Capstone Project

Machine Learning Engineer Nanodegree

Toxic Comments Classification

Mohammed Almohammedsaleh

I. Definition

A. Project Overview

1. *Project domain*

Nowadays, the majority of individuals are active on at least one social networking platform. Not just adults, but even teenagers use them. Nonetheless, some people abuse these platforms to vent their wrath and rage via bullying and the use of filthy language. That is why it is critical to avoid this sort of behavior on the platform by banning anyone who use this language. It will be a tiresome task to force humans to read each remark and determine whether or not it is offensive. That is where Machine Learning comes in help; we can train a model to determine which comment is harmful.

2. *Dataset and input*

The dataset was obtained for free from a Kaggle competition page. It contains a large number of comments taken from Wikipedia talk page.

The dataset is divided into three csv files:

- train.csv: containing 150000 comments for training.
- test.csv: containing another 150000 comments for testing (without labels).
- test_labels.csv: containing labels for test.csv entries.

B. Problem Statement

This project will design a machine learning model that will evaluate a huge set of comments and then utilize the trained model to predict the level of toxicity of a given comment. The level of toxicity will fall into one or more of the following types:

- toxic
- severe_toxic
- obscene
- threat
- insult
- identity_hate

C. Evaluation Metrics

In my proposal I chose accuracy score to be my evaluation metrics to the model. However, after further research I found that accuracy score does not describe the whole story. Accuracy score will give us the relation between the correct predictions to the whole dataset. It describe how much true positives and true negatives the model got right. However, since our dataset contains imbalance data towards unlabeled data, the accuracy will be higher if the model was to randomly predict zeros.

However since our dataset contains a large number of negative (0) values, compared to the positive ones. It is better to evaluate the model based on how well it can predict the positive (1) values, which can be accomplished with precision and recall metrics.

So, Precision and Recall will be the main evaluation metrics that will be used in this project.

- Precision: describes the proportion of accurately predicted positive observations to all observed positive observations in the actual class.
- Recall: describe the ratio of correctly predicted positive observations to the all observations in actual class

II. Analysis

A. Data Exploration

Data Input Fields:

- **train.csv:**
- id: comment id, will not be used
- comment_text: contains the raw text for the comment
- toxic: binary field indicating whether the comment is toxic
- sever_toxic: binary field indicating whether the comment is extremely toxic
- obscene: binary field indicating whether the comment is obscene
- threat: binary field indicating whether the comment is a threat
- insult: binary field indicating whether the comment is an insult
- identity_hate: binary field indicating whether the comment consist of identity has

This dataset has a total of 159571 records. By contrast, only 16225 comments have been categorized into at least one of the six categories. This demonstrates that the dataset is not evenly distributed.

The testing dataset is pre divided into two files:

- test.csv: consist of the comments and their IDs
- test_labels.csv: consist of the same comment IDs with their labels.

B. Exploratory Visualization

The first figure shows that class 'toxic' takes the majority of the labeled data. However, the second figure outlines that the classified data consist about ten percent of the overall dataset.

Figure 1: Number of occurrences of each category

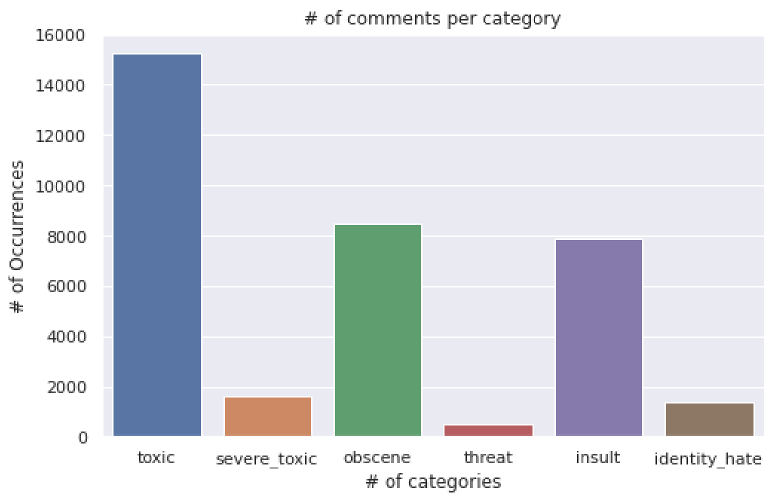
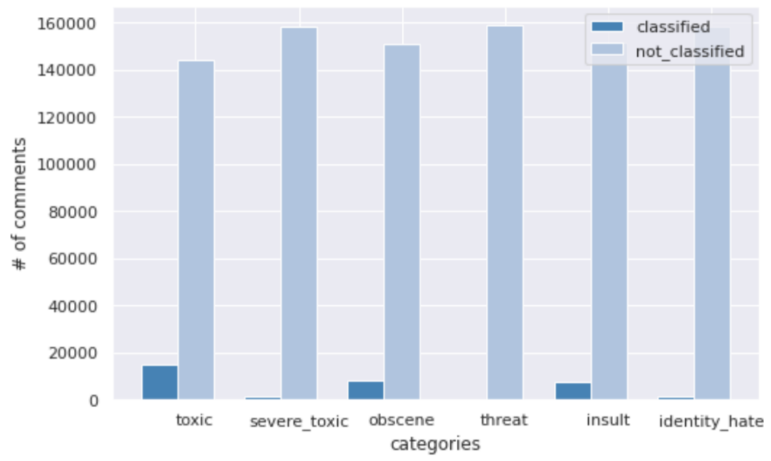


Figure 2 destribution of classification for each category



C. Algorithms and Techniques

All model will be used with OneVsRestClassifier. OneVsRestClassifier will help us with the multilabel classification, as we can define it with any classification algorithm and it will handle the training of each label separately.

We used each of the following algorithms:

- LogisticRegression
- MultinomilNB
- LinearSVC

Along with OneVsRestClassifier, we will be using Pipline to iterate over each row in the dataset. This process will help in increasing the performance of the model and reducing the time required for training.

D. Benchmark Model

Since we have a binary classification problem, Logistic Regression algorithm might be a good choice to use as a benchmark. As discussed earlier, OneVsRestClassifier will be used throughout the whole project.

So, the benchmark model for this project will be the Logistic Regression Algorithm with OneVsRestClassifier.

III. Methodology

A. Data Preprocessing

In this step, we tried to clean the data as much as possible. Since the data consist of comments taken from a website, it is possible that some comments might contains HTML tags, punctation marks, or numbers. All of these does not provide any information about the comment toxicity. So, the first thing we did in this step is to remove all non-alphabetical characters from the comments.

Then, we used text stemming to remove similar words and combine them into basic English words. For instance, Playing, Plays, Played, and Player are treated as “Play”.

The last thing for data preprocessing is to remove all stop words in each comment. Stop words are the most frequent words in the English language that does not provide any information about the text meaning. Those word can be such as, pronouns, prepositions, or conjunctions … etc.

B. Refinement

The best score gained before refinement is:

- Precision: 0.57
- Recall: 0.75

The refinment process consist into two steps, the first one is to remove some of the non-categorized comments. The second step is to tune TfidfVectorizor’s parameters.

The gained scores after refinement are:

- Precision: 0.81
- Recall: 0.77

C. Implementation

The implementation phase is divided into three stages:

- ➢ Initial Stage

In this stage the whole data set will be used for training. However, entries that contains label values of -1 from the testing set is dropped which according to the dataset provider website they does not held any meaningful value. So, eventually we will have about 150,000 entries for training and about 60,000 for testing.

Then all comments of both training and testing sets will be processed with the discussed attributes in part A.

The next step is to extract features form the comments using TfidfVectorizer. However extracting those features form all comments at once to be fed into the model requires a huge amount of memory, nearly 100 GB of memory. So, the solution is to make a pipline that will extract the features from one comment at a time and giving it to the training model. This method will be held on throughout all stages.

The next step is to train the three proposed model using a pipline and TfidfVectorizer used with the pipline to avoid storing a

and evaluate their performance based on their precision and recall scores. Starting from the benchmark model we get:

- Logistic Regression (benchmark)
  - Average Precision: 0.55
  - Average Recall: 0.47
- MultinomialNB
  - Average Precision: 0.90
  - Average Recall: 0.04
- LinearSVC
  - Average Precision: 0.57
  - Average Recall: 0.75

From the results above MultinomialNB has the highest precisison score. However, recall score of 0.04 tells that there is something went wrong.

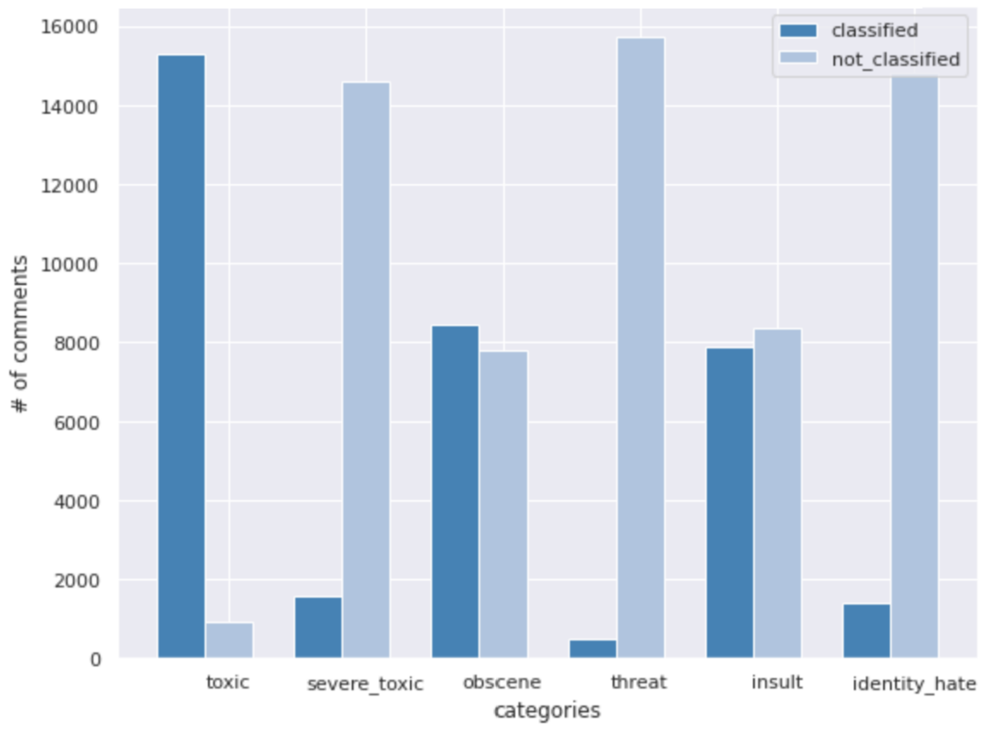
After further investigation, confusion matrix of MultinomailNB indicates that the model did not predict any true positive value for the category with small ration of labeled values. Which means that MultinomialNB is not suitable for imbalanced classification.

On other hand, both Logistic Regression and LinearSVC performed fine.

- ➢ Refinement Stage

In this stage, not all the training data will be used. Half of the uncategorized entries from the training set will be dropped, and 30% of the testing data is dropped also. This will reduce the unbalance the disproportion between positive and negative values. Only half will be dropped because if all the uncategorized entries is dropped the disproportion of the data will become biased towards positive values as shown in the figure below.

Figure 3: Training data with all non-categorized entries dropped



The resulted score after refinement:

- Logistic Regression (benchmark)
    - Average Precision: 0.84
    - Average Recall: 0.70
- MultinomialNB
    - Average Precision: 0.88
    - Average Recall: 0.23
- LinearSVC
    - Average Precision: 0.81
    - Average Recall: 0.77

All models scores has improved after modifying the dataset.

- Final stage

In this stage modifications will be done on the vectorizer for better feature extractions. Also, the modified dataset is used in this stage.

The modifications is done on the vectorizer's parameters as follows:

- max_df: indicates the maximum frequency allowed for each term.
  - Test values are: 0.25, 0.5
- n_gram: determines how many characters each term must contain in order to be considered by the vectorizer
  - Test values are: (1, 1), (1,2) , (1,3), (1,4)

The best model found is:

- Algorithm: LinearSVC(),
- Vectorizer's parameter: max_df = 0.25 , n_gram(1, 2)
- Score:
  - Precision: 0.79
  - Recall: 0.80

IV. Results

A. Justification

Benchmark scores:

- Precision = 0.55
- Recall = 0. 47

Final scores:

- Precision = 0.79
- Recall = 0.80

Based on the nature of the dataset, for which some categories like 'threat' contains a huge imbalance between positive and negative values. And deleting more entries will result in increasing the imbalance for the other categories which contain higher positive values. A precision of 0.8 is considered reasonable.

V. Conclusion

To sum up, this project tried to construct a machine learning model to predict the level of toxcicty of a comment. A dataset provided by a Kaggle competition has been used along with three machine learning algorithms. A precision score of 0.8 has been obtained, which means that for every ten comments eight of them will be classified correctly. Because the dataset was imbalanced a score of eighty percent is considered reasonable given the fact that the dataset was classified by humans.