Detect hate speech in tweets

Safaa Alraddadi

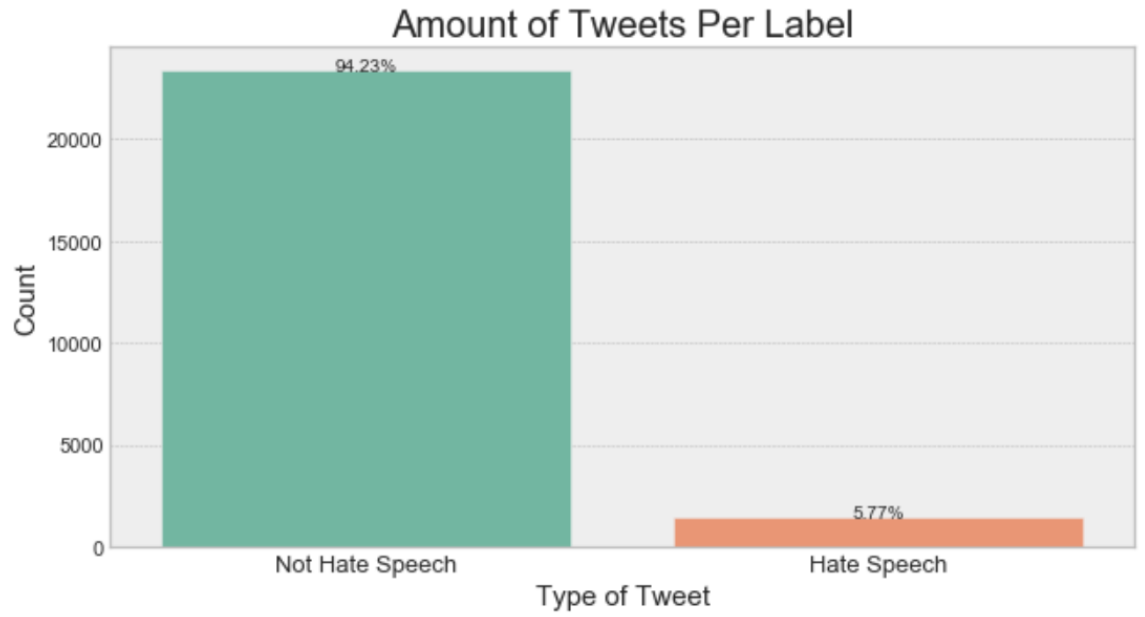
problem and objective

- Messages posted in social media may contain heat message
  - Target individual and group

- aims to automate content moderation to identify hate speech using machine learning binary

Data and method

- The dataset is provided as a .csv file with 24,802 text posts from Twitter where 6% of the tweets were labeled as hate speech.
- **Class Imbalance**



Prepressing Text data

- **cleaning step:**
  - Reassigning labels
  - Lowercasing tweet text
  - Removing hashtags, mentions, quotes and punctuation from tweet text
  - checking for missing value
  - Tokenization & removing stop word
  - Lemmatization

Feature Engineering

- **TF-IDF Vectorization**
- **Count Vectorization**

Modeling process

- **Baseline Random Forest**

```
Testing Set Evaluation Metrics:
Precision: 0.4128
Recall: 0.1613
F1 Score: 0.232
Weighted F1 Score: 0.9272
```

These scores are not ideal because the F1 is being brought down on how the model predicts the "Hate Speech" label.

Modeling process

Baseline Logistic Regression

```
Testing Set Evaluation Metrics:
Precision: 0.2939
Recall: 0.5699
F1 Score: 0.3878
Weighted F1 Score: 0.9134
```

Compared the first Random Forest baseline, the Logistic Regression baseline performed much better. The F1 score increased from 0.232 to 0.3878.

Modeling process

Baseline Naive Bayes

```
Testing Set Evaluation Metrics:
Precision: 0.4118
Recall: 0.1254
F1 Score: 0.1923
Weighted F1 Score: 0.9255
```

The F1 score dropped down to .1923. So, this model performed worse than both the Random Forest and Logistic Regression models.

Modeling process

- **Baseline Support Vector Machine**

```
Testing Set Evaluation Metrics:
Precision: 0.3609
Recall: 0.4373
F1 Score: 0.3955
Weighted F1 Score: 0.9281
```

- This model produced the highest F1 so far, with a score of .3955.

Modeling process

- **Evaluation Metrics for All Baseline Models**

	precision	recall	f1_score
Baseline Random Forest - TFIDF	0.412844	0.161290	0.231959
Baseline Log Reg - TFIDF	0.293900	0.569892	0.387805
Baseline Naive Bayes - TFIDF	0.411765	0.125448	0.192308
Baseline SVM - TFIDF	0.360947	0.437276	0.395462

Modeling process-Count vectorization

- **Baseline Linear SVM with Count Vectorization**

```
Testing Set Evaluation Metrics:
Precision: 0.2712
Recall: 0.5365
F1 Score: 0.3603
Weighted F1 Score: 0.9104
```

- Unfortunately, this model did not achieve a higher F1 than the TF-IDF version of the SVM model.

Modeling process-Count vectorization

- **Baseline Logistic Regression with Count Vectorization**

```
Testing Set Evaluation Metrics:
Precision: 0.2898
Recall: 0.6241
F1 Score: 0.3958
Weighted F1 Score: 0.9121
```

- 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

- **Over sampling with smooth**
  - This method over-samples the minority class, "Hate Speech".

```
Testing Set Evaluation Metrics:
Precision: 0.2326
Recall: 0.4745
F1 Score: 0.3121
Weighted F1 Score: 0.9024
```

Seems that the uniform F1 score went down with SMOTE, from 0.3958 to 0.3121. It also had a lower Recall score.

Dealing with class imbalance

- **Under-Sampling with Tomek Links**
  - This method under-samples the majority class, "Not Hate Speech."
-

```
Testing Set Evaluation Metrics:
Precision: 0.5702
Recall: 0.2372
F1 Score: 0.3351
Weighted F1 Score: 0.9377
```

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.

Final model Analysis-Logistic regression with count Vectorize

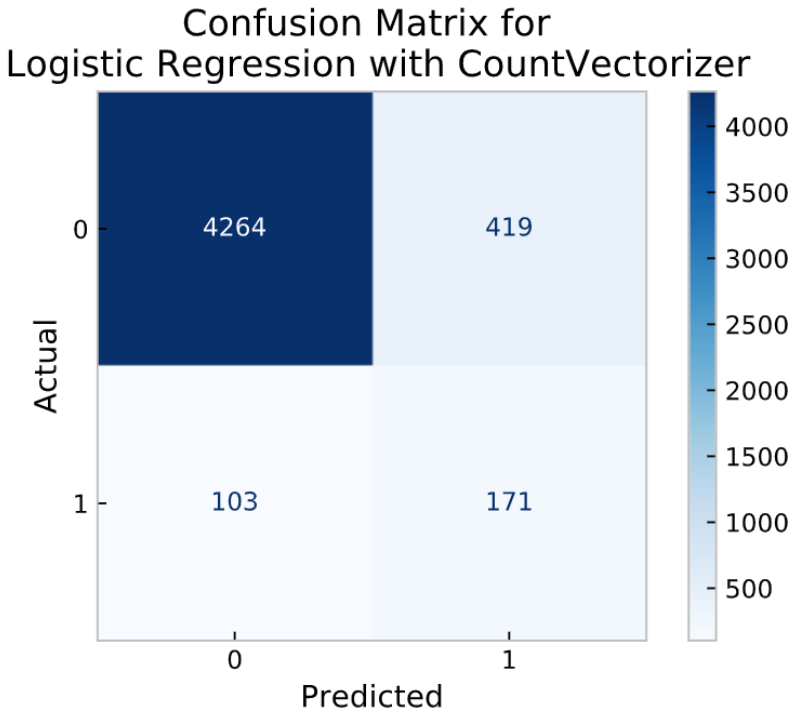
- **Evaluation Metrics for Testing Set**

```
Testing Set Evaluation Metrics:
Precision: 0.2898
Recall: 0.6241
F1 Score: 0.3958
Weighted F1 Score: 0.9121
```

Ultimately, the uniform F1 score of .3958 is so low because it is brought down by the poor predicting ability for the "Hate Speech" label.

Final model Analysis

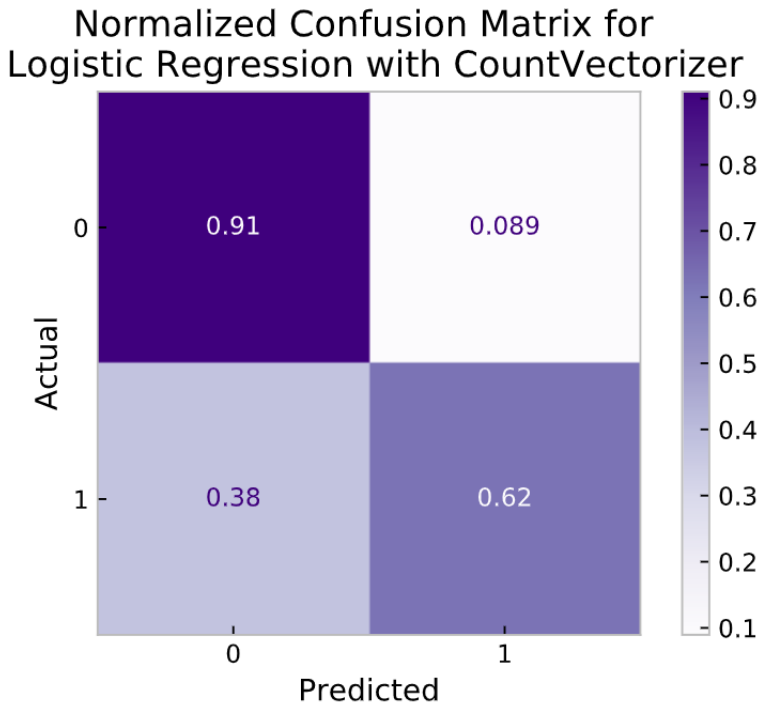
- **Confusion matrix**



From this confusion matrix, we can see that the True Negative rate is high, but the True Positive rate is much lower.

Final model Analysis

- **Confusion matrix**



The final model has a True Negative Rate of 91% and a True Positive Rate of 62%.

Thank You