Twitter Sentiment Analysis Project Report

By: - Piyush Chandra & Ashwin Krithekavasan

Abstract

We are given a list of tweets, which are either positive or negative or neutral or mixed. The goal of our project is given the existing set of tweets, we need to train our classifier on the set of training data and then classify the unseen test data as positive, negative or neutral.

Introduction

Sentiment analysis is about finding whether a statement is positive, negative or neutral. Twitter sentiment analysis is classifying tweets as positive, negative or neutral. In this project, we must classify test data of tweets. Nowadays a lot of people tweet; twitter statistics shows that 500 million of tweets are sent per day. Since twitter has become so popular, it is now used for predicting the results of major elections by classifying the tweets, even businesses try to find if their product is a success in the market by analyzing if people are talking good or bad about the product.

Since there are many tweets, it becomes difficult to analyze and find whether those tweets are positive, negative or neutral. For this, sentiment analysis is used to classify them.

In our project we have been give tweets of people during 2012 presidential election campaign, writing positive, negative or neutral tweets about Obama and Romney. We have to train our classifier using that as training data, and finally classify the unseen tweets and find its precision, recall, f-score and then the overall accuracy of test data.

Techniques Used

Following were the **preprocessing steps** that we took:

1. Removal of stop-words.
2. HTML tag removal.
3. Punctuation symbol removal.
4. Stemming of words.
5. Parse the words in speech in different parts of speech.

For **feature selection,** we tried the following things:

1. If a word is present or not in the tweet
2. TF-IDF
3. Using parts of speech of the word as feature
4. Position of words in the tweet
5. Separate features for adjectives used in tweet
6. Separate features for verbs used in tweet
7. Separate features for corresponding adverbs used in tweet

Then we finalized on the following attributes as features

- If a word is present or not in a tweet and in what part of speech it is used
- Separate features for adjectives used in tweet
- Separate features for verbs used in tweet
- Separate features for corresponding adverbs used in tweet

Validation:

We have used 10-fold cross-validation to validate the model while using different classifiers.

Different **classifiers** that we have tried are the following:

1. Multinomial Naive Bayesian
2. Bernoulli Naive Bayesian
3. Logistic Regression
4. Linear SVM Classification
5. Nonlinear SVM Classification
6. K Nearest Neighbor Classification
7. Decision Tree Classification
8. Random Forest Classification
9. AdaBoost Classification

Evaluation

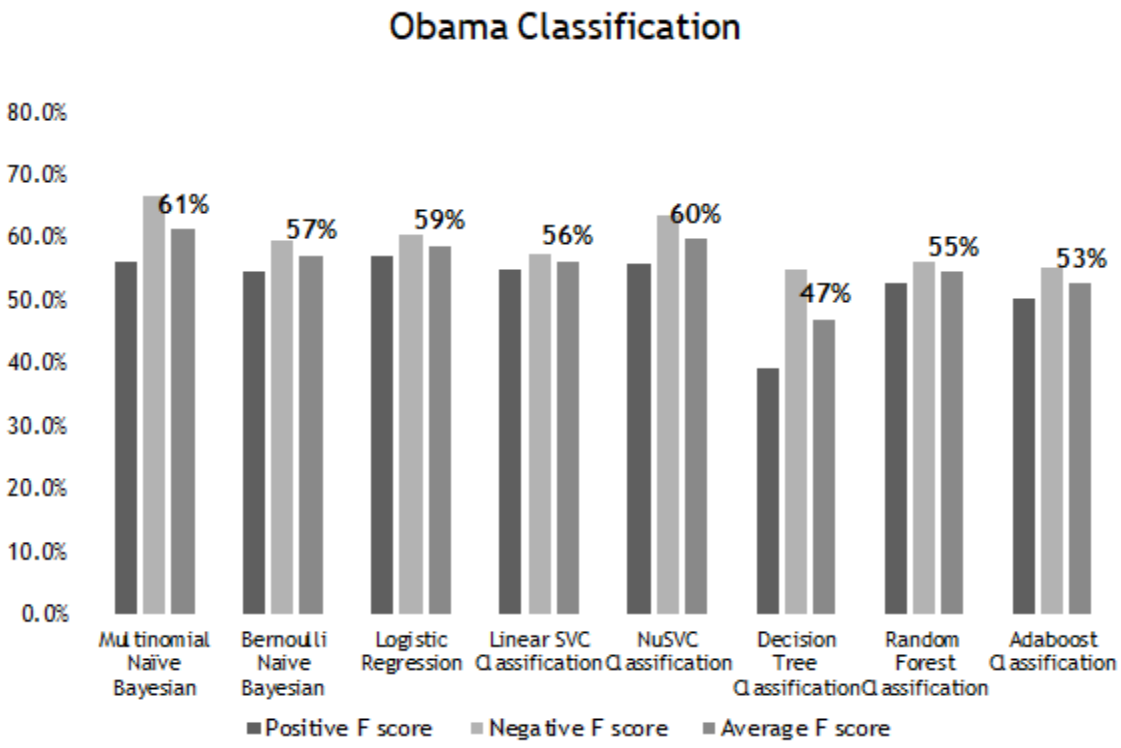
Obama Results:

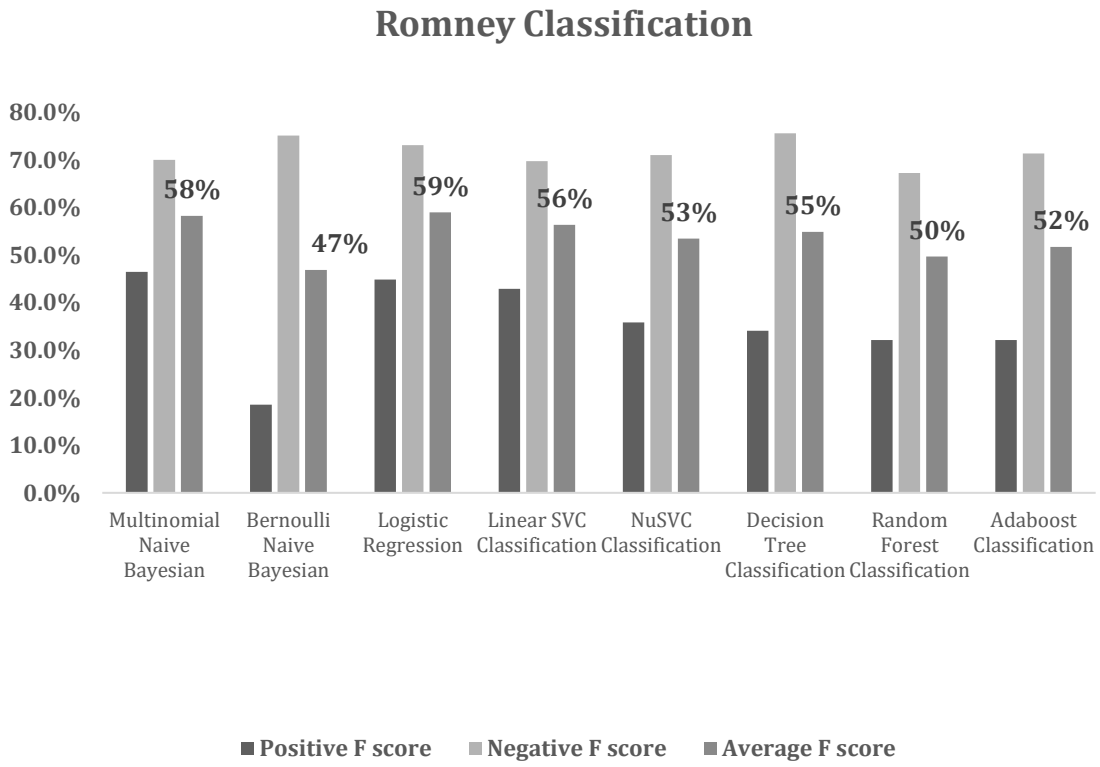
Classification Techniques	Accuracy	Positive Precision	Positive Recall	Positive F score	Negative Precision	Negative Recall	Negative F score	Average F score
Multinomial Naive Bayesian	58.7%	60.9%	52.3%	56.3%	60.2%	75.4%	67.0%	61.6%
Bernoulli Naive Bayesian	53.6%	47.0%	66.9%	55.2%	64.5%	56.2%	60.1%	57.6%
Logistic Regression	55.6%	52.7%	63.1%	57.5%	63.8%	57.8%	60.6%	59.0%
Linear SVC Classification	53.2%	48.8%	63.5%	55.2%	61.8%	54.1%	57.7%	56.5%
NuSVC Classification	56.0%	51.1%	62.7%	56.3%	63.7%	64.4%	64.0%	60.1%
Decision Tree Classification	46.7%	43.2%	36.4%	39.5%	49.4%	62.5%	55.2%	47.3%
Random Forest Classification	51.6%	47.7%	59.6%	53.0%	61.1%	52.6%	56.5%	54.8%
Adaboost Classification	50.3%	44.7%	58.7%	50.8%	59.7%	51.9%	55.5%	53.1%

Romney Results:

Classification Techniques	Accuracy	Positive Precision	Positive Recall	Positive F score	Negative Precision	Negative Recall	Negative F score	Average F score
Multinomial Naive Bayesian	58.5%	40.8%	54.1%	46.5%	72.5%	68.0%	70.2%	58.4%
Bernoulli Naive Bayesian	60.7%	57.9%	11.1%	18.7%	61.8%	96.9%	75.5%	47.1%
Logistic Regression	61.4%	51.6%	39.9%	45.0%	69.7%	77.3%	73.3%	59.1%
Linear SVC Classification	58.0%	44.1%	42.0%	43.0%	70.7%	69.2%	69.9%	56.5%
NuSVC Classification	59.5%	56.0%	26.8%	36.2%	66.3%	77.4%	71.4%	53.8%
Decision Tree Classification	62.4%	59.3%	24.0%	34.2%	64.6%	92.1%	75.9%	55.1%
Random Forest Classification	59.4%	54.6%	22.7%	32.1%	60.6%	75.6%	67.3%	49.7%
Adaboost Classification	57.2%	43.1%	25.9%	32.4%	62.7%	84.2%	71.9%	52.1%

Chart View:


Obama Classification
80.0%
70.0%
60.0%
50.0%
40.0%
30.0%
20.0%
10.0%
0.0%
61%
57%
59%
56%
60%
47%
55%
53%
Multinomial Naïve Bayesian
Bernoulli Naive Bayesian
Logistic Regression
Linear SVC Classification
NuSVC Classification
Decision Tree Classification
Random Forest Classification
Adaboost Classification
Positive F score
Negative F score
Average F score


Romney Classification
80.0%
70.0%
60.0%
50.0%
40.0%
30.0%
20.0%
10.0%
0.0%
58%
47%
59%
56%
53%
55%
50%
52%
Multinomial Naive Bayesian
Bernoulli Naive Bayesian
Logistic Regression
Linear SVC Classification
NuSVC Classification
Decision Tree Classification
Random Forest Classification
Adaboost Classification
Positive F score
Negative F score
Average F score

Conclusion

We tried different algorithms and got different results with each of the algorithms. Then we started working on improving the preprocessing steps which improved the overall f-score slowly.

Thus, after analyzing the average f-score from the 10-cross validation we used

- Multinomial Naive Bayesian for Obama
- Logistic Regression for Romney.

References

1. Web Data mining: Exploring Hyperlinks, Contents and Usage Data by Bing Liu
2. https://www.omnicoreagency.com/twitter-statistics/
3. https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/
4. https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a/
5. http://www.nltk.org/book/ch06.html
6. http://stevenloria.com/how-to-build-a-text-classification-system-with-python-and-textblob/
7. https://github.com/daxiongshu/tradeshift-text-classification
8. http://nadbordrozd.github.io/blog/2016/05/20/text-classification-with-word2vec/
9. https://www.linkedin.com/pulse/short-introduction-using-word2vec-text-classification-mike/
10. http://fastml.com/classifying-text-with-bag-of-words-a-tutorial/
11. http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html