Toxic comment classification

Problem Statement:

Given a comment made by the user, predict the toxicity of the comment.

Columns in train data:

- Comment_text: This is the data in string format which we have to use to find the toxicity.
- target: Target values which are to be predicted (has values between 0 and 1)
- Data also has additional toxicity subtype attributes: (Model does not have to predict these)
  - severe_toxicity
  - obscene
  - threat
  - insult
  - identity_attack
  - sexual_explicit
- a subset of comments have been labelled with a variety of identity attributes, representing the identities that are mentioned in the comment. The columns corresponding to identity attributes are listed below. Only identities with more than 500 examples in the test set (combined public and private) will be included in the evaluation calculation. These identities are shown in bold.
  - male
  - female
  - transgender
  - other_gender
  - heterosexual
  - homosexual_gay_or_lesbian
  - bisexual
  - other_sexual_orientation
  - christian
  - jewish
  - muslim
  - hindu
  - buddhist
  - atheist
  - other_religion
  - black
  - white
  - asian
  - latino

- other_race_or_ethnicity
- physical_disability
- intellectual_or_learning_disability
- psychiatric_or_mental_illness
- other_disability

Example Datapoints and Labels:

Comment: i'm a white woman in my late 60's and believe me, they are not too crazy about me either!!

- Toxicity Labels: All 0.0
- Identity Mention Labels: female: 1.0, white: 1.0 (all others 0.0)

Comment: Why would you assume that the nurses in this story were women?

- Toxicity Labels: All 0.0
- Identity Mention Labels: female: 0.8 (all others 0.0)

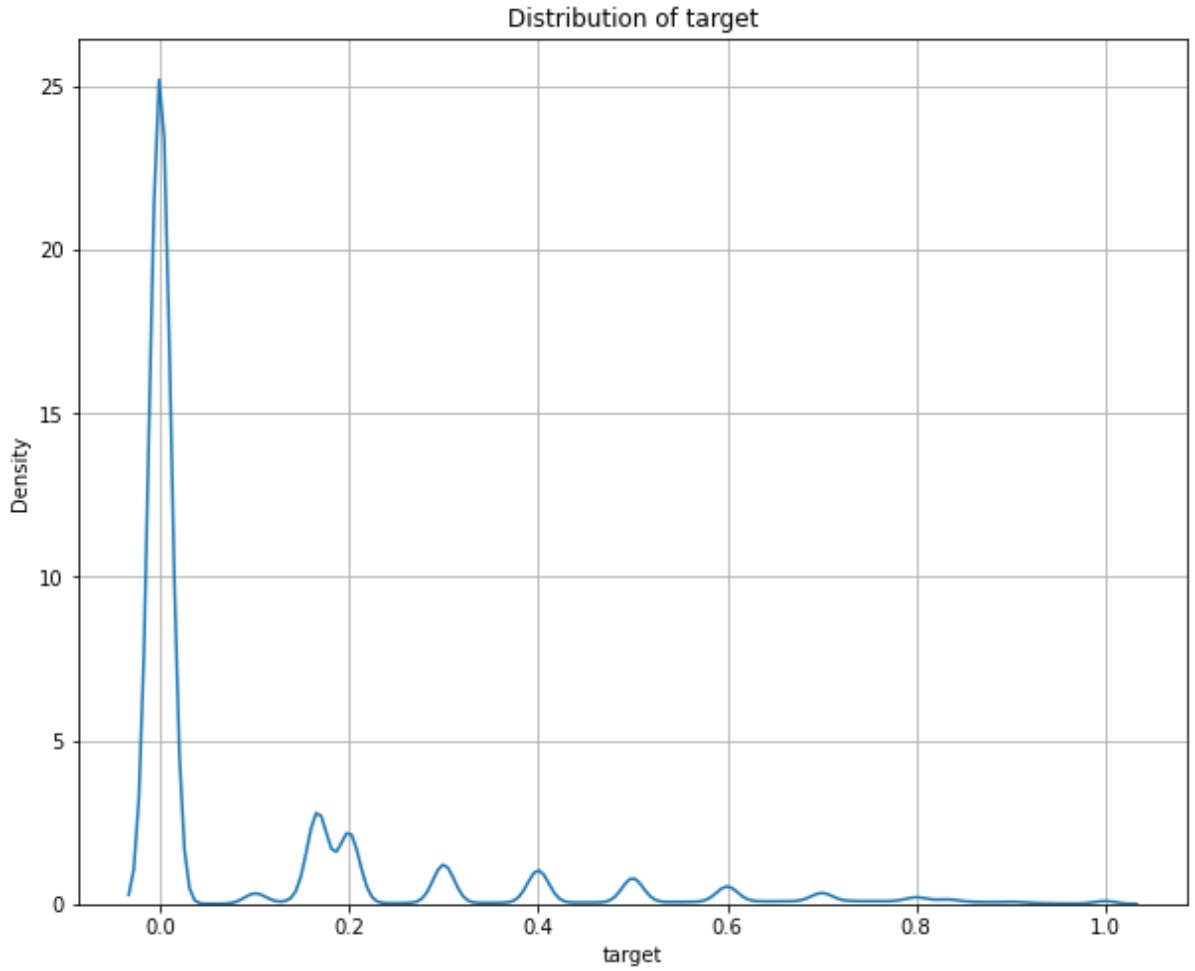
Comment: Continue to stand strong LGBT community. Yes, indeed, you'll overcome and you have.

- Toxicity Labels: All 0.0
- Identity Mention Labels: homosexual_gay_or_lesbian: 0.8, bisexual: 0.6, transgender: 0.3 (all others 0.0)

Data Modelling approach

- Check for missing values:
- Check distribution of toxicity in data
- Pre-process column text
- Split data into train and test
- Built machine learning models on the data
- Hyper parameter tuning
- Compare mean squared error of different model using graph
- feature importance

Distribution of target variable

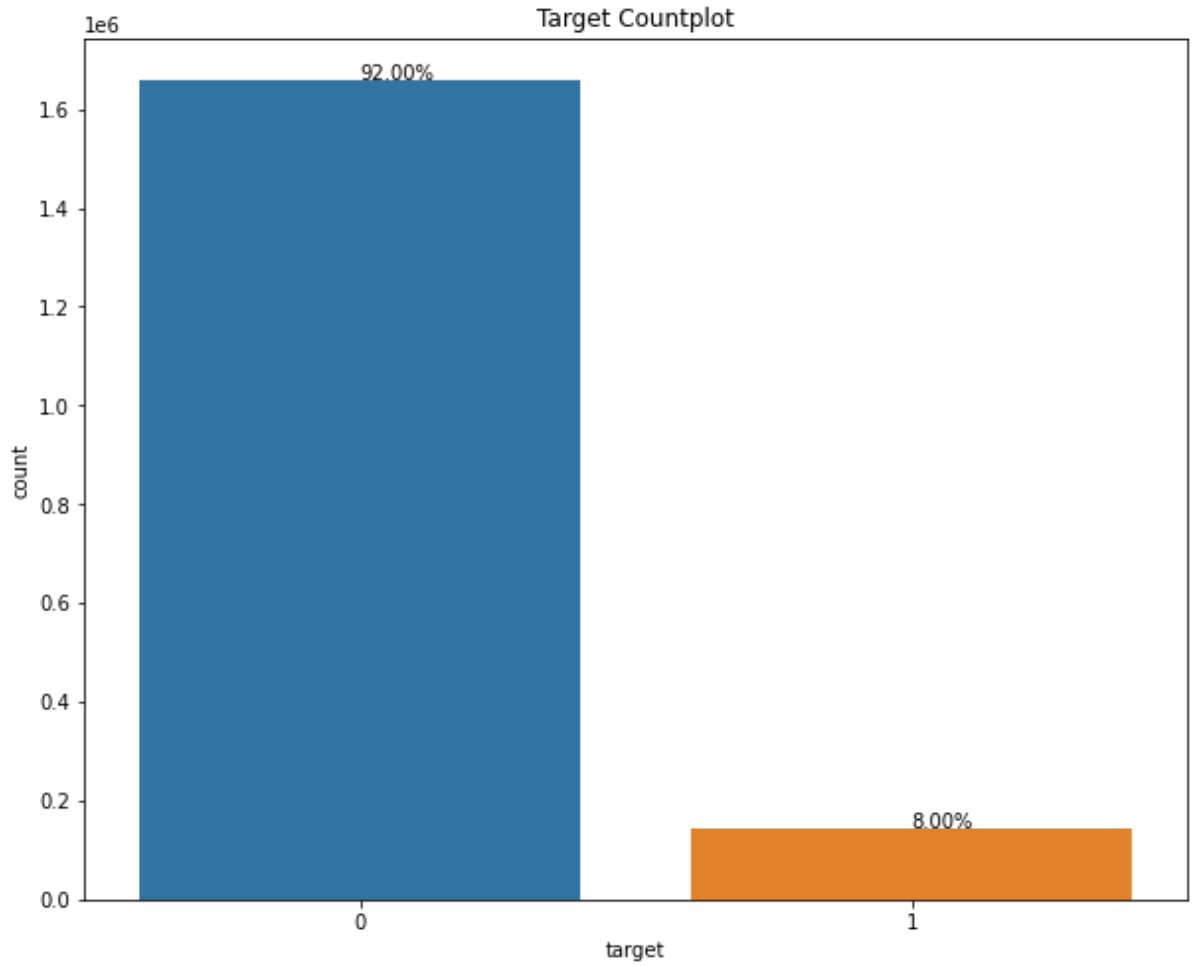


We convert our dataset into classification variable

If toxicity>0.5 then 1

Else 0

Countplot of target variable



Our dataset is unbalanced there 8% toxic comment and 92% non-toxic comment

Divide our dataset features in different category like based on region, gender, and additional toxicity features

And try to see distribution of data on toxic and non-toxic comment

Additional toxicity features:

- severe_toxicity
- obscene
- threat
- insult
- identity_attack
- sexual_explicit

features based on gender

- male
- female
- homosexual_gay_or_lesbian
- bisexual
- heterosexual
- other_gender
- transgender

Build comment_type function that tells which type of comment this is
exa: insult, threat, severe_toxicity,identity_attack

Draw countplot that tells which type of comment most are in on features like religion, gender

Features generated by users feedback

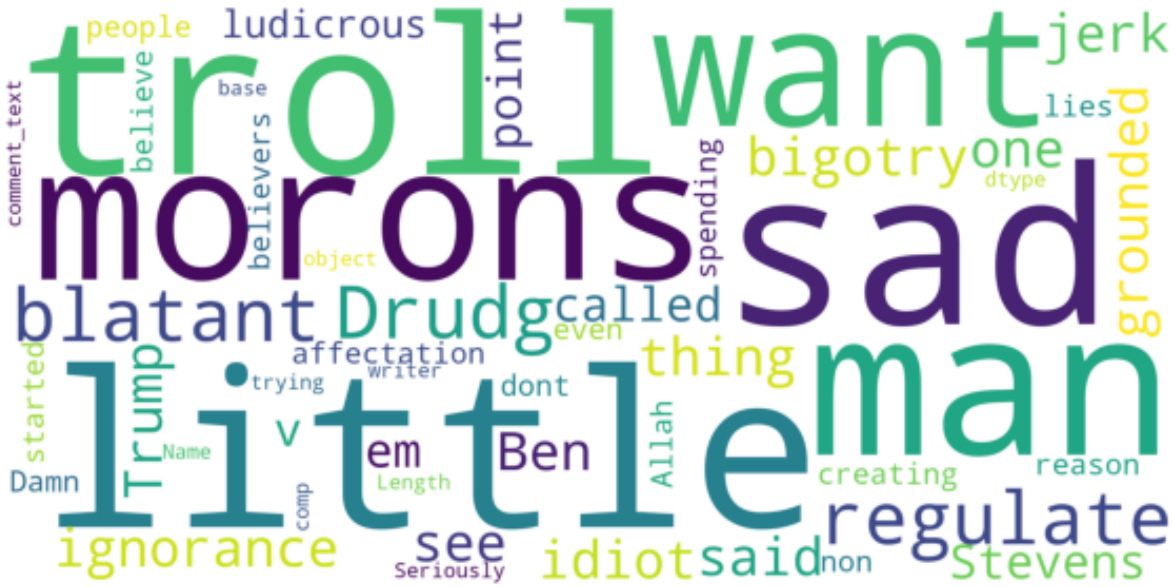
- funny
- sad
- wow
- likes

- disagree

Draw countplot of user feedback features

Draw wordcloud using wordcloud library

Word cloud where toxicity in target variable is grater then 0.75



Prevalent words in comment data where target>0.75

Word cloud where toxicity in insult variable is grater then 0.75

Prevalent words in comment data where threat score>0.75

Pre-process comment_text column

- Convert to lower
- Remove characters other then (A-Z, a-z,0-9)
- Split string
- Stemming words
- Join words

Divide data into train and test

10% test and rest train

Using Count vectorizer and tfidf vectorizer convert text data into numbers

Build model using SGD regressor and decision tree

Compare models using different hyper parameter

See feature importance and coefficient of words

Best model = SGD regressor with tfidf vectorization method

Mean Squared Error on train set: 0.023313194856301823

Mean Squared Error on cv set: 0.023324606373848766