CS 583 Project 2

Twitter Sentiment Analysis

Mayank Raj(mraj3@uic.edu)
Chinmay Nautiyal(cnauti2@uic.edu)

Abstract

In this report we address the problem of sentiment analysis of twitter dataset. We use a number of machine learning methods to perform sentiment analysis. In the end we decide upon a model which gives the best prediction on the training dataset based on a set of parameters like accuracy, precision, recall and Fscore.

1 Introduction

Twitter is a popular social networking website where members create and interact with messages known as "tweets". This serves as a mean for individuals to express their thoughts or feelings about different subjects. Various different parties such as consumers and marketers have done sentiment analysis on such tweets to gather insights into products or to conduct market analysis. Furthermore, with the advancement of machine learning algorithms, we are able to improve the accuracy of sentiment analysis predictions.

In this report, we will attempt to conduct sentiment analysis on "tweets" using various different machine learning algorithms. We will attempt to classify the polarity of the tweet where it is either positive, negative or neutral. The dataset used in the report is a set of tweets which were tweeted for Obama and Romney in the run up to 2012 United States Presidential elections. The model used in the predictions classifies the tweet's sentiment for Obama and Romney. The data provided comes with emoticons, hashtags, usernames and hashtags which are required to processed and converted into standard form. We also need to extract useful features from the text such as unigrams and bigrams which is a form of representation of the tweet.

We use several machine learning algorithms to conduct sentiment analysis using the extracted features. Ensemble methods were also tried for the predictions. Ensembling is a form of meta learning algorithms where we combine different classifiers in order to improve prediction accuracy. Finally, we report our experimental results and findings at the end.

2 Techniques

2.1 Data Description

The data given is in the form of a comma-separated values filled with tweets and their corresponding sentiments. The training dataset is a csv file of type date, time, Annonated_tweet,

Class. Class is either 1 (positive), 0 (neutral) or -1 (negative) and the tweet is in the Annonated_tweet. Similarly, the test dataset is a csv file of type tweet_id and tweet.
The dataset is a mixture of words, emoticons, symbols, URLs and references to people. Words and emoticons contribute to predicting sentiment, but URLs and references may be ignored as they do not contribute substantially to the sentiment. The words are also a mixture of misspelled words, extra punctuations, and words with many repeated letters. The tweets ,thus, have to be preprocessed to standardize the dataset. The training dataset for Obama has 5471 tweets and 5648 tweets for Romney.

		Total
Obama	Words	109667
	User Mentions	2226
	Hashtags	2134
	! And ? Marks	2651
	URLs	1591
Romney	Words	114404
	User Mentions	2055
	Hashtags	2080
	! And ? Marks	2361
	URLs	1629

Table 1: Statistic of preprocessed train dataset

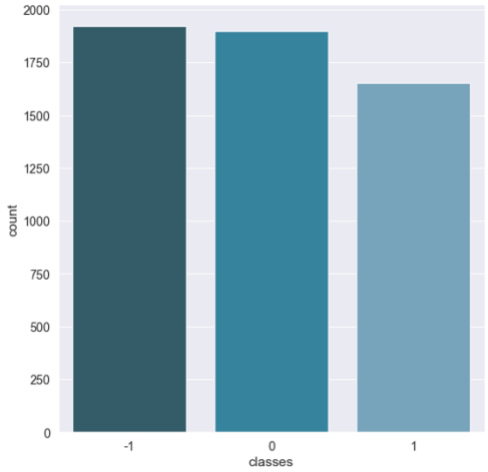

Figure 1: For Obama dataset

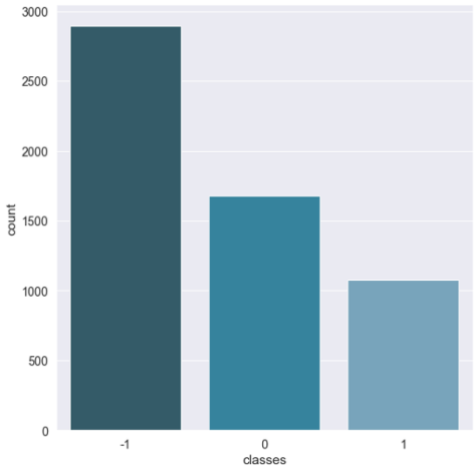

Figure 2: For Romney dataset

2.2 Pre-processing

Raw tweets taken directly from twitter generally result in noisy dataset. This is due to the casual nature of people's usage of social media. Therefor raw twitter data has to be normalized to create a dataset which can be easily learned by various classifiers. We have applied extensive number of

pre-processing steps to standardize the dataset and reduce its size. We first do some general pre-processing on tweets which is as follows:

- Remove the mentions, as we want to generalize the tweet
- Remove the hash tag sign (#) but not the actual hashtag as this may contain information
- Set all words to lowercase
- Remove all punctuations, including question marks and exclamation marks
- Remove the URLs as they do not contain useful information. We did not notice a difference in the number of URLs used between the sentiment classes.
- Converted the emojis to one word
- Remove Stopwords but take care of negation words as they may contain crucial information
- Apply PorterStemmer to keep the stem of the words. We also used lemmatization with POS tags but it did not produce any significant gain in the classification.

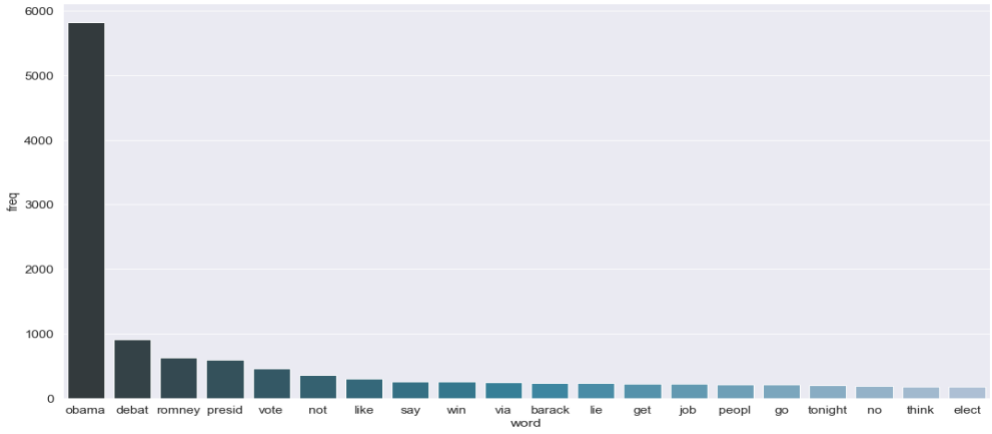

Figure 3: Word distribution for Obama dataset

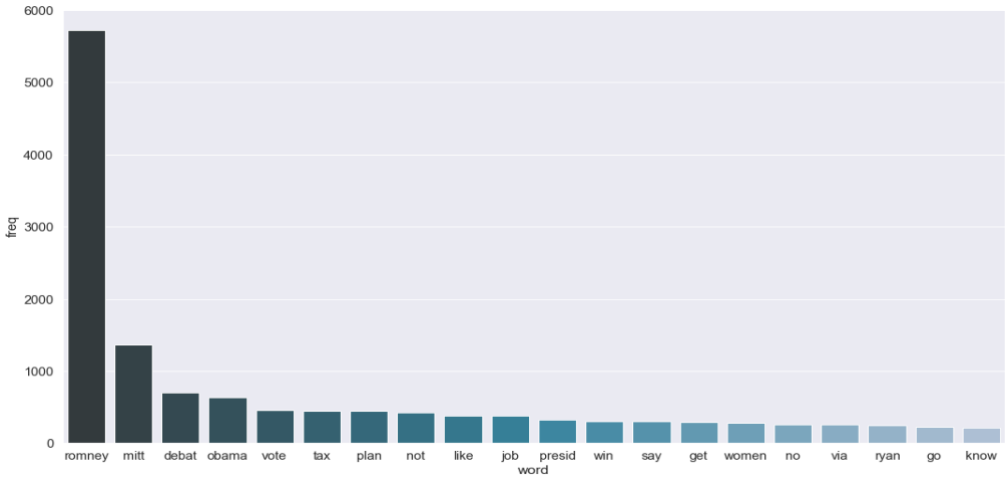

Figure 4: Word distribution for Romney dataset

2.3 Features Extraction

We extract two types of features from our dataset, namely unigrams and bigrams. Probably the simplest and the most commonly used features for text classification is the presence of single words or tokens in the text. Bigrams are word pairs in the dataset which occur in succession in the corpus.
After extracting the unigrams and bigrams, we represented each tweet as a feature vector in either sparse vector representation or dense vector representation depending on the classification used. For sparse vector representation, each unigram (and bigram) is given a unique index depending on its rank. The feature vector for a tweet has a positive value at the indices of unigrams (and bigrams) which are present in that tweet and zero elsewhere which is why the vector is sparse. The positive value at the indices of unigrams (and bigrams) depends on the feature type we specify which is one of presence and frequency.

- Presence – In the case of presence feature type, the feature vector has a 1 at indices of unigrams (and bigrams) present in tweet and 0 otherwise.
- Frequency – In the case of frequency feature type, the feature vector has a positive integer at the indices of unigrams (and bigrams) which is the frequency of that term in the tweet and 0 elsewhere. A matrix of such term-frequency vectors is constructed for the entire training dataset and then each term frequency is scaled by the inverse-document-frequency of the term (idf) to assign higher values to important terms.

  $$Idf(t) = \log [(1 + nd) / (1 + df(d, t))] + 1$$

  where nd is the total number of documents and df(d, t) is the number of documents in which the term t occurs.

2.4 Classifiers

2.4.1 Multinomial Naïve Bayes

Naive Bayes is a simple model which can be used for text classification. In this model, the class $c\hat{}$ is assigned to a tweet t, where

$$c\hat{} = \text{argmax}\ c\ P(c|t)$$
$$P(c|t) \propto P(c)Yn\ i=1\ P(fi\ |c)$$

In the formula above, fi represents the i-th feature of total n features. P(c) and P(fi |c) can be obtained through maximum likelihood estimates.

2.4.2 Support Vector Machines

SVM, also known as support vector machines, is a non-probabilistic binary linear classifier. For a training set of points (xi , yi) where x is the feature vector and y is the class, we want to find the maximum-margin hyperplane that divides the points with yi = 1 and yi = −1.

2.4.3 Logistic Regression

Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one

dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

2.4.4 RandomForest Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

2.4.5 k-Nearest Neighbor

The K-nearest neighbors (KNN) algorithm is a type of supervised machine learning algorithms. KNN is extremely easy to implement in its most basic form, and yet performs quite complex classification tasks. It is a lazy learning algorithm since it doesn't have a specialized training phase. Rather, it uses all of the data for training while classifying a new data point or instance.

3 Evaluation

Now we present the results that were produced after using classification methods on 10-folds cross validation for the training dataset.

Candidate	Classifier	Class	Precision	Recall	F-score	Accuracy
OBAMA	MNB_countvec	1	0.63	0.63	0.63	
		0	0.58	0.48	0.53	
		-1	0.61	0.7	0.65	0.608
	MNB_tfidf	1	0.66	0.59	0.63	
		0	0.61	0.52	0.56	
		-1	0.62	0.75	0.68	0.628
	LR_countvec	1	0.61	0.65	0.63	
		0	0.59	0.56	0.58	
		-1	0.67	0.67	0.67	0.626
	LR_tfidf	1	0.63	0.61	0.62	
		0	0.58	0.57	0.58	
		-1	0.66	0.68	0.67	0.624
	SVM	1	0.47	0.42	0.44	
		0	0.45	0.46	0.45	
		-1	0.65	0.67	0.66	0.56
	RandomForest_tfidf	1	0.64	0.4	0.49	
		0	0.47	0.68	0.56	
		-1	0.63	0.55	0.59	0.552
	RandomForest_countvec	1	0.74	0.32	0.45	
		0	0.49	0.63	0.55	
		-1	0.61	0.72	0.66	0.548
	KNN_tfidf	1	0.46	0.35	0.4	
		0	0.41	0.45	0.43	
		-1	0.47	0.51	0.49	0.445
	KNN_countvec	1	0.48	0.51	0.49	
		0	0.44	0.43	0.43	
		-1	0.5	0.49	0.49	0.449

Candidate	Classifier	Class	Precision	Recall	F-score	Accuracy
ROMNEY	MNB_countvec	1	0.51	0.41	0.45	
		0	0.5	0.34	0.4	
		-1	0.65	0.81	0.72	0.596
	MNB_tfidd	1	0.52	0.26	0.35	
		0	0.49	0.31	0.38	
		-1	0.62	0.87	0.73	0.589
	LR_countvec	1	0.59	0.36	0.45	
		0	0.52	0.36	0.43	
		-1	0.64	0.84	0.73	0.609
	LR_tfidf	1	0.52	0.26	0.35	
		0	0.5	0.3	0.37	
		-1	0.62	0.87	0.72	0.588
	SVM	1	0.29	0.61	0.39	
		0	0.67	0.01	0.02	
		-1	0.64	0.74	0.69	0.501
	KNN_tfidf	1	0.24	0.07	0.11	
		0	0.3	0.22	0.25	
		-1	0.54	0.76	0.63	0.497
	Knn_countvec	1	0.42	0.19	0.26	
		0	0.4	0.27	0.32	
		-1	0.57	0.79	0.66	0.507
	RandomForest_countvec	1	1	0.01	0.02	
		0	1	0.01	0.02	
		-1	0.52	1	0.68	0.52

4 Conclusion

We also tried some under sampling and oversampling techniques to get some sort of class balance, but that did not lead us to a significant gain in performance of the models. Based on the above results, we conclude:

- Classifiers achieve the best results when using the features of Counvectorizer
- Logistic Regression outperforms all the other classification techniques.
- The best performance on the set comes from Logistic Regression with features from Countvectorizer.
- Best Parameters:
    - C value of 0.25
    - L2 regularization
    - max_df: 0.25 or maximum document frequency 0.25
    - min_df: 1 or the words need to appear in at least two tweets
    - n-gram_range: (1,2), both single words and bigrams are used

5 References

1. www.kaggle.com
2. www.stackabuse.com
3. https://scikit-learn-org
4. https://nltk.org
5. Web Data Mining – Bing Liu
6. Sentiment Analysis and Opinion Mining – Bing Liu