NAME OF THE PROJECT

Malignant Comment Classifier Project

SUBMITTED BY

MANISH SINGH

INTRODUCTION

The dataset has approximately 1,59,000 samples. All the data samples contain 8 fields which includes ‘Id’, ‘Comments’, ‘Malignant’, ‘Highly malignant’, ‘Rude’, ‘Threat’, ‘Abuse’ and ‘Loathe’.

1.Malignant: It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.

2.Highly Malignant: It denotes comments that are highly malignant and hurtful.
3.Rude: It denotes comments that are very rude and offensive.

4.Threat: It contains indication of the comments that are giving any threat to someone.
5. Abuse: It is for comments that are abusive in nature.
6.Loathe: It describes the comments which are hateful and loathing in nature.
7. ID: It includes unique Ids associated with each comment text given.
8.Comment text: This column contains the comments extracted from various social media platforms.

Business Problem Framing

The proliferation of social media enables people to express their opinions widely online. However, at the same time, this has resulted in the emergence of conflict and hate, making online environments uninviting for users. Although researchers have found that hate is a problem across multiple platforms, there is a lack of models for online hate detection

Our goal is to build a prototype of online hate and abuse comment classifier which can used to classify hate and offensive comments so that it can be controlled and restricted from spreading hatred and cyberbullying.

Review of Literature

Important point is, making a model which can sort out very accurately malignant, rude, threat or loathe types of comments is difficult task because thousands of languages have in this world and all language have diverse types of negative or malignant words. Those all words keep in while making a model is really difficult task

ANALYTICAL PROBLEM FRAMING

Mathematical/ Analytical Modeling of the Problem

. In Train Dataset malignant types of comments= 10 percent, and non-malignant types of comments= 90 percent

. when we checked distributed of comment's length then we saw that most of higher length of comments are related to malignant types of comments

Data Sources and their formats

The format of this dataset is csv

We below showed a snapshot of dataset which given description of dataset

```
#
train.sample(6)
```

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
75011	c8b202090a15f389	"\n\n ""Encyclopedic value is negligible"" \n\...	0	0	0	0	0	0
115175	67dd79916ed5197e	it is highly unlikely that the police would re...	0	0	0	0	0	0
67320	b4203795df42ccbf	Working on it. I can't just change the expansi...	0	0	0	0	0	0
155203	ba6071c3a9893786	Dear Jeff G. ツ, BLOW ME FAG!	1	0	1	0	1	0
16941	2cb1fe3d153f59b5	MfD nomination of User:Rahulgul\n. Your opinio...	0	0	0	0	0	0
81119	d8fa96d141e4be56	Is somebody here actually trying to say that I...	0	0	0	0	0	0

```
# LET'S CHECK DATASET SHAPE
train.shape
```

```
(159571, 8)
```

```
# NO NULL VALUE IN THIS DATASET
train.isnull().sum()
```

```
id                  0
comment_text        0
malignant           0
highly_malignant    0
rude                0
threat              0
abuse               0
loathe              0
dtype: int64
```

Data Preprocessing Done

First, we created a 'target' feature for keeping all negative meanings comments in one feature because Different types of negative nature comments were distributed in six columns, to do them in one column had to be made

In Data-preprocessing main point was to clean the 'comment_text' data and then stemming with Lemmatization, after that Vectorized the clean text data

. we used regex to clean unnecessary alphabets and numerical character from comment_text and then removed stop-word by using stopwords function of NLP

. Stemming the text data by using Lemmatization

. we remove some blank character rows and after drop some duplicates rows

. we created to 'clean_length' feature which show to how many words in clean_text comment

. make visualization on clean_text data like distribution plot, WordCloud,

. And last, we done Vectorized the clean_text data

Software Requirements and Tools Used

Making this NLP project by using Pandas, NLTK, matplotlib, WordCloud, regex and sklearn.

Pandas is important library for making this project and from using Sklearn we built a ML model on text data

We did clean the text by using regex, NLTK and visualized the data through WordCloud

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

First, we cleaned the text data and after that make visualize to understand the data like to check which type of word is more in non-malignant or malignant comments, WordCloud is plotted

. We vectorized the data by using Tf-idf

In this dataset

. True-Positive = comments are 'Non-Malignant' and model also predict to 'Non-Malignant'

. True-Negative= comments are 'Malignant' and model also predict to 'Malignant'

. False-Positive= comments are 'Non-Malignant' and model predict to 'Malignant'

. False-Negative= comments are 'Malignant' and model predict to 'Non-Malignant'

Testing of Identified Approaches (Algorithms)

. In this Text classification project, we used MultinomialNB, BernauliNB and RandomForestClassifier for making the models

. on this dataset we have to build such types of models whose False-Negative will be minimum

. BernauliNB model performed not so well on this dataset because its FN and FP both are higher compare to another model

. False-Negative of MultinomialNB model is so high compare to another and this important thing is to minimze the FN (comments are 'Malignant' and model predict to 'Non-Malignant')

Key Metrics for success in solving problem under consideration

. In Malignant-comments- classification project, Important point is minimize to False-Negative and we evaluate the model by using confusion_matrix

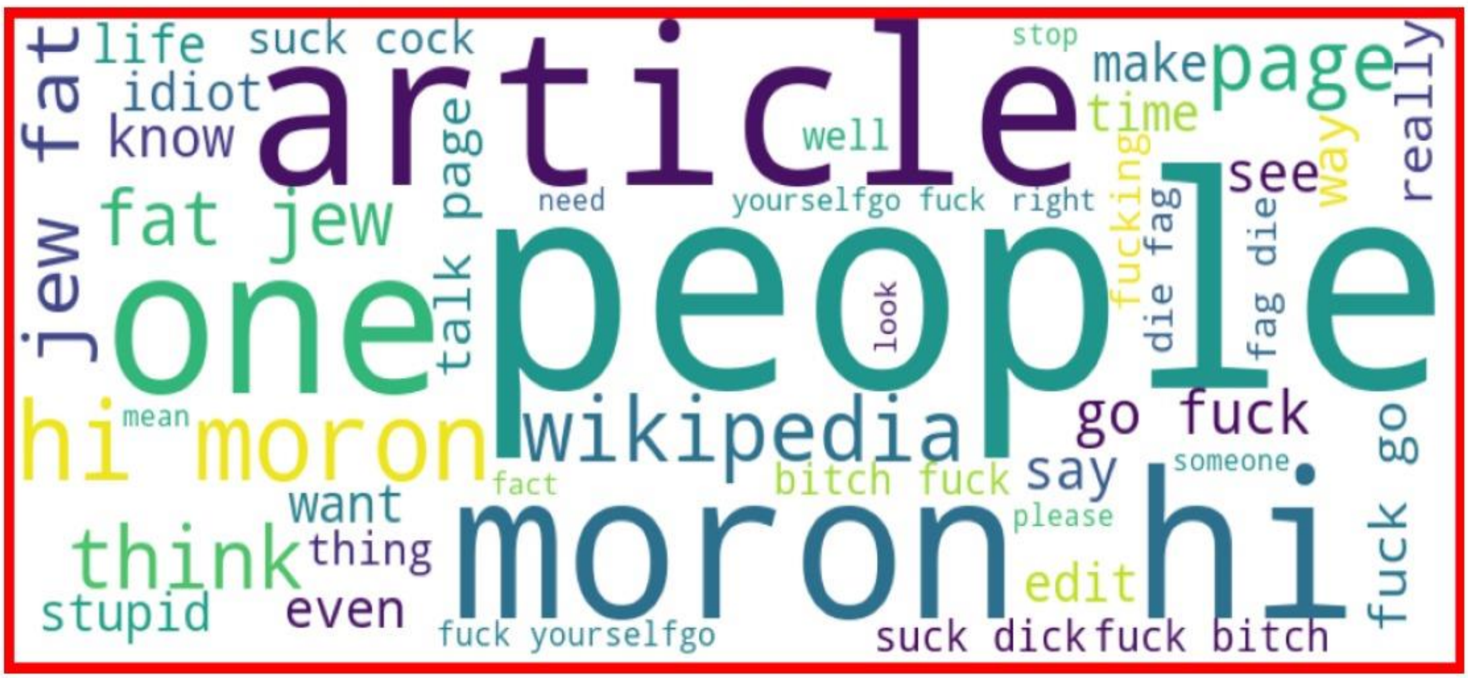
. we used accuracy_score, classification report and confusion_matrix for evaluating the model

Visualizations

We mentioning below some plot related to this project

. Plotting this for getting sense of top cloud words of 'malignant' comment

```
# we can see in this wordcloud picture , maximum words are showing negative meaning
```



. Plotting this for getting sense of top n words of 'non-malignant' comment

```
# we can see in this picture that all words are positive meaning
```

Interpretation of the Results

This is a brief description about the information obtained from Visualization, Preprocessing and Modelling on a Dataset.

. From Wordcloud Visualization, we get the information about specific top n words that which are the words that come more in 'malignant' and 'non-malignant' comments

. In preprocessing part, we done clean the data,removed stopwords and stemming the data , So that the data can be kept as an input in the model

CONCLUSION

Key Findings and Conclusions of the Study

. My observation is that the FP rate of the models is coming higher compare to FP rate because the value distributed ratio of the Target feature is 90:10,

non-malignant=90%   and malignant type= 10%

. And I wanted the model's FN rate to be minimum

. By making a model on this dataset it was found that there are some words that are only in Malignant comments, if we make models with more and more such words, then it will be easier for the model to predict negative comments

Learning Outcomes of the Study in respect of Data Science

. In Visualization part, we did visualize the WordCloud to get the information about specific top n words that which are the words that come more in 'malignant' and 'non-malignant' comments

. text cleaning is more important part of this project, we did know from text-cleaning that 35 percent words of whole data are unnecessary

. If the target feature was balanced, then it is more likely that the result would have been higher than the imbalanced target

. This is a NLP project whose purpose is to identify negative or malignant type and positive types of comments, if we train the model on as many malignant types comments or message as possible while making the model, then there is a more chance that the malignant type of comments can be sorted out