Article Comment Classifier

[1]Mohit Sharma
[1]Author
[1]SRM University

Abstract - **The project aims to develop an article comment classifier system by the use of machine learning techniques. An article comment classifier takes all the comments from an article and filters out the spam comments only showing high quality and genuine comments in a format readable by both websites and apps(API). We intend to train a classifier on a training set and apply it on a test set i.e. comments provided by users and then predicting the labels and producing the desired output. We will be using a Supervised machine learning algorithm and Multinomial Naive Bayes for prediction.**

keywords **- Machine Learning, Classification, Spam detection, Data Prediction, Websites, APIs, comments**

Introduction

A website should be able to "know" how good or useless a comment is posted, so it can be brought to users' attention if it is good or it can be hidden otherwise. So, our goal was to design a machine learning algorithm that trains itself on comments from multiple comments then given a sample test comment, predicts what the rating and modifier of that comment is.

Algorithms

1.**Multinomial Naive Bayes** : Naive Bayes classifier for multinomial models. The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work. We used our own python implementation of Multinomial Naive Bayes . We trained our classifier with a dataset of 5573 comments and tested our algorithm on a variety of articles.

Analysis:

a)Testing and Training with our own dataset,

	Case 1	Case 2	Case 3	Case 4
Train	5545	5249	2786	1671
Test	28	279	2787	3902

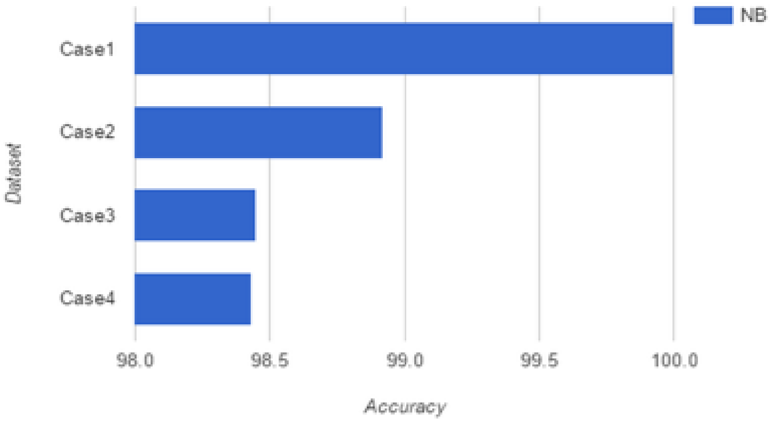

Fig 1(a).

We used Countvectorizer, which is Python's implementation of the Bag of Words model. We split our dataset into test and training data using sklearn's Test train split and tested on 28 comments and so on. We increased the size of our test dataset and got the results as in Fig. 1(a).Predicting time increased and training time decreased. In the first case we got the maximum accuracy and the least predicting time that is 0.0 sec .

b)Testing with Chalkstreet dataset and training with our own dataset.

Chalkstreet is an Online learning platform having a variety of courses. Every course has its reviews based on the user's experience. We exported the Reviews table in CSV format and tested our classifier that we trained using our own dataset of 5573 comments with this data. We got the following results as shown in the table 1(b). We got accuracy almost close to 1 when we used a test set of 4973 reviews.

	Case 1	Case 2	Case 3	Case 4
Train	5573	5573	5573	5573
Test	100	300	500	1000
Accuracy	0.94	0.98	0.98	0.98

c)Testing and Training both on Chalkstreet dataset.
To further test our algorithm's accuracy. We decided to train our classifier with Chalkstreet review dataset and tested on a bunch of comments. We increased the size of the test set in each case. The results we got are as shown in table 1(c).We got an accuracy of 0.96 of our algorithm for Chalkstreet review dataset.

	Case 1	Case 2	Case 3	Case 4
Train	4887	4666	1473	2456
Test	25	246	3439	2456
Accuracy	0.96	0.97	0.97	0.97

1(c).

To check if using some other classifier we can attain better accuracy than Multinomial NB , we used various Supervised Machine Learning algorithms.

2)**Support Vector Machine(SVM)**:

Support Vector Machines are a set of Supervised Learning methods used for classification , regression and outliers detection. SVC and Linear SVC are classes capable of performing multi-class classification on a dataset. The advantages of support vector machines are that it is effective in high dimensional space. SVM does not directly provide probability estimates. We used our own python implementation of Support Vector Machine . We trained our classifier with a dataset of 5573 comments and tested our algorithm on different numbers of comments. We changed the size of the test dataset exactly like 1(a) and got the following results as shown in figure 2(a).Maximum accuracy we reached in this case was approximately 0.99.

	Case 1	Case 2	Case 3	Case 4
Train	5545	5249	2786	1671
Test	28	279	2787	3902

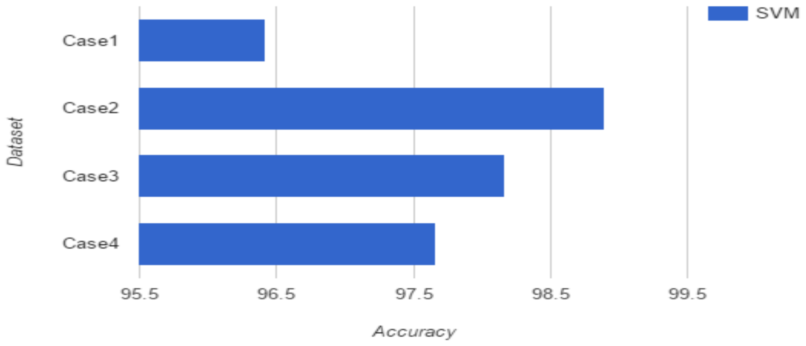

Fig 2(a).

3) **K- Nearest Neighbors** :

The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbor learning), or vary based on the local density of points (radius-based neighbor learning). The distance can, in general, be any metric measure: standard Euclidean distance is the most common choice. Neighbors-based methods are known as non-generalizing machine learning methods, since they simply “remember” all of its training data (possibly transformed into a fast indexing structure such as a Ball Tree).

For the simple task of finding the nearest neighbors between two sets of data, KNN can be used. We used our own python implementation of Nearest Neighbors . We trained our classifier with a dataset of 5573 comments and tested our algorithm on different numbers of comments. We changed the size of the test dataset exactly like 1(a) and got the following results as shown in figure 3(a).Maximum accuracy we reached in this case was approximately 0.93 which is very less as compared to Multinomial NB and Support Vector Machines.

	Case 1	Case 2	Case 3	Case 4
Train	5545	5249	2786	1671
Test	28	279	2787	3902

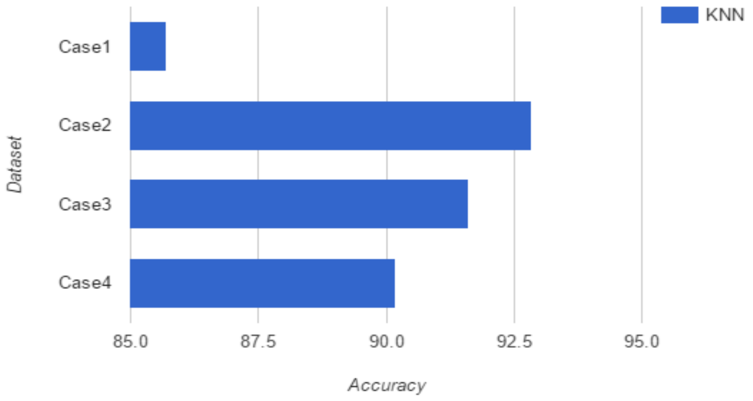

Fig 3(a).

Conclusion

From Figures 1(a).,2(a).,3(a) we can draw a bar chart and compare the difference in accuracy between the three classifiers that are Multinomial Naive Bayes,Support Vector Machine and K Nearest Neighbor.

	Case 1	Case 2	Case 3	Case 4
Train	5545	5249	2786	1671
Test	28	279	2787	3902

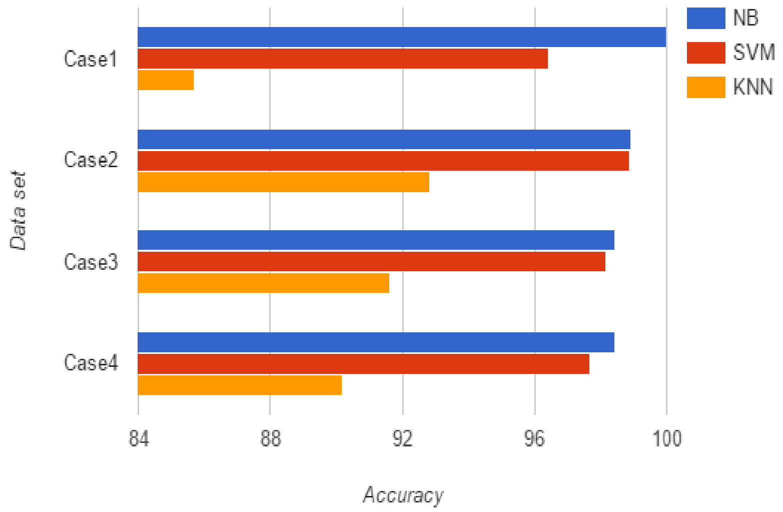

Fig 4(a).

From Figure 4(a). we see that Multinomial Naive Bayes gives the maximum accuracy to our text classification algorithm. Support Vector Machine also gives good accuracy .K Nearest Neighbor gives the least out of the three in each case. Naive Bayes classifiers are a popular choice for classification problems. It outperforms any other algorithm such as in our case where the degree of class overlapping is small and as long as we assume features are independent which is the reason it is called “Naive”.

Future work

Meta Features: We observed some patterns that describe the nature of a comment.

1. Users having more experience posting comments on the platform are less likely to produce bad comments.
2. Good and bad comments are segregated by user ratings.
3. Comments having many special characters are more likely to be characterized as bad comments.

4. Comments having URLs are more likely to be bad as they contain advertisements links.
5. If we switch to a new data set .These meta features will be very helpful in giving good accuracy.

Keywords
1) API : Application Programming Interface.
2) NB : Naive Bayes
3) tf-idf : Term Frequency-Inverse Document Frequency
4) SVM : Support Vector Machine
5) KNN : K Nearest Neighbor
6) CSV : Comma Separated Values
7) SVC : Support Vector Machine Classifier

References:

[1] Internet article comment classifier Matt Jones, Eric Ma, Prasanna Vasudevan, Stanford CS 229 – Professor Andrew Ng, December 2008
[2] Text Categorization with Support Vector Machines: Learning with Many Relevant Features Thorsten Joachims Universitat Dortmund Informatik LS8, Baroper Str. 301 44221 Dortmund, Germany
[3] WIKIPEDIA : https://wikipedia.org
[4] SKLEARN : http://scikit-learn.org