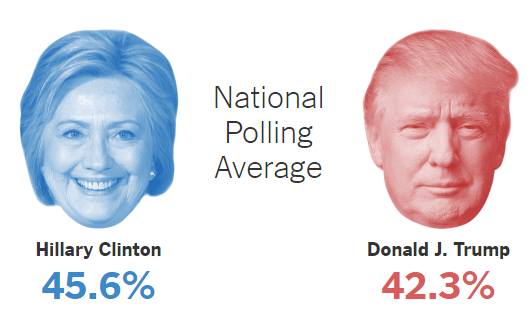
ASSIGMENT 3

# Group 5: 吳桑墨, 佳樂恩, 阮氏惠, 裴春全.



41.52%

58.48%

Group 5 Prediction Result

1. How to run our program?

1.

2

3

4

1. Tasks
2. Tweet extraction.
3. Build training set and testing data
4. Workflow - Naïve bayer Model.
5. Result.
6. **Twitter API and Tweet extraction**

A tweet is structured from 7 dimensions, which are “id” “content” “user” “reply to” …. And for this assignment, we only get the tweet “content” and “id”. Following is tweet extraction API function which provided by Twitter.

**public interface** TwitterApiService {  
 @GET(ApiConstants.***TWITTER\_HASHTAG\_SEARCH\_CODE*** )  
 **void** getTweetList(  
 @Header(**"Authorization"**) String authorization,  
 @Query(**"keywords"**) String hashtag,  
 @Query(**"count"**) **int** count,  
 Callback<TweetList> callback  
 );  
}

Where: **keywords** are the input keywords for enquiring the tweet. For example: “Clinton Trump”; “vote Trump”; “Vote Hillary” …. “**count**” is set to maximum 100. (means we can obtain 100 tweet per query). After extracting tweet list base on input keywords, we save the list into cvs format.

File file = **new** File(path, **"tweets.csv"**);  
**for** (Tweet tweet : **tweetsList**.**tweets**) {  
 s = s.append(tweet.**text** + **"\"\r\n"**);  
}  
writeToFile(s.toString(), file);

1. **Building Training data and Testing data**.

* For testing data, we obtained 8000 tweets from following keywords “Clinton Trump”, “vote trump” “vote Hillary” “2016 election”
* For training data, we manually collected 1000 tweets from testing data. Training tweets will be collected if it express obviously either “vote for Hillary” or “vote for trump”. These tweets then were labeled as “Positive Hillary” and “Positive Trump”
* For example:

|  |  |
| --- | --- |
| *PositiveHillary* | *RT @AdamParkhomenko: Trump's unfit. Get out that vote for Hillary and Democrats up and down the ballot and make generations to come proud hâ€¦* |
| *PositiveHillary* | *Less than a week until history is made. So proud to have voted for @HillaryClinton . Proud to get out the vote @ChelseaClinton* |
| *PositiveHillary* | *RT @abctweet100: Trump is Dangerous bc He is normalizing Racism. Is this what we want? For kids? STOP Him VOTE @HillaryClinton #USA https:/â€¦* |
| *PositiveTrump* | *@nathanTbernard @HillaryClinton \*Are you out of you're mind? Or just stupid to vote HRC* |
| *PositiveTrump* | *@mindyfinn @Evan\_McMullin This is who will be America's Miracle. Vote Trump!!! https://t.co/vKeTt1MFJv* |
| *PositiveTrump* | *@realDonaldTrump Face it, America's in a mess Hillary will be under pressure to Cover It up. America appoint a Clean Sweep Vote TrumpðŸ‘ðŸ»* |

**Work flow diagram**

1. Training Set

Label

Input

CSV File

Machine

Learning

Algorithm

Feature

Extractor

Features Vector

1. Testing Set

Input

Classifier model

(Naïve Bayes)

Feature

Extractor

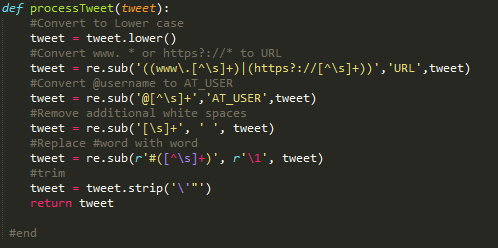
Features Vector

Label

Ordering training the classifier we construct the training set via manual ([Training\_set](data/New_Traning_set.csv)) over than 1000 records. The tweets can have some valuable info about it’s sentiment and some word without meaning. Therefore, we need to remove stop word and preprocess the tweets.

**Preprocess tweets** (processTweet() )

1. Convert the tweets to lower case.
2. Replace URLs with generic word URL.
3. Eliminate "@username" via regex matching or replace it with generic word AT\_USER.
4. Replace #hashtag with exact same word without the hash.
5. Remove punctuation at the start and ending of tweets.

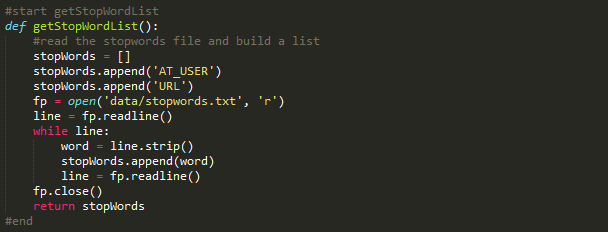
****

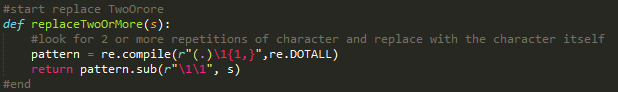
**Feature Vector (getFeatureVector() )**

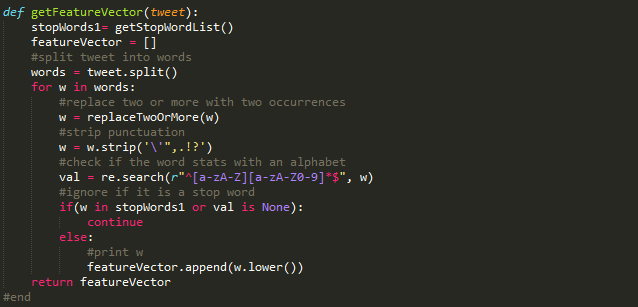
Feature vector is the most important concept in implementing a classifier. A good feature vector directly determines how successful your classifier will be. The feature vector is used to build a model which the classifier learns from the training data and further can be used to classify previously unseen data. In tweets, we can use the presence/absence of words that appear in tweet as feature. In the training data, consisting of vote Hillary Clinton and vote Donald Trump, we can split each tweet into words and add each word to the feature vector. Some of the words might not have any say in indicating the sentiment of a tweet and hence we can filter them out. Before adding the words to the feature vector, we need to preprocess them in order to filter, otherwise, the feature vector will explode.

**Filtering tweet words**

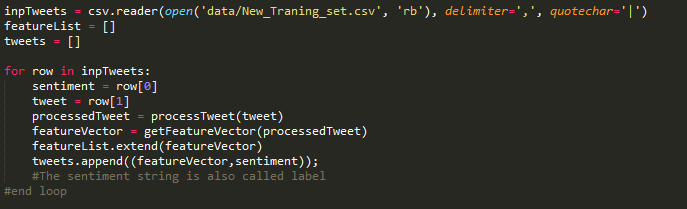
1. Remove stop word (is, a, the, with etc.) You can look in the detail of stop word at [Stopword.txt](data/stopwords.txt). (getStopWordList() )
2. Remove repeating letters (replaceTwoOrMore() )
3. Remove word which don’t start with an alphabet.

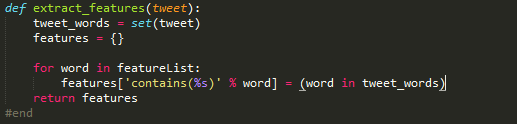






Supposing you want to know the detail of above source code, you can go to <filtering_tweet_word.py>. The main process is <read_tweet.py> ,including : Reading CSV file , Calling preprocess function, feature extraction (extract\_features() ). We use National Language ToolKit (NLTK) library for extract features. NLTK has a neat feature which enables to extract features as above in bulk for all the tweets and can be done using the below code snippet. The line of interest is "**nltk.classify.apply\_features(extract\_features, tweets)**" where you pass in the tweets variable to the extract\_features method.





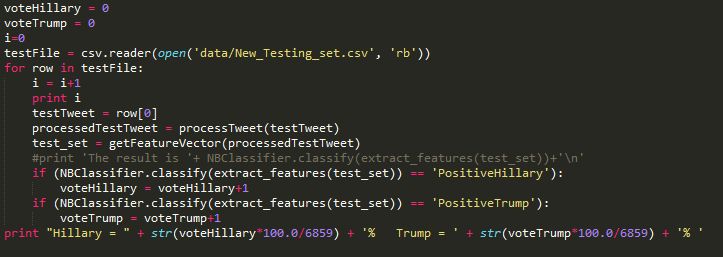
**Naive Bayes Classifier**

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. In this work, we use Naïve Bayes Classifier that is provided by NLTK library for creating the classifier model shown as the following figure.



**Prediction the election result.**

We use the tweet over than 6000 tweets for predict the election result ([New\_Testing\_Set.csv](data/New_Testing_set.csv)). The source code for prediction the election result is shown as the following figure:



**Who will win the presidency?**

**The election result from our system is:**

Hillary = 58.48% Trump = 41.52%