Twitter Sentiment Analysis

Project Overview

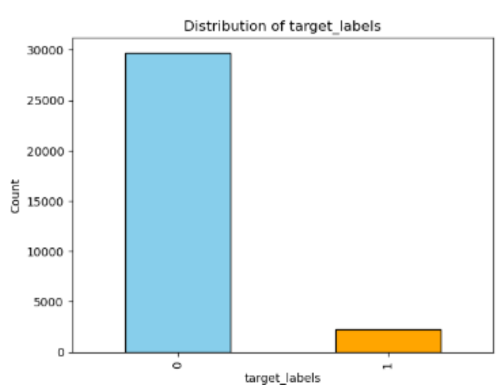
This project focuses on a Kaggle competition hosted by Twitter. The objective is to identify tweets that fall under the category of racism or sexism and potentially block them to reduce bullying and negative tweets as much as possible.

Exploratory-Data-Analysis

```
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      31962 non-null  int64
 1   label   31962 non-null  int64
 2   tweet   31962 non-null  object
dtypes: int64(2), object(1)
```

id	label	tweet
24972	0	be n #healthy do not speak in the hearing of...
24469	0	#late #ff to my #gamedev #indiedev #indiegam...
15099	0	after tonight, no more basketball or football.
27073	0	outrageously busy. we've even sold our bunting...
8386	0	ó¾¬¥ó¾□□ó¾¬□ó¾□ #daughter @user just got #gra...

1. We observe that each row contains one tweet and a label indicating whether they are fall under the cateogry explained above. Thus, this is a `BINARY CLASSIFICATION PROBLEM`.
2. As part of basic preprocessing, we drop any rows that are duplicates or contain null values.
3. Next, we check for **class imbalance**, an important consideration in classification tasks. To address class imbalance, we assign a higher penalty when the model misclassifies a minority class during training. This approach encourages the model to pay attention to minority classes, even if the overall accuracy is high.



Pre-Processing

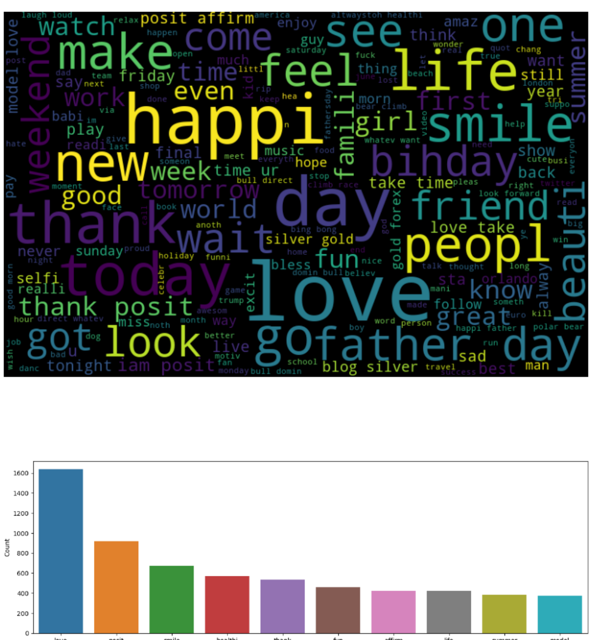
As part of the preprocessing steps, I have performed the following transformations:

1. Converted all text to lowercase.
2. Replaced emojis with their meanings.
3. Some pre-encoded emojis that are not demojized back to their meanings will be simply removed.
4. Expanded contractions (e.g., `'ve` to `have`).
5. Replaced special characters with their names, except # (trends), as it will be useful in creating new features that will assist in classification later.
6. Although @ is used to tag someone, to maintain privacy, the data has usernames replaced by "user" which is not useful here, so we will just drop them.
7. Removed HTML tags.
8. Abbreviated common terms (e.g., `GM` for `Good Morning`).
9. Removed stop words.
10. Applied stemming to reduce words to their root forms.

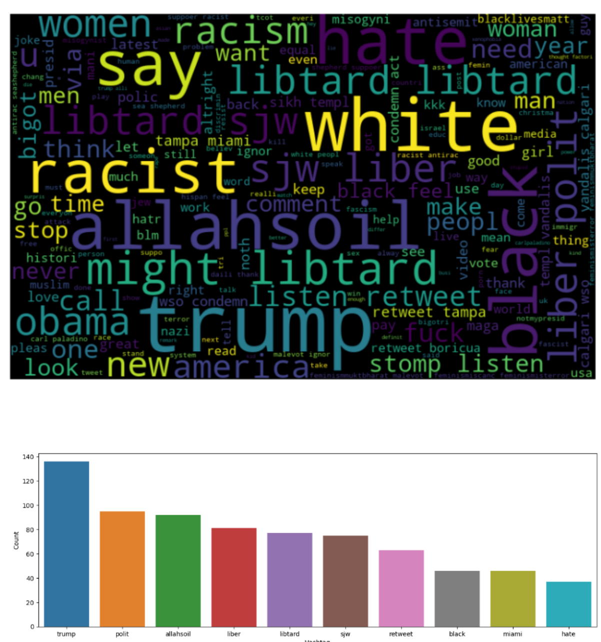
- **Why not perform spelling correction?** Simply because the data is too large, and it would consume a significant amount of time.

Feature-Engineering

As mentioned earlier, we will extract the trends represented by # from the tweets and analyze whether they are mainly used to represent racist or non-racist tweets.



(a) common words used in non-racist/sexist tweets



(b) common words used in racist/sexist tweets

Now, we create a corpus of words from the tweets column of both training and testing data so as to avoid out of vocabulary issue as much as possible. Then, we select the top 1000 most frequent words as the dimensions of vectors to represent each sentence. We will do a comparison between the Bag of Words and TF-IDF methods.

`We will now combine these 1000 columns with the 20 features we created earlier (10 most common labels from each category) making a total of 1020 final features`

Model Training & Evaluation

We will use a Logistic Regression classifier. The f1_score we obtain after training is approximately 0.544 using the Bag of Words (BOW) method and 0.559 using the TF-IDF method.