-*-

Malignant Comment Classifier

Submitted by:

Sumit Santra

ACKNOWLEDGMENT

In the present world of competition there is a race of existence in which those are having will to come forward succeed. Project is like a bridge between theoretical and practical working. With this willing I joined this particular project.

I have taken efforts in this project. However it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I would like to express my special thanks to my SME Shubham Yadav who gave me the golden opportunity to do this wonderful project on the topic **Malignant Comment Classifier**, which also helped me in doing a research and I came to know about so many new things.

I am really thankful to him.

I would also thankful to the online platforms who help me a lot in finishing this project within the limited time.

THANKS AGAIN TO ALL WHO HELPED ME.

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, cyber bullying, 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.

• Conceptual Background of the Domain Problem

So we build a machine learning model that helps to understand that which comment is malignant and which one is Non Malignant based on the sample data. We do Exploratory Data Analysis to visualize the data graphically which is easy to understand. We also used some statistical technique to see the insights of the data.

• Review of Literature

- 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.

- 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. First we do all the data preprocessing steps and do EDA to visualize the data graphically and after that we make a machine learning model in order to improve.

• Motivation for the Problem Undertaken

- Every investigation begins with ideas that are further developed and inspired to address a variety of situations and circumstances.
- The client wants some predictions that could help them in further investment and improvement in selection of comments. So to help them we make this project.
- My motivation behind this project is to do the proper research because research as a process for finding a solution to a problem after making a deep analysis and conducting studies of relevant factors. In general, research is a method designed to ensure that the information obtained is reasonable and supported by the quantitative and qualitative data, and that involves a systematic process. It includes the process of designing research methods, collecting and describing.

ANALYTICAL PROBLEM FRAMING

- **Mathematical/ Analytical Modeling of the Problem**

1. First we check the basic information of the dataset that is it's shape and it's information using pandas library. The basic information tells the data type of our column and number of data present.

2. Then we check the unique information of our data columns.

3. After that we check for null values, if it present then we remove it because our data is text data and we cannot fill it.

4. After that we check the summary statistics of our dataset. This part tells about the statistics of our dataset i.e. mean, median, max value ,min values and also it tell whether outliers are present in our dataset or not.

5. We have to also check whether our dataset is balanced or not.

6. We also create new columns to check the length of data before and after cleaning the comment feature to check the distribution of our data.

7. We use seaborn library to plot the target data and using wordcloud to for getting the sense of loud words in Malignant and Benign comments.

8. Similarly we can also make wordcloud for seprate columns where you can check the loud words for particular features because there are six target features.

9. Before making the model we convert our text into vectors so for that we use technique known as **TF-IDF Vectorizer**

- # Data Sources and their formats

- In this project the sample data is provided to us from our client database. The dataset is in csv (comma seprated values) format.
- The data set contains the training set, which has approximately 1,59,000 samples and the test set which contains nearly 1,53,000samples. All the data samples contain 8 fields which includes 'Id', 'Comments', 'Malignant', 'Highly malignant', 'Rude', 'Threat', 'Abuse' and 'Loathe'.
- The label can be either 0 or 1, where 0denotes a NO while 1 denotes a YES. There are various comments which have multiple labels. The first attribute is a unique ID associated with each comment.
- The data set includes:

- **Malignant:** It is the Label column, which includes values 0 and 1, denoting if the comment is malignant or not.
- **Highly Malignant:** It denotes comments that are highly malignant and hurtful.
- **Rude:** It denotes comments that are very rude and offensive.
- **Threat:** It contains indication of the comments that are giving any threat to someone.
- **Abuse:** It is for comments that are abusive in nature.
- **Loathe:** It describes the comments which are hateful and loathing in nature.
- **ID:**It includes unique Ids associated with each comment text given.

- **Comment text:** This column contains the comments extracted from various social media platforms.

• Data Preprocessing Done

- First we check the information of the given dataset because it tells that how many rows and columns are present in our dataset and data type of the columns whether they are object, integer or float.
- Dropping duplicates rows if present in dataset.
- Then we check for the null values present in our dataset. If null values are present then drop it because we cannot able to fill the text data.
- To visualize the amount of missing values in different-2 columns we use Missingno library.
- After that we check the summary statistics of our dataset. This part tells about the statistics of our dataset i.e. mean, median, max value ,min values and also it tell whether outliers are present in our dataset or not.
- Then we check our label that it is balanced or not. If not balance then we use sampling technique to balanced it.
- We also create new columns to check the length of data before cleaning the Input feature column.

- There are six features in this dataset so we combined them and make one target feature giving the name as label and also we do scaling for that feature.
- In data cleaning we use mailny five steps using function:

- Removing HTML tags
- Removing special characters
- Converting everything to lowercase
- Removing stopwords
- Using WordNetLemmatization for lemmatization

- We create new column (clean_text) after removing puncuations, stopwords from input feature to check how much data is cleaned.

• Data Inputs- Logic- Output Relationships

In this project, we focus on the automatic identification of malignant comments. Our contribution is twofold. First, we introduce the datasets for the task of malignant comment classifier, covering different comments . We describe the collection and present several exploratory analysis on the identification of linguistic differences in malignant and benign comments. Second, we conduct a set of learning experiments to build accurate malignant comment classifier to see how the inputs features affects the target features.

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

- As we know this dataset is imbalance so we don't too much focus on accuracy score . We see the precision and recall value along with f1_score.
- First we see the result without doing any sampling technique and for that I use Logistic Regression with KFold cross validation and hyperparameter tuning.
- We also use Random Forest and XGBoost as our evaluation model without using hyperparameter tuning because our dataset is too large and it takes more than hour to give the result.

• Hardware and Software Requirements and Tools Used

❖ Hardware:

- Processor—RYZEN 9 5900HX 4.20GHz
- Installed Memory(RAM)—16.00 GB
- System type—64-bit Operating System

❖ **Software**: Windos 10 Pro

❖ We have used Python Package because it is powerful and general purpose programming language.

- **NumPy**—It is a math library to work with ndimensional arrays. It enables us to do computation effectively and regurarly. For working with arrays, dictionary, functions data type we need to know NumPy.
- **Pandas**—It is high level Python library and easy to use for data importing , manipulation and data analysis.
- **Matplotlib**—It is a plotting that provide 2D and 3D plotting.

- **Seaborn**-- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- **SciPy**—It is a collection of numerical algorithm and domain specific tool boxes including optimization, statistics and much more.
- **Scikit-learn**—It is a collection of tools and algorithm for machine learning. It works with NumPy and SciPy and it is easy to implement machine learning models.
- **NLTK**-- NLTK is a leading platform for building Python programs to work with human language data.

MODEL/S DEVELOPMENT AND EVALUATION

• Identification of possible problem-solving approaches (methods)

- Before making the model we convert our text into vectors so for that we use technique known as **TF-IDF Vectorizer** .

```
# Importing the library and converting it into vectors
from sklearn.feature_extraction.text import TfidfVectorizer
tf_vec = TfidfVectorizer(max_features = 15000, stop_words='english')
```

- We know that this is classification problem so we use accuracy score, classification report and confusion matrix as our eveluation matrix. We also see the AUC score and also plot the AUC_ROC curve for our final model.
- As we know this dataset is imbalance so we don't too much focus on accuracy score . We see the precision and recall value along with f1_score.

- # Testing of Identified Approaches (Algorithms)

- As we know that this dataset is imbalance so first we see the result without doing any sampling and for that I use Logistic Regression with KFold cross validation and hyperparameter tuning.
- We also use Random Forest and XGBoost Classifiers as our evaluation model without using hyperparameter tuning because our dataset is too large and it takes more than hour to give the result.
- After that I use sampling technique and use different-different models with sampled training data and see the results.
- There are three sampling technique which I have used to handle the imbalance problem of our dataset.After using Sampling technique to balance our dataset and then we apply Random Forest classifier model and see the results.

- **Under Sampling:** By using the under sampling we are trying to reduce the points of maximum labels.

- **Over Sampling:** By using the over sampling we are trying to increase the points of minimum labels.

- **SMOTETomek:** SMOTETomek is a hybrid method which uses on under sampling method (Tomek) in with over sampling method (SMOTE).

- # Run and Evaluate selected models

Logistic Regression by using Cross Validation like KFold and Hyperparameter tunning:

```
#Importing the model Library
from sklearn.linear_model import LogisticRegression

#Importing error metrics
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
from sklearn.model_selection import GridSearchCV, KFold
```

```
log_class=LogisticRegression()
grid={'C':10.0**np.arange(-2,3),'penalty':['l1','l2']}
cv=KFold(n_splits=15,random_state=None,shuffle=False)
```

```
clf=GridSearchCV(log_class,grid,cv=cv,n_jobs=-1,scoring='f1_macro')
clf.fit(x_train,y_train)
```

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

Random Forest Classifer:

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier()
classifier.fit(x_train,y_train)
```

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

XGBoost Classifier:

```
from xgboost import XGBClassifier
xgb = XGBClassifier()
xgb.fit(x_train,y_train)
```

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

We saw that precision and recall is very good in XGBoost as compare to Logistic Regression and Random Forest.

Now we are use some sampling technique and use different-different models and saw the result.

Under Sampling: By using the under sampling we are trying to reduce the points of maximum labels.

```
from collections import Counter
Counter(y_train)
```

```
# Importing the Undersampling library
from imblearn.under_sampling import NearMiss

ns = NearMiss(.8)
x_train_ns,y_train_ns = ns.fit_sample(x_train,y_train)
print("The number of classes before fit {}".format(Counter(y_train)))
print("The number of classes after fit {}".format(Counter(y_train_ns)))
```

```
The number of classes before fit Counter({0: 100342, 1: 11357})
The number of classes after fit Counter({0: 14196, 1: 11357})
```

Over Sampling: By using the over sampling we are trying to increase the points of minimum labels**.**

```
# Importing the Oversampling library
from imblearn.over_sampling import RandomOverSampler
```

```
os = RandomOverSampler(0.75)
x_train_os,y_train_os = os.fit_sample(x_train,y_train)
print("The number of classes before fit {}".format(Counter(y_train)))
print("The number of classes after fit {}".format(Counter(y_train_os)))
```

```
The number of classes before fit Counter({0: 107509, 1: 12169})
The number of classes after fit Counter({0: 107509, 1: 80631})
```

SMOTETomek: It is a method of imblearn. SMOTETomek is a hybrid method which uses on under sampling method (Tomek) in with over sampling method (SMOTE).

```
from imblearn.combine import SMOTETomek
```

```
st = SMOTETomek(0.75)
x_train_st,y_train_st = st.fit_sample(x_train,y_train)
print("The number of classes before fit {}".format(Counter(y_train)))
print("The number of classes after fit {}".format(Counter(y_train_st)))
```

```
The number of classes before fit Counter({0: 100342, 1: 11357})
The number of classes after fit Counter({0: 100328, 1: 75242})
```

From above all results we see that under sampled model gives very poor result. So don't use under sampling unless, untill your dataset is very samll.

XGBoost model using over Sampled data gives the best result among all the sampling models so we keep this model as our final model.

- # Key Metrics for success in solving problem under consideration

Although it is a classification problem so we use accuracy score, classification report and confusion matrix as our evaluation metrics along with AUC ROC score.

As we know that our dataset is imbalance so we do not much focus upon the accuracy score. We mainly see the precision and recall value of our model.

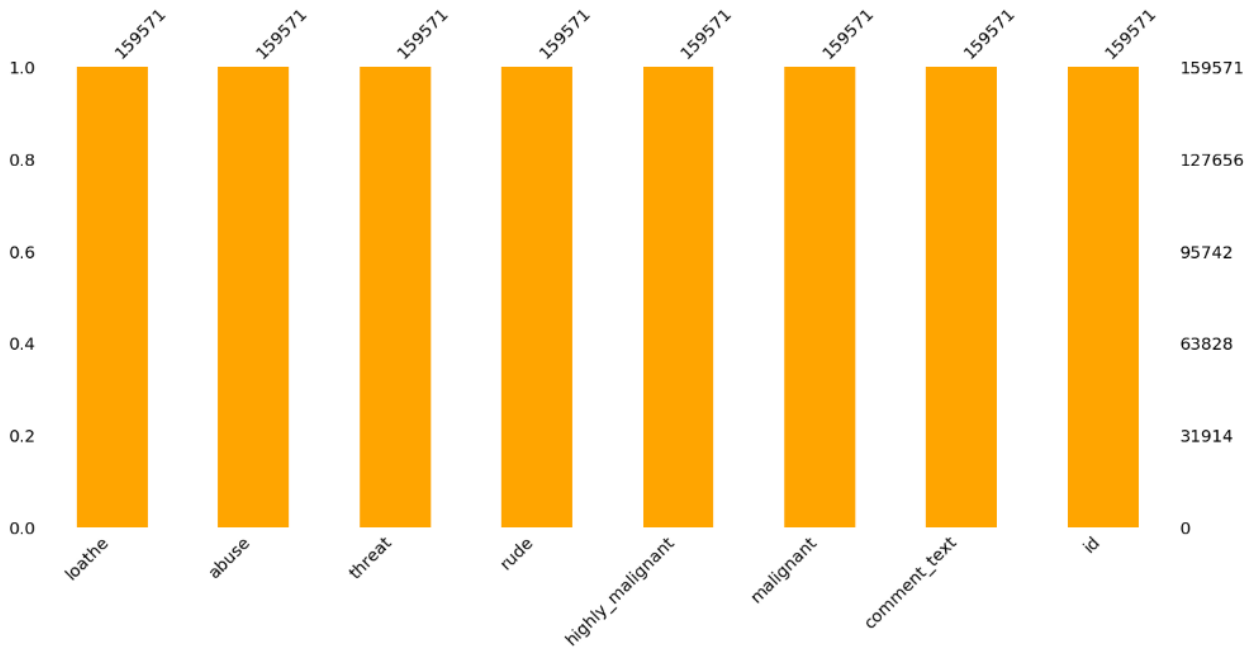
Precision talks about all the correct predictions out of total positive predictions. Recall means how many individuals were classified correctly out of all the actual positive individuals.

Along that we also consider AUC ROC score as our key metrix. Because Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

- **Visualizations**

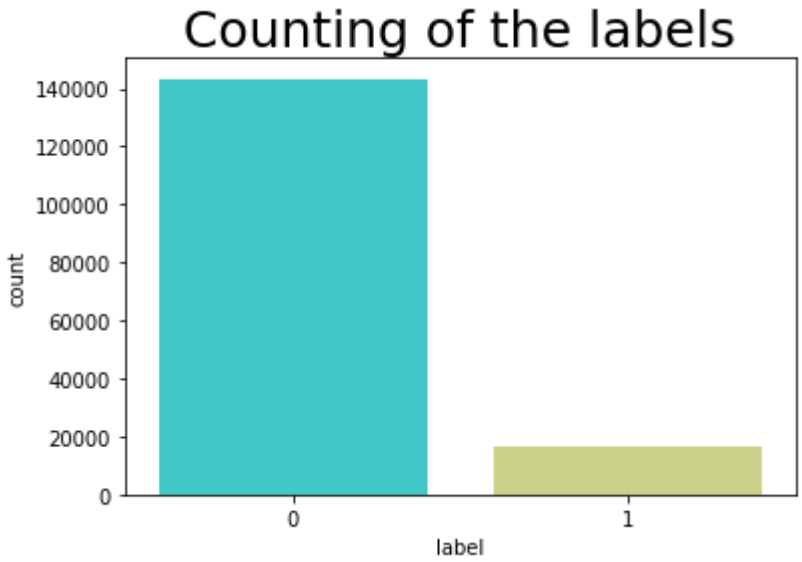
It is the graphical representation of data that is used to check about the presence of outliers, patterns, distribution of the data, etc. There are different data visualisation libraries in python that include matplotlib, seaborn, etc. We will make use of the seaborn and matplotlibtolibrary to visualise the dataset.

Plotting the barplot of the null values:



There are no null value present in our dataset.

Checking the count of labels:

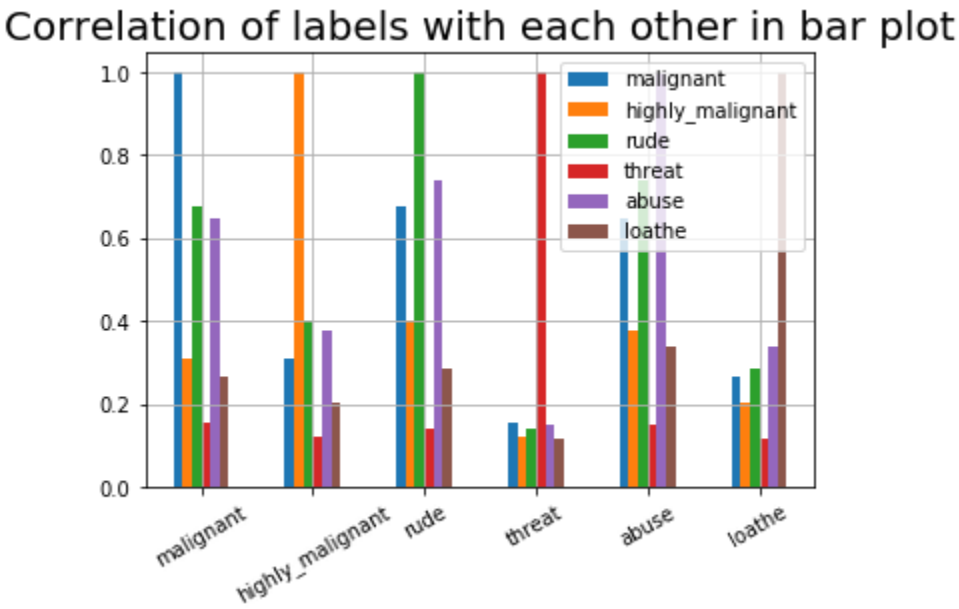


Label 1 indicates malignant Comments and label 0 indicates that benign comment.

1 143346
0 16225

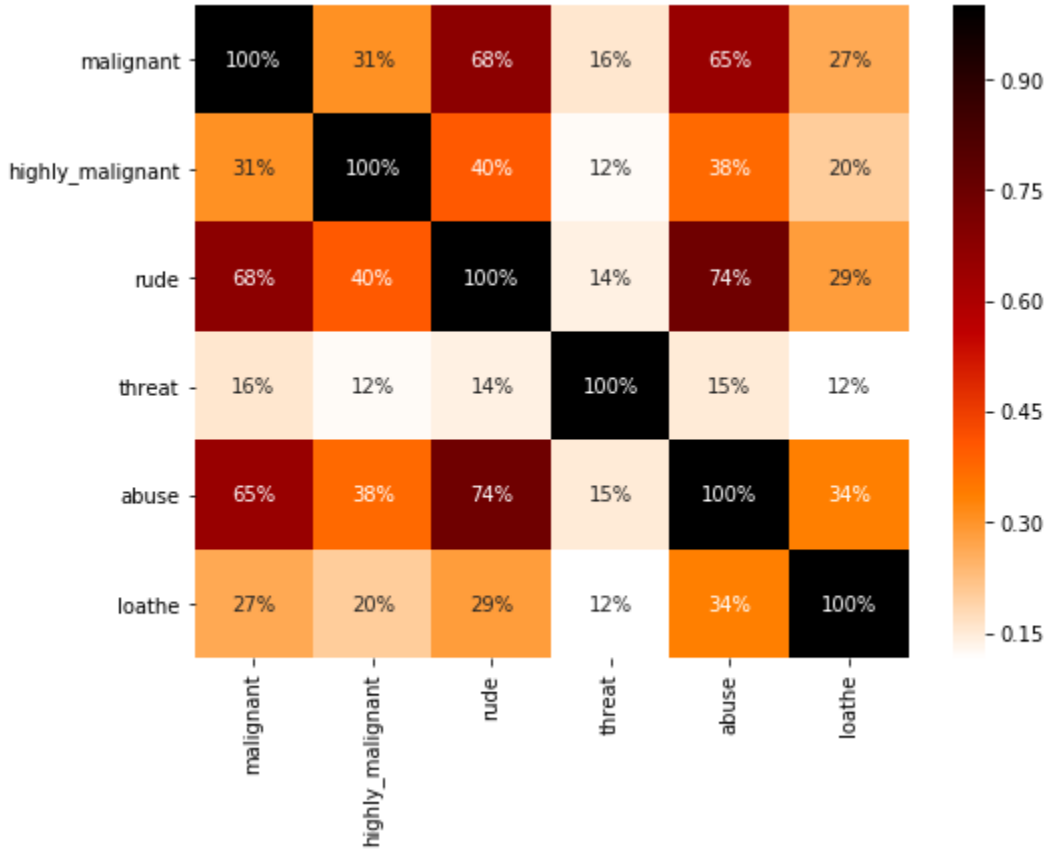
89% of the sample are in benign category and 11% of the sample is in malignant category. We will take this into account when splitting the data into a training and test set. We also set a seed to make this blog reproducible.

Correlation of target features:



In the above graph we can see the correlation of the target features with respect to each other.

Correlation Matrix:



Getting sense of loud words for Malignant Comments in label column:

Getting sense of loud words for Benign Comments in label column:

Getting sense of loud words which are offensive for loathe label:

]

Getting sense of loud words which are offensive for abuse label:

We can also see the others labels as same way.

- # Interpretation of the Results

From the above interpretation we come to know that this is classification based problem so we have learned to build a complete machine learning model for classification based problem.

The goal of any machine learning problem is to find a single model that will best predict our wanted outcome. Rather than making one model and hoping this model is the best/most accurate predictor we can

make, ensemble methods take a myriad of models into account, and average those models to produce one final model..

We also visualize the data and see the outcomes of our result that what percentage of comments are in malignant category and also plot the wordcloud to see the loud words for malignant and benign comments.

The AUC Score,f1-score and recall value is high when we use Logistic Regression with over sampling data. So we choose Logistic Regression model with over sampled data as our final model.

CONCLUSION

- **Key Findings and Conclusions of the Study**
- There are no null values in the dataset.
- The dataset is imbalanced. Label '1' has approximately 11% records, while, label '0' has approximately 89% records.
- ID and Comment_text feature is only object datatype and rest all the features are integer datatype.
- There are six labels in this dataset. So we merge them and make a single target feature.
- Sampling data gives the better precision and recall value along with auc score.

- Under sampled model gives very poor result. So don't use under sampling unless, untill our dataset is very samll.

• Learning Outcomes of the Study in respect of Data Science

- In this project we learn how to build a machine learning model for classification based problem.
- We also learn how to handle the imbalanced dataset for machine learning model. Because when we over sampling and use this over sampled data to build a ML model then it gives the better result.
- The goal of any machine learning problem is to find a single model that will best predict our wanted outcome. Rather than making one model and hoping this model is the best/most accurate predictor we can make, ensemble methods take a myriad of models into account, and average those models to produce one final model.

• Limitations of this work and Scope for Future Work

In this project the sample data is provided from our client database. In this project, we addressed the task of automatic identification of Malignant Comments. We introduced new malignant comment dataset, obtained through crowdsourcing . We developed classification models as well features representing text readability properties. Our best performing models achieved accuracies that are comparable to human ability to spot fake content.

I would conclude the project report by hoping that now you have understood every step that is required to be done to build a machine learning model. We have built the classification model for classifying the labels that is which comment is in label 1 that is malignant comments and which one is in label 0 that is benign comments and then evaluated it using different error metrics.