# **Report for Sentiment Analysis (Trustworthiness)**

**Assignment Description**

**Step 1- To Identify proper Twitter data set.**

**Step 2- Review/prepare data,**

**Step 3- Prepare dictionary for trustworthiness (just a first version),**

**Step 4- use the dictionary to analyses the data**

**Step 5- Algorithm and write code (Python or R).**

**Step 6-Review and send the analysis.**

**---------------------------------------------------------------------------------------------------------**

**Programming Language Used: Python (3.5)**

**IDE used: Pycharm Community edition**

**Python Libraries used: re, string, os, csv**

**Detailed Description of the steps followed:**

1. **proper data set - Load\_Data:**

**Function Used: Load\_data**

* I searched for twitter dataset that could be used for the following assignment on the links provided in the mail ,but didn’t find any relevant dataset and therefore used the sample dataset provided by you

**2. Prepare text for data analysis - Clean\_Data**

**Function Used : Clean \_data**

* The sample dataset could not be used directly for analysis and therefore Data Cleaning was done and the following steps were followed
  + Remove lines starting with RT (Retweet)
  + Remove words attached with @ along with the @
  + Remove punctuation (, ! ., ;, \)
  + Remove html links
  + Remove un-necessary spaces , " "
  + Lower case for all the text

**In the Course of preparing the assignment I first tried to encode the logic to find the polarity of the tweet and therefore followed the following steps**

**3.** **Load negative corpus**

**Function Used: Load\_data**

* List of Negative words

**4. Load Positive Corpus**

**Function Used: Load\_data**

* List of Positive words

**5.** **Run the data through Sentiment -Analysis - Analyse\_sentiment**

**Function Used: Analyse\_sentiment**

* Used the Clean Data and compare each tweet with Positive & Negative Corpus
* Should run through each cleansed tweet and come up with a 3 scores
* Positive score : words in tweet does not Match to any words in Negative dictionary
* Score : words in tweet does Match to any words in Negative dictionary
* Total Score = Positive Score - Negative Score

**6.Store the final Result with the Following columns in a CSV file (names.csv):**

**Function Used: Analyse\_sentiment**

* Tweet\_text
* Positive\_score
* Negative\_score
* neutral\_score
* Total\_score
* Polarity

**The polarity distribution looks like this.**

**After finding the polarity that was on the tweet level encoded the logic to find the Worthiness of a person at the user level**

**7. Load the Trust corpus :**

**Function Used: Load\_data**

* To get the Trust\_Corpus used the following : linkhttp://saifmohammad.com/WebPages/lexicons.html
* The data in the following link had to be mined and had to be made in a proper format to be used for data analysis
* The data was primarily separated line-by-line in a sublime text
* Then the data was moved in an excel file and the username was separated by spaces
* Then filtered the emotion of Trust ,and all the words related to trust was displayed
* Further checked for the association flag, if the flag was set to ‘’1’’ the word was related to trust and if the flag was set to ‘’0’’ the word was related to untrusty

**8. Load the untrust corpus**

**Function Used: Load\_data**

* The above steps were followed for the same

**9. Run the Data for Sentiment Analysis for Worthiness:**

**Function Used: analyse\_sentiment\_for\_trustworthiness**

* Compared the data from Trust & Untrust Corpus with the tweets
* Should run through each cleansed tweet and come up with a 2 scores
* Trust\_score : words in tweet does not Match to any words in Negative dictionary
* Untrust\_Score: words in tweet does Match to any words in Negative dictionary
* Total Score = Trust\_Score – Untrust\_Score
* Depending on the total score Worthiness of a person is decided
* The Results are saved in a csv file (Worthiness.csv)with following fields
  + Username
  + Tweet\_text
  + Trust\_score
  + Untrust\_Socre
  + Neutral\_score
  + Total\_score
  + Worthiness

**10. Finally the Username and Worthiness column are grouped together and stored in a dictionary with username as the key and worthiness options as the values to the corresponding user**

**Function Used: 1.group\_username\_worthiness**

**2. sentiment\_analysis\_score**

# Final Result in a dictionary

{'SteezKid': 'Untrustworthy', 'Enzo\_U': 'Untrustworthy', 'BbeSTACKIN': 'Untrustworthy', 'VitaFiasco': 'Could Not Determine', 'HeardBelleSay': 'Could Not Determine', 'TroyCrossley': 'Untrustworthy', 'BigBlackDave': 'Untrustworthy', 'bbccww383': 'Could Not Determine', 'sotirop\_evi': 'Untrustworthy', 'JukeMeng': 'Untrustworthy', 'DrJohnnyBlaze': 'Untrustworthy', 'jrowee': 'Untrustworthy', 'KBaieva': 'Untrustworthy', 'basiklee': 'Untrustworthy', 'Klumesickle': 'Untrustworthy', 'OVOadem': 'Untrustworthy', 'Jamie\_J19': 'Could Not Determine', 'KizzMcDutchie': 'Untrustworthy', 'BigZoKY': 'Untrustworthy', 'blackgirlsam': 'Untrustworthy', 'KidComeUp': 'Untrustworthy', 'dominickPipes': 'Untrustworthy', 'biggblakk92': 'Untrustworthy', 'therealAbdul\_': 'Untrustworthy', '9MichaelLewis': 'Untrustworthy', 'maryajuanasmoke': 'Untrustworthy', 'xXjungaXx': 'Untrustworthy', 'bettzamparelli': 'Untrustworthy', 'muftimenk': 'Untrustworthy', 'PurpNoShirt': 'Untrustworthy', 'eduardomporto': 'Untrustworthy', 'CopDaWholeThang': 'Untrustworthy', 'Kid\_Wavyy': 'Untrustworthy', 'Username': 'Could Not Determine', 'mellochamp': 'Could Not Determine', 'OhJeyy': 'Untrustworthy', 'MonTanA\_BaBy': 'Untrustworthy', 'DamnItGIO': 'Untrustworthy', 'ImRealTed': 'Untrustworthy', 'AnnieAreYouOkay': 'Untrustworthy', 'JJR\_Undeniable': 'Untrustworthy', 'TheSupplier': 'Untrustworthy', 'WhoButJosh': 'Untrustworthy', 'akiller82': 'Could Not Determine', 'MelloChamp': 'Untrustworthy', '\_JRabbit': 'Untrustworthy', 'BettZamparelli': 'Untrustworthy', 'SincerelyGCx': 'Could Not Determine', 'NbeFly': 'Untrustworthy', 'Basiklee': 'Untrustworthy'}