Predicting Most Popular and Least Popular candidate in Elections 2016 using TWITTER data

AUTHOR: NEEL PATEL
NET ID-NP492454
Email ID: npatel3@aalbany.edu
Department: COMPUTER SCIENCE
CSI-531

1. ABSTRACT:

- The elections plays a major role in the development of country. So we decided to work on a project regarding the prediction of elections of 2016.
- When we started the project, Ted Cruz was also alive in the race of becoming a president. so we have also included Ted Cruz in this project.
- The 4 Candidates that we have selected are TED CRUZ, DONALD TRUMP, HILLARY CLINTON, BERNIE SANDERS. The reason why we selected these candidates is that all of these candidates are the top voted members for the presidential elections.
- Our major goal was to study each and every candidates with their positive comments and negative comments too. So in our project we have taken positive and negative tweets under consideration.
- First we collected the tweets for just 1 city. We manually labeled those tweets. Then we applied sentiment analysis and 4 algorithms for classification. After that we decided the accuracy of each and then we decided which algorithm to apply. At last we did association rules mining to predict the members in each of Republican and democratic. We had most popular and least popular candidates for each city.In total we had 15 cities all around the USA.

2. INTRODUCTION:

- Our Project is completely based on candidates of the Presidential Elections of 2016 for United States of America.
- We selected this project because as we know elections play a major role in developing youngsters of the country. Moreover the election is nowadays a hot topic of interest and gossips.
- When we analysed online data like twitter tweets, we found that we were getting a healthy amount of data on elections and election candidates for th 2016 elections.
- So this was our main motivation behind selecting this topic.
- As there were 4 top candidates in the elections of 2016, so we have selected following 4 candidates:
- **Ted Cruz**
- **Donald Trump**
- **Hillary Clinton**
- **Bernie Sanders**

- Our main goal was to cover all the regions of the United States and study on every detail for each candidate so that we can predict the winning and losing candidate.
- As we started our project collecting the tweets, we had to decide what locations to decide or what cities to select.
- At first we started with the center of USA. But we found that all the regions of USA is not fully involved in the twitter or social media.
- So we went on selecting such cities which were most involved in the twitter as well as the elections. We found it difficult while we were collecting tweets as we were not getting the relevant tweets about the elections.
- So we followed a proper structural approach to proceed and predict the data.

3. RELATED WORK:

- First we started off with collecting the tweets.
- We selected 15 cities of USA covering almost all the geographic locations and coastal regions of the USA.
- These were the 15 cities:
- New York City (NY)
- Los Angeles (CA)
- Washington DC. (MD)
- Atlanta (GEORGIA)
- Boston (MA)
- Jacksonville (FL)
- Illinois (Chicago)
- Houston(TX)
- Wisconsin(Milwaukee)

Predicting Most Popular and Least Popular candidate in Elections 2016 using TWITTER data

- Denver(Colorado)
- Little Rock (Arkansas)
- Louiseville(Kentucky)
- Boise(IDAHO)
- Seattle(WASHINGTON)
- Sioux Falls (South Dakota)

- So these cities cover the western, eastern and middle USA coastal areas.
- We collected 2000 tweets from each city for each candidates.
- So that makes:
- Ted cruz- 2000 tweets
- Donald Trump- 2000 tweets
- Hillary Clinton- 2000 tweets
- Bernie Sanders- 2000 tweets.

- For tweets collection we selected some attributes for collecting the elections relevant tweets regarding our project. Some of the attributes included the names of candidates. we decided the following query for collecting the tweets collectively.

- **QUERY:**

- Donald OR Trump OR
- Hillary OR Clinton OR
- Bernie OR Sanders OR
- Ted OR Cruz
- OR President
- OR Election 2016
- OR Republican OR Democrat
- OR Vote
- OR Campaign
- OR makeamericagreatagain
- OR feelthebern
- OR imwithher
- OR cruzcrew
- But after we collectively focused on all the candidates we decided to take each and every candidate and started collecting the tweets selecting attributes for each of the candidates.
- The new query generated was as follows:

- **HILLARY CLINTON:** Hillary OR Clinton OR President OR Election 2016 OR Democrat OR Vote OR Campaign OR imwithher OR politics OR debate OR poll OR delegate OR voters OR campaign OR caucus OR candidate OR Presidential OR HillaryClinton
- **DONALD TRUMP:** Donald OR Trump OR President OR Election 2016 OR Republican OR Vote OR Campaign OR makeamericagreatagain OR politics OR debate OR poll OR delegate OR voters OR campaign OR caucus OR candidate OR Presidential OR DonaldTrump
- **TED CRUZ:** Ted OR Cruz OR President OR Election 2016 OR Republican OR Vote OR Campaign OR cruzcrew OR cruzers OR politics OR debate OR poll OR delegate OR voters OR campaign OR caucus OR candidate OR Presidential OR TedCruz
- **BERNIE SANDERS:** Bernie OR Sanders OR President OR Election 2016 OR Democrat OR Vote OR Campaign OR feelthebern OR bernie's OR politics OR debate OR poll OR delegate OR voters OR campaign OR caucus OR candidate OR Presidential OR BernieSanders
- This was done at the beginning for only New York city. So we had over 8000 tweets of New York Cities.
- After collecting the tweets we found that there were many tweets with the emojis and images and videos.So we decided to remove all the extra characters and emojis from the tweets. This helped us in the unicode fetching of data and analysing the tweets.
- After removing the emojis we first manually labeled the tweets of New York Cities for Hillary Clinton. So we got a data of all the positive tweets and negative tweets of the candidate: Hillary Clinton.
- The tweets were classified as :

- Positive tweets: **label 1**
- Negative tweets: **label 0**
- While collecting these tweets we also found some tweets which were not related to the elections or which were indecisive.
- For example: a person tweets: “ hey i saw donald trump #trump “ OR “ Today the weather was as cold as Trump” #elections. So such tweets were discarded and kept as rest tweets. For this purpose we used API recall and rest API.

Predicting Most Popular and Least Popular candidate in Elections 2016 using TWITTER data

- The Tweets were then labeled according to their behaviour. After that we combined the positive and negative tweets in a single file and so we got the sample training data for testing the algorithms.

- **SVM(Support Vector Machine):**

- In machine learning, support vector machines (SVMs, also support vector networks[1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.
- So first we took two types of data for SVM:

1. Training dataset
2. Testing dataset.

- As mentioned earlier, the training data was decided and created by manually labelling the tweets of New York city.

- **INPUT:** Training data and testing data. Some of the stop words.
- **OUTPUT:**
- Number of positive tweets
- Number of Negative Tweets.

The data we got from the SVM algorithm was:

	Positive	Negative
Hillary	678	231
Ted	342	984
Bernie	589	424
Trump	460	434

TABLE: 1

- The above table is just the results of NYC for all the CANDIDATES.
- As we had the training data for a candidate we decided to test it on all other candidates. As shown in the stats:
- **HILLARY CLINTON** is with the maximum support.
- While Ted cruz is with the minimum support.
- Similarly , **TED CRUZ** is the candidate with most negative tweets.

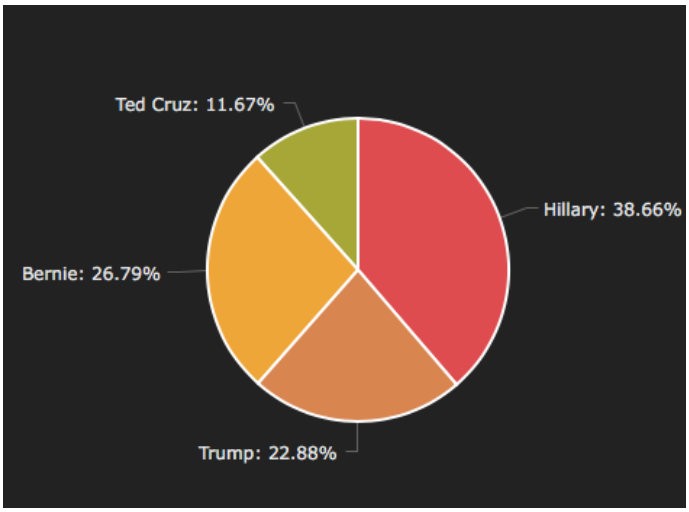


FIGURE 2 : SVM FOR NYC

- Above is the Pie chart for just New York City statistics of all the candidates with just the positive comments or tweets.

- **LR(LOGISTIC REGRESSION):**

- Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

- The logistic regression can be understood simply as finding the β parameters that best fit:

$y = 1$ if $\beta_0 + \beta_1 x + \epsilon > 0$

$y = 0$, otherwise

where ϵ is an error distributed by the standard logistic distribution.

Predicting Most Popular and Least Popular candidate in Elections 2016 using TWITTER data

- So, we decided to perform logistic regression using the same testing and training for each candidate.

- **INPUT:** Training data and testing data. Some of the stop words.
- **OUTPUT:**
- Number of positive tweets
- Number of Negative Tweets.

The data we got from the LR algorithm was:

	Positive	Negative
Hillary	540	314
Ted	247	1190
Bernie	530	424
Trump	404	480

TABLE: 3

- **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.
- Abstractly, naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector $\mathbf{x} = (x_1, \ldots, x_n)$ representing some n features (independent variables), it assigns to this instance probabilities
- $p(C_k|x_1, \ldots, x_n)$
- for each of K possible outcomes or classes.
- Naive bayes classifier was used as it is one of the efficient algorithms of classification oaf data mining and accuracy.
- Naive bayes is an efficient algorithm but it was time consuming as we had a lot of data in our training and testing datasets.

- **INPUT:** Training data and testing data. Some of the stop words.
- **OUTPUT:**
- Number of positive tweets
- Number of Negative Tweets.

The data we got from the Naive Bayes algorithm was:

	Positive	Negative
Hillary	533	314
Ted	289	889
Bernie	457	395
Trump	473	501

TABLE: 4

- **Decision Tree Algorithm :**

- Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called classification trees. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.
- We used Cross validation for Decision tree.
- The training and testing data were changed frequently by changing the data of tweets of New York city.

- **INPUT:** Training data and testing data. Some of the stop words.

Predicting Most Popular and Least Popular candidate in Elections 2016 using TWITTER data

- **OUTPUT:**
- Number of positive tweets
- Number of Negative Tweets.

The data we got from the Decision Tree algorithm was:

	Positive	Negative
Hillary	670	314
Ted	335	889
Bernie	540	395
Trump	393	501

TABLE: 5

- **Cross Validation on DataSets :**

- Let's say you have 10,000 rows and a binary target to train your classifier. The following are the basic validation schemes you can apply.
- 1. Randomly select 70% of the 10,000 rows for training your classifier. Of the remaining 30%, put aside, again choosing randomly, 50% for validation and 50% for Testing (& final model selection).
- So, You should have 3 files: Train set with 7000 rows, Validation set with 1500 rows and Test set with 1500 rows.
- You absolutely cannot use Test set in any way to modify your model training. You may not use validation cases in your training, however you may use it to tune hyper parameters of your model (for example selecting a K in K-nearest neighbours) and to get a feel for how well your model might perform on data that it has not been trained on.
- 2. In some circumstances, using a 70% random split for training and 30% for validation is also fine. For ex: In Kaggle competitions, I treat the Leaderboard as my Test set, so I do not need to split my training data into Train(70%), Validation(15%), & Test (15%).
- 1 & 2 are good schemes if you have plenty of data. In cases where you have very little to begin with, filtering out 30% of the data from training may be wasteful. This is where k-fold cross validation can help.
- 3. Let's say, you decide to use 5-fold CV instead of the regular Train, Validation & Test splitting. Then this is how the validation works.
- 10,000 rows are randomly split into 5-folds.
- First train your model using folds 1,2,3 & 4 i.e, you are training on 8000 rows and test your model fold#5. Make a note of the error on fold#5.
- Now, train your model using folds 2,3, 4 & 5 i.e, you are again training on 8000 rows and test your model fold#1. Make a note of the error on fold#1
- So, we decided to take 10 folds and swap the testing and training datasets to obtain high accuracy.

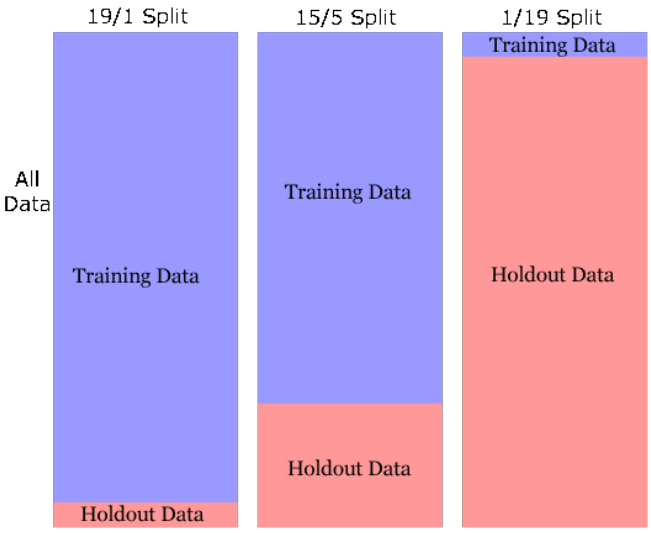


FIGURE 6: CROSS VALIDATION

- After applying the cross validation in decision tree algorithm we predicted this result:

- The training and testing data were changed frequently by changing the data of tweets of New York city.

- **INPUT:** Training data and testing data. Some of the stop words.

- **OUTPUT:**
- Number of positive tweets
- Number of Negative Tweets.

Predicting Most Popular and Least Popular candidate in Elections 2016 using TWITTER data

The data we got from the Decision Tree algorithm was:

	Positive	Negative
Hillary	706	314
Ted	389	889
Bernie	540	395
Trump	457	501

TABLE: 5

- **ASSOCIATION RULES MINING:**

- Association rule learning is a method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.

- **Support**
- The support value of X with respect to T is defined as the proportion of transactions in the database which contains the item-set X. In formula: $\text{supp}(X)$
- In the example database, the item-set {ID,TWEET,LABEL} has a support of 0.5 since it occurs in 20% of all transactions (1 out of 5 transactions). The argument of SUPP() is a set of preconditions, and thus becomes more restrictive as it grows (instead of more inclusive).

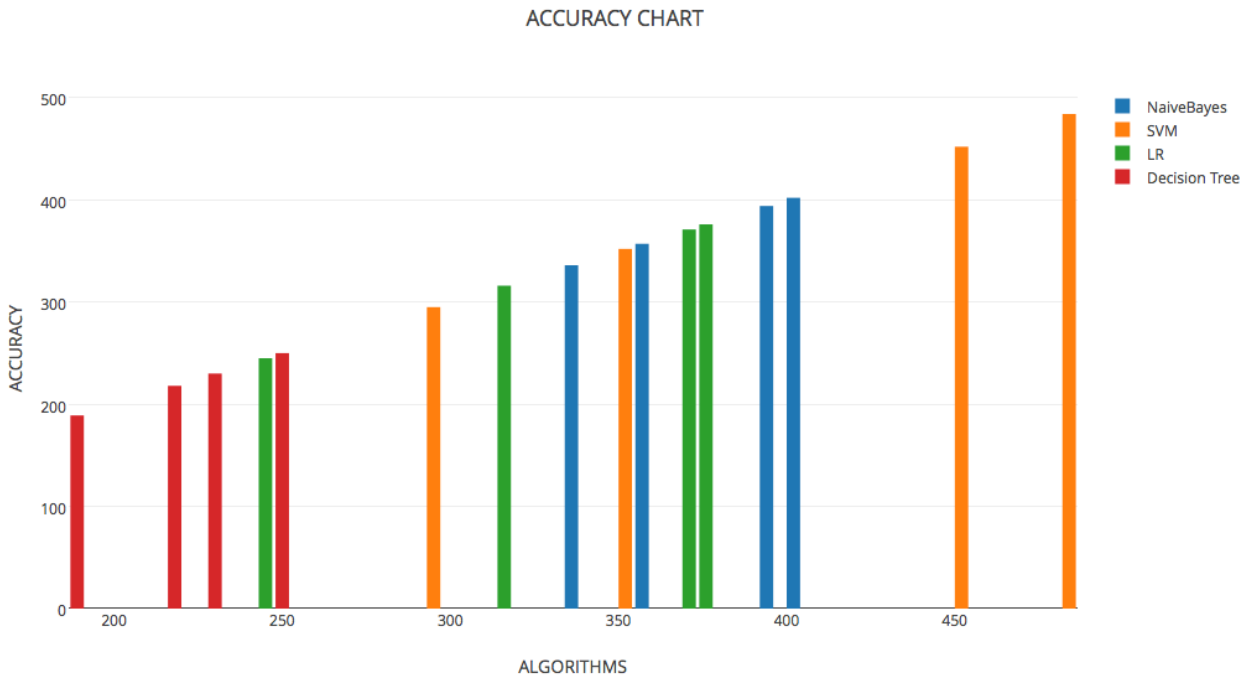
- **Confidence**
- The confidence value of a rule, $X \Rightarrow Y$, with respect to a set of transactions T, is the proportion of the transactions that contains X which also contains Y.
- Confidence is defined as:
- $\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y)/\text{supp}(X)$
.
- For example, the rule {bread,butter}= {milk}has a confidence of 1.0 in the database, which means that for 100% of the transactions containing butter and bread the rule is correct (100% of the times a customer buys butter and bread, milk is bought as well).
- Note that Supp(X union Y) means the support of the union of the items in X and Y. This is somewhat confusing since we normally think in terms of probabilities of events and not sets of items. We can rewrite $supp(X \cup Y)$ as the joint probability $P(E_X \cap E_Y)$, where E_X and E_Y are the events that a transaction contains itemset X or Y, respectively.
- Thus confidence can be interpreted as an estimate of the conditional probability $P(E_Y | E_X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

- Association rules mining is used by taking the datasets online data.
- We selected such a data where a survey was taken for the people who voted the top 4 of the candidates.
- We found the support and confidence according to that datasets.

- **SELECTION:**

- We selected the Support Vector Machine as it was most accurate of all the algorithms.
- As we were analysing the other algorithms we saw that this was the most efficient algorithm.
- Naive bayes was also efficient but it was consuming much time.
- Decision tree was the least accurate algorithm according to our datasets.

Predicting Most Popular and Least Popular candidate in Elections 2016 using TWITTER data



- As you can see the Support vector machine is with the highest accuracy.

- So we selected the training and testing data of tweets such that we took the analysis of New York City and predicted the results of NYC. Then we trained the model of SVM for selecting the training data for other cities with their testing data.
- So expanding this work we took training data of most accurate algorithm to predict other results.

4. SYSTEM DESIGN:

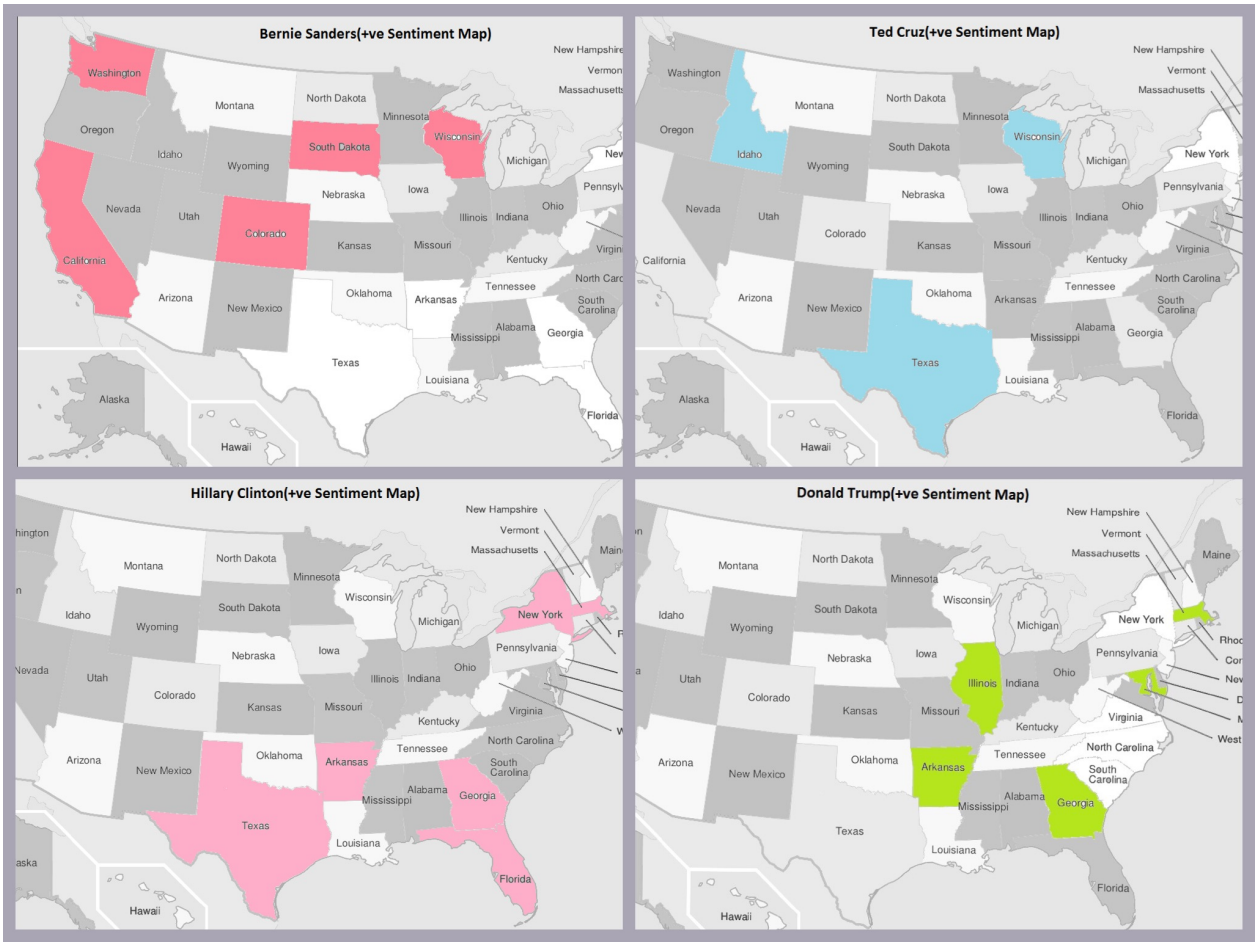
- **For implementing this project we used following tools and Algorithms:**

A. Support Vector Machine (All three contributed)
B. Logistic Regression (Julia Contributed)
C. Decision Tree (Nisarg and Neel Contributed)
D. Naive Bayes (Neel Contributed)
E. Association Rules Mining (Nisarg and Neel Contributed)
F. API Recall (Julia contributed)
G. SENTIMENT ANALYSIS (Nisarg and Julia contributed)
H. Cross Validation (Neel Contributed)
I. Labelling and Spliting (NISARG)
J. Emojis seperation (Julia contributed)
K. Data Visualization (Julia and Nisarg contributed)

- **Tools and Library used:**

A. matplotlib
B. plot.ly
C. amMap
D. google API
E. Twitter API
F. Python libraries

Predicting Most Popular and Least Popular candidate in Elections 2016 using TWITTER data



- **RESULTS:**
- **The predictions for each city were as follows:**

City	Most Popular	Least Popular
Atlanta	Hillary Clinton	Ted Cruz
Boise	Ted Cruz	Bernie Sanders
Boston	Donald Trump	Ted Cruz
Chicago	Donald Trump	Bernie Sanders
Denver	Bernie Sanders	Ted Cruz
Houston	Hillary Clinton	Donald Trump
JacksonVille	Hillary Clinton	Ted Cruz

City	Most Popular	Least Popular
LittleRock	Hillary Clinton	Bernie Sanders
LosAngeles	Bernie Sanders	Donald Trump
Louisville	Donald Trump	Bernie Sanders
Milwaukee	Ted Cruz	Donald Trump
NewYork	Hillary Clinton	Ted Cruz
Seatlle	Bernie Sanders	Hillary Clinton
Sioux Falls	Hillary Clinton	Ted Cruz
Washington DC	Donald Trump	Ted Cruz

Predicting Most Popular and Least Popular candidate in Elections 2016 using TWITTER data

5. REFERENCES:

- **https://www.kaggle.com/forums/f/15/kaggle-forum/t/4217/three-data-sets-and-k-fold-validation**
- **http://stackoverflow.com/questions/20587452/write-a-program-that-computes-mean-medium-maximum-and-standard-derivation**
- **https://github.com/asaini/Apriori/blob/master/apriori.py**
- **https://gist.github.com/mblondel/586753**
- **https://github.com/rmaestre/K-fold-cross-validation**