

E6893 Big Data Analytics:

Tweets analysis and area safety prediction

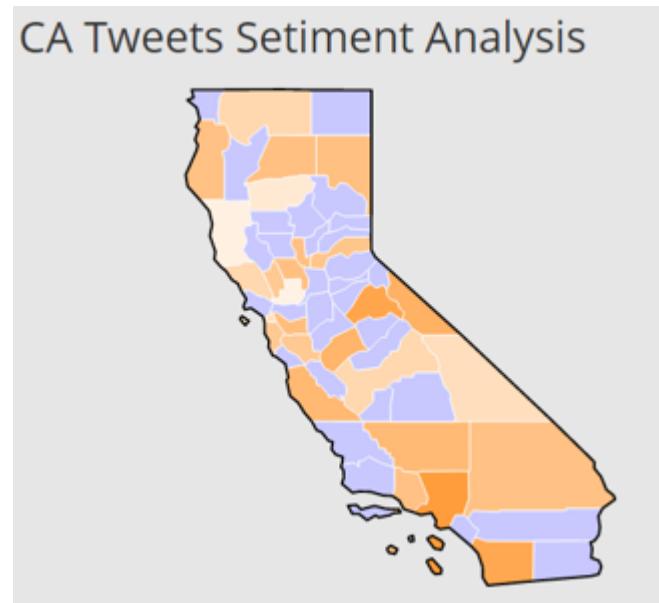
Team Members: Heng Ji, Wanhe Li, Long Long



December 15, 2016

Tweets has become a very efficient method to reflect people's moods. We will extract features from tweets and classify each tweets into positive and negative groups.

Use collected data to reflect the order of a specific area. Find the correlation between tweets and local safety. Accordingly, predict local safety factor and recommend police force.



Technologies Used

Details Settings Keys and Access Tokens Permissions



data mining

<https://www.ee.columbia.edu/~cylin/course/bigdata/>



APIs

Programmatic access to
read and write Twitter
data

[Learn about APIs](#)

Organization

Information about the organization or company associated with your application. This information is optional.



We mainly have two separate sets of data:

- City Open Public Safety Data

Collected by police departments of major cities including date, time, incident type, geo location...

 **OpenData**

LOS ANGELES OPEN DATA

Information, Insights, and Analysis from the City of Los Angeles



NYPD Complaint Data Historic  OFFICIAL Public Safety

[View Data](#)  [Download](#) [API](#) [Share](#) 

This dataset includes all valid felony, misdemeanor, and violation crimes reported to the New York City Police Department (NYPD) from 2006 to the end of last year (2015). For additional details, please see the attached data dictionary in the 'About' section.

Updated November 22, 2016
Data Provided by Police Department (NYPD)

What's in this Dataset?

Rows **5.1M** Columns **24**

Columns in this Dataset

Column Name	Description	Type
CMPLNT_NUM		Number # 
CMPLNT_FR_DT		Date & Time 
CMPLNT_FR_TM		Plain Text T 
CMPLNT_TO_DT		Date & Time 
CMPLNT_TO_TM		Plain Text T 
RPT_DT		Date & Time 

Table Preview

PREM_TYP_DESC	PARKS_NM	HADVELOPT	X_COORD_CD	Y_COORD_CD	Latitude	Longitude	Lat_Lon
RESIDENCE - APT. HOUSE			995,500	149,215	40.576237762	-73.959504022	(40.576237762°, -73.959504022°)
STREET			1,017,933	232,218	40.804004719	-73.878335461	(40.804004719°, -73.878335461°)
STREET			1,012,778	255,259	40.867263716	-73.896857767	(40.867263716°, -73.896857767°)
OTHER			1,054,851	181,458	40.664463748	-73.745517566	(40.664463748°, -73.745517566°)
RESIDENCE - APT. HOUSE			1,023,475	207,175	40.735245416	-73.858463451	(40.735245416°, -73.858463451°)
DEPARTMENT STORE			998,954	233,791	40.808374136	-73.946885823	(40.808374136°, -73.946885823°)
STREET			1,011,365	192,916	40.696153306	-73.902218102	(40.696153306°, -73.902218102°)
DEPARTMENT STORE			987,220	212,676	40.750430768	-73.989282176	(40.750430768°, -73.989282176°)
RESIDENCE-HOUSE			1,045,862	187,367	40.680749693	-73.777864583	(40.680749693°, -73.777864583°)

We mainly have two separate sets of data:

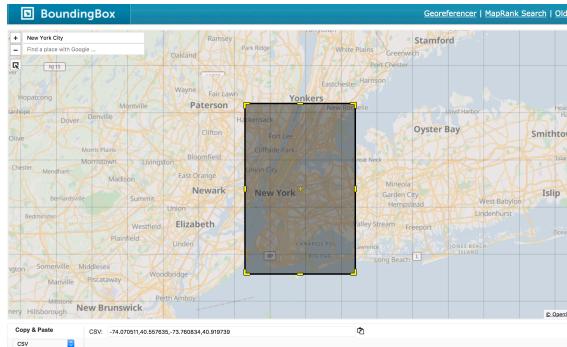
- Tweets with geo location
Used twitter API and tweepy to collect tweets sent out from selected cities

locations

A comma-separated list of longitude/latitude pairs specifying a set of bounding boxes to filter Tweets by. Only geolocated Tweets falling within the requested bounding boxes will be included—unlike the Search API, the user's location field is not used to filter tweets.

Each bounding box should be specified as a pair of longitude and latitude pairs, with the southwest corner of the bounding box coming first. For example:

Parameter value	Tracks Tweets from...
-122.75,36.8,-121.75,37.8	San Francisco
-74.40,-73.41	New York City
-122.75,36.8,-121.75,37.8,-74.40,-73.41	San Francisco OR New York City



```

1 {"created_at": "Fri Dec 09 04:54:37 +0000 2016", "id": "807086021187776512", "id_str": "807086021187776512", "text": "Anyone wanna play Xbox with me \ud83d\udc02", "source": "\ud03ca href=\"http://twitter.com/download/iphone\" rel=\"nofollow\"\ud03eTwitter for iPhone\ud03c/\u20ac\ud03e", "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": "2243061336", "id_str": "2243061336", "name": "cindy ..", "screen_name": "cindybigcat", "location": "Los Angeles", "url": "http://ask.fm/BigCatRT07", "description": "video game enthusiast \u20ac rooster teeth \u20ac achievement hunter \u20ac gallavich \u20ac photography enthusiast \u20ac wwa 09/13/14 \u20ac rowyso 11/14 \u20ac slfl 09/07/16 \u20ac", "protected": false, "verified": false, "followers_count": 13201, "friends_count": 2814, "listed_count": 18, "favourites_count": 27019, "statuses_count": 52873, "created_at": "Thu Dec 12 23:29:47 +0000 2013", "utc_offset": "-28800", "time_zone": "Pacific Time (US & Canada)", "geo_enabled": true, "lang": "en", "contributors_enabled": false, "is_translator": false, "profile_background_color": "C0DEED", "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_sidebar_border_color": "#00CED", "profile_sidebar_fill_color": "#DDEEF6", "profile_text_color": "#333333", "profile_use_background_image": true, "profile_image_url": "http://pbs.twimg.com/profile_images/765068306172686336/Pi7zHrLr_normal.jpg", "profile_image_url_https": "https://pbs.twimg.com/profile_images/765068306172686336/Pi7zHrLr_normal.jpg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/2243061336/1481014651", "default_profile": true, "default_profile_image": false, "following": null, "follow_request_sent": null, "notifications": null, "geo": null, "coordinates": null, "place": {"id": "3b77ca94bfc81fe", "url": "https://api.twitter.com/1.1/geo/id/3b77ca94bfc81fe.json", "place_type": "city", "name": "Los Angeles", "full_name": "Los Angeles, CA", "country_code": "US", "country": "United States", "bounding_box": {"type": "Polygon", "coordinates": [[[ -118.668404, 33.704538], [-118.668404, 34.337041], [-118.155409, 33.704538], [-118.155409, 34.337041], [-118.668404, 33.704538]]]}, "attributes": {}, "contributors": null, "is_quote_status": false, "retweet_count": 0, "favorite_count": 0, "entities": {"hashtags": [], "urls": []}, "user_mentions": [], "symbols": []}, "favorited": false, "retweeted": false, "filter_level": "low", "lang": "en", "timestamp_ms": "1481259277041"}  

2 {"created_at": "Fri Dec 09 04:54:37 +0000 2016", "id": "807086022290878464", "id_str": "807086022290878464", "text": "Man. I'm so happy by this L..", "source": "\ud03ca href=\"http://twitter.com/download/iphone\" rel=\"nofollow\"\ud03eTwitter for iPhone\ud03c/\u20ac\ud03e", "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": "2243061336", "id_str": "2243061336", "name": "cindy ..", "screen_name": "cindybigcat", "location": "Los Angeles", "url": "http://ask.fm/BigCatRT07", "description": "video game enthusiast \u20ac rooster teeth \u20ac achievement hunter \u20ac gallavich \u20ac photography enthusiast \u20ac wwa 09/13/14 \u20ac rowyso 11/14 \u20ac slfl 09/07/16 \u20ac", "protected": false, "verified": false, "followers_count": 13201, "friends_count": 2814, "listed_count": 18, "favourites_count": 27019, "statuses_count": 52873, "created_at": "Thu Dec 12 23:29:47 +0000 2013", "utc_offset": "-28800", "time_zone": "Pacific Time (US & Canada)", "geo_enabled": true, "lang": "en", "contributors_enabled": false, "is_translator": false, "profile_background_color": "C0DEED", "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_sidebar_border_color": "#00CED", "profile_sidebar_fill_color": "#DDEEF6", "profile_text_color": "#333333", "profile_use_background_image": true, "profile_image_url": "http://pbs.twimg.com/profile_images/765068306172686336/Pi7zHrLr_normal.jpg", "profile_image_url_https": "https://pbs.twimg.com/profile_images/765068306172686336/Pi7zHrLr_normal.jpg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/2243061336/1481014651", "default_profile": true, "default_profile_image": false, "following": null, "follow_request_sent": null, "notifications": null, "geo": null, "coordinates": null, "place": {"id": "3b77ca94bfc81fe", "url": "https://api.twitter.com/1.1/geo/id/3b77ca94bfc81fe.json", "place_type": "city", "name": "Los Angeles", "full_name": "Los Angeles, CA", "country_code": "US", "country": "United States", "bounding_box": {"type": "Polygon", "coordinates": [[[ -118.668404, 33.704538], [-118.668404, 34.337041], [-118.155409, 33.704538], [-118.155409, 34.337041], [-118.668404, 33.704538]]]}, "attributes": {}, "contributors": null, "is_quote_status": false, "retweet_count": 0, "favorite_count": 0, "entities": {"hashtags": [], "urls": []}, "user_mentions": [], "symbols": []}, "favorited": false, "retweeted": false, "filter_level": "low", "lang": "en", "timestamp_ms": "1481259277041"}}

```

1. Feature method: Using Tf-idf, Bigram, Unigram or Combination.
2. Classify algorithm: (a) If we use decision tree or Neural network, the result could be more accurate but will spend more time for calculating.
(b) If we apply binary classification method, we may get more error because not all datasets could be binary classified.
3. Correlation of tweets and crime rate seems not so obvious as we expected. As we use positive and negative to classify each sentence, a more accurate rating process might give good results.



1. Feature extraction

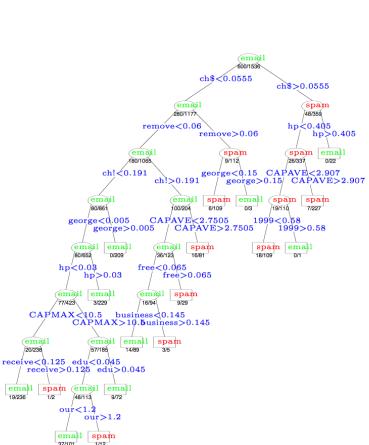
2. Spark decision tree algorithm

3. Prediction.

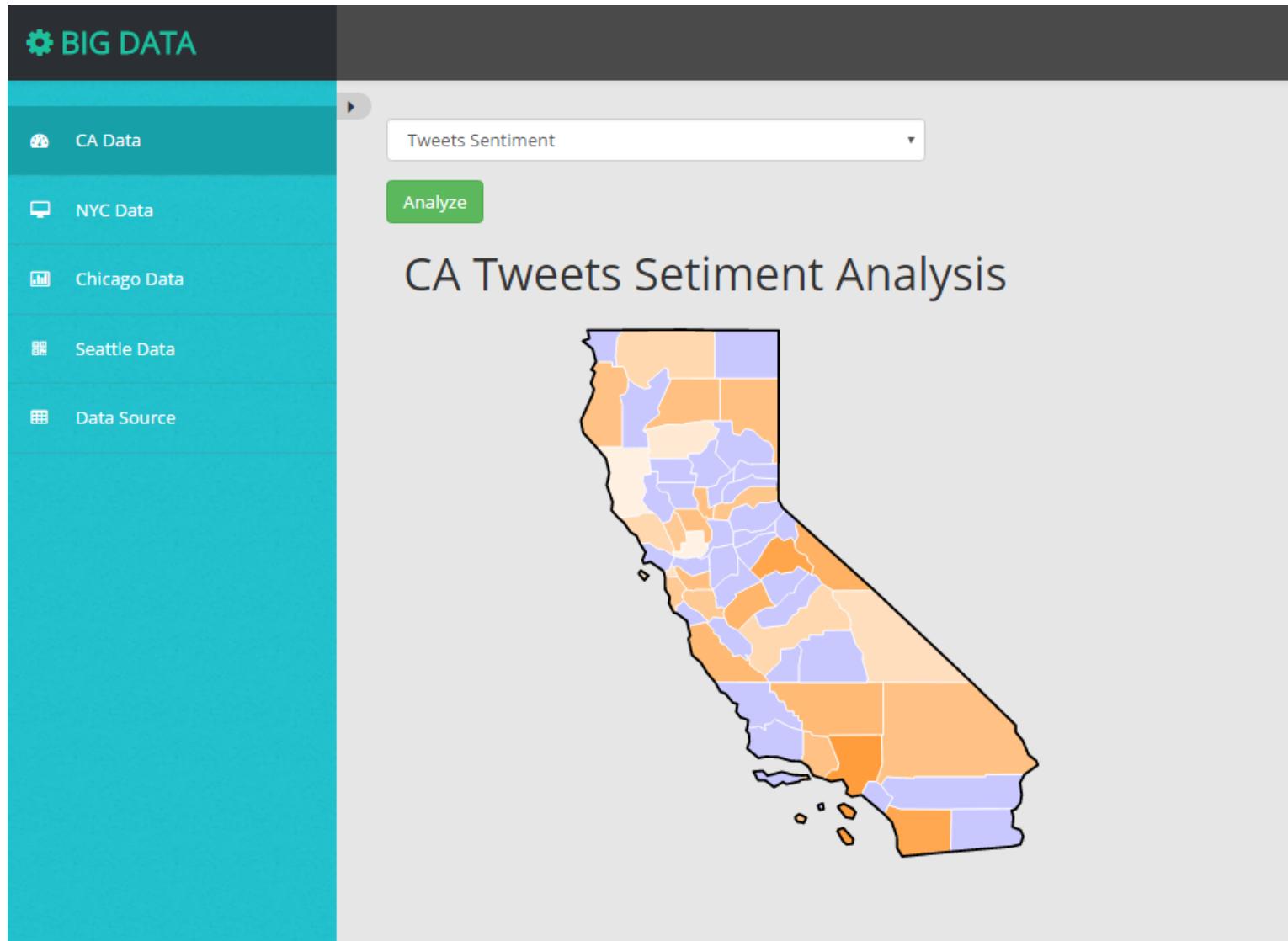
```

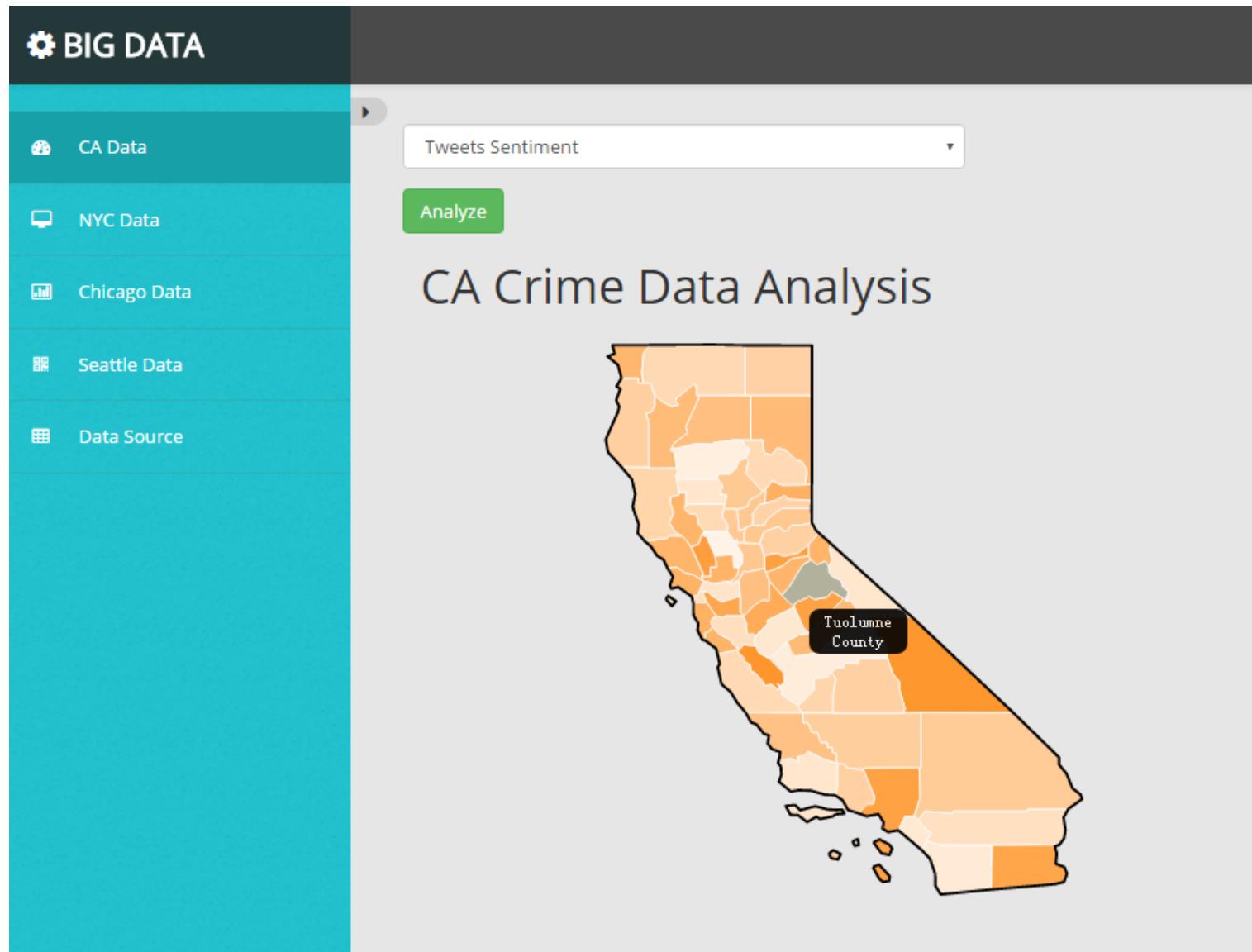
6      .....
7      import pandas as pd
8      import numpy as np
9      from sklearn.cross_validation import KFold
10     from sklearn.feature_extraction.text import CountVectorizer
11     from pyspark.mllib.tree import DecisionTree, DecisionTreeModel
12     from pyspark.mllib.util import MLUtils
13     data = pd.read_csv("Sentiment.csv")
14     test=pd.read_csv("geoTweets.csv")
15     text=test['text']
16     location=test['place_full_name']
17     traindata=data['SentimentText']
18     train=[]
19     label=[]
20     a=0
21     trainlabel=data['Sentiment']
22     for i in range(0,len(traindata)):
23         try :
24             temp=traindata[i].encode('utf-8')
25             train.append(temp.lower())
26             label.append(trainlabel[i])
27         except:
28             a=a+1
29
30
31     ngram_vectorizer = CountVectorizer(analyzer='word', ngram_range=(1, 1), min_df=1)
32     traindata = ngram_vectorizer.fit_transform(train)
33     model = DecisionTree.trainClassifier(traindata, numClasses=2, categoricalFeaturesInfo={}, impurity='gini', maxDepth=5, maxBins=32)
34     testdata=[]
35     testlocation=[]
36
37
38     for i in range(0,len(text)-1):
39         if type(text) is str:
40             print i
41             print text[i].lower()
42             testdata.append(text[i].lower())
43             testlocation.append(location[i])
44
45     testdata=ngram_vectorizer.transform(testdata)
46     predictions = model.predict(testdata.map(lambda x: x.features))
47

```



Bayonne, NJ	negative
Manhattan Beach, CA	neutral
Gardena, CA	negative
Tenafly, NJ	neutral
Fremont, CA	negative
Palos Heights, IL	neutral
Cambridge, MA	negative
Redwood City, CA	positive
Oakland, CA	neutral
Mountain View, CA	neutral
Hammond, IN	positive
Renton, WA	negative
Milpitas, CA	neutral
Hoboken, NJ	negative
Massachusetts, USA	neutral
Carson, CA	negative
Elmwood Park, IL	negative
Hackensack, NJ	neutral
Albany Park, Chicago	neutral
Downey, CA	negative
Santa Cruz, CA	negative
Oakland, CA	neutral
Franklin Park, IL	neutral
Carson, CA	negative
White Center, WA	neutral
Alhambra, CA	neutral
Newton, MA	neutral
Massachusetts, USA	neutral
New York, USA	positive





THANK YOU!