Offensive Language detection

Kusum Pareek

Email:pareekkp20@gmail.com

PROBLEM STATEMENT

Hate Speech Detection is the automated task of detecting if a piece of text contains hate speech.

ABSTRACT

The presence of offensive language on social media platforms and the implications this poses is becoming a major concern in modern society. Given the enormous amount of content created every day, automatic methods are required to detect and deal with this type of content. With the rise of hate speech phenomena significant research efforts have been undertaken in order to provide automatic solutions for detecting hate speech, varying from simple machine learning models to more complex deep neural network models

Data Preparation

Our target variable is a categorical variable and denotes which class each tweet belong to.And there are three classes: 0 denotes that a tweet contains hate language; 1 denotes that a tweet do not contain hate language but contain offensive language; 2 denotes that a tweet includes neither hate nor offensive language

Machining learning models are not able to deal with text directly, so we need to transform text of each tweet into numerical or categorical features. Tweets are highly unstructured and not formal writing including emoij and abbreviation, so the first step is to clean text of tweets for tokenization. We converted text of each tweet into a list of words through tokenization and further cleaning, and then transformed the

word list of each tweet into a feature vector based on the bag of words model.

Data preprocessing-

steps for preprocessing:

1.removal of punctuations and numbers

2. remove whitespace with a single space

3. removal of capitalization

4. tokenizing

5.remove words beginning with @

	class	tweet	text length	processed_tweets
0	2	!!! RT @mayasolovely: As a woman you shouldn't...	140	rt as a woman you shouldn t complain about cle...
1	1	!!!!! RT @mleew17: boy dats cold...tyga dwn ba...	85	rt boy dats cold tyga dwn bad for cuffin dat h...
2	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...	120	rt dawg rt you ever fuck a bitch and she start...
3	1	!!!!!!!! RT @C_G_Anderson: @viva_based she lo...	62	rt she look like a tranny
4	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...	137	rt the shit you hear about me might be true or...
5	1	!!!!!!!!!!!!!!!"@T_Madison_x: The shit just...	158	the shit just blows me claim you so faithful a...
6	1	!!!!!!"@__BrighterDays: I can not just sit up ...	105	i can not just sit up and hate on another bitc...
7	1	!!!!“@selfiequeenbri: cause I'm tired of...	98	cause i m tired of you big bitches coming for ...
8	1	" & you might not get ya bitch back & ...	58	amp you might not get ya bitch back amp thats ...
9	1	" @rhythmixx_ :hobbies include: fighting Mariа...	55	hobbies include fighting mariam bitch

Data Understanding-

The class label id defined majority of users:
class 0: hatespeech
class 1:offencive language
class 2:neither or non offencive

count: no of users who coded each tweet
hate_speech: no of users who judged the tweet to be hatespeech
offencive_language: no of users who judged the tweet to be offencive language
neither: no of users who judged the tweet to be neither hate nor offencive

Technology used:

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Modeling and Evaluation

LR:

Pros:

- Provides probabilities for outcomes
- Multi-col linearity is not really an issue and can be conquered by regularization to some extent
- Low variance

Cons:

- High bias
- Does not perform well when feature space is too large
- Sensitive to outliers

Result

Logistic Regression

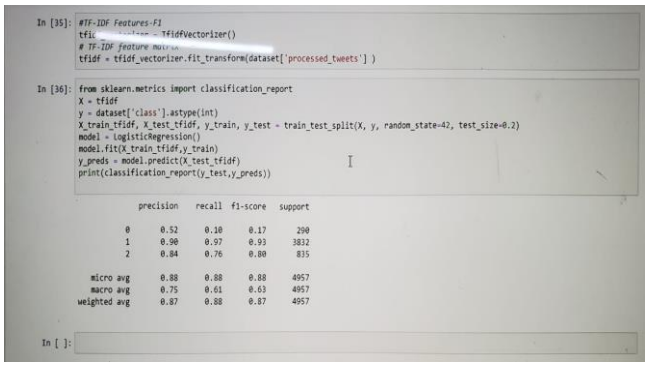
It is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X.

Conclusion:

The use of offensive language in user-generated content is a serious problem that needs to be addressed with the latest technology.
the area of Offensive Language Detection helped us to explore the dataset

We explored the various techniques of handling the imbalance in the training
set, but for our dataset; TFIDF vectorizer worked well

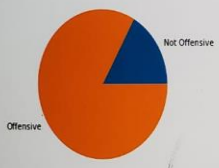
output



```
In [35]: #TF-IDF Features-F1
         tfi[redacted] = TfidfVectorizer()
         # TF-IDF feature matrix
         tfidf = tfidf_vectorizer.fit_transform(dataset['processed_tweets'] )

In [36]: from sklearn.metrics import classification_report
         X = tfidf
         y = dataset['class'].astype(int)
         X_train_tfidf, X_test_tfidf, y_train, y_test = train_test_split(X, y, random_state=42, test_size=0.2)
         model = logisticRegression()
         model.fit(X_train_tfidf,y_train)
         y_preds = model.predict(X_test_tfidf)
         print(classification_report(y_test,y_preds))

                       precision    recall  f1-score   support

                    0       0.52      0.10      0.17       290
                    1       0.90      0.97      0.93      3832
                    2       0.84      0.76      0.80       835

             micro avg       0.88      0.88      0.88      4957
             macro avg       0.75      0.61      0.63      4957
          weighted avg       0.87      0.88      0.87      4957

In [ ]:
```

to show percentage of offencive and non offencive tweets:

```
In [131]: import matplotlib.pyplot as plt
          offensive=sum(dataset['class']==1)+sum(dataset['class']==0)
          Not_Offensive=sum(dataset['class']==2)
          total=dataset['class'].count()
          print("Offensive tweets:",round(offensive*100/total,2),'%')
          print("Non Offensive tweets:",round(Not_Offensive*100/total,2),'%')
          y = np.array([round(Not_Offensive*100/total,2),round(offensive*100/total,2)])
          mylabels = ["Not Offensive", "Offensive"]
          plt.pie(y, labels = mylabels)
          plt.show()

          Offensive tweets: 83.2 %
          Non Offensive tweets: 16.8 %
```



text-length:

```
In [132]: dataset['text length'] = dataset['tweet'].apply(len)
          print(dataset.head())

             Unnamed: 0  count  hate_speech  offensive_language  neither  class  \
          0           0      3            0                   0        3      2
          1           1      3            0                   3        0      1
          2           2      3            0                   3        0      1
          3           3      3            0                   2        1      1
          4           4      6            0                   6        0      1

                                                      tweet  text length
          0  !!! RT @mayasolovely: As a woman you shouldn't...          140
          1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...           85
          2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...          120
          3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...           62
          4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...          137

In [133]: import seaborn as sns
          import matplotlib.pyplot as plt
          graph = sns.FacetGrid(data=dataset, col='class')
          graph.map(plt.hist, 'text length', bins=50)

Out[133]: <seaborn.axisgrid.FacetGrid at 0x1f18ef4bac8>
```