Senty

Michael Gigiolio, Maxim Moghadam, Andrew Grillot, Stacy Hayes, Jonathan Bayer, and Juan Ferrer Peasley

What is Senty

- Corporate sentiment analysis software
- Assesses public opinion of companies
- Draws from twitter to determine corporate standing
- Determines news media sentiment towards company
- Allows users to view their most recent search

Project objective:

Provide users with a way to determine how the public feels about a company at the current time

Principal Components

Sentiment Analysis

- Draw data from twitter and news media
- Scan for bot accounts and ignore posts
- Run sentiment analysis on received data

User Registration

- Give users ability to register and login
- Catalogue user searches and interest
- Display past searches to user to show how sentiment has changed since last search

Front End

- Provide users with easily navigable website
- Give clear access to important features
- Deliver a visually appealing user experience

Tools Used

Python 	• Basic programming code • Creating functional features	5 Stars
Github 	• Compiling all work • Facilitating peer work/testing	3 Stars
MySQL 	• Store each sentiment analysis • Store data about twitter users	5 Stars
Flask 	• Python microframework • Login system	4 Stars

Tools Used

HTML 	• Website Markup • Interactive Forms	5 Stars
CSS 	• Website Style • Website Layout	5 Stars
Trello 	• Visualizing progress • Dictating workload	2 Stars
JQUERY/AJAX 	• Get/Post methods	2 Stars

Methodologies

Iterative

- Allowed us to finish components portion by portion
- Allowed for optimization before moving further
- Paired well with Github

★★★★★

Agile

- Emphasized teamwork and collaboration
- Allowed easy response to circumstance
- Many software iterations
- Good pacing

★★★★★

Gantt Chart

- Allowed cataloguing of timeframes
- Too rigid
- Emphasized personal tasks rather than collaborative
- More or less unused due to changing specs

★★

Product Execution

Challenges Encountered

Web Framework

- Learning flask
- Ensuring every element of the web framework is properly functioning and integrated

Login

- Difficult to implement
  - Experimented with multiple languages
- Required own database
  - Solved with effort
- Pairing user data with profiles
  - Integration of multiple databases

Challenges Encountered

Bot Detection

- Proper bot detection is very time consuming and requires a lot of data
- Needed to cut corners in order to get the user's request in a reasonable amount of time

Sentiment Analysis

- Originally to be coupled with natural language processing
  - Scrapped, beyond our capabilities
- Planned to display graph over time
  - Would require populating database first, scrapped
- Removing spam that could weight scores
  - Implemented bot detection

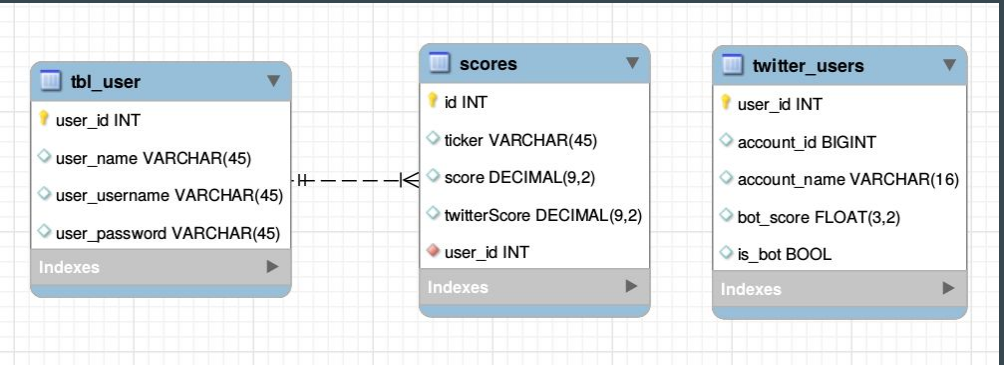
Database

- scores- Records sentiments from news and twitter with the user that made the request
- tbl_user- stores account data for users
- Twitter_users - stores data about twitter accounts.

ticker	score	twitterScore	user_id
aapl	10.49	27.31	1
AAPL	9.61	18.74	1
AAPL	9.36	20.95	1
TWTR	10.89	16.28	1
TSLA	9.10	16.28	1

user_id	user_name	user_username	user_password
87	wer	wef@sdfa	adf
88	andrew	sdf@dnfkjb	sdwa
89	dfwjkb2	WEFKN2@DFknd.c	adf2fe

user_id	account_id	account_name	bot_score	is_bot
1	3239118067	kingrootlion	0.39	0
2	831929636	CalvinKlesmith	0.16	0
3	364589590	ExactOptionPick	0.52	1
4	1095563516	Ironcookies	0.26	0
5	1097179518	basarav	0.21	0



Testing Procedures

Automated Testing

- Performing assertion tests with PyUnit library
- Confirming variable data falls within acceptable ranges
- Verifying error handling
- Ensuring proper website pathing

User Acceptance Testing

- User testing website
- Providing erroneous queries
- Confirming bot detection
- Demonstrating website intuitiveness
- Ensuring home page text is accurate and informative