MALIGNANT COMMENTS CLASSIFIER

Submitted by:

KEWALARAMANI SHILPA

ACKNOWLEDGMENT

I would like to acknowledge the contributions of the following people without whose help and guidance this report would not have been completed. I acknowledge the counsel and support of our training coordinator, **Mr. Shubham Yadav**, SME of Internship14 and **Mr. Sajid Chaudhary SME** of Internship15 with respect and gratitude, whose expertise, guidance, support, encouragement, and enthusiasm has made this report possible. Their feedback vastly improved the quality of this report and provided an enthralling experience.

I am indeed proud and fortunate to be supported by them. I am also thankful to **Miss. Vaishali Singh** HR **of FlipRobo Technologies**, and **Miss Neha** Mentor of Batch No 1827 of **DataTrained Institute, Noida** for her constant encouragement, valuable suggestions and moral support.

Although it is not possible to name individually, I shall ever remain indebted to the faculty members **of Data Trained Institute of Technology & Management, Noida for** their persistent support extended during this work.

This acknowledgement will remain incomplete if I fail to express our deep sense of obligation to my parents and God for their consistent blessings and encouragement

INTRODUCTION

- Business Problem Framing

  Describe the business problem and how this problem can be related to the real world.

  Out of 7.7 Billion people around 3.5 billion people are active on the social platforms .There are positive as well as negative things on this platforms. A part of this things are comments on people, product, company, organisation etc this comments may be positive or negative . These comments may affect someone's life or someone's reputation very badly. It is very necessary to avoid such negative things to happen. As social platforms are never built with the intention of creating negative environment. Social platforms always aim at bringing people, ideas, community, information together through their services

- **Conceptual Background of the Domain Problem**

  Describe the domain related concepts that you think will be useful for better understanding of the project.

  The domain we can relate here is social platforms that updates people with the current affairs of the world . Here we can take examples of twitter. Twitter is that social platform that allows people to come and caste their view on any trending topic related to any topic like if some celebrity, who worked in movie and said some dialogue that can hurt any third person's feeling that third person can tweet (comment) publically and many people do support that third person and or sometimes a national political leader may announce some changes in certain policy. The beneficiaries of that policy usually use tweets to deny the changes made, sometimes many important news are shared through twitters like if Microsft announced new version of windows 11 to be officially launch after some date and many other things can be shared through twitter sometimes fans of celebrity do wishes birthday to their favourite stars online.

Review of Literature

This is a comprehensive summary of the research done on the topic. The review should enumerate, describe, summarize, evaluate and clarify the research done.

The research done on this topic is by looking at the comments_text is Online platforms when used by normal people can only be comfortably used by them only when they feel that they can express themselves freely and without any reluctance. If they come across any kind of a malignant or toxic type of a reply which can also be a threat or an insult or any kind of harassment which makes them uncomfortable, they might defer to use the social media platform in future. Thus, it becomes extremely essential for any organization or community to have an automated system which can efficiently identify and keep a track of all such comments and thus take any respective action for it, such as reporting or blocking the same to prevent any such kind of issues in the future. This is a huge concern as in this world, there are 7.7 billion people, and, out of these 7.7 billion, more than 3.5 billion people use some or the other form of online social media. Which means that every one-in-three people uses social media platform. This problem thus can be eliminated as it falls under the category of Natural Language Processing. In this, we try to recognize the intention of the speaker by building a model that’s capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate. Moreover, it is crucial to handle any such kind of nuisance, to make a more user-friendly experience, only after which people can actually enjoy in participating in discussions with regard to online conversation.

- **Motivation for the Problem Undertaken**
  Describe your objective behind to make this project, this domain and what is the motivation behind.

  The aim behind working on this project is to stop getting such comments posted on the websites or categorize the comments as malignant or not malignant by creating the filter system on the website that first checks the decency of the comment and then let it post or not. Or the website may warn the person commenting on the post. Or let the positive comments be shown publically but the negative comments to be read out only by that person on which the comment is passed

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

- **Data Sources and their formats**

  What are the data sources, their origins, their formats and other details that you find necessary? They can be described here. Provide a proper data description. You can also add a snapshot of the data.

  The data is provided in the two sets the train dataset, and the test dataset. The train dataset contains 1597571 rows and 8 columns including id, comment_text, malignant, highly_malignant, rude, loathe, threat abuse. And the test data only contains id and comment_text and there were 1,53,164 rows. In the training data set if the comment text was malignant and rude then in the respective column, it was marked as 1 and all the columns were marked as 0. Similarly, if the comment text was only a threat then the threat column was marked as 1and other columns as 0. Sometimes if the comment was both malignant as well as abuse then both the column was marked as 1 and others as zero

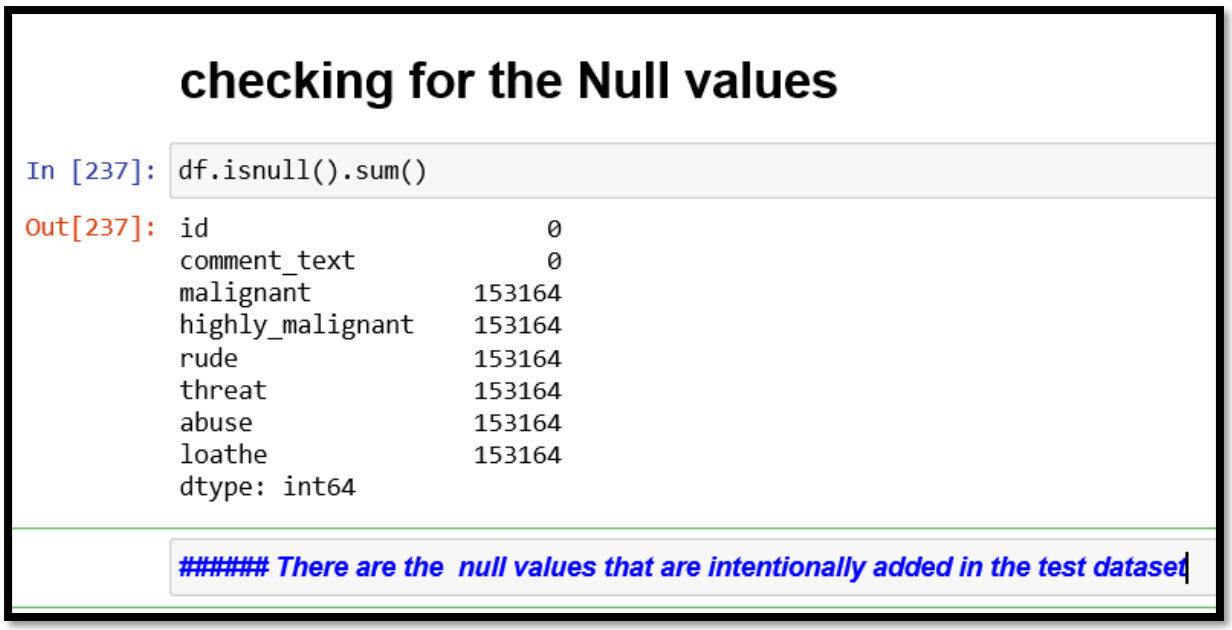
- **Data Preprocessing Done**

What were the steps followed for the cleaning of the data? What were the assumptions done and what were the next actions steps over that?

Checking the presence of Null values :-

The very first thing was to check whether there are null values or not in the dataset. Null Values or missing values usually occurs whenever the entry is left empty, but there were no null values in the data. The null values were checked using dataframe.isnull().sum() function which checks and returns the sum of all null values of individual columns

and it was observed that :- There were no null values in the train dataset. But Intentionally null values were added in the test dataset for preprocessing the dataset for model building.



Dropping the column

: The columns were not dropped but the required columns like label column and the comment_text was taken for model training and model building so all the columns other than this two were dropped inkind

Cleaning of The Data:

The Data Cleaning is the most important step in the NLP projects The text in the review column was cleaned thoroughly

1) The Original Length of Each comment_text was calculated and saved as a column for our understanding that up to what amount our data was cleaned.

processing data

In [266]:
```
# creating new column that counts the original lenth of the comments
df['Original_comment_length']=df.comment_text.str.len()
```

In [267]:
```
df.head()
```

Out[267]:

		id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe	label	Original_comment_length
	0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0.0	0.0	0.0	0.0	0.0	0.0	0	264
	1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0.0	0.0	0.0	0.0	0.0	0.0	0	112
train	2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0.0	0.0	0.0	0.0	0.0	0.0	0	233
	3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0.0	0.0	0.0	0.0	0.0	0.0	0	622
	4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0.0	0.0	0.0	0.0	0.0	0.0	0	67

2) At last the stopwords were removed but in the stopwords the words which have negative impressions were kept because the review is kind on sentiment analysis in it the negative word has equal importance as positive words so only words like for, it , is an , the me you etc were removed

```
In [275]: # remove stopwords and punctuations from above sentence

          import nltk
          import string
          from nltk.corpus import stopwords
          from nltk.tokenize import word_tokenize

          stopwords=stopwords_list
          punct=string.punctuation

In [276]: df['comment_text']=df['comment_text'].apply(lambda x: ' '.join( term for term in x.split() if term not in stopwords ))
```

3) After taking length then All the data was converted into lower case
4) After that Punctuation was removed in which all the special characters were removed
5) All the whitespace were removed

```
In [277]: # converting all the comments to lower case
          df['comment_text']=df['comment_text'].str.lower()

In [278]: # Remove punctuation
          df['comment_text'] = df['comment_text'].str.replace(r'[^\w\d\s]', ' ')

          # Replace whitespace between terms with a single space
          df['comment_text'] = df['comment_text'].str.replace(r'\s+', ' ')

          # Remove leading and trailing whitespace
          df['comment_text'] = df['comment_text'].str.replace(r'^\s+|\s+?$', '')

In [279]: df['Cleaned_comment_length']=df.comment_text.str.len()
```

6) A new column cleaned text was also created that calculated the text length after cleaning data and a good difference was found among it

7) And also a new column label was created in the comments were categorised as malignant, highly_malignant, rude, loathe, threat and abuse separately and saved and the columns that do not showed any of such comments were labelled as not_malignant.

8) As in the problem statement it was told to do the binary classification then in the label columns all highly_malignant, rude, threat, loathe, abuse etc were replaced as malignant and 0 was replaced as not malignant

Label Encoding The Column label which at last having only two values i.e., ‘malignant’ and ‘not_malignant’ was manually label encoded as malignant=1 and not_malignant as 0.

Label Encoding- malignant with 1 and not malignant with 0

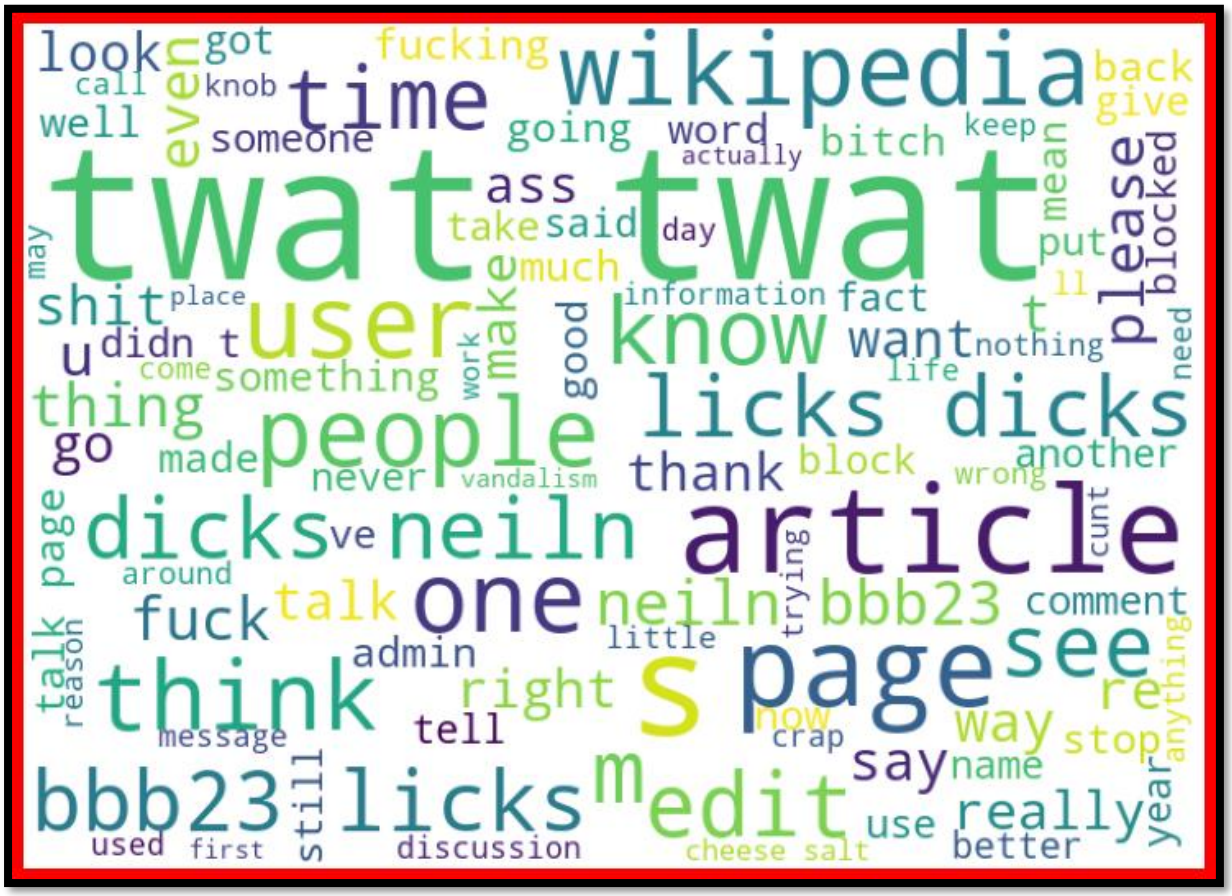
```
In [292]: # label encoding the columns manually
          train_data['label']=train_data['label'].replace('malignant',1).replace('not_malignant',0)

In [293]: train_data['label'].value_counts()

Out[293]: 0    143346
          1     16225
          Name: label, dtype: int64
```

- **Data Inputs- Logic- Output Relationships**

  Describe the relationship behind the data input, its format, the logic in between and the output. Describe how the input affects the output.

  The relationship between the output and the input i.e., in the comment_text and label columns was clearly seen in the form of wordcloud. The wordcloud of 100 words was used to show the relation as follow

  The wordcloud of rude comments was as follows



The wordcloud of abuse comments was as follow

The wordcloud of threat comments was as follows

The wordcloud of loathe comment was as follows

The wordcloud of Highly_malignant comment is

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

  As we are asked to build a binary class model of malignant and non_malignant comments so I created an extra column named as label in it I assume that if the comments was abuse and malignant then it was assigned as in whichever column the 1 comes first like in the label column it will be assigned either malignant or abuse but not both as given in the dataset. The same procedure was followed for all columns . and the column which was having zero in all columns was assigned as zero then again at last the values in label columns were replaced with 1 and 0. All this process is done during feature engineering.

- **Hardware and Software Requirements and Tools Used**

Listing down the hardware and software requirements along with the tools, libraries and packages used. Describe all the software tools used along with a detailed description of tasks done with those tools.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows to launch applications and easily manage conda packages , environments , and channels without using command-line commands. Navigators are searching for packages on Anaconda.org or in a Anaconda Repository.

Anaconda Navigator comprises of many applications like Jupyter Notebook, Jupyter lab, Pyshell and many more which are more essential for the Data Scientist.

Jupyter Notebook is an open-source web application that allows to create and share document that contain live code, equations, visualiazations and narrative text.

we used jupyter notebook for importing libraries, loading dataset, data cleaning and transformation, numerical simulation,statistical modelling, visualization of data, building the models and much more

Pandas Library 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.

In our project we use pandas for various purposes like for importing dataset we used read_csv() ,

At last for separating the features and target variable pandas library was used like drop function to drop target from feature and to create new feature column again dataframe of target was created using library function.

Numpy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and metrics. NumPy was created in 2005 by Travis Oliphant.

In our project we used numpy for various things like for converting negative into positive values we used absolute abs() function. Besides min,max ,Iqr are also derived from Numpy library.

Matplotlib and Seaborn is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Using matplotlib library we plot various graphs like histogram, countplot, boxplot, distplot. Using seaborn we plot heatmap of the correlation

Scikit-learn is a library in python that provides many unsupervised and supervised learning algorithms. Its built

upon some of the technology we used like NumPy , pandas, and Matplotlib

The functionality that scikit-learn provides include:

i. Regression, including Linear and Logistic Regression
ii. Classification , including K-Means and K-Means++
iii. Model selection
iv. Preprocessing including Min-Max Normalization

Like using sklearn.model_selection we used train_test_split, cross_val_score, GridSearchCV**,**

And we used classification library like

MultinomialNB, Logistic_Regression, DecisionTreeClassifier and even for ensemble technique we used sklearn.ensemble to import RandomForestClassifier

And for metrices to check the performance we used accuracy_score, classification_report , confusion_matrix from the sklearn.metrices

Imblearn

sampling

joblib or pickle

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

  Describe the approaches you followed, both statistical and analytical, for solving of this problem.

- **Testing of Identified Approaches (Algorithms)**

  Listing down all the algorithms used for the training and testing.

1) For Splitting the columns in dependent and independent features

   i)train_test_split

   train_test_split function was used which is imported from sklearn.model_selection

2) For Training the Model

   i)MultinomialNB

   MultinomailNB algorithm was used to train the model which was imported from sklearn.naive_bayes

   ii)RandomForestClassifier

   RandomForestClassifier was used to train the model which was imported from sklearn.ensemble

iii) DecisionTreeClassifier

DecisionTreeClassifier was used to train the model which was imported from sklearn.tree

iv) SupportVectorClassifier

SupportVectorClassifier was used to train the model which was imported from sklearn.svm

v) Logistic Regression

LogisticRegression was used to train the model which was imported from sklearn.linear_model

3) For Balancing the imbalanced datasets sampling techniques were used

i) RandomOverSampler

RandomOverSamppler Technique was used to oversample the dataset which was imported from imblearn.over_sampling

4) For coverting text to Vector

i) TfidfVectorizer

TfidfVectorizer was used to convert the text into vectors which was imported from sklearn.feature_extraction.text

5) For Saving the model

   i) Pickle

      Pickle library is imported to save the model in .pkl file

- **Run and Evaluate selected models**

  Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics.

1) Multinomial Naïve Bayes

Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.

Naive Bayes classifier is a collection of many algorithms where all the algorithms share one common principle, and that is each feature being classified is not related to any other feature. The presence or absence of a feature does not affect the presence or absence of the other feature.

But the Multinomial NB gave good accuracy and the f1-score, recall and precision score were not so good

Training the Data With MultinomialNB

```
In [299]: from sklearn.naive_bayes import MultinomialNB

          mnb=MultinomialNB()

          mnb.fit(x_train,y_train)

          y_pred= mnb.predict(x_test)

          print ('Accuracy score is  = > ', accuracy_score(y_test,y_pred))

          Accuracy score is  = >  0.9157124138323933
```

2) RandomForestClassifier

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests create decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random forests has a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

Using RandomForestClassifier to train model

In [311]:
```
# Train and predict
from sklearn.ensemble import RandomForestClassifier
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=42)

rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)

y_pred= rfc.predict(x_test)

print ('Final score = > ', accuracy_score(y_test,y_pred))
```

```
Final score = >  0.9415686962625022
```

3) DecisionTreeClassifier

A Decision Tree is constructed by asking a serious of questions with respect to a record of the dataset we have got. Each time an answer is received, a follow-up question is asked until a conclusion about the class label of the record. The series of questions and their possible answers can be organised in the form of a decision tree, which is a hierarchical structure consisting of nodes and directed edges. A tree has three types of nodes:

- root node that has no incoming edges and zero or more outgoing edges.
- Internal nodes, each of which has exactly one incoming edge and two or more outgoing edges.
- Leaf or terminal nodes, each of which has exactly one incoming edge and no outgoing edges.

In a decision tree, each leaf node is assigned a class label. The non-terminal nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics

Using DecisionTreeClassifier to train model

In [317]:
```
# Train and predict
from sklearn.tree import DecisionTreeClassifier
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=42)

dtc=DecisionTreeClassifier()
dtc.fit(x_train,y_train)

y_pred= dtc.predict(x_test)

print ('Final score = > ', accuracy_score(y_test,y_pred))
```

```
Final score = >  0.9370816935301932
```

4) Logistic Regression

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.

It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.

Using LogisticRegression to train model

In [305]:
```
# Train and predict
from sklearn.linear_model import LogisticRegression
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=42)

lg= LogisticRegression()
lg.fit(x_train,y_train)

y_pred= lg.predict(x_test)

print ('Final score = > ', accuracy_score(y_test,y_pred))
```

```
Final score = >  0.9543528939914271
```

5) SupportVectorClassifier

Support Vector Machines is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.

Using SupportVectorClassifier to train model

In [323]:
```
# Train and predict
from sklearn.svm import SVC
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=42)

svc=SVC()
svc.fit(x_train,y_train)

y_pred= svc.predict(x_test)

print ('Final score = > ', accuracy_score(y_test,y_pred))
```

```
Final score = >  0.9576868122226957
```

- Key Metrics for success in solving problem under consideration

  What were the key metrics used along with justification for using it? You may also include statistical metrics used if any.

  For justification log_loss, f1_score, recall , precision and accuracy were used

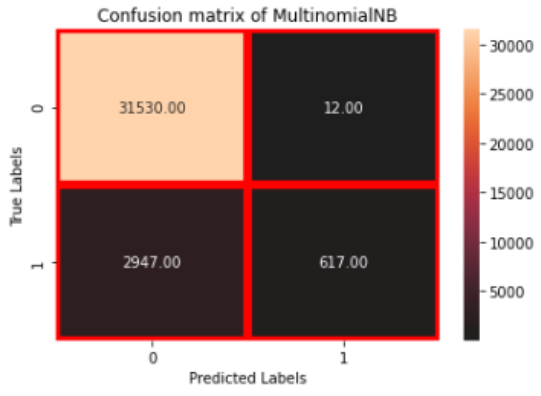
  1) MultinomialNB

  In MultinomialNB it was found that the accuracy was good as 91%, The log Loss was 0.33% , but the recall , f1_score was not so good of class 1

In [302]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.91      1.00      0.96     31542
           1       0.98      0.17      0.29      3564

    accuracy                           0.92     35106
   macro avg       0.95      0.59      0.62     35106
weighted avg       0.92      0.92      0.89     35106
```

In [303]:
```
# plot confusion matrix heatmap
conf_mat = confusion_matrix(y_test,y_pred)

ax=plt.subplot()

sns.heatmap(conf_mat,annot=True,ax=ax,linewidths=5,linecolor='r',center=0,fmt="0.2f")

ax.set_xlabel('Predicted Labels');ax.set_ylabel('True Labels')

ax.set_title('Confusion matrix of MultinomialNB')


plt.show()
```



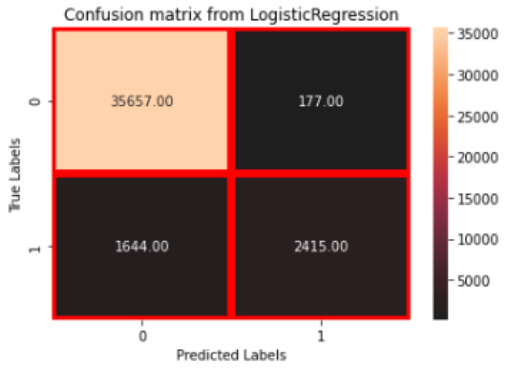
2) LogisticRegression

In MultinomialNB it was found that the accuracy was too good as 95%, The log Loss was 0.12 ,least as compared to all and but the recall , f1_score were also good of class 1 as compared to all

In [308]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     35834
           1       0.93      0.59      0.73      4059

    accuracy                           0.95     39893
   macro avg       0.94      0.80      0.85     39893
weighted avg       0.95      0.95      0.95     39893
```

In [309]:
```
# plot confusion matrix heatmap
conf_mat = confusion_matrix(y_test,y_pred)

ax=plt.subplot()

sns.heatmap(conf_mat,annot=True,ax=ax,linewidths=5,linecolor='r',center=0,fmt="0.2f")

ax.set_xlabel('Predicted Labels');ax.set_ylabel('True Labels')

ax.set_title('Confusion matrix from LogisticRegression')


plt.show()
```



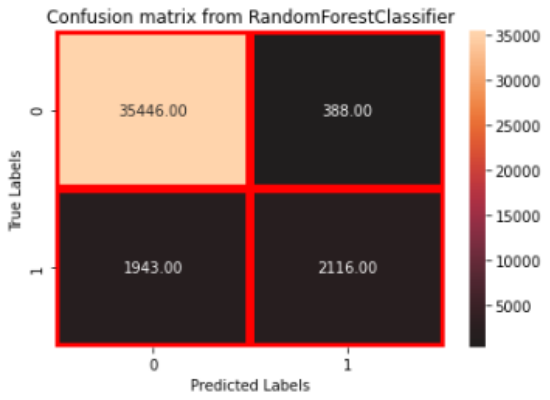
3) RandomForestClassifier

In RandomForestClassifier it was found that the accuracy was good as 94%, The log Loss was 0.15, and the recall , f1_score and precision was also good of class 1

In [314]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.95      0.99      0.97     35834
           1       0.85      0.52      0.64      4059

    accuracy                           0.94     39893
   macro avg       0.90      0.76      0.81     39893
weighted avg       0.94      0.94      0.94     39893
```

In [315]:
```
# plot confusion matrix heatmap
conf_mat = confusion_matrix(y_test,y_pred)

ax=plt.subplot()

sns.heatmap(conf_mat,annot=True,ax=ax,linewidths=5,linecolor='r',center=0,fmt="0.2f")
ax.set_xlabel('Predicted Labels');ax.set_ylabel('True Labels')

ax.set_title('Confusion matrix from RandomForestClassifier')


plt.show()
```



4) DecisionTreeClassifier

In DecisionTreeClassifier it was found that the accuracy was good as 93%, The log Loss was bad 2.17 , but the recall , f1_score was good of class 1

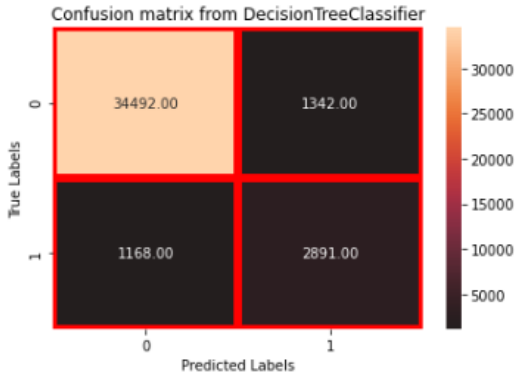
In [320]:

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.97      0.96      0.96     35834
           1       0.68      0.71      0.70      4059

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

In [321]:

```
# plot confusion matrix heatmap
conf_mat = confusion_matrix(y_test,y_pred)

ax=plt.subplot()

sns.heatmap(conf_mat,annot=True,ax=ax,linewidths=5,linecolor='r',center=0,fmt="0.2f")

ax.set_xlabel('Predicted Labels');ax.set_ylabel('True Labels')

ax.set_title('Confusion matrix from DecisionTreeClassifier')


plt.show()
```



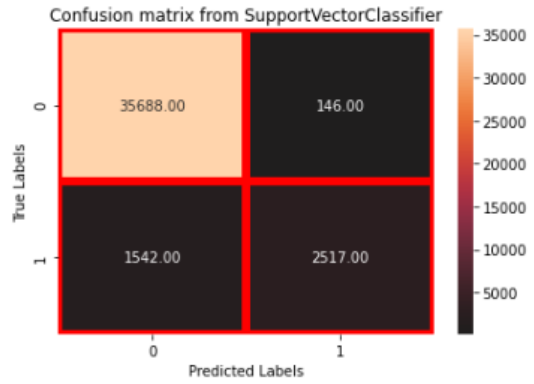
5) SupportVectorClassifier

In SupportVectorClassifier it was found that the accuracy was good as 95%, The log Loss was not calculated because in supportvectorClassifier the probability is false , but the recall was not so good, of class 1

In [326]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     35834
           1       0.95      0.62      0.75      4059

    accuracy                           0.96     39893
   macro avg       0.95      0.81      0.86     39893
weighted avg       0.96      0.96      0.95     39893
```

In [327]:
```
# plot confusion matrix heatmap
conf_mat = confusion_matrix(y_test,y_pred)

ax=plt.subplot()

sns.heatmap(conf_mat,annot=True,ax=ax,linewidths=5,linecolor='r',center=0,fmt="0.2f")

ax.set_xlabel('Predicted Labels');ax.set_ylabel('True Labels')

ax.set_title('Confusion matrix from SupportVectorClassifier')


plt.show()
```



- **Visualizations**

  Mention all the plots made along with their pictures and what were the inferences and observations obtained from those. Describe them in detail.

  If different platforms were used, mention that as well.

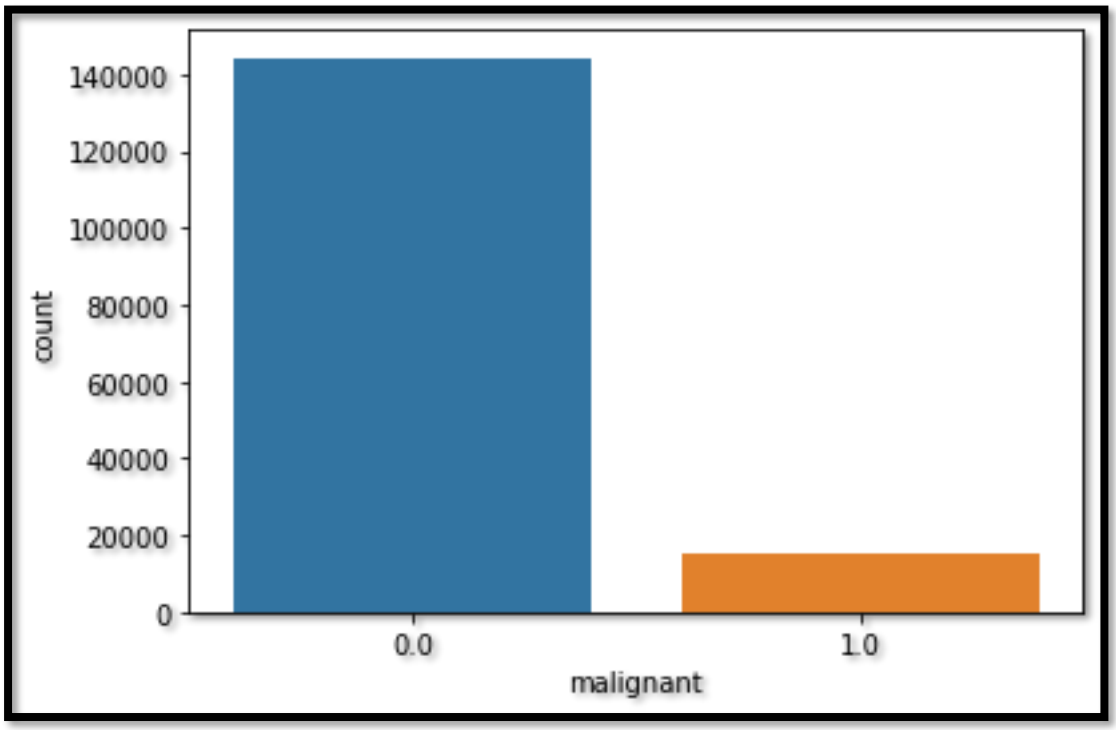
&

- **Interpretation of the Results**

  Give a summary of what results were interpreted from the visualizations, preprocessing and modelling.

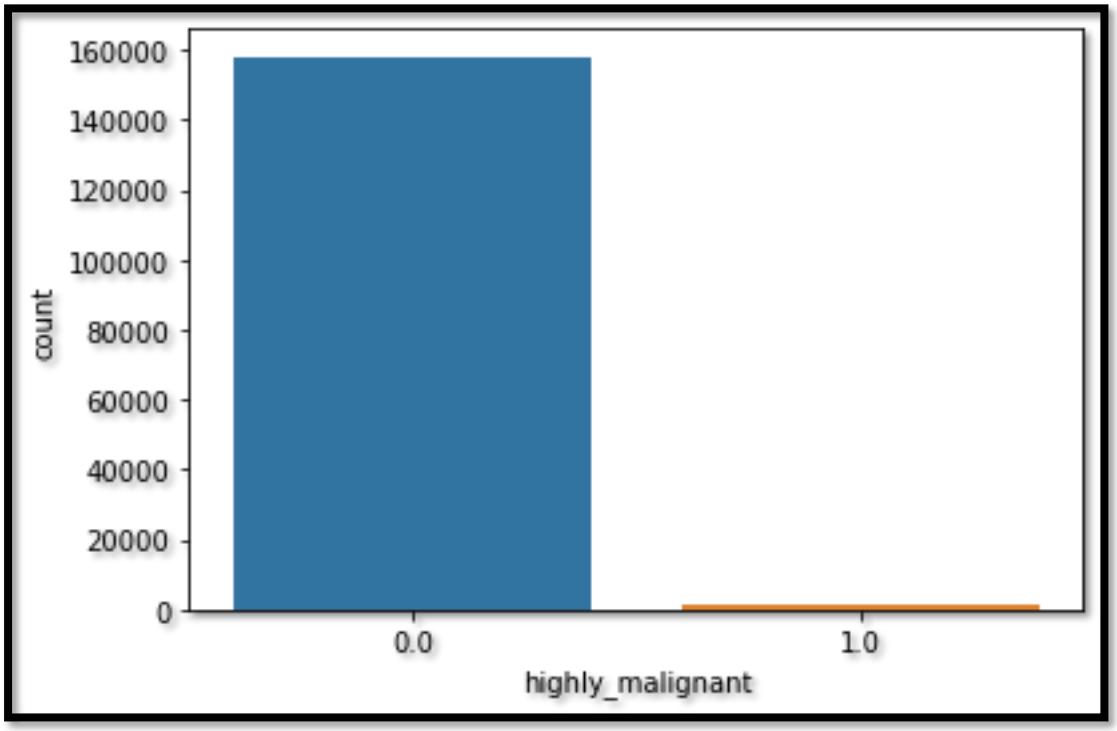
  Well in this project I have done univariate analysis and for that I used the countplot . I visualized the columns like malignant, highly_malignant, loathe, abuse, threat ,rude etc

  Malignant



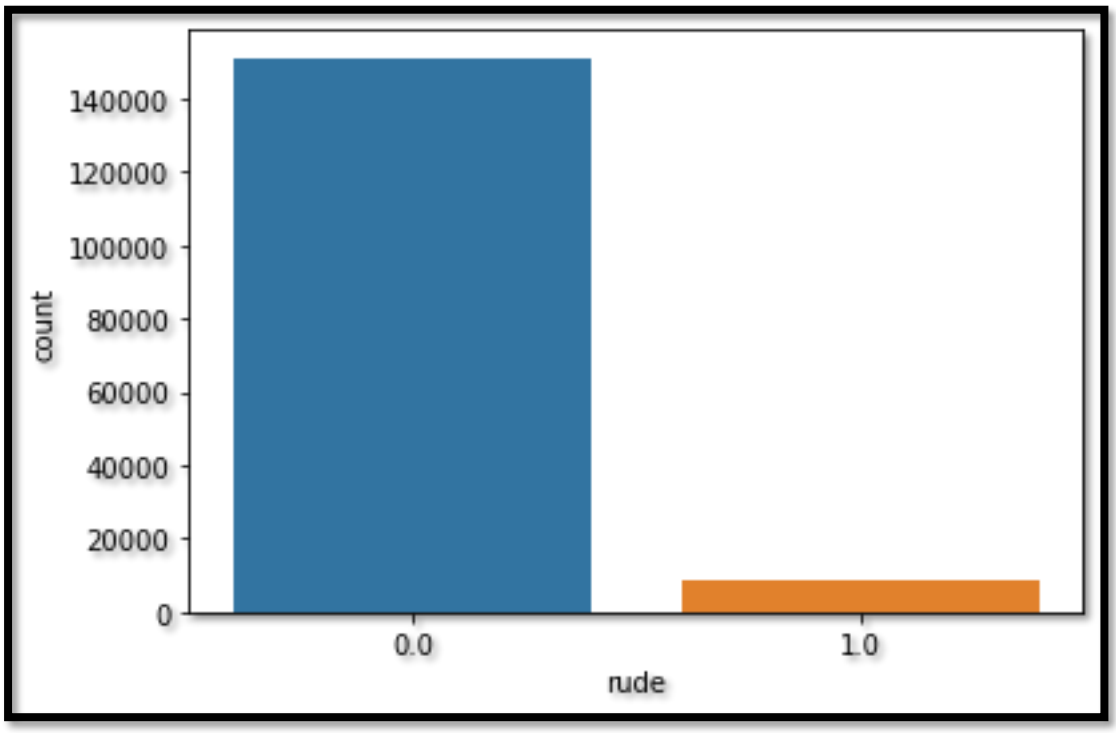
It was found that there were only 15294 malignant comments and other 144277 comments were not malignant.

Highly_Malignant



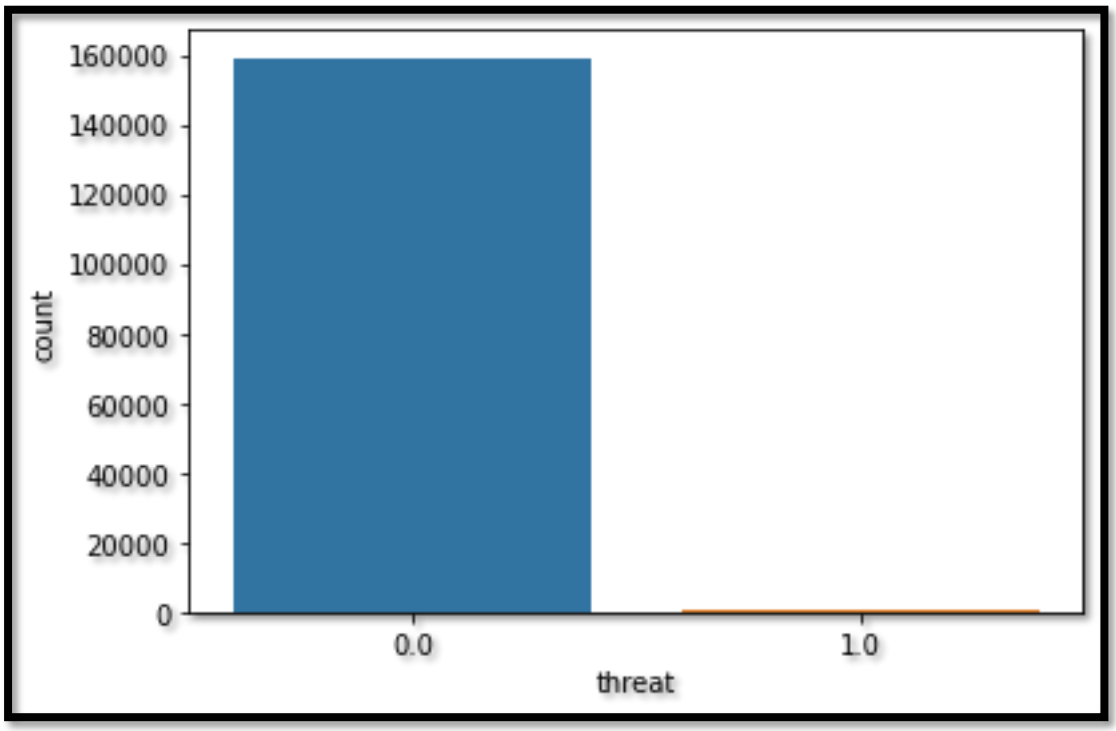
It was found that there were malignant and highly_malignat comments are same in number i.e., 15294 and other 144277 comments were not malignant

Rude



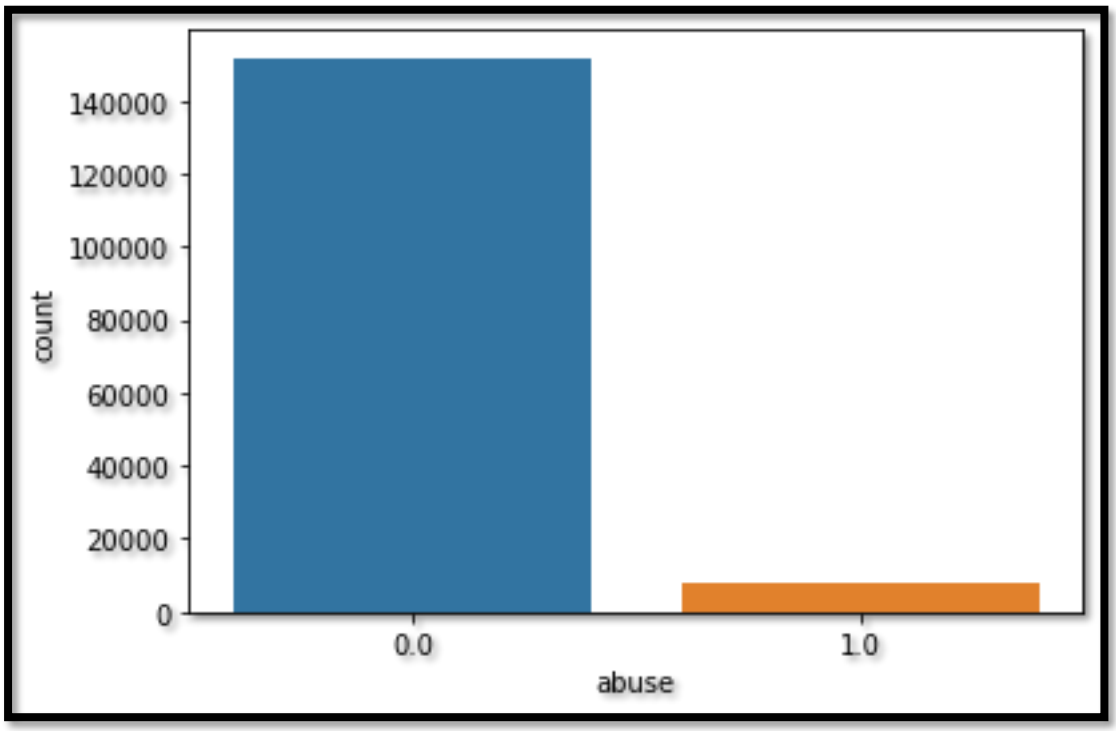
There were 8449 rude comments and 151122 non malignant comments

Threat



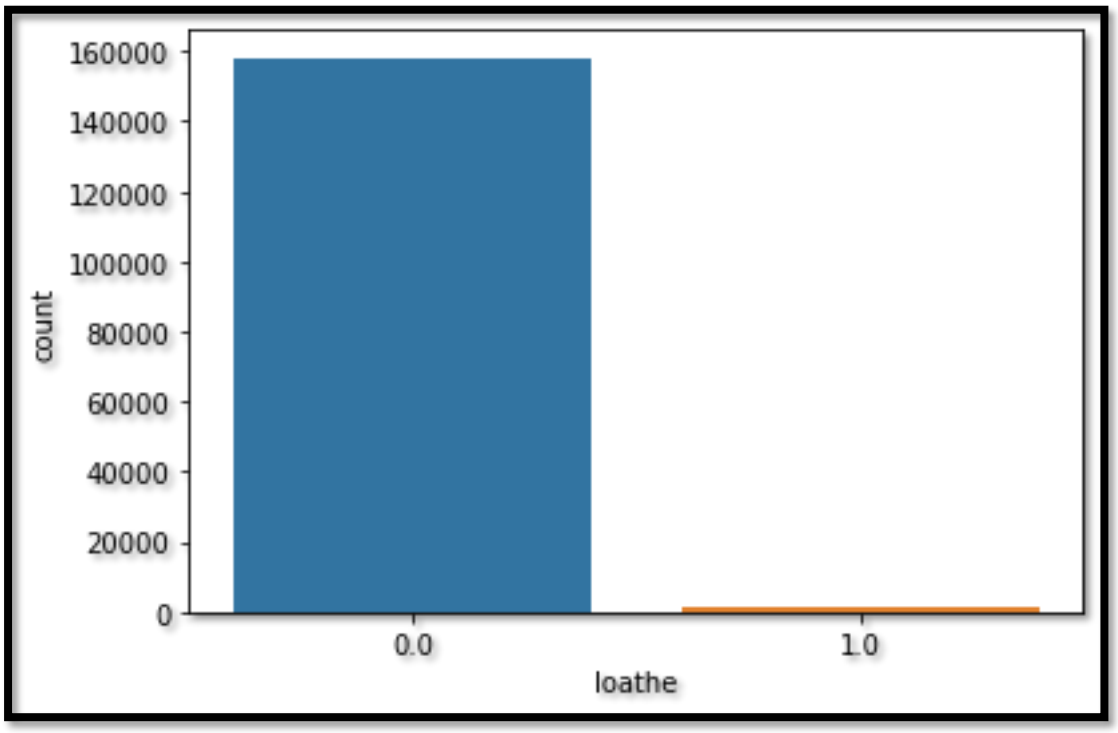
There were 478 Threat comments and 159093 non malignant comments

Abuse



There were 7877 abuse comments and 151694 non_malignant comments.

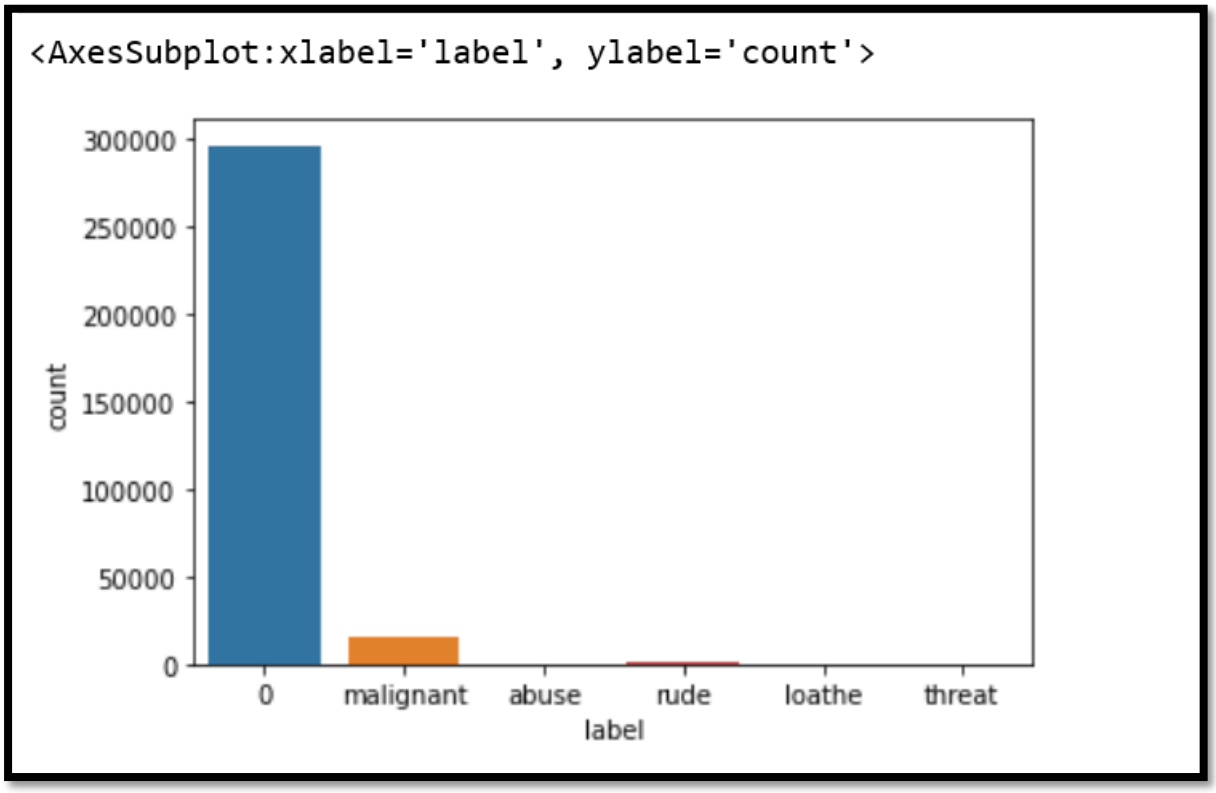
Loathe



There were 1405 loathe comments and 158166 non_malignant comments

label

A label column was created that giving the counts of each comments in a single column in it number of comments like there were 15,294 malignant comments 523 rude comments 329 abuse comments 54 loathe and 25 threat comments and remaining were not_malignant comments



And the other plots that were visualized were the confusion matrix . all the confusion matrix are specified above with explanations

CONCLUSION

- Key Findings and Conclusions of the Study

  Describe the key findings, inferences, observations from the whole problem.

&

- Learning Outcomes of the Study in respect of Data Science

  List down your learnings obtained about the power of visualization, data cleaning and various algorithms used. You can describe which algorithm works best in which situation and what challenges you faced while working on this project and how did you overcome that.

  From the whole project I learned many things like

  About Visualization

  1) countplot :
     the countplot from seaborn is very useful visualization plot it helps to clearly visualize the counts of unique values of columns in the form of bar graph. It is very useful in univariate analysis of a column.

  Data Cleaning

  Data Cleaning involves many process like Handling Null Values, Removing Outliers ,Removing Skewness, Handling Categorical Variables, Normalization of Data in data cleaning I learned but as this was NLP based projects so I learned that

  1) converting text to lower case is really usefull

2) Punctuations and special characters do not contribute well in model building
3) Stopwords create useless noise.
4) Max_feature parameter of TfIdfVectorizer is very useful that helps in solving the value error or the dimension mismatch error.

- Limitations of this work and Scope for Future Work

  What are the limitations of this solution provided, the future scope? What all steps/techniques can be followed to further extend this study and improve the results.

  Well I did not found any limitation, but to improve the result we need more malignant comments that will help the model to learn more better .