**Detect hate speech in tweets**

**Safaa Alraddadi**

**Abstract :**

Social Media are sensors in the real world that can be used to measure the pulse of societies. However, the massive and unfiltered feed of messages posted in social media is a phenomenon that nowadays raises social alarms, especially when these messages contain hate speech targeted to a specific individual or group. This project aims to automate content moderation to identify hate speech using machine learning binary classification algorithms. Baseline models included Random Forest, Naive Bayes, Logistic Regression and Support Vector Machine (SVM). The final model was a Logistic Regression model that used Count Vectorization for feature engineering which outperform other models. It produced an F1 of 0.3958 and Recall (TPR) of 0.624.

**Design:**

Dataset using Twitter data, is was used to research hate-speech detection. The text is classified as: hate-speech, offensive language, and neither. The dataset consists of 24783 rows and 7 columns. I am expecting on working on tweet features .

the labels on this dataset were voted on by crowdsource and determined by majority-rules. The “class” column labels each tweet as 0 for hate speech, 1 for offensive language or 2 for neither. I will be treating the data as a binary classification problem.

Therefore, the final model will be predicting whether a tweet is hate speech or not. To prepare the data for this, I will be manually replacing existing 1 and 2 values as 0, and replacing 0 as 1 to indicate hate speech.

**Algorithm :**

I have used Baseline models included Random Forest, Naive Bayes, Logistic Regression and Support Vector Machine (SVM). The dataset suffers from imbalanced classes which was handled by using SMOTE. CountVectorizer and TF-IDF have been used as feature engenering.The results showing in the tables below

Table

Description automatically generatedUnfortunately, Baseline Linear SVM with Count Vectorization did not achieve a higher F1 than the TF-IDF version of the SVM model. Using Count Vectorization on the Logistic Regression baseline actually produced the highest F1 and Recall out of all the other models.

Dealing with class imbalance we use Over sampling with smooth This method over-samples the minority class, "Hate Speech". Seems that the uniform F1 score went down with SMOTE, from 0.3958 to 0.3121. It also had a lower Recall score. Moreover , Under-Sampling with Tomek Links Although using Tomek Links performed better than using SMOTE, the resulting F1 still isn't as good as the initial Logistic Regression model's F1 score of 0.3958.

**Tools**

* Numpy and Pandas for data manipulation
* Scikit-learn for modeling, confusion matrix and feature extraction
* Matplotlib and Seaborn for plotting
* NLP library such as nltk