PREDICTING ELECTION RESULTS WITH SOCIAL MEDIA AND NEWS DATA

Under Guidance of-

Ms. Archana Purwar
Dr.Shikha Mehta

Undertaken By-

PRATIGYA AGARWAL 14103180
GAURAV SINGH 14103159
AMAN 14103168

CERTIFICATE

This is to certify that the work titled **" ELECTION PREDICTION**" submitted by "**PRATIGYA AGARWAL,GAURAV SINGH, AMAN** " in partial fulfilment for the award of degree of B.TECH IN COMPUTER SCIENCE of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor:

Name of Supervisor: Ms. Archana Purwar

Dr. Shikha Mehta

Date: 4-May-2017

ACKNOWLEDGEMENT

A lot of efforts have been taken in this project.This project consumed huge amount of work, research and dedication. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

We are highly indebted to Ms. Archana Purwar , Dr. Shikha Mehta for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

My thanks also go to my colleagues in developing the project and people who have willingly helped me out to the best their abilities.

Objective

This project aims to predict the result of elections with the use of data collected from twitter and news website.

Abstract

Micro-blogging provider Twitter has become very popular communication tools for Internet and Mobile users.People write about their life, share opinions on a variety of topics and discuss current political issues.

This huge amount of raw data can be used for industrial or studies purpose by organizing according to our requirement and processing.

Data is in the form of tweets which are opinions of people on different topics which lie in political category .

We present the results of machine learning algorithms for classifying the sentiment of Twitter messages using a novel feature vector. Our training data consists of publicly available tweets obtained through automated means. We show that machine learning algorithms (Naive Bayes, Maximum Entropy, and SVM) can achieve competitive accuracy when trained using feature vector and the publicly available dataset. Our project also describes the preprocessing steps of the dataset needed in order to achieve high accuracy. The main contribution is the novel feature vector of weighted unigrams used to train the machine learning classifiers.

DIVISION OF WORK AMONGST THE GROUP MEMBERS

The project has been built with equal contribution from all the group members and a lot of time and effort has been put into collaborating common slots to built the project collectively as a group. Each and every member has contributed in the implementation of every feature in this project.

On a broad basis,

Pratigya Agarwal -14103180

- **Data collection of twitter and news data using scraping.**
- **Data cleaning and pre-processing and algorithms (Naive Bayes,Support VectorMachine,Maximum entropy and Modified polarity lexicon method) implementation.**

Gaurav Singh -14103159

- **Front end website designing.**
- **Resultant pie charts and data dictionary of positive, negative words from the tweet data.**
- **Preparation of training dataset for sentiment analysis.**

Aman -14103168

- **Data Dictionary preparation of positive and negative tweets.**
- **Preparation of training dataset for sentiment analysis and data loading in hive table.**

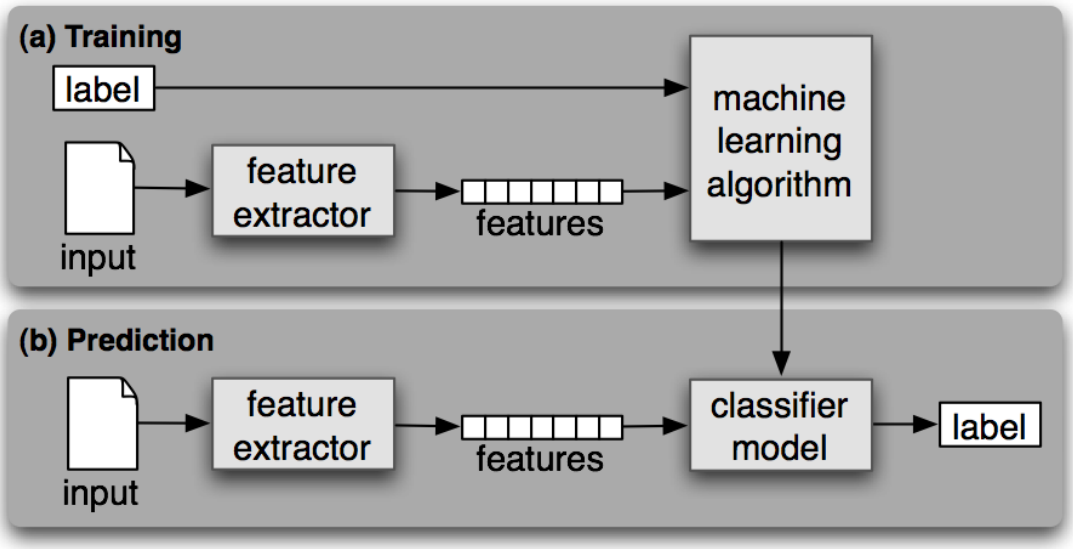
BACKGROUND STUDY

- **In order to build a sentiment analyzer, first we need to equip ourselves with the right tools and methods. Machine learning is one such tool where people have developed various methods to classify. Classifiers may or may not need training data. In particular, we will deal with the following machine learning classifiers, namely, Naive Bayes Classifier, Maximum Entropy Classifier and Support Vector Machines. All of these classifiers require training data and hence these methods fall under the category of supervised classification.**
- **After reading numerous research papers related to predictive analytics and text mining, we found out that**
- **A large amount of data generated today is in unstructured form.**
- **To extract useful information from it,we have to perform text mining techniques.**
- **In text mining, we have to clean the data and save it in a particular format so that information can be obtained from it.**
- **We learn about various classification algorithms like naive bayes,svm and maximum entropy classification.**
- **Also we learnt about various data collection technologies like selenium ,beautiful soup for web scraping.**

- **We learnt to use and work on anaconda framework using pandas and sklearn libraries.**
- **We learn to connect hive with python and running map reduce jobs from python script.**
- **We learn how to use pandas data frame technology to visualize the results obtained after implementing algorithms.**
- **We learn to apply modified polarity lexicon method to predict the results and percentage of probability of winning of a party.**

FINDINGS

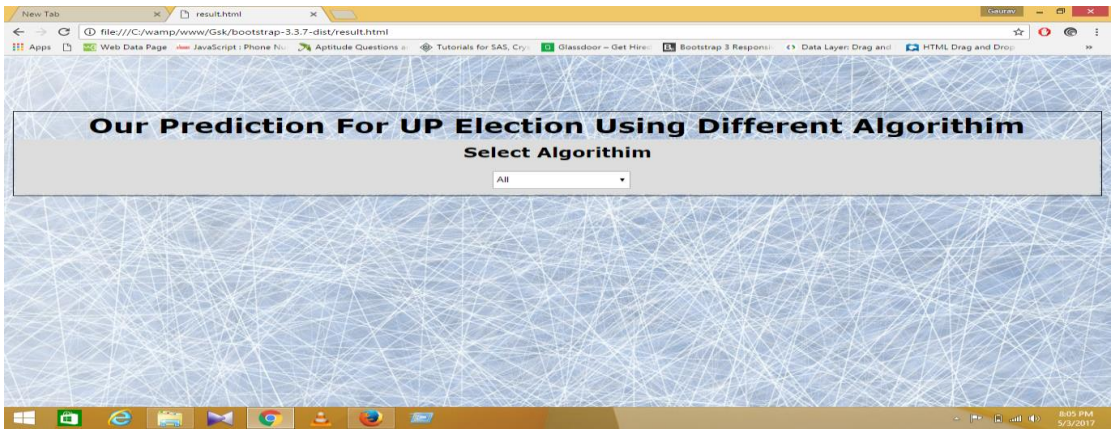
Since social media is usually formed and constructed by daily and continuous communication between participants, we have decided to investigate its potential in predicting real-world outcome. That is, using the information posted by social media participants to detect their sentiment or opinion about the different 2017 UP political parties. The sentiment used to predict the outcome of the election.

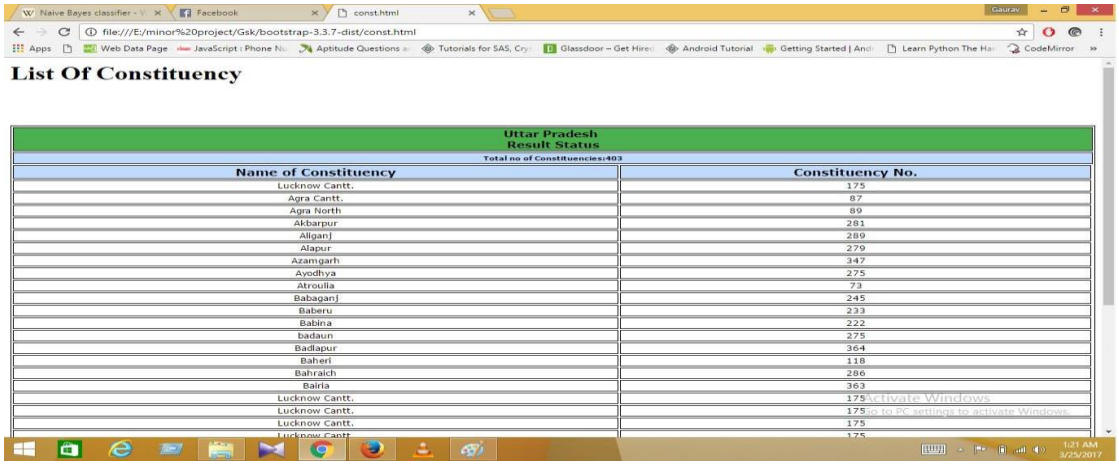


DESIGNING

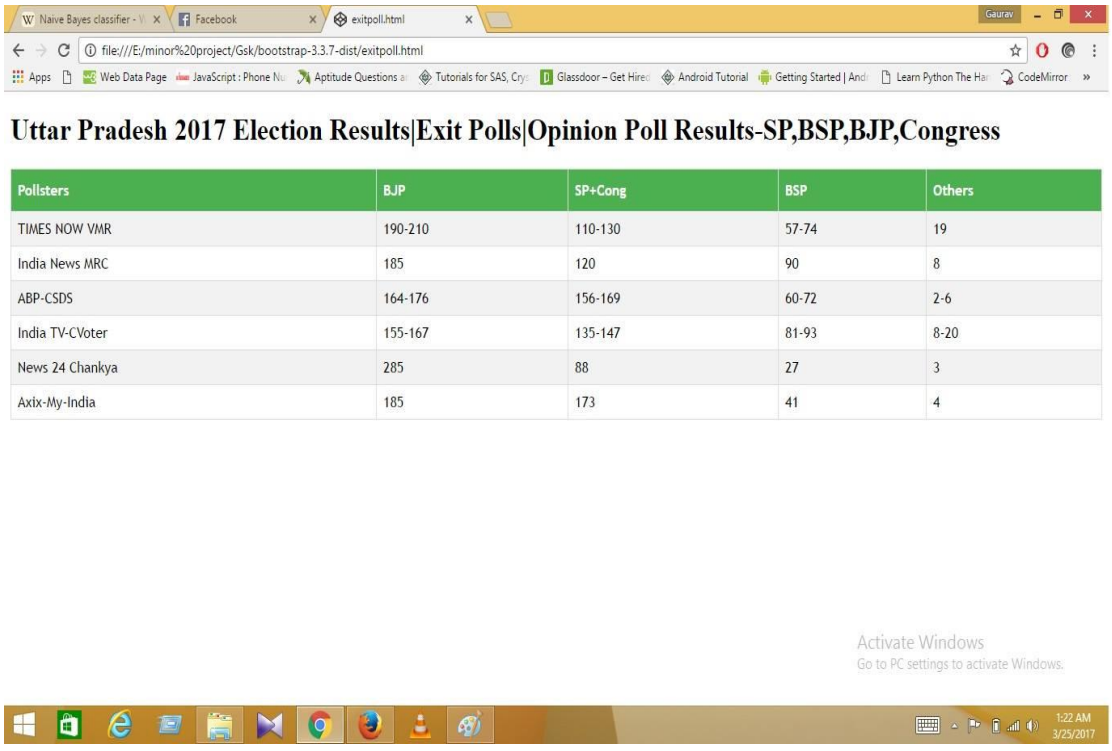
Website to display final result of our prediction.

FRONTEND OF THE WEBSITE



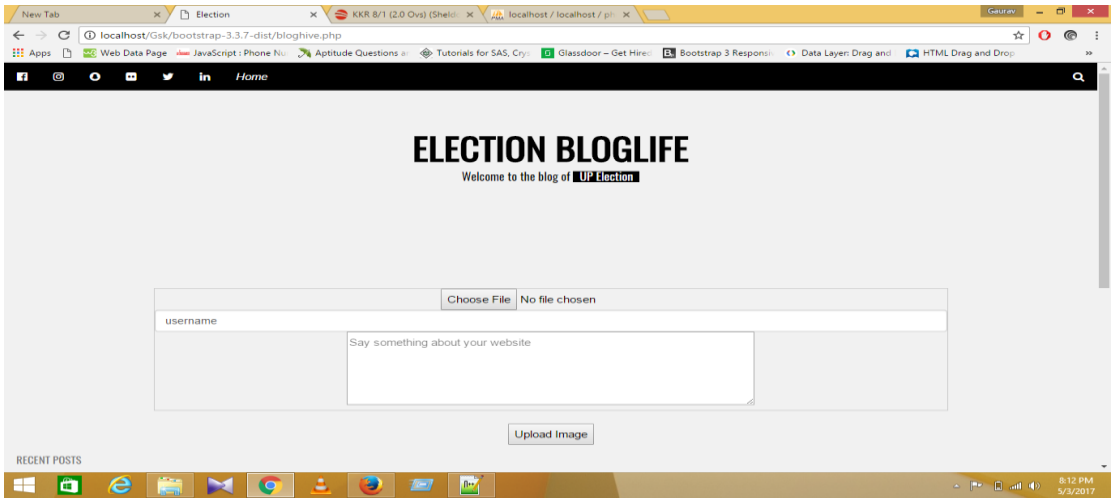


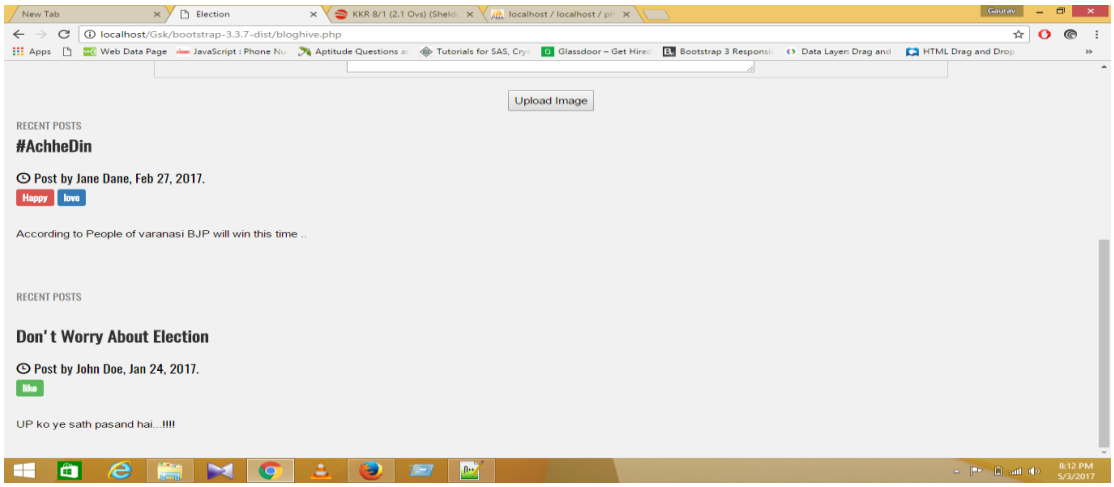
EXIT POLLS RESULTS



BLOG

- Blog is provided where users can share their opinions to others.





Step by step implementation

Implementation Details

We will be using Python (3.x) along with the Natural Language Toolkit (nltk) and libsvm libraries to implement the classifiers.

- **Data collection-**

  Data is collected by web scraping. A technique used to collect information from web pages. We have collected data from 1st October until 8th march for our analysis.

Website: Twitter

News Website: Indian express

Language used: Python

Selenium Web driver for scrolling through the web page

Libraries used:

- of twitter
- **Beautiful soup** for collecting data from multiple pages in next link.

- **Format for data storage:**

  Text file storage for tweets and news website data.

CODE:-

1.Data Cleaning:-

One of the first steps in working with text data is to pre-process it. It is an essential step before the data is ready for analysis. Majority of available text data is highly unstructured and noisy in nature – to achieve better insights or to build better algorithms, it is necessary to play with clean data. For example, social media data is highly unstructured – it is an informal communication – typos, bad grammar, usage of slang, presence of unwanted content like stop-words , special characters etc. are the usual suspects.

Preprocess tweets

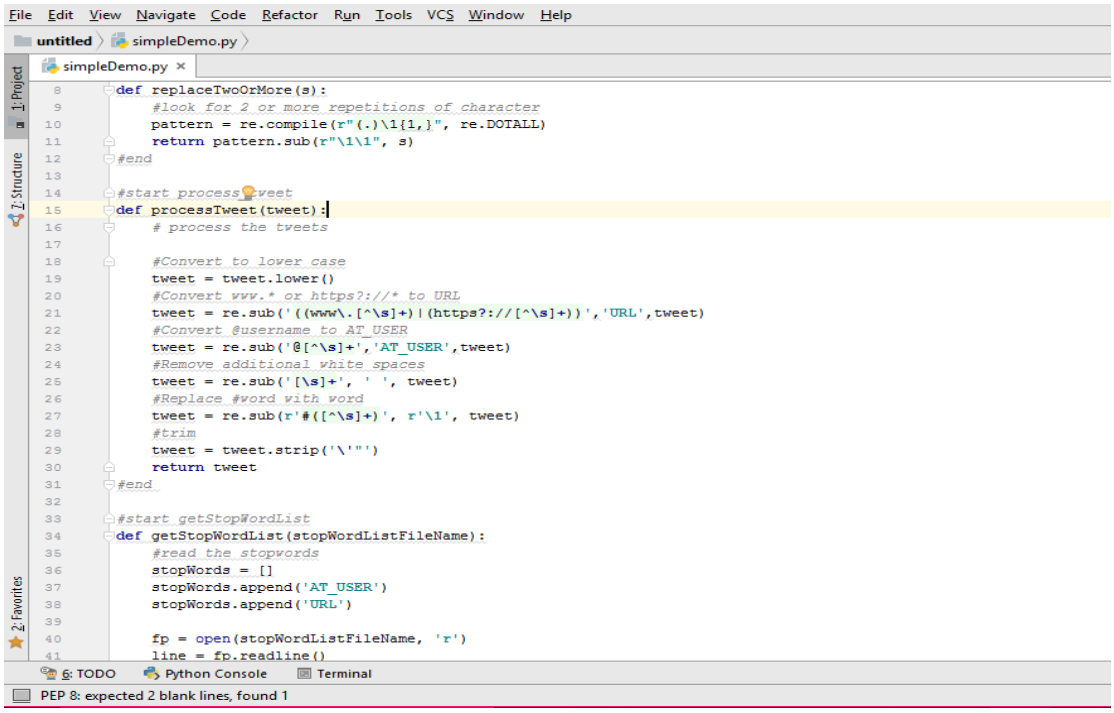
- Lower Case - Convert the tweets to lower case.
- URLs - I don't intend to follow the short urls and determine the content of the site, so we can eliminate all of these URLs via regular expression matching or replace with generic word URL.

- @username - we can eliminate "@username" via regex matching or replace it with generic word AT_USER.
- #hashtag - hash tags can give us some useful information, so it is useful to replace them with the exact same word without the hash. E.g. #BJP replaced with 'BJP'.
- Punctuations and additional white spaces - remove punctuation at the start and ending of the tweets. It is also helpful to replace multiple whitespaces with a single whitespace.

Language used: Python

Library used: Regex

Code:-



[Reference:-www.analyticsvidhya.com/blog/2014/11/text-data-cleaning-steps-python]

3.Data loading into Hadoop file storage system:-

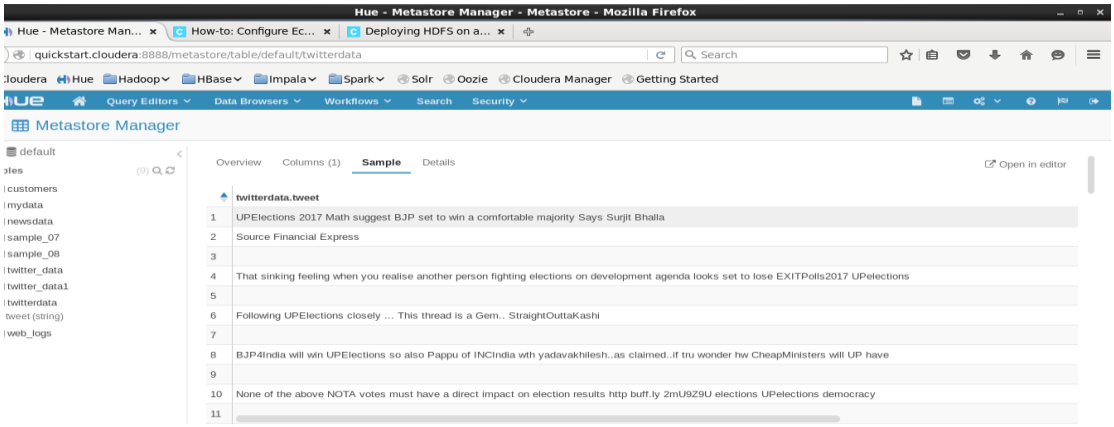
Using put command load text data files into file storage.

Command:-hdfs dfs -put textfilename.txt

4.Data loading into hive table:-

For loading data from hdfs into hive, we have used external table named twitterdata and newsdata1 using load data statement.

Snapshot:-



5.Data dictionary of positive and negative words :-

We have prepared 4 separate text files containing positive and negative words . For opinion mining, data dictionary is needed.

[Reference:-Kabir Ismail Umar Data mining for social media analysis using twitter.]

6.Feature Vector

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.

Similarly, in tweets, we can use the presence/absence of words that appear in tweet as features. In the training data, consisting of positive, negative and neutral tweets, 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. Adding individual (single) words to the feature vector is referred to as 'unigrams' approach.

7.Filtering tweet words (for feature vector):

- Stop words - a, is, the, with etc. The full list of stop words can be found at Stop Word List. These words don't indicate any sentiment and can be removed.
- Repeating letters - if you look at the tweets, sometimes people repeat letters to stress the emotion. E.g. hunggrryyy, huuuuuuungry for 'hungry'. We can look for 2 or more repetitive letters in words and replace them by 2 of the same.
- Punctuation - we can remove punctuation such as comma, single/double quote, question marks at the start and end of each word. E.g. beautiful!!!!!! replaced with beautiful

- Words must start with an alphabet - For simplicity sake, we can remove all those words which don't start with an alphabet. E.g. 15th, 5.34am

simpleDemo.py

```
34      def getStopWordList(stopWordListFileName):
35          #read the stopwords
36          stopWords = []
37          stopWords.append('AT_USER')
38          stopWords.append('URL')
39
40          fp = open(stopWordListFileName, 'r')
41          line = fp.readline()
42          while line:
43              word = line.strip()
44              stopWords.append(word)
45              line = fp.readline()
46          fp.close()
47          return stopWords
48      #end
49
50      #start getfeatureVector
51      def getFeatureVector(tweet, stopWords):
52          featureVector = []
53          words = tweet.split()
54          for w in words:
55              #replace two or more with two occurrences
56              w = replaceTwoOrMore(w)
57              #strip punctuation
58              w = w.strip('\'"?,.')
59              #check if it consists of only words
60              val = re.search(r"^[a-zA-Z][a-zA-Z0-9]*[a-zA-Z]+[a-zA-Z0-9]*$", w)
61              #ignore if it is a stopWord
62              if(w in stopWords or val is None):
63                  continue
64              else:
65                  featureVector.append(w.lower())
66          return featureVector
67      #end
```

8.Feature Extraction:

The following code, extracts the tweets and label from the csv file and processes it as above and obtains a feature vector and stores it in a variable called "tweets".

```
#Read the tweets one by one and process it
inpTweets = csv.reader(open('data/215.csv', 'r'), delimiter=',')
stopWords = getStopWordList('data/feature_list/stopwords.txt')
count = 0;
featureList = []
tweets = []
for row in inpTweets:
    sentiment = row[0]
    tweet = row[1]
    processedTweet = processTweet(tweet)
    featureVector = getFeatureVector(processedTweet, stopWords)
    featureList.extend(featureVector)
    tweets.append((featureVector, sentiment));
#end loop
```

9.Extract Features Method

```
#start extract_features
def extract_features(tweet):
    tweet_words = set(tweet)
    features = {}
    for word in featureList:
        features['contains(%s)' % word] = (word in tweet_words)
    return features
#end
```

10.Naive Bayes Classifier:

For classifying the data/tweets into positive,negative and neutral. Naive Bayes is a simple model which works well on text categorization . We use a multinomial Naive Bayes model. Class $c*$ is assigned to tweet *d*, where

$$c* = argmac_c P_{NB}(c|d)$$

$$P_{NB}(c|d) := \frac{(P(c)\sum_{i=1}^{m} P(f|c)^{n_i(d)})}{P(d)}$$

In this formula, *f* represents a feature and *ni(d)* represents the count of feature *fi* found in tweet d. There are a total of m features. Parameters *P(c)* and *P(f|c)* are obtained through maximum likelihood estimates, and add-1 smoothing is utilized for unseen features. I used the Python based Natural Language Toolkit library to train and classify using the Naive Bayes method.

Code:-

```
# Generate the training set
training_set = nltk.classify.util.apply_features(extract_features, tweets)

# Train the Naive Bayes classifier
NBClassifier = nltk.NaiveBayesClassifier.train(training_set)

# Test the classifier
list_of_parties=['BJP','Congress','SP','BSP']
iTweet= csv.reader(open('data/b.csv', 'r'),delimiter=',')
#myfile = open('data/naive_result.csv', 'w')
#wr = csv.writer(myfile, quoting=csv.QUOTE_ALL)
#testTweet = 'Disappointing day. Attended a car boot sale to raise some funds
fp = open('final_naive_bayes.txt', 'w')
```

[Reference:-The Predictive Power of Social Media: On the Predictability of U.S. Presidential Elections using Twitter by Kazem Jahanbakhsh ,Yumi Moon]

11.Support Vector Machine:

Support Vector Machines (SVM) is pretty much the standard classifier which is used for any general purpose classification. I will use the libsvm library (written in C++ and has a python handle)

Support Vector Machines is another popular classification technique . I have used libsvm library with a linear kernel. My input data are two sets of vectors of size m. Each entry in the vector corresponds to the presence a feature. In the unigram feature extractor, each feature is a single word found in a tweet. If the feature is present, the value is 1, but if the feature is absent, then the value is 0. I use feature presence, as opposed to a count, so that I do not have to scale the input data, which speeds up overall processing .

Code:-

simpleDemo.py × svm_final.py ×

```
def getSVMFeatureVector (tweet,featureList):
    sortedFeatures= sorted(featureList)
    test_feature_vector=[]
    for t in tweet:
        map1={}
        for w in sortedFeatures:
            map1[w] = 0

        tweet_word = t[0]

        for word in tweet_word:
            word = replaceTwoOrMore(word)
            word = word.strip('\'"?,.')
            if word in map1:
                map1[word] = 1

        values = list(map1.values())
        test_feature_vector.append(values)
        return test_feature_vector
```

12.Maximum Entropy Classifier:

We use the 'General Iterative Scaling' algorithm and stick to 10 iterations.

The idea behind Maximum Entropy models is that one should prefer the most uniform models that satisfy a given constraint . MaxEnt models are feature-based models. In a two class scenario, it is the same as using logistic regression to find a distribution over the classes. MaxEnt makes no independence assumptions for its features, unlike Naive Bayes. The model is represented by the following:

$$P_{ME}(c|d,\lambda) = \frac{\exp[\Sigma_i \lambda_i f_i(c,d)]}{\Sigma_{c'} \exp[\Sigma_i \lambda_i f_i(c,d)]}$$

In this formula, *c* is the class, *d* is the tweet, and λ is a weight vector. The weight vectors decide the significance of a feature in classification. A higher weight means that the feature is a strong indicator for the class. The weight vector is found by numerical optimization of the *λi's* so as to maximize the conditional probability.

We use the Python NLTK library to train and classify using the Maximum Entropy method. For training the weights we use conjugate gradient ascent.

Theoretically, Max Entropy performs better than Naive Bayes because it handles feature overlap better. However, in practice, Naive Bayes can still perform well on a variety of problems..

```
featureList = list(set(featureList))

# Generate the training set
training_set = nltk.classify.util.apply_features(extract_features, tweets)

# Train the Naive Bayes classifier
#NBClassifier = nltk.NaiveBayesClassifier.train(training_set)
MaxEntClassifier = nltk.classify.maxent.MaxentClassifier.train(training_set, max_iter = 10)#, '
#testTweet = 'just had some bloodwork done. My arm hurts'

list_of_parties=['BJP','Congress','SP','BSP']
iTweet= csv.reader(open('data/b.csv', 'r'),delimiter=',')
#myfile = open('data/maxent_result.csv', 'w')
#wr = csv.writer(myfile, quoting=csv.QUOTE_ALL)
fp = open('final_maxent.txt', 'w')
```

14.Results for three algorithms stored on hive:

We stored result data from three algorithms in 3 hive tables.

These tables are then used to retrieve no of positive and negative instances for a articular party.

15.Code for retrieving count from hive table:

We connected python with hive using pyhive package and wrote HiveQL queries in the python script.

6 map reduce jobs were run using this script.

16.Polarity lexicon model modified by Gayo-Avello et al.:- Used for prediction of the results

$$(c_1) = \frac{pos(c_1) + neg(c_2)}{pos(c_1) + neg(c_1) + pos(c_2) + neg(c_2)}$$

c1=party 1

c2=party 2

Pos(c1) positive words for party 1

Pos(c2) positive words for party 2

Neg(c1) negative words for party 1

Neg(c2) negative words for party 2

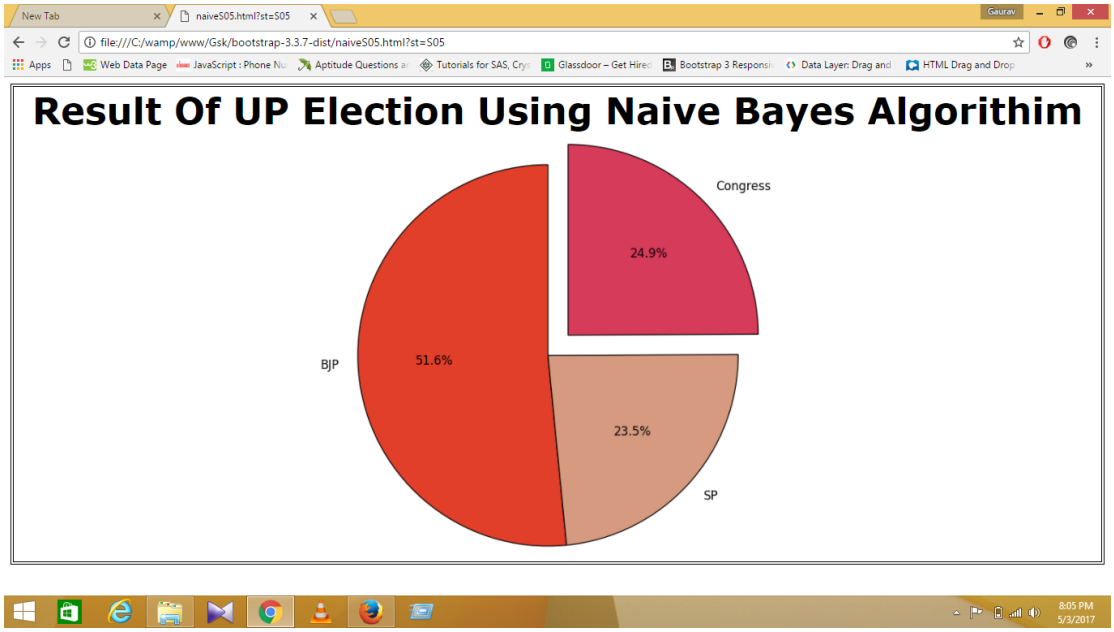
However, the equation does not use the neutral tweets as they don't express a candidate preference.

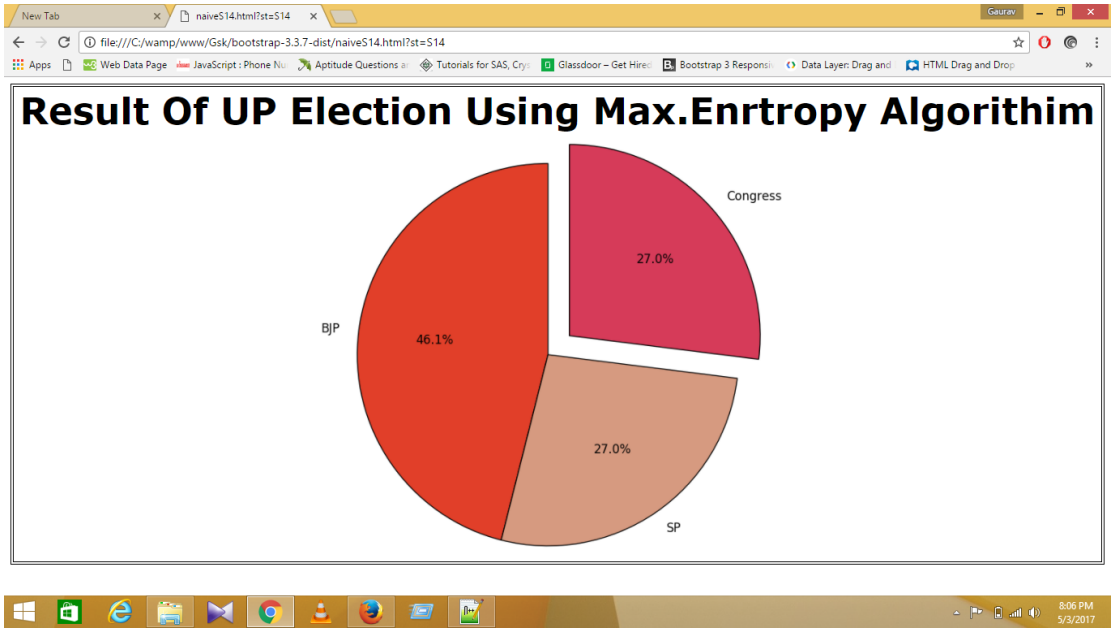
[Reference:-Kabir Ismail Umar Data mining for social media analysis using twitter.]

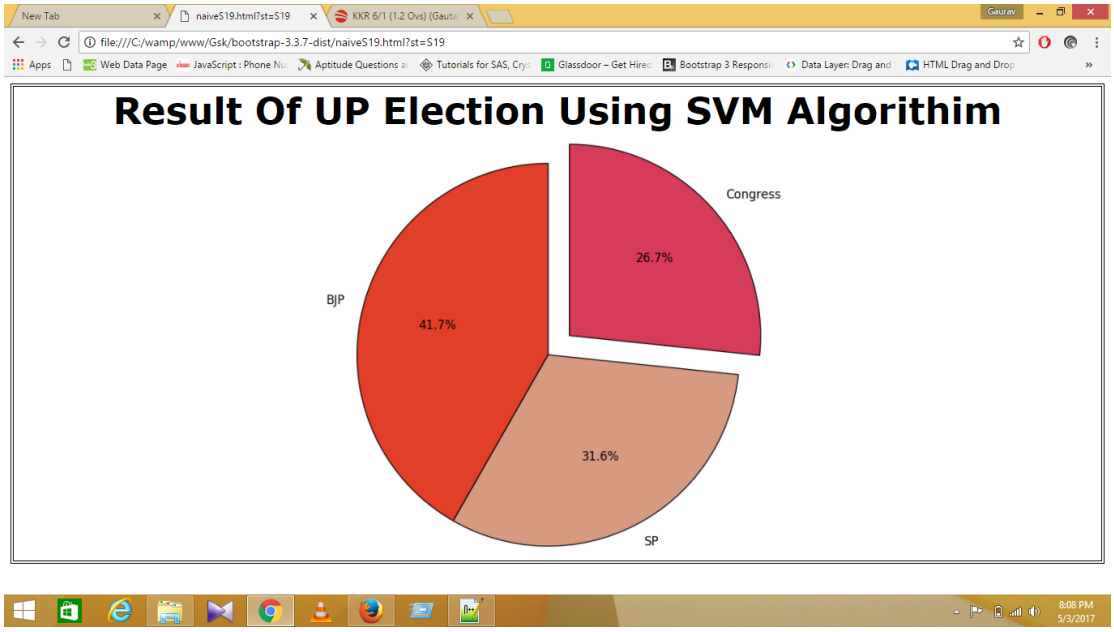
17. Result visualization:

For the visualization of final result, we used pandas library of python. In that we created data frame from the data obtained after prediction. We used functions for creation of pie chart.

Final pie charts are displayed on website.





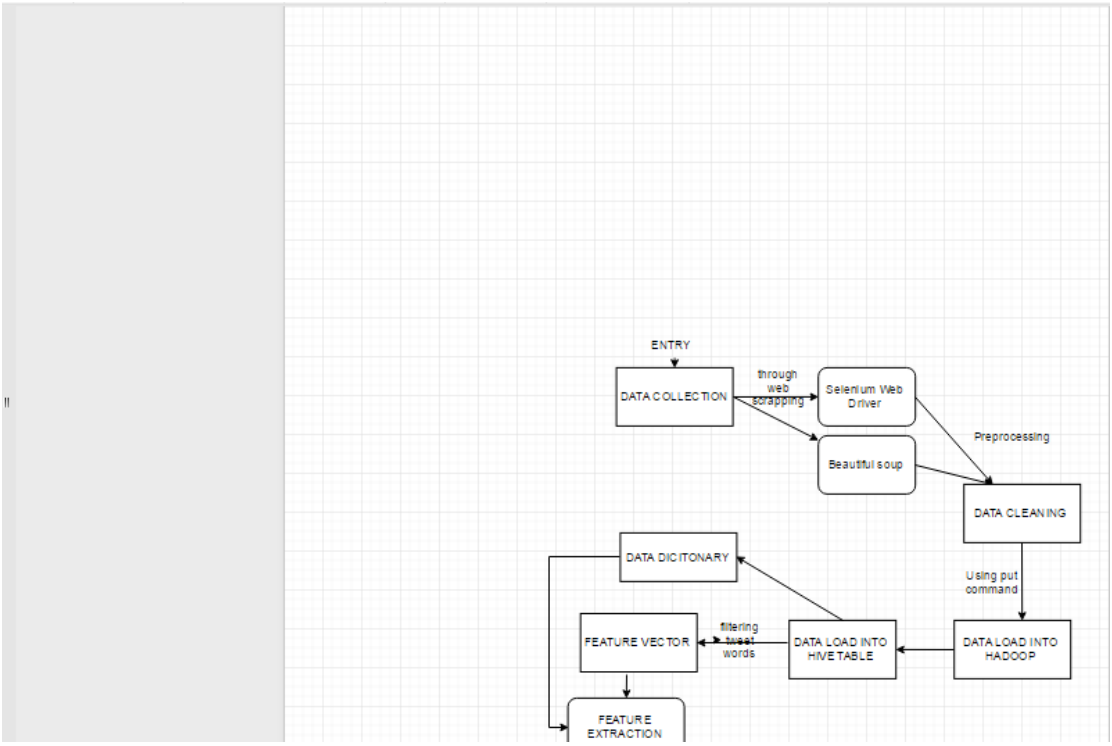


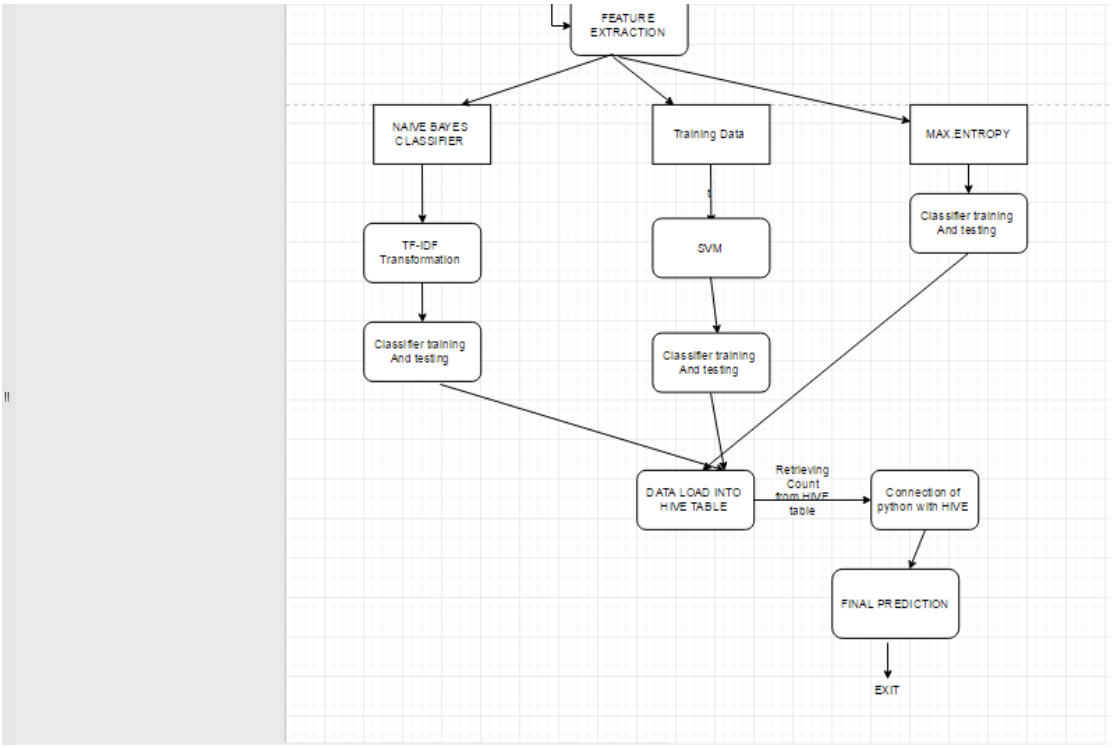
Website designing:-

An online portal to display our prediction in the form of pie charts . In addition to it, various constituencies and political parties list can be found on the website.

[Source:- election commission website.]

FLOWCHART


ENTRY
DATA COLLECTION
through web scrapping
Selenium Web Driver
Beautiful soup
Preprocessing
DATA CLEANING
DATA DICITONARY
Using put command
FEATURE VECTOR
filtering tweet words
DATA LOAD INTO HIVE TABLE
DATA LOAD INTO HADOOP
FEATURE EXTRACTION


FEATURE EXTRACTION
NAIVE BAYES CLASSIFIER
Training Data
MAX.ENTROPY
TF-IDF Transformation
SVM
Classifier training And testing
Classifier training And testing
Classifier training And testing
DATA LOAD INTO HIVE TABLE
Retrieving Count from HIVE table
Connection of python with HIVE
FINAL PREDICTION
EXIT

Language, database systems, packages used

We have used python for implementing all the algorithms and also for scraping twitter and data collection.

Database System used:-

We have stored the data in hive table. In addition to it party and sentiment for the tweets is stored in 3 different tables for different algorithms.

Packages Used:-

- Nltk for naive bayes and maximum entropy.
- Libsvm for svm.
- Pandas for visualization.
- Pyhive for connecting python script with hive table and performing queries.
- Selenium and BeautifulSoup for scraping of twitter data.
- In addition to it we have used anaconda framework for all the machine learning packages on windows as well as cloudera.

Testing

Test case	Input tweet	Naive Bayes Output	SVM Output	Maximum Entropy Output
1	"The only belief of BJP is Vikas Vikas Vikas people of UP believe that only BJP can make this possible.."	bjp | positive	bjp | positive	bjp | positive
2	"UPElections 2017 Math suggest BJP set to win a comfortable majority Says Surjit Bhalla"	bjp | positive	bjp | positive	bjp | positive
3	UPElections Strongly condemn road show. Urge the EC to take action against BJP for violation of model code of conduct.	bjp | negative	bjp | negative	bjp | negative
4	UPElections So how did rape accused Prajapati get SP ticket writes Rajesh Singh	sp | negative	sp | negative	sp | positive
5	Maharashtra civic poll result indicates clear majority of BJP4UP in UPElections thanks to people of Maha. for showing their faith BJP	bjp | positive	bjp | positive	bjp | positive

FUTURE WORK

Machine learning techniques perform well for classifying sentiment in tweets. I believe the accuracy of the system could be still improved. Below is a list of ideas I think could help the classification:-

Semantics The algorithms classify the overall sentiment of a tweet. The polarity of a tweet may depend on the perspective you are interpreting the tweet from. For example, in the tweet "BJP beats SP :)", the sentiment is positive for BJP and negative for SP. In this case, semantics may help. Using a semantic role label may indicate which noun is mainly associated with the verb and the classification would take place accordingly. This may allow "BJP beats SP :)" to be classified differently from "SP beats BJP :)".

Bigger Dataset The training dataset in the order of millions will cover a better range of twitter words and hence better unigram feature vector resulting in an overall improved model. This would vastly improve upon the existing classifier results.

Internationalization Currently, we focus only on English tweets but Twitter has a huge international audience. It should be possible to use our approach to classify sentiment in other languages with a language specific positive/negative keyword list.

REFERENCES:-

[1] Kabir Ismail Umar & Fatima Chiroma .Data Mining for Social Media Analysis using Twitter to predict the 2016 US Presidential Election.1 September, 2016.

[2] Harald Schoen, University of Bamberg (Germany), Daniel Gayo-Avello, University of Oviedo (Spain), Panagiotis Takis Metaxas, Wellesley College and Harvard University.The Power of Prediction with Social Media.

[3] Donald E. Brown and Ahmed Abbasi, University of Virginia. Predictive analytics.

[4] Patrick Hummel (Google) ,David Rothschild (Microsoft Research) PHummel@alumni.gsb.stanford.edu. Fundamental Models for Forecasting Election

[5] David W. Nickerson University of Notre Dame ,Todd Rogers Harvard Kennedy School .Political Campaigns and Big Data Faculty Research Working Paper Series

[6] Lei Shi Opera Solutions,Neeraj Agarwal, Ankur Agrawal,Predicting US Primary Elections with Twitter.

[7] Ronald MacDonald1 and Xuxin Mao2 .18 January 2016. Forecasting the 2015 General Election with Internet Big Data: An Application of the TRUST Framework

[8] Rohan Sampath, Yue Teng.Classification and Regression Approaches to Predicting United States Senate Elections

[9] Data Mining in US presidential election campaign. https://www.promptcloud.com/blog/data-mining-in-presidential-election-campaign/

[10] Robert Tibshirani. The lasso method for variable selection .In Statistics in Medicine, pages 385–395, 1997.

[11] D. O. Computer, C. wei Hsu, C. chung Chang, and C. jen Lin. A practical guide to support vector classification chih-wei hsu, chih-chung chang, and chih-jen lin. Technical report, 2003.

[12] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, March 2000.

[13] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Micro-blogging as online word of mouth branding. In CHI EA ‟09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, pages 3859–3864, New York, NY, USA, 2009. ACM.

[14] T. Joachims. Making large-scale support vector machine learning practical. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in kernel methods: support vector learning, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.

[15] C. D. Manning and H. Schutze. Foundations of statistical natural language processing. MIT Press, 1999.

[16] G. Mishne. Experiments with mood classification in blog posts. In 1st Workshop on Stylistic Analysis Of Text For Information Access, 2005.

[17] K. Nigam, J. Lafferty, and A. Mccallum. Using maximum entropy for text classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61–67, 1999.

[18] B. Pang and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2):1–135, 2008.

[19] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79–86, 2002.