TWITTER SENTIMENT ANALYSIS USING SUPERVISED LEARNING TECHNIQUES

Keshav Malpani & Venkata Ramkiran Chevendra
CS-583 Data Mining and Text Mining, Spring 2017
Department of Computer Science
University of Illinois at Chicago, Illinois

ABSTRACT

Twitter is a popular social networking platform where the users post their opinions or use it as an online questionnaire platform. The posts are called tweets and each tweet must be restricted to 140 characters. In this project, sentimental analysis is carried on the political tweets related to 2012 US presidential elections between Barack Obama and Mitt Romney. A labelled data classifying the tweets as positive, negative, neutral and mixed tweets are provided. A learning model was created using the labelled data to classify any given tweet as a positive, negative or neutral opinion on Obama and Romney. The mixed case is ignored. Data preprocessing is done first on the provided labelled data (training data). Various classifiers are used to create the model to classify new set of tweets (test data) and their relative performances are discussed in this report. The performance of the model is determined by various parameters such as accuracy, precision, recall and F-Score.

INTRODUCTION

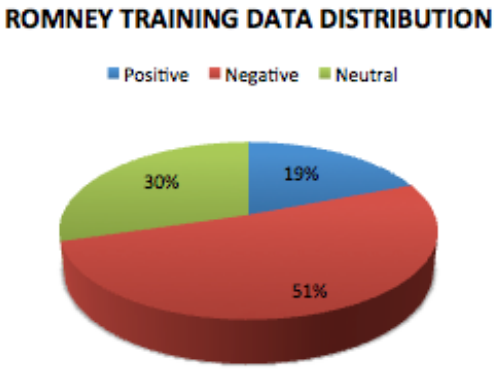
Twitter is a microblogging site with several millions of tweets per day posted by millions of users every day. According to a survey (Collaborative Search Revisited by Morris), approximately 33% of twitter users posted questions on the twitter platform. Rest of the users made the twitter their opinion and review destination. Thus, twitter is an ideal tool to perform the sentimental analysis on the current and trending issues. In our project, the labelled twitter data is collected about 2012 US presidential election. The tweets corresponding to position, negative, neutral and mixed sentiments of both the candidates Obama and Romney. The labelled data is used to train the classifiers to predict the other such tweets and determine whether a tweet falls under a positive or a negative opinion. The positive opinion is given a class of 1 where as a negative opinion is given a class of -1 and neutral opinion with 0. There is a mixed opinion with class 2 in the twitter dataset which we ignore.

As twitter is a social networking site and is not bound by any formal restrictions, there tends to be usage of internet acronyms, spelling mistakes, emoticons, other characters that express special meanings and colloquial slangs by diverse twitter population. Hence the preprocessing is to be done on the dataset before training and building a classifier. Preprocessing is the one of the important steps in building an ideal classifier which improves the evaluation parameters.

SAMPLING OF TRAINING DATA

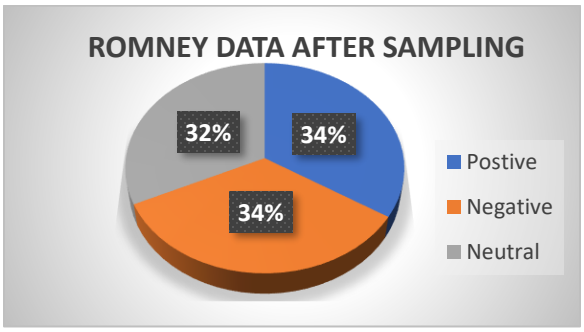
Sampling allows any skewed dataset to become balanced and thus the balanced inputs are provided to the classifier learning and building. Sampling is concerned with the selection of a subset of individuals from within a statistical population to estimate characteristics of the whole population. If one of the classes of the population is outnumbering the other, then either oversampling or under sampling

techniques are used to make the balanced dataset.



In the training dataset provided in the project, Obama dataset is fairly balanced as shown in the pie charts with similar proportions of the positive, negative and neutral datasets. Romney dataset is imbalanced with negative dataset dominating the other two datasets.

Hence the sampling is done to make the dataset balanced.



Sampled Romney was used to train the model with 34% positive, 34% negative and 32% neutral tweets.

TECHNIQUES

Pre-processing:

1. **Removal of URLs:** The URLs in the tweets are removed as they do not contribute to any significant sentiment of the tweet. The URLs are of the form http://sh.eu/HgfDsW. There were parsed and replaced with a blank space.
2. **Removal of usernames:** The usernames that are referred to or pointed to in a tweet also doesn't add to any sentiment of the tweet. Hence the usernames are also parsed and replaced with a blank space.
3. **Removal of HTML, punctuation and special characters:** The HTML tags, punctuation and special characters like[!;,?.'"\-%$<>&\(\)\{\}\[\]__=,;:*\\~\+#] are removed from the tweets as they too do not any other information to the tweet.
4. **Removal of stop words:** The tweets are in natural language and as such there are many stop words, which are present for the sake of grammar of the language. The stop words corpus was obtained from Internet and removed through NLTK stop word removal process. Some modifications were required to this as the corpus also had some negative words such as nor, not, neither which are important in identifying negative sentiments and should not be removed.
5. **Stemming:** Stemming is the process of reducing a word to its root form. NLTK provides various packages for stemming such as the Porter Stemmer, Lancaster Stemmer and so on. The Porter Stemmer was used in this project, which uses various rules for suffix stripping. In addition to stemming the train and test data, the positive and negative word corpus was also stemmed. Stemming reduces the feature space as many derived words are

reduced to the same root form. Multiple features now point to the same word and hence it increases the probability of the word.

6. **Tokenization:** Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing such as parsing or text mining. Tokenization is useful both in linguistics (where it is a form of text segmentation), and in computer science, where it forms part of lexical analysis.
7. **Unigrams, bigrams and lemmatization:** All the three techniques were employed to check if there is any improvement in the performance of the evaluation parameters, but they are hindering the evaluation parameters. Hence those were removed from the code.

CLASSIFIERS EMPLOYED:

BernouliNB: BernouliNB implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, Boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a BernoulliNB instance may binarize its input (depending on the binarize parameter). The results of this classifier is one of the best among many classifiers employed in the project.

SVM: Support Vector Machines is another popular classification technique. It constructs a hyper plane or set of hyper planes in a high-dimensional space such that the separation is maximum. The hyper plane identifies certain examples close to the plane, which are called as support vectors.

Decision Tree Classifier: A Decision Tree is a flowchart-like tree structure, in which each internal node represents a test on an attribute (features) and each branch represents an outcome of the test, and each leaf node represents a class (+ve, -ve or neutral). The results of decision tree haven't been consistent.

Logistic Regression: Logistic Regression models are feature-based models. The idea behind this model is that one should prefer the most uniform models that satisfy a given constraint. After BernouliNB and SVM, logistic Regression is the classifier which produced accurate and consistent results.

Random Forest Classifier: Random forests or random decision forests are an ensemble learning method or classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Like decision trees, the results of decision tree haven't been consistent.

SGD classifier: This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). SGD allows minibatch (online/out-of-core) learning, see the partial_fit method. For best results using the default learning rate schedule, the data should have zero mean and unit variance.

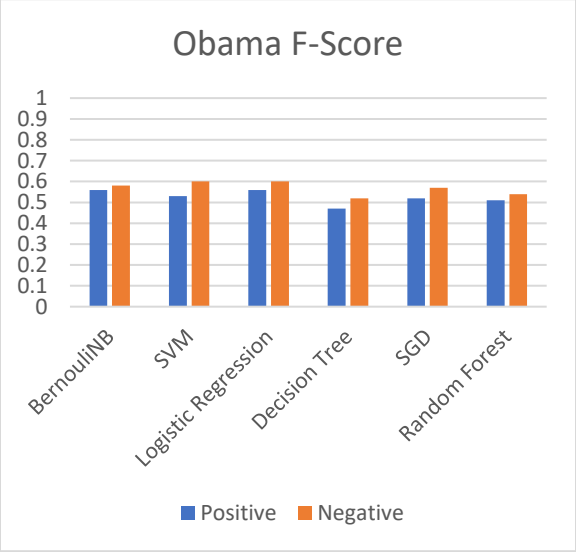
Neural Networks MLP Classifier: The neural network classifier was used in the project and the execution time of the classification took so long time and hence this was not employed. The results using this classifier were also not that

consistent when compared to classifiers like BernouliNB and SVM.

Voting Classifier: It is an ensemble classifier. Soft Voting/Majority Rule classifier for unfitted estimators. All the classifiers employed are parsed in the voting classifier. The voting classifier determines the best classifiers and averages the evaluation parameters obtained from those best classifiers.

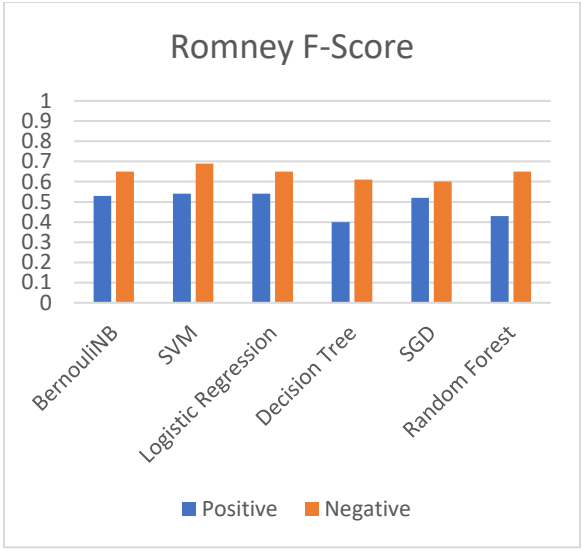
EVALUATION

The evaluation parameters of the classifiers are F-Score, accuracy, precision and recall. The below chart shows the Obama F-Score for various classifiers.



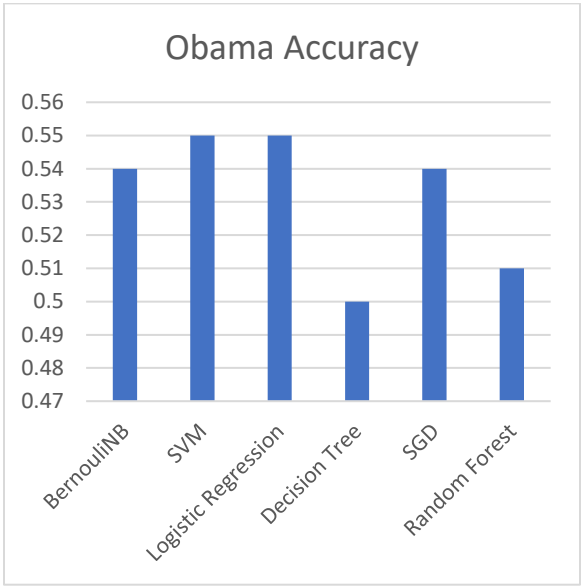
BernouliNB, SVM and Logistic Regression are the best classifiers which showed significant results than the rest of the classifiers.

Similarly F-Score of Romney is in the chart below

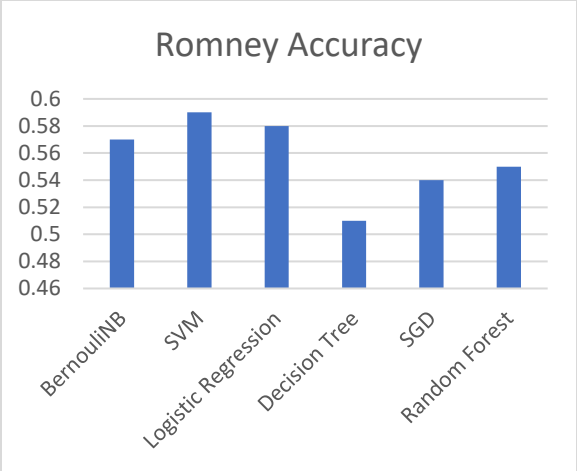


For Romney data too, the three classifiers BernouliNB, SVM and Logistic Regression showed significant results.

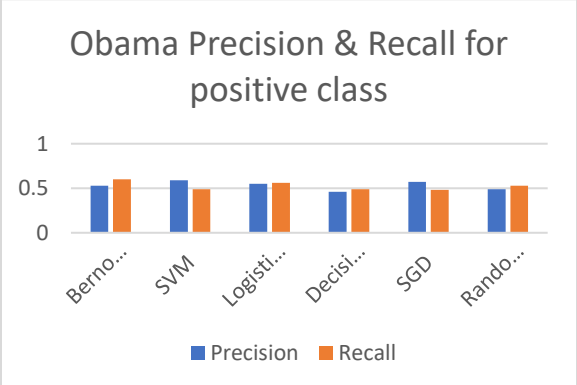
The accuracy results of both the data sets are as follows:

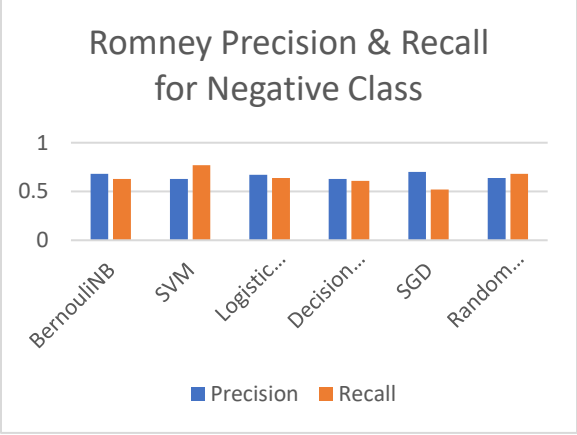


Similarly, the accuracy scores for Romney with various classifiers are as follows



The sample precision and recall of Obama and Romney datasets are as follows:





CONCLUSION:

We experimented with several classifiers as shown above. We used kfold cross validation for the training dataset and trained the classifiers. We used SMOTE(Synthetic Minority OverSampling technique) for the test which balances the skewed data if any and produces better results. The results with and without SMOTE is quite drastic. We also tried many steps in the preprocessing which might improve the results such as lemmatization and replacing emoticons and frequently used words like #teamobama, #romneyfor2012 etc.. replace with corresponsing sentiments. These changes had minor impact on the final results. The n-grams such as unigram and bigram were tried but they are decreasing the performance and hence were not used.

With many classifiers used in our project, Naïve bayes BernouliNB, SVM and LogisticRegression produced better results than the rest. The Voting classifier is used at the end to produce the average of the results from the best classifiers. In this project, only the texts of the tweets are considered and other information like the users who tweet them, the times of the retweets and other factors are also potentially useful and as a future scope of this project we would like to experiment with these attributes and few more supervised learning optimization algorithms such as Stochastic Gradient Descent and Vectorization and some semi supervised learning models.

REFERENCES:

1. Scikit-learn: http://scikit-learn.org/

2. NLTK: http://www.nltk.org/genindex.html

3. Tf-Idf Vectorizer: scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

4. SMOTE: http://contrib.scikit-learn.org/imbalanced-learn/generated/imblearn.over_sampling.SMOTE.html