MALIGNANT COMMENTS CLASSIFICATION

Submitted by:

Neha Chand

ACKNOWLEDGMENT

In this project the dataset is given by the company and the project is done on the Jupyter Notebook in Anaconda.

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.

- ## Conceptual Background of the Domain Problem

  Monitoring of words written on social platform which can be reported and reduced for clean and better social media and interned

- ## Review of Literature

  We are going to build 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

- ## Motivation for the Problem Undertaken

  As cyberbullying, hatred speech, giving threat to someone is easy nowadays, while other can handle such situations others often takes stress which further reaches to a Mantel Disorder. By controlling the above social issues, we can create a clean and better social platform for individuals

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem
    - Multi-label classification originated from the investigation of text categorisation problem, where each document may belong to several predefined topics simultaneously.
    - Multi-label classification of textual data is an important problem. Examples range from news articles to emails. For instance, this can be employed to find the genres that a movie belongs to, based on the summary of its plot.

- Data Sources and their formats

  The sample data is provided is provided by the client database, and the data is in the form of csv i.e. in the excel sheet. Here is the dataset.

- Data Preprocessing Done

  In the Data Preprocessing, At first we drop the unnecessary variables or the features which are not related to the further processes, after this we remove those feature which are very highly correlated to each other. And then we divide the dataset into dependent and independent sets. After this we scaled our dataset but not the target variable by the Normalizer method and then we select top 25 most import features as the data is very huge and this is done by the sklearn library by importing RFE Recursive feature elimination and based on the ranking of top 25 we select the top 25 columns for the process. And after that outliers were handled with the help of zscore method and then Undersampling is done as dataset is class imbalance and then splitting the data for Training and Testing.

- Data Inputs- Logic- Output Relationships

By the visualization of the data we see the relationship between the features and also with the correlation tells about the all features related to the Target variable. By the correlation method we can see the relationship and effects of features on the target variable. And also we see the correlation between the features and we deleted those columns which are highly correlated with each other as both of them carry the same type of data.

- Hardware and Software Requirements and Tools Used

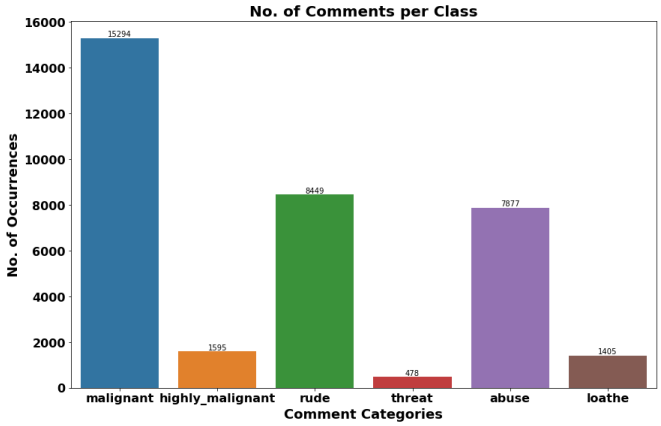
  The Project is done on the Window 10, here we use the Software Anaconda platform (Python 3.8.5 64 bit) and the project is done on the Jupyter Notebook where we run different python libraries such as

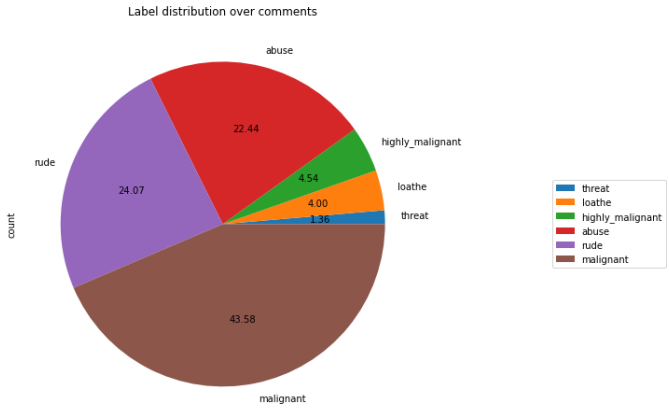
Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Here we first check the data is supervised or unsupervised by checking the target column and then when we see the output it shows it's a classification problem. Then we split the data into training and testing and then we trained the data with different model using various classifiers (logistic Regression, Random Forest Classifier, Support Vector Classifier, Ada Boost Clasifier) and we check the accuracy score, classification report and confusion matrix and then we check the aucroc curve and score for the better prediction.

Exploration of Target Variable Ratings





- Testing of Identified Approaches (Algorithms)

The different classification algorithm used in this project to build ML model are as below:

- Random Forest classifier
- Support Vector Classifier
- Logistics Regression
- AdaBoost Classifier

- Run and evaluate selected models

  Running the four classifier and selecting the best modal according to accuracy score

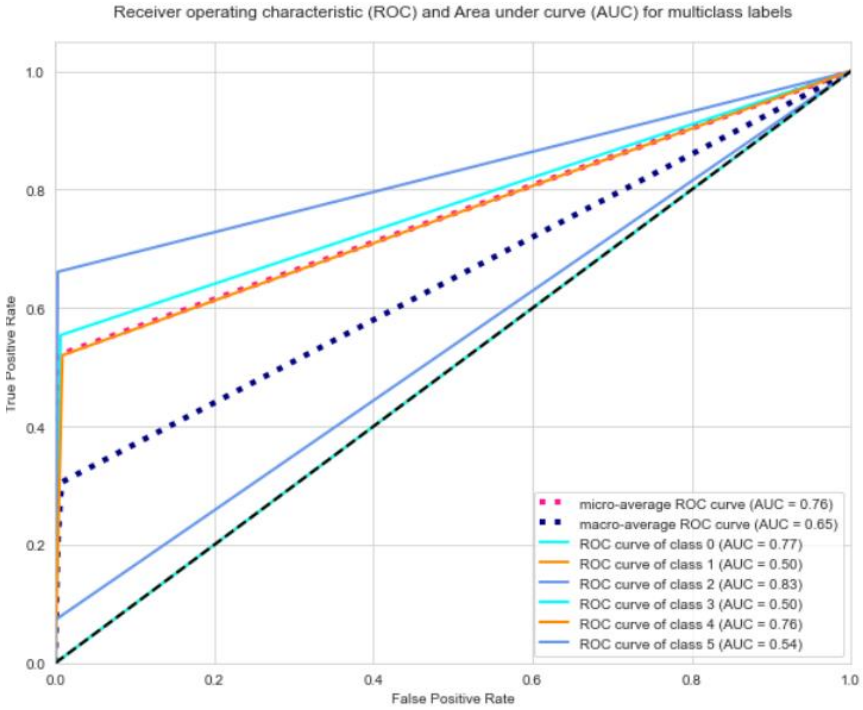
In [50]:

```
# Preparing the list of models for classification purpose
models = {
          "Logistic Regression": {"name": LogisticRegression()},
          "Random Forest Classifier": {"name": RandomForestClassifier()},
          "Support Vector Classifier": {"name": LinearSVC(max_iter = 3000)},
          "Ada Boost Classifier": {"name": AdaBoostClassifier()},
         }

# Taking one forth of the total data for training and testing purpose
half = len(df)//4
trained_models = build_models(models,X[:half,:],Y[:half,:])
```

```
================================================================================
Current Model in Progress: Support Vector Classifier
================================================================================
Training:  BinaryRelevance(classifier=LinearSVC(max_iter=3000), require_dense=[True, True])
Testing:

        Hamming Loss  : 0.02124319534118243
        Accuracy Score: 0.9109760729206229
              precision    recall  f1-score   support

           0       0.84      0.58      0.69      1281
           1       0.51      0.19      0.27       150
           2       0.91      0.65      0.76       724
           3       0.70      0.32      0.44        44
           4       0.76      0.52      0.62       650
           5       0.71      0.20      0.31       109

   micro avg       0.83      0.55      0.66      2958
   macro avg       0.74      0.41      0.51      2958
weighted avg       0.82      0.55      0.65      2958
 samples avg       0.05      0.05      0.05      2958
Completed in [11.988146070999846 sec.]
================================================================================
```

- Key Metrics for success in solving problem under consideration

Here we use the Aucroc curve and score for the final prediction.



- Visualizations
  - Word Cloud is a visualization technique for text data wherein each word is picturized with its importance in the context or its frequency.
  - The more commonly the term appears within the text being analysed, the larger the word appears in the image generated.
  - The enlarged texts are the greatest number of words used there and small texts are the smaller number of words used.

- Interpretation of the Results

  Final Model is giving us Accuracy score of 91.26% which is slightly improved compare to earlier Accuracy score

Final Model

```
Final_Model = OneVsRestClassifier(LinearSVC(loss='hinge',
            multi_class='ovr', penalty='l2', random_state=42))

Classifier = Final_Model.fit(x_train, y_train)
fmod_pred = Final_Model.predict(x_test)
fmod_acc = (accuracy_score(y_test, fmod_pred))*100
print("Accuracy score for the Best Model is:", fmod_acc)
h_loss = hamming_loss(y_test,fmod_pred)*100
print("Hamming loss for the Best Model is:", h_loss)
```

```
Accuracy score for the Best Model is: 91.26002673796792
Hamming loss for the Best Model is: 2.0819407308377897
```

CONCLUSION

- Key Findings and Conclusions of the Study

  The Maximum feature used while vectorization is 2000. Employing more feature in vectorization lead to more accurate model which I not able to employed due computational resources.

- Learning Outcomes of the Study in respect of Data Science

  Data is imbalanced in nature but due to computational limitation we have not employed balancing techniques here.

- Limitations of this work and Scope for Future Work

  Deep learning CNN, ANN can be employed to create more accurate model.