Capstone Project Proposal

Machine Learning Engineer Nanodegree

Toxic Comments Classification

Mohammed Almohammedsaleh

1. Domain Background

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. Problem Statement

This project will design a machine learning model that will evaluate over 15000 different comments and then utilize the trained model to create a web application that will accept user comments and display the level of toxicity contained inside.

Toxicity types is as follows:

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

3. Datasets and Inputs

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

Data Input Fields:

- id: comment id, will not be used
- comment_text: contains the raw text for the comment

The following columns are the targets that we want the model to predict:

- 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

4. Solution Statement

This is a multi-label classification problem, where we want to classify each comment into all of the six categories. Hence, a comment can be toxic, severe toxic, obscene, threat , insult, and identity hate at the same time. Since these categories are not mutually exclusives.

My approach to solve this problem is to train a machine learning model with a multi-label classification algorithm, after processing and splitting the data into training and testing sets. Then use this model to predict a comment provided by the user and return an output as a JSON object that contains all the labels that describe the comment.

5. Benchmark Model

BinaryRelevance algorithm provided by scikit-multilearn library will be my benchmark model for this project.

6. Evaluation Metrics

To evaluate our model, we will use the following metrics:

- Prediction accuracy score: which describe how will the model is predicting the testing data.
- Hamming Loss: which describe the percentage incorrect predictions.

7. Project Design

7.1 Data Exploration

In this step we will load the data and explore it by plotting several graphs to grasp an idea how the data is distributed, and how many comments falls into each category.

We will also use correlation-matrix to see how the categories relate to each other.

7.2 Data Preprocessing

For data preprocessing we will

- Remove all symbols and numbers from the comments, and any html tag if they exist.
- Remove stopwords
- Convert all letters into lower case

7.3 Feature Extraction

In this step we will use sklearn’s Tfidf -Vectorizer to convert comments into a matrix of TF-IDF features.

7.4 Model Training

Data will be splited into training and testing sets. Then staring from the proposed model earlier we will feed the training data into a BinaryRelevance model. We will test the model with the testing set and calculate the prediction accuracy.

We will try two other models from scikit-mulitlearn library, which are BRkNNaClassifier and MLkNN. Then compare the result for the three model then pick whichever produce the be accuracy.

References

1. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview

2. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

3. http://scikit.ml/modelselection.html#Estimating-hyper-parameter-k-for-MLkNN