# PROJECT REPORT

**ON**

**TOXIC COMMENTS CLASSIFICATION IN SOCIAL NETWORKING**

By

SINGAM TRIVENI

ON

DEEP LEARNING

Under the guidance

**Smart Bridge**

Mentor/project guide: Mahankali Surya Tej

# Toxic Comments Classification In Social Networking

## Project Overview

Nowadays, the flow of data over the internet has grown dramatically, especially with the appearance of social networking sites. Social networks sometimes become a place for threats, insults, and other components of cyberbullying. A huge number of people are involved in online social networks.

Toxic comments are textual comments with threats, insults, obscene, racism, etc. In recent years there have been many cases in which authorities have arrested some users of social sites because of the negative (abusive) content of their personal pages. Hence, the protection of network users from anti-social behaviour is an important activity. One of the major tasks of such activity is automated detecting the toxic comments.

Bag of words statistics and bag of symbols statistics are the typical source information for the toxic comments detection. Usually, the following statisticsbased features are used: length of the comment, number of capital letters, number of exclamation marks, number of question marks, number of spelling errors, number of tokens with non-alphabet symbols, number of abusive, aggressive, and threatening words in the comment, etc. A neural network model is used to classify the comments.

## Problem Statement

Given a group of sentences or paragraphs, which was used as a comment by a user in an online platform, our task is to classify it to belong to one or more of the following categories – toxic, severe-toxic, obscene, threat, insult or identityhate with approximate probabilities or discrete values .This is a Multi-label Classification problem.

A multi-label classification problem differs from a multi-class classification problem (in which each sample can only be assigned to one of the many-labels). Hence in our problem each comment can belong to more than one label for eg, a comment can be obscene, toxic and insult, all at the same time.

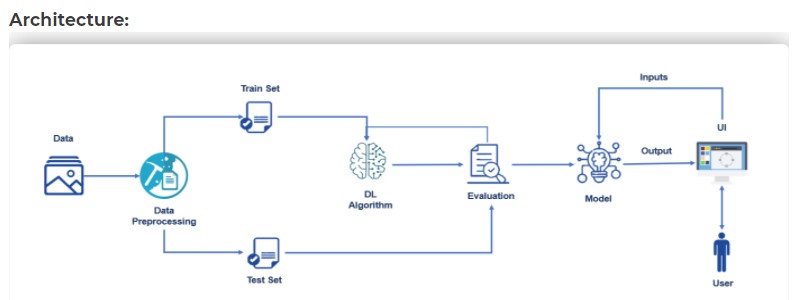
Solving a multi-label classification problem

1.One way to approach a multi-label classification problem is to transform the problem into separate single-class classifier problems. This is known as 'problem transformation'. There are three methods:

1. Binary Relevance: This is probably the simplest which treats each label as a separate single classification problems. The key assumption here though, is that there are no correlation among the various labels.

1. Classifier Chains: In this method, the first classifier is trained on the input X. Then the subsequent classifiers are trained on the input X and all previous classifiers' predictions in the chain. This method attempts to draw the signals from the correlation among preceding target variables.

1. Label Powerset: This method transforms the problem into a multiclass problem where the multi-class labels are essentially all the unique label combinations. In our case here, where there are six labels, Label Powerset would in effect turn this into a 2^6 or 64-class problem.



### Data Exploration

The dataset consists of the following fields-

Id: A hexadecimal value, to identify the person who had written this comment

Comment \_text: A multi-line text field containing the exact comment

Toxic: binary label containing 0/1

Severe \_toxic: binary label containing 0/1

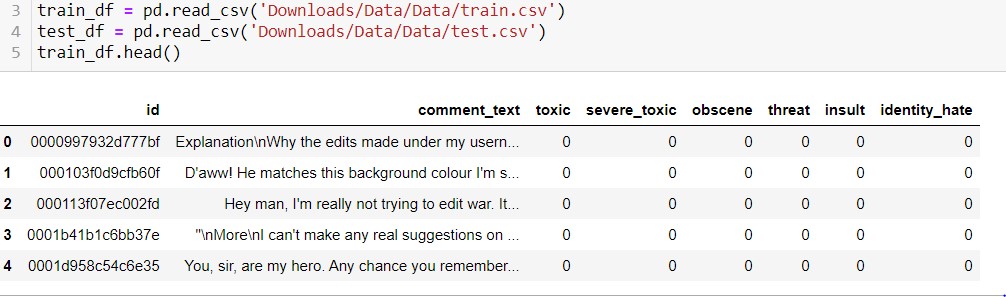
Obscene: binary label containing 0/1

Threat: binary label containing 0/1

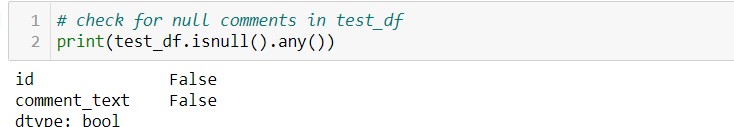
Insult: binary label containing 0/1

Identity \_hate: binary label containing 0/1

Out of these fields, the comment\_text field will be preprocessed and fitted into different classifiers to predict whether it belongs to one or more of the labels/outcome variables.

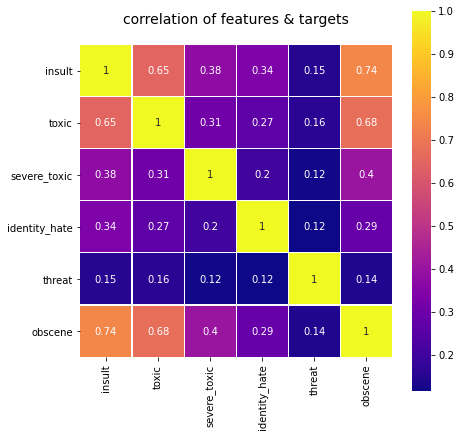


### Checking for null comments



### Correlations

Correlation analysis is a statistical method used to evaluate the strength of relationship between two quantitative variables. A high correlation means that two or more variables have a strong relationship with each other, while a weak correlation means that the variables are hardly related



Indeed, it looks like some of the labels are higher correlated, e.g.insultobscene has the highest at 0.74, followed by toxic-obscene and toxic-insult. This co-relation can be taken in use for other kind of algorithms, such as multichain labelling

### Data Preprocessing

Preprocessing involved the following steps:

✤ **Preparing a string containing all punctuations to be removed :**

1. The string library contains punctuation characters. This is imported and all numbers are appended to this string.
2. Also, we can notice that our comment\_text field contains strings such as won't, didn't, etc which contain apostrophe character('). To prevent these words from being converted to wont/didnt, the character ' represented as \' in escape sequence notation is replaced by empty character in the punctuation string.

✤ Applying Count Vectoriser :

1. Count Vectoriser is used for converting a string of words into a matrix of words with column headers represented by words and their values signifying the frequency of occurrence of the word.
2. It accepts stop words, convert to lowercase( set as true), and regular expression as its parameters. Here, we will be supplying our custom list of stop words created earlier and using lowercase option. Regular expression will have its default value.



Binary Relevance - build a multi-label classifier using Logistic Regression

### Model Building and Fitting

Import and instantiate the Logistic Regression model. The model is trained using the train data and then the training accuracy is computed.



# Application Building

**Building an Application to integrate the model**

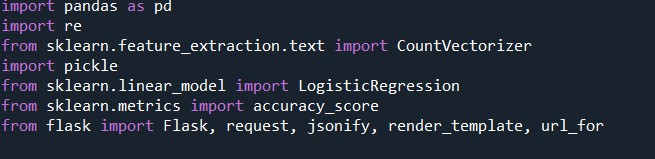
After the model is built, we will be integrating it into a web application so that normal users can also use it to know if any comment is toxic or not. In the application, the user provides any text(comment) to check and analyzed by calling the different label model files generated during cleaning. Finally, the toxicity of the comment is displayed on the UI.

# Flask App

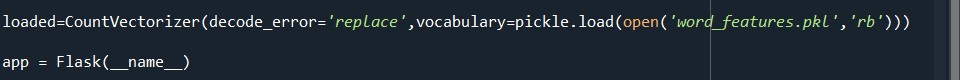
**Build the python flask app**

In the flask application, the comment is taken from the HTML page and it is cleaned and processed. Using this processed text predictions are performed and are sent back to the HTML page to notify the user.

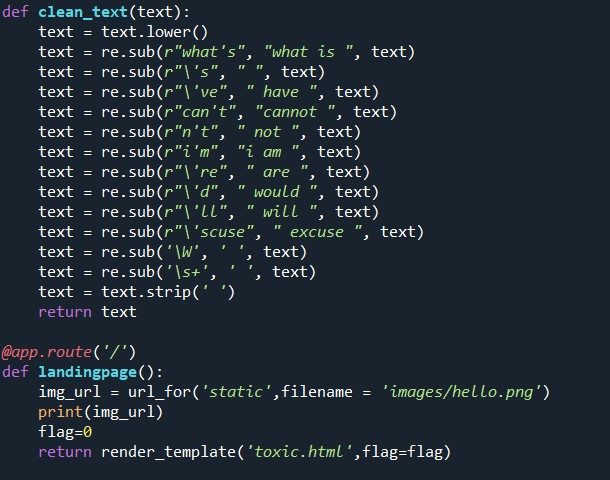
**Step - 1:** Import the required libraries

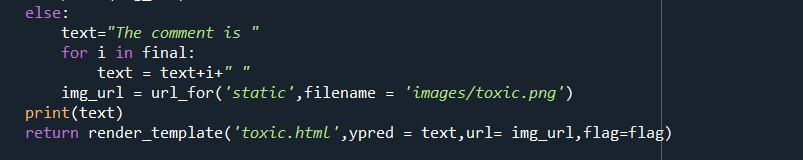
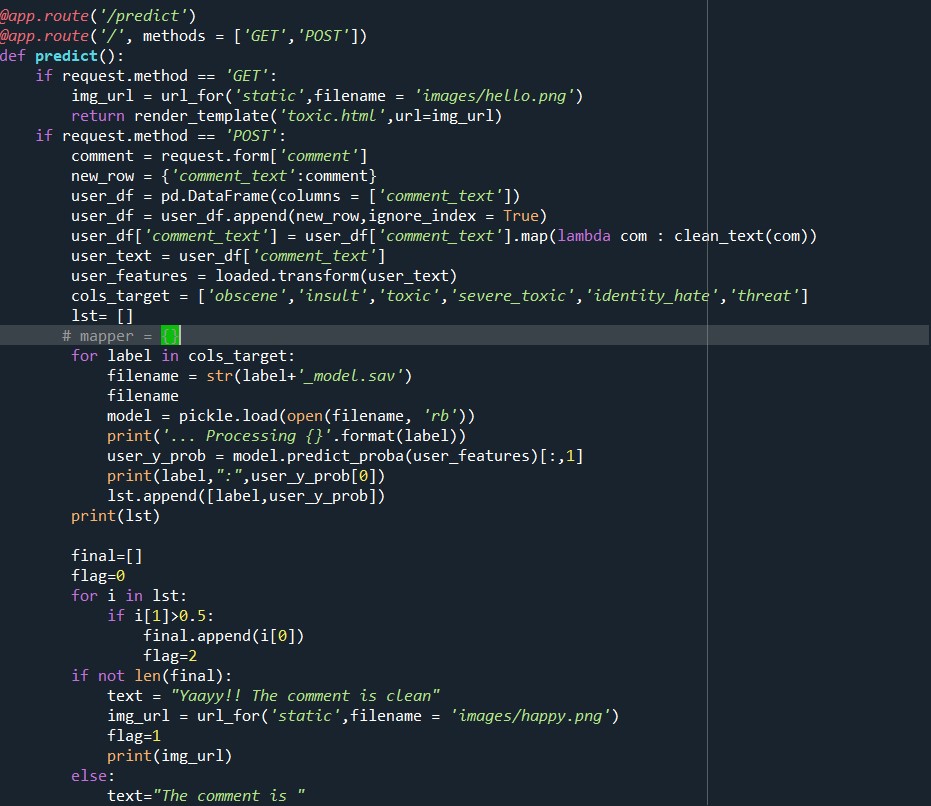


**Step - 2:** Load the count vectorizer and initialize flask app

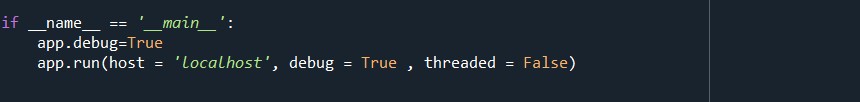


**Step - 3:** Configure app.py to fetch the URL from the UI, process the text, and return the prediction





**Step 4:** Run the app



# Build An HTML Page

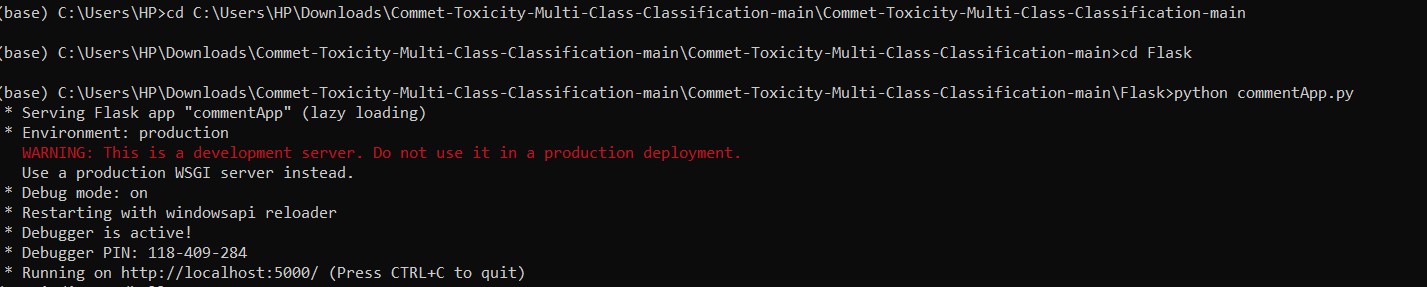
We Build an HTML page to take the text from the user and upon clicking on the button for submission it has to redirect to the URL for “**y\_predict**” which returns if the text given is toxic or safe. The output is to be then displayed on the page.

The HTML pages are put under the templates folder and any style sheets if the present is put in the static folder.

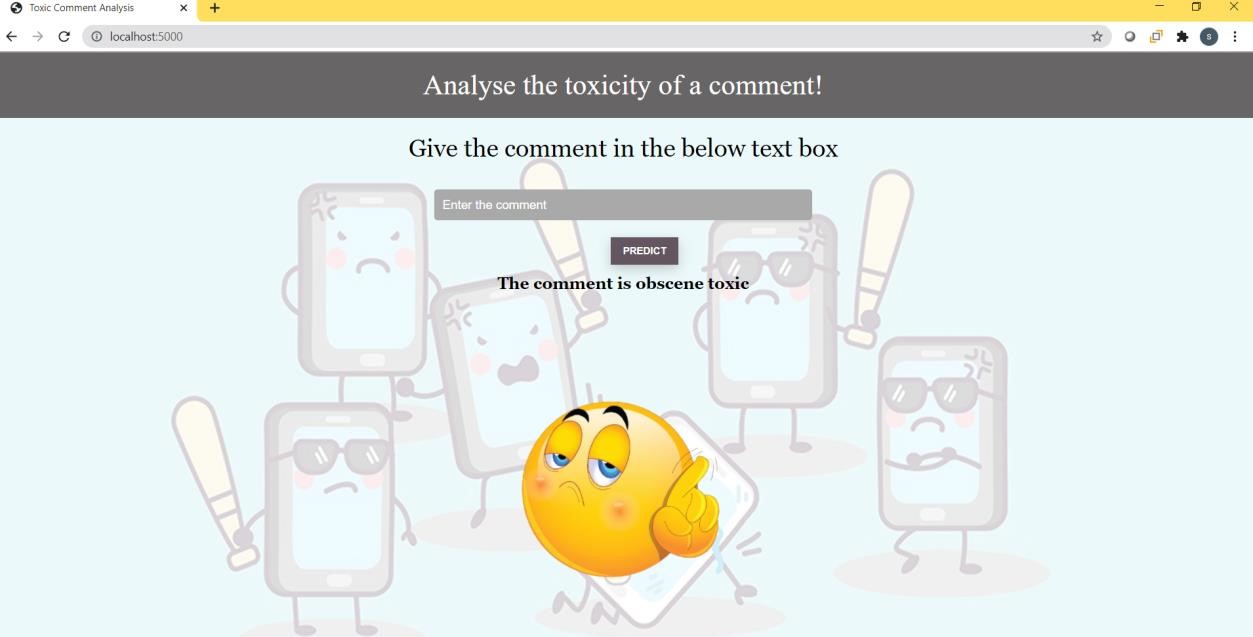
# Execute And Test The Model

**Now we execute the model using Anaconda Prompt**

Execute the python code and after the module is running, open the localhost page and give the text to be predicted or tested.



For a toxic comment, the prediction is:



For the comment "it is amazing", the prediction is:

