MALIGNANT COMMENT CLASSIFICATION

Submitted by: Ganesh Kumbhar

ACKNOWLEDGMENT

- Wikipedia : https://en.wikipedia.org/wiki/Multi-label_classification
- Kaggle challenge page for datasets and ideas : https://www.kaggle.com/c/
- jigsaw-toxic-comment-classification-challenge
- Conversation AI git page : https://conversationai.github.io/
- Research Paper titled “Multi-label Classification: Problem Transformation methods in Tamil Phoneme classification” : https://ac.els-cdn.com/
- S1877050917319440/1-s2.0-S1877050917319440-main.pdf? _tid=eced1a38-f8fa-11e7- b8ef-00000aab0f27&acdnat=1515914406_0f244d3e6313bb049c435bf43504bd52
- Research Paper titled “Benchmarking Multi-label Classification Algorithms” :
- http://ceur-ws.org/Vol-1751/AICS_2016_paper_33.pdf
- Problem Transformation Methods : https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/
- Research Paper on BP-MLL : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.507.910&rep=rep1&type=pdf
- GridsearchCV on Sequential Models : https://dzubo.github.io/machine-learning/2017/05/25/increasing-model-accuracy-by-tuning-parameters.html
- Ml-knn - A Lazy Learning Approach to Multi-Label Learning : https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/pr07.pdf
- Average Precision score : http://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html
- SOCIAL MEDIA DATA MINING PROJECT REPORT, by Nikhil Gohil, Tejas Mehta, Pratik Bhatia, Nikhil Bharadwaj, Hemanta Pattnaik in Syracuse University, NY

INTRODUCTION

• Business Problem Framing

Online platforms when used by normal people can only be comfortably used by them only when they feel that they can express themselves freely and without any reluctance. If they come across any kind of a malignant or toxic type of a reply which can also be a threat or an insult or any kind of harassment which makes them uncomfortable, they might defer to use the social media platform in future. Thus, it becomes extremely essential for any organization or community to have an automated system which can efficiently identify and keep a track of all such comments and thus take any respective action for it, such as reporting or blocking the same to prevent any such kind of issues in the future.

There has been a remarkable increase in the cases of cyberbullying and trolls on various social media platforms. Many celebrities and influences are facing backlashes from people and have to come across hateful and offensive comments. This can take a toll on anyone and affect them mentally leading to depression, mental illness, self-hatred and suicidal thoughts.

Internet comments are bastions of hatred and vitriol. While online anonymity has provided a new outlet for aggression and hate speech, machine learning can be used to fight it. The problem we sought to solve was the tagging of internet comments that are aggressive towards other users. This means that insults to third parties such as celebrities will be tagged as inoffensive, but “u are an idiot” is clearly offensive.

Our goal is to build a prototype of online hate and abuse comment classifier which can used to classify hate and offensive comments so that it can be controlled and restricted from spreading hatred and cyberbullying.

• Conceptual Background of the Domain Problem

Given a number of tweets (in Twitter) or any kind of other comments, sentences or paragraphs being used as a comment by a user, our task is to identify the comment as whether it is a malignant comment or no. After that, when we have a collection of all the malignant comments, our main task is to classify the tweets or comments into one or more of the following categories – toxic, severe-toxic, obscene, threat, insult or identity-hate. This problem thus comes under the category of multi-label classification problem.
There is a difference between the traditional and very famous multi-class classification, and the one which we will be using, which is the multi-label classification. In a multi-class classification, each instance is classified into one of three or more classes, whereas, in a multi-label classification, multiple labels (such as – toxic, severe-toxic, obscene, threat, insult or identity-hate) are to be predicted for the same instance.
Multiple ways are there to approach this classification problem. It can be done using –

- Multi-label methods which belong to the problem transformation category: Label Power Set (LP), Binary Relevance (BR), BR+ (BRplus), and classifier chain.
- Base and adapted algorithms like: J48 (Decision Tree), Naïve Bayes, k-Nearest-Neighbour (KNN), SMO (Support Vector Machines), and, BP-MLL neural networks.

• Review of Literature

Evaluation Metrics

- **Label bases metrics** include one-error, average precision, etc. These can be calculated for each labels, and then can be averaged for all without taking into account any relation between the labels if exists.

  Average Precision (AP): Average precision is a measure that combines recall and precision for ranked retrieval results. For one information need, he average precision is the mean of the precision scores after each relevant document is retrieved, where, and are the precision and recall at the threshold.
- **Example based metrics** include accuracy, hamming loss, etc. These are calculated for each of the examples and then averaged across the test set.

  Accuracy is defined as the proportion of correctly predicted labels to the total no. of labels for each instance.

• Motivation for the Problem Undertaken

This is a huge concern as in this world, there are 7.7 billion people, and, out of these 7.7 billion, more than 3.5 billion people use some or the other form of online social media. Which means that every one-in-three people uses social media platform. This problem thus can be eliminated as it falls under the category of Natural Language Processing. In this, we try to recognize the intention of the speaker by building a model that's capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate. Moreover, it is crucial to handle any such kind of nuisance, to make a more user-friendly experience, only after which people can actually enjoy in participating in discussions with regard to online conversation.

Analytical Problem Framing

• Mathematical/ Analytical Modelling of the Problem

One of the most time-consuming tasks in Data science is collection and labelling of data. Even when we collect data, a major problem we face is the availability of useful data in the collected dataset. What we noticed was that if we gave the track parameter as (" ") that is space or "a" which is available in all comments, out of 100000 tweets collected around 14000 tweets were toxic which means that many of them were vague or non-toxic. We could not use this

collected data for our analysis. So, we collected data using mainly cuss words in the track parameter because of which almost all of the data we collected could be classified as one of the malignant comment classes and we could perform analysis on the predicted results.

- Data Sources and their formats

The data set contains the training set, which has approximately 1,59,000 samples and the test set which contains nearly 1,53,000 samples. All the data samples contain 8 fields which includes 'Id', 'Comments', 'Malignant', 'Highly malignant', 'Rude', 'Threat', 'Abuse' and 'Loathe'.

The label can be either 0 or 1, where 0 denotes a NO while 1 denotes a YES. There are various comments which have multiple labels. The first attribute is a unique ID associated with each comment.

The data set includes:

- **Malignant:** It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.
- **Highly Malignant:** It denotes comments that are highly malignant and hurtful.
- **Rude:** It denotes comments that are very rude and offensive.
- **Threat:** It contains indication of the comments that are giving any threat to someone.

- **Abuse:** It is for comments that are abusive in nature.
- **Loathe:** It describes the comments which are hateful and loathing in nature.
- **ID:** It includes unique Ids associated with each comment text given.
- **Comment text:** This column contains the comments extracted from various social media platforms.

- Data Pre processing Done

The dataset consists of the following fields-

• id: An 8-digit integer value, to get the identity of the person who had written this comment

• comment_text: A multi-line text field which contains the unfiltered comment

• toxic: binary label which contains 0/1 (0 for no and 1 for yes)

• severe_toxic: binary label which contains 0/1

• obscene: binary label which contains 0/1

• threat: binary label which contains 0/1

• insult: binary label which contains 0/1

• identity_hate: binary label which contains 0/1

Out of these fields, the comment_text field will be pre processed and fitted into different classifiers to predict whether it belongs to one or more of the labels/outcome variables (i.e. toxic, severe_toxic, obscene, threat, insult and identity_hate).

Stemming and Lemmatizing:

1. The process of converting inflected/derived words to their word stem or the root form is called stemming. Many similar origin words are converted to the same word e.g. words like "stems", "stemmer", "stemming", "stemmed" as based on "stem".
2. Lemmatizing is the process of grouping together the inflected forms of a word so they can be analyzed as a single item. This is quite similar to stemming in its working but differs since it depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighbouring sentences or even an entire document.
3. The wordnet library in nltk will be used for this purpose. Stemmer and Lemmatizer are also imported from nltk.

Applying Count Vectorizer:

1. To convert a string of words into a matrix of words with column headers represented by words and their values signifying the frequency of occurrence of the word Count Vectorizer is used.
2. Stop words were accepted, convert to lowercase, and regular expression as its parameters. Here, we will be supplying our custom list of stop words created earlier and using lowercase option. Regular expression will have its default value.

- # Data Inputs- Logic- Output Relationships

This project is more about exploration, feature engineering and classification that can be done on this data. Since the data set is huge and includes many categories of comments, we can do good amount of data exploration and derive some interesting features using the comments text column available.

The data set contains the training set, which has approximately 1,59,000 samples and the test set which contains nearly 1,53,000 samples. All the data samples contain 8 fields which includes ‘Id’, ‘Comments’, ‘Malignant’, ‘Highly malignant’, ‘Rude’, ‘Threat’, ‘Abuse’ and ‘Loathe’.

The label can be either 0 or 1, where 0 denotes a NO while 1 denotes a YES. There are various comments which have multiple labels. The first attribute is a unique ID associated with each comment.

The data set includes:

- **Malignant:** It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.
- **Highly Malignant:** It denotes comments that are highly malignant and hurtful.
- **Rude:** It denotes comments that are very rude and offensive.

- **Threat:** It contains indication of the comments that are giving any threat to someone.

- **Abuse:** It is for comments that are abusive in nature.
- **Loathe:** It describes the comments which are hateful and loathing in nature.
- **ID:** It includes unique Ids associated with each comment text given.
- **Comment text:** This column contains the comments extracted from various social media platforms.

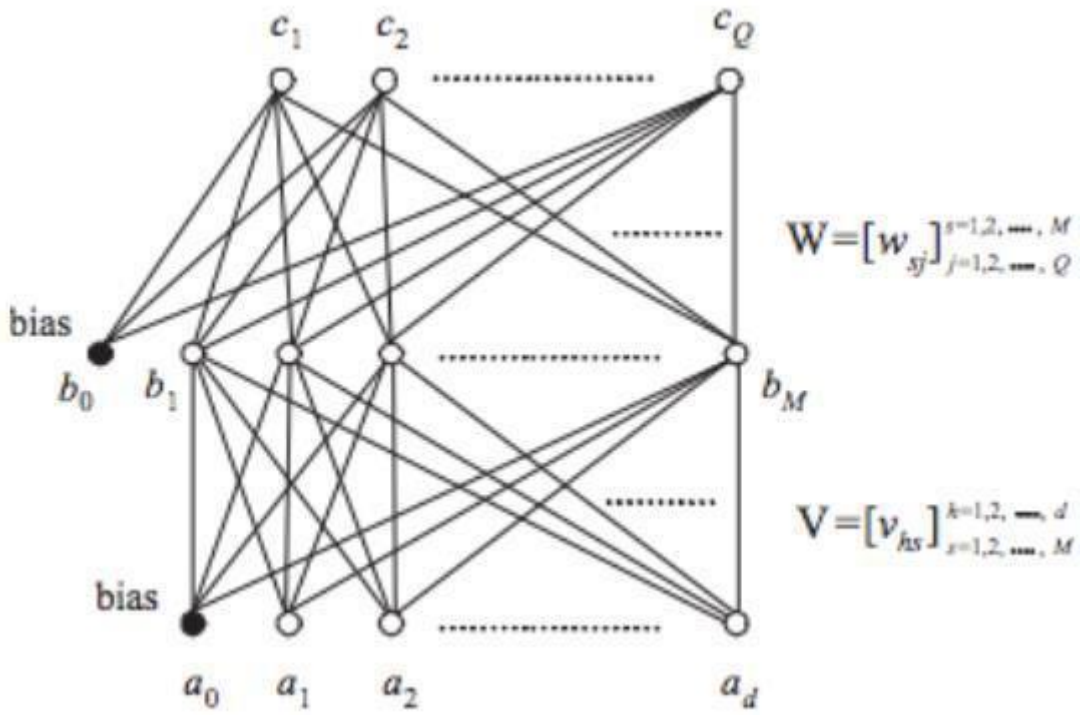
• Hardware and Software Requirements and Tools Used

PROBLEM TRANSFORMATION METHODS:

- **Binary Relevance Method**: This method does not take into account the interdependence of labels. Each label is solved separately like a single label classification problem. This is the simplest approach to be applied.
- **Classifier Chain Method**: In this method, the first classifier is trained on input data and then each of the next classifier is trained on the input space and previous classifier, and so on. Hence this method takes into account some interdependence between labels and input data. Some classifiers may show dependence such as toxic and severe_toxic. Hence it is a fair deal to use this method.
- **Label Powerset Method**: In this method, we consider all unique combinations of labels possible. Any one particular combination hence serves as a label, converting our multi-label problem to a multi class classification problem. Considering our dataset, many comments are such that they have 0 for false labels all together and many are such that obscene and insult are true together. Hence, this algorithm seems to be a good method to be applied. In this, we find that x1 and x4 have the same labels, similarly, x3 and x6 have the same set of labels. So, label power set transforms this problem into a single multi-class problem.

ADAPTION ALGORITHMS:

- **MLKNN**: This is the adapted multi label version of K-nearest neighbors. Similar to this classification algorithm is the BRkNNaClassifier and BRkNNvClassifier which are based on K-Nearest Neighbors Method. This algorithm proves to give superior performance in some datasets such as yearst gene functional analysis, natural scene classification and automatic web page categorization.
  Since this is somewhat similar to the page categorization problem, it is expected to give acceptable results. However, the time complexity involved is large and therefore it will be preferable to train it on smaller dataset.
- **BP-MLL Neural Networks**: Back propogation Multi-label Neural Networks is an architecture that aims at minimizing pair-wise ranking error.
  An architecture of one hidden layer feed forward neural network is as follows:



The input layer is fully connected with the hidden layer and the hidden layer is fully connected with the output layer. Since we have 6 output labels, the output layer will have 6 nodes. This algorithm can be trained in a reasonable amount of time with appropriate number of nodes in the hidden layer.

- **Benchmark Model**
  As we proposed in our proposal, we will be using Support Vector Machine with radial basis kernel (rbf) as the benchmark model (using Binary Relevance Method) Implementation of this model has been done along with other models using the Binary Relevance Method in the implementation section.
  Since we have a large dataset, other classifiers using the bag of words model such as the MultinomialNB and GausseanNB are expected to work than this model.
  But, in practice, it performs quite well. A detailed comparison and analysis between all these classifiers will hence be provided.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

We will be defining **function evaluate_score** for printing all evaluation metrics: Some implementations return a sparse matrix for the predictions and others return a dense matrix. So, a try except block were used to handle it.
Then, we will start implementing various **problem transformation methods**. Binary Relevance, Label Powerset and Classifier Chain methods were included.

Binary Relevance method were implemented from scratch. It does not consider the interdependence of labels and basically creates a separate classifier for each of the labels. Next, Binary Relevance method for other classifiers (SVC, multinomialNB, gausseanNB) is directly

imported from the **Scikit-multilearn** library. Classifiers for **Classifier Chain** and **Label Powerset** are imported and tested.
Then **Adaptation Algorithms** were used. To be precise, MLkNN will be imported from the scikit-multilearn library and BP-MLL will be implemented from scratch.

Back Propagation MultiLabel Neural Networks (BP-MLL) can be implemented using the Sequential Model from Keras. Checkpointer is used to show the intermediate results from different epochs.

- Run and Evaluate selected models

Decision Tree Classifier

```
Training accuracy is 0.9988898736783678
Test accuracy is 0.9406542446524064
[[41638  1312]
 [ 1529  3393]]
              precision    recall  f1-score   support

           0       0.96      0.97      0.97     42950
           1       0.72      0.69      0.70      4922

    accuracy                           0.94     47872
   macro avg       0.84      0.83      0.84     47872
weighted avg       0.94      0.94      0.94     47872
```

Logistic Regression

```
Training accuracy is 0.9595609629450577
Test accuracy is 0.9553183489304813
[[42729   221]
 [ 1918  3004]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.93      0.61      0.74      4922

    accuracy                           0.96     47872
   macro avg       0.94      0.80      0.86     47872
weighted avg       0.95      0.96      0.95     47872
```

RandomForestClassifier

```
Training accuracy is 0.9988719684151156
Test accuracy is 0.9550676804812834
[[42392   558]
 [ 1593  3329]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.86      0.68      0.76      4922

    accuracy                           0.96     47872
   macro avg       0.91      0.83      0.87     47872
weighted avg       0.95      0.96      0.95     47872
```

XGboost

```
Training accuracy is 0.9614052050600274
Test accuracy is 0.9526236631016043
[[42689   261]
 [ 2007  2915]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.97     42950
           1       0.92      0.59      0.72      4922

    accuracy                           0.95     47872
   macro avg       0.94      0.79      0.85     47872
weighted avg       0.95      0.95      0.95     47872
```

AdaBoost Classifier

```
Training accuracy is 0.951118631321677
Test accuracy is 0.9490307486631016
[[42553   397]
 [ 2043  2879]]
              precision    recall  f1-score   support

           0       0.95      0.99      0.97     42950
           1       0.88      0.58      0.70      4922

    accuracy                           0.95     47872
   macro avg       0.92      0.79      0.84     47872
weighted avg       0.95      0.95      0.94     47872
```

CONCLUSION

- Key Findings and Conclusions of the Study

- The first step involved collecting data and deciding what part of it is suitable for training: This step was extremely crucial since including only very small length comments would give poor results if the length was increased whereas including very long length comments would increase the number of words drastically, hence increasing the training time exponentially and causing system (jupyter kernel) to go out of memory and die eventually.
- The second major step was performing cleaning of data including punctuation removal, stop word removal, stemming and lemmatizing: This step was also crucial since the occurrence of similar origin words but having different spellings will intend to give similar classification, but computer cannot recognize this on its own. Hence, this step helped to a large extent in both removing and modifying existing words.
- The third step was choosing models to train on: Since I had a wide variety of models(3 for problem transformation) and classifiers(not bounded) along with number of adaptation models in BP-MLL, selecting which all models to train and test took lots of efforts.

- # Learning Outcomes of the Study in respect of Data Science

Although we have tried quite several parameters in refining my model, there can exist a better model which gives greater accuracy.

Yes. We were unable to find a clear implementation of the Adaboost.MH decision tree model which we had planned to use. The scikit-multi learn library doesn't even mention of such a model. Also, the research papers were a little vague regarding implementation details.

- # Limitations of this work and Scope for Future Work

- The current project predicts the type or toxicity in the comment. We are planning to add the following features in the future:
- Analyse which age group is being toxic towards a particular group or brand.
- Add feature to automatically sensitize words which are classified as toxic.
- Automatically send alerts to the concerned authority if threats are classified as severe.
- Build a feedback loop to further increase the efficiency of the model.
- Handle mistakes and short forms of words to get better accuracy of the result.