DESCRIPTION

Using NLP and machine learning, make a model to identify toxic comments from the Talk edit pages on Wikipedia. Help identify the words that make a comment toxic.

**Problem Statement:**

Wikipedia is the world’s largest and most popular reference work on the internet with about 500 million unique visitors per month. It also has millions of contributors who can make edits to pages. The Talk edit pages, the key community interaction forum where the contributing community interacts or discusses or debates about the changes pertaining to a particular topic.

Wikipedia continuously strives to help online discussion become more productive and respectful. You are a data scientist at Wikipedia who will help Wikipedia to build a predictive model that identifies toxic comments in the discussion and marks them for cleanup by using NLP and machine learning. Post that, help identify the top terms from the toxic comments.

**Domain**: Internet

**Analysis to be done**: Build a text classification model using NLP and machine learning that detects toxic comments.

**Content:**

id: identifier number of the comment

comment\_text: the text in the comment

toxic: 0 (non-toxic) /1 (toxic)

**Steps to perform:**

Cleanup the text data, using TF-IDF convert to vector space representation, use Support Vector Machines to detect toxic comments. Finally, get the list of top 15 toxic terms from the comments identified by the model.

**Tasks:**

1. Load the data using read\_csv function from pandas package
2. Get the comments into a list, for easy text cleanup and manipulation
3. Cleanup:
   1. Using regular expressions, remove IP addresses
   2. Using regular expressions, remove URLs
   3. Normalize the casing
   4. Tokenize using word\_tokenize from NLTK
   5. Remove stop words
   6. Remove punctuation
   7. Define a function to perform all these steps, you’ll use this later on the actual test set
4. Using a counter, find the top terms in the data.
   1. Can any of these be considered contextual stop words?
   2. Words like “Wikipedia”, “page”, “edit” are examples of contextual stop words
   3. If yes, drop these from the data
5. Separate into train and test sets
   1. Use train-test method to divide your data into 2 sets: train and test
   2. Use a 70-30 split
6. Use TF-IDF values for the terms as feature to get into a vector space model
   1. Import TF-IDF vectorizer from sklearn
   2. Instantiate with a maximum of 4000 terms in your vocabulary
   3. Fit and apply on the train set
   4. Apply on the test set
7. Model building: Support Vector Machine
   1. Instantiate SVC from sklearn with a linear kernel
   2. Fit on the train data
   3. Make predictions for the train and the test set
8. Model evaluation: Accuracy, recall, and f1\_score
   1. Report the accuracy on the train set
   2. Report the recall on the train set:decent, high, low?
   3. Get the f1\_score on the train set
9. Looks like you need to adjust  the class imbalance, as the model seems to focus on the 0s
   1. Adjust the appropriate parameter in the SVC module
10. Train again with the adjustment and evaluate
    1. Train the model on the train set
    2. Evaluate the predictions on the validation set: accuracy, recall, f1\_score
11. Hyperparameter tuning
    1. Import GridSearch and StratifiedKFold (because of class imbalance)
    2. Provide the parameter grid to choose for ‘C’
    3. Use a balanced class weight while instantiating the Support Vector Classifier
12. Find the parameters with the best recall in cross validation
    1. Choose ‘recall’ as the metric for scoring
    2. Choose stratified 5 fold cross validation scheme
    3. Fit on the train set
13. What are the best parameters?
14. Predict and evaluate using the best estimator
    1. Use best estimator from the grid search to make predictions on the test set
    2. What is the recall on the test set for the toxic comments?
    3. What is the f1\_score?
15. What are the most prominent terms in the toxic comments?
    1. Separate the comments from the test set that the model identified as toxic
    2. Make one large list of the terms
    3. Get the top 15 terms