SENTIMENT ANALYSIS FOR INVESTMENT RESEARCH MODELS

This data analysis report targets professionals within the financial investment industry who are looking to incorporate non-traditional data, also referred to as “alternative data”, into financial investment models.

Course: INFO 640-01, Data Analysis
Assignment: Final Project

ERIK HANNELL
M.Sc. Data Analytics & Visualization

INTRODUCTION

Big data has become one of the most talked-about fields over the last couple of decades. Whether a topic relates to finance, sustainability, politics, or any other area, there is practically always a way to implement big data as an interesting addition to the conversation. The possibility to perform analysis by studying comprehensive data sets, as opposed to relying on traditional methods, such as sample studies, heavily enhances the accuracy of analysis models.

Unstructured, human-written, text is one of the many origins of big data. In today's society, people are expressing their thoughts, feelings, and opinions through various social media channels. Twitter is undeniably one of the most popular platforms for public online conversations, not the least when it comes to global leaders. Considering the massive amount of unstructured data currently available, and all the new data constantly being created, there is a lot of untapped value ready to be extracted. One of the most common methods for capturing value from unstructured text is by performing sentiment analysis in order to make sense of the underlying feelings expressed in the text. Although sentiment analysis technology is not yet fully reliable, it can still be used for several purposes, investment research is one of them.

This data analysis project is an extension of previous research in the field of using sentiment analysis as a tool within investment research. Mittal and Goel (2011) found that they could successfully predict stock market fluctuations by observing public moods via Twitter sentiment analysis. The study yielded statistically significant results when certain sentiment moods were correlation tested against stock market fluctuations. The current quality of Mittal and Goel's findings, as of December 2019, is subject to discussion, considering that the fast-paced technological environment could have altered the outcome were a similar study carried out today. However, the assumption for this project is that their conclusions are still statistically significant by the time of writing this report, in December 2019. Important to note is that, even if sentiment analysis could not be used to predict stock price fluctuations, the method would still be highly relevant in order to gain additional insight into the assessment of potential investments, or other subjects of analysis.

The project culminates in an exploratory data analysis model which attempts to determine the recent and current public perception of an organization, based on Tweets addressed to that organization. The model would optimally be incorporated as a building block of a more comprehensive data analysis model, in order to make as informed decisions as possible. Such an extensive model could be a predictive analysis for evaluating popularity rates for politicians. However, the main purpose of the sentiment model in this project is to be built into investment research models.

METHOD

All the code for this project was built using the statistical computing programming language R, in the R-Studio environment. Several library tools were applied in order to generate the end result, the most important being; the twitteR library, to feed Twitter data directly into the code, the NRC library, which was used to carry out the actual sentiment analysis, and the ggplot2 library for creating visualizations.

Tesla was selected as the test subject for this particular analysis, however, the test subject is not the main focus of this project. Attention should rather be directed to the actual sentiment analysis model as a tool for investment management. The reason for selecting Tesla was mainly because the company had just introduced the latest edition to its car model catalog by the start of this project. The unveiling of Tesla's new venture, the Cybertruck, received mixed opinions, and immediately became a hot topic on social media. Hence, the company made a good fit for the purpose of this project.

Following the picking of Tesla as the analysis subject, code production was initiated. The twitteR-library was used to collect Tweets that held the word "Tesla", and other parameters were set to only gather the most recent Tweets, dating back one day at its maximum. A criterion to disregard retweets was set, in order to avoid duplicate texts. Various search queries were tested, such as "Tesla"+"Elon Musk", which would return Tweets with both the words "Tesla" and "Elon Musk", or either one in the text. However, the decision was eventually made to proceed with "Tesla" as the only search term, as it returned a sufficient amount of Tweet and would maintain a unanimous focus on the

company. It should be mentioned that in order to request Twitter data using the twitteR-library, users are required to insert API credentials. These may be provided by registering for a Twitter Developer account.

Once the Tweets were downloaded, they were transformed into a data frame. In this data frame, Twitter information is spread out across 17 variables, ranging from user data to time of posting a Tweet. From this data frame, a corpus holding only the Tweets was created. Unnecessary text information can disrupt the quality of the sentiment analysis outcome. Hence, text cleaning was applied to the corpus, such as the removal of punctuations and other special characters, as well as the removal of particular words that are assessed to be unnecessary. Upon the completion of data cleaning and manipulation, the main segment of the code scripting was initiated, namely sentiment analysis using the NRC-library. The NRC-library comes with a large dictionary, with word categorizations according to ten different feelings; positive, joy, anticipation, trust, surprise, negative, anger, disgust, sadness, and fear. This method is defined as “bag of words”, i.e. classification of individual words, meaning that the underlying message of the text might be disregarded. In order to attempt to capture the actual message of the text, one would have to apply natural language processing techniques. This part of the code could, therefore, be further developed in future versions of this project, in order to potentially improve the efficiency of the model. The bag of words method was used because that method is easier to carry out and implement, while an NLP approach would simply extend beyond the scope of this project, as this is more of a machine learning method.

After conducting the analysis and assigning sentiment scores for all Tweets, the corpus was transformed back into a data frame and merged with the original data frame. This resulted in a Twitter data set with ten new columns, a total of 27 variables. Three new columns were added; one holding the cumulative score for all positive feelings per Tweet, one column for storing the total negative score per Tweet, and one column holding the absolute total score. For the positive total, four feelings were cumulated; *positive*, *joy*, *anticipation*, and *trust*. The negative total was assigned five feelings; *negative*, *fear*, *disgust*, *sadness,* and *anger*. The feeling *surprise* was left out of the

totals as it can be both positive or negative, or even neutral. Finally, sentiment ratios were created, by dividing each feeling score by the total score.

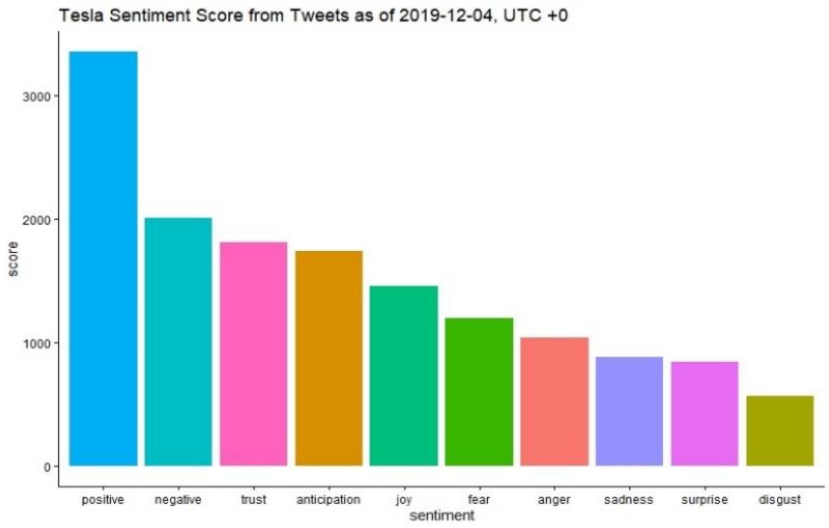
The last stage of data manipulation, before moving on to the exploratory data analysis, was to aggregate the data per hour. This time period could be altered, either being extended or minimized, depending on desired data granularity. Shorter time periods, such as by the minute, would increase the possibility of measuring public opinion more often. However, the hourly frequency was selected as this was optimal for visualizing the data in this project.

RESULTS

The following visualizations were built in order to demonstrate the data that was generated from running the code. The purpose of the first two visualizations is solely to showcase the data to the observer, while the last visualization of the positivity ratio is the main component in terms of investment model integration.

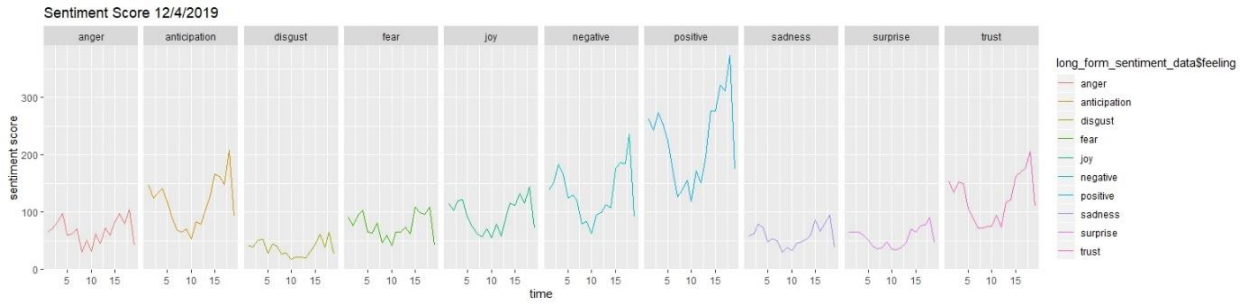
Visualization 1: Sentiment Ranking

This is a ranked bar chart that shows which of the ten feelings from the NRC-library were the most associated with Tesla on the 4th of December 2019. It is clear that the single most associated feeling was positivity, and the least associated was disgust. This visualization provides a simple and clear overview of recent Twitter opinions.



Output 2: Faceted Sentiment Graph

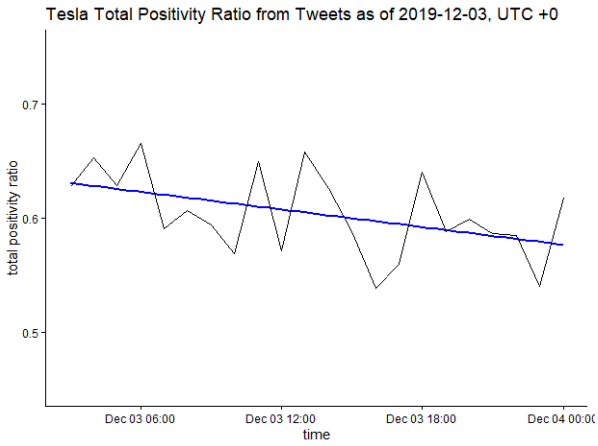
The second graph is faceted by each of the ten NRC feelings. This graph is not automatically generated by the code, as its main purpose is to show the underlying data for the sake of the report. However, the graph can be reproduced by transforming the Twitter data frame from a wide to long format. This can be done rather easily by using Google's open-source data tool OpenRefine.

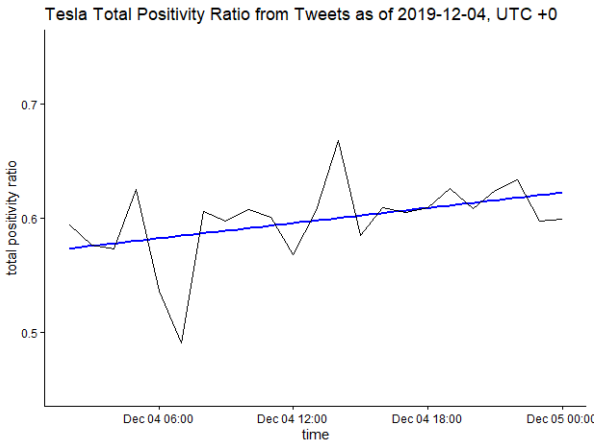


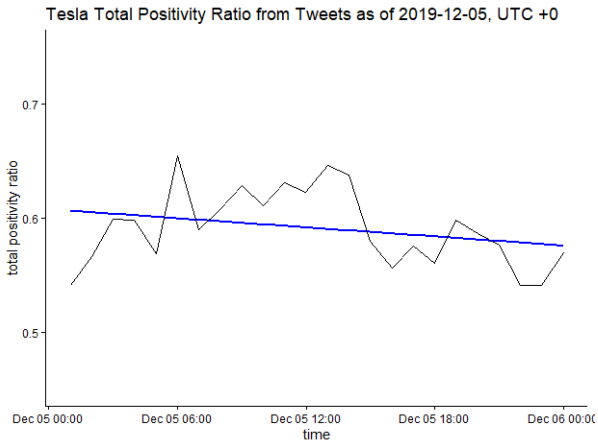
Output 3: Positivity Ratio

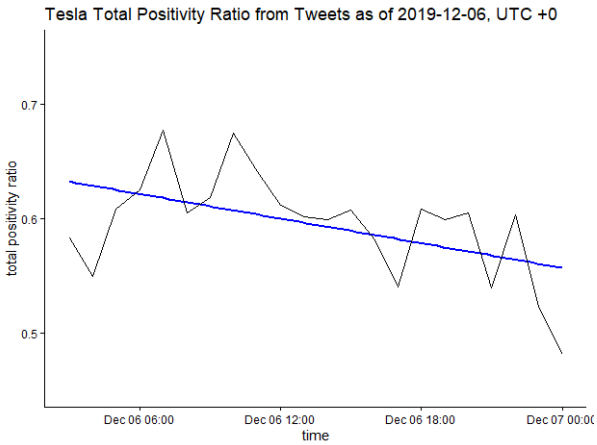
The positivity ratio is the most important part of the code output, as this is what would eventually drive the insight generation for the investment model. The graphs below demonstrate how the positivity ratio of Tesla developed over time, with data collected for each day between December 3 and December 10. It allows analysts to make insightful takeaways, by studying how the perception on Twitter fluctuates over time. One would assume that these highs and lows are related to news publications or Tweets from Tesla's CEO Elon Musk, but they could also be caused by less obvious reasons.

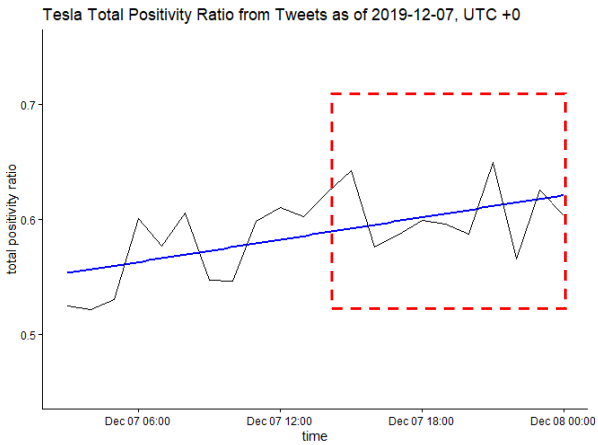
By observing the different graphs, we can search for interesting patterns. For example, we see a peak in positivity around 11 AM – 1 PM on December 9. This is approximately the time when a news story came out stating that Elon Musk had driven into a road sign with his new Tesla Cybertruck. However, one would assume that the positivity ratio would dip after such a news article, but instead, we saw an incline. Considering the sentiment fluctuations above, potentially caused by an Elon Musk event, it is clear that sentiment movements can be rather unpredictive. Hence, it is of great importance to carefully study the data prior to making decisions. It would have been interesting to study the stock price movement of 11 AM – 1 PM on December 9, but unfortunately, the NASDAQ, where Tesla is traded, was closed at the time.

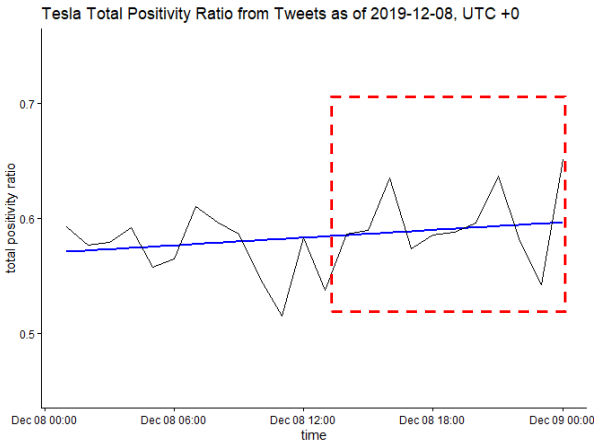

Tesla Total Positivity Ratio from Tweets as of 2019-12-03, UTC +0
total positivity ratio
0.7
0.6
0.5
Dec 03 06:00
Dec 03 12:00
Dec 03 18:00
Dec 04 00:0
time

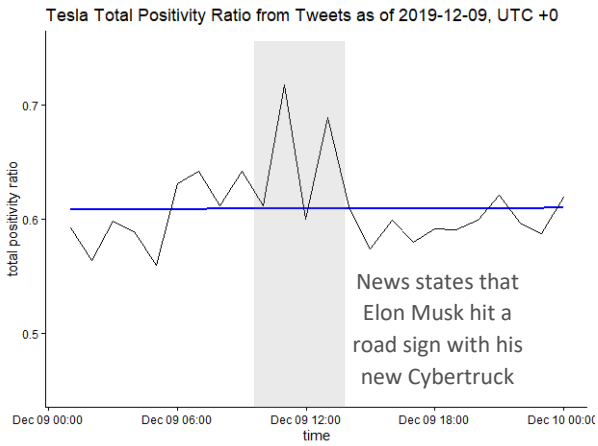

Tesla Total Positivity Ratio from Tweets as of 2019-12-04, UTC +0
total positivity ratio
0.7
0.6
0.5
Dec 04 06:00
Dec 04 12:00
Dec 04 18:00
Dec 05 00:0
time

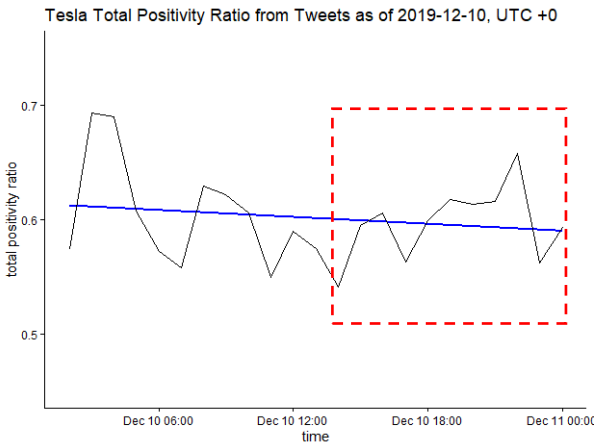

Tesla Total Positivity Ratio from Tweets as of 2019-12-05, UTC +0
total positivity ratio
0.7
0.6
0.5
Dec 05 00:00
Dec 05 06:00
Dec 05 12:00
Dec 05 18:00
Dec 06 00:0
time


Tesla Total Positivity Ratio from Tweets as of 2019-12-06, UTC +0
total positivity ratio
0.7
0.6
0.5
Dec 06 06:00
Dec 06 12:00
Dec 06 18:00
Dec 07 00:0
time


Tesla Total Positivity Ratio from Tweets as of 2019-12-07, UTC +0
total positivity ratio
0.7
0.6
0.5
Dec 07 06:00
Dec 07 12:00
Dec 07 18:00
Dec 08 00:00
time


Tesla Total Positivity Ratio from Tweets as of 2019-12-08, UTC +0
total positivity ratio
0.7
0.6
0.5
Dec 08 00:00
Dec 08 06:00
Dec 08 12:00
Dec 08 18:00
Dec 09 00:00
time


Tesla Total Positivity Ratio from Tweets as of 2019-12-09, UTC +0
total positivity ratio
0.7
0.6
0.5
Dec 09 00:00
Dec 09 06:00
Dec 09 12:00
Dec 09 18:00
Dec 10 00:0
time
News states that Elon Musk hit a road sign with his new Cybertruck


Tesla Total Positivity Ratio from Tweets as of 2019-12-10, UTC +0
total positivity ratio
0.7
0.6
0.5
Dec 10 06:00
Dec 10 12:00
Dec 10 18:00
Dec 11 00:0
time

The average positivity ratio appears to be slightly below 0.6, which can be concluded by studying the positions of the blue trend lines. This is a significant finding since such a figure is necessary for the process of building the decision-making algorithm. For example, if the average number would eventually be statistically set to 0.6, which would require far more observations and calculations, a signal could be sent out for each time the actual positivity ratio exceeds the average and a threshold margin, such as (0.6 + 20%) would be the threshold for a positive signal, while (0.6 - 20%) would be the threshold for the contradiction. Of course, it would have to be carefully investigated whether (average + 20%) is a legitimate threshold for sending out decision recommendations, for example by correlation testing against stock price.

Furthermore, it appears as if the positivity ratio follows similar paths for some of the days. For example, when observing the graph parts within the dashed red box, you will find intentions of reoccurring patterns. The graph starts with an incline, followed by a decline, then a slight upward trend, followed by a rather significant peak, then again, another steep decline before eventually starting to trend upwards. It is important to take note of these reoccurring patterns and incorporate them into the decision-making algorithm. Analysts should search for abnormalities from these reoccurring patterns. Unusual patterns should then be tested, for example, against stock price, to investigate whether certain abnormalities can trigger particular stock price movements.

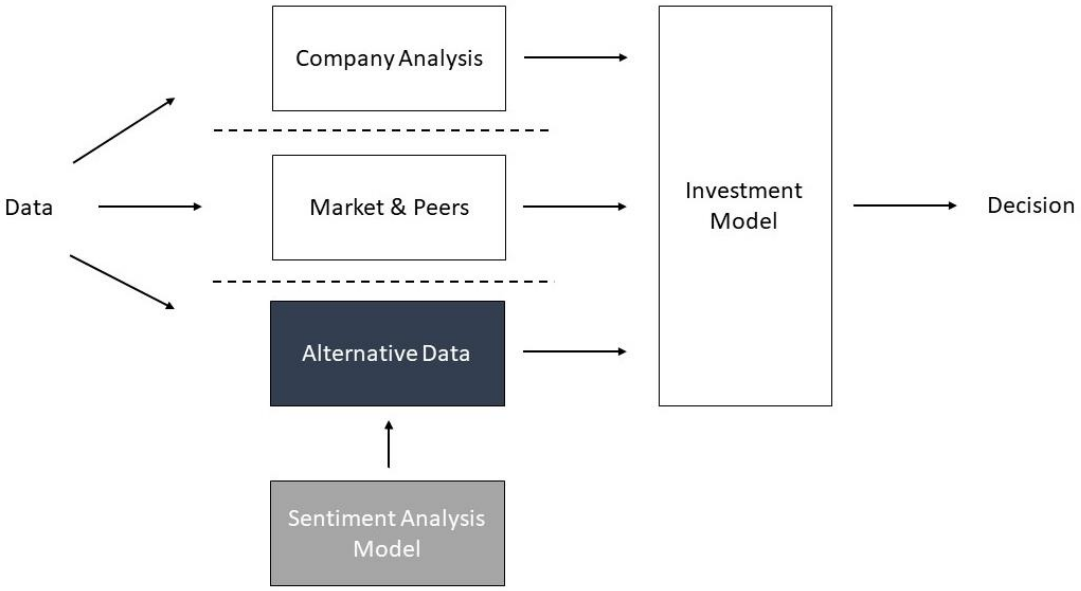
INVESTMENT MODEL INCORPORATION

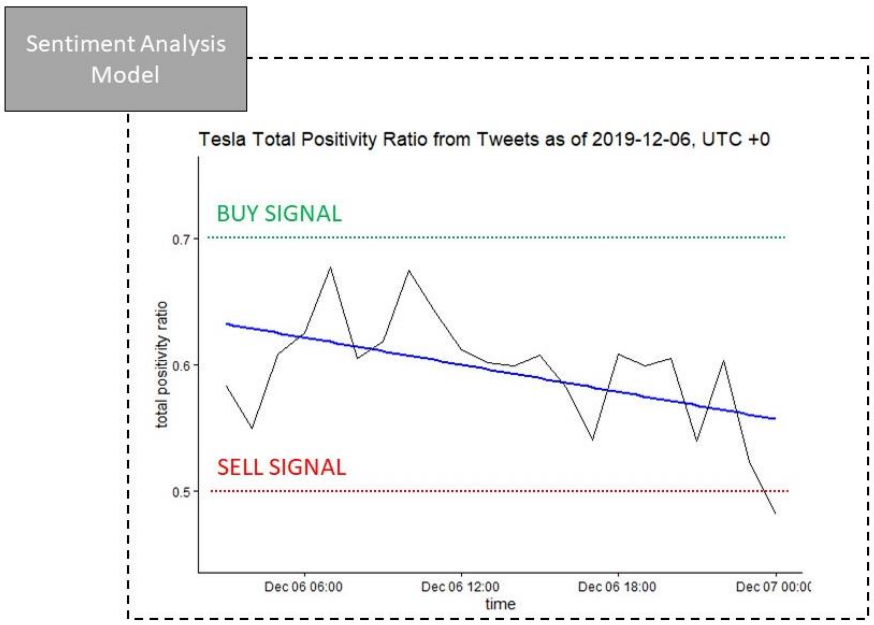
Considering that stock market fluctuations adhere to a wide range of information sources and several other factors, it would not be optimal to run a sentiment analysis program as a standalone investment model. The sentiment analysis should rather function as part of the “alternative data” fraction of an extensive investment model, taking a wide range of information sources into account, to try and catch as many signals as possible.

The best practice for incorporating this particular sentiment analysis project as part of a more extensive investment research model would ultimately be to apply machine learning in order to search for potential predictable patterns. These patterns were

discussed earlier as abnormalities from common movements. If reoccurring sentiment patterns were detected prior to bearish or bullish stock fluctuations, it would allow the investment model to respond to the information, weighing in on decisions to either increase or decrease a long or short position.

The visualization below is a mock-up of how the sentiment model could be designed using the positivity ratio. The buy and sell signals are determined by assuming a 0.6 average ratio +/- 17% (0.1).





CONCLUSION

This project was an extension of previous research within the field, which had successfully proved a statistically significant correlation between the sentiment of Tweets and stock market fluctuations. The report showcased a sentiment analysis model aimed to be incorporated into extensive investment research models, in order to feed public opinions into decision-making algorithms. Public opinion data was collected from Twitter and stored as a data set in R Studio, where all the code scripting was carried out. Sentiment analysis was applied using the NRC-library package. Finally, a positivity ratio was calculated by dividing positive feelings with negative feelings. This ratio was used to analyze the opinion of the research object, in this case, the electric car manufacturer Tesla.

Some interesting patterns were detected by running the sentiment model on Tesla. Eight days' worth of sentiment data was collected, and it appeared as if some days followed a similar pattern. Furthermore, we were able to identify an average ratio of around 0.6 for the positivity measure. These findings are all interesting for eventually constructing a decision recommendation algorithm.

It would be interesting to further investigate the reasons for the patterns that could be found in the positivity ratios which were previously discussed. If similar patterns were identified when performing the same analysis on another company, other than Tesla, we could potentially assume that the patterns are related to macro behavior. Additionally, the next step for this project would be to improve the code, for example by adding NLP-methods as discussed earlier, but also to eventually test different positivity ratio movements against stock price, and finally, given successful pattern recognitions, integrating the code into a larger investment model.

REFERENCES

Mittal, A. & Goel, A. (2011). Stock Prediction Using Twitter Sentiment Analysis. Stanford.

APPENDIX (CODE)

```
#FINAL PROJECT - Twitter sentiment analysis

install.packages("xts")

#setup packages
library(twitteR)
library(dplyr)
library(lubridate)
library(tidytext)
library(tidyverse)
library(stringr)
library(textdata)
library(RColorBrewer)
library(tm)
library(ROAuth)
library(plyr)
library(quantmod)
library(ggplot2)
library(syuzhet)
library(lmtest)
library(ggplot2)
library(xts)

#_____________________SCRAPTE TWITTER, CREATE DATASET_____________________

#API keys
api_key = "[INSERT KEY]"
api_secret = "[INSERT KEY]"
access_token = "[INSERT KEY]"
access_token_secret = "[INSERT KEY]"

#_____________twitter scraping_____________________
setup_twitter_oauth(api_key, api_secret, access_token, access_token_secret)

#get today's date
date = Sys.Date()
date <- toString(date)

#INSERT DESIRED DATES HERE, AND SET AS SINCE & UNTIL VARIABLES:
since_date = "2019-12-04" #since =
until_date = "2019-12-05" #until =

#tweet search query
#CHANGE "Tesla" TO DESIRED ANALYSIS SUBJECT:
tweets_tesla = searchTwitter("\"Tesla\"-filter:retweets", since = date, n=10000,
lang="en")

#_____________________________________________________________________________

#convert to dataset
tweets_tesla_df <- twListToDF(tweets_tesla)

#manipulate
tweets_tesla_df <- data.frame(doc_id=index(tweets_tesla_df),
coredata(tweets_tesla_df))
tweets_tesla_df$doc_id <- as.character(tweets_tesla_df$doc_id)

#inspect
glimpse(tweets_tesla_df)
head(tweets_tesla_df["text"])

#set min and max time stamps of collected tweets (to match with potential stock
data)
min_time <- min(tweets_tesla_df$created)
```

```
max_time <- max(tweets_tesla_df$created)

#__________________DATA CLEANING AND MANIPULATION_____________________

tweets_tesla_text_df <- tweets_tesla_df["text"]
tweets_tesla_text_df

#create corpus
tweets_tesla_corpus <- Corpus(VectorSource(tweets_tesla_text_df$text))
tweets_tesla_corpus
#inspect pre-cleaning
inspect(tweets_tesla_corpus[1:10])

#text cleaning functions
  tweet.removeURL = function(x) gsub("http[^[:space:]]*","",x)
  tweet.removeATuser = function(x) gsub("@[a-z,A-Z]*", "", x)
  tweet.removeEmoji = function(x) gsub("\\p{So}|\\p{Cn}", "", x, perl=TRUE)
  tweet.removeSpecialChar = function(x) gsub("[^[:alnum:]///' ]", "", x)

#clean the corpus
tweets_tesla_corpus = tm_map(tweets_tesla_corpus,
content_transformer(tweet.removeURL))
tweets_tesla_corpus = tm_map(tweets_tesla_corpus,
content_transformer(tweet.removeATuser))
tweets_tesla_corpus = tm_map(tweets_tesla_corpus,
content_transformer(tweet.removeSpecialChar))
tweets_tesla_corpus = tm_map(tweets_tesla_corpus, content_transformer(tolower))
tweets_tesla_corpus = tm_map(tweets_tesla_corpus, removeNumbers)
tweets_tesla_corpus = tm_map(tweets_tesla_corpus, removePunctuation)
#tweets_tesla_corpus = tm_map(tweets_tesla_corpus, stemDocument)

#remove stopwords
tweets_tesla_corpus = tm_map(tweets_tesla_corpus, removeWords,
                             c(stopwords("english"), "model", "cars",
                               "analysis", "cyber", "id", "cybertruck",
                               "cybertrucks", "truck", "RT", "rt", "teslas",
                               "tesla", "stock"))
#strip whitespace
tweets_tesla_corpus = tm_map(tweets_tesla_corpus, stripWhitespace)

#inspect post-cleaning
inspect(tweets_tesla_corpus[1:10])

#transform corpus to DF
tweets_tesla_corpus_df <- data.frame(text = sapply(tweets_tesla_corpus,
as.character), stringsAsFactors = FALSE)
tweets_tesla_corpus_df <- data.frame(ID=index(tweets_tesla_corpus_df),
coredata(tweets_tesla_corpus_df))

#___________________________DOCUMENT TERM MATRIX (might expand)__________________

#DocumentTermMatrix
tesla_dtm <- DocumentTermMatrix(tweets_tesla_corpus)
tesla_dtm

tesla_dtm_tidy <- tidy(tesla_dtm)
tesla_dtm_tidy


#_____________________________SENTIMENT ANALYSIS_____________________________

get_sentiments("nrc")
tesla_sentiment <- get_nrc_sentiment(tweets_tesla_corpus_df$text)

#generate ID-column to tesla_sentiment
```

```
tesla_sentiment_df <- data.frame(doc_id=index(tesla_sentiment),
coredata(tesla_sentiment))

tesla_sentiment_scores <- data.frame(colSums(tesla_sentiment[,]))
names(tesla_sentiment_scores) <- "score"
tesla_sentiment_scores <- cbind("sentiment" = rownames(tesla_sentiment_scores),
                                tesla_sentiment_scores)
rownames(tesla_sentiment_scores) <- NULL

my_theme <-  theme(panel.background = element_blank(),
                   plot.background = element_blank(),
                   legend.background = element_blank(),
                   legend.key = element_blank(),
                   strip.background = element_blank(),
                   axis.text = element_text(colour="black"),
                   axis.ticks = element_line(colour="black"),
                   panel.grid.major = element_blank(),
                   panel.grid.minor = element_blank(),
                   axis.line = element_line(colour = "black"),
                   strip.text = element_blank(),
                   panel.spacing = unit(1, "lines"))

#variable for ggtitle, to combine with today's date
title = "Tesla Sentiment Score from Tweets as of "
timezone = ", UTC +0"

#visualize the sentiment frequency
ggplot(data = tesla_sentiment_scores, aes(x=reorder(sentiment, -score), y=score))+
  geom_bar(aes(fill = sentiment), stat="identity")+
  theme(legend.position = "none")+
  xlab("sentiment") + ylab("score")+
  my_theme+
  ggtitle(paste0(title, date, timezone))


#_________________JOIN CORPUS BACK WITH CREATED (DATE)_____________________ ___
#This will allow me to tell when the tweets were made

#join DF's to get score per tweet
tweets_tesla_sentiment_dates <- merge(tweets_tesla_df, tesla_sentiment_df,
by="doc_id")

#total negativity and total positivity columns
tweets_tesla_sentiment_dates$sum_positivity <-
rowSums(tweets_tesla_sentiment_dates[,c("positive", "joy", "trust",
"anticipation")])
tweets_tesla_sentiment_dates$sum_negativity <-
rowSums(tweets_tesla_sentiment_dates[,c("negative", "sadness", "fear", "disgust",
"anger")])

#_____________________AGGREGATE ALL MOODS BY HOUR____________________________

glimpse(tweets_tesla_sentiment_dates)

#sum_positivity
dat.xts.pos <- xts(tweets_tesla_sentiment_dates$sum_positivity,
               as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_sum_positivity <- period.apply(dat.xts.pos, endpoints(dat.xts.pos, "hours"),
sum) #sum to hours
hourly_sum_positivity <- data.frame(time=index(hourly_sum_positivity),
coredata(hourly_sum_positivity)) #create "time"
hourly_sum_positivity <- data.frame(id=index(hourly_sum_positivity),
coredata(hourly_sum_positivity)) #create "id"

#sum_negativity
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$sum_negativity,
```

```
               as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_sum_negativity<- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"),
sum) #sum to hours
hourly_sum_negativity <- data.frame(time=index(hourly_sum_negativity),
coredata(hourly_sum_negativity)) #create "time"
hourly_sum_negativity <- data.frame(id=index(hourly_sum_negativity),
coredata(hourly_sum_negativity)) #create "id"

#positive
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$positive,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_positive <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"), sum)
#sum to hours
hourly_positive <- data.frame(time=index(hourly_positive),
coredata(hourly_positive)) #create "time"
hourly_positive <- data.frame(id=index(hourly_positive), coredata(hourly_positive))
#create "id"

#negative
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$negative,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_negative <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"), sum)
#sum to hours
hourly_negative <- data.frame(time=index(hourly_negative),
coredata(hourly_negative)) #create "time"
hourly_negative <- data.frame(id=index(hourly_negative), coredata(hourly_negative))
#create "id"

#trust
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$trust,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_trust <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"), sum)
#sum to hours
hourly_trust <- data.frame(time=index(hourly_trust), coredata(hourly_trust))
#create "time"
hourly_trust <- data.frame(id=index(hourly_trust), coredata(hourly_trust)) #create
"id"


#anticipation
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$anticipation,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_anticipation <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"),
sum) #sum to hours
hourly_anticipation <- data.frame(time=index(hourly_anticipation),
coredata(hourly_anticipation)) #create "time"
hourly_anticipation <- data.frame(id=index(hourly_anticipation),
coredata(hourly_anticipation)) #create "id"


#joy
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$joy,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_joy <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"), sum) #sum
to hours
hourly_joy <- data.frame(time=index(hourly_joy), coredata(hourly_joy)) #create
"time"
hourly_joy <- data.frame(id=index(hourly_joy), coredata(hourly_joy)) #create "id"


#fear
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$fear,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_fear <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"), sum) #sum
to hours
```

```
hourly_fear <- data.frame(time=index(hourly_fear), coredata(hourly_fear)) #create
"time"
hourly_fear <- data.frame(id=index(hourly_fear), coredata(hourly_fear)) #create
"id"


#anger
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$anger,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_anger <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"), sum)
#sum to hours
hourly_anger <- data.frame(time=index(hourly_anger), coredata(hourly_anger))
#create "time"
hourly_anger <- data.frame(id=index(hourly_anger), coredata(hourly_anger)) #create
"id"


#sadness
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$sadness,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_sadness <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"), sum)
#sum to hours
hourly_sadness <- data.frame(time=index(hourly_sadness), coredata(hourly_sadness))
#create "time"
hourly_sadness <- data.frame(id=index(hourly_sadness), coredata(hourly_sadness))
#create "id"


#surprise
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$surprise,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_surprise <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"), sum)
#sum to hours
hourly_surprise <- data.frame(time=index(hourly_surprise),
coredata(hourly_surprise)) #create "time"
hourly_surprise <- data.frame(id=index(hourly_surprise), coredata(hourly_surprise))
#create "id"


#disgust
dat.xts.neg <- xts(tweets_tesla_sentiment_dates$disgust,
                   as.POSIXct(tweets_tesla_sentiment_dates$created))
hourly_disgust <- period.apply(dat.xts.neg, endpoints(dat.xts.neg, "hours"), sum)
#sum to hours
hourly_disgust <- data.frame(time=index(hourly_disgust), coredata(hourly_disgust))
#create "time"
hourly_disgust <- data.frame(id=index(hourly_disgust), coredata(hourly_disgust))
#create "id"


#_____________________Visualizing the feelings over time_________________

#sum_positivity over time
ggplot(hourly_sum_positivity, aes(x=hourly_sum_positivity$time,
y=hourly_sum_positivity$coredata.hourly_sum_positivity.))+
  my_theme+
  xlab("time") + ylab("total positivity score")+
  ggtitle(paste0(title, date, timezone))+
  geom_line()

#sum_negativity over time
ggplot(hourly_sum_negativity, aes(x=hourly_sum_negativity$time,
y=hourly_sum_negativity$coredata.hourly_sum_negativity.))+
  my_theme+
  xlab("time") + ylab("total negativity score")+
  ggtitle(paste0(title, date, timezone))+
```

```
  geom_line()

#MERGE ALL MOODS OVER TIME
#merged sum_negativity and sum_positivity and created a positivity score
hourly_data_gathered <- merge(hourly_sum_negativity, hourly_sum_positivity,
by="time")
names(hourly_data_gathered)[names(hourly_data_gathered)=="id.x"] <- "unique"
hourly_data_gathered <- merge(hourly_data_gathered, hourly_anger, by="time")
hourly_data_gathered <- merge(hourly_data_gathered, hourly_disgust, by="time")
hourly_data_gathered <- merge(hourly_data_gathered, hourly_fear, by="time")
hourly_data_gathered <- merge(hourly_data_gathered, hourly_joy, by="time")
hourly_data_gathered <- merge(hourly_data_gathered, hourly_negative, by="time")
hourly_data_gathered <- merge(hourly_data_gathered, hourly_positive, by="time")
hourly_data_gathered <- merge(hourly_data_gathered, hourly_anticipation, by="time")
hourly_data_gathered <- merge(hourly_data_gathered, hourly_sadness, by="time")
hourly_data_gathered <- merge(hourly_data_gathered, hourly_surprise, by="time")
hourly_data_gathered <- merge(hourly_data_gathered, hourly_trust, by="time")

#drop columns: id.x and id.y
hourly_data_gathered = subset(hourly_data_gathered, select = -c(id.x, id.y))
hourly_data_gathered = subset(hourly_data_gathered, select = -c(id.x, id.y))
hourly_data_gathered = subset(hourly_data_gathered, select = -c(id.x.1, id.y.1))
hourly_data_gathered = subset(hourly_data_gathered, select = -c(id.x.2, id.y.2))
hourly_data_gathered = subset(hourly_data_gathered, select = -c(id.x.3, id.y.3))
hourly_data_gathered = subset(hourly_data_gathered, select = -c(id.y.4))

#rename unique back to id
names(hourly_data_gathered)[names(hourly_data_gathered)=="unique"] <- "id"

#create total-column
hourly_data_gathered$total <-
rowSums(hourly_data_gathered[,c("coredata.hourly_sum_positivity.",

"coredata.hourly_sum_negativity.")])

#create ratios
hourly_data_gathered$sum_positivity_ratio <-
hourly_data_gathered$coredata.hourly_sum_positivity./hourly_data_gathered$total
hourly_data_gathered$anger_ratio <-
hourly_data_gathered$coredata.hourly_anger./hourly_data_gathered$total
hourly_data_gathered$disgust_ratio <-
hourly_data_gathered$coredata.hourly_disgust./hourly_data_gathered$total
hourly_data_gathered$fear_ratio <-
hourly_data_gathered$coredata.hourly_fear./hourly_data_gathered$total
hourly_data_gathered$joy_ratio <-
hourly_data_gathered$coredata.hourly_joy./hourly_data_gathered$total
hourly_data_gathered$negative_ratio <-
hourly_data_gathered$coredata.hourly_negative./hourly_data_gathered$total
hourly_data_gathered$positive_ratio <-
hourly_data_gathered$coredata.hourly_positive./hourly_data_gathered$total
hourly_data_gathered$anticipation_ratio <-
hourly_data_gathered$coredata.hourly_anticipation./hourly_data_gathered$total
hourly_data_gathered$sadness_ratio <-
hourly_data_gathered$coredata.hourly_sadness./hourly_data_gathered$total
hourly_data_gathered$surprise_ratio <-
hourly_data_gathered$coredata.hourly_surprise./hourly_data_gathered$total
hourly_data_gathered$trust_ratio <-
hourly_data_gathered$coredata.hourly_trust./hourly_data_gathered$total


#_____________MAIN VISUALIZATION (for analyzing general opinion)_____________


#Visualize SUM POSITIVITY RATIO
title_ratio = "Tesla Total Positivity Ratio from Tweets as of "
```

```
#positivity ratio over time
ggplot(hourly_data_gathered, aes(x=hourly_data_gathered$time,
y=hourly_data_gathered$sum_positivity_ratio))+
  xlab("time") + ylab("total positivity ratio")+
  ggtitle(paste0(title_ratio, date, timezone))+
  my_theme+
  stat_smooth(method = "lm", se=FALSE, col="blue")+
  coord_cartesian(ylim=c(0.45, 0.75))+  #change this scale if
  geom_line()

#___________SIGNAL CODE FOR IMPLYING INVESTMENT DECISION_____________________

#the figures 0.8 and 0.4 would have to be statistically determined to provide
strong signal
if(hourly_data_gathered$sum_positivity_ratio > 0.8){
  status <- "buy"
} else if (hourly_data_gathered$sum_positivity_ratio < 0.4){
  status <- "sell"
} else{
  status <- "hold"
}


#____________FACETED VISUALIZATION - FOR PRESENTATION_____________________

#I am using a Excel-modified data frame, in long form, to facet.
#This step is not automated

# Write to CSV
write.csv(hourly_data_gathered, file = "C:/Users/erikh/Desktop/Pratt/Data
Analysis/Assignment 3 - Final Project/hourly_data_gathered_ratios.csv")
#I transform the output CSV file above into a long data format using Excel
#It is inserted here again to show a faceted graph
long_form_sentiment_data <- read.csv("C:/Users/erikh/Desktop/Pratt/Data
Analysis/Assignment 3 - Final Project/hourly_data_gathered_ratios1.csv",
header=TRUE)

ggplot(long_form_sentiment_data, aes(x=reorder(Time, -Ratio),
                                     y=Ratio,
                                     color=Sentiment,
                                     group = 1))+
  geom_line() +
  facet_grid(.~Sentiment) +
  xlab("time") + ylab("sentiment score")+
  labs(title = "Sentiment Score 12/4/2019")
```