Categorizing Inappropriate Texts

Aditi Gupta (2019292) | Mansi Singhal (2019370) | Nimisha Gupta (2019315)

Problem Statement

Conversations that take place over online platforms can become inappropriate and discourteous, degrading the quality of online conversations and having serious consequences like trolling, cyberbullying, and accessing adult explicit content. To avoid such situations, we want to build a **multi-label classification model that will detect inappropriate texts** and help us to further categorize them.

Data Set

We will be using a kaggle dataset containing Wikipedia comments which have been labeled with multiple classes like toxic, obscene, threat etc. by human raters. The dataset in total contains 312,737 samples. We will be splitting this dataset into train, validation and test sets.

id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
	Explanation						
0000997932d77	Why the edits made under my use	0	0	0	0	0	0
000103f0d9cfb6	D'aww! He matches this backgrou	0	0	0	0	0	0
0002bcb3da6cb:	COCKSUCKER BEFORE YOU PI	1	1	1	0	1	0
000113f07ec002	Hey man, I'm really not trying to ec	0	0	0	0	0	0

Preprocessing Techniques

We will first be employing **text cleaning** techniques like replacing contractions with their full forms, removing special characters, lemmatization, removing stopwords, making all the text lowercase and tokenization.
After that we will be using the **TF-IDF feature extraction** technique to convert the data into a numerical form which can be fed into a model. We would also like to explore techniques like Word2Vec and BERT for the same.

Learning Algorithms

1. **Logistic Regression** with various classifiers (Binary Relevance/Classifier Chains/ Label Powerset)
2. Naive Bayes with various classifiers (Binary Relevance/Classifier Chains/ Label Powerset)
3. **Random Forest with MultiOutputClassifier**
4. **SVM with Classifier Chains**
5. **MLkNN**

We will also be exploring neural network techniques for multi-label classification like defining a MLP model using Sigmoid activation and Binary cross-entropy loss function.

Training Approach

We will be using the **Quasi-Newton method** as our optimization algorithm since it has the perfect trade-off between speed and memory requirements as compared to other optimization algorithms.

Model selection and Hyperparameter Tuning

We will be selecting models based on their performance on the **validation set**. For hyperparameter tuning we will be using **Bayesian Optimization** technique and also try manual tuning to learn how drastically a parameter affects the model's performance. To avoid overfitting the model on the validation set we will be using **Stratified K-Fold Cross Validation** technique, with a random seed.

Ensemble Approach

We will be using **Random forests** which is an ensemble of decision tree algorithms and is an **extension of bootstrap aggregation (bagging)** of decision trees.

Evaluation Metric

We will be using **micro-averaging recall as our optimizing metric and micro-averaging precision as our satisficing metric**. We have chosen this evaluation metric since the cost of the model classifying a decent text as obscene (false positive) is far less than the cost of it missing an obscene text (false negative). So we are planning to prioritize minimizing the false-negative rate (optimizing recall), subject to there being no more than a certain threshold of false positives (satisfying precision).

Error Analysis Approach

We will try the following error analysis approaches to identify the most promising directions to improve our algorithm.

1. Manually going through the miscategorized data to **identify misclassifications** and count the major categories of errors to prioritize the categories of errors that we need to rectify.
2. Using training and validation learning curves to determine overfitting or underfitting of the model.
3. Calculating **variance** and **avoidable bais** to determine which will be more rewarding to reduce. For calculating avoidable bais in this problem the bayes optimal error is the human-level performance.

Work distribution

1. Aditi: Error Analysis, MLkNN, Binary Relevance with logistic regression and Naive bayes
2. Mansi: Preprocessing, Random Forest, Classifier Chains with Logistic Regression and Naive bayes.
3. Nimisha: EDA, SVM, Label Powerset with Logistic Regression and Naive Bayes.