Final project report for data mining course

On

Political Social Media Posts

(Classify partisan bias, audience, and goal based on politicians' social media)

Xuechun Dong
Information Science
University of Pittsburgh
xud8@pitt.edu

Ashutosh Burde
Information Science
University of Pittsburgh
asb161@pitt.edu

Tianxing Le
Information Science
University of Pittsburgh
til57@pitt.edu

ABSTRACT

We are going to analyze the dataset which was from Crowdflower's Data For Everyone Library, provides text of 5000 messages from politicians' social media accounts, along with human judgments about the purpose, partisanship, and audience of the messages. This task requires us to classify partisan bias, audience, and goal based on politicians' social media. To solve this problem, we should analyze data, preprocess data, implement algorithms(classification) and evaluate the performance (choose best model).

For this problem, we are going to implement several classification algorithms on this dataset. For example, we'll implement naïve Bayesian classifier, logistic regression, K-means, decision tree, adaptive boosting(ensembles), support vector machine to classify partisan bias, audience and use network to improve the results. Besides, we use text mining methods to extract information from the 'text' data and do some extra sentiment analysis. But before we implement supervised learning algorithms, the first thing we need to do is preprocessing this text dataset and extract features from results we get.

KEYWORDS

Text Mining, Sentiment Analysis, Naïve Bayesian Classifier, Logistic Regression, K-means, Decision Tree, Adaptive Boosting(ensembles), Support Vector Machine, Social Media, Machine Learning, Network

1 Introduction

From several reading assignments, we deeply felt the power and importance of Text-mining. The analysis of Political Social Media Posts gives us a novel approach to access to the political predictions. For this dataset, the most challenging part is extracting proper features from dataset preprocessed. For us, mastering the knowledge is not enough in the field of data mining, especially text mining. This project requires our group to combine supervised learning algorithms with text mining, which means we should make good use of both of fields of knowledge. Furthermore, accurately predicting partisan bias and audience of a person according to his/her Political Social Media Posts can help people to build good connection with each other-people who have same biases may become good friends. Besides, we perform sentiment analysis and display word-clouds. Lastly, we evaluate the performance of the different models.

2 Related Work

The dissertation "Tracking Bias in News Sources Using Social Media: the Russia-Ukraine Maidan Crisis of 2013-2014" [1] illustrates how machine learning can solve the problem of extracting important features from dataset and how to do classification of unstructured data. Since all of us are not very experienced with analyzing unstructured data. We'll implement some data preprocessing methods mentioned in this paper.

Another Paper I want to mention is "A Framework for the Classification of Unstructured Data" [2], which will help us to have deeper comprehension of unstructured data analysis.

3 Data Preprocessing

Since some of our data are unstructured data, we must do the data preprocessing first and extract useful features before implementing supervised learning algorithms.

Firstly, we take a look at the summary of our data. Some features are not really important for classification and prediction tasks. So, we remove features we'll discuss below.

Secondly, some important features like labels and text, should be preprocessed before they are used as features to train models. We're planning to convert these texts to lowercase, remove

punctuations, numbers and repeated strings (some url addresses are same).

Finally, we can redefine new features from texts which we have preprocessed. (new strings related to important information, like attitude towards to a party, state where politicians are from, the most frequent words or sentences these politicians mentioned in their post, etc.)

3.1 Drop useless features

We dropped the 'orig_golden', 'audience_gold' and 'bias gold' because they were empty.

We dropped the 'X_golden' and 'X_unit_state' because they had the same value. They are useless to do the predictions.

We dropped the 'embed' because we already have had the information it contains. The 'embed' contains source, label and text. So we can just use these features separately and drop the 'embed' feature.

3.2 Keep the id features

Each politician has an id and label. Even though the id features make no sense in the prediction, we still decide to keep it(this id will not be used as feature, only for plots). Because we plan to use the label text and use the id for labels in plots because they are compressed.

3.3 Set weights

We use 'Bias_Confidence' as weight to 'bias' feature, 'Audience_Confidence' as weight to 'audience' feature, 'Message_Confidence' and 'Trust_Judgement' as weight to 'message' feature.

3.4 Extract states from 'label' feature

The 'label' feature contains the users' information (name and states), we think that the states should be a useful feature for our prediction, so we extract it from 'label' feature by text mining methods.

label <chr>
From: Trey Radel (Representative from Florida)
From: Mitch McConnell (Senator from Kentucky)
From: Kurt Schrader (Representative from Oregon)
From: Michael Crapo (Senator from Idaho)
From: Mark Udall (Senator from Colorado)
From: Heidi Heitkamp (Senator from North Dakota)

Figure 1: Label before text mining

label <fctr>
Florida
Kentucky
Oregon
Idaho
Colorado

Figure 2: Label after text mining

4 Methods

4.1 Supervised Learning Classification

4.1.1 K-nearest neighbor (kNN) algorithm

kNN is a lazy algorithm. There is minimal training data. Most of the data is in testing. The main principle is feature similarity.
kNN is used for classification. In some cases it is used for regression as well.

In this project, we implement three kNN models which k=1, k=3 and k=5. We choose 'message', 'bias', 'source' and 'label' to predict the 'audience'. We choose 'message', 'audience', 'source' and 'label' to predict the 'bias'.

4.1.2 Adaboost (Adaptive Boosting)

It is an Ensemble classifier. It chooses the training set based on the accuracy of the model of the previous iteration.
It is similar to random forest.

We choose 'message', 'audience', 'source' and 'label' to predict the 'bias'. We choose 'message', 'bias', 'source' and 'label' to predict the 'audience'

4.1.3 Logistic regression

In statistics is a widely used model. In a regression analysis is estimating the parameters of a logistic model.
A binary logistic model has a dependent variable with two possible values.

4.1.4 Naïve Bayesian classification

Naïve Bayes is a conditional probability model. Naïve Bayes classifiers can be trained effectively in a supervised learning setting.

4.1.5 Decision Trees

Decision Tree is a model of possible outcomes of a series of related choices.

A node branches into multiple outcomes. Each of those outcomes in turn branch out into other outcomes. Thus, the structure starts to resemble a tree.

We built the Decision Tree model as well as the pruned version. We choose 'message', 'audience', 'source' to predict the 'bias'. We chose 'message', 'bias', 'source' and 'label' to predict the 'audience'.

4.1.6 Support Vector Machines

The model is a classifier, that creates a hyperplane that separates the data points.
The model selects the plane that best optimizes the classification.

We built SVM models in all 3 kernels – Linear, Radial and Sigmoid. We choose 'message', 'audience', 'source' and 'label' to predict the 'bias'. We choose 'message', 'bias', 'source' and 'label' to predict the 'audience'.

4.2 Text Mining

4.2.1 Word-cloud

We use the word-cloud methods to solve our basic questions in Kaggle and it is the preprocessing part for the network mining part.

The word-cloud computes the word counts used in a particular text first and then create an image composed of words, in which the size of each word indicates its frequency and importance.

4.2.2 Sentiment analysis

I transform our text into tidy text format. The tidytext package contains several sentiment lexicons in the sentiments dataset. I only use the BING lexicon (from Bing Liu and collaborators) which categorizes words in a binary fashion into positive and negative categories.[3]

I use inner join to do the sentiment analysis of different bias, audience and massages texts.

4.3 Network Mining

We want to find out more information from word cloud. We built four networks showing relationships between high-frequency words. Digging out these relationships helps us predict politicians' bias and audience more accurately and iconically.

5 Evaluations

5.1 Supervised Learning

Before adding the weights:

Measures	Logistic Regression	Naïve Bayes	kNN1	kNN3	kNN5	Ada Boost	SVM Radial	SVM Linear	SVM Sigmoid	Dtree	Dtree Prune
Accuracy	0.5202738	0.5319827	0.3655925	0.3750347	0.3727505	0.6902179	0.5650913	0.5655923	0.5564001	0.6886426	0.6868706
Precision	0.3004756	0.3584355	0.2065723	0.1987507	0.1914207	0.5198443	0.4209959	0.4220138	0.4139017	0.5775204	0.5847179
Recall	0.5790751	0.5981758	0.5296369	0.4943697	0.4738383	0.6413701	0.6877468	0.6909636	0.6713627	0.4954477	0.4906747
F-Score	0.3526803	0.3729960	0.2923629	0.2753984	0.2626037	0.5067703	0.4452045	0.4473538	0.4341097	0.4231126	0.4144128
AUC	0.5818243	0.6122817	0.3685207	0.3515915	0.3355251	0.7870718	0.7210065	0.7242807	0.6988370	0.7540205	0.7556357

Table 1: Performance of bias

Measures	Logistic Regression	Naïve Bayes	kNN1	kNN3	kNN5	Ada Boost	SVM Radial	SVM Linear	SVM Sigmoid	Dtree	Dtree Prune
Accuracy	0.6666667	0.7049437	0.7987645	0.8382977	0.8522748	0.6666667	0.6796261	0.6769159	0.6761286	0.6666667	0.6666667
Precision	0.8971000	0.9680991	0.8081667	0.8495044	0.8772177	0.8971000	0.9702848	0.9699964	0.9705427	0.8971000	0.8971000
Recall	0.6666667	0.6658186	0.9791513	0.9697136	0.9515899	0.6666667	0.6313425	0.6282702	0.6266623	0.6666667	0.6666667
F-Score	0.9426485	0.7267276	0.8919558	0.9083566	0.9181959	0.9426485	0.6965710	0.6929848	0.6913617	0.9426485	0.9426485
AUC	1.0000000	0.9964093	0.5610795	0.7093721	0.8215013	1.0000000	0.9999973	0.9999983	0.9999968	1.0000000	1.0000000

Table 2: Performance of audience

After adding the weights:

Measures	Logistic Regression	Naïve Bayes	kNN1	kNN3	kNN5	Ada Boost	SVM Radial	SVM Linear	SVM Sigmoid	Dtree	Dtree Prune
Accuracy	0.5214684	0.5306991	0.3598382	0.3764035	0.3746541	0.6876887	0.5658275	0.5626364	0.5606439	0.6854431	0.6838612
Precision	0.3114240	0.3543580	0.2122226	0.1984270	0.1931322	0.5165219	0.4235606	0.4183025	0.4187495	0.5749022	0.5731375
Recall	0.5834797	0.6000601	0.5600367	0.4976782	0.4758823	0.6389229	0.6784502	0.6708214	0.6756256	0.4860232	0.4973303
F-Score	0.3554935	0.3728604	0.3033475	0.2755200	0.2646679	0.5022349	0.4431440	0.4360305	0.4383871	0.4094966	0.4161109
AUC	0.5874052	0.6117949	0.3767314	0.3506218	0.3408691	0.7863203	0.7140757	0.7054961	0.7063527	0.7511554	0.7566592

Table 3: Performance of bias

Measures	Logistic Regression	Naïve Bayes	kNN1	kNN3	kNN5	Ada Boost	SVM Radial	SVM Linear	SVM Sigmoid	Dtree	Dtree Prune
Accuracy	0.9786189	0.8881757	0.7986131	0.8336620	0.8513799	0.7339000	0.6792316	0.6797506	0.6773859	0.6666667	0.6666667
Precision	0.9949276	0.8854580	0.8079576	0.8460395	0.8795103	0.9063981	0.9699080	0.9701677	0.9706926	0.8971000	0.8971000
Recall	0.9790652	0.9928855	0.9793049	0.9683305	0.9472112	0.7500000	0.6313237	0.6316561	0.6280305	0.6666667	0.6666667
F-Score	0.9961799	0.9354805	0.8920015	0.9057844	0.9119970	0.9487776	0.6955326	0.6965359	0.6947852	0.9426485	0.9426485
AUC	1.0000000	0.9938661	0.5600782	0.7036624	0.8183541	1.0000000	0.9999831	0.9999895	0.9999853	1.0000000	1.0000000

Table 4: Performance of audience

5.2 Text Mining

5.2.1 Word-cloud

What words predict partisan v. neutral messages?

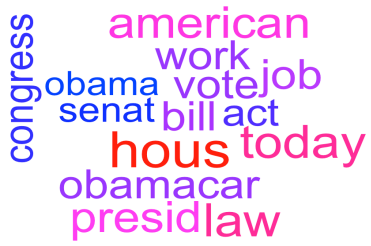

Figure 3: Wordcloud of partisan texts

Figure 4: Wordcloud of neutral texts

What words predict national messages v. constituency messages?

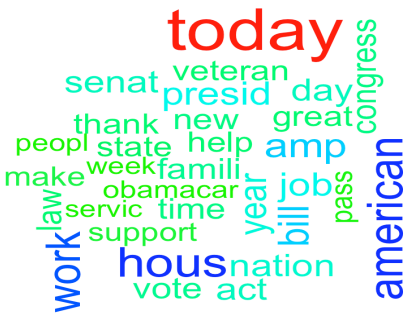

Figure 5: Wordcloud of national texts

Figure 6: Wordcloud of constituency texts

What words predict support messages v. attack messages?

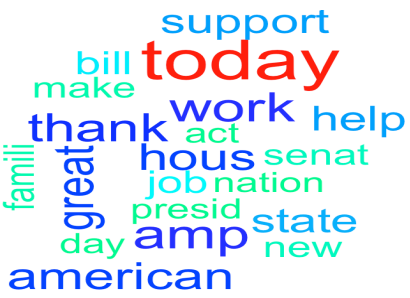

Figure 7: Wordcloud of support texts

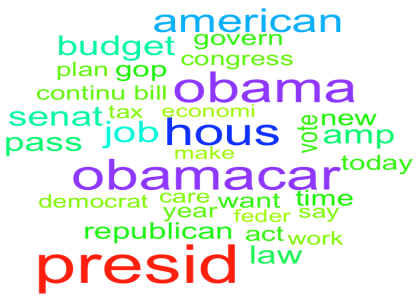

Figure 8: Wordcloud of attack texts

5.2.2 Sentiment analysis

I transform our text into tidy text format. The tidytext package contains several sentiment lexicons in the sentiments dataset. I only use the BING lexicon (from Bing Liu and collaborators) which categorizes words in a binary fashion into positive and negative categories. [3]

I use inner join to do the sentiment analysis of different bias, audience and massages texts and plot the sentiment scores.

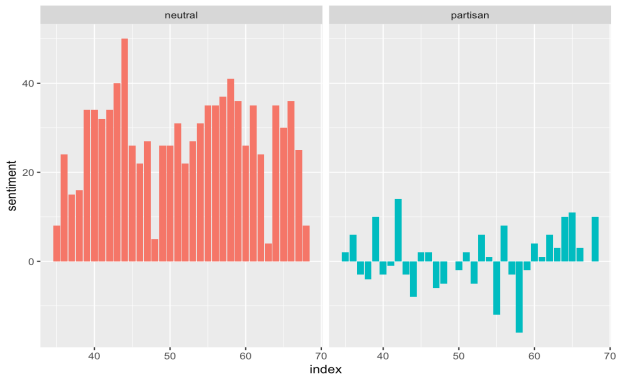

Figure 9: Sentiment scores of different bias texts

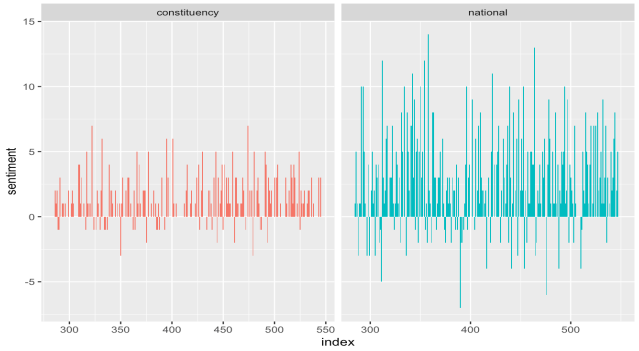

Figure 10: Sentiment scores of different audience texts

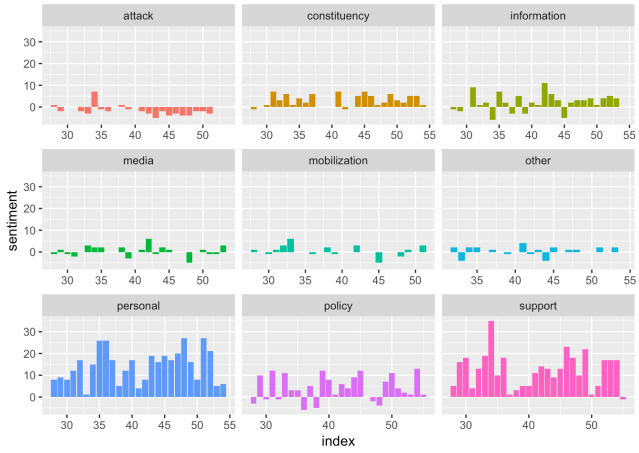

Figure 11: Sentiment scores of different kinds of messages

We can see different sentiment contribution in different kinds of text above. The partisan texts always have positive sentiment, however, the neutral texts have half positive and half negative sentiment. The difference between constituency and national texts is not obvious. For different kinds of messages, personal and support texts always have positive sentiment. However, the attack texts have negative sentiment.

Then we compute the word counts of different kinds of texts contribute to each sentiment. The result is here:

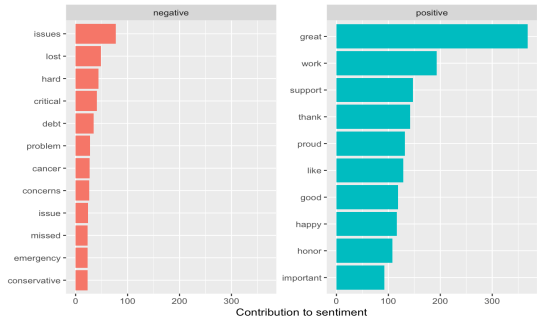

Figure 12: The most common negative and positive words in partisan text

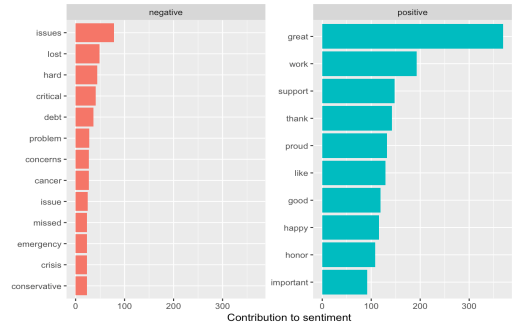

Figure 13: The most common negative and positive words in neutral text

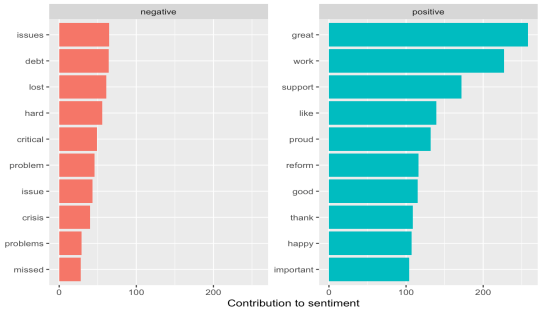

Figure 14: The most common negative and positive words in national text

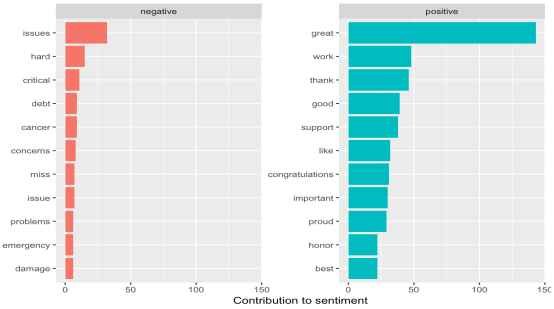

Figure 15: The most common negative and positive words in constituency text

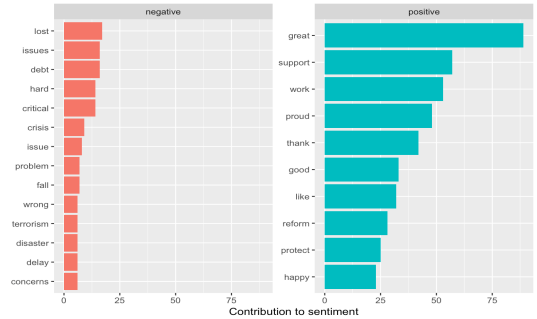

Figure 16: The most common negative and positive words in support text

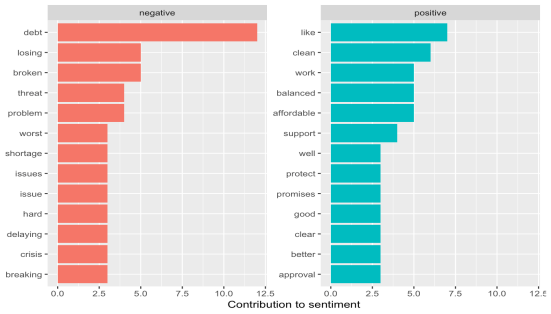

Figure 17: The most common negative and positive words in attack text

4.3 Network Mining

With different combinations of these words in word cloud, we can calculate the degree, betweeness or closeness of them and precisely make a conclusion whether this poster using these combinations in posts is partisan or neutral, as well as audiences of them are constituency or national. Four networks show below:

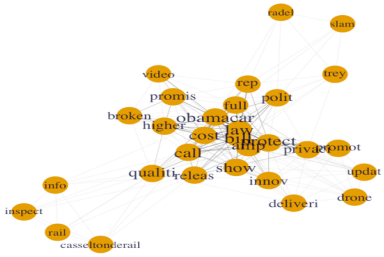

Figure 18: Partisan words network

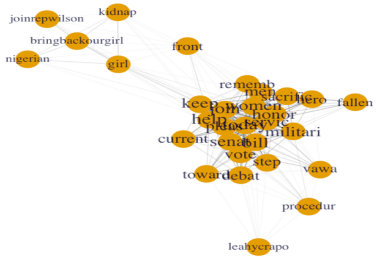

Figure 19: Neutral words network

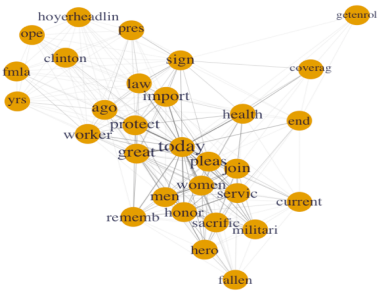

Figure 20: Support words network

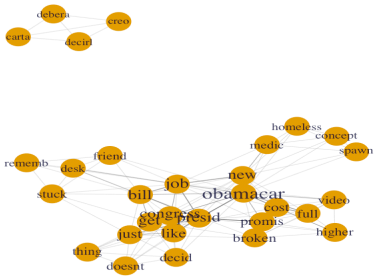

Figure 21: Attack words network

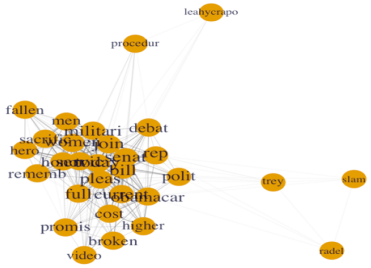

Figure 22: National words network

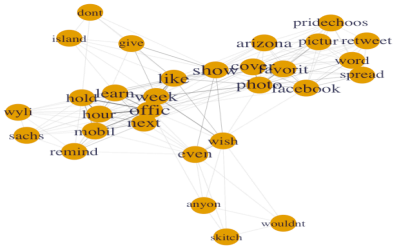

Figure 23: Constituency words network

Different sizes and depths of colors of lines, as well as sizes of fonts means how strong relationship between two words and how important this word is to predict bias or audience. For example, in attack words network, 'Obamacar' and 'presid' have largest sizes of fonts and lines with deepest colors between them, which means a document with words 'Obamacar' and 'presid' has very high probability of being a attack post.

DISCUSSION

As results show above, classification algorithm is not perfect.

Table 5(text mining-word cloud) shows the accuracy of words that used to predict bias, audience and goals (attack or support).

	accuracy	precision	recall	F1-score
partisandata	0.7242000	0.4873606	1.0000000	0.6553362
neutraldata	0.6162000	1.0000000	0.4798048	0.6484704
supportdata	0.6753529	1.0000000	0.3507058	0.5192926
attackdata	0.6453488	1.0000000	0.2906977	0.4504505
nationaldata	0.8966000	0.8848039	1.0000000	0.9388817
constituencydata	0.9328000	1.0000000	0.6734694	0.8048780

Table 5: Performance of wordcloud based on network mining

As table shows, word cloud used for predicting audience (national or constituency) performs very well (high accuracy, high precision, high recall and high F1-score). However, the performance of prediction for bias is not as perfect as we expected. To improve performance, we tried our best to set proper parameters (the number of high-frequency words chosen from word cloud, the degree of each word in network). Words with high degree in network always dominate the property of a post. For example, high-frequency word 'Obama' in a post indicates this post has high probability of being a partisan post. From this basic idea, we picked up 30 highest-frequency words whose degrees are above average value as key strings to predict corresponding posts. Table 3 shows performance of improved models:

	accuracy	precision	recall	F1-score
partisandata	0.7856000	0.5501469	1.0000000	0.7097997
neutraldata	0.6258000	1.0000000	0.4928165	0.6602506
supportdata	0.7285559	1.0000000	0.4571118	0.6274218
attackdata	0.6627907	1.0000000	0.3255814	0.4912281
nationaldata	0.9124000	0.9006577	1.0000000	0.9477327
constituencydata	0.9388000	1.0000000	0.7026239	0.8253425

Table 6: Performance of improved wordcloud based on network mining

To figure out the relationship between sources and goals, we made two lists of percentages of different goals based on source:

	attack	policy	support	information	media	constituency	mobilization	other	personal
percentage	0.0436	0.2668	0.1904	0.1316	0.0676	0.038	0.0296	0.016	0.2164

Table 7: Percentages of goals based on Twitter

	attack	policy	support	information	media	constituency	mobilization	other	personal
percentage	0.0436	0.2668	0.1904	0.1316	0.0676	0.038	0.0296	0.016	0.2164

Table 8: Percentages of goals based on Facebook

After comparison of different models and our calculation, we made two conclusions. First, word cloud is the best tool to predict audience and bias. Second, goals (messages) is not related to source.

CONCLUSION

We are by no means the first to classify politicians' bias and audience, as well as their goals based on their social medias. Posts are the ideal resources of choice for such tasks and has been applied to classification of different US senators or American politicians, text categorization, clustering, searching, as well as other similar problems. Under most conditions, posts reflect true thoughts of politicians, which means posts offer a large amount of information for mining and predicting politicians' properties. From this task, we are able to not only perfect a political information push system, but a basic friends-making recommendation system that helps people find friends in different domains with same political learnings.

In this paper, to achieve the best performance of prediction, we addressed the issue of predicting the bias and audience of posts, as well as goals based on social media. After implementing several basic classification algorithms on dataset preprocessed (without text information). We've found the limit of these classification algorithm on predicting posts. The best performance(accuracy) of model KNN5 for predicting audience is about 85% and the best performance of model adaptive boosting for predicting bias is about 69%, which doesn't even meet our basic expectations. To further improve our projects, it is important to consider more elements like 'states' from labels and 'political bias words' from texts. Word cloud is a good tool for presenting prediction. To predict goal based on social media, we abstractly calculate percentages of posts with different goals based social media. We find just a little portion of politicians and senators show their attack tendency either on twitter or Facebook. The large amount of posts are about policy or personal either on twitter or Facebook, which means goals of posts are not related to social medias.

Combining network mining and text mining analysis can help us predict bias and audience much more accurately. Network shows relationship of different words used for prediction. After our test and analysis, we find some combinations of words dominate the result of prediction. Contents of text and labels do help improve models. We find it is frequent for some combinations of words appear in posts belonging to same kind.

ACKNOWLEDGEMENT

We thank the Crowdflower's Data For Everyone Library for making the dataset available. We would also like to thank our Data Mining course professor, Dr. Yuru Lin for her guidance. And we thank our team members for finishing this project together.

CONTRIBUTION

Ashutosh Burde: To remove unnecessary features from the data frame. Building the 3 SVM models and the 2 Decision Tree Models. Combining all the classification models and collecting and analyzing the results. Separate analysis of results with and without adding weights.

Xuechun Dong: To do the data preprocessing. Extract the 'states' from 'label' text. Building the 3 kNN models and the Adaboost Models and made evaluations for two models. Implementing wordcloud and sentiment analysis on text mining. Making evaluations of the sentiment of different bias, audience and message and find the most common positive and negative words. Doing the final combination of report and codes.

Tianxing Le: Building logistic regression model and naïve Bayesian model on dataset preprocessed and made evaluations for two models. Building six networks for term-term matrices based on word clouds to find relationships between high-frequency words Making evaluations of models predicting different kinds of posts with high-frequency words from word clouds. Improving models

by adding degrees of networks as weights. Making comparison and discussion of original word cloud based model and improved model.

REFERENCES

[1] Peter Potash, Alexey Romanov, Mikhail Gronas, Anna Rumshisky, Mikhail Gronas. Tracking Bias in News Sources Using Social Media: the Russia-Ukraine Maidan Crisis of 2013–2014. W17-4203, 2017.

[2] David Alfred Ostrowski. A Framework for the Classification of Unstructured Data. 10.1109/ICSC.2009.48, 2009.

[3] Julia Silge and David Robinson, Text Mining with R, A Tidy Approach, https://www.tidytextmining.com/sentiment.html

[4] Osamu Yamakawa, Takahiro Tagawa, Koichi Yastake. Combining study of complex network and text mining analysis to understand growth mechanism of communities on SNS. EDM 2011- Proceedings of the 4th International Conference on Education Data Mining (pp. 335-336), 2011.