

# Boston University

## CAS CS 660 - Introduction To Database Systems

### PA3-extra: NoSQL Databases and Twitter Analysis

**This is ONLY for CS 660 students**

**Due on: Tuesday, Dec 12, 2017 at 11:59PM.**

#### 1. Introduction

In this project you will learn how to get tweets from the Twitter Website in real-time (streaming mode), how to store them in a MongoDB database and retrieve them in a Python code using PyMongo, in addition to playing with the data within the Mongo shell itself.

For this extra project which is geared only for grad students in CS660, we expect students to be able to install all the necessary packages on their own and be able to search and research for ways to do things. For some of the tasks we have provided suggestions on how to perform them but you could use any other methods to get the task done as long as you are using PyMongo within Python except for the last **extra point** task which you might want to use other methods and languages. We tried to keep it fun and engaging and we wish you a great rest of semester ahead.

For each part you should write related Python code either using PyMongo API or pure Python code or using other 3rd party libraries. You need to **gsubmit** your entire code in a **zip** file in the format of **firstname\_lastname\_CS660.zip** by **Tuesday, December 12, 2017 at 11:59PM**.

## Part 1)

For this part of the project, you use the Twitter data mining script (pymongo\_tweepy.py) given to you and modify it such that it mines tweets with the keywords **#deeplearning**, **#computervision**, **#datascience**, and **#bigdata**. Your streamer, similar to the original file, should stream on track (search for keywords) (while in Part 2 you stream based on location).

Here's what a single tweet would look like when stored in MongoDB:

Use the command **> db.twitter\_search.find().limit(1)**

```
> db.twitter_search.find().limit(1)
{ "_id" : ObjectId("5a2334cae977e59684f710d2"), "created_at" : "Sat Dec 02 23:18:34 +0000 2017", "id" : NumberLong("937098695891398656"), "id_str" : "937098695891398656", "text" : "Advanced Linear Models for Data Science 1: Least Squares #machinelearning #bigdata #ai https://t.co/5MKcCCHVWE", "source" : "<a href='\"https://ifttt.com/\"' rel='\"nofollow\"'>IFTTT</a>", "truncated" : false, "in_reply_to_status_id" : null, "in_reply_to_status_id_str" : null, "in_reply_to_user_id" : null, "in_reply_to_user_id_str" : null, "in_reply_to_screen_name" : null, "user" : { "id" : NumberLong("3585208755"), "id_str" : "3585208755", "name" : "Ave Johnson", "screen_name" : "xtools_at", "location" : "Vienna, Austria", "url" : "https://www.leasingrechner.at", "description" : "eCommerce Hero, UX-/Android-/Web-Developer, Tinkerer tweeting Simply Amazing Stuff ;)\n\nLatest project: https://www.leasingrechner.at", "translator_type" : "none", "protected" : false, "verified" : false, "followers_count" : 6210, "friends_count" : 5868, "listed_count" : 798, "favourites_count" : 328, "statuses_count" : 63452, "created_at" : "Tue Sep 08 07:18:29 +0000 2015", "utc_offset" : -28800, "time_zone" : "Pacific Time (US & Canada)", "geo_enabled" : false, "lang" : "en", "contributors_enabled" : false, "is_translator" : false, "profile_background_color" : "222222", "profile_background_image_url" : "http://abs.twimg.com/images/themes/theme1/bg.png", "profile_background_image_url_https" : "https://abs.twimg.com/images/themes/theme1/bg.png", "profile_background_tile" : false, "profile_link_color" : "6699CC", "profile_sidebar_border_color" : "000000", "profile_sidebar_fill_color" : "000000", "profile_text_color" : "000000", "profile_use_background_image" : false, "profile_image_url" : "http://pbs.twimg.com/profile_images/642027067878600704/778HY_sY_normal.jpg", "profile_image_url_https" : "https://pbs.twimg.com/profile_images/642027067878600704/778HY_sY_normal.jpg", "profile_banner_url" : "https://pbs.twimg.com/profile_banners/3585208755/1469396450", "default_profile" : false, "default_profile_image" : false, "following" : null, "follow_request_sent" : null, "notifications" : null }, "geo" : null, "coordinates" : null, "place" : null, "contributors" : null, "is_quote_status" : false, "quote_count" : 0, "reply_count" : 0, "retweet_count" : 0, "favorite_count" : 0, "entities" : { "hashtags" : [ { "text" : "machinelearning", "indices" : [ 57, 73 ] }, { "text" : "bigdata", "indices" : [ 74, 82 ] }, { "text" : "ai", "indices" : [ 83, 86 ] } ], "urls" : [ { "url" : "https://t.co/5MKcCCHVWE", "expanded_url" : "http://bit.ly/2BGPrtQ", "display_url" : "bit.ly/2BGPrtQ", "indices" : [ 87, 110 ] } ], "user_mentions" : [ ], "symbols" : [ ] }, "favorited" : false, "retweeted" : false, "possibly_sensitive" : false, "filter_level" : "low", "lang" : "en", "timestamp_ms" : "1512256714209" }
>
```

In order to find the number of tweets in your database, you could use the following command:

**> db.twitter\_search.find().count()**

For the purpose of the project please retrieve ~1000 tweets using the given instruction in [https://github.com/monajalal/mongo\\_tweets](https://github.com/monajalal/mongo_tweets). You would need to do a **git pull** to get the latest version of the code if you

already have **git cloned** the repository. For further instruction on how to get the repo and get started with Twitter API please check Lab11\_extra *in case you didn't attend the lab on December 1st, 2017.*

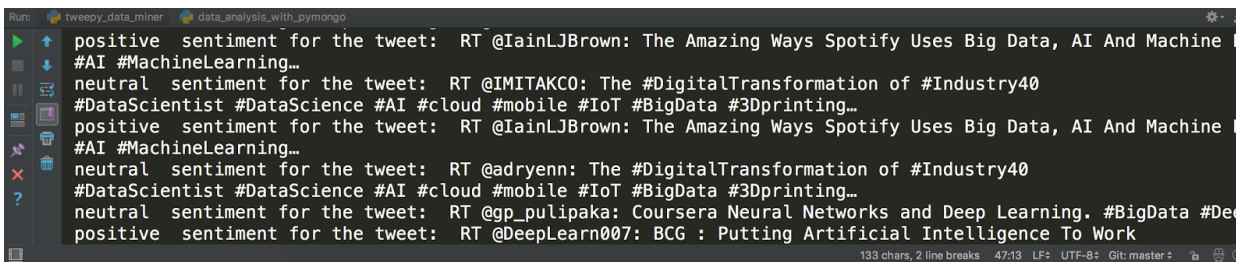
**Lab11\_extra:**

<https://docs.google.com/document/d/1rCAgy7V1q8u4E33XwW0-3E0d9xU07WN6prpfbnoATRA/edit?usp=sharing>

For this project, you would need to refer to Tweet Object definitions here <https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object>

- A. Find the number of tweets that have data somewhere in the tweet's text (case insensitive search using [regex](#))
  
- B. From all the data related objects, how many of them are geo\_enabled?
  
- C. For all the data related tweets, use the [TextBlob](#) Python library to detect if the Tweet's sentiment is "**Positive**", "**Neutral**", or "**Negative**". You are free to use other sensible methods and libraries to do so. (Hint: To get better results you could **clean your Tweet's text** of unwanted characters/emoji/etc--not obligatory and we wouldn't deduct point based on accuracy, whatsoever).

Your final results should look like something like below:



```

Run: tweepy_data_miner data_analysis_with_pymongo
positive sentiment for the tweet: RT @IainLJBrown: The Amazing Ways Spotify Uses Big Data, AI And Machine
#AI #MachineLearning...
neutral sentiment for the tweet: RT @IMITAKC0: The #DigitalTransformation of #Industry40
#DataScientist #DataScience #AI #cloud #mobile #IoT #BigData #3Dprinting...
positive sentiment for the tweet: RT @IainLJBrown: The Amazing Ways Spotify Uses Big Data, AI And Machine
#AI #MachineLearning...
neutral sentiment for the tweet: RT @adryenn: The #DigitalTransformation of #Industry40
#DataScientist #DataScience #AI #cloud #mobile #IoT #BigData #3Dprinting...
neutral sentiment for the tweet: RT @gp_pulipaka: Coursera Neural Networks and Deep Learning. #BigData #De
positive sentiment for the tweet: RT @DeepLearn007: BCG : Putting Artificial Intelligence To Work
133 chars, 2 line breaks 47:13 LF UTF-8 Git: master

```

## Part 2)

- A. Create a new script that mines Tweets from Twitter using the Tweepy API so that instead of mining Tweets based on keywords, it mines tweets based on location. Basically, change the **stream.filter** so that for the location field it takes the **United States bounding box** (visit <http://boundingbox.klokantech.com> for finding this info) longitude, latitude. Additionally, modify the script so that from all the real-time tweets that it streams, it only saves those into the MongoDB using PyMongo that their '**coordinates**' field '**is not None**'. Also, in the mining code given to you, modify the twitterdb to usa\_db and twitter\_search to usa\_tweets\_collection. Leave it running until you mine ~**10000** tweets by checking **db.usa\_tweets\_collection.find().count()**
- B. (mostly Python coding) Do some searching in Google to find how to extract emojis from a text. We suggest you to use **from emoji import UNICODE\_EMOJI** for this purpose but feel free to use any library that can help you. Find the tweets that have at least one emoji in them and use **defaultdict**(or **dictionary**) to save the count of emoji per state and state per emoji.
1. What are the top 15 emojis used in the entire tweets?

1. You should report something like this: [('♥', 61), ('🌲', 39), ('💙', 23), ('😍', 22), ('🔥', 19), ('👶', 18), ('👍', 13), ('😁', 12), ('❄', 12), ('😂', 12), ('📺', 11), ('💕', 11), ('👊', 11), ('🎵', 10), ('💖', 9)]

2. What are the top 5 states for the emoji 🌲?

3. What are the top 5 emojis for MA?

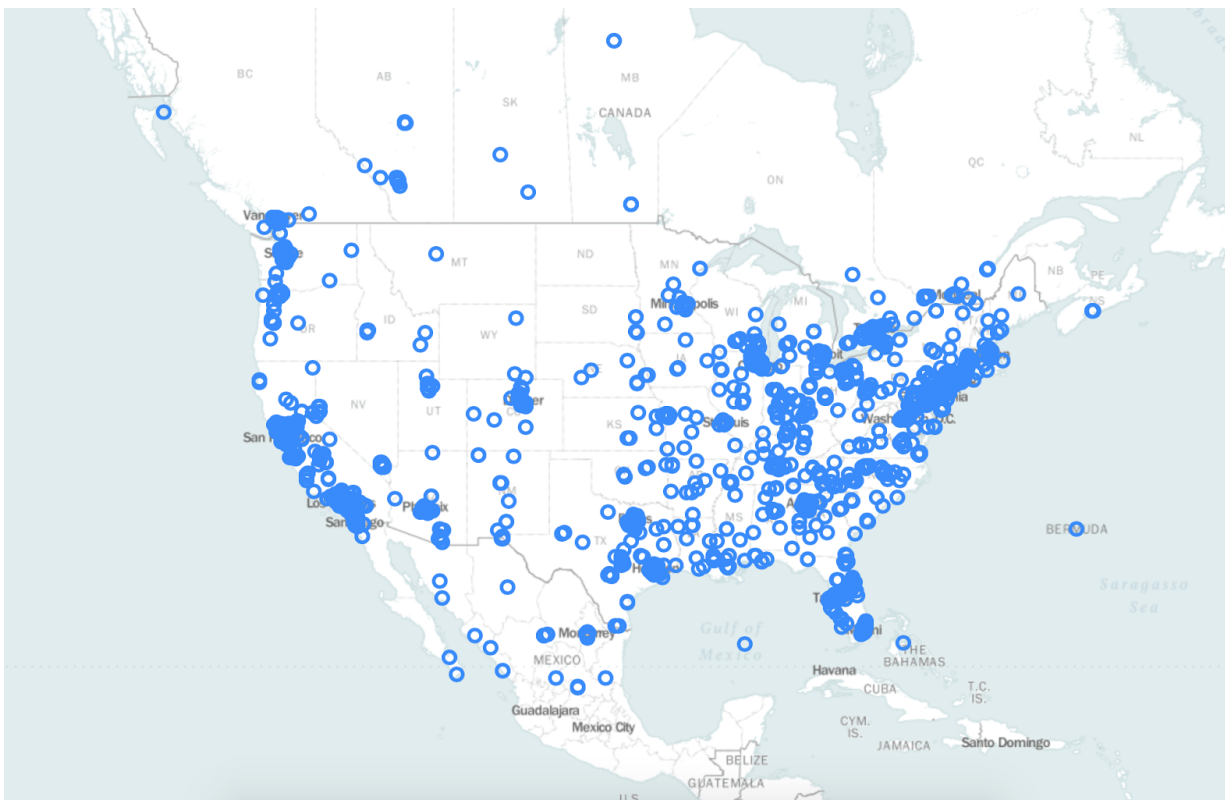
4. What are the top 5 states that use emojis?

C. Use MongoDB queries within PyMongo API to answer the following:

1. What are the top 5 states that have tweets?

2. In the state of California, what are the top 5 cities that tweet?

D. We have given you the file **json\_to\_csv.py** which converts the database saved in MongoDB to csv format. You use that to create a map of all the tweets using **Folium**--you are free to use other methods-- Python library. Your eventual map should look like something below (and of course brownie points if it looks better). Some of the methods that you would possibly find useful from Folium are: **Map** and **CircleMaker**. Eventually you have to use the **save** method to save your map in “**map.html**” file. We expect the students to read more about the necessary methods from Folium documentation which has plenty of examples here <https://media.readthedocs.org/pdf/folium/latest/folium.pdf> (this task will not really require more than 10 lines of code).



**Extra credit (10 out of 100)**

**For part 2) B, create the map of USA with top 2 emojis per state (you could use any language you want).**

**Resources:**

<https://www.thesisscientist.com/docs/Dr.JakeFord/f900e601-2cc7-4c34-a45e-a9d363e43026.pdf>