FLIP ROBO

Malignant-Comments-Classifier

Submitted by:

Saurabh Srivastava

ACKNOWLEDGMENT

A project is a bridge between theoretical and practical learning and with this thinking I worked on the project and made it successful due to timely support and efforts of all who helped me. This is to acknowledge all those without whom this projects would not have been reality. I would wish to thank our company ‘Flip Robo’ and our mentor ‘Kushboo Garg ‘ who gave me an opportunity to work on this project.

INTRODUCTION

- Business Problem Framing

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

The goal is to create a classifier model that predict if input text is inappropriate(malignant). People used to comment on Social Networking Platform to which helps us to identify such toxic comments.

- ## Conceptual Background of the Domain Problem

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

It is necessary to know about the words which are called to be Toxic.

- ## 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 dataset contains Train and test dataset.

Train dataset contains 159571 rows and 8 columns

Test dataset contains 153164 rows and 2 columns

Explore the dataset to get a better picture of how labels are distributed , how they are correlate with each other and what defines toxic or clean comments.

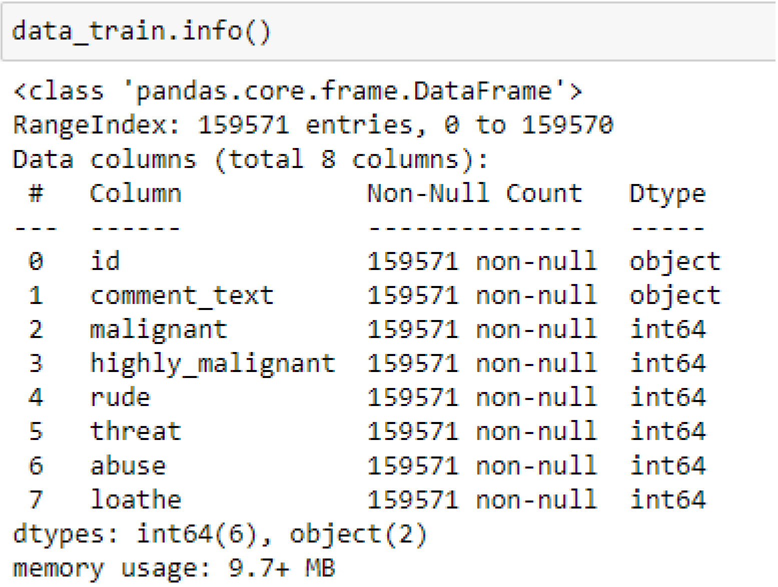
- ## Motivation for the Problem Undertaken

The objective is to create a classifier model that predict if input text is inappropriate(malignant) which helps people to identify the morally wrong and unfair people.

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.



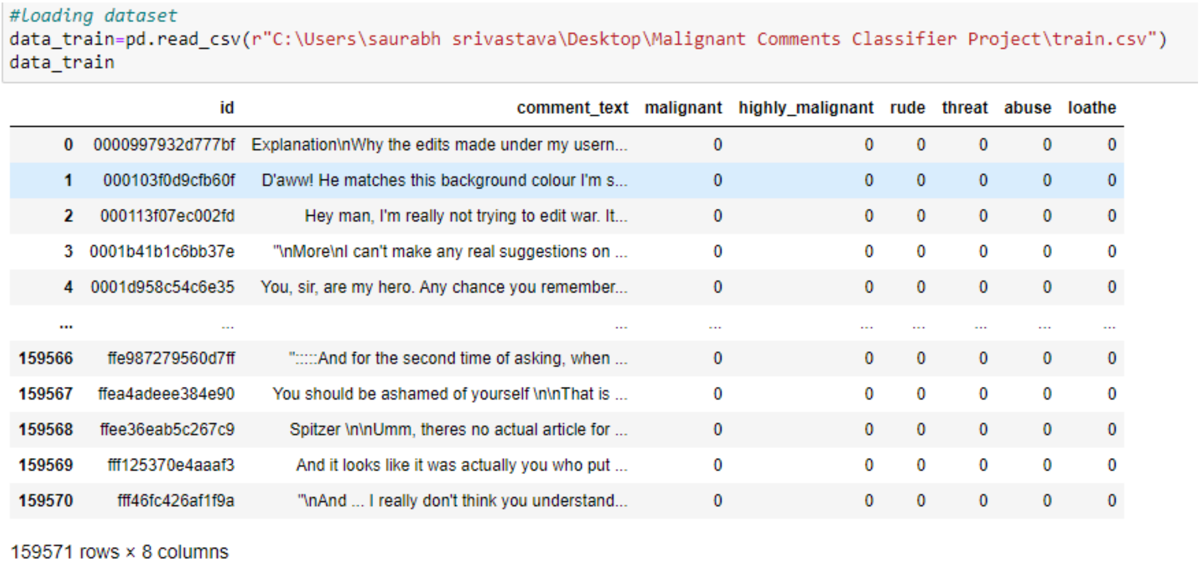
```
data_train.info()
```

```
<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
```

From the above screenshot there are two object type data and six integer type data and comment_text column will be consider as input feature and rest integer date type label column are consider as Target variable.

- ## 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.



```
#loading dataset
data_train=pd.read_csv(r"C:\Users\saurabh srivastava\Desktop\Malignant Comments Classifier Project\train.csv")
data_train
```

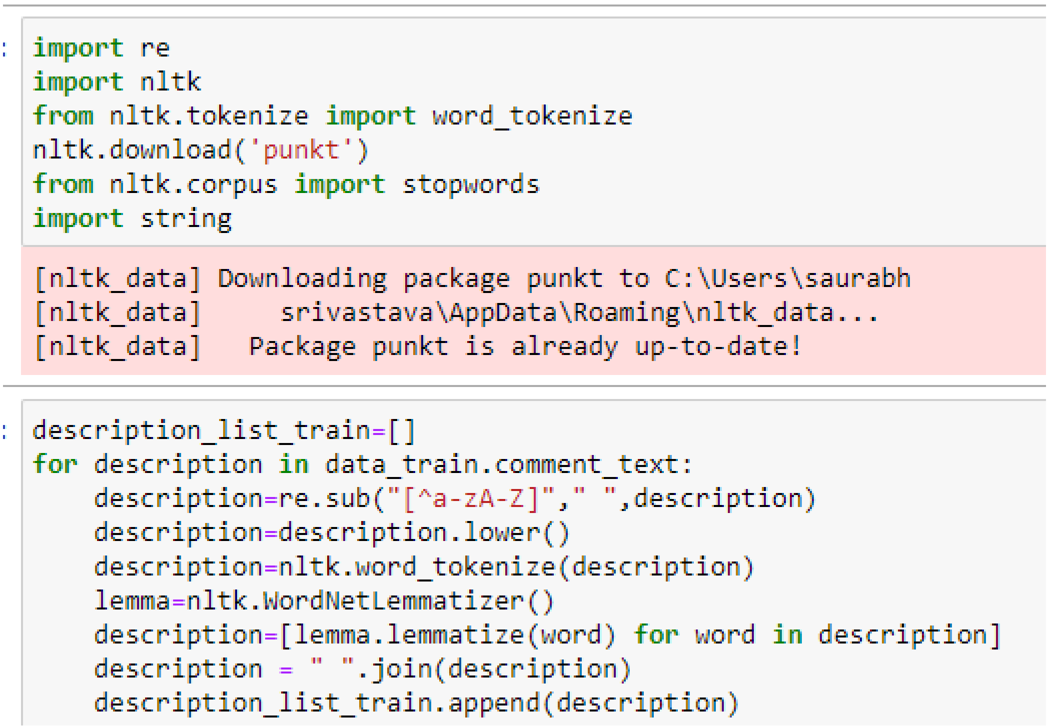
	id	comment_text	malignant	highly_malignant	rude	threat	abuse	loathe
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	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
...	...	...	...	...	...	...	...	...
159566	ffe987279560d7ff	":::::And for the second time of asking, when ...	0	0	0	0	0	0
159567	ffea4adeee384e90	You should be ashamed of yourself \n\nThat is ...	0	0	0	0	0	0
159568	ffee36eab5c267c9	Spitzer \n\nUmm, theres no actual article for ...	0	0	0	0	0	0
159569	fff125370e4aaaf3	And it looks like it was actually you who put ...	0	0	0	0	0	0
159570	fff46fc426af1f9a	"\nAnd ... I really don't think you understand...	0	0	0	0	0	0

159571 rows × 8 columns

From the above screenshot there are two object type data and six integer type data and comment_text column will be consider as input feature and rest integer date type label column are consider as Target variable.

- ## 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?



```
import re
import nltk
from nltk.tokenize import word_tokenize
nltk.download('punkt')
from nltk.corpus import stopwords
import string
```

```
[nltk_data] Downloading package punkt to C:\Users\saurabh
[nltk_data]     srivastava\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

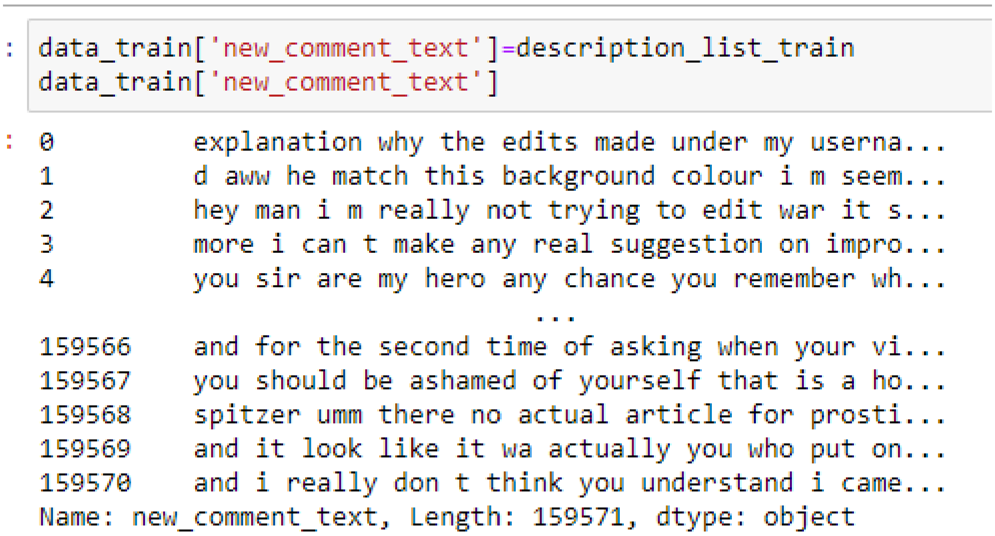
```
description_list_train=[]
for description in data_train.comment_text:
    description=re.sub("[^a-zA-Z]"," ",description)
    description=description.lower()
    description=nltk.word_tokenize(description)
    lemma=nltk.WordNetLemmatizer()
    description=[lemma.lemmatize(word) for word in description]
    description = " ".join(description)
    description_list_train.append(description)
```

Replacing all the characters which is not equal to a-z or A-Z with space.

Bringing down all the characters to lower case.

word_tokenize is a function in Python that splits a given sentence into words using the NLTK library.

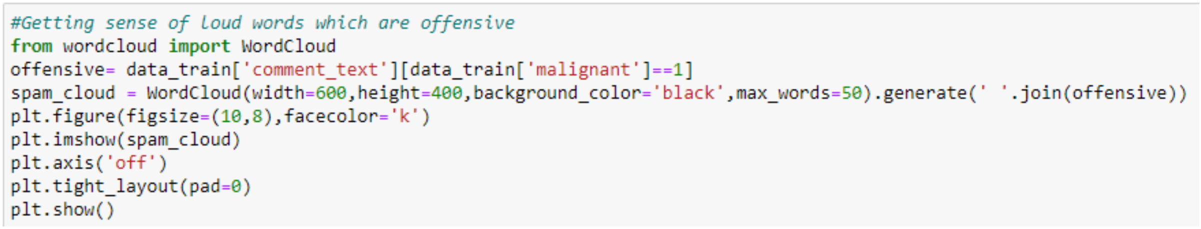
By using WordNetLemmatizer it converts the word to its meaningful base form.



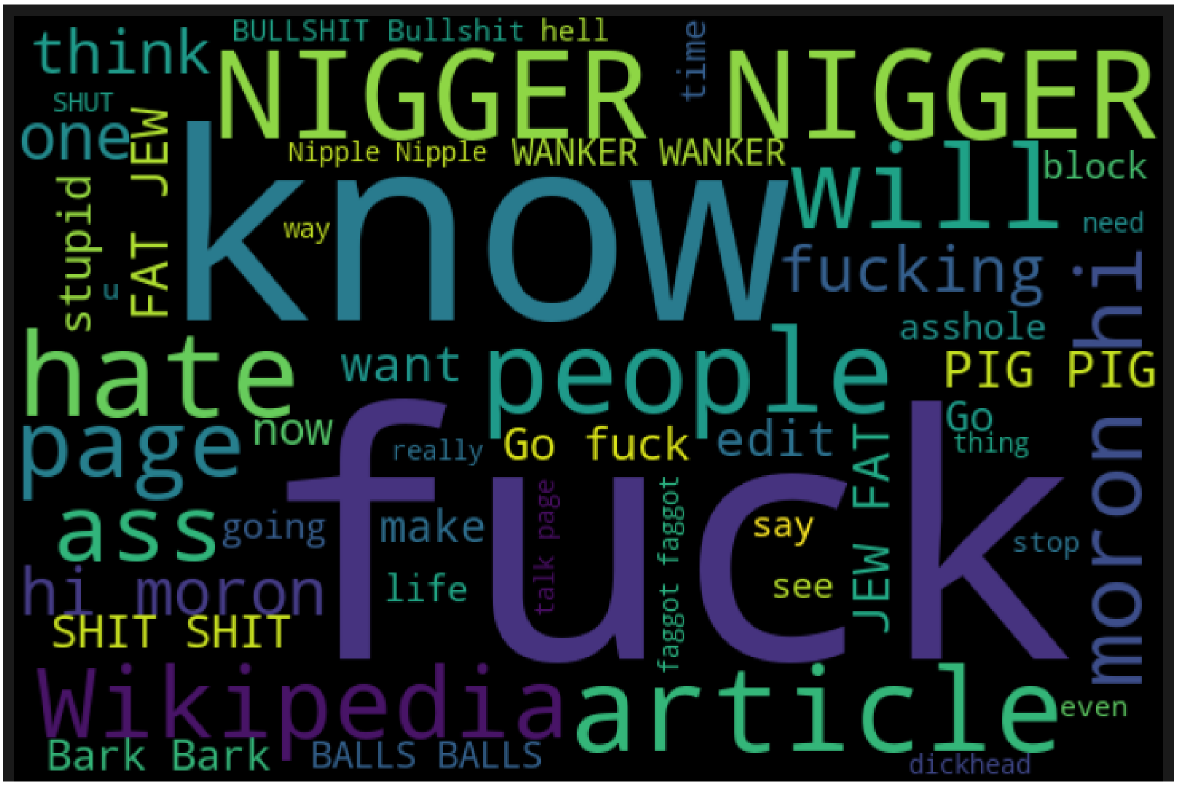
```
: data_train['new_comment_text']=description_list_train
  data_train['new_comment_text']
```

```
: 0         explanation why the edits made under my userna...
  1         d aww he match this background colour i m seem...
  2         hey man i m really not trying to edit war it s...
  3         more i can t make any real suggestion on impro...
  4         you sir are my hero any chance you remember wh...
                                  ...
  159566    and for the second time of asking when your vi...
  159567    you should be ashamed of yourself that is a ho...
  159568    spitzer umm there no actual article for prosti...
  159569    and it look like it wa actually you who put on...
  159570    and i really don t think you understand i came...
  Name: new_comment_text, Length: 159571, dtype: object
```

Now assigning the clean data to [new_comment_text] column

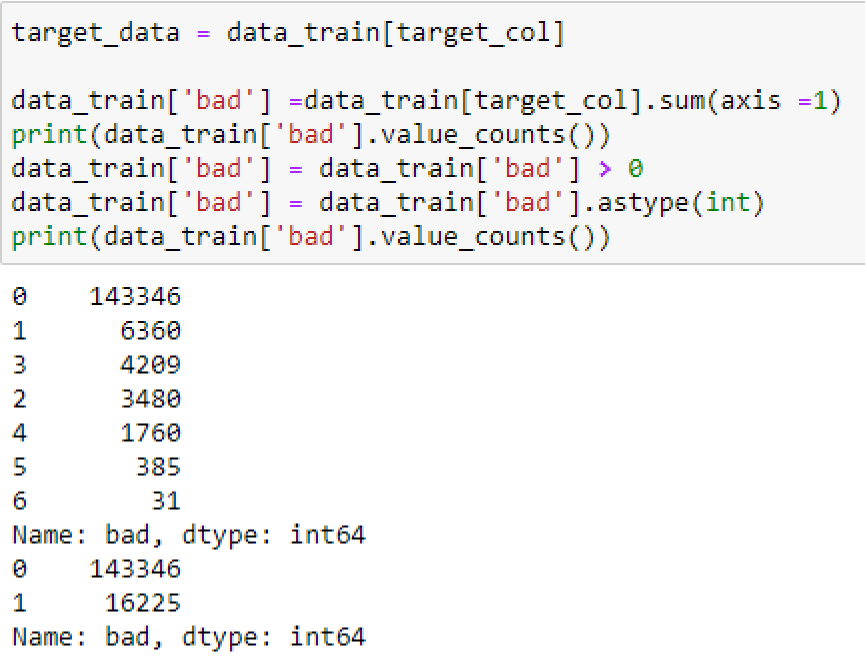


```
#Getting sense of loud words which are offensive
from wordcloud import WordCloud
offensive= data_train['comment_text'][data_train['malignant']==1]
spam_cloud = WordCloud(width=600,height=400,background_color='black',max_words=50).generate(' '.join(offensive))
plt.figure(figsize=(10,8),facecolor='k')
plt.imshow(spam_cloud)
plt.axis('off')
plt.tight_layout(pad=0)
plt.show()
```



Word Cloud is **a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance**.

From the Visualization it is clear that Text which is larger in size has more importance with context to its features.



```
target_data = data_train[target_col]

data_train['bad'] =data_train[target_col].sum(axis =1)
print(data_train['bad'].value_counts())
data_train['bad'] = data_train['bad'] > 0
data_train['bad'] = data_train['bad'].astype(int)
print(data_train['bad'].value_counts())
```

```
0    143346
1      6360
3      4209
2      3480
4      1760
5       385
6        31
Name: bad, dtype: int64
0    143346
1     16225
Name: bad, dtype: int64
```

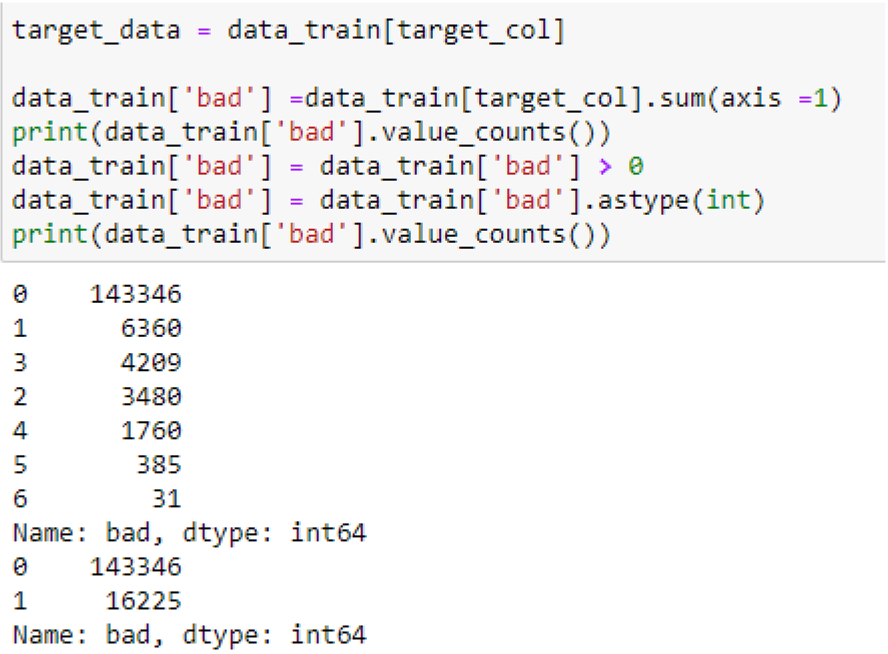
From the above screen-shot that new column named ‘bad’ is created and all target variable are stored in new data_train[‘bad’] column

- 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.

target_col=['malignant','highly_malignant','rude','threat','abuse','loathe']

all labels columns are stored in variable named-target_col



```
target_data = data_train[target_col]

data_train['bad'] =data_train[target_col].sum(axis =1)
print(data_train['bad'].value_counts())
data_train['bad'] = data_train['bad'] > 0
data_train['bad'] = data_train['bad'].astype(int)
print(data_train['bad'].value_counts())
```

```
0    143346
1      6360
3      4209
2      3480
4      1760
5       385
6        31
Name: bad, dtype: int64
0    143346
1     16225
Name: bad, dtype: int64
```

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

Here, you can describe any presumptions taken by you.

All labels have integer type data in the form of 0 and 1

Consider that all malignant data represents 1 and other as 0.

- 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.

Pandas is used for DataAnalysis

Numpy is used for Mathematical operations

Matplotlib and seaborn for data visualization.

Nltk is a text processing liabraries

WordCloud for text data visualization

Logistic Regression, Knn, Decision Tree and Random Forest algorithm used for building model.

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.

  To solve the text problem nltk approach was used .

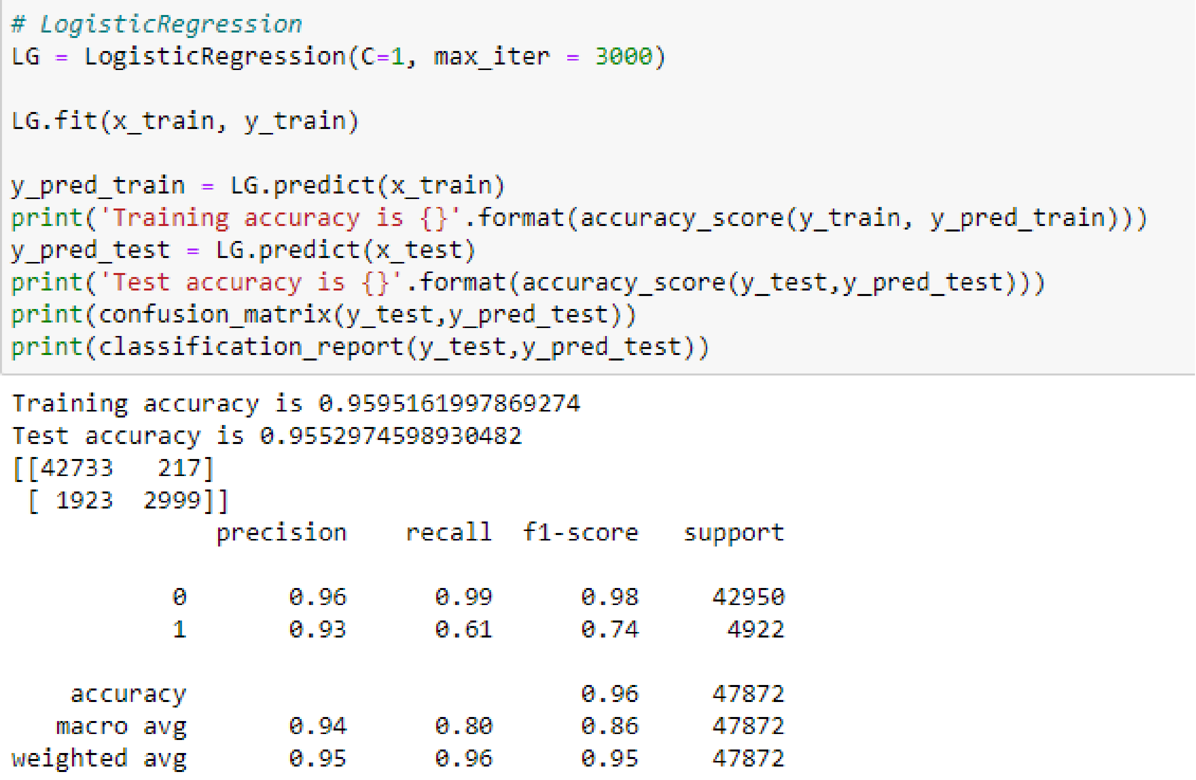
- ## Testing of Identified Approaches (Algorithms)

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

  Logistic Regression, Knn, Decision Tree and Random Forest algorithm used for building model.

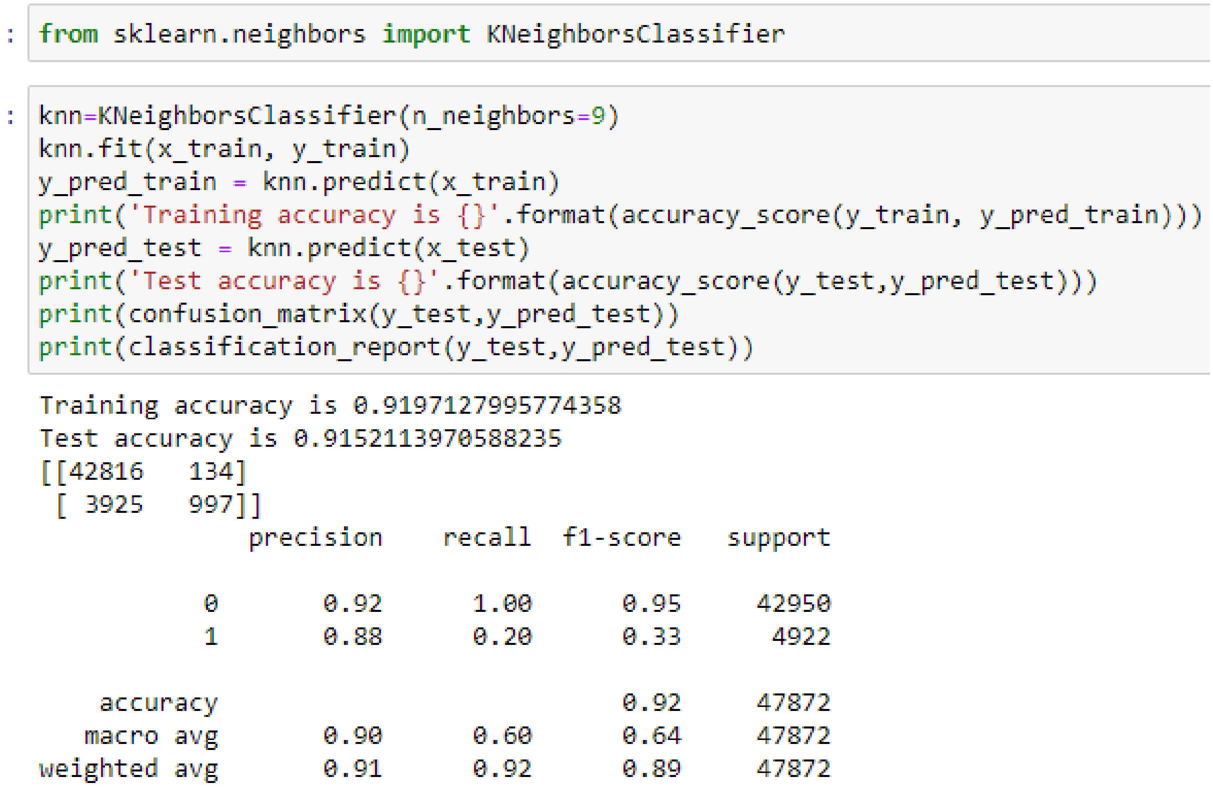
- ## Run and Evaluate selected models

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



```
# LogisticRegression
LG = LogisticRegression(C=1, max_iter = 3000)

LG.fit(x_train, y_train)

y_pred_train = LG.predict(x_train)
print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
y_pred_test = LG.predict(x_test)
print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
print(confusion_matrix(y_test,y_pred_test))
print(classification_report(y_test,y_pred_test))
```

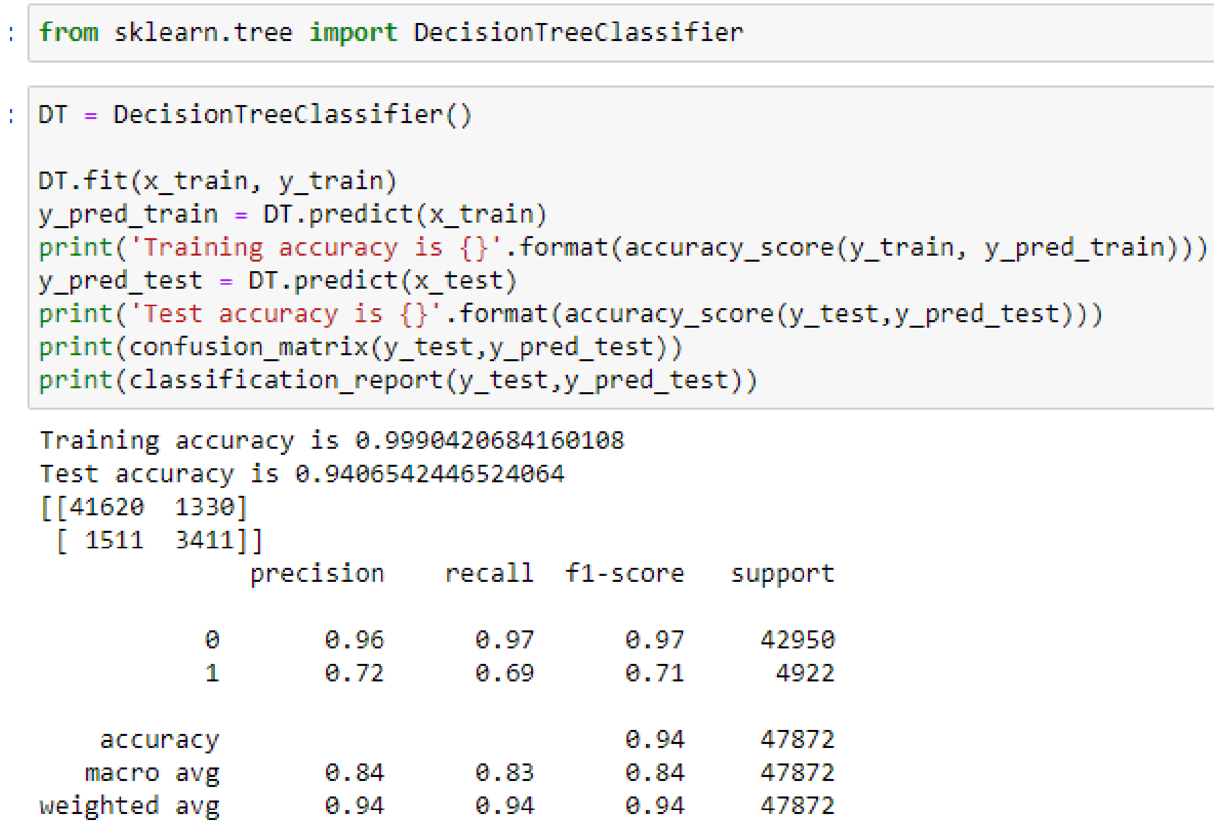
```
Training accuracy is 0.9595161997869274
Test accuracy is 0.9552974598930482
[[42733   217]
 [ 1923  2999]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.93      0.61      0.74      4922

    accuracy                           0.96     47872
   macro avg       0.94      0.80      0.86     47872
weighted avg       0.95      0.96      0.95     47872
```



```
: from sklearn.neighbors import KNeighborsClassifier
```

```
: knn=KNeighborsClassifier(n_neighbors=9)
  knn.fit(x_train, y_train)
  y_pred_train = knn.predict(x_train)
  print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
  y_pred_test = knn.predict(x_test)
  print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
  print(confusion_matrix(y_test,y_pred_test))
  print(classification_report(y_test,y_pred_test))
```

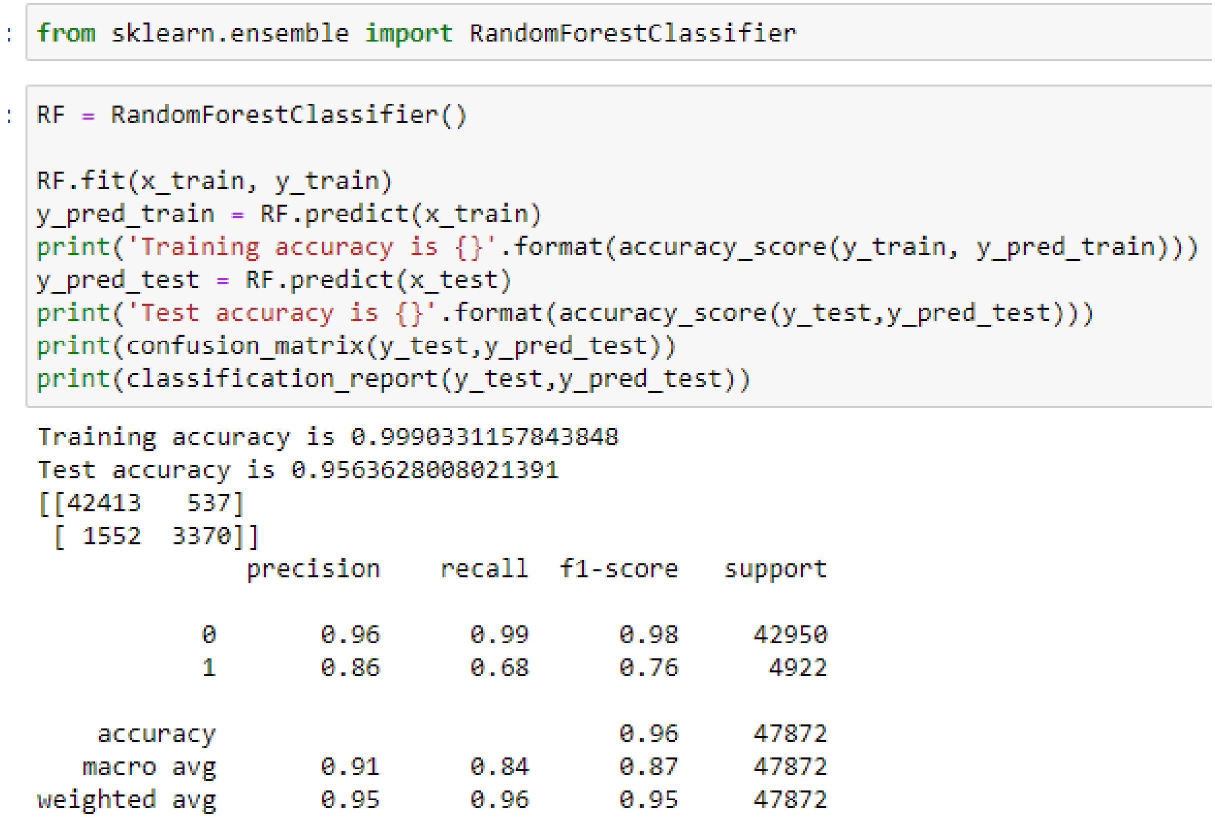
```
Training accuracy is 0.9197127995774358
Test accuracy is 0.9152113970588235
[[42816   134]
 [ 3925   997]]
              precision    recall  f1-score   support

           0       0.92      1.00      0.95     42950
           1       0.88      0.20      0.33      4922

    accuracy                           0.92     47872
   macro avg       0.90      0.60      0.64     47872
weighted avg       0.91      0.92      0.89     47872
```



```
: from sklearn.tree import DecisionTreeClassifier
```

```
: DT = DecisionTreeClassifier()

  DT.fit(x_train, y_train)
  y_pred_train = DT.predict(x_train)
  print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
  y_pred_test = DT.predict(x_test)
  print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
  print(confusion_matrix(y_test,y_pred_test))
  print(classification_report(y_test,y_pred_test))
```

```
Training accuracy is 0.9990420684160108
Test accuracy is 0.9406542446524064
[[41620  1330]
 [ 1511  3411]]
              precision    recall  f1-score   support

           0       0.96      0.97      0.97     42950
           1       0.72      0.69      0.71      4922

    accuracy                           0.94     47872
   macro avg       0.84      0.83      0.84     47872
weighted avg       0.94      0.94      0.94     47872
```



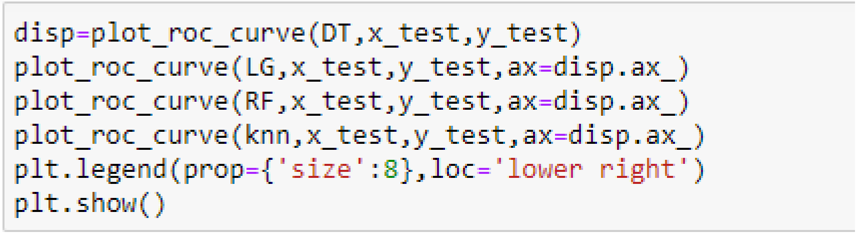
```
: from sklearn.ensemble import RandomForestClassifier
```

```
: RF = RandomForestClassifier()

  RF.fit(x_train, y_train)
  y_pred_train = RF.predict(x_train)
  print('Training accuracy is {}'.format(accuracy_score(y_train, y_pred_train)))
  y_pred_test = RF.predict(x_test)
  print('Test accuracy is {}'.format(accuracy_score(y_test,y_pred_test)))
  print(confusion_matrix(y_test,y_pred_test))
  print(classification_report(y_test,y_pred_test))
```

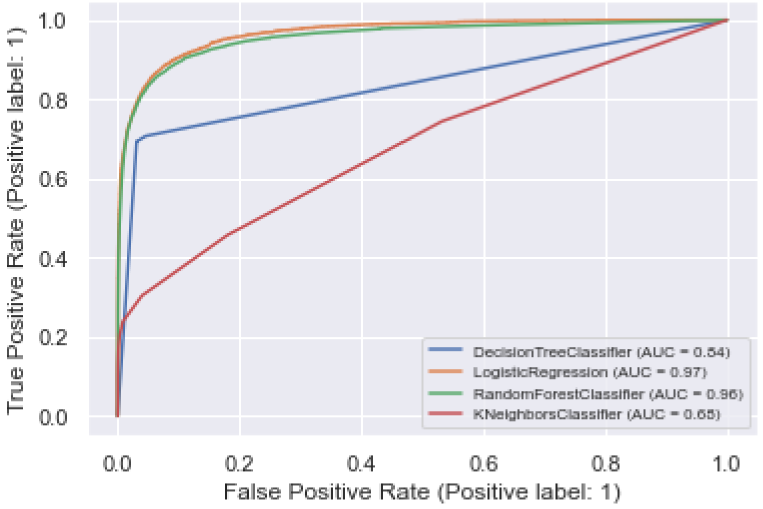
```
Training accuracy is 0.9990331157843848
Test accuracy is 0.9563628008021391
[[42413   537]
 [ 1552  3370]]
              precision    recall  f1-score   support

           0       0.96      0.99      0.98     42950
           1       0.86      0.68      0.76      4922

    accuracy                           0.96     47872
   macro avg       0.91      0.84      0.87     47872
weighted avg       0.95      0.96      0.95     47872
```

- 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.



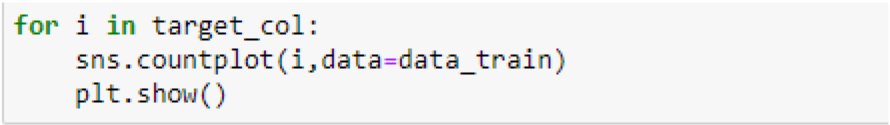
```
disp=plot_roc_curve(DT,x_test,y_test)
plot_roc_curve(LG,x_test,y_test,ax=disp.ax_)
plot_roc_curve(RF,x_test,y_test,ax=disp.ax_)
plot_roc_curve(knn,x_test,y_test,ax=disp.ax_)
plt.legend(prop={'size':8},loc='lower right')
plt.show()
```



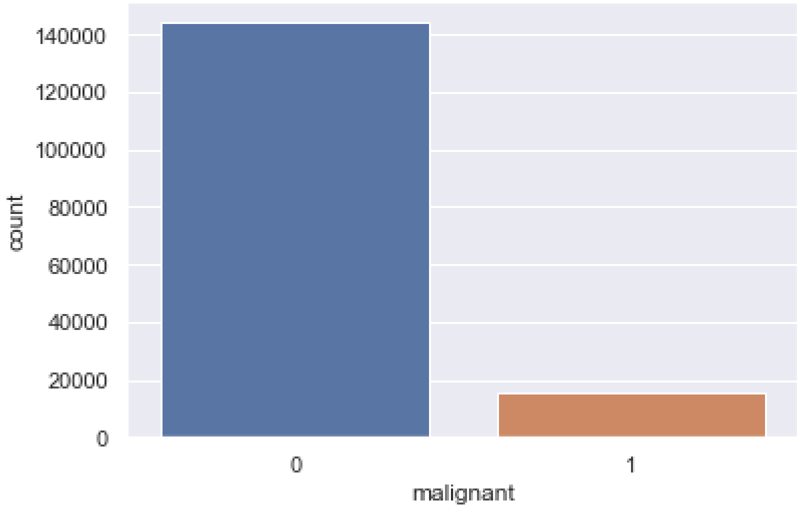
- Visualizations

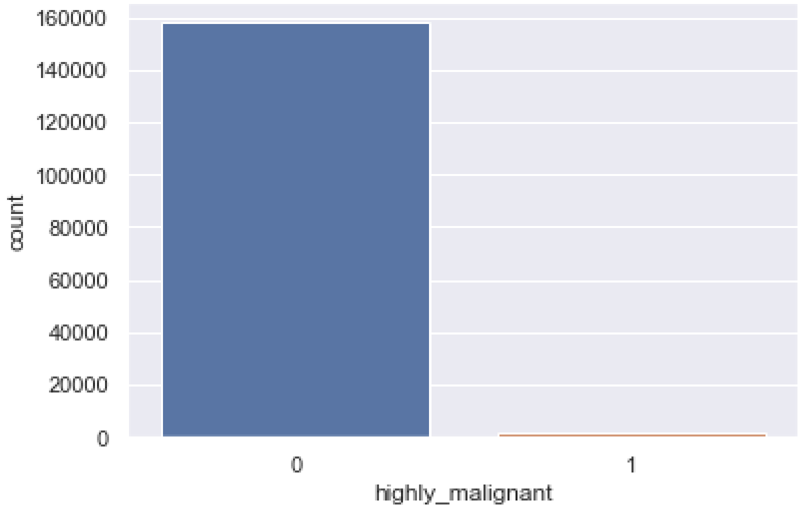
Mention all the plots made along with their pictures and what were the inferences and observations obtained from those. Describe them in detail.

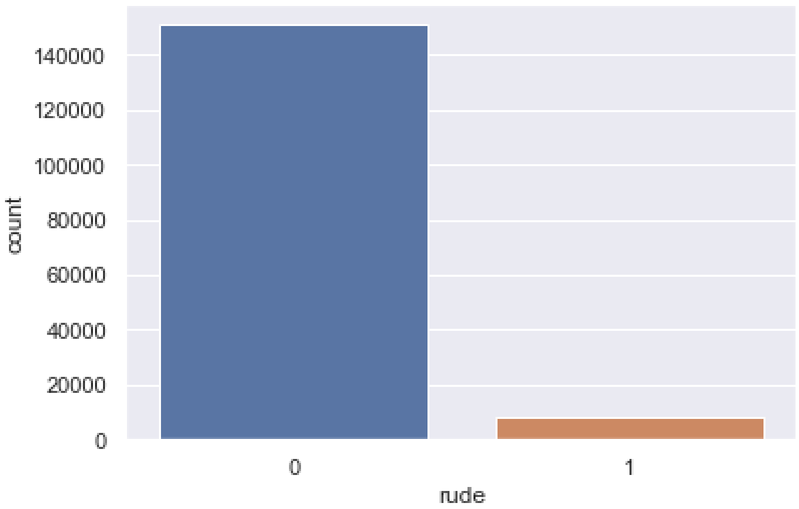
If different platforms were used, mention that as well.

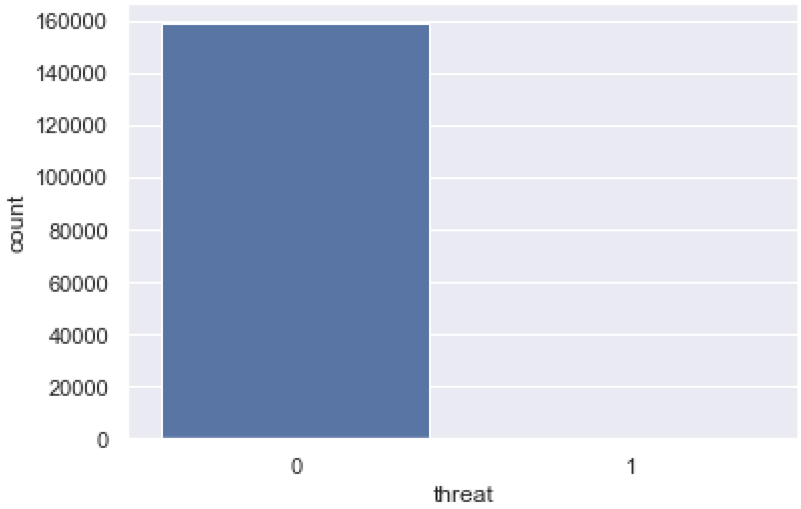


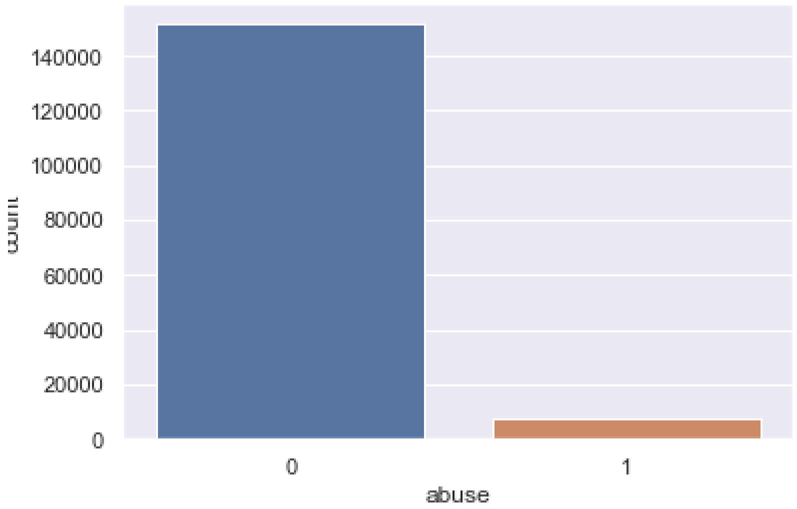
```
for i in target_col:
    sns.countplot(i,data=data_train)
    plt.show()
```

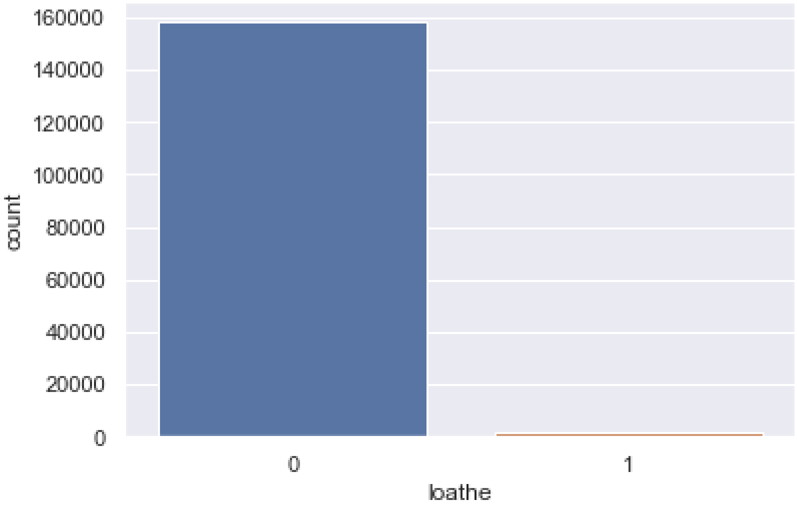





140000
120000
100000
80000
60000
40000
20000
0
count
0
1
rude


160000
140000
120000
100000
80000
60000
40000
20000
0
count
0
1
threat


140000
120000
100000
80000
60000
40000
20000
0
count
0
1
abuse



From the above visualization it is clear that malignant ratio is more as compared to other toxic comments.

- Interpretation of the Results

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

Dataset has imbalanced label variables.

Random Forest gives 96% accuracy and F1 score is about 76%.

CONCLUSION

- Key Findings and Conclusions of the Study

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

It has a imbalanced datasets.

Random Forest gives better results as compared to other algorithm.

- 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.

Random Forest algorithm able to predict well as it has imbalance dataset.