A drawing of a cartoon character

Description automatically generated

Kent Appiah, Seán Hayes & Jeff Levesque

https://github.com/jeff1evesque/ist-736 | Final Project

IST-736: Market Sentiment

PRofessor gates

Table of Contents

[Introduction 2](#_Toc11190807)

[Analysis 2](#_Toc11190808)

[Data Preparation 2](#_Toc11190809)

[Twitter API 2](#_Toc11190810)

[Quandl API 3](#_Toc11190811)

[Joining Data 3](#_Toc11190812)

[Exploratory 3](#_Toc11190813)

[Word Clouds 3](#_Toc11190814)

[Sentiment Analysis 3](#_Toc11190815)

[Stop Words 3](#_Toc11190816)

[Baseline Results 3](#_Toc11190817)

[Time series 4](#_Toc11190818)

[Granger Causality 4](#_Toc11190819)

[Classification 4](#_Toc11190820)

[Results 4](#_Toc11190821)

[Conclusions 4](#_Toc11190822)

# Introduction

# Analysis

# Data Preparation

## Twitter API

The Twitter API[[1]](#footnote-1) was implemented using an approved Twitter developer[[2]](#footnote-2) account. A python template file (config--TEMPLATE.py) was created containing dummy text representing the secret key and tokens provided with the Twitter developer account. This file was copied as config.py, with values properly substituted. Changes were a requirement of the general application implementing the Twython[[3]](#footnote-3) package. Within the application, two main twitter functionalities were streamlined. The first allowed general querying through a set of parameters[[4]](#footnote-4), while the second allowed querying content for specified twitter screen names using the timeline[[5]](#footnote-5) component.

Five screen names were queried:

* Jimcramer
* ReformedBroker
* TheStalwart
* LizAnnSonders
* SJosephBurns

Corresponding code generated dataframe structures for each of the above screen names, then outputted the result to an associated csv file[[6]](#footnote-6). On future executions, if the corresponding csv file already exists, then the twitter api did not create duplicate files.

Furthermore, the parameters collected from the twitter accounts were screen\_name, created\_at, and full\_text. Each account was collected using a rate\_limit=900[[7]](#footnote-7). This ensured that the maximum number of tweets could be collected per screen name. However, due to the request limit, roughly 15 minutes needed to transpire, before re-executing to obtain the maximum content for the successive account. Thus, a little over 1.5 hours was required to initially generate local csv files.

## Quandl API

Like the Twitter API, the python Quandl API[[8]](#footnote-8) was utilized to acquire market data, including the Nasdaq index. An account was needed to obtain the associated API key, and the same config.py was utilized, respectively. Moreover, the date range was maximized[[9]](#footnote-9) in order to obtain the largest possible dataset. While obtaining data was not as restricted by the same rate limit as Twitter, a local csv file was created. This ensures integrity and optimization in case a future study extends with additional datasets. While five different columns were returned, only the Index Value was utilized for successive calculations, described in sections below:

* Trade Date
* Index Value
* High,
* Low
* Total Market Value
* Dividend Market Value

## Joining Data

# Exploratory

The exploratory tasks were performed for each twitter screen names, as well as an overall exercise.

## Word Clouds

## Sentiment Analysis

## Stop Words

A rich set of stop words was created[[10]](#footnote-10), utilized prior to vectorization:

# Baseline Results

## Time series

## Granger Causality

## Classification

# Results

# Conclusions

1. <https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets> [↑](#footnote-ref-1)
2. <https://developer.twitter.com/en/apps> [↑](#footnote-ref-2)
3. <https://twython.readthedocs.io/en/latest/> [↑](#footnote-ref-3)
4. <https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html> [↑](#footnote-ref-4)
5. <https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline.html> [↑](#footnote-ref-5)
6. <https://github.com/jeff1evesque/ist-736-hw/tree/master/data> [↑](#footnote-ref-6)
7. <https://developer.twitter.com/en/docs/basics/rate-limiting.html> [↑](#footnote-ref-7)
8. <https://www.quandl.com/tools/python> [↑](#footnote-ref-8)
9. <https://github.com/jeff1evesque/ist-736/blob/9652d7aa79dc576ca5ad671effbb76362beaa72a/app.py#L227> [↑](#footnote-ref-9)
10. <https://github.com/jeff1evesque/ist-736/blob/master/utility/stopwords.py> [↑](#footnote-ref-10)