Data Mining and Pattern Recognition

To obtain status messages from Twitter and run some simple data mining procedures in order to draw some conclusions about currently popular topics.

Project 1

By:

Dhaval Ravindra Shah

Student ID : 802973719

CPSC 483

Spring, 2016

Professor: Mr. Kenytt Avery

Department of Computer Science

California State University, Fullerton

Feb 23, 2016

1. The programming environment, libraries, and tools used?

Ans) The **Environment** used in this project is: Linux – Ubuntu 64bit. I have used Visual Studio Code Version 0.10.8 as **tool** for developing. **Libraries** used for this project are "json, operator, urllib and oauth2".

2. The procedures used to obtain, analyze, and draw conclusions?

Ans) The First step for this project was to collect "DATA" from twitter. From the help of python file provided in courseara course. Here are the steps to download data:
The steps below will help you set up your twitter account to be able to access the live 1% stream.

1. Create a twitter account if you do not already have one.
2. Go to **https://dev.twitter.com/apps** and log in with your twitter credentials.
3. Click "Create New App"
4. Fill out the form and agree to the terms. Put in a dummy website if you don't have one you want to use.
5. On the next page, click the "API Keys" tab along the top, then scroll all the way down until you see the section "Your Access Token"
6. Click the button "Create My Access Token".
7. You will now copy four values into twitterstream.py. These values are your "API Key", your "API secret", your "Access token" and your "Access token secret". All four should now be visible on the API Keys page. (You may see "API Key" referred to as "Consumer key" in some places in the code or on the web; they are synonyms.) Open twitterstream.py and set the variables corresponding to the api key, api secret, access token, and access secret. You will see code like the below:
8. api_key = "<Enter api key>"
   api_secret = "<Enter api secret>"
   access_token_key = "<Enter your access token key here>"
   access_token_secret = "<Enter your access token secret here>"
9. Run the following and make sure you see data flowing and that no errors occur.

$ python twitterstream.py > output.txt

The twitter data file collected is 164.3 mb.

Second step:

The data collected is in json format. It need to be parse using ptyhon to read the data from it. Then, to parse the data in output.json, you want to apply the function json.loads to every line in the file.
This function will parse the json data and return a python data structure. There are many objects in the file. This object may be tweets and may not be tweets. Non- tweets has to filter out from the data. "Created_at" and"text" in on object means there is a valid tweet in it. Then text is compared in all the objects. If the match is found, counter is incremented for the valid tweets. AFFINN-111 file has all the sentiment score for the pre-define words. From this we get the score for each tweet. This score is used to determine the sentiment of the tweet. Scores are generated for each tweets and one tweet may have many hashtags so the sentiment of the hashtags re-present the total number of score of the tweet. For example: there are two tweets 1) I love python and 2) Python and the hash tag are #love, #python and #python respectively. So assume that python has sentiment score 2 and love has 1 so the #python will have total of sentiment score 4 and #love will have only 1. Then the average of the sentiments is taken and on the basis of this positive and negative is gathered.

Hash tags are collected from the entities which are in the object. Then here top 10 Hash tag is searched in frequency the occur after getting all the Hash tags from the objects. Dictionary is used to store the objects from the json file. To get the average sentiment for a hashtag and sorting them in descending order to find the most positive and most negative hash tags.

3. The answers to each of the questions posed above?

3.1) How large is the file?

Ans) File is 164.3 MB

3.2)How many objects are in the file?

Ans) Total number of objects are 47536

3.3) How many are tweets?

Ans) Total number of tweets are 37273

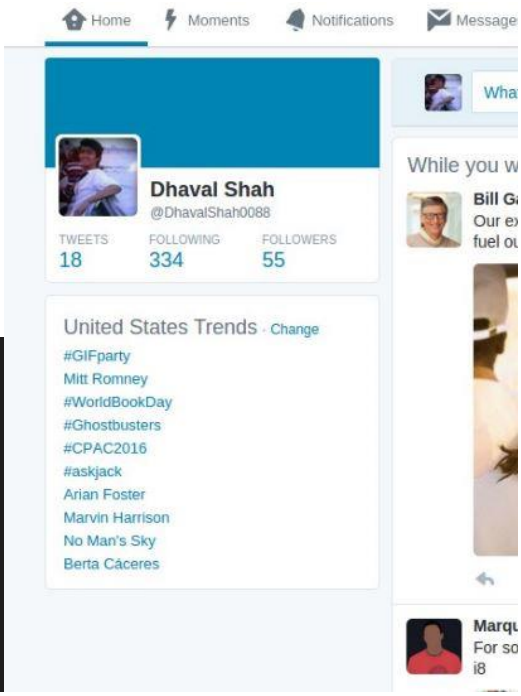
3.4) How many of the tweets have hashtags?

Ans) Total number of tweets have hash tag are 6032

3.5) What are the 10 most popular hashtags?

Ans) #KCA
#GIFparty
#iHeartAwards
#gameinsight
#KanhaiyaKumar
#PorTuSonrisaDamosTodoGravano
#androidgames
#BestFanArmy
#android
#WorldBookDay

3.6) How does your list compare with the trending hashtags shown on Twitter?
Ans).




```
OUTPUT
HashTag Name   -   Frequency    -    Score
#KCA -  463  -  0.082073
#GIFparty -  102  -  0.205882
#iHeartAwards -  81  -  0.024691
#gameinsight -  70  -  -0.028571
#KanhaiyaKumar -  63  -  0.301587
#PorTuSonrisaDamosTodoGravano -  54  -  -0.055556
#androidgames -  53  -  -0.037736
#BestFanArmy -  52  -  0.038462
#android -  50  -  0.020000
#WorldBookDay -  49  -  1.142857
```

There are 2 hastags which were common that is “GIFparty” and other one is “WorldBookDay”.

3.7) What is the average sentiment for each hashtag?

Ans).

```
OUTPUT
HashTag Name   -   Frequency    -    Score
#KCA -  463  -  0.082073
#GIFparty -  102  -  0.205882
#iHeartAwards -  81  -  0.024691
#gameinsight -  70  -  -0.028571
#KanhaiyaKumar -  63  -  0.301587
#PorTuSonrisaDamosTodoGravano -  54  -  -0.055556
#androidgames -  53  -  -0.037736
#BestFanArmy -  52  -  0.038462
#android -  50  -  0.020000
#WorldBookDay -  49  -  1.142857
```

The score in the above image represent the average sentiments for each tags. As there are many of hashtags.

3.8) What are the most positive and most negative of the popular hashtags?

Ans). Most positive hashtags from 10 popular tags are: WorldBookDay : 1.14285714286 and the most negative from 10 popular tags are: PorTuSonrisaDamosTodoGravano : -0.0555555555556

2. Source code, scripts, screenshots, configuration files, and any other artifacts associated with the project.

References:

https://class.coursera.org/datasci-002/assignment/view?assignment_id=3

https://dev.twitter.com/apps

https://github.com/wooorm/afinn-111