4/16/16
Group 8: Alex B., Gabrielle C., Adam F., Pit K., Ben R.

TO 412 – Final Project Write-Up

Introduction:

This project aims to display the relationship between **stock data** and **Twitter tweets** about a single company. We chose to analyze Tesla stock because of its volatile and speculative, emotion-driven nature. In particular, we hope to investigate the relationship between Twitter sentiment and stock price movements for Tesla (TSLA). Below, we've outlined the steps we've taken for our analysis, as well as some of the challenges we faced. Please feel free to use this material for your class next year!

Sources:

Our data sources are categorized into three major areas, including financial, contextual, and third-party sentiment scoring sources. To gather financial data on Tesla' stock prices, we used Yahoo! Finance to obtain stock returns data and StreetInsider to obtain earnings data. We downloaded a total of 1,457 records from June, 2010 to April, 2016. From these two sources, we are able to obtain 5 fields of financial data:

Date, Tesla Close, S&P Close, TSLA Returns, S&P Returns

To find relevant tweets, we used Klout, a web app that uses social media analytics to rank Twitter users according to their social influence. We identified three relevant expert groups with Tesla: Electric Car Enthusiasts, Environmental Activists, and Tesla Motors Enthusiasts. We then extracted tweets from Twitter, specifying tweets from the 35 influencers that obtain a list of Tesla relevant words. Each of the influencers that made our final list had at least 5,000 followers, as we felt their tweets would have the most amount of influence on the TSLA.

For our sentiment analysis algorithm, we obtained a third-party source that contained a list of words, along with a sentiment score. There are 18,543 words in this databased and each is rated with a sentiment score that ranges from -1 to 1. "-1" indicates the most negative word while "1" indicates the most positive word. Along with the sentiment score, we downloaded a total of 18,330 tweets and obtained 4 fields of tweet info:

User_screen_name, created_at, followers_count, retweet_count

A key goal of our project was to get tweet data that dated back to the inception of TLSA as a publicly traded company. This was difficult at first because the Tweepy and Twitter APIs limited us to download only 200 of the most recent tweets for each user in our list. This did not give us enough information dating back to 2010. To solve this, we used a somewhat confusing technique/"hack" that worked around Twitter's 200 tweet limit. It saves the tweet ID from the last tweet captured and then starts the next query with that last ID, this way we can make multiple queries that allowed us to collect over 18000 tweets (lines 51-71 in the code) (Figures 10 and 11). A summary of the data can be found in (Figure 1).

Sentiment Analysis Algorithm

We used a GitHub sentiment analysis algorithm (Figure 17). It creates a dictionary object with each word and its corresponding score from -1 to 1, it checks each word in each tweet and if the word is in the dictionary, the sentiment score for that word is added to the sentiment score value for that tweet. For each Tweet, we aggregated the sentiment score with the following score:

```
if word in sentiment.keys():
        sent_score = sent_score + float(sentiment[word])
else:
        sent_score = sent_score
```

SQL Joining Algorithm

Before joining the two data files, the Twitter data and the stock data, we calculated the "idiosyncratic movement," which is the stock movement beyond what we would expect. Our method is outlined in the "Analysis" section.

On SQL, we joined the two data files by the date of the stock record ("Date") and the date the tweets were created on ("Created_at") using the following query (Figure 18):

```
SELECT *
FROM TeslaOutput2 O, TeslaStockData T
WHERE O.created_at = T.date
```

Analysis

In order to accurately match sentiment data with stock data, we had to extrapolate macro-economic impacts on the TSLA stock price, since none of these impacts are likely related to Twitter. We did this using the following formula:

$$Idiosyncratic\ Return = r_T - \beta_T * r_M$$

$$r_T = TSLA\ Return(log(r_T/r(_T-1))$$
$$\beta_T = TSLA\ Beta(Risk)$$
$$r_M = Market\ Return$$

This formula reduces the stock return to Tesla's idiosyncratic return, i.e. the return that is solely due to events internal to Tesla. After adjusting the returns, we were able to calculate the correlation between sentiment scores and the stock price.

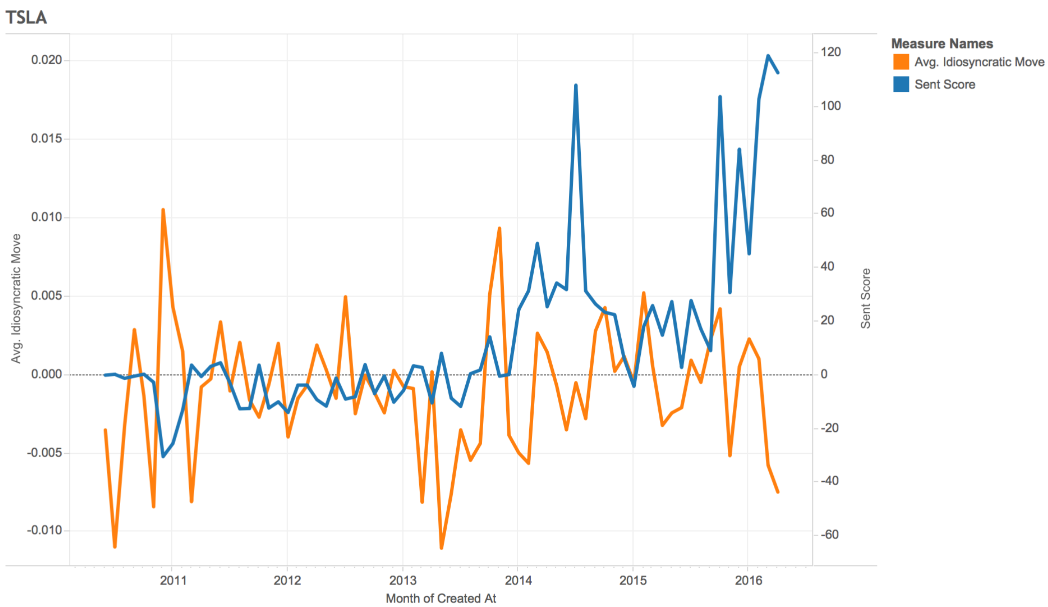
Note: we attempted to adjust for a small time lag (i.e. tweets today impact the stock price tomorrow or vice-versa). However, since the result were not significantly different, we decided to leave it out.

Results

Our results were dismal and discouraging (Figure 4). Below, you will find the end result of our analysis.

When we correlated the daily idiosyncratic move and daily sentiment score, we found a value of **R = 0.1993**.

Though we tried many variations of this analysis, ultimately, we found a very low correlation between the idiosyncratic move and the Twitter score (Figures 12-16). Upon further inspection, we discovered that over 80% of TSLA is owned by institutional investors such as Vanguard and BlackRock, and insiders, such as Elon Musk (Figure 2). We rationalized that these types of investors would not trade stock on sentiment, particularly those on Twitter, but would likely employ a buy-and-hold strategy.



The trends of Avg. Idiosyncratic Move and Sent Score for Created At Month. Color shows details about Avg. Idiosyncratic Move and Sent Score.

Nonetheless, we found a few notable results:

- There is a clear association between the number of Tesla tweets and the sentiment scores: people are tweeting more about Tesla and have a more positive view towards Tesla since 2014 (Figures 5 and 6)
- Average sentiment score follows a trend with local optimum at months where Tesla releases earnings reports (Figure 7)
- Similarly, the months before these earnings releases appear to have the lowest sentiment scores (Figure 7)
- @TeslaModelS had the lowest sentiment score and one of the fewest retweet counts among users, while those with higher sentiment scores tended to have higher retweet counts (Figures 8 and 9)
- Some days had major event-driven outliers for sentiment scores on Twitter (Figure 1)

<u>Further Analysis/Conclusion:</u>

For our thoughts on further analysis, please refer to the last page of our write-up.

Appendix:

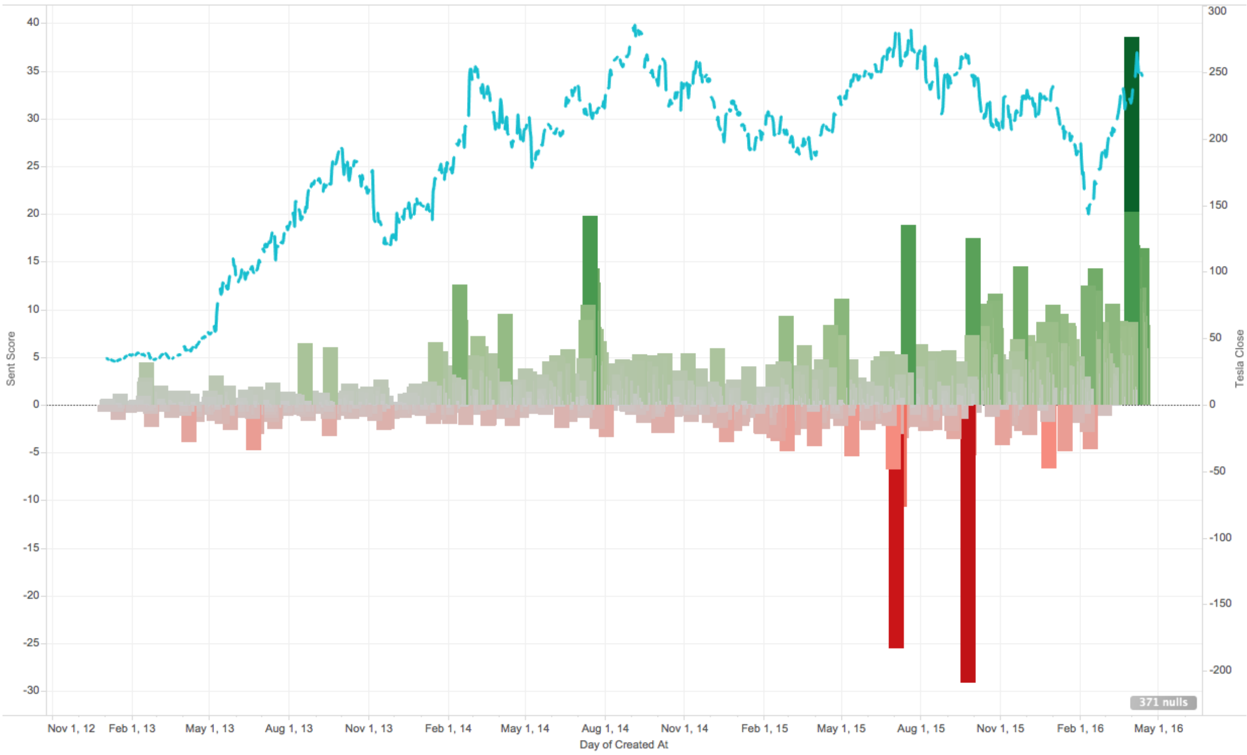


Figure 1: TSLA close price and daily sentiment score

Above is an overall summary of TSLA stock data and sentiment score for the last three years. Certain major events can be shown in the peaks of both directions for the sentiment scores.

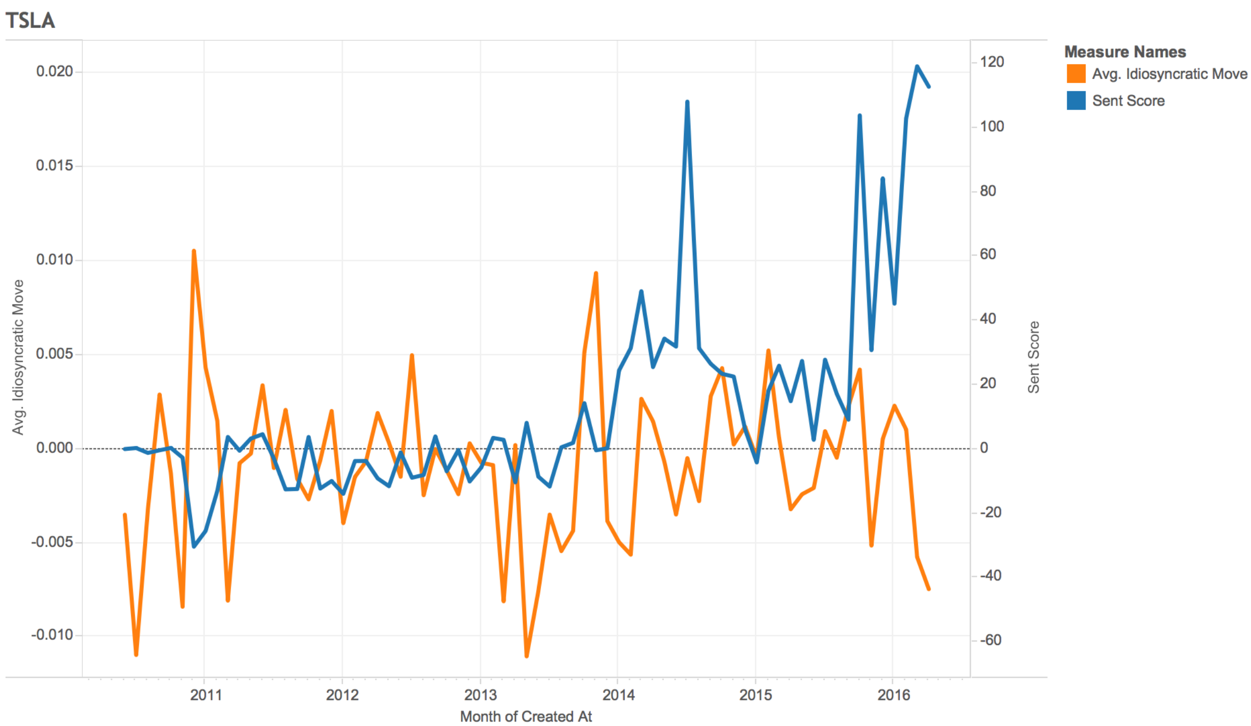
Breakdown	
% of Shares Held by All Insider and 5% Owners:	22%
% of Shares Held by Institutional & Mutual Fund Owners:	62%
% of Float Held by Institutional & Mutual Fund Owners:	80%
Number of Institutions Holding Shares:	572

Figure 2: Tesla Major Investors
Source: http://finance.yahoo.com/q/mh?s=tsla+Major+Holders

Breakdown	
% of Shares Held by All Insider and 5% Owners:	10%
% of Shares Held by Institutional & Mutual Fund Owners:	37%
% of Float Held by Institutional & Mutual Fund Owners:	41%
Number of Institutions Holding Shares:	260

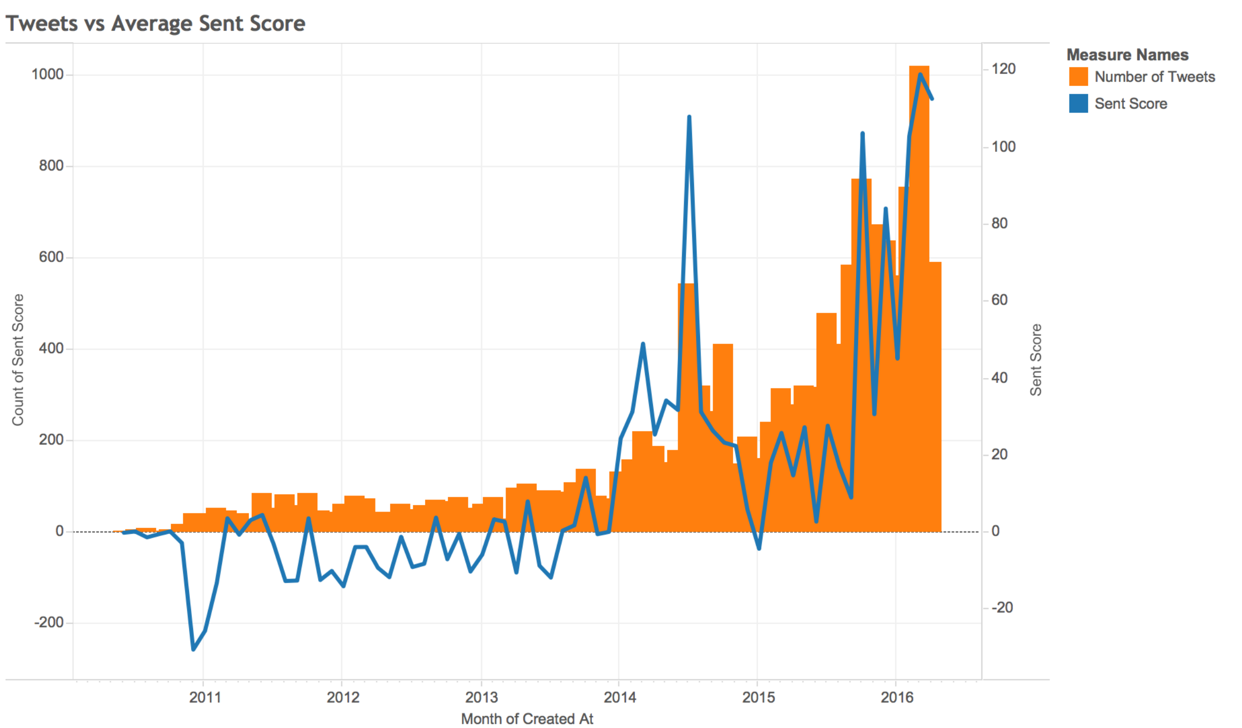
Figure 3: GoPro Major Investors
Source: https://finance.yahoo.com/q/mh?s=GPRO+Major+Holders

GoPro (GPRO) might be more volatile -- only 41% of it is held by institutional investors (compared to 80% for TSLA). A future project might yield more "actionable" insights with a more volatile stock which is held less by institutions and more by regular people.



The trends of Avg. Idiosyncratic Move and Sent Score for Created At Month. Color shows details about Avg. Idiosyncratic Move and Sent Score.

Figure 4: TSLA idiosyncratic move vs. sentiment score



The trends of Number of Tweets and Sent Score for Created At Month. Color shows details about Number of Tweets and Sent Score.

Figure 5: Tesla sentiment score vs. number of tweets

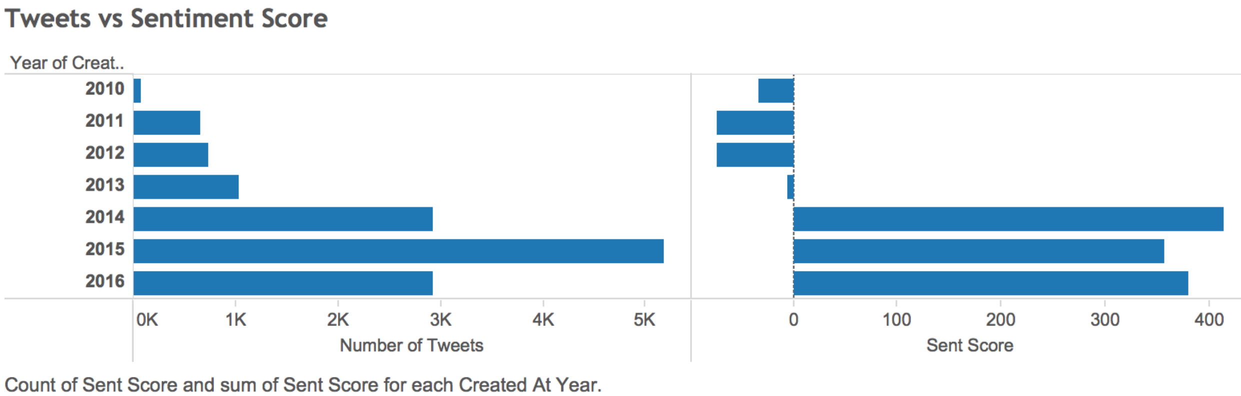


Figure 6: Tesla sentiment score vs. number of tweets per year

As shown in the figures above, the amount of tweets about Tesla skyrocketed after 2014 and has stayed relatively high ever since. It is no coincidence that the sentiment score associated with these Tesla-related tweets also skyrocketed in 2014. The yearly sentiment score from inception through 2013 was negative every single year, while it has been positive every year since. There is a clear association between the number of Tesla tweets and the sentiment scores: people are tweeting more about Tesla and have a more positive view towards Tesla.

A possible explanation for Tesla's skyrocket in 2014 coincides with the unveiling of Telsa's Model D for Dual Drive, meaning that the car operates using an engine on both axles. This gave future Tesla models the ability to accelerate at ludicrous speeds. For the Model D unveiled in 2014 (now called the Model S) it could reach 0-60mph in 3.2 seconds, which is unheard of for prior electric cars. This gave Tesla its 'sexy' image that it upholds today, being a car that is both efficient and fast. Another media boom in 2014 was the SpaceX CRS-3 (Dragon 9) mission to resupply the international space station. Elon Musk was the CEO of SpaceX at the time and likely received positive sentiment from tweets that mentioned both him and Tesla.

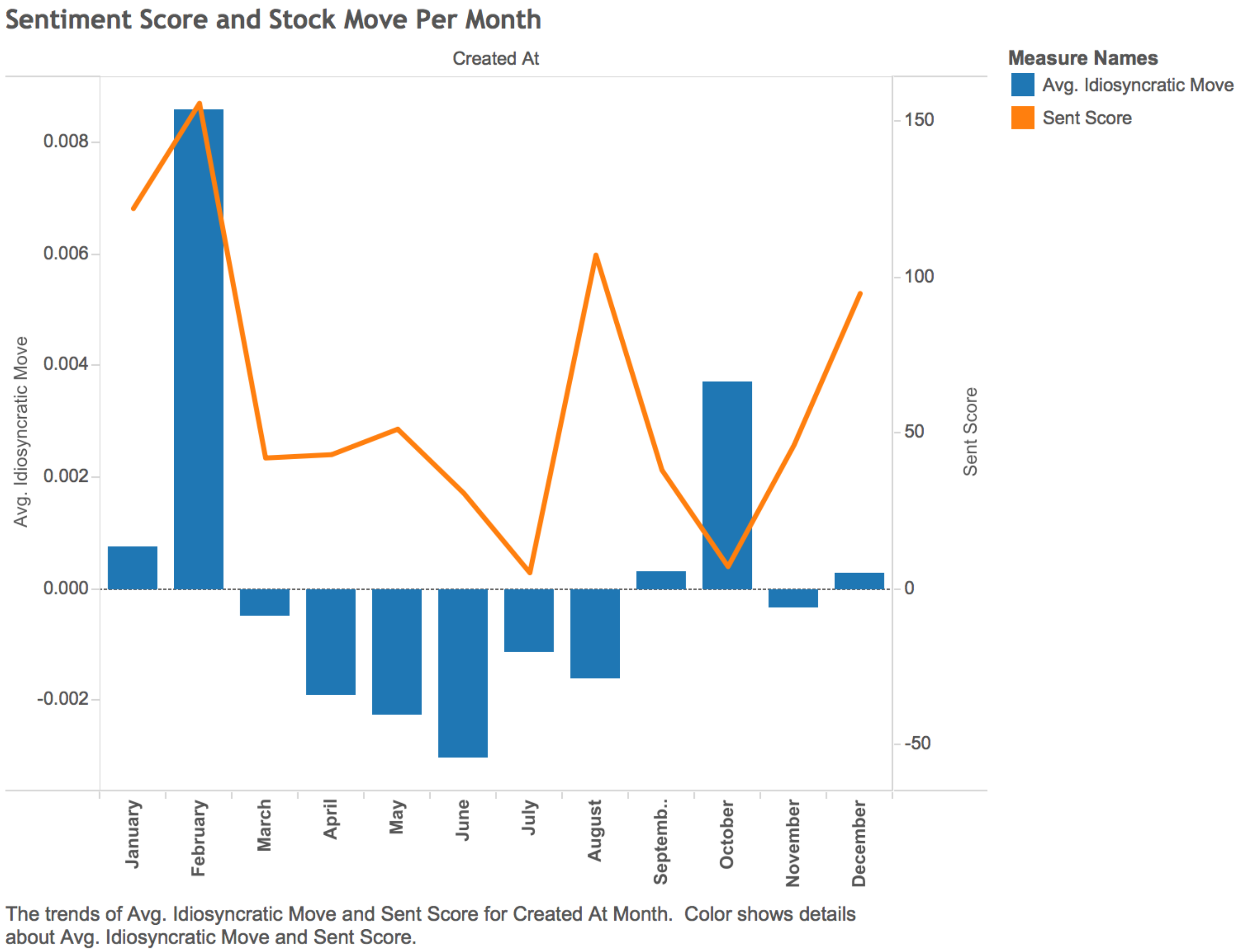
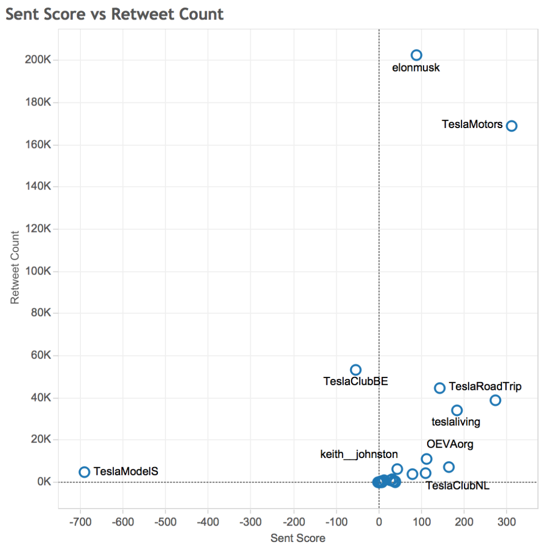


Figure 7: Tesla sentiment score and idiosyncratic move per month

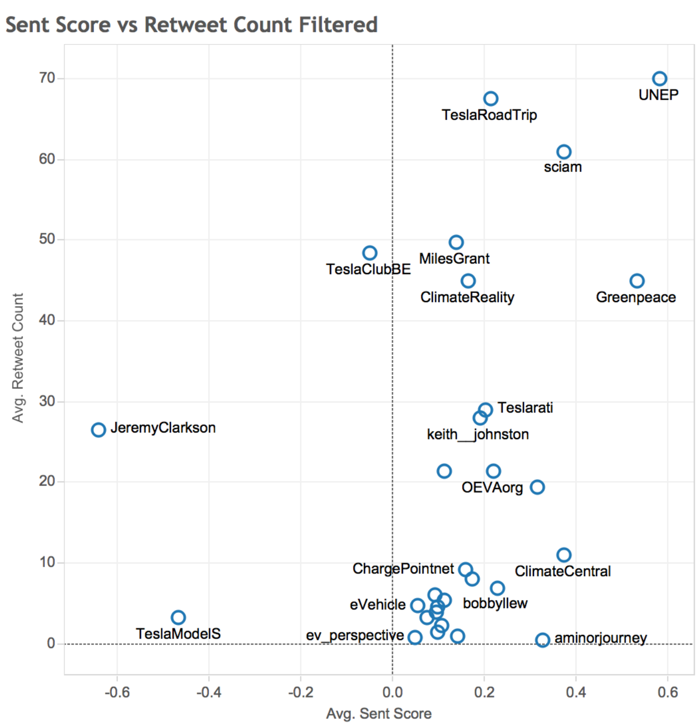
Average sentiment score follows a trend with local optimum at months where Tesla releases earnings reports. February, August, May, and November all have relatively high sentiment scores as these months coincide with earnings releases. Similarly, the months before these earnings releases appear to have the lowest sentiment scores. For example, July and October both come before an earnings release month and have the two lowest sentiment scores.

Furthermore, December and January also have two of the highest sentiment scores. During the holiday season, the public is most likely planning on buying new cars and tweeting about their highly anticipated purchase.



Sum of Sent Score vs. sum of Retweet Count. The marks are labeled by User Screen Name. Details are shown for User Screen Name.

Figure 8: Top influencers total sentiment score vs. total retweet count

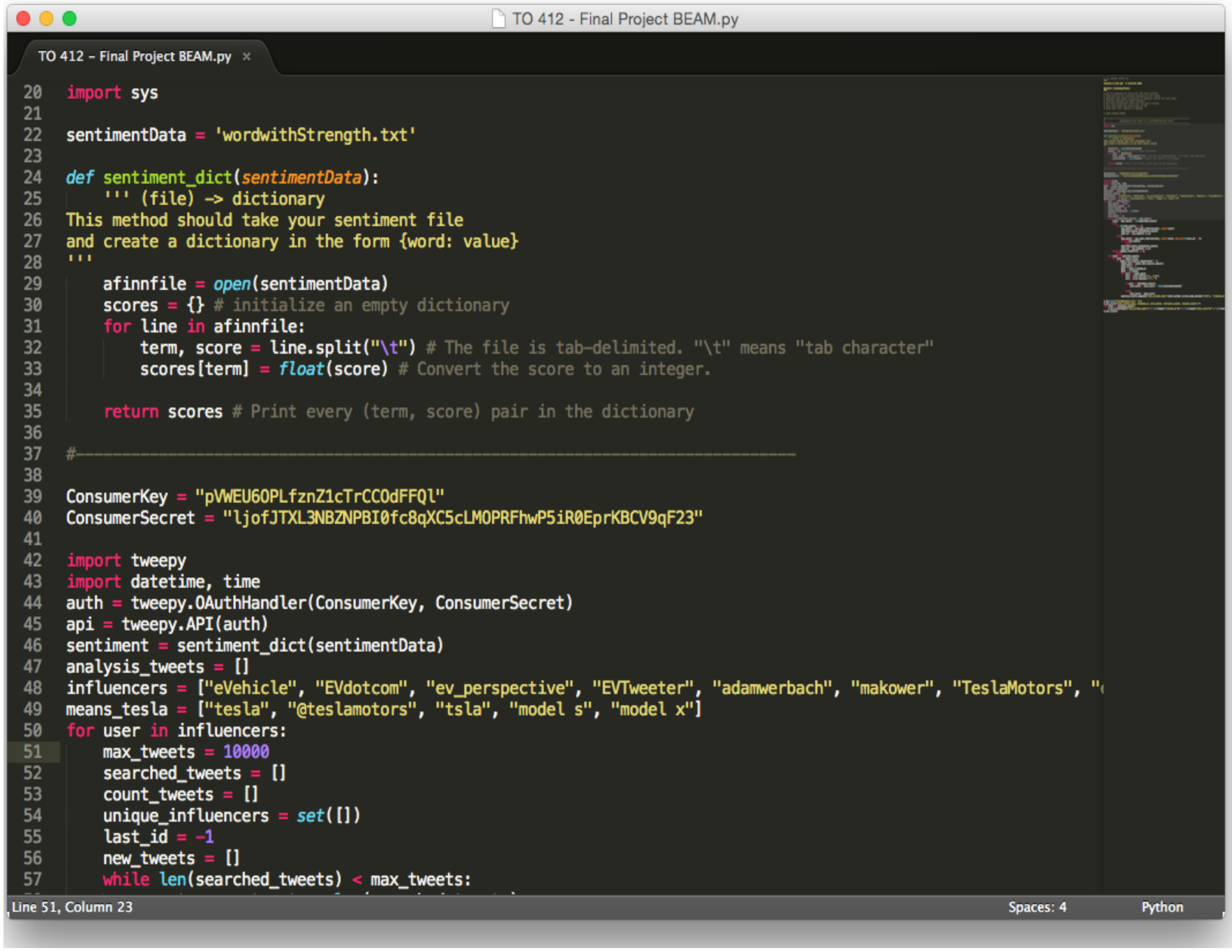


Average of Sent Score vs. average of Retweet Count. The marks are labeled by User Screen Name. Details are shown for User Screen Name. The view is filtered on User Screen Name, which excludes elonmusk and TeslaMotors.

Figure 9: Top influencers average sentiment score vs. average retweet count (filtered out outliers: @elonmusk and @TeslaMotors)

We measure virality of individual Twitter users by the total number of retweets they have. As shown in the graph below, Elon Musk (@elonmusk) and Tesla Motors (@TeslaMotors), are two outliers amongst the 31 Twitter users in our data set. They had a total of (NUMBER) and (NUMBER) retweets respectively. The result was not surprising as they are the easy figures to associate Tesla with as compared to users in the Electric Car Enthusiasts and the Environmental Enthusiasts categories.

We filtered out Elon Musk and Tesla Motors to further look into the virality of other users. We found that tweets from the Environmental Enthusiasts group, including users such as @UNEP, @Greenpeace, @MilesGrant, and @sciam, are generally higher in virality than the other groups. Moreover, their average sentiment scores towards Tesla are positive. Interestingly, users from the Electric Car Enthusiasts generally are low in virality. In particular, @JeremyClarkson gave Tesla the lowest sentiment score, (NUMBER). Clarkson has been critical of Tesla's reliability and performance in the past. Due to Clarkson's high number of followers and visibility on Twitter (NUMBER followers), Tesla unsuccessfully sued Clarkson's reviews (Halliday, 2013).



```
TO 412 - Final Project BEAM.py
20  import sys
21
22  sentimentData = 'wordwithStrength.txt'
23
24  def sentiment_dict(sentimentData):
25      ''' (file) -> dictionary
26  This method should take your sentiment file
27  and create a dictionary in the form {word: value}
28  '''
29      afinnfile = open(sentimentData)
30      scores = {} # initialize an empty dictionary
31      for line in afinnfile:
32          term, score = line.split("\t") # The file is tab-delimited. "\t" means "tab character"
33          scores[term] = float(score) # Convert the score to an integer.
34
35      return scores # Print every (term, score) pair in the dictionary
36
37  #------------------------------------------------------------------------------
38
39  ConsumerKey = "pVWEU6OPLfznZ1cTrCCOdFFQl"
40  ConsumerSecret = "ljofJTXL3NBZNPBI0fc8qXC5cLMOPRFhwP5iR0EprKBCV9qF23"
41
42  import tweepy
43  import datetime, time
44  auth = tweepy.OAuthHandler(ConsumerKey, ConsumerSecret)
45  api = tweepy.API(auth)
46  sentiment = sentiment_dict(sentimentData)
47  analysis_tweets = []
48  influencers = ["eVehicle", "EVdotcom", "ev_perspective", "EVTweeter", "adamwerbach", "makower", "TeslaMotors", "
49  means_tesla = ["tesla", "@teslamotors", "tsla", "model s", "model x"]
50  for user in influencers:
51      max_tweets = 10000
52      searched_tweets = []
53      count_tweets = []
54      unique_influencers = set([])
55      last_id = -1
56      new_tweets = []
57      while len(searched_tweets) < max_tweets:
Line 51, Column 23                                   Spaces: 4        Python
```

Figure 10: Python Code

Note that we made an effort to limit the keywords in the Tweets that would represent relevant tweets (line 49) and manually picked the influencers who met our criteria (line 48).

TO 412 - Final Project BEAM.py

```python
    while len(searched_tweets) < max_tweets:
        count = max_tweets - len(searched_tweets)
        try:
            if len(new_tweets) == 0:
                new_tweets = api.user_timeline(user, count=count)
                searched_tweets.extend(new_tweets)
                last_id = new_tweets[-1].id
            else:
                new_tweets = api.user_timeline(user, count=count, max_id=str(last_id - 1))
                if not new_tweets:
                    break
                searched_tweets.extend(new_tweets)
                last_id = new_tweets[-1].id
        except tweepy.TweepError as e:
            break
    for tweet in searched_tweets:
        for word in means_tesla:
            if tweet.text.lower().count(word) > 0:
                tweet_word = tweet.text.lower().split()
                sent_score = 0
                date = tweet.created_at
                date = str(date)
                for word in tweet_word:
                    word = word.rstrip('?:!.,;"!@')
                    word = word.replace("\n", "")

                    if word in sentiment.keys():
                        sent_score = sent_score + float(sentiment[word])

                    else:
                        sent_score = sent_score
                analysis_tweets.append({'user_screen_name':tweet.author.screen_name.encode('utf8'), 'created_at'

f_out=open("TeslaOutput2.csv", 'w')
f_out.write("user_screen_name, created_at, sent_score, followers_count, retweet_count\n")
for tweet in analysis_tweets:
    f_out.write(tweet['user_screen_name']+","+str(tweet['created_at'])+","+str(tweet['sent_score'])+","+str(twee
f_out.close()
```

Line 51, Column 23 Spaces: 4 Python

Figure 11: Python code (continued)

Regression Analysis (code):

```
title: "Tesla Regression"
author: "Pit Kauffmann"
date: "2016 M04 19"
output: html_document
---
```

```
```{r}
tesla <- read.csv("output.csv", header = TRUE, sep = ";")
head(tesla)
tesla <- tesla[,-c(1:3)]
tesla <- as.data.frame(sapply(tesla, as.numeric))

tesla_z <- as.data.frame(scale(tesla))

teslaregression <- lm(TSLAReturn ~ SentScore + SPReturn, data = tesla)
summary(teslaregression)
teslaregression_1 <- lm(TSLAReturn ~ SentScore, data = tesla_z)
summary(teslaregression_1)
teslaregression_2 <- lm(SentScore ~ TSLAReturn, data = tesla_z)
summary(teslaregression_2)

tesla <- read.csv("output.csv", header = TRUE, sep = ";")
head(tesla)
teslaClose <- aggregate(tesla$TSLAReturn, by=list(Category=tesla$Month), FUN=mean)
teslaSent <- aggregate(tesla$SentScore, by=list(Category=tesla$Month), FUN=mean)
tesla_1 <- cbind(teslaClose, teslaSent)
head(tesla_1)
tesla_1 <- tesla_1[,-3]
names(tesla_1) <- c("Month", "AVG.Price", "AVGSentScore")
teslaregression_3 <- lm(AVG.Price ~ AVGSentScore, data = tesla_1)
summary(teslaregression_3)
teslaregression_4 <- lm(AVGSentScore ~ AVG.Price, data = tesla_1)
summary(teslaregression_4)

```
```

Figure 12

In order to calculate regression statistics, we used R and its statistical package: the first block of code reads in the csv file, deletes 2 columns and converts all columns to numeric types. The second block runs 3 regression models on the read-in data, whereas the third block of code first groups the data by month and then calculates a regression based on the resulting reduced data table.

Regression Analysis (results):

```
> teslaregression_1 <- lm(TSLAReturn ~ SentScore, data = tesla_z)
> summary(teslaregression_1)

Call:
lm(formula = TSLAReturn ~ SentScore, data = tesla_z)

Residuals:
    Min      1Q  Median      3Q     Max
-6.5016 -0.5075  0.0164  0.5075  6.4896

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.905e-17  2.723e-02   0.000    1.000
SentScore   -9.387e-04  2.724e-02  -0.034    0.973

Residual standard error: 1 on 1348 degrees of freedom
Multiple R-squared:  8.811e-07, Adjusted R-squared:  -0.000741
F-statistic: 0.001188 on 1 and 1348 DF,  p-value: 0.9725
```

Figure 13

```
> teslaregression_2 <- lm(SentScore ~ TSLAReturn, data = tesla_z)
> summary(teslaregression_2)

Call:
lm(formula = SentScore ~ TSLAReturn, data = tesla_z)

Residuals:
    Min      1Q  Median      3Q     Max
-4.8009 -0.6519  0.0043  0.6592  4.6664

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.475e-15  2.723e-02   0.000    1.000
TSLAReturn  -9.387e-04  2.724e-02  -0.034    0.973

Residual standard error: 1 on 1348 degrees of freedom
Multiple R-squared:  8.811e-07, Adjusted R-squared:  -0.000741
F-statistic: 0.001188 on 1 and 1348 DF,  p-value: 0.9725
```

Figure 14:

```
> teslaregression_3 <- lm(AVG.Price ~ AVGSentScore, data = tesla_1)
> summary(teslaregression_3)

Call:
lm(formula = AVG.Price ~ AVGSentScore, data = tesla_1)

Residuals:
       Min         1Q     Median         3Q        Max 
-1.492e-03 -7.311e-04  3.261e-05  8.023e-04  1.363e-03 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)  
(Intercept)  -0.0006989  0.0003045  -2.295   0.0446 *
AVGSentScore -0.0032910  0.0034619  -0.951   0.3642  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.001055 on 10 degrees of freedom
Multiple R-squared:  0.08288,	Adjusted R-squared:  -0.008832 
F-statistic: 0.9037 on 1 and 10 DF,  p-value: 0.3642
```
Figure 15

```
> teslaregression_4 <- lm(AVGSentScore ~ AVG.Price, data = tesla_1)
> summary(teslaregression_4)

Call:
lm(formula = AVGSentScore ~ AVG.Price, data = tesla_1)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.14938 -0.04687 -0.01226  0.03620  0.15627 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.01725    0.03246  -0.531    0.607
AVG.Price   -25.18362   26.49144  -0.951    0.364

Residual standard error: 0.09226 on 10 degrees of freedom
Multiple R-squared:  0.08288,	Adjusted R-squared:  -0.008832 
F-statistic: 0.9037 on 1 and 10 DF,  p-value: 0.3642
```

Figure 16

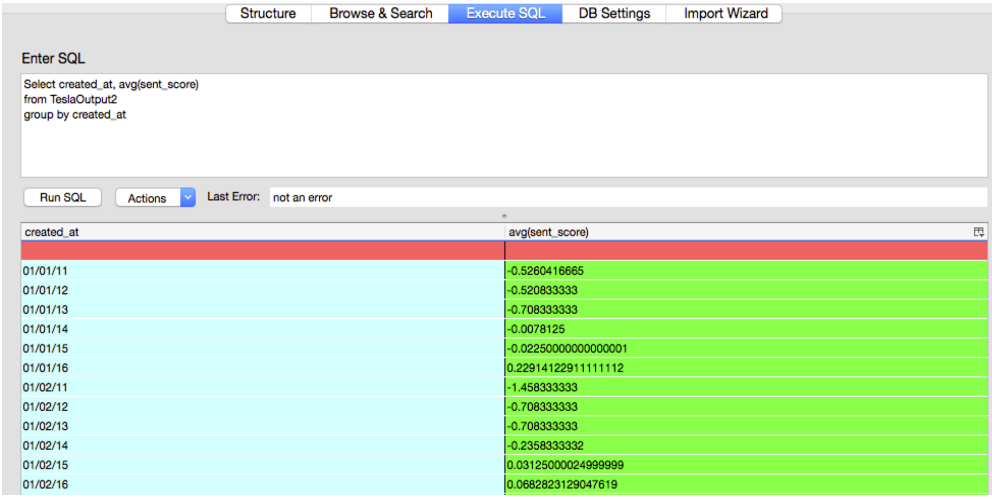
wordwithStrength.txt

```
epic	1
good	1
upright 1
fortunate	0.958333333
wonderfulness	0.875
wonderful	0.875
wide-eyed	0.875
wholesomeness	0.875
well-to-do	0.875
well-situated	0.875
well-heeled	0.875
well-fixed	0.875
vermifuge	0.875
utter	0.875
unpretentious	0.875
unmannerly	0.875
unmannered	0.875
understated	0.875
topping 0.875
top-hole	0.875
top-flight	0.875
therapeutic	0.875
sweetmeat	0.875
superb	0.875
successfulness	0.875
staring 0.875
spiffing	0.875
sodding 0.875
serendipity	0.875
self-respecting 0.875
```

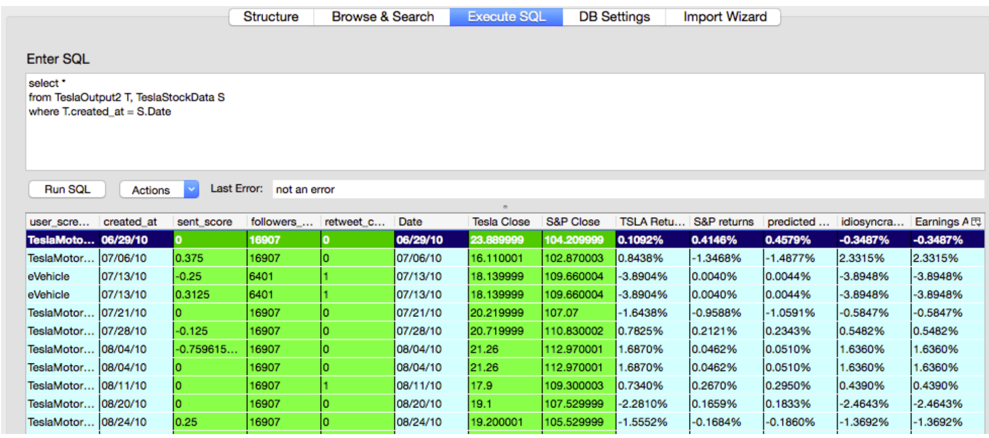
```
abominable	-0.875
incorrect	-0.9375
damaging	-0.9375
slanderous	-1
shit	-1
screw	-1
piteous -1
misfortunate	-1
libelous	-1
libellous	-1
hapless -1
denigratory	-1
denigrative	-1
denigrating	-1
```

Figure 17: Positive and negative sentiment scores

Source: https://github.com/hitesh915/sentimentstrength



created_at	avg(sent_score)
01/01/11	-0.5260416665
01/01/12	-0.520833333
01/01/13	-0.708333333
01/01/14	-0.0078125
01/01/15	-0.022500000000000001
01/01/16	0.22914122911111112
01/02/11	-1.458333333
01/02/12	-0.708333333
01/02/13	-0.708333333
01/02/14	-0.2358333332
01/02/15	0.03125000024999999
01/02/16	0.0682823129047619



user_scre…	created_at	sent_score	followers_…	retweet_c…	Date	Tesla Close	S&P Close	TSLA Retu…	S&P returns	predicted …	idiosyncra…	Earnings A
TeslaMoto…	06/29/10	0	16907	0	06/29/10	23.889999	104.209999	0.1092%	0.4146%	0.4579%	-0.3487%	-0.3487%
TeslaMotor…	07/06/10	0.375	16907	0	07/06/10	16.110001	102.870003	0.8438%	-1.3468%	-1.4877%	2.3315%	2.3315%
eVehicle	07/13/10	-0.25	6401	1	07/13/10	18.139999	109.660004	-3.8904%	0.0040%	0.0044%	-3.8948%	-3.8948%
eVehicle	07/13/10	0.3125	6401	1	07/13/10	18.139999	109.660004	-3.8904%	0.0040%	0.0044%	-3.8948%	-3.8948%
TeslaMotor…	07/21/10	0	16907	0	07/21/10	20.219999	107.07	-1.6438%	-0.9588%	-1.0591%	-0.5847%	-0.5847%
TeslaMotor…	07/28/10	-0.125	16907	0	07/28/10	20.719999	110.830002	0.7825%	0.2121%	0.2343%	0.5482%	0.5482%
TeslaMotor…	08/04/10	-0.759615…	16907	0	08/04/10	21.26	112.970001	1.6870%	0.0462%	0.0510%	1.6360%	1.6360%
TeslaMotor…	08/04/10	0	16907	0	08/04/10	21.26	112.970001	1.6870%	0.0462%	0.0510%	1.6360%	1.6360%
TeslaMotor…	08/11/10	0	16907	1	08/11/10	17.9	109.300003	0.7340%	0.2670%	0.2950%	0.4390%	0.4390%
TeslaMotor…	08/20/10	0	16907	0	08/20/10	19.1	107.529999	-2.2810%	0.1659%	0.1833%	-2.4643%	-2.4643%
TeslaMotor…	08/24/10	0.25	16907	0	08/24/10	19.200001	105.529999	-1.5552%	-0.1684%	-0.1860%	-1.3692%	-1.3692%

Figure 18: SQL JOIN Results

Further Analysis:

Our information was not perfect. Given more time and resources, we would consider incorporating the following into future analysis:

Time: To account for the stock market hours that open at 9am and close at 5pm, the tweets created after 5pm each day could be accounted on the next day. This will be a more realistic behavioral reflection if the stock is driven by sentiments on Twitter. There is also a time lag between the time of tweets created and the time stock owners react based on the tweet sentiments. Therefore, it could be accounted when we join two data files. A good estimation of the specific time lagged would require more research on the behavior of stock traders.

Sentiment Quantification: We chose our sentiment analysis package based on its large number of words, its values for each word, and its ability to translate sentiment into quantitative values. The package was also somewhat easy to implement; there were other packages with more complicated models for sentiment scoring that we could have used on GitHub or other places online. Some of these packages had better feedback from users and could improve the accuracy of our sentiment analysis.

Unrelated & Outlier Events: There were some events that affected Tesla's sentiment values that were unrelated to Tesla as a company and should not be reflected in its sentiment. Including: a significant event in Nikola Tesla's life, a concert played by the band called 'Tesla', a car accident that killed three people in which a Tesla vehicle rear ended another another car, and a Tesla World 2015 event where Elon Musk rudely arrived three hours late. If we were able to remove these outliers from our data set, it would give us a better depiction of Tesla's true sentiment.