# Classification of Online Pernicious Comments using Machine Learning

**Aniket L Sulke[1] Akash S Varude[2]**
[1,2]Researcher
[1,2]MIT Academy of Engineering, Pune, India

*Abstract*— Internet has become the biggest platform to represent our skills. Various websites allow people to use their platform to showcase their skills through videos, articles and other information in different formats. Most of the websites provide facility of commenting on any of uploaded information. However, there is possibility that people can use abominable language in their comments. This paper mainly focuses on identification of these inadequate comments and classification of these into different categories. The required data is taken from machine learning site 'Kaggle' (www.kaggle.com). The comments are classified into 6 different categories- toxic, severe toxic, obscene, threat, insult and identity hate. We have used four different machine learning algorithms- logistic regression, support vector machine (SVM), K nearest neighbour and decision tree. All these mentioned models successfully predict classes of the comments. Out of these models support vector machine gives best result.

***Keywords:*** Abominable Language; Machine Learning; Comment Classification; Logistic Regression; Support Vector Machine; Social Media

## I. Introduction

Now a day's data on internet is increasing day by day. The growth is exponential. This blast of data is contributing development of new machine learning algorithms. One of the emerging trend in machine learning is text analysis. Machine learning has opened various doors for researchers in text analysis. Text classification is one of them which means a task of classifying text into different predefined categories [1]. The origin of text classification goes back to early 60's while machine learning approaches was successfully applied in 90's.

Initially text classification used for limited purpose. Further its scope increased from English language and spread over multiple languages in the world moreover its application use also increased. Classification of task has several applications, including filing patents into patent directories, selective dissemination of information to information consumers, automated population of hierarchical catalogues of Web resources, spam filtering, identification of document genre, authorship attribution, automated survey coding, and even automated essay grading [2].

Nowadays every social media sites and applications use machine learning approach. Machine learning has simplified the task that may take long duration to complete without it. Most of the approaches require text analysis and classification techniques [4,5,6]. These sites helps to establish communication between people all over the world. Many sites provide comment section allowing the user to comment their opinion, so that corresponding people can use these opinions for improvement. As every coin has two sides, some irresponsible people are using them in negative ways. It is found that many people using this platform in some terrible ways. There cases of online threating to people as well as use of rubbish words to people or society.

The threat of harassment and abuse online means that many people stop expressing themselves and give up on seeking different opinions.

Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments.[3] Hence this type of activity is unacceptable and those comments must be prevented for posting to prevent possible harm to the society.

Classification of the comments is necessary before posting on online platform. This paper discuss different methodologies like logistic regression, k nearest neighbor [7], support vector machine and decision tree for comment classification into 6 different categories viz. toxic, severe toxic, threat, obscene, identity hate and insult.

## II. Methodology

The multi-label classification of comments needs to be performed. The process methodology consists of the following steps - dataset description, text pre-processing, solving multi-label classification problem, feature extraction, model training.

### A. Dataset Description

The dataset used for classification consisted of Wikipedia comments which have been labeled by human raters for toxic behaviour. There are total 1,53,165 comments labeled as toxic, severe toxic, obscene, threat, insult and identity hate. Each comment has an output of 0 or 1 for each class.

### B. Text Pre-Preprocessing

The efficiency depends upon the pre-processing of data as pre-processing reduces ambiguity in feature extraction. Hence data pre-processing in an important step in text classification. The pre-processing includes conversion of all text to lowercase along with removal of special characters, numbers, punctuation marks, URLs, usernames and stop words.

### C. Feature Extraction

It is easy for human to classify images or text, but it is difficult for computers, which deal only with numbers, and to be more accurate, they process numbers in the form of electrical impulses. Therefore, any data must be converted into that form so computer can process it and give us back the result. So feature extraction plays important role in processing text. So before training the model we must vectorize the input data.

Feature extraction can be done by applying the Term Frequency Inverse Document Frequency (TF-IDF) using n-gram Features- Unigram (1-word) and bigram (2-words) to find the weight of a particular feature in a text. Thus, features are filter based on the maximum weight.

#### 1) TfidfVectorizer

It is used to convert a collection of raw documents to a matrix of TF-IDF features. It performs the task of CountVectorizer followed by TfidfTransformer. TF-IDF is not a single method, but a class of techniques to calculate the similarity between queries and documents.

It computes word counts, TF values and IDF values all at once and then gets Tfidf scores of a set of documents. TF-IDF is a numerical statistic that reflects the value of a word for the whole document (here, comment). The TF values, IDF values and the Tfidf score is calculated using the following formulas:

TF(w) = (Number of times term w appears in a document) / (Total number of terms in the document)

IDF(w) = log_e(Total number of documents / Number of documents with term w in it)

Tfidf score = TF(w) * IDF(w)

The max_df, max, ngram_range, min_df, stop_words are some parameters that can be passed to the TfidfVectorizer.

*2) N-gram Features*

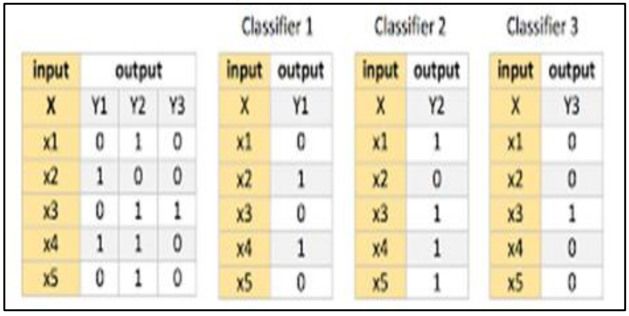
When n-gram feature is applied, more than one word is considered at a time. This is advantageous as many time some words convey sentiments more effectively when used together. A n-gram of size 1 is referred to as a unigram; size 2 is a bigram. N-grams captures more context around each word. N-grams almost always boosts accuracy. The N-gram frequency method provides an inexpensive and highly effective way of classifying documents.

### *D. Solving Multi-label Classification Problem*

Each instance in a dataset is assigned to multiple categories, so this is a multi-label classification problem. This problem can be solved by using methods which are listed below.

*1) Problem Transformation*

Traditional single label classification methods are concerned with learning from a set of examples that only correlated with a single label y from a set of disjoint labels. Problem transformation method is widely used to transform multi-label learning task into one or more single label learning tasks. Each independent feature is mapped to more than single dependent label, the multi-label problem is decomposed into several independent binary classification problems, one for each label which participates in the multi-label problem. This method is called binary relevance. This method assumes that each single label is independent of other. The problem is broken into N binary classification problems with N being the total number of Labels (Figure ). Each one of the binary classifiers predicts if the label belongs to the sample or not. The final result for the multi-label classification is determined by aggregating the classification results from each binary classifier.



|  |  |  |  | Classifier 1 |  | Classifier 2 |  | Classifier 3 |  |
|---|---|---|---|---|---|---|---|---|---|
| input | output |  |  | input | output | input | output | input | output |
| X | Y1 | Y2 | Y3 | X | Y1 | X | Y2 | X | Y3 |
| x1 | 0 | 1 | 0 | x1 | 0 | x1 | 1 | x1 | 0 |
| x2 | 1 | 0 | 0 | x2 | 1 | x2 | 0 | x2 | 0 |
| x3 | 0 | 1 | 1 | x3 | 0 | x3 | 1 | x3 | 1 |
| x4 | 1 | 1 | 0 | x4 | 1 | x4 | 1 | x4 | 0 |
| x5 | 0 | 1 | 0 | x5 | 0 | x5 | 1 | x5 | 0 |

Fig. 1: An example of transforming multi-label classification problem into binary classification problem

Another approach to solve multi-label classification using problem transformation method is classifier chain. In this method the first classifier is trained on the input data that is nothing but the comments and then each next classifier is trained on the input space and the previous classifier. To use this method we should first check the correlation between the labels. Pearson's correlation coefficient is used to calculate the correlation between the labels x and y. Classifier chain is more effective and accurate than the binary relevance when the two labels are highly correlated. The formula to calculate Pearson's correlation coefficient is given as:

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$

*2) Algorithm Adaptation*

The algorithms are adapted to directly perform multi-label classification, instead of transforming the problem into different subsets of problems. We have used the multi-label version of KNN, represented by MLkNN and decision trees.

### *E. Model Training*

The process of training the model involves providing the learning algorithm with training data to learn from. To train the model Support Vector Machine (SVM), Logistic Regression and KNN algorithms were used. These algorithms are described briefly as follows:

*1) Support Vector Machine (SVM)*

Given a labeled training data the algorithm outputs the optimal hyper plane which categorizes new input data. It works efficiently on large datasets without too much computation. To maximize the margin between the data points and hyper plane hinge loss function is used. The SVM algorithm implemented in practice using kernel. The hyper plane is learn in linear SVM by transforming the problem using some linear algebra.

*2) Logistic Regression*

Logistic regression is a simple classification algorithm. We try to predict the probability that it belongs to "0" class or "1" class. It uses a logistic function that always returns a value between 0 and 1 to model a binary dependent variable. The logistic function is given by:

$$f(x) = \frac{1}{1+e^{-(x)}}$$

If the value is above the threshold, the class is predicted as "1" class else it is predicted as "0" class.

*3) K-Nearest-Neighbour (KNN)*

The KNN algorithm won't learn in training phase, it only save all the training examples. At the time of prediction, for test instance xt the algorithm finds the training example xi, yi which is closest to xt. Euclidean distance is used to calculate the distance between the test data and each row of training data. The formula to calculate the Euclidean distance between two data points v and u is given by:

$$d(v,u) = \sqrt{\sum_{i=1}^{n} |v_i - u_i|^2}$$

K neighbours are selected which are having the shortest distance from the test data and voting is carried out in order to decide the output.

### 4) *Decision Tree*

It is flow-chart like tree structure, where each internal node denotes a test on an attribute, each branch denotes an outcome of test, and each leaf node holds a class label. The topmost node in a tree is the root node. Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. The learning and classification steps of decision tree induction are simple and fast. Decision tree is useful because construction of decision tree classifiers does not require any domain knowledge.

Their representation of acquired knowledge in tree form is easy to assimilate by users. Decision tree classifiers have good accuracy. The problem of constructing a decision tree can be expressed recursively. First, it is necessary to select an attribute to place at the root node, and make one branch for each possible value. The attribute with a highest information gain is selected as root node. Information gain is a measure of the effectiveness of the attribute in classifying the training data.

When complexity of tree increases over fitting may occurs, which in turn reduces the accuracy of test data. To avoid over fitting pre-pruning and post-pruning methods are used. In pre-pruning, we stop early (while growing the tree) if gain is not statistically significant. While in post-pruning we grow full tree and remove the nodes based on cross validation.

## III. RESULT

In this section we report the results of the experiments, i.e. the performance of the machine learning algorithms when applied to the dataset.

### A. *Accuracy Score*

To quantify the quality of predictions and interpret the performance, accuracy score is calculated for different algorithms and compared.

Table 1 shows the comparison of the accuracies between Logistic regression, KNN and SVM by using binary relevance (BR) method and classifier chain (CC) methods. One more approach used was 'Adaptation Algorithm' (AA) for multi label KNN and Decision trees. Table 2 shows accuracies for Ml-KNN and decision tree for this approach. According to result SVM gives best accuracy of 98.972 percentage by binary relevance method and 98.991 percentage by classifier chain method.

| Model | BR | CC |
|---|---|---|
| Logistic Regression | 98.145 | 98.454 |
| KNN | 98.386 | 98.864 |
| SVM | 98.972 | 98.991 |

Table 1: Comparing accuracies using BR and CC method

| Model | AA |
|---|---|
| Decision Tree | 97.756 |
| Ml-KNN | 98.452 |

Table 2: Comparing accuracies using Adaptation Algorithm method

### B. *Precision, Recall and F-measure*

We will only discuss the precision, recall and f-measure of SVM as it has performed better than other models. The classification report of SVM using BR is as follows:

```
              precision    recall  f1-score   support
           0       0.99      0.99      0.99     45334
           1       0.76      0.76      0.76      2538
   micro avg       0.97      0.97      0.97     47872
   macro avg       0.87      0.87      0.87     47872
weighted avg       0.97      0.97      0.97     47872
```

From classification report of SVM using BR, the f1-score for class 0 is higher than for class 1.

The classification report for SVM using CC method is as follows:

```
              precision    recall  f1-score   support
           0       0.98      0.98      0.98     45494
           1       0.60      0.60      0.60      2378
   micro avg       0.96      0.96      0.96     47872
   macro avg       0.79      0.79      0.79     47872
weighted avg       0.96      0.96      0.96     47872
```

### C. *Time for Training Classifiers*

The table 3 shows time required for training Logistic Regression classifier, KNN classifier and SVM classifier using BR and CC method.

| Model | BR | CC |
|---|---|---|
| Logistic Regression | 3.234 | 2.542 |
| KNN | 235.8 | 210.5 |
| SVM | 195.8 | 205.1 |

Table 3: Time performance in seconds

## IV. DISCUSSION

Using classifier chain method for multi-label classification improves the accuracy of Logistic Regression, KNN and SVM. The accuracy and f1-score is higher for SVM. Hence, SVM is a better algorithm to predict compared to Logistic Regression, KNN and Decision Tree considering accuracy.

Time required for training Logistic Regression algorithm is very less compared to other algorithms. Training of KNN and SVM using both BR and CC method takes huge amount of time due to complex nature of their working and also due to complex dataset.

## V. CONCLUSION

This paper focused on pernicious comment classification where online comments are classified into various predefined categories. There have been ceaseless trials experimenting and computing the presence of toxicity of various kinds on the online platforms including the micro and macro blogging sites by the industries as well as the research communities for an efficient model that detects and predicts the online pernicious comments. This holds importance in the research field due to the tremendously growing online interactive communication among users. This work is dedicated to finding the best possible optimum solutions for classification of online pernicious comments which further classifies the toxic comments into 6 labels provided by the datasets on kaggle platform.

Dataset belongs to Wikipedia's talk page edits. Four different machine learning algorithms- k nearest neighbor, support vector machine, logistic regression and decision tree were applied and corresponding comment is classified accordingly. We can also use this model to classify the comments which are written in different languages.

Higher accuracy is obtained by taking extra efforts in pre-processing of dataset, as it helps us to reduce the noise in dataset. The multi-label classification problem is solved by using Binary Relevance, Classifier Chain and Algorithm Adaptation methods.

The analysis shows that support vector machine (SVM) is preferable than other algorithms.

## REFERENCES

[1] Yutaka Sasaki, "Introduction to Text Classification" [online] Available: https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/tutorial-TC.html.

[2] Fabrizio Sebastiani, "Research in Automated Text Classification: Trends and Perspectives".

[3] Fatima Chiroma, Han Liu, Mihaela Cocea, "Suicide related text classification with prism algorithm "Proceedings of the 2018 International conference on Machine Learning and Cyberbernetics, Chengdu, China 15-18 July 2018.

[4] Purvi Prajapati, Amit Thakkar, Amit Ganatra, "A Survey And Current Research Challenges in Multi-Label Classification Methods", (IJSCE) ISSN: 2231-2307, Vol. 2, March 2012.

[5] Menaka S, Radha N, "Text Classification using Keyword Extraction Technique", (IJARCSSE) ISSN: 2277 128X, Vol. 2, 12, December 2013.

[6] Xiaogang Peng, Ben Choi, "Document Classifications Based on Word Semantic Hierarchies", in proceedings of ACM International conference on Artificial Intelligence and Applications, pp 362-367.

[7] O. W. Kwon, J. H. Lee, "Text Categorization Based on K-nearest Neighbor Approach for Web Site Classification", published in an International Journal of Information Processing and Management, Vol. 39, p 25-44.