Mini Project On

Toxic Comments Classfication

By

Tejas Kamble (2021510024)

Under the guidance of

Internal Supervisor

Prof. Dr. Pooja Raundale

Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University

2020-21

CERTIFICATE OF APPROVAL

This is to certify that the following students

Tejas Kamble (2021510024)

Have satisfactorily carried out work on the project
entitled

"Toxic Comments Classfication"

Towards the fulfilment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2020-21.

Project Guide:
Prof. Dr. Pooja Raundale

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Tejas Kamble (2021510024)

Have successfully completed the Project report on

"Toxic Comments Classfication",

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER EXTERNAL EXAMINER

HEAD OF DEPARTMENT PRINCIPAL

Contents

Abstract

Discussing things you care about can be difficult. The threat of abuse and harassment online means that many people stop expressing themselves and give up on seeking different opinions. Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments.

In the modern world, where the internet has reached worldwide providing a source for connecting people from all around the world it is also spreading love and hate at the same speed. Social media has become an online platform where people can express themselves freely and communicate with anyone from all around the world.

People talk about their thoughts and opinions and try to connect with other people. Although most people accept others as they are there are always some people who use toxicity when showing anger or hate. Popular social media platforms have strict rules for this kind of thing. The platforms classify these comments into several categories and then block or remove those comments which are too toxic. This is done using machine learning.

Objectives

Our objective is to create social media better place.

- To create a safe and secure environment for people to express their feelings.
- To avoid unnecessary hatred spreading though social media.
- To make social media safer and better place.

List of Figures

List of Tables

1 Introduction

In today's world where social media is also a big part of our life somehow become the biggest platform where we can share our thoughts and opinions. Nearly 4,480,000,000 people use social media daily. Teenagers are the ones who use social media most of the time. And is the most influenced age group by social media. Most social media includes Instagram, Facebook, Twitter, Pinterest, Reddit, Youtube, Tumblr, etc.

In this social media, people need to be able to express their thoughts and opinions freely without any fear. But some of the population always wants to oppose them with toxicity. To keep the social media free space to express themselves. Most people oppose using comments or using the report option. As though the platforms check this report to verify and then they take action so it's somehow safe. But when it comes to comments which are open to the public eye get the attention of the public quickly. In those cases, these platforms use Sentimental Analysis to filter out these toxic comments.

Sentiment analysis is a process of classifying textual data into classes. For this research, I've used the Toxic Comments Classification Challenge dataset. The dataset contains toxic comments collect from Wikipedia. The objective of this research is to create a model which can help filter out these toxic comments from normal ones. We will be using Multi-label Classification for this purpose which can categorize the comments into one or more categories by their toxicity level.

1.1 Problem Definition

The threat of abuse and harassment online means that many people stop expressing themselves and give up on seeking different opinions. Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments.

Wikipedia has created a challenge called "Toxic Comments Classification" on Kaggle for creating the most model that can classify a toxic comment accurately. We will try to take solve this problem.

1.2 Objectives and Scope

1.2.1 Objectives

Our objective is to build a multi-headed model that's capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate better than Perspective's current models.

- To create a safe and secure environment for people to express their feelings.

- To avoid unnecessary hatred spreading though social media.
- To make social media safer and better place.

1.2.2 Scope

The model is able to classify the toxic comments from the dataset accurately. The model can be used on different social media platforms and will be used to stop or block toxic comments before publishing.

- The model can be used to enhance the quality of social media.
- Categorize comments in different-different types.
- Increase accuracy of the model to create better results.
- This model will also be used by the various organizations and media platforms.
- Also it can be used in e-commerce site to review the reviews.

1.3 Existing System

All social media use machine learning to classify the comments before publishing them. These social media are able to classify if the comment is toxic while this comment is being published. Even though social media is aware that those comments are toxic they might not able to tell the level of toxicity of those comments.

Some of the disadvantages of the existing system are as follows :

- The social media is unable to classify the comments into categories by their toxicity level.
- Social media platforms can categorize toxic words and ban them.

1.4 Proposed System

The model is able to classify the comment and how many categories it belongs to. We will be using Multi-Label Classification for this purpose and thus the model will be able to classify the comment for each label separately.

We will be using the dataset provided by Kaggle's Toxic Comments Challenge which includes Wikipedia comments.

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:
Some of the advantages of our system are as follows :

- Binary Relevance: This is probably the simplest which treats each label as a separate single classification problem. The key assumption here though, is that there is no correlations among the various labels.
- 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.
- Label Powerset: This method transforms the problem into a multi-class problem where the multi-class labels are essential all the unique label combinations. In our case here, where there are six labels, Label Powerset would in effect turn this into a 2 raise to 6 or 64-class problem.

1.5 System Requirements

- Harware Requirements

Table 1.5.1: Harware Requirements

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space for smooth run

- Software Requirements

Table 1.5.3: Software Requirements

Operating System	Windows, MacOS, Linux.
Web Browser	Mozilla Firefox, Google Chrome, Microsoft Edge.
Application	Google Colab, Jupyter, Anaconda.

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

The purpose of this project is to analyze and research different methods and finding the best solution for Multi-Label Classification. The Kaggle website had created a challenge by Wikipedia. In the challenge the datasets are provided of the comments from wikipedia. The comments are categories in one or more label by their toxicity level. Our purpose is to classify comments provided by wikipedia into this six categories and analyze them.

The comments are classified in following categories:

- Toxic
  Toxicity on normal level which is only meant to show anger on the individual, society or an organization.
- Severe Toxic
  Extreme toxicity in which people intend to hurt the individual mentally. One can lead the troma or the mental health issues. One can lead to riots in when commented on one group of society.
- Obscene
  Obscene meaning the comment is to offend people or organization. This comments are not that toxic but are intend to hurt their emotions.
- Insult
  Insult may or may not be included in severe toxic. It is meant to disrespect an individual or an organization.
- Threat
  Threat is intend to make people threated not only mentally but to show that it meant for harming physically (Individual) or legally (Organizations).
- Identity Hate
  This comments are just intended show hate and generally do have severe toxicity in it.

2.2 Definition

To build a Training and Placement Application so the students can have an easy to go placement process.

- ML - Machine Learning
  Machine learning is a branch of artificial intelligence (AI) and computer science that focuses on using data and algorithms to imitate how humans learn, gradually improving its accuracy.

- NLP - Natural Language Processing
  NLP stands for Natural Language Processing, which is a part of Computer Science, Human language, and Artificial Intelligence. It is the technology that is used by machines to understand, analyse, manipulate, and interpret human's languages. It helps developers to organize knowledge for performing tasks such as translation, automatic summarization, Named Entity Recognition (NER), speech recognition, relationship extraction, and topic segmentation.

- Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

- Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

- Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, and WxPythonotTkinter. It can also be used in Python and IPython shells, Jupyter notebook, and web application servers.

- Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top of the matplotlib library and is also closely integrated with the data structures from pandas.

- Scikit-learn (Sklearn) is the most useful and robust library for machine learning. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistency interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy, and Matplotlib.

2.3 Overall Description

2.3.1 Functional Requirements

1. Collecting Data

   The data is collected from the Kaggle website which is one of the largest website which provides datasets for all sorts of purposes. The Wikipedia holded a competition named "Toxic Comments Classification Challenge". In which the Wikipedia has provided four datasets.

   The number of the datasets is 4.
   Train, Test, Testlabels and Samplesubmission in .csv format.

2. Preparing the Data

   After you have your data, you have to prepare it. You can do this by:

   Putting together all the data you have and randomizing it. This helps make sure that data is evenly distributed, and the ordering does not affect the learning process.

   Cleaning the data to remove unwanted data, missing values, rows, and columns, duplicate values, data type conversion, etc. You might even have to restructure the dataset and change the rows and columns or index of rows and columns.

   Visualize the data to understand how it is structured and understand the relationship between various variables and classes present.

   Splitting the cleaned data into two sets - a training set and a testing set. The training set is the set your model learns from. A testing set is used to check the accuracy of your model after training.

3. Choosing a Model

   A machine learning model determines the output you get after running a machine learning algorithm on the collected data. It is important to choose a model which is relevant to the task at hand. Over the years, scientists and engineers developed various models suited for different tasks like speech recognition, image recognition, prediction, etc. Apart from this, you also have to see if your model is suited for numerical or categorical data and choose accordingly.

4. Splitting the Data

The dataset will bewsplit into two sets the training set and the testing set. The 80

5. Training the Model

   Training is the most important step in machine learning. In training, you pass the prepared data to your machine-learning model to find patterns and make predictions. It results in the model learning from the data so that it can accomplish the task set. Over time, with training, the model gets better at predicting.

6. Evaluating the Model

   After training your model, you have to check to see how it's performing. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that you split our data into earlier. If testing was done on the same data which is used for training, you will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give you disproportionately high accuracy.

   When used on testing data, you get an accurate measure of how your model will perform and its speed.

7. Parameter Tuning

   Once you have created and evaluated your model, see if its accuracy can be improved in any way. This is done by tuning the parameters present in your model. Parameters are the variables in the model that the programmer generally decides. At a particular value of your parameter, the accuracy will be the maximum. Parameter tuning refers to finding these values.

8. Making Predictions

   In the end, you can use your model on unseen data to make predictions accurately.

3 Project Analysis and Design

3.1 Methodologies Adapted

Data Collection

Size:
Train.csv: 65.6Mb
Test.csv: 57.6Mb
Testlabels.csv: 4.7Mb
Sample$_submission.csv$: $6Mb$

Rows and Columns:

Train.csv:
8 Columns: id, columntext, toxic, severetoxic, obscene, threat, insult, identity-hate
159572 Rows.

Test.csv:
2 Columns: id, comment$_text$
153165$Rows$

Test$_labels.csv$:
$7Columns : id, toxic, severe_toxic, obscene, threat, insult, identity_hate$
153165$Rows$

Sample$_submission.csv$:
$7Columns : id, toxic, severe_toxic, obscene, threat, insult, identity_hate$
153165$Rows$

- Data Preprocessing
  Data Preprocessing includes processing the data, finding null values, reducing the rows with null values, cleaning the data.

- Data Visualization
  Data Visualization means visualizaing the data using charts, graphs, histograms or maps. We have used two histograms and one heatmap to visualize the data.

- Splitting dataset into Training and Testing Data
  The provided dataset as already provided two separate datasets for Training and Testing the data by Kaggle.com.

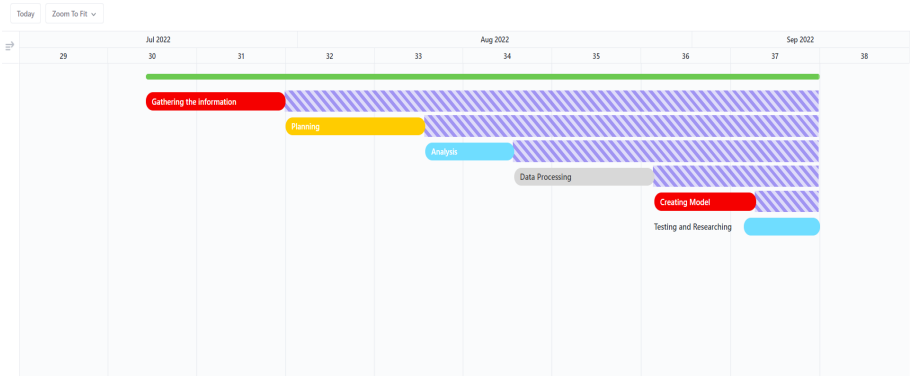
- Training the data with Binary Relevance using Logistic Regression
  We feed our training and testing data to the model to learn the data and provide the accuracy of the model for each category of the comment.

- Training the data with Classifier Chains using Logistic Regression
  Similar to the previous step we feed the training and testing data to the model to learn the data and it gives us the accuracy of our model for each category.

- Exporting the Result
  After the training model we provide a sample dataset to both the models and it gives us the accuracy for each category. We store the results separatly for both models Binary Relevance and Classifier Chains in .csv format.

3.2 Modules

3.2.1 Gantt Chart



3.2.1: Gantt Chart

4 Project Implementation and Testing

4.1 Importing Libraries

Importing Dependencies

```
[ ]  1 #importing numpy for arrays
     2 import numpy as np
     3
     4 #importing pandas for manipulation of the data
     5 import pandas as pd
     6
     7 #importing matplotlib for using plots
     8 from matplotlib import pyplot as plt
     9
    10 #importing seaborn for using plots
    11 import seaborn as sns
    12
    13 #importing re for regular expressions
    14 import re
```

4.1.1: Importing Libraries

4.2 Data Collection

Data Collection and Pre-Processing

```
[ ]  1 from google.colab import drive
     2 drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[ ]  1 #importing the datasets
     2 train_df = pd.read_csv('/content/drive/MyDrive/Summer-Project/train.csv')
     3 test_df = pd.read_csv('/content/drive/MyDrive/Summer-Project/test.csv')
     4 print(train_df)
```

```
                     id                                       comment_text  \
0       0000997932d777bf  Explanation\nWhy the edits made under my usern...
1       000103f0d9cfb60f  D'aww! He matches this background colour I'm s...
2       000113f07ec002fd  Hey man, I'm really not trying to edit war. It...
3       0001b41b1c6bb37e  "\nMore\nI can't make any real suggestions on ...
4       0001d958c54c6e35  You, sir, are my hero. Any chance you remember...
...                  ...                                                ...
159566  ffe987279560d7ff  ":::::And for the second time of asking, when ...
159567  ffea4adeee384e90  You should be ashamed of yourself \n\nThat is ...
159568  ffee36eab5c267c9  Spitzer \n\nUmm, theres no actual article for ...
159569  fff125370e4aaaf3  And it looks like it was actually you who put ...
159570  fff46fc426af1f9a  "\nAnd ... I really don't think you understand...

        toxic  severe_toxic  obscene  threat  insult  identity_hate
0           0             0        0       0       0              0
1           0             0        0       0       0              0
2           0             0        0       0       0              0
3           0             0        0       0       0              0
4           0             0        0       0       0              0
...       ...           ...      ...     ...     ...            ...
159566      0             0        0       0       0              0
159567      0             0        0       0       0              0
159568      0             0        0       0       0              0
159569      0             0        0       0       0              0
159570      0             0        0       0       0              0
```

4.2.1: Data Collection

4.3 Data Preprocessing 1

```
[ ]  1 train_df.comment_text[15]

'"\n\nJuelz Santanas Age\n\nIn 2002, Juelz Santana was 18 years old, then came February 18th, which makes Juelz turn 19 making songs with The Diplomats. The third neff to be signed to
Cam\'s label under Roc A Fella. In 2003, he was 20 years old coming out with his own singles ""Santana\'s Town"" and ""Down"". So yes, he is born in 1983. He really is, how could he b
e older then Lloyd Banks? And how could he be 22 when his birthday passed? The homie neff is 23 years old. 1983 - 2006 (Juelz death, god forbid if your thinking about that) equals 23.
Go to your caculator and stop changing his year of birth. My god."'

[ ]  1 #printing 5 random tuples
     2 train_df.sample(5)
```

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
158613	f0ab4dd74fc15809	"\n\n VfD on Jews by country \n\nThis has not ...	0	0	0	0	0	0
42830	7244097e417e6e27	Douglas Maine attack article = vandalism \n\nP...	0	0	0	0	0	0
131254	be4cd537a75e5fbb	ami killed sailor moon \n\n.	1	0	0	0	0	0
25009	422146e60fcc1bf7	"\n\nI can give you the IP because it was when...	0	0	0	0	0	0
138146	e33ea5cff4eb2e09	"\n\n Hindus Love the Word Controversy\n\I am s...	0	0	0	0	0	0

```
[ ]  1 #declaring target columns
     2 cols_target = ['obscene','insult','toxic','severe_toxic','identity_hate','threat']

[ ]  1 train_df.describe()
```

	toxic	severe_toxic	obscene	threat	insult	identity_hate
count	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000
mean	0.095844	0.009996	0.052948	0.002996	0.049364	0.008805
std	0.294379	0.099477	0.223931	0.054650	0.216627	0.093420
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

4.3.1: Data Preprocessing 1

4.4 Data Preprocessing 2

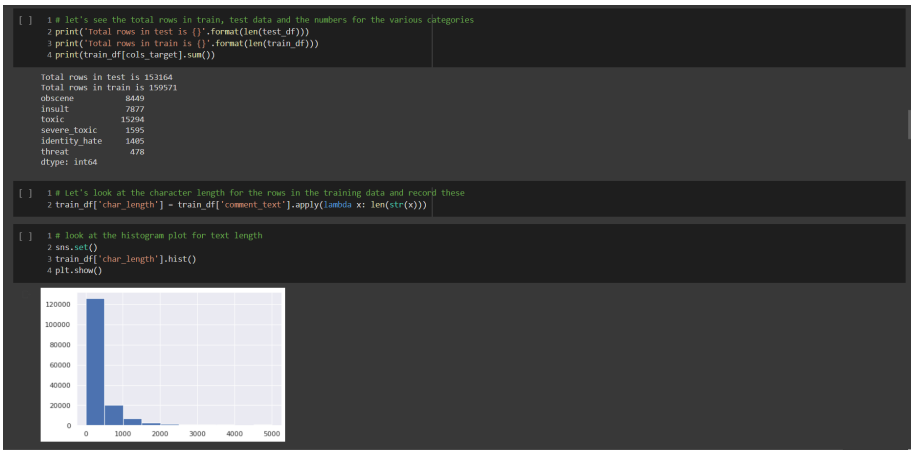
```
[ ]  1 #checking for the unlabeled data
     2 unlabelled_in_all = train_df[(train_df['toxic']!=1) & (train_df['severe_toxic']!=1) & (train_df['obscene']!=1) & (train_df['threat']!=1) & (train_df['insult']!=1) & (train_df['ider
     3 print('Percentage of unlabeled comments is ', len(unlabelled_in_all)/len(train_df)*100)

Percentage of unlabeled comments is  89.83211235124176

[ ]  1 # check for any 'null' comment
     2 no_comment = train_df[train_df['comment_text'].isnull()]
     3 len(no_comment)

0

[ ]  1 #print first 5 tuples of dataset
     2 test_df.head()
```

	id	comment_text
0	00001cee341fdb12	Yo bitch Ja Rule is more succesful then you'll...
1	0000247867823ef7	== From RfC == \n\n The title is fine as it is...
2	00013b17ad220c46	" \n\n == Sources == \n\n * Zawe Ashton on Lap...
3	00017563c3f7919a	:If you have a look back at the source, the in...
4	00017695ad8997eb	I don't anonymously edit articles at all.

```
[ ]  1 print(no_comment)

Empty DataFrame
Columns: [id, comment_text, toxic, severe_toxic, obscene, threat, insult, identity_hate]
Index: []

[ ]  1 # let's see the total rows in train, test data and the numbers for the various categories
     2 print('Total rows in test is {}'.format(len(test_df)))
     3 print('Total rows in train is {}'.format(len(train_df)))
```

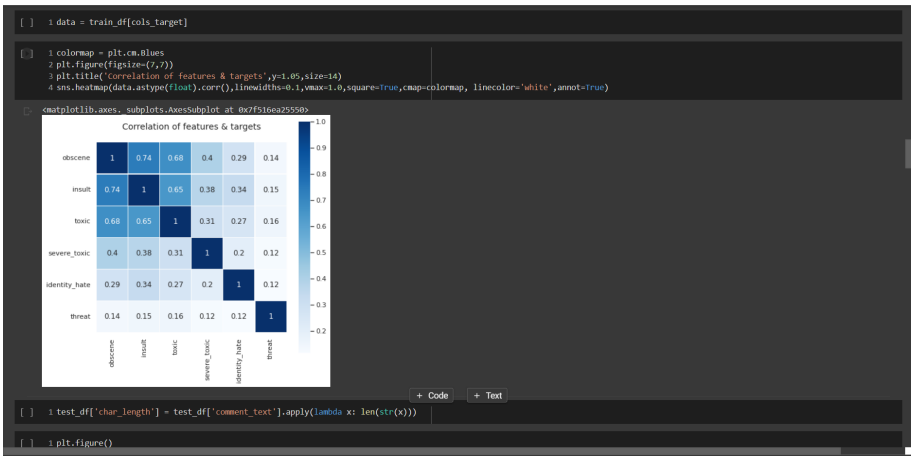
4.4.1: Data Preprocessing 2

4.5 Data Visualization 1



4.5.1: Data Visualization 1

4.6 Data Visualization 2

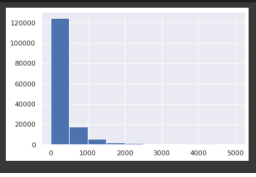


4.6.1: Data Visualization 2

4.7 Data Visualization 3

```
[ ]  1 test_df['char_length'] = test_df['comment_text'].apply(lambda x: len(str(x)))
```

```
[ ]  1 plt.figure()
     2 plt.hist(test_df['char_length'])
     3 plt.show()
```



4.7.1: Data Visualization 3

4.8 Data Cleaning 1

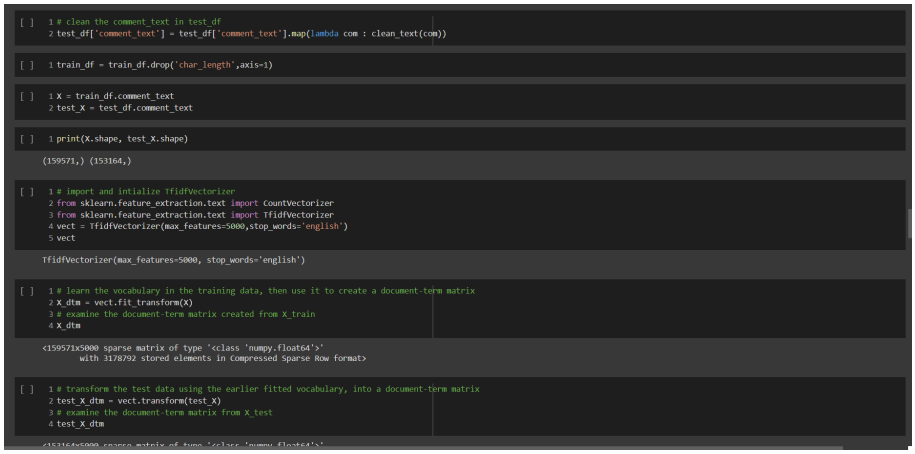
```
[ ]  1 # defining the clean text using regex
     2 def clean_text(text):
     3     text = text.lower()
     4     text = re.sub(r"what's", "what is ", text)
     5     text = re.sub(r"\'s", " ", text)
     6     text = re.sub(r"\'ve", " have ", text)
     7     text = re.sub(r"can't", "cannot ", text)
     8     text = re.sub(r"n't", " not ", text)
     9     text = re.sub(r"i'm", "i am ", text)
    10     text = re.sub(r"\'re", " are ", text)
    11     text = re.sub(r"\'d", " would ", text)
    12     text = re.sub(r"\'ll", " will ", text)
    13     text = re.sub(r"\'scuse", " excuse ", text)
    14     text = re.sub('\W', ' ', text)
    15     text = re.sub('\s+', ' ', text)
    16     text = text.strip(' ')
    17     return text
```

```
[ ]  1 # clean the comment_text in train_df
     2 train_df['comment_text'] = train_df['comment_text'].map(lambda com : clean_text(com))
     3 train_df.describe()
```

	toxic	severe_toxic	obscene	threat	insult	identity_hate	char_length
count	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000	159571.000000
mean	0.095844	0.009996	0.052948	0.002996	0.049364	0.008805	394.073221
std	0.294379	0.099477	0.223931	0.054650	0.216627	0.093420	590.720282
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	6.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	96.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	205.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	435.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	5000.000000

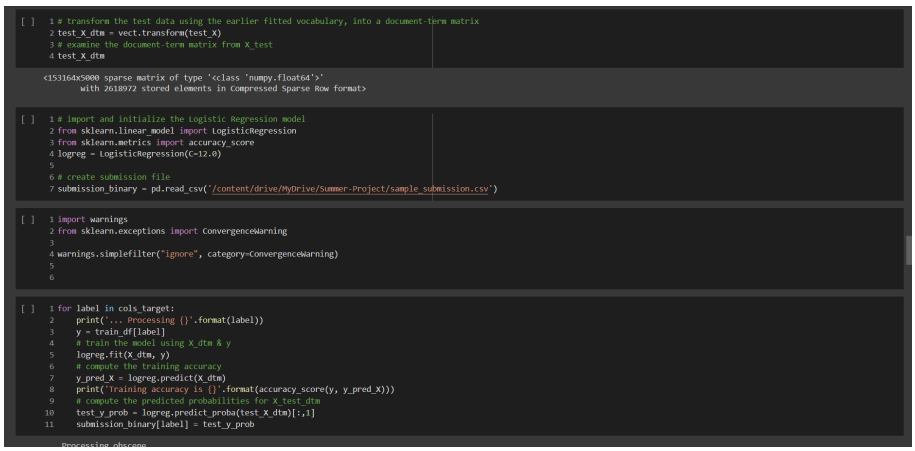
4.8.1: Data Cleaning 1

4.9 Importing TfidVectorizer



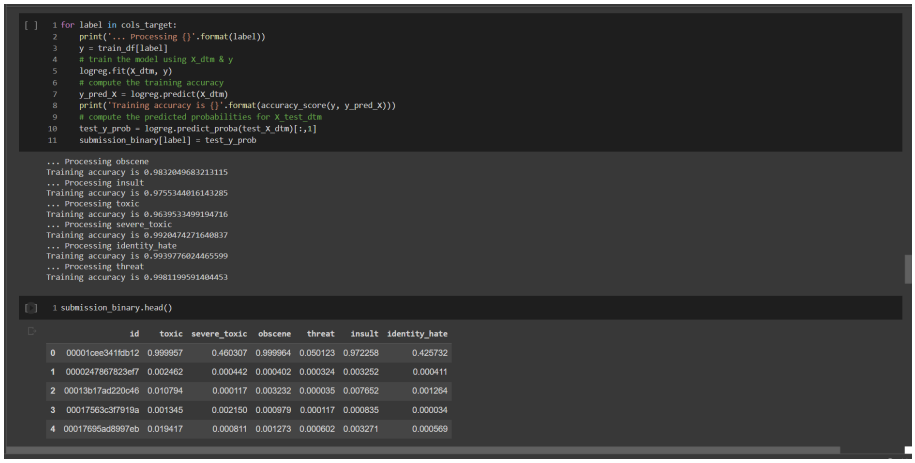
4.9.1: Importing TfidVectorizer

4.10 Importing Logistic Regression



4.10.1: Importing Logistic Regression

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

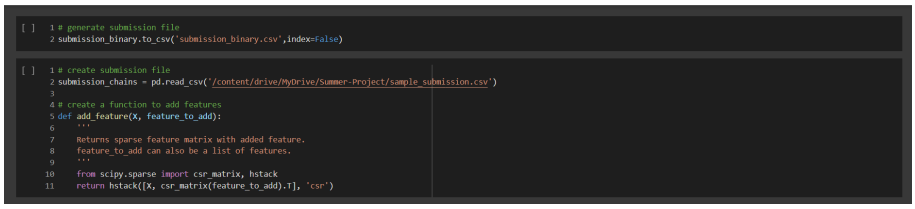


```
[ ] 1 for label in cols_target:
    2     print('... Processing {}'.format(label))
    3     y = train_df[label]
    4     # train the model using X_dtm & y
    5     logreg.fit(X_dtm, y)
    6     # compute the training accuracy
    7     y_pred_X = logreg.predict(X_dtm)
    8     print('Training accuracy is {}'.format(accuracy_score(y, y_pred_X)))
    9     # compute the predicted probabilities for X_test_dtm
   10     test_y_prob = logreg.predict_proba(test_X_dtm)[:,1]
   11     submission_binary[label] = test_y_prob

... Processing obscene
Training accuracy is 0.9832049683213115
... Processing insult
Training accuracy is 0.9755344016143285
... Processing toxic
Training accuracy is 0.9639533499194716
... Processing severe_toxic
Training accuracy is 0.9920474271640837
... Processing identity_hate
Training accuracy is 0.9939776024465599
... Processing threat
Training accuracy is 0.9981199591404453

[ ] 1 submission_binary.head()
```

	id	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	0.999957	0.460307	0.999964	0.050123	0.972258	0.425732
1	0000247867823ef7	0.002462	0.000442	0.000402	0.000324	0.003252	0.000411
2	00013b17ad220c46	0.010794	0.000117	0.003232	0.000035	0.007652	0.001264
3	00017563c3f7919a	0.001345	0.002150	0.000979	0.000117	0.000835	0.000034
4	00017695ad8997eb	0.019417	0.000811	0.001273	0.000602	0.003271	0.000569

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

4.12 Classifier Chains - build a multi-label classifier using Logistic Regression



```
[ ] 1 # generate submission file
    2 submission_binary.to_csv('submission_binary.csv',index=False)

[ ] 1 # create submission file
    2 submission_chains = pd.read_csv('/content/drive/MyDrive/Summer-Project/sample_submission.csv')
    3
    4 # create a function to add features
    5 def add_feature(X, feature_to_add):
    6     '''
    7     Returns sparse feature matrix with added feature.
    8     feature_to_add can also be a list of features.
    9     '''
   10     from scipy.sparse import csr_matrix, hstack
   11     return hstack([X, csr_matrix(feature_to_add).T], 'csr')
```

4.12.1: Classifier Chains - build a multi-label classifier using Logistic Regression

4.13 Classifier Chains 2

```
[ ]  1 for label in cols_target:
     2     print('... Processing {}'.format(label))
     3     y = train_df[label]
     4     # train the model using X_dtm & y
     5     logreg.fit(X_dtm,y)
     6     # compute the training accuracy
     7     y_pred_X = logreg.predict(X_dtm)
     8     print('Training Accuracy is {}'.format(accuracy_score(y,y_pred_X)))
     9     # make predictions from test_X
    10     test_y = logreg.predict(test_X_dtm)
    11     test_y_prob = logreg.predict_proba(test_X_dtm)[:,1]
    12     submission_chains[label] = test_y_prob
    13     # chain current label to X_dtm
    14     X_dtm = add_feature(X_dtm, y)
    15     print('Shape of X_dtm is now {}'.format(X_dtm.shape))
    16     # chain current label predictions to test_X_dtm
    17     test_X_dtm = add_feature(test_X_dtm, test_y)
    18     print('Shape of test_X_dtm is now {}'.format(test_X_dtm.shape))

... Processing obscene
Training Accuracy is 0.9832049683213115
Shape of X_dtm is now (159571, 5001)
Shape of test_X_dtm is now (153164, 5001)
... Processing insult
Training Accuracy is 0.9818137380852411
Shape of X_dtm is now (159571, 5002)
Shape of test_X_dtm is now (153164, 5002)
... Processing toxic
Training Accuracy is 0.9675630283698166
Shape of X_dtm is now (159571, 5003)
Shape of test_X_dtm is now (153164, 5003)
... Processing severe_toxic
Training Accuracy is 0.9930626492282433
Shape of X_dtm is now (159571, 5004)
Shape of test_X_dtm is now (153164, 5004)
... Processing identity_hate
Training Accuracy is 0.9956195047972376
Shape of X_dtm is now (159571, 5005)
Shape of test_X_dtm is now (153164, 5005)
```

4.13.1: Classifier Chains 2

4.14 Exporting Data 1

```
[ ]  1 submission_chains.head()
```

	id	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	0.999970	4.168872e-01	0.999964	0.193270	0.893687	0.600495
1	0000247867823ef7	0.002696	7.459653e-10	0.000402	0.000072	0.003677	0.000136
2	00013b17ad220c46	0.008088	2.060120e-10	0.003232	0.000009	0.004084	0.000536
3	00017563c3f7919a	0.001138	1.124507e-08	0.000979	0.000063	0.000569	0.000004
4	00017695ad8997eb	0.019776	1.095576e-09	0.001273	0.000158	0.001547	0.000104

```
[ ]  1 # generate submission file
     2 submission_chains.to_csv('submission_chains.csv', index=False)

[ ]  1 # create submission file
     2 submission_combined = pd.read_csv('/content/drive/MyDrive/Summer-Project/submission_combined.csv')

[ ]  1 # corr_targets = ['obscene','insult','toxic']
     2 for label in cols_target:
     3     submission_combined[label] = 0.5*(submission_chains[label]+submission_binary[label])

[ ]  1 submission_combined.head()
```

	id	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	0.999964	0.438597	0.999964	0.121697	0.932973	0.513114
1	0000247867823ef7	0.002579	0.000221	0.000402	0.000198	0.003465	0.000274
2	00013b17ad220c46	0.009441	0.000059	0.003232	0.000022	0.005868	0.000900
3	00017563c3f7919a	0.001241	0.001075	0.000979	0.000090	0.000702	0.000019
4	00017695ad8997eb	0.019597	0.000405	0.001273	0.000380	0.002409	0.000336

4.14.1: Exporting Data 1

4.15 Exporting Data 2

```
[ ]  1 submission_combined.head()
```

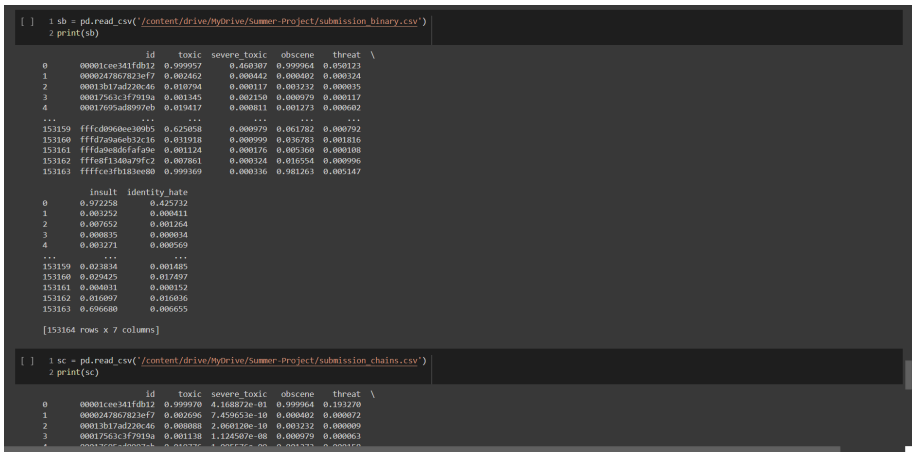
	id	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	0.999964	0.438597	0.999964	0.121697	0.932973	0.513114
1	000024786782 3ef7	0.002579	0.000221	0.000402	0.000198	0.003465	0.000274
2	00013b17ad220c46	0.009441	0.000059	0.003232	0.000022	0.005868	0.000900
3	00017563c3f7919a	0.001241	0.001075	0.000979	0.000090	0.000702	0.000019
4	00017695ad8997eb	0.019597	0.000405	0.001273	0.000380	0.002409	0.000336

```
[ ]  1 # generate submission file
     2 submission_combined.to_csv('submission_combined.csv', index=False)
```

4.15.1: Exporting Data 2

5 Test Cases

5.1 Exported csv for Binary Relevance



```
[ ]  1 sb = pd.read_csv("/content/drive/MyDrive/Summer-Project/submission_binary.csv")
     2 print(sb)
```

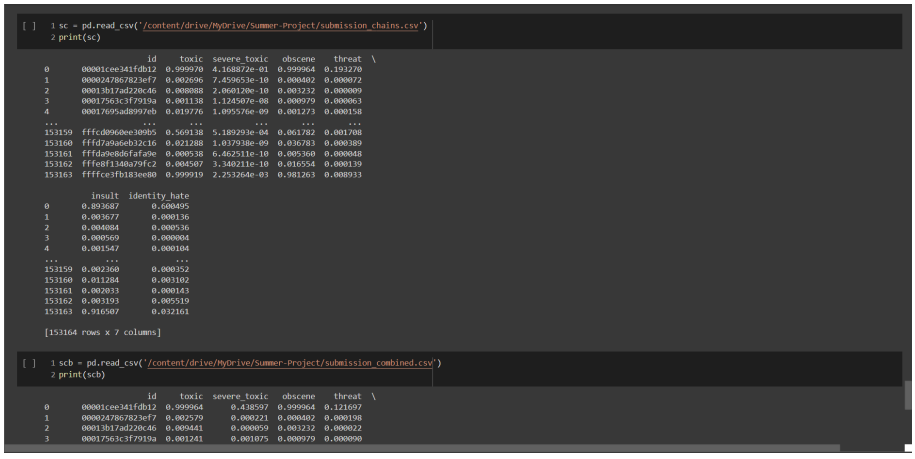
```
                     id     toxic  severe_toxic   obscene    threat  \
0       00001cee341fdb12  0.999957      0.460387  0.999964  0.050123
1       0000247867823ef7  0.002462      0.000442  0.000402  0.000324
2       00013b17ad220c46  0.010794      0.000117  0.003232  0.000035
3       00017563c3f7919a  0.001345      0.002150  0.000979  0.000117
4       00017695ad8997eb  0.019417      0.000811  0.001273  0.000602
...                  ...       ...           ...       ...       ...
153159  fffcd0960ee309b5  0.625058      0.000979  0.061782  0.000792
153160  fffd7a9a6eb32c16  0.031918      0.000999  0.036783  0.001816
153161  fffda9e8d6fafa9e  0.001124      0.000176  0.005360  0.000108
153162  fffe8f1340a79fc2  0.007861      0.000324  0.016554  0.000996
153163  ffffce3fb183ee80  0.999369      0.000336  0.981263  0.005147

          insult  identity_hate
0       0.972258       0.425732
1       0.003252       0.000411
2       0.007652       0.001264
3       0.000835       0.000034
4       0.003271       0.000569
...          ...            ...
153159  0.023834       0.001485
153160  0.029425       0.017497
153161  0.004031       0.000152
153162  0.016097       0.016036
153163  0.696680       0.006655

[153164 rows x 7 columns]
```

```
[ ]  1 sc = pd.read_csv("/content/drive/MyDrive/Summer-Project/submission_chains.csv")
     2 print(sc)
```

```
                     id     toxic  severe_toxic   obscene    threat  \
0       00001cee341fdb12  0.999970  4.168872e-01  0.999964  0.193270
1       0000247867823ef7  0.002696  7.459653e-10  0.000402  0.000072
2       00013b17ad220c46  0.008088  2.060120e-10  0.003232  0.000009
3       00017563c3f7919a  0.001138  1.124507e-08  0.000979  0.000063
```

5.1.1: Exported csv for Binary Relevance

5.2 Exported csv for Classifier Chains



```
[ ]  1 sc = pd.read_csv('/content/drive/MyDrive/Summer-Project/submission_chains.csv')
     2 print(sc)
```

```
                     id     toxic  severe_toxic   obscene    threat  \
0       00001cee341fdb12  0.999970  4.168872e-01  0.999964  0.193270
1       0000247867823ef7  0.002696  7.459653e-10  0.000402  0.000072
2       00013b17ad220c46  0.008088  2.060120e-10  0.003232  0.000009
3       00017563c3f7919a  0.001138  1.124507e-08  0.000979  0.000063
4       00017695ad8997eb  0.019776  1.095576e-09  0.001273  0.000158
...                  ...       ...           ...       ...       ...
153159  fffcd0960ee309b5  0.569138  5.189293e-04  0.061782  0.001708
153160  fffd7a9a6eb32c16  0.021288  1.037938e-09  0.036783  0.000389
153161  fffda9e8d6fafa9e  0.000538  6.462511e-10  0.005360  0.000048
153162  fffe8f1340a79fc2  0.004507  3.340211e-10  0.016554  0.000139
153163  ffffce3fb183ee80  0.999919  2.253264e-03  0.981263  0.008933

          insult  identity_hate
0       0.893687       0.600495
1       0.003677       0.000136
2       0.004084       0.000536
3       0.000569       0.000004
4       0.001547       0.000104
...          ...            ...
153159  0.002360       0.000352
153160  0.011284       0.003102
153161  0.002033       0.000143
153162  0.003193       0.005519
153163  0.916507       0.032161

[153164 rows x 7 columns]
```

```
[ ]  1 scb = pd.read_csv('/content/drive/MyDrive/Summer-Project/submission_combined.csv')
     2 print(scb)
```

```
                     id     toxic  severe_toxic   obscene    threat  \
0       00001cee341fdb12  0.999964      0.438597  0.999964  0.121697
1       0000247867823ef7  0.002579      0.000221  0.000402  0.000198
2       00013b17ad220c46  0.009441      0.000059  0.003232  0.000022
3       00017563c3f7919a  0.001241      0.001075  0.000979  0.000090
```

5.2.1: Exported csv for Classifier Chains

5.3 Exported csv for combined Binary Relevance and Classifier Chains

```
1 scb = pd.read_csv('/content/drive/MyDrive/Summer-Project/submission_combined.csv')
2 print(scb)
```

```
                    id     toxic  severe_toxic   obscene    threat  \
0       00001cee341fdb12  0.999964      0.438597  0.999964  0.121697
1       0000247867823ef7  0.002579      0.000221  0.000402  0.000198
2       0001b17ad220c46  0.009441      0.000059  0.003232  0.000022
3       00017563c3f7919a  0.001241      0.001075  0.000979  0.000090
4       00017695ad8997eb  0.019597      0.000405  0.001273  0.000380
...                  ...       ...           ...       ...       ...
153159  fffcd0960ee309b5  0.597098      0.000749  0.061782  0.001250
153160  fffd7a9a6eb32c16  0.026603      0.000499  0.036783  0.001102
153161  fffda9e8d6fafa9e  0.000831      0.000088  0.005360  0.000078
153162  fffe8f1340a79fc2  0.006184      0.000162  0.016554  0.000567
153163  ffffce3fb183ee80  0.999644      0.001295  0.981263  0.007040

           insult  identity_hate
0        0.932973       0.513114
1        0.003465       0.000274
2        0.005868       0.000900
3        0.000702       0.000019
4        0.002409       0.000136
...           ...            ...
153159   0.013097       0.000918
153160   0.020354       0.010300
153161   0.003032       0.000147
153162   0.009645       0.010778
153163   0.806594       0.019408

[153164 rows x 7 columns]
```

5.3.1: Exported csv for combined Binary Relevance and Classifier Chains

6 Limitations

- It needs internet to be accessed.
- It is only used for comments and review classification
- The data needs to be provided in .csv format
- It doesn't have any user interface as it is a model that can be implemented by social media platforms.

7 Future Enhancements

- A model can be used in social media sites and apps.
- It can be also be used in E-Commerce websites for filtering toxic reviews.
- The model can be also trained with other algorithms and model and compare them for better performance.

8 Bibliography

8.1 Web References

[1.] Research Paper 1 by Pallam Ravi, Hari Narayana Batta, Greeshma S, Shaik Yaseen (International Journal of Trend in Scientific Research and Development)
https://bit.ly/3CGBWKS

[2.] Research Paper 2 by Grigorios Tsoumakas, Ioannis Katakis (Dept. of Informatics, Aristotle University of Thessaloniki, 54124, Greece)
https://bit.ly/3CZUkjo

[3.] Research Paper 3 by Department of Computer Engineering Army Institute of Technology Pune, India
https://bit.ly/3ENzzsq

[4.] Research Paper 4 by NMIMS's Mukesh Patel School of Technology Management Engineering
https://bit.ly/3ERpR8B

[5.] Research Paper 5 by Department of Information Technology, Sreenidhi Institute of Science and Technology, India
https://bit.ly/3EQ9jxx

[6.] Hands-on Machine Learning Course - Youtube Channel (Siddhardhan)
https://www.youtube.com/c/Siddhardhan/playlists?view=50&sort=dd&shelf_id=4