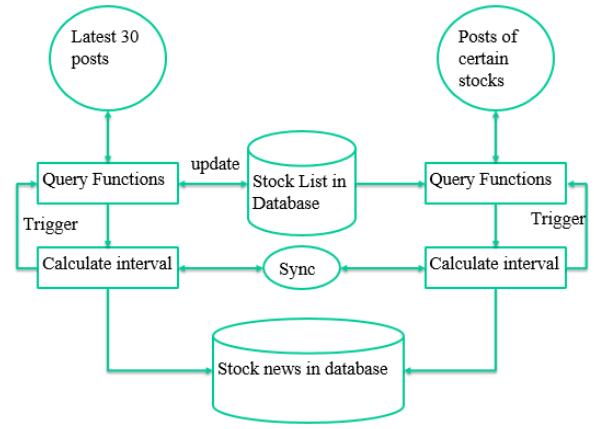


Fig. 2. General Flow of our database

```
if __name__ == '__main__':
    database = STDataset("user1", "abcd1234", "45.33.45.39", "bigdata")
    cl = database.findMessages()
    database.findDailyRank("2017-04-11")
```

Fig. 3. A simple example to use our database API

B. Data collection

In order to collect the data from StockTwits, we got a partner-level access to StockTwits' API. We wrote a script that can automatically query the StockTwits' API for real-time information. For each hour, we can query the StockTwit's API for 400 times, and we save all the information related to stock prediction to our database.

Our database is designed to automatically gather messages. It will try to find the newest 30 messages. Because of the hit rate limitation of StockTwits, we can only query their website 400 times an hour, and only get the newest 30 messages per query. From these messages, we extract symbols from all of them. We keep a set of all seen symbols. If during extracting symbols from the new messages, we find an unseen symbol, we will add that new symbol to our explored symbol set. And then, we used this new explored symbol set to find newest 30 messages for all these symbols. Through this method, both our symbol set and message set will grow automatically.

C. Query API

We also created a Python API to for user to easily access our database. Figure 3 shows an example to use our Python API. We will provide this API to any other teams that want to use StockTwits dataset for their project. The table I are list of functions that we provide through API.

IV. DATA ANALYSIS

After collecting and storing all the data in our StockTwits database, we start analyzing the data and extracting the public sentiment from these data. We query the data by using our Python API of StockTwits database. This can also be served as a good example of how to use our database API to analysis

TABLE I

A TABLE OF ALL THE FUNCTIONALITIES THAT WE PROVIDE THROUGH OUR API

Attribute	notation
findDailyMessage	find all the messages posted during the input date
findDailyRank	find all the analysis results for the input date. Ex: sentiment scores
getStockPrice	input a start date, end date, and the stock name. This function will provide open price, close price, max price, and minimum price for each stock
getCorrelation	input a list of stocks, and this function will provide the correlation of the stock price and our prediction

TABLE II

A TABLE OF ALL THE ATTRIBUTES WE GATHERED FOR CALCULATING MESSAGE SCORES

Attribute	notation	Detail
reshare_count	r	the number of people reshare this post
like_count	l	the number of people liked this post
followers_count	f	the number of followers of the message poster
subscribers_count	s	the number of subscribers of the message poster

StockTwits information. For each stock, we calculate a daily popularity score and a daily sentiment score. The popularity score is calculated using a formula that utilizes various information such as message counts and subscribers count. And the sentiment score is calculated based on sentiment analysis of the posts.

A. popularity score

1) *message score*: In order to calculate the popularity score for each stock, the first important information we gathered is the messages posted for that stock. We give a score to each message based on various information related to the message, such as how many like this message gets. We also account in the information related to the poster of the message. The most popular and active the poster is, the more score we give to that message. Table II is the information we collected for each message.

Then based on this information, we use the following formula to calculate the mes_popularity score m_i for each message.

$$m_i = 10 + r + l + f + s * 10 \quad (1)$$

In the formula, we use give each message a base score 10. Because even if nobody liked or shared the post, and the poster is a new user, the message should still get a base score. Then for each message, we will its reshared_count r, like_count l, followers_count f, and subscribers_count s. Also, we multiply the subscribers_count s by 10 because

TABLE III

A TABLE OF ALL THE ATTRIBUTES WE GATHERED FOR CALCULATING TOTAL POPULARITY SCORE

Attribute	Detail
total_message_score	the sum of all message scores
watchlist_count	the number of people following this stock
trending_score	the number to estimate whether this stock is trending or not given by StockTwits

subscribers means they will receive notifications if the poster post anything. The subscribers_count shows more importance than followers_count, because following a posters just means the followers would be able to see the posts but they won't be notified. Therefore, we decide to give more weight to the subscribers_count.

2) *Total popularity score*: After calculating all the message scores, we sum all of them and take a log of the sum. And then we add another two information related to each stock instead of individual messages. Table III is the attributes we used for calculating the total popularity score.

Then based on this information, we use the following formula to calculate the today_popularity score for each message.

$$tp = \log_2\left(\sum_{i=0}^k m_i\right) + \log_2 w + t * 10 \quad (2)$$

In the formula, we first get the sum of the all message_scores m_i , and then we take the log of the sum. k is the total number of messages posted about this stock. We take the log because this score tends to get really huge, and we want it to be scaled. And then we take the log of watchlist_count w for the same reason. In the end, we multiply the trending_score t by 10 because it is usually a really small number. We tried to scale these three numbers so they all have a reasonable weight.

$$dp = tp * 0.75 + yp * 0.25 \quad (3)$$

After calculating today_popularity score tp, we also count past popularity score yp. Today's score gets weight 0.75 and yesterday's score gets weight 0.25. Because yesterday's score is also calculated use the day before yesterday's score, so this imply that the day before yesterday's score gets weight $(0.25)^2$. Therefore, we can see as the days get older, their contributions to the current popularity score go down. So we used the following formula to calculate the daily_popularity dp for each stock.

B. sentiment score

1) *message sentiment score*: Similar to the popularity score, we give each message a sentiment score ms based on sentiment analysis of the message body and the tags. If based on the sentiment analysis, a message is bearish, we used the following formula:

$$ms_i = -10 * \log_2(m_i) \quad (4)$$

And if a message is bullish, we used a similar formula:

$$ms_i = 10 * \log_2(m_i) \quad (5)$$

To calculate ms_i , We weighted each message by its message popularity score m_i , because if a message is posted and reshared by a lot of the people, it can represent a stronger public sentiment. We also take a log of the popularity scores because we do not want each message sentiment score ms get too large by multiplying m_i . For each message, if it is bullish, it will get a base sentiment score 10. On the other hand, if it is bearish, it will get a base sentiment score -10. Therefore, an overall positive sentiment score implies that this stock is bullish, and an overall negative sentiment score implies that this stock is bearish.

2) *total sentiment score*: For calculating the daily sentiment score ds , we also first calculate the today's total sentiment score ts , and then add the weighted yesterday's sentiment score ys . Today's total sentiment score ts is the sum of all message sentiment scores ms_i . The total number of messages is k . Similar to the popularity score, as the days get older, their contributions to the current sentiment score also go down. Here are the formulas we used to calculate the daily_sentiment score.

$$ts = \sum_{i=0}^k (ms_i) \quad (6)$$

$$ds = ts * 0.75 + ys * 0.25 \quad (7)$$

C. Daily Rank Lists

By calculating the popularity scores and sentiment scores, we can give a list of top 10 most popular stocks, a list of top 10 most bullish stocks, and a list of top 10 most bearish stocks. For example, here are the three lists we got for March 20, 2017.

Top 5 popular stocks:

stock: DUK, popularity score: 23.687876306
stock: AMGN, popularity score: 23.6721293661
stock: CTSB, popularity score: 23.4721210623
stock: UNP, popularity score: 21.8510142453
stock: MMM, popularity score: 21.6931601479

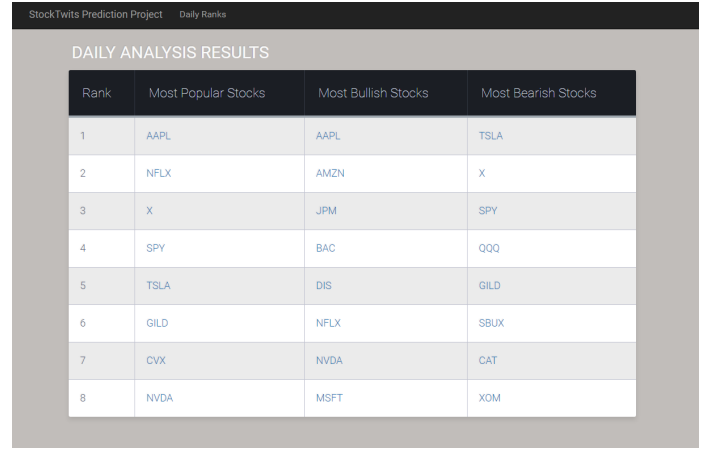
Top 5 Bullish stocks:

stock: NVDA, sentiment score: 10364.0736287
stock: AAPL, sentiment score: 6863.09809951
stock: DIS, sentiment score: 2343.64813327
stock: AMZN, sentiment score: 1929.0605209
stock: SPY, sentiment score: 1585.35898829

Top 5 Bearish stocks:

stock: QQQ, sentiment score: -618.98469022
stock: X, sentiment score: -179.545631625
stock: MMM, sentiment score: -74.9715096773
stock: D, sentiment score: -62.743473381
stock: CMCSA, sentiment score: -47.8901525697

For each hour, our program would automatically analysis the scores and the daily rank lists based on our current data, and



The screenshot shows a web browser window with the title 'StockTwits Prediction Project - Daily Rank'. The main content area is titled 'DAILY ANALYSIS RESULTS' and contains a table with four columns: 'Rank', 'Most Popular Stocks', 'Most Bullish Stocks', and 'Most Bearish Stocks'. The table lists the top 10 stocks for each category on a specific day.

Rank	Most Popular Stocks	Most Bullish Stocks	Most Bearish Stocks
1	AAPL	AAPL	TSLA
2	NFLX	AMZN	X
3	X	JPM	SPY
4	SPY	BAC	QQQ
5	TSLA	DIS	GILD
6	GILD	NFLX	SBUX
7	CVX	NVDA	CAT
8	NVDA	MSFT	XOM

Fig. 4. The main page of our website

then store these scores in our database too. Future researchers can access this lists from our database too, if they find this information useful.

V. DATA VISUALIZATION AND WEB INTERFACE

The third part of our research is creating a standard user interface for people to check our prediction. We will use flask to create a website which provides daily updated stock market analysis. For each day, we will provide the 10 most popular stocks, 10 most bullish stocks, and 10 most bearish stocks. These results are created based on our analyzed popularity scores and sentiment scores. Users can use these information as a reference for predicting the movement of stock market. We hosted our website on our server using Linode and Apache service, you can access our website through the following url: <http://45.33.45.39:8050/>. Please contact us if you encounter any problem while accessing this website, because our server might be restarting or being used for other purpose.

There are three main parts of web interface. The first part is the daily rank page. This page shows the top 10 most popular stocks, most bullish stocks, and most bearish stocks. The second part is the individual page for each stock. On each stock's page, you can see the messages posted on StockTwits for each stock. The third main part is the graphs that show the past weeks' price and score changes.

A. Daily rank page

As shown in Figure 4. This is the main page of our website. This page shows the top 10 most popular stocks, most bullish stocks, and most bearish stocks. Users can also click on the stock symbols to enter their individual stock pages.

B. Individual Stock page

As shown in Figure 5. This is the individual stock page of our website. This page shows the current popularity score and sentiment score of this stock. Also, it shows our current accuracy of predicting this stock. And we also show a message board that provides new messages posted on StockTwits related to this stock.

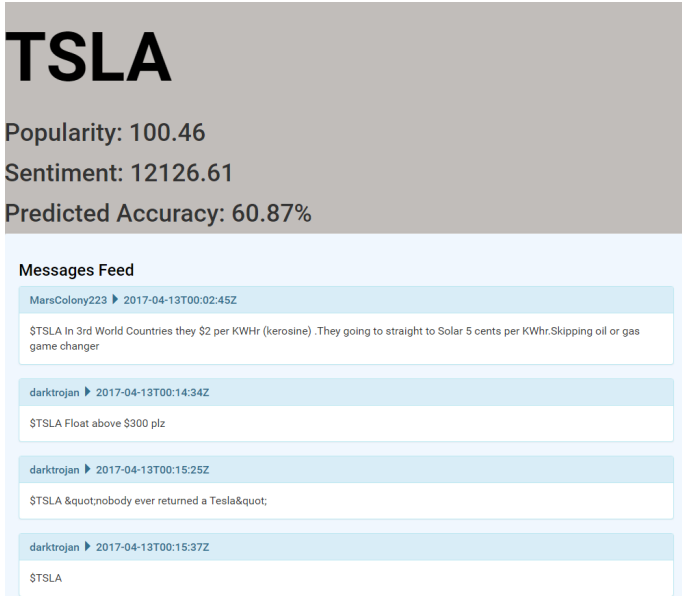


Fig. 5. An example stock page of our website

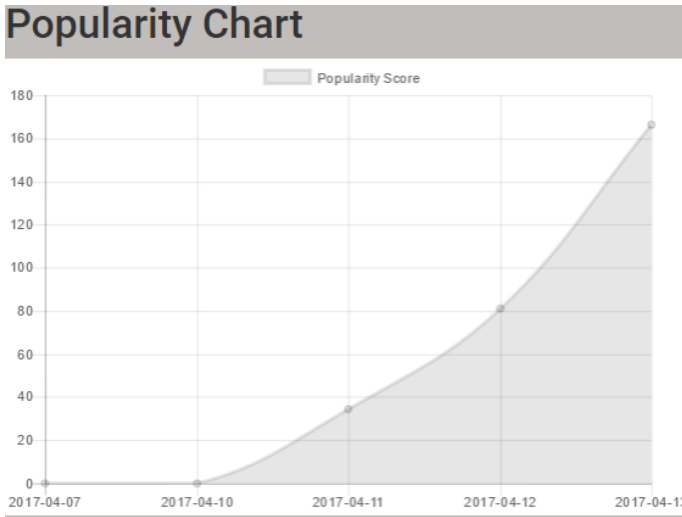


Fig. 6. An example of popularity plot

C. Graphs

1) *Popularity Chart*: As shown in Figure 6, this is an example of our popularity chart. For this figure, we used the popularity score of U.S. Steel's stock from April 7th to April 13th. You can see the we remove April 8th and April 9th from our plot because these two days are weekends and the market was closed. Therefore, We didn't include weekends in all of our graphs.

2) *Sentiment Chart*: As shown in Figure 7, this is an example of our sentiment chart. For this figure, we used the sentiment score of U.S. Steel's stock from April 7th to April 13th. We also didn't include weekends in this graph. We can see from this chart that the public sentiment for 'X' this week is mostly negative.

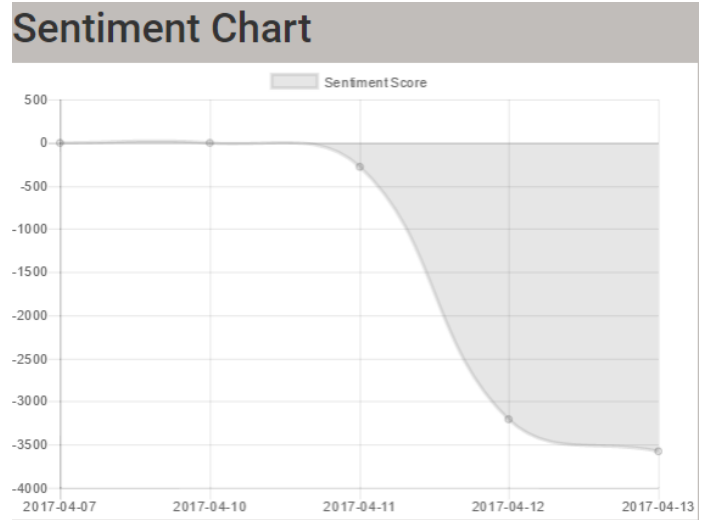


Fig. 7. An example of sentiment plot

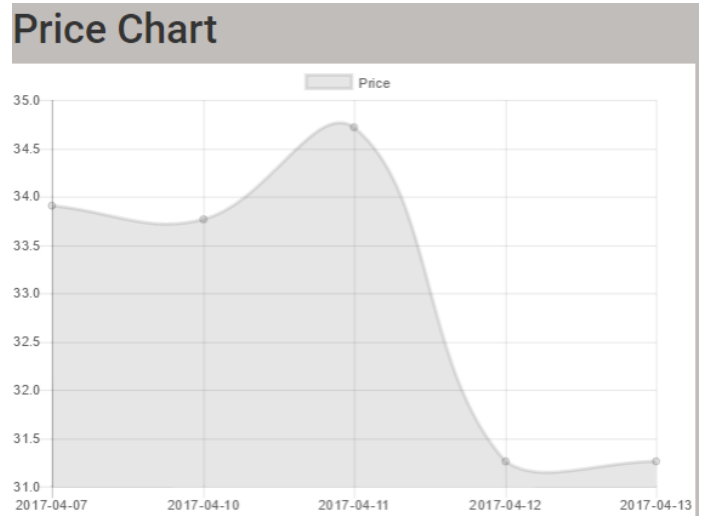


Fig. 8. An example of price plot

3) *Price Chart*: As shown in Figure 8, this is an example of our price chart. For this figure, we used the market close price of Stock 'X' from April 7th to April 13th. We also didn't include weekends in this graph. We can see from this chart that the price actually went down from April 11th to April 13th. In this example, we can definitely see a correlation between the sentiment chart and the price chart. Even though there are other examples that the sentiment charts cannot represent the price chart at all, but we can find a lot of cases that public sentiment can be used for predicting market trending.

VI. EVALUATION

We will evaluate our scores by comparing our prediction with the actual movement of the stock market. We used two main methods to evaluate our prediction. The first method is calculating the accuracy of our binary prediction, and the

second method is calculating the correlation between the scores and the actual prices.

A. Accuracy

We evaluate our model by calculating the accuracy of our prediction. So for each stock, we predict the market price will go up tomorrow if today's public sentiment is positive, and we predict the market price will go down tomorrow if today's public sentiment is negative. And We use the market close price every day as our measurement. We create a function that will measure the accuracy of our prediction for each stock for any given time period. For example, if user want to know the accuracy of our prediction for March, they just need to set the start state as March 1st, and the end date as March 31th. Then our function will calculate and provide the accuracy to the user.

B. Correlation

We also evaluate our model by calculating the correlation of our prediction. We calculate the correlation between the popularity score and the real stock price, and the correlation between sentiment score and the real stock price.

To calculate the correlation, we use the Pearson product-moment correlation coefficients, which is "relationship between the correlation coefficient matrix R, and the covariance matrix C." Formula 8 is the formula we used for calculating the correlation.

$$R_{ij} = \frac{C_{ij}}{\sqrt{C_{ii} * C_{jj}}} \quad (8)$$

C. Results

Figure 9 and 10 are our evaluation results. We used one month data that we gather this semester for calculating our evaluation results. The "corr vs pop" column shows the correlation between popularity score and market price. The "corr vs senti" column shows the correlation between sentiment score and market price. And the "predict" column shows the accuracy of our prediction. Figure 9 is a table sorted by highest to lowest accuracy. Figure 10 is a table sorted by the "corr vs senti" column.

VII. APPLYING MACHINE LEARNING

For our final project, we used all the data and analysis that we got from our task to train a machine learning model for stock prediction. We hope that through machine learning, we would be able to get a better accuracy for stock trend prediction.

A. Training and testing Dataset

For the training dataset, we used the historical data of 2015 that we got from StockTwits. We were unable to obtain the historical data of 2016 from StockTwits even though we emailed them and asked for whether we can get it. Therefore, even though using data from 2015 probably cannot get us the best results, we decided to use these data as a proof of concept. We picked the same 44 stocks that we used for our previous

symbol	corr vs pop	corr vs senti	predict
CTL	-0.1288	0.3730	68.75
X	0.0976	-0.0254	64.29
GE	0.0426	-0.1676	62.50
MMM	-0.0118	0.3661	62.50
EIX	-0.0089	-0.3610	62.50
CVX	0.2031	-0.4645	60.00
XOM	0.1633	0.3114	56.25
BA	-0.3121	-0.0449	56.25
TSLA	-0.0591	0.0370	56.25
GILD	0.2721	-0.0837	56.25
MSFT	0.3438	0.0591	56.25
CAT	-0.3748	-0.1039	53.33
PXD	-0.3815	-0.3916	50.00
COP	-0.0451	0.7135	50.00
FCX	0.3950	0.1347	50.00
IP	-0.1668	-0.5080	50.00
VMC	-0.0700	-0.3610	50.00
UNP	0.1055	-0.1613	50.00
DIS	0.3867	-0.2512	50.00

Fig. 9. Our evaluation results sorted by accuracy

tasks, and we got 10,736 data points from the whole historical dataset. After preprocessing and clean the dataset, we used 80 percent of the dataset for training, and 20 percent of it for testing.

We were worried that just using the historical data from 2015 might not be enough for creating an accurate model. Therefore, we designed a dynamic learning system, which dynamically included the newest data we obtain from 2017. So everyday, we collect data from StockTwits, analyze, and processes it. Then, we feed both the historical data and new 2017 data to train our machine learning model. Our hope is that as our model get more and more data every single day, its accuracy would also slowly increase.

B. Machine learning features

The following table IV lists all the features that we extract from the raw data, and used as our features to train our machine learning model. Most of these features are really similar to what we used to calculate the popularity scores and sentiment scores. If you want to understand each feature in detail, please check table II and table III.

symbol	corr vs pop	corr vs senti	predict
COP	-0.0451	0.7135	50.00
T	0.1874	0.4416	40.00
CTL	-0.1288	0.3730	68.75
MMM	-0.0118	0.3661	62.50
C	0.0990	0.3251	46.67
XOM	0.1633	0.3114	56.25
CMCSA	0.2269	0.3018	50.00
SBUX	0.4342	0.2833	43.75
NVDA	0.0462	0.2471	43.75
QQQ	0.2834	0.2341	37.50
AMZN	0.3657	0.1794	50.00
FCX	0.3950	0.1347	50.00
VZ	-0.0963	0.1157	37.50
BAC	0.2559	0.1024	37.50
LYB	0.2315	0.0822	33.33
AAPL	0.3183	0.0734	37.50
MSFT	0.3438	0.0591	56.25
JNJ	-0.1665	0.0381	25.00
TSLA	-0.0591	0.0370	56.25

Fig. 10. Our evaluation results sorted by correlation of sentiment and price

TABLE IV
A TABLE OF ALL THE FEATURES WE USED

features	Example
Total likes	698
Total reshares	10
Total post users	441
Total followers	24786
Total subscribers	5815
Watchlist Count	91084
Trending score	0.58
Total number of messages	1601
Total number of bullish messages	346
Total number of bearish messages	135
Date	2015-01-02

C. Machine learning models

For machine learning model, we picked Naive Bayes and Neutral Networks. We picked these two model because after reading related papers and studies, these are two commonly used model for stock market prediction.

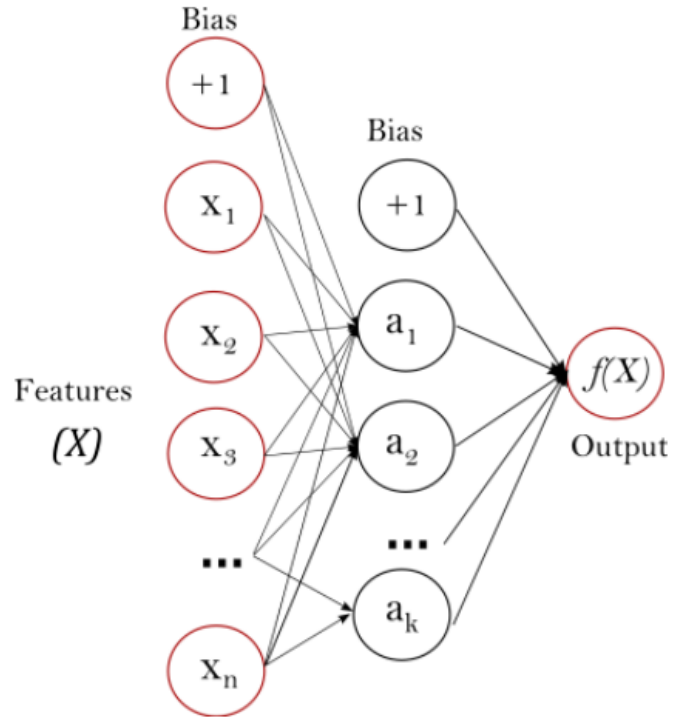


Fig. 11. Multi-layer perceptron

1) *Naive Bayes*: Naive Bayes is a classification technique based on Bayes' theorem. This model is really useful for analyzing very large dataset because it's comparatively faster to train. Also, this model naturally gives a confidence score for the prediction, which can be really useful information for stock prediction.

2) *Neural Networks*: We also applied neural networks, more specifically, multi-layer perceptron (MLP) model to analyze our data. MLP is a supervised learning algorithm as shown in the figure 11. We used Scikit-learn for building both of the models [4]. After using grid search for finding the best hyper-parameters, we found out that using 2 hidden layers with 5 neurons each gives us the best accuracies.

D. Dynamic Learning System

We also created a dynamic learning system to let our machine learning continuously get the newest data, and automatically improve itself everyday. As shown in figure 12 Every single day, this system would retrieve the newest data from StockTWits website, clean the raw data, analysis and preprocesses the data, and generate features. Then we will feed both the new data, and all the historical data to train a new model. And then we use this newest model to make predictions for tomorrow's stock market trend. After getting the new prediction, we would also update our website, so our users can get the newest prediction from us. We set the system to automatically go through this whole process and create a new model every day at 7pm. So user can use these information to make their investment decision for the next day.

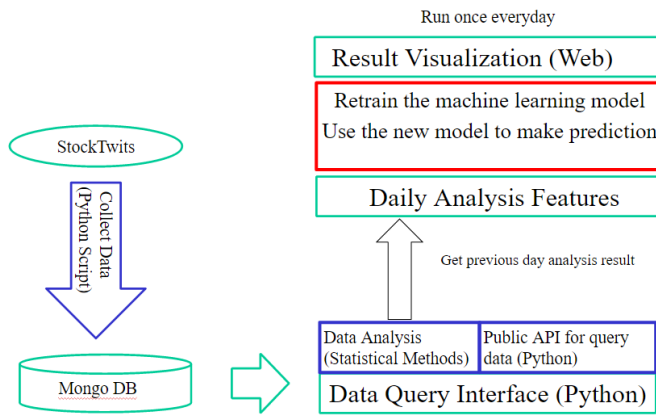


Fig. 12. Dynamic Learning System Process

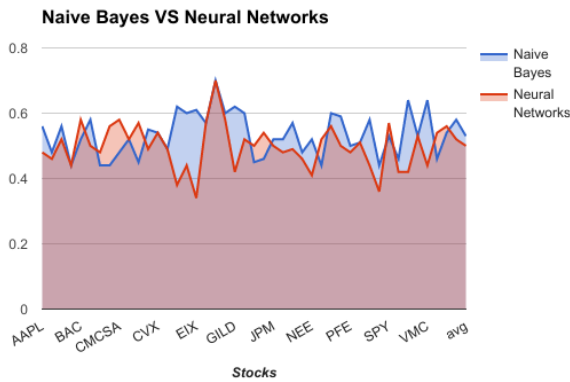


Fig. 13. Plot of the accuracies of Naive Bayes model and Neural Networks model. The vertical axis is the accuracies, and the horizontal axis is the stocks.

E. Results

Figure 14 shows the testing accuracy for training on 80 percent of the 2015's data, and testing on remaining 20 percent. The average accuracy for Naive Bayes model is 0.55. And the average accuracy for the neural networks model is 0.5. We plotted the accuracies of these two model in figure 13. From this figure, we can see that the Naive Bayes model got slightly better results. We understand that the accuracy is not too promising. Therefore, we created the dynamic learning system to improve the performance of our machine learning model. Our hope is that as our dynamic learning system gets more and more new and up-to-date data from 2017, our machine learning model would improve itself, and get better accuracy than our current model.

F. Web interface

Every day, after retrained our model and used it to make the newest predictions, we update our website. From our interface, users can see the newest prediction, and also the corresponding confidence scores generated by our model. As shown in figure 15, we created a new web-page for showing the results, which shows all the symbols, the prediction, and the confidence

DAILY ANALYSIS RESULTS		
Symbol	Prediction	Confidence Score
AAPL	Bullish	100.00
AMGN	Bullish	89.56
AMZN	Bullish	99.97
BA	Bullish	100.00
BAC	Bearish	80.92
BMJ	Bullish	99.90
C	Bullish	77.07
CAT	Bullish	80.30
CMCSA	Bearish	97.44
COP	Bearish	77.95
CTL	Bearish	99.62
CTSH	Bullish	57.92
CVX	Bearish	86.45
D	Bearish	79.48

Fig. 14. Table of Accuracies for our Machine Learning Model

score. Our users can use this page as a reference to make their investment decision for the next day.

G. Compare with Microsoft Sentiment Analysis

After our final demo, based on professor's feedback, we also conduct a comparison between our system and Microsoft Azure sentiment analysis. We accessed Microsoft Azure sentiment analysis through their text analytics API. We registered an free account which allows us to perform 5,000 calls per 30 days to Azure sentiment analysis. Because we are limited by the number of calls we can make for free using Microsoft's API, We tested both our model and Microsoft sentiment analysis on one month of 2017's StockTwits data.

For each day, we feed all messages to both Microsoft analysis and our model, and let both of them predict tomorrow's stock market trend. Then we compare the predictions with the ground truth. Figure 16 shows the table of Microsoft's accuracy and our model's accuracy.

Based on the figure, for predicting the stock market trending of March 2017, the average accuracy of Microsoft Azure is 0.70 , and our model's average accuracy is 0.72. Our model has a slightly better accuracy, this might because our model only trained on related stock market message dataset, while Azure probably trained on more noisy and unrelated dataset. Therefore, we can perform slightly better at predicting the sentiment of messages related to stock market. However, in order to truly compare the accuracy of Microsoft and our model, we need to test them on a larger dataset. Even though because of time and resource constrains, we can't perform

Stocks	Naive Bayes	Neural Networks
EIX	0.61	0.34
EXC	0.57	0.57
FCX	0.7	0.7
GE	0.6	0.58
GILD	0.62	0.42
HAL	0.6	0.52
PFE	0.5	0.48
PXD	0.51	0.51
QQQ	0.58	0.44
SBUX	0.44	0.36
SPY	0.53	0.57
T	0.46	0.42
TSLA	0.64	0.42
UNP	0.53	0.53
VMC	0.64	0.44
VZ	0.46	0.54
X	0.54	0.56
XOM	0.58	0.52
Average	0.55	0.5

Fig. 15. Web Interface for our Machine Learning Model

a larger evaluation, this could be an interesting task for future research.

VIII. COLLABORATION WITH OTHER TEAMS

We collaborated with another group B3 who is also doing market intelligence analysis. Zijun Hao and Qing Zhou kindly offered their sentiment analysis program for analyzing social media message to us. So when some stockTwits's messages are not tagged with sentiment, we would use their model to perform sentiment analysis instead. We also collaborated with group B9, who want to use our data for optimizing user portfolio. We provided our website and API to her.

IX. CONCLUSION AND FUTURE WORKS

Based on our evaluation result, we can see that there are correlations between public sentiment extracted from StockTwits and the stock market trend. It shows that StockTwits dataset has the potential to be used for future stock prediction projects or used to train machine learning models.

For the future project, it would be really interesting to combine the Twitter dataset with the StockTwits dataset for

Stocks	Microsoft Accuracy	Our Accuracy
AAPL	0.823529412	0.823529412
AMGN	0.529411765	0.764705882
AMZN	0.764705882	0.764705882
BA	0.647058824	0.588235294
BAC	0.647058824	0.647058824
BMJ	0.647058824	0.647058824
C	0.705882353	0.764705882
CAT	0.705882353	0.705882353
CMCSA	0.588235294	0.764705882
COP	0.705882353	0.764705882
CTL	0.647058824	0.764705882
CTSH	0.823529412	0.823529412
CVX	0.588235294	0.764705882
D	0.588235294	0.647058824
DIS	0.647058824	0.647058824
DUK	0.647058824	0.588235294
EIX	0.647058824	0.705882353
EXC	0.764705882	0.588235294

Fig. 16. Comparison of Microsoft's Accuracy and Our model's Accuracy

training machine learning model. We hope to achieve better accuracy than models that just used twitter dataset. Also, it would be interesting to incorporate other types of data such as news into our system. We can see a lot of potential usage of the StockTwits dataset for predicting and analyzing stock market.

REFERENCES

- [1] Stock Market Prediction Using Twitter Sentiment Analysis. pdf (2012).
- [2] Si, Jianfeng, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. "Exploiting Topic based Twitter Sentiment for Stock Prediction." ACL (2) 2013 (2013): 24-29.
- [3] Si, Jianfeng, et al. "Exploiting Topic based Twitter Sentiment for Stock Prediction." ACL (2) 2013 (2013): 24-29.
- [4] Pedregosa, Fabian, Gal Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12, no. Oct (2011): 2825-2830. <http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>
- [5] Schumaker, Robert P., and Hsinchun Chen. "Textual analysis of stock market prediction using breaking financial news: The AZFin text system." ACM Transactions on Information Systems (TOIS) 27, no. 2 (2009): 12.
- [6] Yoo, Paul D., Maria H. Kim, and Tony Jan. "Machine learning techniques and use of event information for stock market prediction: A survey and evaluation." In Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on, vol. 2, pp. 835-841. IEEE, 2005.
- [7] Preis, Tobias, Helen Susannah Moat, and H. Eugene Stanley. "Quantifying trading behavior in financial markets using Google Trends." (2013).
- [8] Stefan van der Walt, S. Chris Colbert and Gal Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation." Computing in

Science and Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37
<http://www.numpy.org/>

- [9] Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, <http://www.scipy.org/> [Online; accessed 2017-03-07].
- [10] Python Software Foundation. "Python Language Reference, version 2.7." Python Software Foundation. May. 1995. <http://www.python.org>
- [11] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E."Installing Scikit-learn." Google Machine Learning Group. Mar. 2017. <http://scikit-learn.org/stable/install.html>