Twitter Sentiment Analysis

Detection of hate speech in tweets

Sumit Mishra

1. Problem Statement

The task is to detect hate speech in tweets. 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 data set of tweets and labels, where label '1' denotes the tweet is racist/sexist and label '0' denotes the tweet is not racist/sexist, the objective is to predict the labels.

2. Data Set

The Data Set is taken from Kaggle and is also available on Analytics Vidhya. It 3 columns named id, tweets and label. The tweet columns contain the tweets and the label column contain the info if the tweet is hate speech or not. label '1' denotes the tweet is racist/sexist and label '0' denotes the tweet is not racist/sexist. The Tweets are Strings. The tweets contain URL (Web Addresses), twitter handle, hashtags, Special characters etc.

3. Preprocessing

- Converting the tweets to lower case.
- Replacing two or more spaces with single space.
- Replacing two or more dots (.) with single space.
- Removing all the Special characters and the Non-ASCII characters.
- Removing all the leading and trailing whitespaces.
- **URL:** Users often shares URL and web addresses in the tweets and any particular URL is not important for text classification, hence any URL is replaced with “URL”.
- **User Mention:** Twitter user very often mentions other users in the tweets by @Handle. All the User Mentions are Replaced with “user”.
- **Hashtag:** Hashtags are unspaced words prefixed by a hash (#) symbol. All the hashtags are replaced by “hash”.
- Removal of all the stopwords from the tweets as they don’t help in training the model and only increase the computational time.

- Removing the word Stems using the Snowball Stemmer from the nltk.stem . I've used the Snow ball stemmer of the English language.
- Used the Bag of Word approach and used Frequency Distribution to find the frequency of each words in the tweets and used the first 5000 words as the features of the tweets and Found the frequency of all the 5000 words in each tweet and used the output as the feature set to train the models.
- Used the train_test_split to split the data set in to train and test data.

4. NLTK Models

- **Decision Tree Classifier:** Decision Tree Classifier gave an accuracy of 92.06 and have a decent Precision and recall scores according to the classification report.

- **Stochastic Gradient Descent:** SGD gave an accuracy of 94.49 (with max_iter = 1000) but in the classification report it has a very low F1 score and recall and hence can't be used for the predictions.

- **Random Forest:** Random Forest gave an accuracy of 94.71 and in the classification report it has a high precision and reasonable recall.

- **Logistic Regression:** Logistic Regression gave an accuracy of 94.48 and in the classification report it had a very low recall and F1 score. Comparing with the other models it's the least favorable.

5. RESULT

The Best Model is the Decision Tree Classifier and I have Used it for the final predictions.