# Twitter hate speech detection using ML:

In today’s world, where online threats and cyber bullying has increased exponentially, the detection and filtering of any speech that may be harmful or have a negative effect in the community has become a vital tool. However, it is difficult to create a detector that is so strong that can manage to eliminate all forms of ‘hate’ speech as hate speech is actually quite ambiguous, and to extract meanings from words or letters may be harder because one word may be used in many different forms of expression, and a string of words may express different meanings that may have hateful words but the meaning simple may not be ‘hateful’ .

In this project, we create a hate speech detection tool for tweets using ML and NLP. The objective behind this project is to understand the difficulties that arise while working with unstructured data and also to predict hateful tweets from a set of collected tweets.

## Data:

We have used two datasets in this project. The train dataset has three columns, ID, tweet and label for denoting whether a tweet is a hate tweet or not(which was predetermined by a group of individuals). It has about 32000 tweets. The test dataset has only the tweets and a column of id in it. The total number of tweets in the test data is around 17200. We will be working with the train data only . We check the duplicates in the data, and it is found to be none. There are no missing values as well. Now to gather some idea about what we are dealing with, we produce two wordclouds to get an idea about the frequency of words and the type of words. Based on the wordcloud of hate tweets, we can observe some underlying themes of the form of hate: racist, derogatory and political.

## Feature engineering:

We extracted a few characteristics from the set of tweets: like the tweet length, number of hashtags, exclamation marks, question marks, tags, punctuations, number of words. However, when we created histograms of the above extracted features and plotted it on top of each other for both hate and non-hate tweets, we found out that these features had no distinguishing effect on the classification of the tweets. Hence, we did not use it in our model building.

## Data preprocessing:

We split the train data into two parts: train and test. The division was done in the ratio of 80:20. We also tokenize the tweets, and make all the letters lowercase, remove the user names as those will not add value to the data and punctuations, stop words etc.

Now we convert every tweet to a numerical vector using tf-idf vectorisation.

One problem we have with the data is that the dataset is highly imbalanced. From the pie chart, it is observed that only 7% of the data is labelled as hate tweet. If we build a model on this data, then the model formed will not be able to capture the hate tweets’ characteristics, and instead will perform unrealistically. Hence, we used a technique called SMOTE- synthetic minority oversampling technique to balance the class. That is, we synthetically generated some tweets from the hate tweets.

## ML Model:

We tried fitting four models: logistic regression, Naïve Bayes classifier, random forest classifier, Gradient boosting classifier. Out of all the models, the random forest and the extreme gradient boosting gave the highest F1 scores for both the training and the testing data.

However, the F1 score for the training data for random forest was too perfect ( equal 100%) and the F1 score for the training data for gradient boosting was around 94%. This suggests some overfitting and underfitting respectively, hence we tried to resolve this with hyperparameter tuning.

## Hyperparameter Tuning:

We tweaked the models by using various combinations of hyperparameters. After which the problem of overfitting and underfitting was resolved.

Final model that came out the best is : random forest classifier with a validation F1 score of 71%.

Then we predicted the labels for the tweets in the test data that we had collected before.

From the test data, we can see that more or less the prediction was satisfactory. However, the problem that we had discussed before, that is the internal meaning of the sentence when it is contrary to the meaning of the words in cases such as those, the detector failed to label them as ‘non hate’ tweets. This suggests that we may need to dive deeper into text analytics to solve this issue.