ECE 4813 - Final Project

A Real-Time Twitter Public Sentiment Analysis Framework

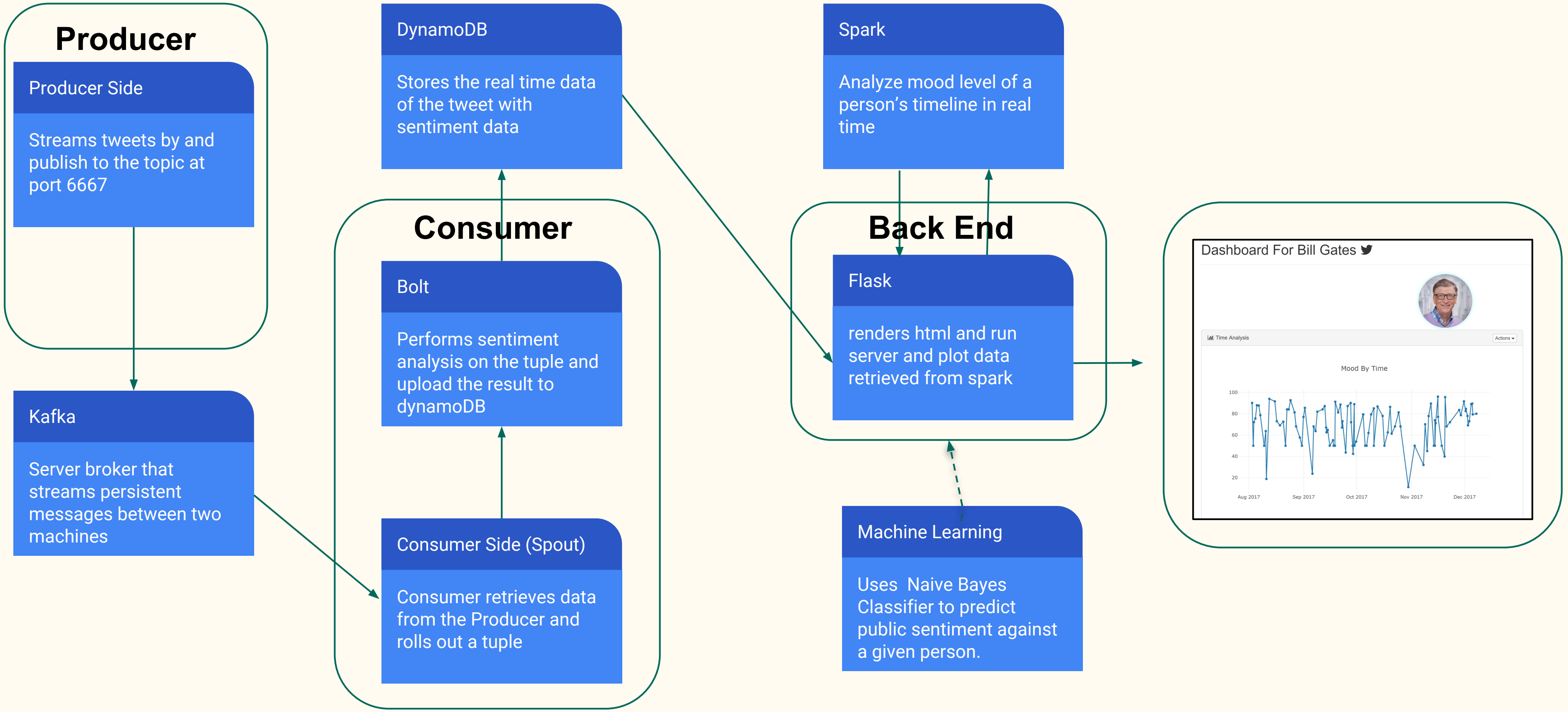
Seung Kwang Son, Chang Min Lee, YoungBin Byun, Hokyung Hwang, Pratik Sharma

Georgia Tech School of Electrical and Computer Engineering
College of Engineering

Background

- Objective: Our platform is catered to users looking for a way to follow the moods of celebrities. The objective is not only to find a place for **sentiments of the tweets of celebrities** through predictive analysis but also to provide users with constant updates on **public sentiment and live tweets** about that person. In this way, researchers can identify patterns between public opinion of celebrities and how they affect each others.
- Problem: News articles are not always accurate source of knowing the facts of the person.

Overall Architecture



Producer Side

Streams tweets by and publish to the topic at port 6667

Kafka

Server broker that streams persistent messages between two machines

DynamoDB

Stores the real time data of the tweet with sentiment data

Bolt

Performs sentiment analysis on the tuple and upload the result to dynamoDB

Consumer Side (Spout)

Consumer retrieves data from the Producer and rolls out a tuple

Spark

Analyze mood level of a person's timeline in real time

Flask

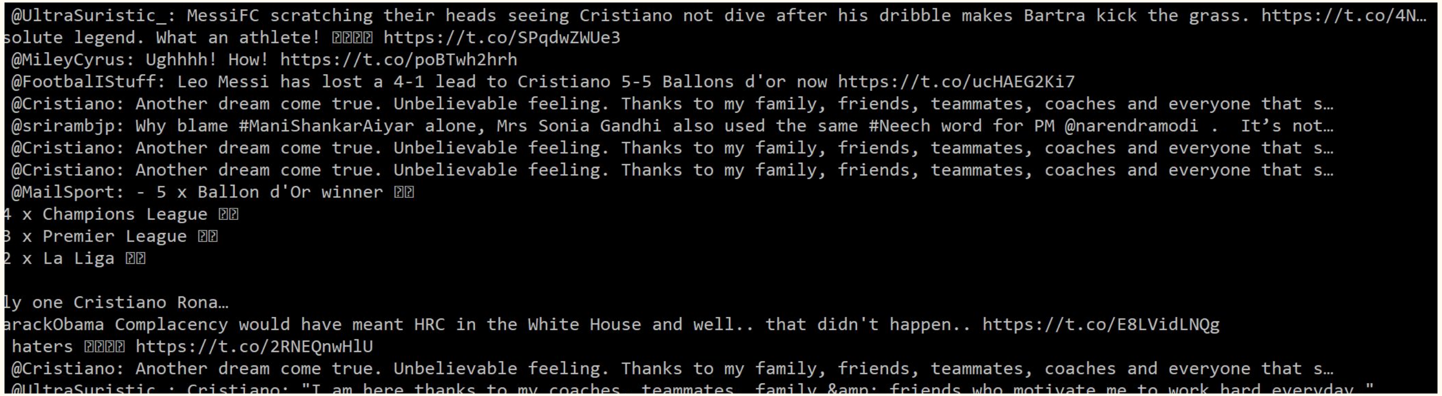
renders html and run server and plot data retrieved from spark

Machine Learning

Uses Naive Bayes Classifier to predict public sentiment against a given person.

Producer - (Stream.py)

- Tweepy StreamListener()
- Filter keys set to top Twitter accounts: “@JustinBiber, @realDonaldTrump …”
- Retrieve real time data of public tweets and publish to the Kafka stream

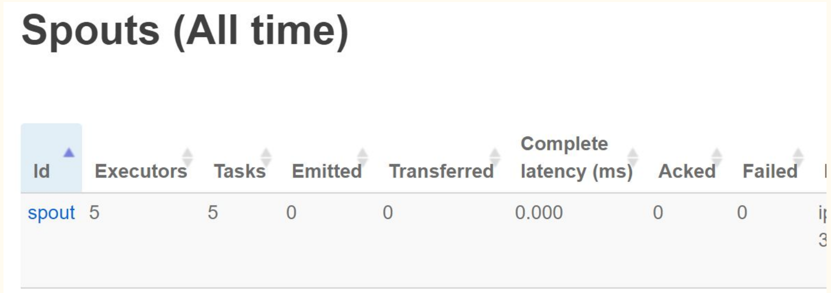


```
 @UltraSuristic_: MessiFC scratching their heads seeing Cristiano not dive after his dribble makes Bartra kick the grass. https://t.co/4N…
solute legend. What an athlete! □□□□ https://t.co/SPqdwZWUe3
 @MileyCyrus: Ughhhh! How! https://t.co/poBTwh2hrh
 @FootbalIStuff: Leo Messi has lost a 4-1 lead to Cristiano 5-5 Ballons d'or now https://t.co/ucHAEG2Ki7
 @Cristiano: Another dream come true. Unbelievable feeling. Thanks to my family, friends, teammates, coaches and everyone that s…
 @srirambjp: Why blame #ManiShankarAiyar alone, Mrs Sonia Gandhi also used the same #Neech word for PM @narendramodi .  It’s not…
 @Cristiano: Another dream come true. Unbelievable feeling. Thanks to my family, friends, teammates, coaches and everyone that s…
 @Cristiano: Another dream come true. Unbelievable feeling. Thanks to my family, friends, teammates, coaches and everyone that s…
 @MailSport: - 5 x Ballon d'Or winner □□
4 x Champions League □□
3 x Premier League □□
2 x La Liga □□

ly one Cristiano Rona…
arackObama Complacency would have meant HRC in the White House and well.. that didn't happen.. https://t.co/E8LVidLNQg
 haters □□□□ https://t.co/2RNEQnwHlU
 @Cristiano: Another dream come true. Unbelievable feeling. Thanks to my family, friends, teammates, coaches and everyone that s…
 @UltraSuristic_: Cristiano: "I am here thanks to my coaches, teammates, family &amp; friends who motivate me to work hard everyday."
```

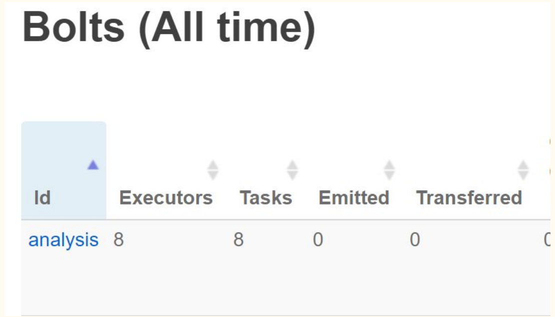
Storm

- Given data of streams feeded into Kafka producer, Spout will consume data from producer and transfer to bolts.
- Bolts will use the data from the Spout, calculate the mood level, using NLTK, and insert it into Dynamodb.



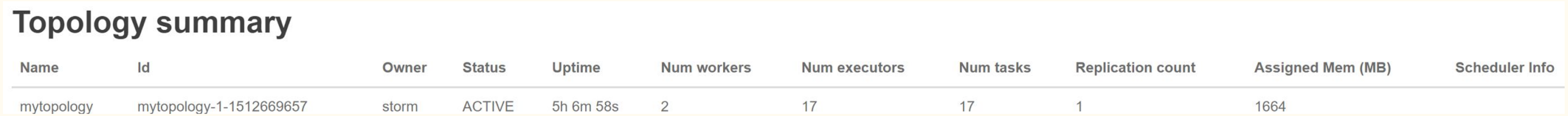
Spouts (All time)

Id	Executors	Tasks	Emitted	Transferred	Complete latency (ms)	Acked	Failed	I
spout	5	5	0	0	0.000	0	0	i 3



Bolts (All time)

Id	Executors	Tasks	Emitted	Transferred	
analysis	8	8	0	0	(

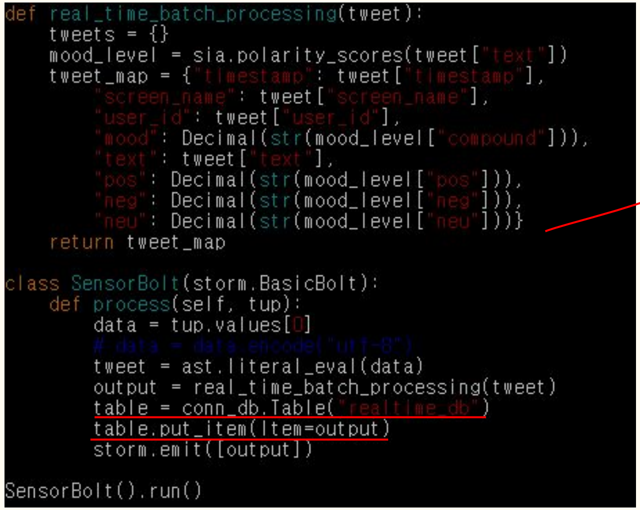


Topology summary

Name	Id	Owner	Status	Uptime	Num workers	Num executors	Num tasks	Replication count	Assigned Mem (MB)	Scheduler Info
mytopology	mytopology-1-1512669657	storm	ACTIVE	5h 6m 58s	2	17	17	1	1664	

Storm

- Used Dynamo DB for storing real time tweets about the user.
  - Tweet's text, timestamp, screen_name, and user_id are passed into the real_time_batch_processing()
  - NLTK SentimentIntensityAnalyzer outputs sentiment scores that are inserted into Dynamo DB

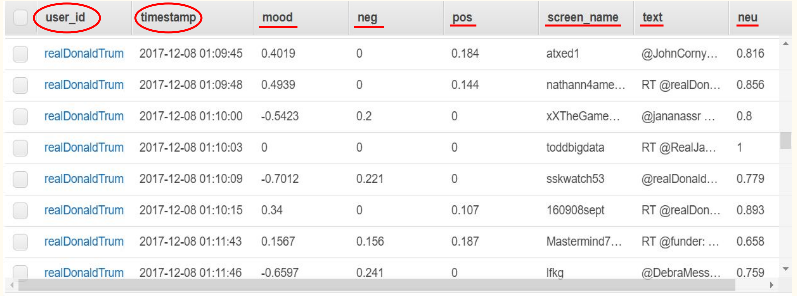


```
def real_time_batch_processing(tweet):
    tweets = {}
    mood_level = sia.polarity_scores(tweet["text"])
    tweet_map = {"timestamp": tweet["timestamp"],
        "screen_name": tweet["screen_name"],
        "user_id": tweet["user_id"],
        "mood": Decimal(str(mood_level["compound"])),
        "text": tweet["text"],
        "pos": Decimal(str(mood_level["pos"])),
        "neg": Decimal(str(mood_level["neg"])),
        "neu": Decimal(str(mood_level["neu"]))}
    return tweet_map

class SensorBolt(storm.BasicBolt):
    def process(self, tup):
        data = tup.values[0]
        # data = data.encode("utf-8")
        tweet = ast.literal_eval(data)
        output = real_time_batch_processing(tweet)
        table = conn_db.Table("realtime_db")
        table.put_item(Item=output)
        storm.emit([output])

SensorBolt().run()
```

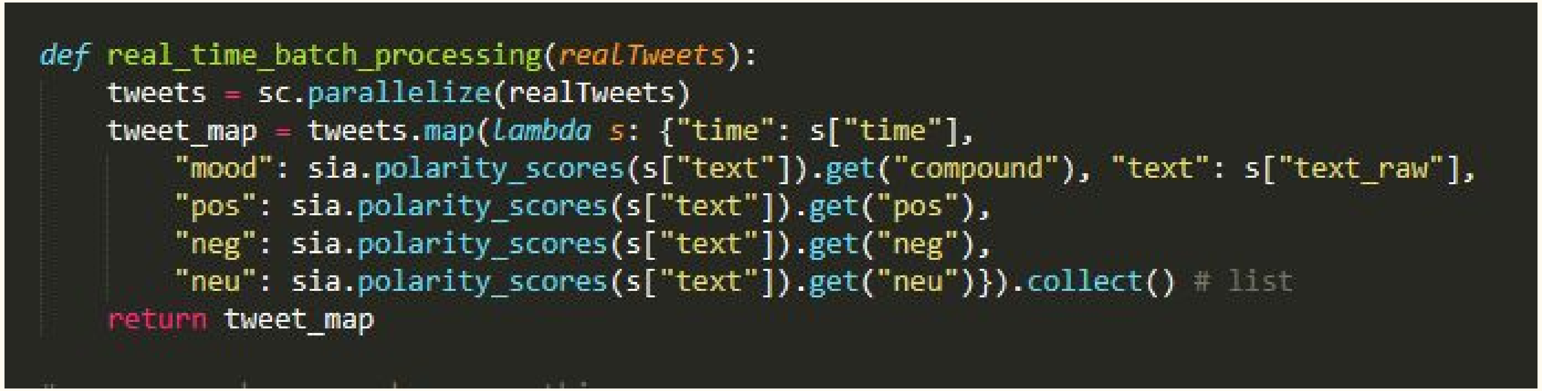
⬭	-	**Key attributes**
—	-	**Regular attributes**



user_id	timestamp	mood	neg	pos	screen_name	text	neu
realDonaldTrum	2017-12-08 01:09:45	0.4019	0	0.184	atxed1	@JohnCorny…	0.816
realDonaldTrum	2017-12-08 01:09:48	0.4939	0	0.144	nathann4ame…	RT @realDon…	0.856
realDonaldTrum	2017-12-08 01:10:00	-0.5423	0.2	0	xXTheGame…	@jananassr …	0.8
realDonaldTrum	2017-12-08 01:10:03	0	0	0	toddbigdata	RT @RealJa…	1
realDonaldTrum	2017-12-08 01:10:09	-0.7012	0.221	0	sskwatch53	@realDonald…	0.779
realDonaldTrum	2017-12-08 01:10:15	0.34	0	0.107	160908sept	RT @realDon…	0.893
realDonaldTrum	2017-12-08 01:11:43	0.1567	0.156	0.187	Mastermind7…	RT @funder: …	0.658
realDonaldTrum	2017-12-08 01:11:46	-0.6597	0.241	0	lfkg	@DebraMess…	0.759

Spark

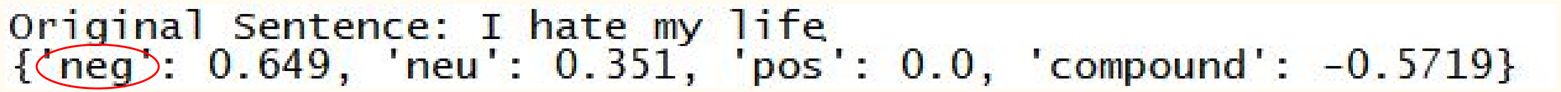
- Analyze a given user’s timeline in real time.
- Calculates the mood level of 100 timeline data of given user in a real time.



```python
def real_time_batch_processing(realTweets):
    tweets = sc.parallelize(realTweets)
    tweet_map = tweets.map(lambda s: {"time": s["time"],
        "mood": sia.polarity_scores(s["text"]).get("compound"), "text": s["text_raw"],
        "pos": sia.polarity_scores(s["text"]).get("pos"),
        "neg": sia.polarity_scores(s["text"]).get("neg"),
        "neu": sia.polarity_scores(s["text"]).get("neu")}).collect() # list
    return tweet_map
```

Spark Batch Process backed by NLTK

- Use natural language processing to find the positive, negative, neutral level of a given sentence from negative one to positive one.
- This also outputs aggregated score as "compound"



```
Original Sentence: I hate my life
{'neg': 0.649, 'neu': 0.351, 'pos': 0.0, 'compound': -0.5719}
```

Naive Bayes Classifier (Machine Learning)

- Wrote and implemented a Naive Bayes Classifier based on opensourceforu.com
- Supervised learning classifier
- Trained it using 3 datasets (Amazon, Yelp, IMDB)
- Classifies tweets based on the training as positive or negative
- The main problem was the amount of time taken for the algorithm to analyze the tweets based on the training data

Flask Back End

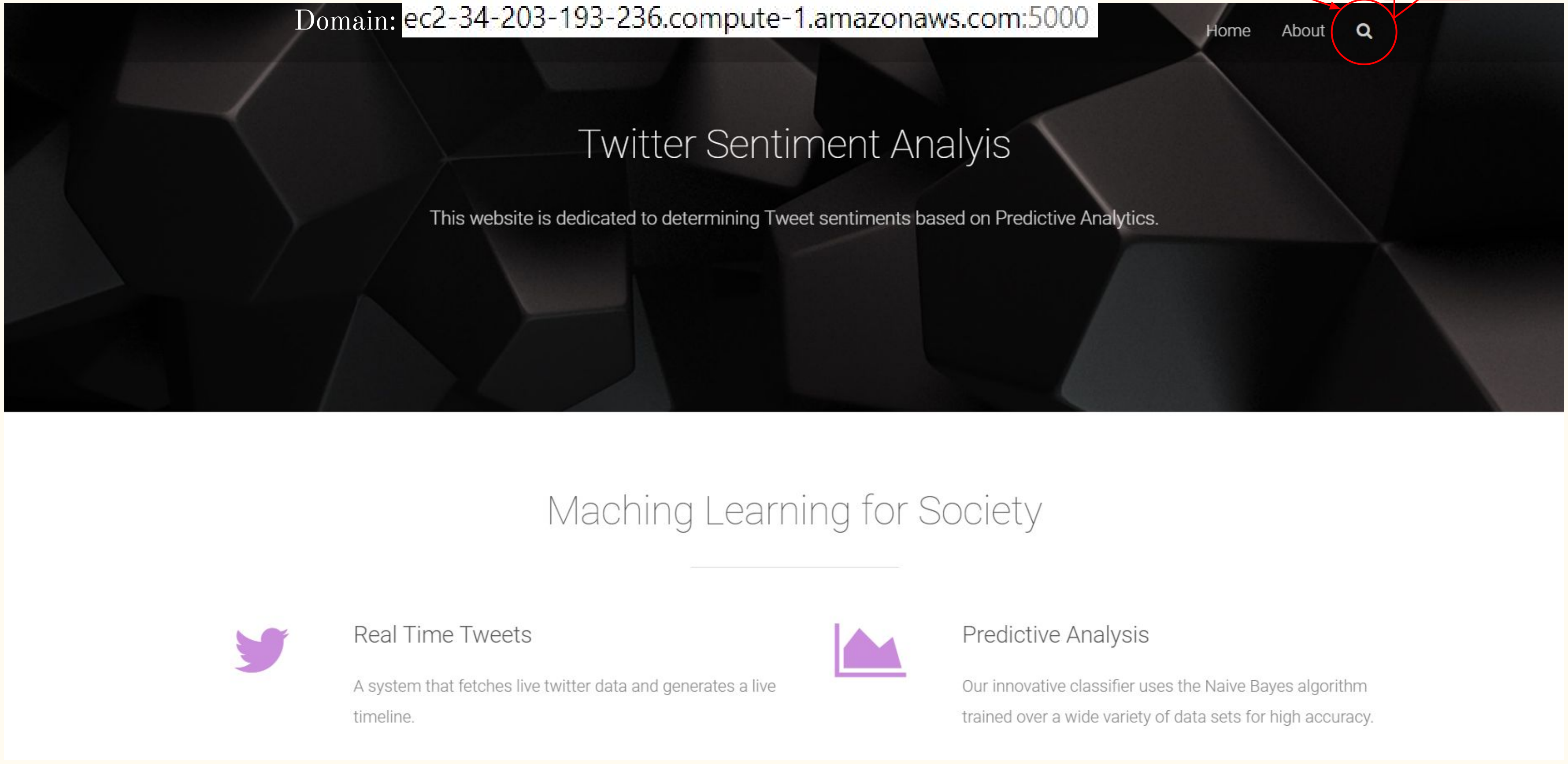
- Interacts with Spark to retrieve mood levels of a given user's over recent 100 tweets
- Pass dataset of the timeline of tweets and mood levels scaled from 0 to 100 to the front end setting update interval to 10 seconds
- Give the user's ten most recent tweets and their corresponding mood level from the timeline to the webpage
- Receive 5 most recent tweets about the user from Bolt, store them in Dynamo DB and displays

Front End

- Plot the mood level in Y axis and timestamp in X axis.
- Display original tweets of a given user with their mood levels.
- Show the change “mood” level of public opinion about certain celebrity (0-100) by using Gauge API.
- Display tweets about a given user in real time.

Front End (index.html)

search for the person you want.



Domain: ec2-34-203-193-236.compute-1.amazonaws.com:5000

Twitter Sentiment Analyis

This website is dedicated to determining Tweet sentiments based on Predictive Analytics.

Maching Learning for Society

Real Time Tweets

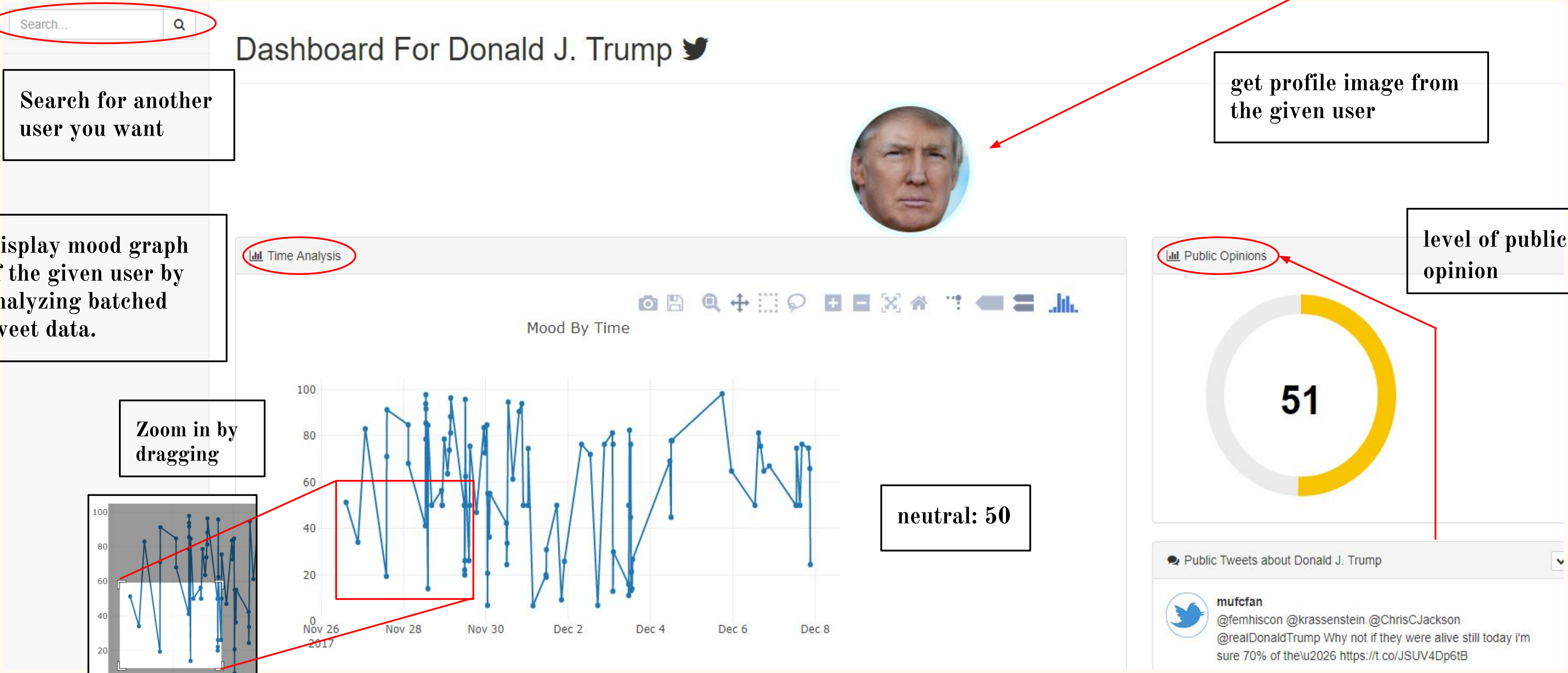
A system that fetches live twitter data and generates a live timeline.

Predictive Analysis

Our innovative classifier uses the Naive Bayes algorithm trained over a wide variety of data sets for high accuracy.

Front End (result.html)

Domain: ec2-34-203-193-236.compute-1.amazonaws.com:5000/user/realDonaldTrump



Search...

Dashboard For Donald J. Trump

Public Tweets about Donald J. Trump

mufcfan
@femhiscon @krassenstein @ChrisCJackson @realDonaldTrump Why not if they were alive still today i'm sure 70% of the\u2026 https://t.co/JSUV4Dp6tB

Search for another user you want

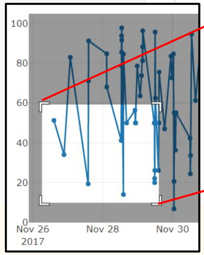
get profile image from the given user

Display mood graph of the given user by analyzing batched tweet data.

level of public opinion

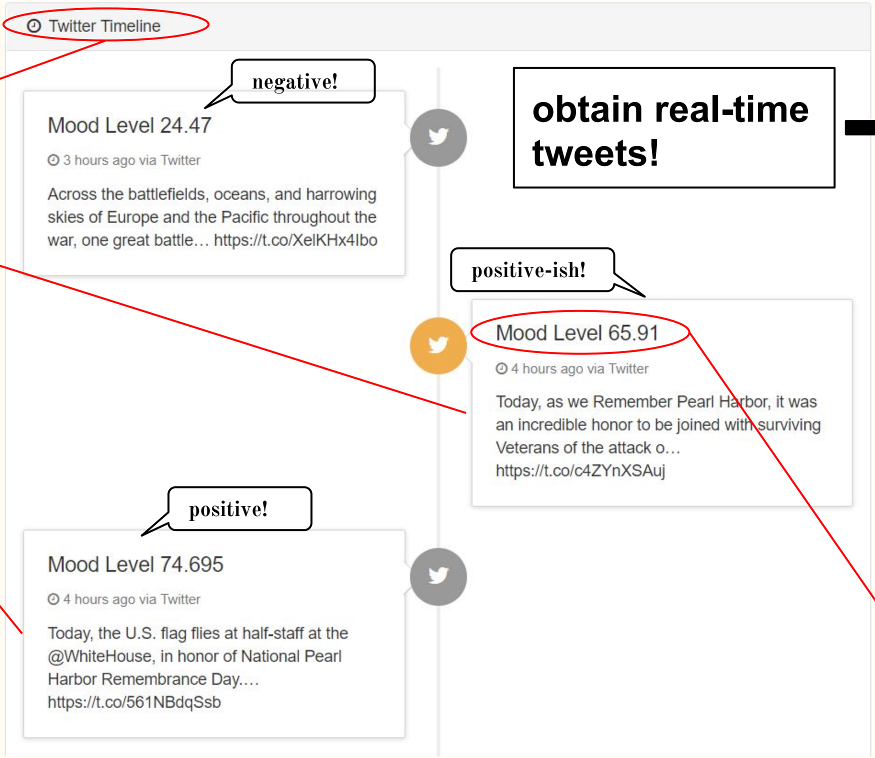
Zoom in by dragging

neutral: 50



Front End

Display tweets from the given user in chronological order.



Twitter Timeline

negative!

Mood Level 24.47

3 hours ago via Twitter

Across the battlefields, oceans, and harrowing skies of Europe and the Pacific throughout the war, one great battle… https://t.co/XelKHx4lbo

positive-ish!

Mood Level 65.91

4 hours ago via Twitter

Today, as we Remember Pearl Harbor, it was an incredible honor to be joined with surviving Veterans of the attack o… https://t.co/c4ZYnXSAuj

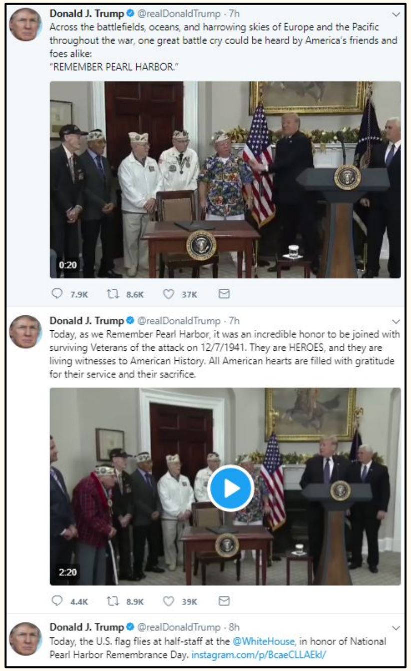
positive!

Mood Level 74.695

4 hours ago via Twitter

Today, the U.S. flag flies at half-staff at the @WhiteHouse, in honor of National Pearl Harbor Remembrance Day.… https://t.co/561NBdqSsb

obtain real-time tweets!



Donald J. Trump @realDonaldTrump · 7h

Across the battlefields, oceans, and harrowing skies of Europe and the Pacific throughout the war, one great battle cry could be heard by America's friends and foes alike:
"REMEMBER PEARL HARBOR."

0:20

7.9K 8.6K 37K

Donald J. Trump @realDonaldTrump · 7h

Today, as we Remember Pearl Harbor, it was an incredible honor to be joined with surviving Veterans of the attack on 12/7/1941. They are HEROES, and they are living witnesses to American History. All American hearts are filled with gratitude for their service and their sacrifice.

2:20

4.4K 8.9K 39K

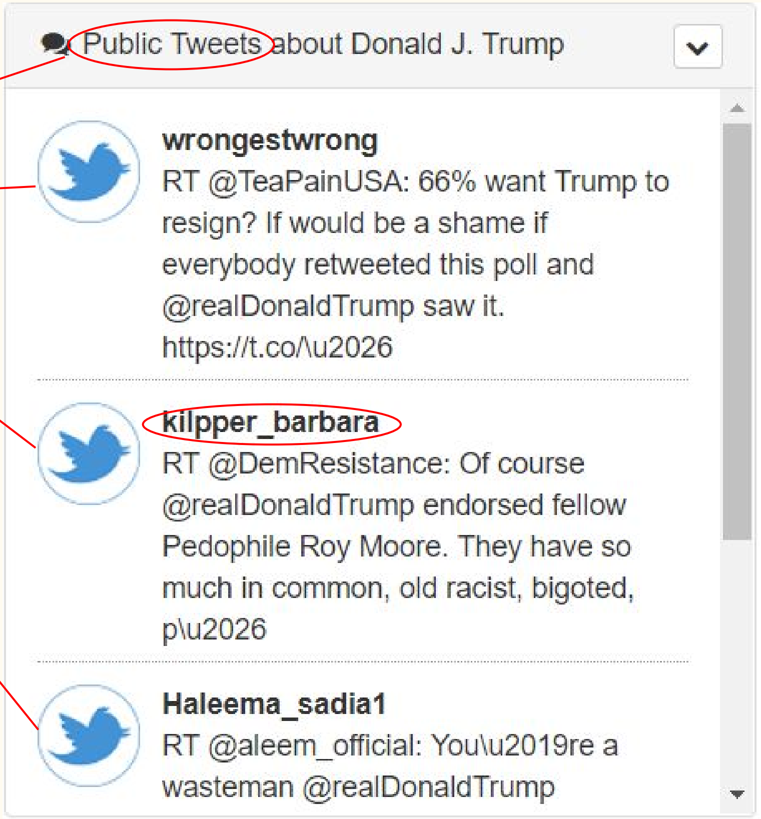
Donald J. Trump @realDonaldTrump · 8h

Today, the U.S. flag flies at half-staff at the @WhiteHouse, in honor of National Pearl Harbor Remembrance Day. instagram.com/p/BcaeCLLAEkl/

Through the sentimental analysis, the mood value of given user is calculated.
(0 ------------------100)
negative positive

Front End

Display public tweets about given user



wrongestwrong
RT @TeaPainUSA: 66% want Trump to resign? If would be a shame if everybody retweeted this poll and @realDonaldTrump saw it. https://t.co/\u2026

kilpper_barbara
RT @DemResistance: Of course @realDonaldTrump endorsed fellow Pedophile Roy Moore. They have so much in common, old racist, bigoted, p\u2026

Haleema_sadia1
RT @aleem_official: You\u2019re a wasteman @realDonaldTrump

This embedded timeline is updated every 10 seconds.

Code Analysis (Codes Attached)

Bolt.py	**Description**
real_time_batch_processing()	Get the user id from the user entry and post 10 recent posts
Class SensorBolt: process():	Renders the entire page with plotting results and streamed tweet

Stream_producer.py	**Description**
topAccounts.txt	List of target accounts that we want to stream "realDonaldTrump; justinbieber; billgates ..."
publish(tweet)	Data["timestamp"] = "2017-03-22 14: 22:13" Data["text"] = "@james wow this project #ece4813 #finalproject" Data["user_id"] = "abcd341" Data["screen_name"] = "realDonaldTrump"

Code Analysis (Codes Attached)

Dashboard.py	Description
@App.route(‘/’, methods=’GET’, ‘POST’) real_time_get_timeline(used_id)	Get the user id from the user entry and post 10 recent tweets
@App.route(‘/user/<userAccount>’, methods = [‘GET’, ‘POST’] get_profile(userAccount)	Renders the entire page with plotting results and streamed tweets
@App.route(‘/user/<userAccount>’, methods=[‘GET’, ‘POST’] get_basic_info(userAccount)	Returns user’s name, screen name, picture URL, and the user ID
@App.route(‘/get_publicopinion’/<userAccount>’) get_publicopinion(userAccount)	Scales public sentiment scores and returns Jsonified result
@App.route(‘/get_public_tweet/<userAccount>’, methods=[‘GET’] get_public_tweet(userAccount)	get_stream(UserID)
real_time_batch_processing(realTweets)	NLTK Vader processing on Spark
naive_bayes_analysis(realTweets)	NLTK Bayes processing on Spark (*Unused. Check the performance analysis)

Performance Analysis

- Spark Real Time
  - Tweepy Stream > Producer > Storm > DynamoDB > Flask App
  - Travel Time = timestamp_Flask App - timestamp_Tweepy Stream

- Spark Batch
  - Tested 100 tweets, 1000 tweets (Iterated 10 times)

- NLTK ML-Trained Algorithm
  - Trained over 3 datasets (IMDB, Amazon, and Yelp)
  - Utilizes a Naive Bayes

	Spark Real-Time (NLTK Vader)	Spark Batch (NLTK Vader)	NLTK Naive Bayes
Processing Time	**~1.5 seconds / tweet**	**~3s / 100 tweets** ~40s / 10,000 tweets	**~7.5 seconds / tweet**

Summary

- Earlier prototypes processed tweets of pre-selected accounts (Locally stored tweets)
  - Very low latency but limited to fixed number of Twitter accounts
  - To make the app more ***scalable***, it now processes 100 tweets per user's request on any account
- Batch processing time varies by number of tweets (100 tweets vs 10,000 tweets). 100 tweets is more optimal number to analyze user's mood while not slowing down the speed of the app
- For sentiment analysis we evaluated the performance of NLTK Vader and a Naive Bayes Classifier and found a significant difference in run time. Due to this finding we decided to use the NLTK Vader sentiment analysis

Contribution

- Hokyung Hwang
  - Backend (Flask)
  - Kafka (Pipelining Tweepy, Producer, Consumer)
  - Apache Storm
- Seung Kwang Son
  - Instantiated EC2, set-up Ambari
  - Kafka (Pipelining Tweepy, Producer, Consumer)
  - Backend (Flask)
- Youngbin Byun
  - DynamoDB
  - UI (front-end)
  - Backend (Flask)
- Chang Min Lee
  - Apache Storm
  - UI design (front-end)
  - Backend (Flask)
- Pratik Sharma
  - HTML/Javascript (front-end)
  - Backend (Flask)
  - NLTK ML Implementations