**Offensive Tweet Detection Model**

1. **Introduction**

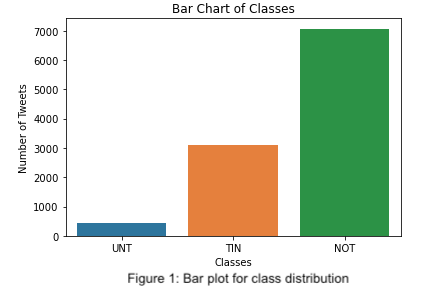
As the world of internet social media grows, posts on a wide variety of issues are posted every day. These posts are a useful way of expressing one's opinion to the public. With this comes the use of offensive words and texts which are either simply expressing their dissatisfaction or frustration at something or some or targeted to hurt individuals or groups. It is important to distinguish tweets in terms of their offensive content to identify and take mitigating actions. Text classification using machine learning can be a useful tool. As part of this project I selected a dataset that includes a total of 10,592 labeled tweets and tasked to create a machine learning model capable of identifying offensive tweets that are targeted at individuals or groups. In the process I created several features and experimented using classification methods to derive a better performing model.

The main objective of this project is to improve the predicting performance of the machine learning model, which is measured by the micro and macro F1 values.

1. **Methodology**

10,592 labeled test data was loaded and data was represented into tweets and label features. The training dataset has three classes, namely - NOT: Not offensive, TIN: Targeted Insult, and UNT: Untargeted. A bar chart of classes in the training data is shown below. Another 2,648 tweets were provided and kept aside to be predicted with the best model.

The data is not balanced, the majority of the tweets are not offensive (64%) and only about 4% of the tweets are considered to be an untargeted insult. The train-validation split was used to test the various models and the creation of features. A base model was then run on the original data provided and other models with a combination of created features and classification methods were run to improve this base model.



**2.1. Data Preparation:**

For the purpose of lexicon-based features, I cleaned the tweets for white space, special characters, apostrophes, commas, numeric digits, periods, URLs, @USERS, emojis etc. using python functions and regular expression packages. Comparison of clean and raw tweets is shown below.

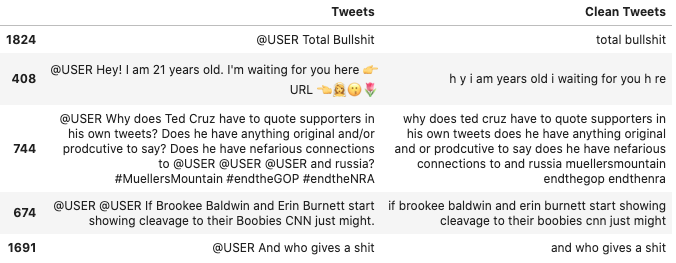


Figure 2: Sample cleaned tweets

**2.2. Creating new features:**

Several features were created to improve the model. The following new features were created:count of punctuation, count of capitalized words, titled words to identify possible names, ellipses, twitter objects which can be indicative of whether a tweet is targeted (@ mentions, # mentions and URLs).

Three more lexicon-based features were created from cleaned-data and the features are count of offensive words, pronouns, words/terms with most offensive words are targeted at.

**2.3. Models Explored:**

The following classifiers from the Sklearn package were used to find the best model that would be able to classify the tweets correctly. LinearSVC, SGDClassifier, RandomForest and Decision Tree etc. Micro and Macro F1 scores as well as the were used to compare the models.

1. **Results and Discussion**

**3.1. Feature Engineering**

New features were created to improve the model's prediction score. Since the classification required identifying offense/profane words, CMU’s [Offensive/Profane Word List](https://www.cs.cmu.edu/~biglou/resources/bad-words.txt) was used to create a count of offensive words in the data. To help identify whether the words are targeted at an individual or group, a list of pronouns and potential list of targets in the context of American politics was used. For this I used the list and modified it based on the frequency of common targets such as Trump and Hillary(reference). Moreover, a count of twitter objects (URLs, @USER and # mentions) indicating reference or mention were created from the raw data to aid classification.

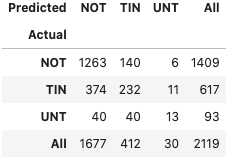
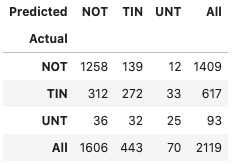
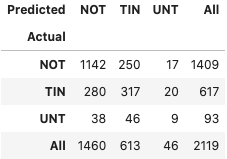
Other features that signify the tweets intentions and emphasis such as the use of capitalized words, use of question marks, titled words to identify potential names and exclamation marks were counted.

**3.2. Modelling and Evaluation**

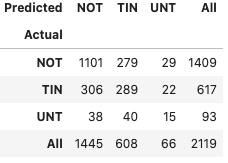
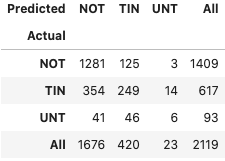
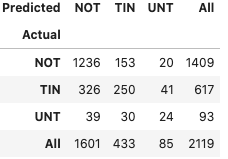
I started off by splitting the row data to train and validation split and vectorized it using the CountVectorizer method and ran the LinearSVC classifier. Subsequently, LinearSVC(tuning parameters) SGDclassifier, RandomForest, and DecisionTree classifiers were explored on preprocessed data.

I used 1 unigram and 1 bigram and kept other default parameters in the base LinearSVC model. I can see that the model is good at predicting non offensive tweets, but the number of true untargeted insults is very low. This shows that the model is not good at predicting unintended offensive tweets. The goal is to improve the predicting as measured by precision, recall and ultimately the micro and macro F1 score.

Next, I ran the same LinearSVC model on cleaned tweets with the addition of new features. During fine tuning of the model, setting df\_min = 5, max\_df = 0.9 and english - stopwords in CountVectorizer increased the recall value for TIN. To account for the unbalanced data, the parameter ‘class\_weight’ was set to balanced. This method uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data allowing better prediction of less frequent classes. These changes increased the number of UNT’s correctly from 9 to 25, leading to higher precision, recall and the macro F1 score (0.5515). This improvement came at the expense of the count of true positive predictions for targeted insult (TIN) tweets. The prediction for non-offensive tweets increased as well. The validation F1 score is slightly lower than the micro F1 score.



a) LinearSVC(Base) b) LinearSVC(with new Feature) c) SGDClassifier



d) LinearSVC (NF+tfidfVectorizer) e) RandomForest f) DecisionTree

Figure 3: Confusion matrix

The validation F1, Recall and micro average F1 scores improved dramatically, while the improvements for the macro F1 were smaller. These improvements were noted in all explored methods except for Randomforestclassifier. Looking at the confusion matrix, predictions (in terms of quantity of returned true positives) improved non offensive tweets for most of the method and the most improvement for UNT class came from LinearSVC classifiers. The quantity of returned true positive TINs decreased. Applying the tfidfVectorizer for linear didn’t improve the predictions.

Due to the strong imbalance in the data, all the algorithms seem to be biased towards the majority class and overwhelmingly predicting the non-offensive tweets. Lexicon based features and twitter mentions (@, #) seems to improve prediction of non-targeted insults. Other features such as count of ‘ellipses and name items (titled words) didn’t play a significant role in improving prediction, hence were not used in the final models. Summary of scores for all the metrics is provided below and classification reports are included in the code for further reference.

Given the higher score across all metrics, I chose the model with LinearSVC classifier where the balance weight\_level is set.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Classifier** | **Validation F1** | **Precision** | **Recall** | **F1 Macro** | **F1 Micro** | **Remark** |
| 1 | LinearSVC - Raw Data | 0.4652 | 0.4983 | 0.4737 | 0.4803 | 0.6928 | CountVectorizer, Base Model |
| 2 | LinearSVC | 0.7218 | 0.5342 | 0.5848 | 0.5515 | 0.7338 | CountVectorizer, Nef features |
| 3 | LinearSVC | 0.7173 | 0.5133 | 0.5434 | 0.5221 | 0.7121 | TfidfVectorizor, New Features |
| 4 | RandomForest | 0.7399 | 0.4591 | 0.5393 | 0.4714 | 0.7249 | CountVectorizer- New features |
| 5 | SGDclassifier | 0.7049 | 0.4707 | 0.5832 | 0.4936 | 0.7117 | CountVectorizer- New features |
| 6 | DecisionTreeClassifier | 0.4882 | 0.5036 | 0.5036 | 0.4949 | 0.672 | TfidfVectorizor, New Features |

Figure 4: Summary of evaluation metrics

**3.3 Error analysis**

Some aspects of the error analysis were expained in the previous section. Observing the confusion matrix reveals that these models are a good fit for the non-offensive class (NOT). With this best model, the recall values increased by 8% though the improvement in precision is negligible.

For the UNT class the number of true positives increased, while the number of false negatives and false negatives decreased leading to about 16% increase in precision and recall values. These improvements in recall came with a decrease in the number of returned true positive TIN tweets, as a result the recall value decreased by about 6%. Despite the decrease in true positive TIN tweets, the precision increased by 11% because of the decreased number of false positives. Prediction of 5 random tweets is listed below to examples of false positives and false negatives from the models.

|  |  |
| --- | --- |
| **LinearSVC Base Model**  precision recall f1-score support  NOT 0.78 0.81 0.80 1409  TIN 0.52 0.51 0.52 617  UNT 0.20 0.10 0.13 93 | **Model 2: Linear SVC with Features**  precision recall f1-score support  NOT 0.7833 0.8928 0.8345 1409  TIN 0.6140 0.4408 0.5132 617  UNT 0.3571 0.2688 0.3067 93 |

Figure 5: Classification Report for Linear SVC (Base Model vs Selected Model)

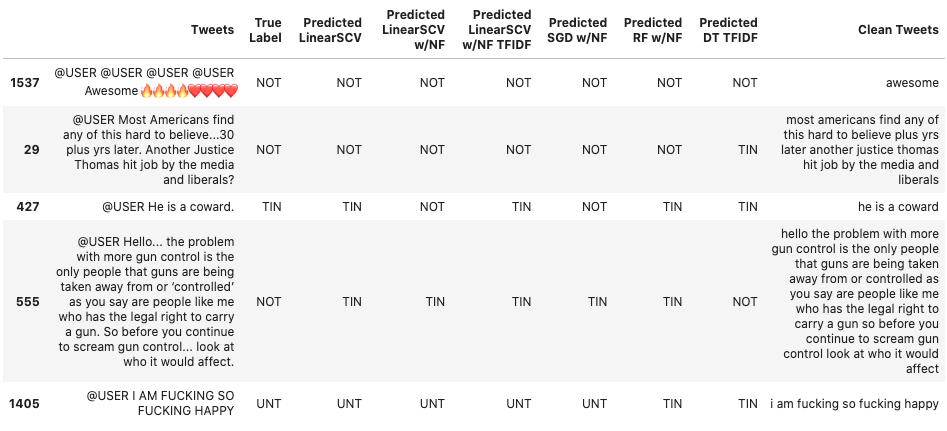
****

Figure 6: Summary of the tweet predictions by different models

Tweet **id 1537** is a good example of true positive returning across all the explored models, not surprisingly the class with all models in agreement is NOT. A good example for false negative is tweet **id# 427**, which is a targeted insult (TIN) class but the two of the models [final model (LinearSVC with new features) and SGD classifier], predicted it as non-offensive. This is an example of the limitations of the models specially with lexicon-based features.

**3.4 Prediction on test dataset**

The selected LinearSVC model was applied to predict the reserved test dataset and the predicted classes were appended to the test file provided. The resulting output of the prediction is summarised in the bar chart below. As expected, the majority of tweets were predicted to be not offensive (NOT: 75%) followed by TIN (22%) and UNT (3.5%).

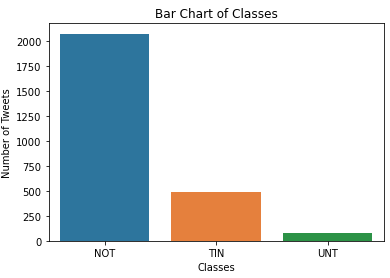


Figure 7: Bar plot for class distribution of the test predictions

1. **Conclusion**

Using relevant features such as lexicon-based features led to a better score. This is specially important as the data provided has a highly unbalanced class, addition of features that boost detection of the less frequent class were applied. Fine tuning the parameters in the vectorizers and classifiers was found to be useful. Using min\_df = 5 and max\_df = 0.9 and unigram-bigram features improved the scores. Selecting the balanced class\_level parameter in the classifiers was applied to address the unbalanced classes and bias to more frequent classes and prediction was improved in all models.

The LinearSVC model with additional features and fine tuning was selected as the best model as it had the highest scores across all metrics, with the most improvement achieved in validation F1 and micro scores. It was able to predict more UNT class compared to other models. The prediction on the test tweets labeled 75% of the tweets as not offensive.