Problem Statement

The objective of this task is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets.

Formally, given a training sample of tweets and labels, where label '1' denotes the tweet is racist/sexist and label '0' denotes the tweet is not racist/sexist, your objective is to predict the labels on the test dataset.

Motivation

Hate  speech  is  an  unfortunately  common  occurrence  on  the  Internet.  Often social media sites like Facebook and Twitter face the problem of identifying and censoring  problematic  posts  while weighing the right to freedom of speech. The  importance  of  detecting  and  moderating hate  speech  is  evident  from  the  strong  connection between hate speech and actual hate crimes. Early identification of users promoting  hate  speech  could  enable  outreach  programs that attempt to prevent an escalation from speech to action. Sites such as Twitter and Facebook have been seeking  to  actively  combat  hate  speech. In spite of these reasons, NLP research on hate speech has been very limited, primarily due to the lack of a general definition of hate speech, an analysis of its demographic influences, and an investigation of the most effective features.

Data

Our overall collection of tweets was split in the ratio of 65:35 into training and testing data. Out of the testing data, 30% is public and the rest is private.

Data Files

1. **train.csv** - For training the models, we provide a labelled dataset of 31,962 tweets. The dataset is provided in the form of a csv file with each line storing a tweet id, its label and the tweet.

There is 1 test file (public)

1. **test\_tweets.csv** - The test data file contains only tweet ids and the tweet text with each tweet in a new line.

Submission Details

The following 3 files are to be uploaded.

1. **test\_predictions.csv** - This should contain the 0/1 label for the tweets in test\_tweets.csv, in the same order corresponding to the tweets in test\_tweets.csv. Each 0/1 label should be in a new line.

1. **A .zip file of source code** - The code should produce the output file submitted and must be properly commented.

Evaluation Metric:

The metric used for evaluating the performance of classification model would be F1-Score.

The metric can be understood as -

**True Positives (TP)** - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.

**True Negatives (TN)** - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.

**False Positives (FP)** – When actual class is no and predicted class is yes.

**False Negatives (FN)** – When actual class is yes but predicted class in no.

**Precision** = TP/TP+FP

**Recall**= TP/TP+FN

**F1 Score** = 2\*(Recall \* Precision) / (Recall + Precision)

F1 is usually more useful than accuracy, especially if for an uneven class distribution.