

Machine Learning With Social Media

By Sidharth Srivastava

A decorative network diagram in the top-left corner, consisting of various sized grey circles connected by thin grey lines, some with dashed outlines.

What is Machine Learning?

The Basics

A decorative network diagram in the bottom-right corner, consisting of various sized grey circles connected by thin grey lines, some with dashed outlines.



“

A breakthrough in machine learning would be worth ten Microsofts.

-Bill Gates

Machine Learning

The use of statistical processes by machines to analyze data and make decisions



Machine Learning

- ◎ A few key steps:
- ◎ Data gathering
- ◎ Model Training
- ◎ Evaluation
- ◎ Prediction



Types of Machine Learning

Unsupervised

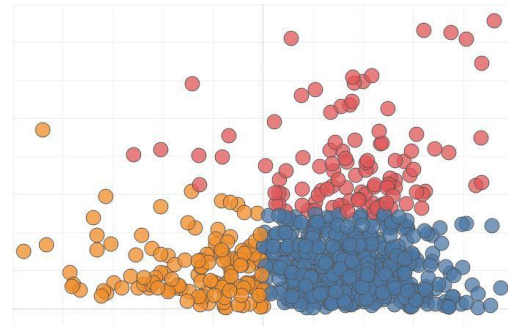
- ⦿ Data without given labels
- ⦿ Algorithm discovers layers and groups in data
- ⦿ Can be used to find labels

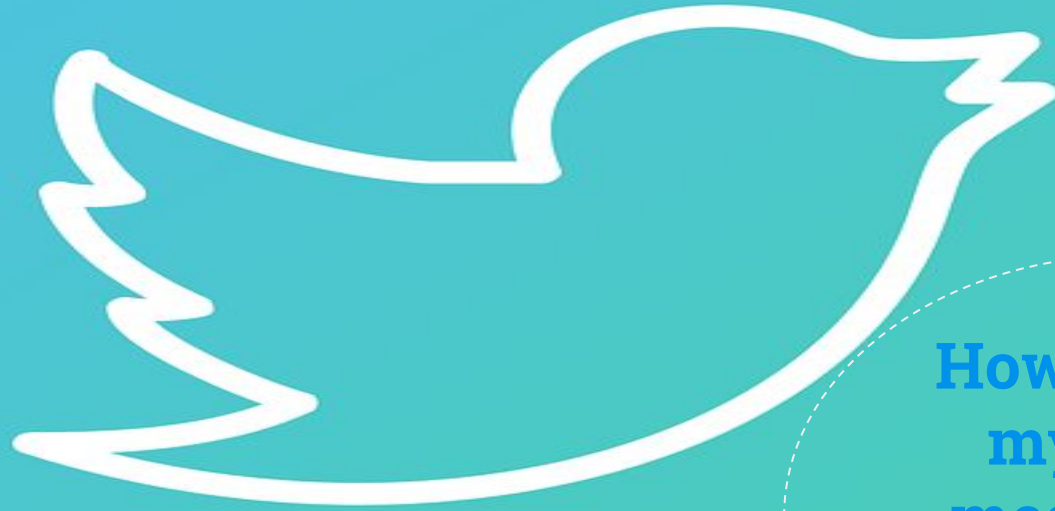
Supervised

- ⦿ Data with labels
- ⦿ Model is trained to find correlation
- ⦿ Prediction is based on correlation

Clustering

Clustering is an unsupervised task where data is grouped into clusters to find groups in data.





**How did I get
my social
media data?**

A look at data
mining.

The Program

Python

- ◎ The language I programmed my algorithm in.
- ◎ All the APIs ran on my Python instance.
- ◎ I chose Python because it has a large library of machine learning and data mining APIs.

Tweepy API

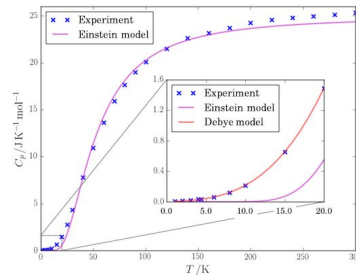
- ◎ The commands that Tweepy made to get data from Twitter.
- ◎ I downloaded this API, and my program could use the commands it gave.
- ◎ I used it to “stream” or get 500 tweets live.
- ◎ I then got the user from each tweet and got their follower count and tweet count.

The TensorFlow Environment

- Machine Learning requires a lot of preprocessing and algorithms.
- TensorFlow is an API that does it for you.
- I used the k-means method, which does clustering.

Matplotlib

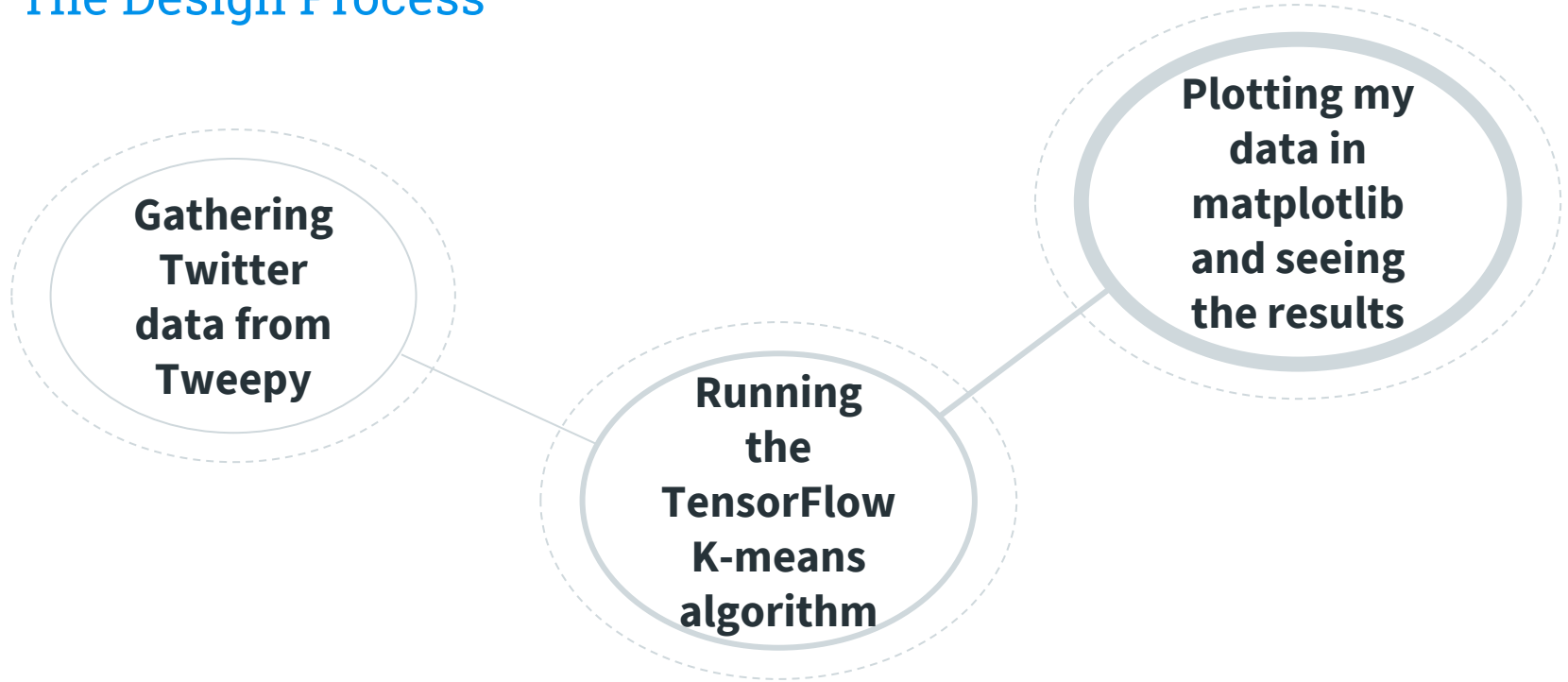
- Matplotlib is another API, designed to plot data.
- It can plot scatter plots, points, and more.
- It served as a tool to display my data.



Putting It All Together

- ◎ First, I gathered 500 tweets that referenced “north korea” and put them into a file.
- ◎ Then, I got the username from each tweet, and got that user’s follower and tweet count.
- ◎ Then, I ran my clustering algorithm.
- ◎ Finally, I plotted the data in each cluster.

The Design Process



Program

```
        print(a[i][0], a[i][1], i)
        if(i==499):
            break
        i+=1

def input_fn():
    return tf.train.limit_epochs(tf.convert_to_tensor(a, dtype=tf.float32),
num_epochs=1)

num_clusters = 3
kmeans = tf.contrib.factorization.KMeansClustering(num_clusters=num_clusters,
use_mini_batch=False)

num_iterations = 1000
previous_centers = None
for _ in range(num_iterations):
    kmeans.train(input_fn)
    cluster_centers = kmeans.cluster_centers()
    previous_centers = cluster_centers
print ('cluster centers:', cluster_centers)

cluster_indices = list(kmeans.predict_cluster_index(input_fn))
charar = np.array(['ro', 'bo', 'go'])
for p, point in enumerate(a):
    cluster_index = cluster_indices[p]
    center = cluster_centers[cluster_index]
    print(a[p][0], a[p][1], cluster_index)
    plt.plot(a[p][0], a[p][1], charar[cluster_index])
plt.show()
```

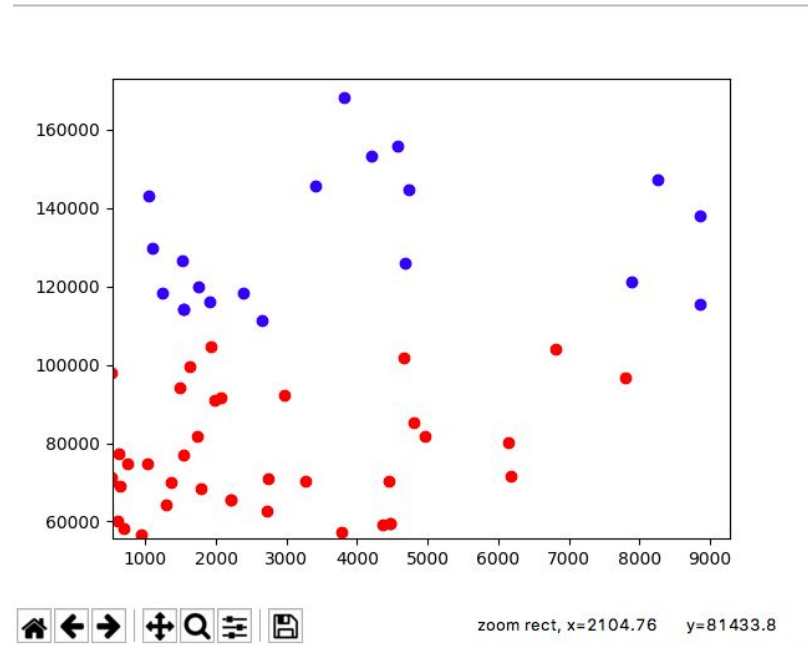
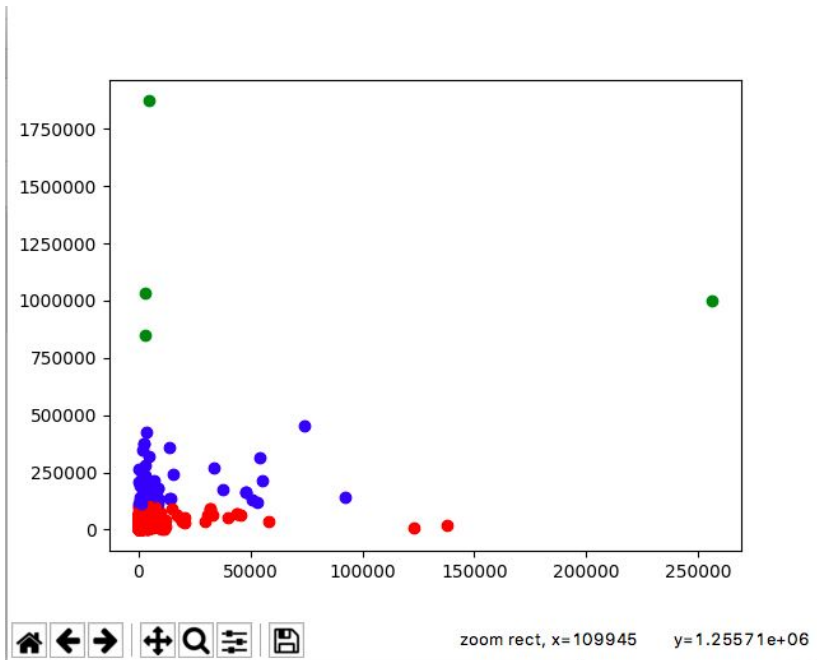
```
api = tweepy.API(auth)
i=0

class listener(StreamListener):
    def on_data(self, data):
        global i
        all_data = json.loads(data)
        username = all_data["user"]["screen_name"]
        user_data = api.get_user(username)
        a[i][0]=user_data.followers_count
        a[i][1]=user_data.statuses_count
        print(username, a[i][0], a[i][1])
        i += 1
        if(i>9):
            return False
        else:
            return True
    def on_error(self, status):
        print (status)

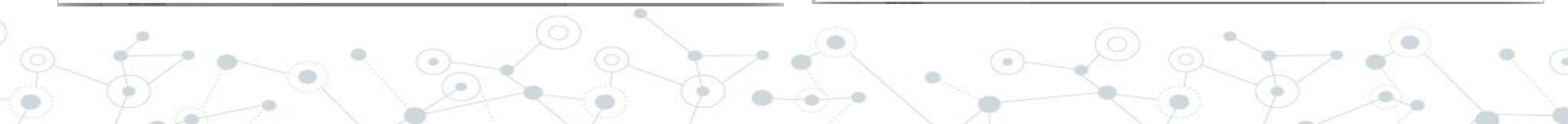
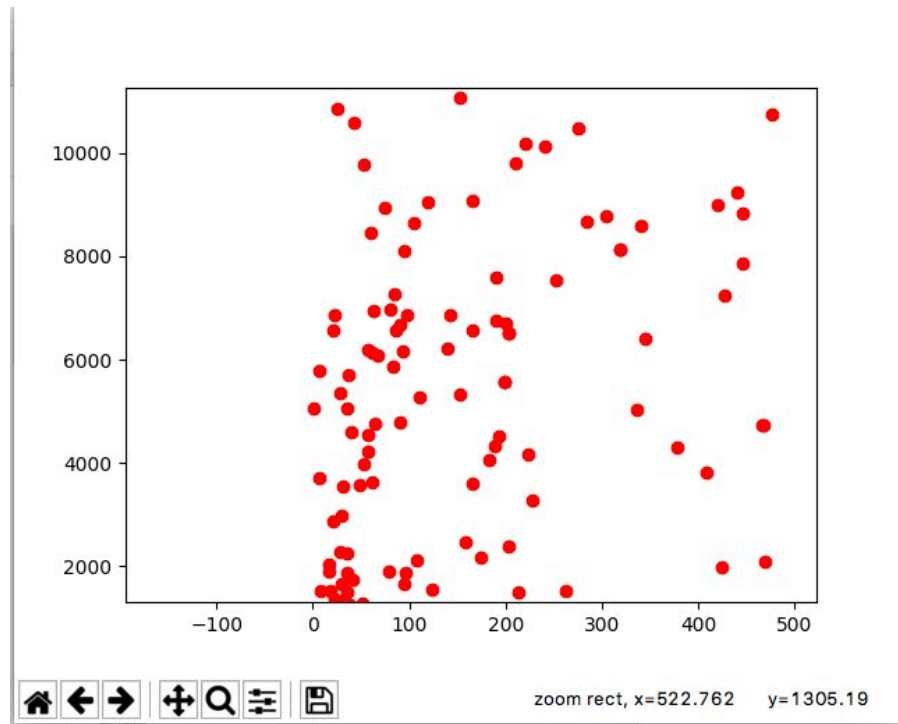
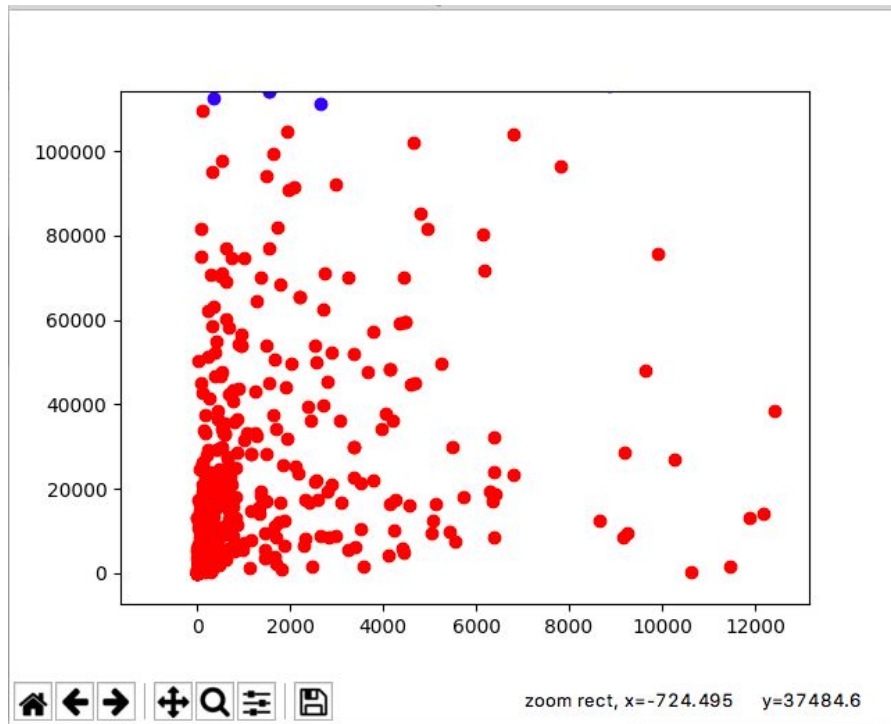
twitterStream = Stream(auth, listener())
twitterStream.filter(track=['north korea'])

def input_fn():
    return tf.train.limit_epochs(tf.convert_to_tensor(a, dtype=tf.float32),
num_epochs=1)
```

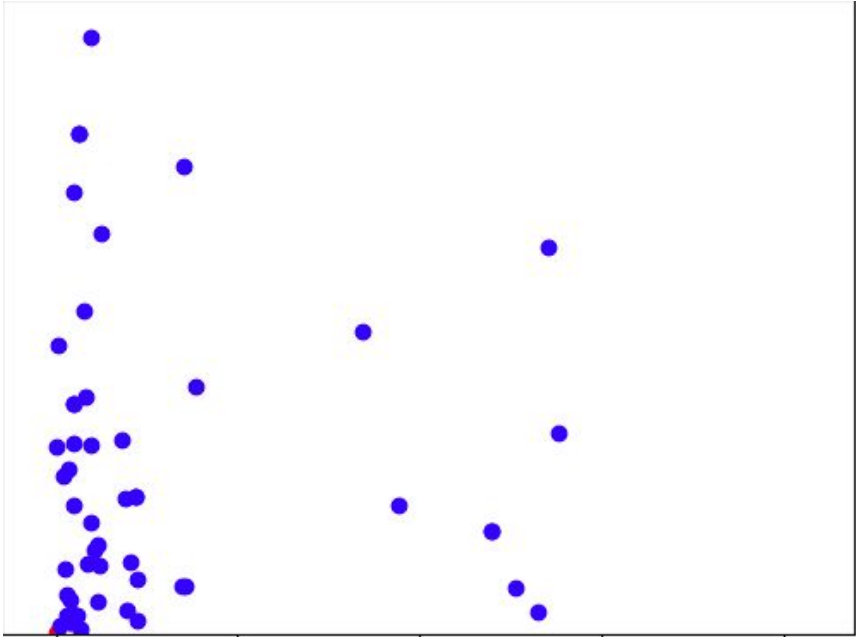
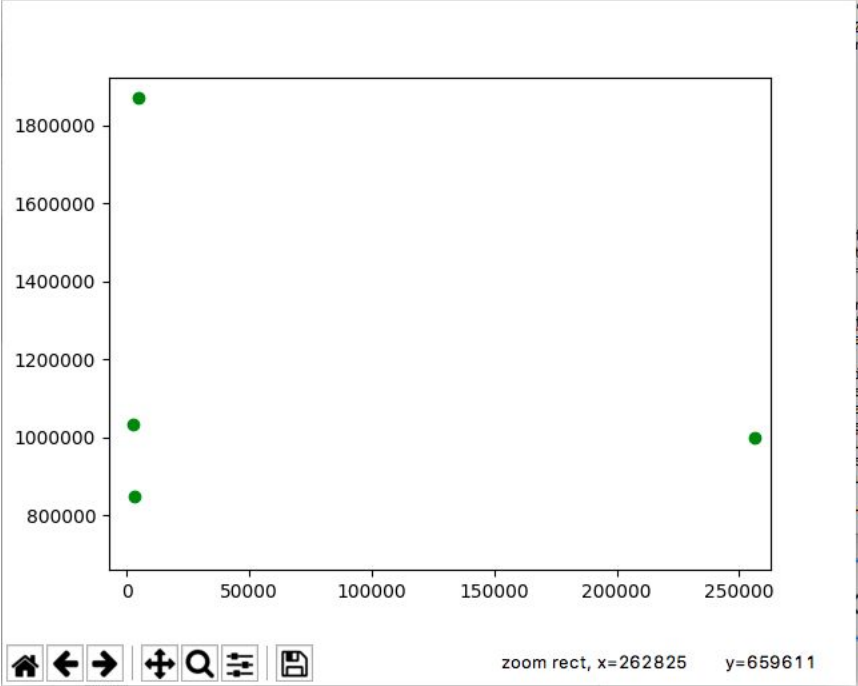
Graphs



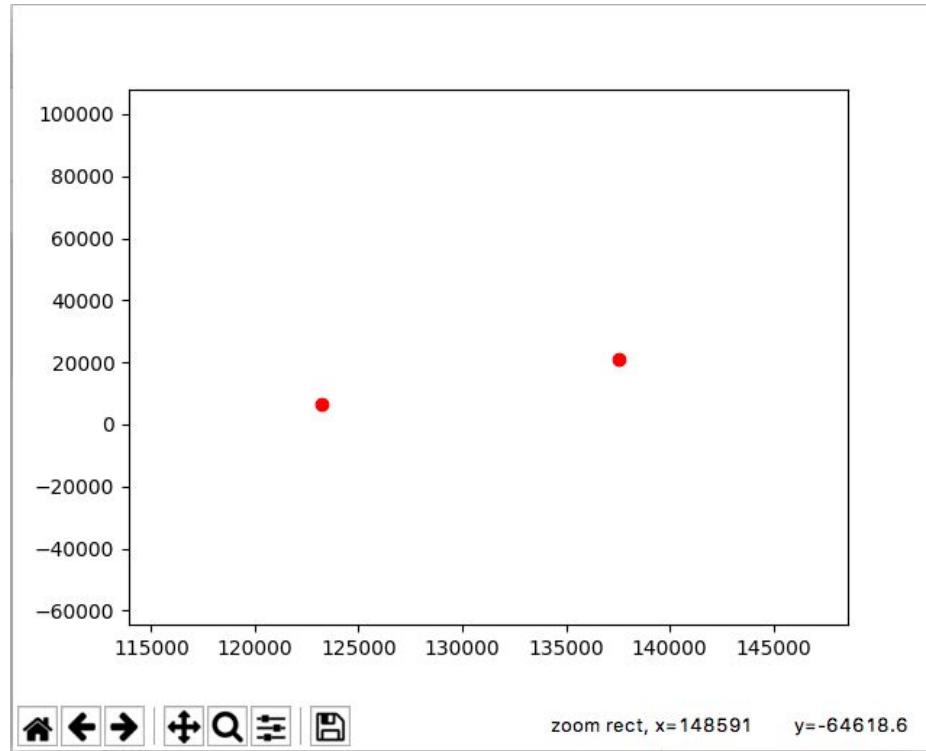
Graphs



Graphs



Graphs



What Next?



Geolocation

Geolocation can be used with any other variables to generate a map of users based on their location and another parameter.



Cloud

I could migrate my algorithm to a Hadoop cluster or other cloud computer, giving me more processing power to process more tweets.



Tweet Rate

Using tweet rate, I can cluster users and determine which communities are bots.



Neural Network

This clustering algorithm could be used as a preprocessor for a supervised algorithm, such as a neural net.



Text

I could use the spherical k-means algorithm to cluster tweets based on texts, and then do sentiment analysis or another algorithm.



Different Clustering

I could use hierarchical clustering to build clusters of users and their status based on followers.

Works Cited



- ⊙ <http://www.anc.ed.ac.uk/machine-learning/>
- ⊙ <https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/unsupervised.pdf>
- ⊙ <https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e>
- ⊙ <http://www.stat.columbia.edu/~madigan/W2025/notes/clustering.pdf>

Works Cited

- ① https://matplotlib.org/api/as_gen/matplotlib.pyplot.scatter.html
- ① http://docs.tweepy.org/en/latest/api.html#API.get_status
- ① <https://twitter.com/sid58352832>
- ① <https://docs.scipy.org/doc/numpy-1.14.0/reference/generated/numpy.random.uniform.html>

Works Cited

- © <https://marcobonzanini.com/2015/03/09/mining-twitter-data-with-python-part-2/>
- © <https://learningtensorflow.com/lesson6/>
- © <https://apps.twitter.com/app/15040379/keys>
- © <https://github.com/tensorflow/tensorflow/blob/r1.8/tensorflow/contrib/factorization/python/ops/kmeans.py>

- 
- 
- © <https://github.com/tensorflow/tensorflow/blob/r1.8/tensorflow/contrib/factorization/python/ops/kmeans.py>
 - © <https://stackoverflow.com/questions/49418325/use-tf-contrib-factorization-kmeansclustering>