Malignant Comments Classification

Submitted by:

VARSHA PATANKAR

ACKNOWLEDGMENT

Various references were used for the implementation of this project. Different research papers by experts were referred for the best method and for the best result. Different sites also referred for this project which are useful for data science projects like github, kaggle, towardsdatascience and so on.

INTRODUCTION

- ## 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.

  Online hate, described as abusive language, aggression, cyberbullying, hatefulness and many others has been identified as a major threat on online social media platforms. Social media platforms are the most prominent grounds for such toxic behaviour.

  There has been a remarkable increase in the cases of cyberbullying and trolls on various social media platforms. Many celebrities and influences are facing backlashes from people and have to come across hateful and offensive comments. This can take a toll on anyone and affect them mentally leading to depression, mental illness, self-hatred and suicidal thoughts.

  Internet comments are bastions of hatred and vitriol. While online anonymity has provided a new outlet for aggression and hate speech, machine learning can be used to fight it. The problem we sought to solve was the tagging of internet comments that are aggressive towards other users. This means that insults to third parties such as celebrities will be tagged as unoffensive, but “u are an idiot” is clearly offensive.

- ## Conceptual Background of the Domain Problem

  To protect users from being exposed to offensive language on online forums or social media sites, companies have started flagging comments and blocking users who are found guilty of using

unpleasant language. Several Machine Learning models have been developed and deployed to filter out the unruly language and protect internet users from becoming victims of online harassment and cyberbullying.

- Review of Literature

  Having worked on an NLP use-case before , the aim in this project was to focus on data pre-processing and feature engineering and ensure that the data, which will be consumed by machine learning models, is as clean as possible.

- Motivation for the Problem Undertaken

  There has been a remarkable increase in the cases of cyberbullying and trolls on various social media platforms. Many celebrities and influences are facing backlashes from people and have to come across hateful and offensive comments. This can take a toll on anyone and affect them mentally leading to depression, mental illness, self-hatred and suicidal thoughts.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

  The general problem one faces in the modelling is that to find best suitable algorithm and mathematical and statistical approach for our model. In real world the datasets are very large, noisy and unstructured. So to find the best strategy for our model we need such approach that combines the statistics and machine learning together.

  Statisticians are using so many approaches to find the best accuracy and we are also trying some of the machine learning algorithms along with the statistics. So while using those methods we will came to know that which algorithm has limitations and which algorithm is best fitted for our model.

- Data Sources and their formats

  There are so many sources are available on the internet from which can get the data on which we have to work on. So for this project you can get the data from github site, kaggle site, towardsdatascience etc.

  The data you found can be in any form i.e. csv file, excel file etc.

  Here I have got the data in the form of csv file both train and test data sperately which we have to import that file using pandas library. Below is the snapshot of data and how I have load the data for further process.

```
#lets import pandas library to read the CSV data
import pandas as pd
df = pd.read_csv('train.csv')
df
```

	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0

- Data Pre-processing Done

  There is some process that we have to do while pre-processing. Following are the process that we have used-

1. **Data Wrangling:** Our dataset is not always in proper format. Sometimes there maybe some missing values, sometimes there must be out of the box values. Our data is always a raw data so for that we have to do some wrangling process like.
   a. Check if there are any null values are present in the dataset or not.
   b. Check the data type of each column.

```
import seaborn as sns             #visualization library to see null values graphically
import matplotlib.pyplot as plt   #another visualization library to plot the output
pd.options.display.max_info_columns = 9   #check the columns info if any null values present
df.info()                                 #displays the info
print(sns.heatmap(df.isnull()))
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159571 entries, 0 to 159570
Data columns (total 8 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   id                159571 non-null  object
 1   comment_text      159571 non-null  object
 2   malignant         159571 non-null  int64
 3   highly_malignant  159571 non-null  int64
 4   rude              159571 non-null  int64
 5   threat            159571 non-null  int64
 6   abuse             159571 non-null  int64
 7   loathe            159571 non-null  int64
dtypes: int64(6), object(2)
memory usage: 9.7+ MB
AxesSubplot(0.125,0.125;0.62x0.755)
```

c. Check the summary of our dataset using

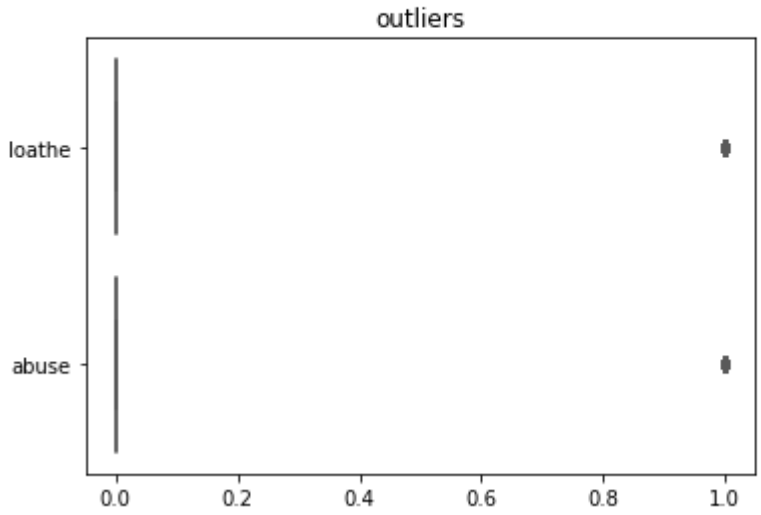
```
df.describe()   #Calculates the data statistically of all columns
```

	malignant	highly_malignant	rude	threat	abuse	loathe
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
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

From above we can say that all values for each column is different. There are no more difference in 75% values and max value , so we can say that our data is proper.

Checking Outliers: Our data is a raw data so we have to check whether any value is out of the box or not. If any outliers are present in any column, so in that case we have remove that particular value from our dataset. Otherwise it will give us wrong predictions.

```
sns.boxplot(data=df[['loathe','abuse']],orient='h',palette='Set2')
plt.title('outliers')
plt.show()
```

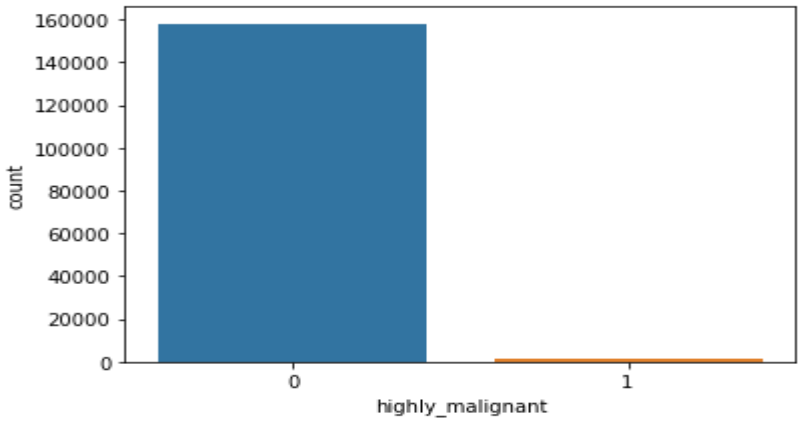


- Data Inputs- Logic- Output Relationships

  In this step we have to do the analysis of each column using univariate method. So for this I have used Count method to check the number of unique values present in that particular column and also some visualization libraries to understand our data more clearly.

```
c=df['highly_malignant'].value_counts()
print(c)
sns.countplot(x='highly_malignant',data=df)
plt.show()
```

```
0    157976
1      1595
Name: highly_malignant, dtype: int64
```



Now we have to find the correlation between all variables.

```
df.corr()  #find the pairwise correlation of all columns in the dataframe
```

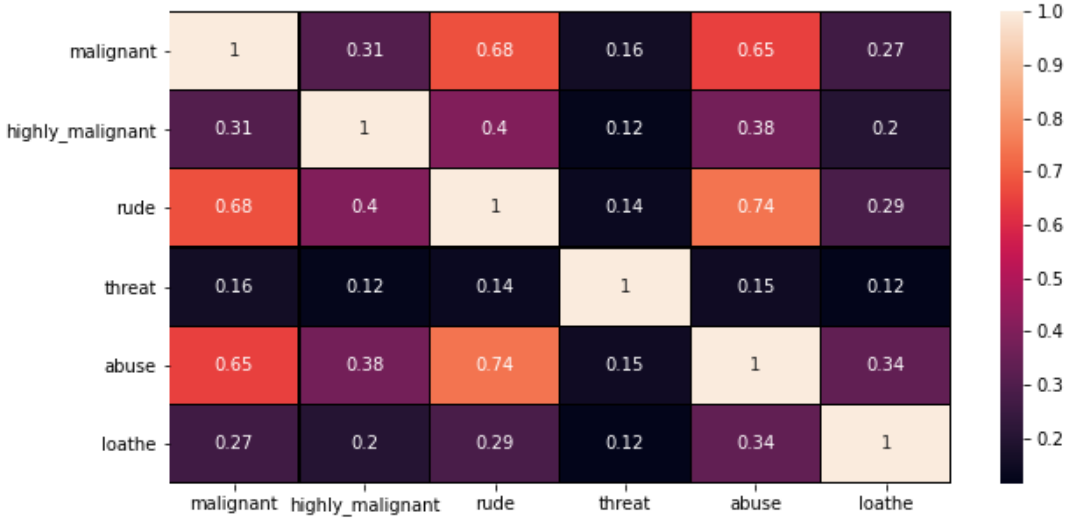
	malignant	highly_malignant	rude	threat	abuse	loathe
malignant	1.000000	0.308619	0.676515	0.157058	0.647518	0.266009
highly_malignant	0.308619	1.000000	0.403014	0.123601	0.375807	0.201600
rude	0.676515	0.403014	1.000000	0.141179	0.741272	0.286867
threat	0.157058	0.123601	0.141179	1.000000	0.150022	0.115128
abuse	0.647518	0.375807	0.741272	0.150022	1.000000	0.337736
loathe	0.266009	0.201600	0.286867	0.115128	0.337736	1.000000

- State the set of assumptions (if any) related to the problem under consideration

  This above method is used to check the correlation between all columns. Heatmap method gives the graphical visualization from which we can easily understand the correlation. So from the diagram below we can say that-

```
#lets check the correlation using heatmap for better understanding
plt.subplots(figsize=(10,5))
sns.heatmap(df.corr(),linewidths=.1,linecolor='black', annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ade3fbcbe0>
```



In this dataset all columns are making good positive correlation with other columns.

- Hardware and Software Requirements and Tools Used

  No external hardware required for this project. Only you have to do in laptop.

  Jupyter notebook with any latest version is required as a software.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Machine learning helps to solve the all above problems. It will not only used for the recent data but also helpful to predict the future outcomes with the help of past outcomes.

  Machine learning algorithms help to create a new model using existing and historical data that we can use for the training and testing our model to predict the future outcome.

  In this project we have used various machine learning algorithms to predict the proper outcome which gives as much as accuracy for our model.

- Testing of Identified Approaches (Algorithms)

Following algorithms we have used for our model which gives better performance for the dataset.

1. Logistic regression – Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

```
#Import all model library
from sklearn.linear_model import LogisticRegression
lg=LogisticRegression()
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn.model_selection import GridSearchCV,cross_val_score
import warnings
warnings.filterwarnings('ignore')
```

2. Decision tree classifier – Decision tree is a powerful algorithm in machine learning which can used when working on the real world datasets.

Decision Tree Classifier is a simple and widely used classification technique. It applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receives an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.

- Run and Evaluate selected models

Following are the methods used for our model-

```
from sklearn.tree import DecisionTreeClassifier
dc=DecisionTreeClassifier()
dc.fit(x_train,y_train)
dc.score(x_train,y_train)
pred=dc.predict(x_test)
print(accuracy_score(y_test,pred)*100)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

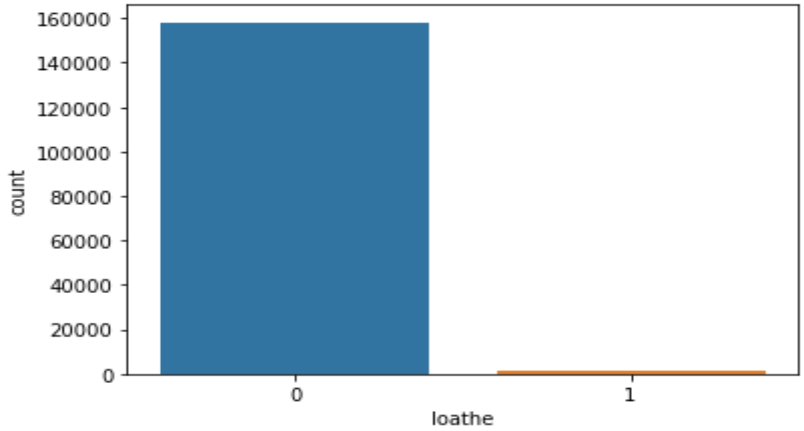
```
94.02150753264984
[[34726  1161]
 [ 1224  2782]]
              precision    recall  f1-score   support

           0       0.97      0.97      0.97     35887
           1       0.71      0.69      0.70      4006

    accuracy                           0.94     39893
   macro avg       0.84      0.83      0.83     39893
weighted avg       0.94      0.94      0.94     39893
```

From above diagram we can say that logistic regression gives the accuracy above 95% and decision tree gives accuracy above 94%.

- Visualizations

```
a=df['loathe'].value_counts()
print(a)
sns.countplot(x='loathe',data=df)
plt.show()
```

```
0    158166
1      1405
Name: loathe, dtype: int64
```



Seaborn and Matplotlib are two visualization libraries that I used to visualize graphically my data. This is the univariate method because I have visualized only one variable. The same method I have used for remaining variables.

- Interpretation of the Results

From all above process we can say that whatever algorithms or method we used for prediction of our model all process with visualization is very beneficial. Using any visualization library it is easy for us to understand the all data.

We have to interpret the result on test dataset.

```
#Loading the test data
test_df=pd.read_csv('test.csv')
test_df
```

	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.

```
test_df =tf_vec.fit_transform(test_df['comment_text'])
test_df
```

```
<153164x10000 sparse matrix of type '<class 'numpy.float64'>'
        with 2940213 stored elements in Compressed Sparse Row format>
```

```
pickle.load(open('malignant.pkl','rb'))
```

```
LogisticRegression()
```

From above image we can say that the best fitted model for our training dataset was logistic regression. So the same we have used for the testing dataset and saved our model using pickle method with logistic regression.

CONCLUSION

- Key Findings and Conclusions of the Study

1. In the experimentation phase two different machine learning algorithms were employed for the proposed model development and their performances were evaluated on various parameters.

2. From all above process we can say that whatever algorithms or method we used for prediction of our model all process with visualization is very beneficial. Using any visualization library it is easy for us to understand the all data.

3. For this data Logistic regression proves the best and others we have used also has good result but while using Logistic regression with different random state, it also gives the more than 95% accuracy.

Learning Outcomes of the Study in respect of Data Science

In this project we have worked on different machine learning models and additionally, I was able to implement them on a Natural Language Processing use-case. The various data pre-processing and feature engineering steps in the project made me cognizant of the efficient methods that can be used to clean textual data.