EBT Call Notes

* + Used Twitter API to get data
  + JSON output was converted to data frames
  + Appended multiple files to create one master file
  + Important Columns
    - ID
    - Tweet Body
    - Posted Time
    - Hashtags
    - Mentions

“Important Columns” – We use the Tweet Body to get clean data. We can get insight at Hashtag level – i.e. sentiment attached to a particular hashtag or Sentiment attached to mention.

Cleaning The data:

* + Removed hyperlinks
  + Removed special characters
  + Removed stop words
  + Removed punctuation, except comma and full stop
  + Converted to lower case

Getting data labels:

* + Matched Primary, Inclusion and Exclusion Words – Used dictionary given by Coke Team
  + Spam Flags (Rule Based) - Must Include both Primary and Inclusion words but if an exclusion word is present then tag it as Spam.
  + Matched Brands
  + Matched Trends
    - Used pattern matching to find Brands/Trends with more than one word.

Explanation of each of the Matched Columns and the rule used to get flags. The issue with limited query that lead to exclusion words being present can also be discussed.

**Spam Flag Prediction Using Labelled Data**

* Using train-test split
* Use labelled data as training set
* Build an ML model to predict spam on new data
* Logistic, Naïve Bayes Classification, SVM, NLTK Corpus (Universe of all the relevant words) – ***Will be done***
* Under consideration -
  + Markov Chain *(NOTES AT THE END OF THE DOCUMENT)*
  + XG Boost, Neural Networks

Topic Modelling:

* Find topics from the processed tweet text
* Find optimum K for no. of topics
* Challenges
  + The tweet size would further reduce when we remove mentions and hashtags.
  + Would not give robust results with less than 100-120 characters

Brand/Trend Detection

* + Find mentions of Brand/Trend in each tweet

Opinion Mining

* + Find the descriptor words against the brand/trend – ex. Delicious, Tangy, Bitter etc.
  + Extract polarity of the descriptor word and contextual sentiment for the Brand/trend

Word Cloud

* + Create word cloud of Top descriptor words
  + Further drill-down.
    - Top 10/15 tweets for a given descriptor word

Sentiment Scoring

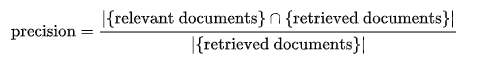
* + Sentiment Score @Tweet level
  + Sentiment Score @ Brand level @ Trend level
  + Hashtags vs sentiment score Analysis

Confusion Matrix:

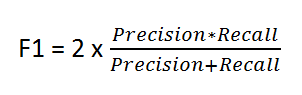
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Manual Labels* | | |  |
| **Labels** | **NOT SPAM** | **SPAM** | *Grand Total* |
| *Predicted Labels (Rule – based)* | **Not Spam** | **857** | **276** | *1133* |
| **Spam** | **248** | **627** | *875* |
|  | *Grand Total* | *1105* | *903* | *2008* |

|  |  |
| --- | --- |
| **True Positives** | **627** |
| **True Negatives** | **857** |
| **False Positives** | **248** |
| **False Negatives** | **276** |

|  |  |
| --- | --- |
| **Precision** | **0.72** |
| **Recall** | **0.69** |
| **F1 Score** | **0.7** |







**Using Markov Chains for Spam detection**

* Using Markov Chains, we can tell the probability of transition between two states.

*Assumption of a Markov is chain is that the future state depends only on the current state*

* Here each state is the word in a tweet ( we can also use the same logic with character in a word)
* *Eg: Possibility of 'f' coming after 'z' is almost zero but possibility of 'u' coming after 'q' is almost 100%*
  + We'll be able to find the probability of transition from one word/character to another word/character.
  + If the probability is very low (less than threshold), then we label is as spam

* Given we train the model on sufficiently large 'not spam' data, we'll be able to label spam (sentences that make no sense and words that are gibberish) with good accuracy

Eg:  The columns here are future states, rows are current states.

This example here shows probability of next day being 'Rainy' when today is 'Sunny' as 0.19

Machine generated alternative text:
States 
Sunny 
Rainy 
Snowy 
Sunny 
0.8 
0.2 
0.1+ 
Rainy 
0.19 
0.7 
0.2 
Snowy 
0.01 
0.1 
0.7 

With good training data, we'll create a large multi-dimensional transition matrix (similar to above matrix) where rows and columns are the corpus of words in the tweets

We'll get probabilities of next word (in columns) when the current word is 'x' ( in rows)

Then label spam based on threshold