Detection of Insulting Comments in Online Discussion

Wasi Uddin Ahmad
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
wua4nw@virginia.edu

Md Masudur Rahman
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
mr5ba@virginia.edu

Abstract

In this project, we aim to detect the comments that may appear insulting to other participants in an online discussion or conversation. We will use text processing techniques to generate features and then we will use supervised learning techniques to detect insulting vs non-insulting comments from the text. We are planning to use naive Bayes and logistic regression to set a baseline. After setting the baseline, we are aiming to use models like support vector machine, random forests. Finally we will use adaptive ensemble classifier to maximize accuracy. In order to avoid over-fitting, we will tune and evaluate our model via K-folds cross validation. The dataset is obtained from the popular data science competition portal, Kaggle.

1 Introduction

With the prosperity of the Internet, ample amount of ways or spaces for public discussion has emerged which changes how do people communicate with each other. Either it is a section for comments or reviews for a news article or a forum to discuss some aspects of a particular product or event, these online discussion gives us opportunity to share our opinion and findings, as well as to know about the thoughts of others. People from different cultures and age group participates in such discussions. Sometimes the discussion topic is more sensitive than general case. Even though these discussions are expected to be productive, it allows people to post insulting or inappropriate comments.

As a result, these comments can hurt others feelings and often create a hostile or uncomfortable environment for some users, who might stop visiting the site in future. This problem is a serious one that website owners commonly face. Hence special focus need to be given on insulting comments of the online (ex. blog/forum) discussion. However, the comments containing insults but are targeted to a non-participant of conversation (like a celebrity etc.) are not marked as insults. Insults are of many types like: Taunts, reference to handicaps, improper language, slurs and racism which are aimed to attack the other person in an online discussion. While some other type of insults which mainly aim to embarrass the reader (not an attack) like crude language, provocative words, sarcasm, indirect reference [5].

Also, sometimes if we look for some information on some site and find insults then it leads to frustration. Though, the *Terms of Service* of social networking sites like Facebook, Twitter, Yahoo etc. prohibits from posting content that is unlawful, abusive and harassing, user' posts are only partially filtered for some particular collection of offensive words. Also, while some sites like YouTube, some newsgroups etc. provide flag facility to mark content as insulting/ inappropriate, they are prone to collusion and are highly misused (marking a non-insulting comment because it wasn't liked). Also, it is not possible to have a human moderator to review the comments before posting because of the increasing amount of online data.

An effective solution to mitigate this problem is to build a system that can detect whether a comment is insulting or not. With such a system, website owners would have a lot of flexibility in dealing with this problem. For instance, the owner could choose to automatically block or hide these insulting comments, or flag them so that they can more easily be found by site moderators. So, we are primarily interested in detecting comments that are intended to be insulting to other participants in an online discussion. The objective of this project is to do: build a machine learning system that can accurately classify online comments as insulting to other participants or not.

This report is organized in the following order. In Section 2, we are discussing previous solutions of our target task followed by the reasoning of using machine learning techniques to detect insulting comments in section 3. In section 4, the full pipeline of our proposed method is discussed which we have followed to build a strategy to get a prediction. Then we have described our experimental design and dataset in Section 5 followed by the description of evaluation metrics in section 6. In section 7, we have presented our experimental results. Then finally in section 8, we have concluded and discussed some future works of our project. In addition, section 9 and 10 covers the reasoning about why we were the right team for implementing this project and the description of our individual contribution in the project since we have worked in a group.

2 Previous solutions for the target task

Different attempts have been made to classify the insulting comments in online discussions. The work by Ellen Spertus [1] used static dictionary approach to build a feature vector for training but it suffers from high false positive rates. Another work by Altaf Mahmud et. al [2] used semantic rules but couldn't distinguish between the insults directed to non-participant and participant of conversation. The work by Razavi et. al [3] makes use of insulting and abusing language dictionary on top of bag-of words features in their proposed three-level classification machine learning model but they mostly depend on dictionary which is not easily available.

Another recent work by Carolyn P. Rose et. al [4] in classifying offensive tweets builds topical features and lexicon features using some dictionary and uses various machine learning approach. However, besides the limited approach of using seed words and pattern matching, the above mentioned works also do not distinguish between the insult directed towards the people participating in blog/forum conversation and non-participants such as celebrities, public figures etc [5]. Comments which contain obscene language or racial slurs may not necessarily be insulting to other person.

Insult directed towards the people participating in an online discussion is addressed in some of the recent works. Priya Goyal and Kalra [5] used support vector machine and logistic regression to train their model but there were some false positives. But the interesting feature that was applied by them was the inclusion of the count of words that followed the phrases such as "you are a", "you're", "you", "your". It was observed that this feature was a constant theme in abusive comments, as they were the most direct way to append an insult to one of these second person phrases. In their paper, Priya Goyal and Kalra discussed how they had originally had a classifier right around 82% accurate and with the inclusion of this feature it rose to 86%, as it was found to be a strong classifier.

In the work by Heh [6], logistic regression and stochastic gradient descent is used for classification and training their parameter vector respectively. As discussed by Heh, there were a number of things that increased accuracy to his project. These include, adding Google's bad word list [13] to count the number of bad words in the sentence as a feature, the inclusion Stanford's NLTK tokenizer and the Lancaster Stemmer. Such methods greatly increased the cross validation accuracy

Prashant Ravi experimented with SVM, Naive Bayes Multinomial, Random Forest, AdaBoostM1 classifiers and eventually found SVM, the best classifier in terms of the most number of correctly classified instances in his work [7]. In another work by Foley [8], they trained a variety of supervised learning algorithms to classify insults directed at other forum members. After meticulous text processing to generate features, they fit naive Bayes and regularized logistic regression to establish a baseline. Then they attempt to maximize accuracy with more complex models: random forests, SVM, and boosted regression trees.

So, in our work, we aim to build an efficient ensemble classifier excelling previous works to detect peer to peer insults involving machine learning approaches.

3 Why is this related to machine learning?

As we all know machine learning explores the study and construction of algorithms for learning to do tasks. The learning that is being done is always based on some sort of observations or data, such as examples (the most common case in this course), direct experience, or instruction. So in general, machine learning is about learning to do better in the future based on what was experienced in the past.

The emphasis of machine learning is on automatic methods. In other words, the goal is to devise learning algorithms that do the learning automatically without human intervention or assistance. Since we are trying to build an automated system to detect insults, we need to particularly investigate and analyze different techniques in the field of machine learning to come up with an efficient and accurate classifier. So, this work is entirely based on machine learning techniques.

4 Proposed Method

The basic strategy of our work is depicted in the following flow chart. The following sub-sections discuss each step in detail.

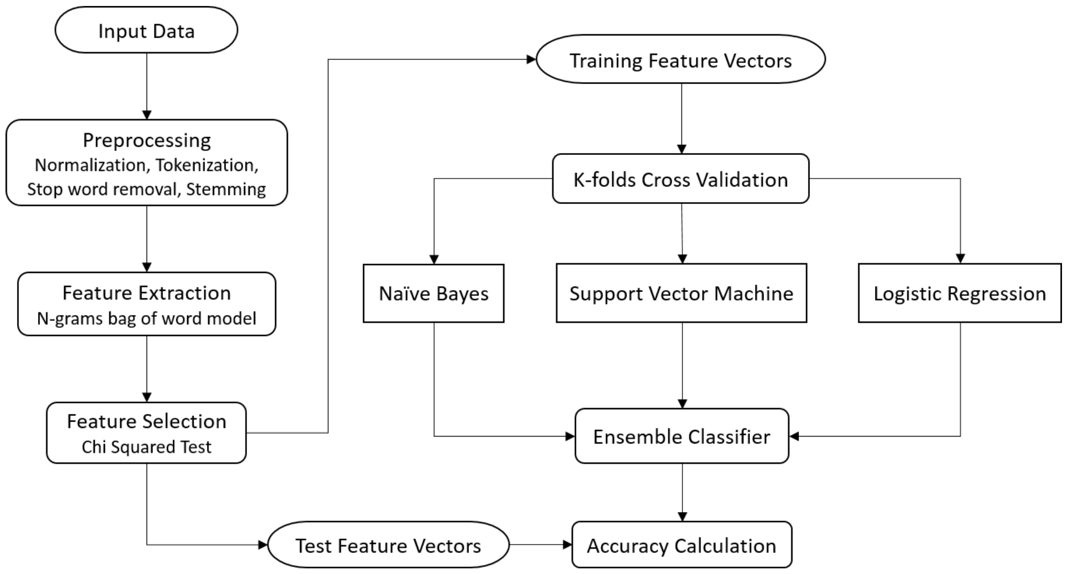

Figure 1: Flow chart of our classification approach

4.1 Preprocessing

The dataset obtained from kaggle cant be used directly for learning models. Some pre-processing of data is necessary and we need to convert it to the desired format of various machine learning algorithms. However, pre-processing should not lead to loss of information. So depending on the task, we have identified that the following pre-processing steps are required.

4.1.1 Normalization

We have used various techniques for this dataset after observing its nature. They are given below.

1. Removed punctuation symbols from the comments.
2. Removed unwanted strings like \\xa0, \\xc2, \\n, etc. and some unwanted html tags. These may put bias on the results if not removed.

4.1.2 Tokenization

Tokenization means slitting the text into tokens. Tokens can be characters, 'words' or n-grams which are the sequence of n consecutive words. We took words as tokens. So, far we have considered unigrams (words) and bigrams (2 consecutive words) for the feature vector construction. To construct feature vector we have used bag of word model.

4.1.3 Stop Words Removal

This is optional. We have tested our model with and without removing stop words. Stop words usually refer to the most common words in a language and they do not carry significant information. We are using a popularly used list of stop words: Smart system's stop word list [12]. Removal of stop words adds value if the training data set is large. In our case, since our data set is small and we are selecting the top k features through a feature selection process later on, we didn't remove the stop words.

4.1.4 Correcting Common Words

People prefer to write short forms of words like ”ur” for ”your”, ”nope” for ”no” etc. If we can convert these types of words to their original correct form, then it reduces the size of feature set and also improves the accuracy of models. Also, due to the flexibility in an online discussion, people often manipulates the dictionary while writing insult words like ”@$$hole” for ”asshole”, ”!d!ot” for ”idiot” etc. These words are necessary for the insult detection and so these must be identified my models correctly. We constructed a dictionary of approximately 500 words by combining [13] and [14] that contains all the possible ways an insult word can be written and when these words are encountered we convert then to their true form using this dictionary.

4.1.5 Stemming

Stemming involves reducing the words to their root or stem form like ”running” to ”run”, ”beautiful” to ”beauti” etc. For large dataset this is necessary because otherwise it results in unnecessary increase in the number of features. However, for our project, this is optional because this might result in the loss of information from dataset as it may wrongly reduce some words which effect insult detection to their root.

4.2 Feature Extraction

The strings should be converted to a numeric vector so that they can be used by machine learning algorithms. We use the various N-grams model (1 & 2 only) to construct the feature vector in terms of bag of word model. We have used the following two different measure to construct the feature vector. In future, we will consider 3 or 4 grams model to check our models accuracy.

4.2.1 Term Frequency

We have counted the frequency (number of times) of each token occurs in a comment. We constructed a (generally sparse) matrix of size N by V where N is the size of the training data (number of comments in our case) and V is the size of the vocabulary (the length of feature vector constructed over the whole training set using n-grams) representing all the text strings (comments in our case) where the number of occurrences of each token is a feature for that text string.

4.2.2 TF–IDF Weight

TF-IDF, short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. We constructed a similar matrix of size N by V using tf–idf weight as we did using term frequency also.

4.3 Feature Selection

Considering N-grams model, we are getting features in the order of a hundred thousand which is too big a number to be handled efficiently by algorithms like SVM and Logistic Regression. So we need to select a few best features from of our set of features. We have applied a statistical test known as "Chi Squared Test" [15] to our feature matrix to select best k number of features where k will be a parameter for our model.

4.4 Model Selection

After constructing the feature vectors and extracting the top features, we have applied naive bayes algorithm to set a baseline. We want to use top two classifiers from SVM, logistic regression, Decision tree, K-nearest neighbor, random forests [9] and AdaBoost after exploring their behavior over our training data. Based on our observation, finally we will develop an ensemble learning model consists of three different machine learning classifiers to use majority vote to predicted target labels. This way we will be able to achieve better accuracy. In order to avoid over-fitting, we will tune and evaluate our model via K-folds cross validation.

5 Experimental Design

In this section we are giving details of our data set. Then we will describe different parameter settings for the machine learning classifiers that we have used.

5.1 Details of the Data

We obtained data for training and testing from Kaggle [6], which is a popular website that hosts machine learning competitions. The training data set contains 3947 examples and the testing data set contains 2647 examples, each of which consists of the text of a particular comment and its desired label. A label of 1 represents an insulting comment, while a label of 0 represents a neutral comment. For instance, two examples from the training data set are:

- Text: "Either you are fake or extremely stupid...maybe both...", Label: 1
- Text: "We afford what we HAVE to afford, Marco.", Label: 0

In the training data, total 1049 of the examples are labeled as "insulting", while the remaining 2898 examples are labeled as "neutral". In the testing data, total 693 of the examples are labeled as "insulting", while the remaining 1954 examples are labeled as "neutral".

5.2 Parameter Settings

We have tested many machine learning classifiers with different parameter settings and eventually used the best three classifiers to form our ensemble model. Except Naive Bayes, SVM and Logistic Regression, all other learning techniques fell short to be considered for our ensemble model. We have observed the performance of Bernoulli, Multinomial and Gaussian naive bayes and eventually selected Multinomial naive bayes to be incorporated in our model.

For support vector machine, we have found liner kernel to be most effective. We have set the parameter value to 1 for moderate amount of regularization. In this parameter setting, SVM worked best with our data set. For logistic regression, we have set identical value for the parameter to have similar regularization. With that setting, performance of logistic regression is very good.

We got the best value of k = 3, when we tested k-nearest neighbor classifier with different k values. For k-nearest neighbor classifier, we have used tf-idf weight to form the feature vector and used cosine similarity as distance measuring metric. But the performance of the classifier was not satisfactory, eventually we didn't consider it for our ensemble learner. We also implemented decision tree with different combination of parameter values but we found that decision tree is not suitable for our purpose.

Since our model is based on ensemble paradigm, we have also implemented random forest and adaBoost algorithm to compare the performance of our model and those classifiers. Our ensemble model outperformed all those ensemble model by good margin.

6 Evaluation Metrics

To evaluate our proposed method, we have used accuracy can be described as the proportion of the total number of predictions that were correct. It can be calculated using the following equation:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

where

$TP = True\ Positive,\ TN = True\ Negative, FP = False\ Positive, FN = False\ Negative$

Besides, we have used use F-measure which is also a measure of a test's accuracy. It is also known as F_1 score or F-score [7]. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F_1 score can be interpreted as a weighted average of the precision and recall, where an F_1 score reaches its best value at 1 and worst at 0.

The traditional F-measure or balanced F-score (F_1 score) is the harmonic mean of precision and recall. It can be calculated using the following equation:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

7 Experimental Results

We have used Naive Bayes, SVM and Logistic Regression to build our ensemble classifier. Though we have considered other effective machine learning classifiers like decision tree, k-nearest neighbor but none of them was accurate enough to compete with our model. Naive Bayes, SVM and Logistic Regression are effective methods to classify text data. For the better performance of SVM and Logistic regression, we find out the top k features from the bag of word model. For different number of features, we have compared our ensemble classifier's accuracy with the accuracy of Naive Bayes, SVM and Logistic Regression. Figure 2 shows our findings.

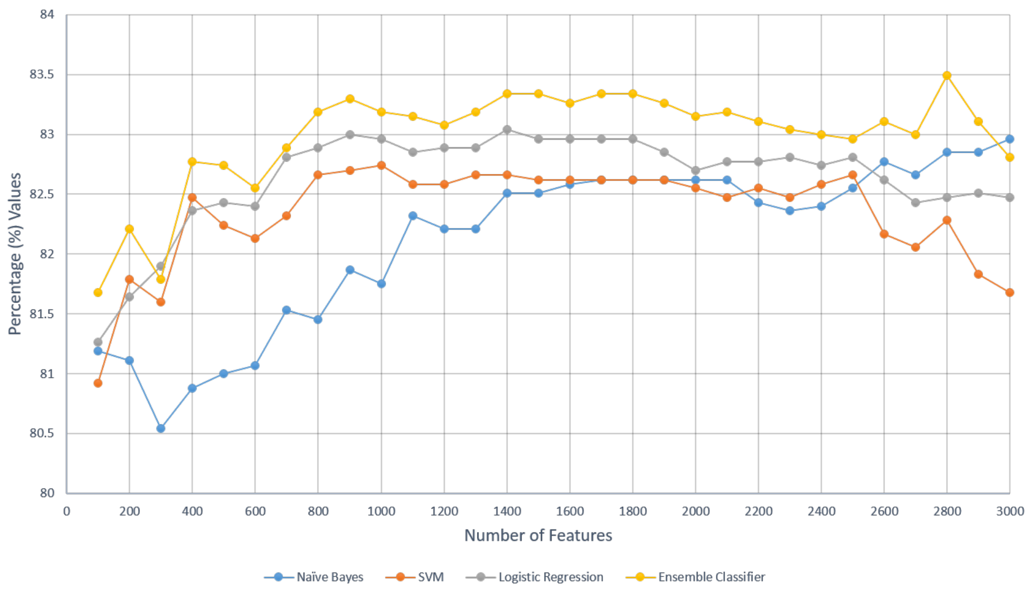

Figure 2: Our Ensemble Model vs. Naive Bayes vs. SVM vs. Logistic Regression

Almost with any number of features, our proposed ensemble model works better than those individual classifiers. But the difference between their performance is marginal. We have also implemented random forest and adaBoost algorithm to get check the performance with the data set. We have compared our ensemble learner with random forest and adaBoost. In both cases, our model outperformed them by significant margin. Our findings are depicted in figure 3.

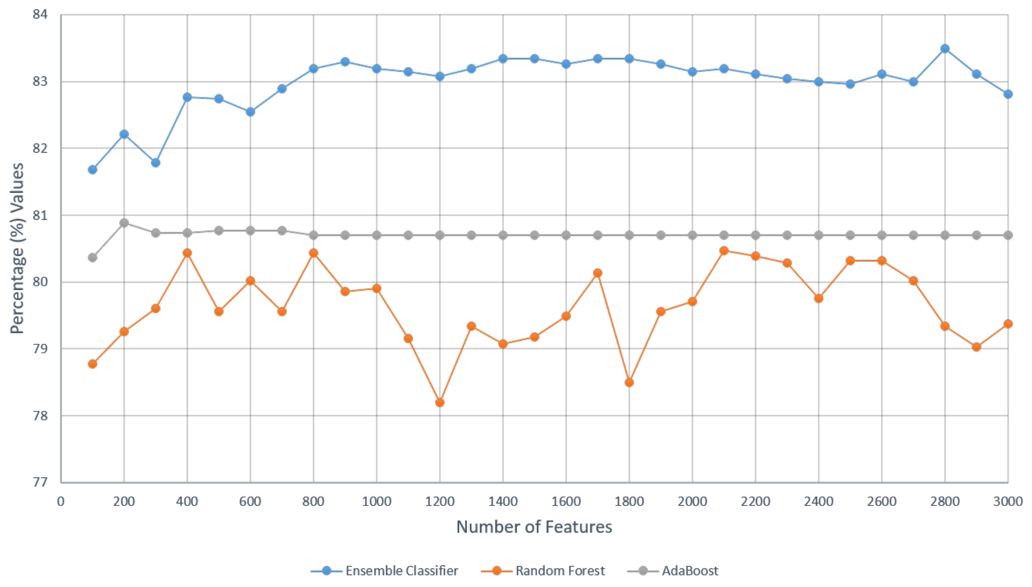

Figure 3: Our Ensemble Model vs. Random Forest vs. AdaBoost

We have used miss-classification accuracy to measure the effectiveness of our proposed model. But as we mentioned previously, we have evaluated our model through precision, recall and F score. The value of these evaluation metric is depicted in figure 4.

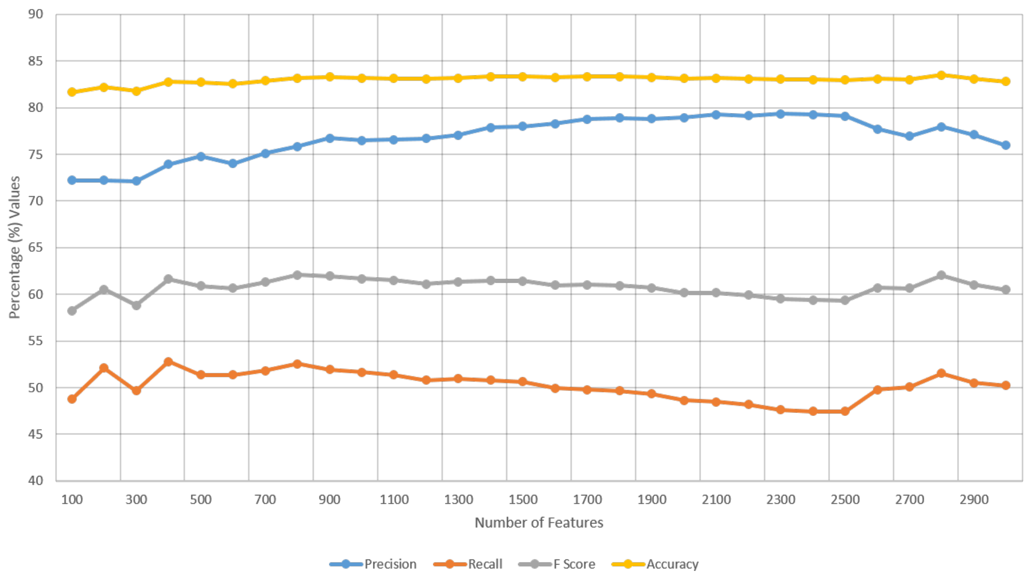

Figure 4: Accuracy, F score, Precision and Recall of our model

In all the previous works, everybody considered only the miss-classification accuracy. We did check that for precision, recall and F score. Precision of our model is quite good which represents that our model can correctly classify the insulting comments though there are few false positives. But we believe with some improvement measure, our model will be able to perform even better.

8 Conclusion and Discussion

We have tried to investigate how to detect insults, as define in the introduction, that appear in online discussions. To deal with it, we have used some knowledge from machine learning field. We have tested many classifiers to observe their nature on the training data set and eventually, we developed an ensemble classifier by combining our predictions from three different classifier via voting. We get significant improvement over some of the previous works but not all of them. We need to find better way to find significant features.

Though we were unable to excel the previous works by margin but it leads us to generate some idea for future work as improvement. We think the most effective area for future work is in gathering new types of features. For example, we could get user based features such as location, post history, and past comments. This might allow us to construct a profile of each user, to classify users themselves as toxic. Additionally, it would be useful to follow a conversation in its entirety to look at characteristics like thread length, number of replies to a comment, and nested comments.

Besides all the previous works in this regard didn't considered any unseen words during testing and we did the same. In future, we will try to use language models for smoothing, so that we can handle the unseen words during testing. We need to consider 3 and 4 grams also to get better accuracy. Moreover, using some natural language processing techniques, we can extract underlying semantic meaning of the comments posted in online discussion.

9 Why we were the right team for implementing this plan?

We have done this project for our machine learning course. We have learned machine learning techniques and how they can be employed in different applications. To grasp these concepts cogently, we need to implement some of these techniques in a practical work. Through this project, we have learned how ensemble classifier works and some of the well established machine learning techniques like support vector machine or logistic regression to handle text data. We have explored different aspects of these learning techniques and did some research how to exploit those techniques in solving the issue that we have addressed in this project. With the implementation of this project, we have improved our breadth in the field of machine learning. So considering all these facts, we believe as a team we were the right choice to implement this plan.

10 Individual Contribution

Our individual contribution is illustrated in the following two subsections.

10.1 Contribution of Wasi Uddin Ahmad

My contributions are as follows.

1. From text preprocessing to feature extraction in terms of bag of word model, everything is done by me.
2. Implemented Bernoulli, Multinomial and Gaussian Naive Bayes classifier and compared their performance to select the best approach.
3. Implemented Support Vector Machine with different kernel and hyper parameters and compared their performance to select the best parameter setting
4. Implemented Decision Tree classifier with different parameter settings
5. Implemented AdaBoost and Random Forest classifier and compared their performance with our proposed model
6. Worked on experimental analysis and result formulation.
7. Contributed in project proposal writing, mid phase and final report writing
8. Contributed in preparing presentation also.

10.2 Contribution of Md Masudur Rahman

My contribution is highlighted below.

1. Used the Chi Squared test and selected the top k features to train our model.
2. Implemented Logistic Regression and tuned it parameter to get better testing accuracy for our model.
3. Implemented K-nearest neighbor classifier and observed accuracy with different k values.
4. Contributed in experimental analysis and result formulation.
5. Contributed in project proposal writing, mid phase and final report writing
6. Contributed in preparing presentation also.

Acknowledgements

We would like to thank Professor Yanjun Qi to introduce us with the amazing field of machine learning through the course CS 6316. We have explored many supervised and unsupervised machine learning techniques and used some of them in solving the problem that we have addressed in this report. Also we would like to thank both the teaching staff for providing support and advice on this project.

References

[1] Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence, pages 1058 - 1065.

[2] Altaf Mahmud, Kazi Zubair Ahmed, and Mumit Khan, 2008. Detecting flames and insults in text. In Proceedings of the Sixth International Conference on Natural Language Processing.

[3] Amir H. Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010 Offensive language detection using multi-level classification. In Proceedings of the 23rd Canadian Conference on Artificial Intelligence, pages 1627.

[4] Xiang, G., Hong, J., & Ros, C. P. (2012). Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus. In Proceedings of The 21st ACM Conference on Information and Knowledge Management, Sheraton, Maui Hawaii, Oct. 29- Nov. 2, 2012.

[5] Dr. Amitabha Mukherjee, Priya Goyal and Gaganpreet Singh Kalra. 2013. Peer-to-peer insult detection in online communities.

[6] Kevin Heh. 2013. Detection of insults in social commentary.

[7] Prashant Ravi. 2014. Detecting Insults in Social Commentary.

[8] Foley, Ryan and Kasper, Ben and MacNguyen, Robert. 2012. Text Mining to Detect Insults in Online Discussion.

[9] Breiman Leo, 2001. Random Forests. Machine Learning 45(1):5-32.

[10] Forman George, 2003. An extensive empirical study of feature selection metrics for text classification. The Journal of Machine Learning Research 3:1289-1305.

[11] For dataset: www.kaggle.com/c/detecting-insults-in-social-commentary/data

[12] Smart system's stopword list – http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

[13] Badwords list: www.urbanoalvarez.es/blog/2008/04/04/bad-words-list

[14] Dubs, Jamie. "Google's Official List of Bad Words." Free Art & Technology. N.p., 28 July 2011. Web.

[15] Chi–squared test: https://en.wikipedia.org/wiki/Chi-squared_test

[16] F-score: https://en.wikipedia.org/wiki/F1_score