University of Sunderland

[Cyberbully Detection Model]

[By Victor Shimane]

Table of Contents

Introduction

CET313 Artificial Intelligence Module's aim and desired outcome is to produce knowledgeable students with a wide range of artificial intelligence skills to help combat societal problem and improve the quality of living as well as to contribute to the technology community. To emphasize this, students are tasks with a real-life problem to test, experiment and solve using the skills acquired from attending CET313 Artificial Intelligence Module. Th desired output is an intelligent prototype solving a social existing problem from a given research area fields that is path finding and machine learning.

E-portfolio Justification

The problem in the study, which is cyberbully, is addressed using Natural Language Processing under Machine learning through a developed prototype which will detect and rate social media post and comments from non-bully to bully comment or post. The decision to pursue and investigate this field of research was encouraged by the vast knowledge and confidence gained during AI lessons, tutorial and during my own individual research. During that time, the author discovered and was fascinated by how machine learning techniques can help computers mimic the human intelligence with the sole purpose of improving the quality of life through solving everyday problem, with these techniques evolving and improving over the years they are becoming more efficient and accurate. Some of the skills gained during the learning period are:

Artificial Intelligence:

The knowledge gained from the module raised the awareness on how AI has not only gained popularity in the last two decades but how it is integrated and solving on day-to-day activity around us, improving the quality of living. For instance, how now through computer vision cancer can be detected through image recognition giving more accurate and precise results or the use of self-driving cars and manufacturing robots.

Machine Learning:

This is a subset of Artificial intelligence which allows computers to mimic human intelligence using historical data or experience, it uses input to predict output. This allows systems and models to learn from data by analyzing and identifying patterns on data.

Natural Language Processing:

The AI class tutorial four and five which were under Natural Language processing which included the installation of Natural Language Tool Kit (NLTK) packages, text preprocessing by removing stop words, punctuation, normalizing text, Unicode characters and stemming and lemmatization, and text sentiment analysis which influenced the decision to choose NLP. Sentiment analysis is contextual mining of text to identify, extract, quantify and study subjective information and output if subject at study is positive, negative, or neutral. With the knowledge gained from class and research Natural language proved to be the best approach for cyberbully detection model because with pattern recognition and historical data analysis from machine learning integrated with sentimental analysis to distinguish bad bully sentiment from good neural and non-bully sentiments from social media post and comment.

Literature Review on Prototype Identification

Problem Scope

Internet connectivity has in the last decade proved to be a double-edged sword. As significant as it is to prove both educational and social benefits to people through improved communication and democratized access of information in platforms like social media which allow users to share data like text, picture, videos etc with the purpose of socializing and meeting new people. With all the good the Internet has brought it has also led to the mushrooming of anonymity and concealed individuals who serve to stalk, bully, harass and mistreat let alone threaten others online. Cyber-bullying is the type of harassment that occurs through the internet online.

According to the findings of (Zois et al., 2018) the evidence of cyber bullying on teenagers is staggering with over 50% of teenagers either have been engaged in or have been a victim of cyber bullying with a 10 to 20% experience rate on daily basis. The research showed the impact exerted by cyber bullies can lead to suicide, psychological suffering, and isolation, learning difficulties and escalated physical fights.

The Northon Cybercrime predicted that 1 million people are victims of cybercrime globally with the National Crime Prevention predicting that about 50% of children having experienced online harassment while in the United Kingdom insights from statistics outline that 1 in 5 teens are victims of cybercrime (A com). Research has proved and indicated that cyberbully is increasing at an alarming rate globally.

The impact of cyber bullying has also stretched enough to affect productivity at workplace, as investigated by (Kwanya et al., 2021) on which they discovered that although there is no standalone statistics which support the claim cyber bullying affecting the turnaround productivity at work place, it is evident that cyber bullying is a psychological violence which affect the wellbeing of employees which in-turn affect production, not only that it can poison an organization by undermining employees morale and eroding any sense of loyalty, trust or teamwork (Kwanya et al., 2021).

Research has demonstrated that being involved in cyber bullying, either being the perpetrator or victim, has psychological and emotional consequences. Victims are more likely to develop psychological pathology. Both the victims and perpetrators are likely to suffer from low self-esteem. However the greatest impact associated with cyber bullying to victims is that they are likely to suffer from severe depression which could lead to suicide and self-harm.(Watanabe, Bouazizi and Ohtsuki, 2018) The impact is not only limited to psychological impact but also physical as there is evidence that victims are more likely to weight loss or gain, headaches, abdominal pains and sleeping problems.(Watanabe, Bouazizi and Ohtsuki, 2018)

Proposed Solution

Cyber bullying has proved to be a global social problem which affects all age groups be it adolescents and or teenagers, middle age, and senior citizens hence there is need to address the problem. The author proposes an automated machine learning model which composes of semantic analysis, machine learning algorithms and classifiers like SVG, Random Forest, Naive Bayes, and logical regression to detect bully post and comments. The solution will achieve the best results through intensive model training for more precision and accuracy.

Current Solutions

Different solutions have been put in-place to address cyber bullying which could be classified into two thus Legislation and Filtering. This paper will be focused on filtering bully or hate speech recognition using machine learning.

(Dalvi, Baliram Chavan and Halbe, 2020) employed a language-based machine learning technique which filtered cyber bullying, on which they managed to have an accuracy of 83.2. They confirmed that their model indeed was able to identify cyber bully and acknowledged that there is room for development taking account of the severity of the problem at study.

Online programs which educate. Most social media use the 3 unfriend/follow, Block, and report system on which the victim I give a choice and power to detach themselves from the bully. With reporting the system automatically blocks the user then opens a case on which authorized personnel from social media investigates the case and could lead to temporary ban or permanent ban based on the magnitude of the hate speech.

(Akhter et al.,2019) used a multinomial Naive Bayes classifier to classify Facebook comments data with three main data categories, shimming, harassment, and races. The occurrence of the model at the end of the experiment was 88.76% and although the authors found the results satisfactory (Akhter et al.,2019) did emphasize on the model's accuracy's need for improvements for a more accurate and precise model.

Planning

Phase 1: Project Planning: 1.1 Requirements gathering and Project Scope
1.2 Project Feasibility
1.3 Project Schedule
1.4 Gantt chat planning

Phase 2 Data Processing2.1 Data Cleaning
2.2 Text preprocessing
2.3 Text quality validation
2.4 Text transformation

Phase 3 Model Training 3.1 Model and classifier fitting
3.2 Model Evaluation
3.3 Optimization and model tuning

Phase 4 Testing and evaluation 4.1 functionality and performance evaluation
4.2 Classifier comparison
4.3 conclusion
Phase 5 Deployment 5.1 Model and flask integration
5.2 Deployment of model

Reflection on prototype Identification

The pursue of this project has been insightful and informative as tit acted as a learning curve and gained new knowledge and skills about cyber bullying, natural language processing and machine learning. Choosing machine learning natural language processing has boosted the authors confidence due insights discovered during research for instance text sentimental analysis which divides and breaks down words, statements and phrases from paragraphs finding patterns and how the use of one relates to another which is helpful in the pursuit of finding a solution for cyber bully on which repetitive use phrases, words and statement is evident.

The background search for existing solutions which address this problem proved to be fruitful as it acts as a guide on the projection of the project, with the knowledge, experimental results, hypothesis, and conclusions from other research help in building a more accurate, precise, and efficient solution which address the problem.

The experience gained from class tutorials complements the chosen field and problem such that when combined with knowledge from literature review a more effective, enhanced, and accurate model or solution is achievable. This includes data preprocessing techniques and Data cleaning which deals with removing outliers, duplicates and corrupt data, machine learning algorithms like Random Forest, Support vector machine and Naive Bayer. Data transformation techniques which model the data into a more compatible state or comprehensive form though the use of text normalization, stemming and lemmatization, removal of noise in data.

This project has increased and raised the authors awareness on the selected problem, for at the beginning of the project the goal was to solve a social problem existing in the community but through research, it proves not only to be a community problem but global problem which is growing at an alarming rate. However, through the success of this project, the problem will be solved by curbing a global issue.

Prototype Development

Prototype Benchmark

To ensure the success of this project, the author decided to use a project that has achieved success which have been developed by machine learning and natural language expect as a reflection to benchmark with the aim to yield similar results of success. The project is Sentiment Analysis for marketing and was developed by Koosha Tahmasebipour who is a Principal ML Engineer. This project was chosen not only because of its success but also to avoid code plagiarism because the two projects are distinct although they follow under NLP.

The Github repository for Sentimental analysis for Marketing: https://github.com/koosha-t/Sentiment-Analysis-NLP-for-Marketting

Code development

Technical specification for developed Model.

The programming language used is python 3.10.1
Machine learning algorithm implemented is Logical Regression after the data was experimented upon on different LM algorithms with the best classifier selected.
Software used: Microsoft visual code.

Technical Specification for technology used.

Based on the magnitude of the project, all the development of the model were implemented on the author's personal computer with the specifications:

Processor: Intel® Core™ i7-8550U CPU @ 3.00GHz
RAM: 16 GB
GPU Radeon™ 530
System type: 64 bits
Storage: Samsung 520SSD

Deployment and use

Open Visual Code
Click File then open "select folder" then select the directory for the application.
Open app.py on the left navigation showing contents of the selected folder.
Run the code.
If it does not open the browser automatically then click CTRL + URL on the terminal

Data Cleaning and Exploration

The origin of the data is from Kaggle, the machine learning and data science community. The dataset includes 47692 statements to study which composed of 70 percent of ham and 30% being bully text.

During data cleaning, there was removal of corrupt data which included incomplete and missing values. Removal of duplicate values is to ensure all values are distinct and unique to maximize the best performance and obtain the most accurate model.

```
In [14]: STOPWORDS = set(stopwords.words('english'))
STOPWORDS.update(['rt', 'mkr', 'didn', 'bc', 'n', 'm',
                  'im', 'll', 'y', 've', 'u', 'ur', 'don',
                  'p', 't', 's', 'aren', 'kp', 'o', 'kat',
                  'de', 're', 'amp', 'will', 'wa', 'e', 'like'])
stemmer = SnowballStemmer('english')
def clean_text(text):

    # Remove Hashtag, Mention, URLs
    pattern = re.compile(r"(#[A-Za-z0-9]+|@[A-Za-z0-9]+|https?://\S+|www\.\S+|\S+\.[a-z]+|RT @)")
    text = pattern.sub('', text)
    text = " ".join(text.split())

    # Make all text lowercase
    text = text.lower()

    # Stemming
    text = " ".join([stemmer.stem(word) for word in text.split()])

    # Removing Punctuations
    remove_punc = re.compile(r"[%s]" % re.escape(string.punctuation))
    text = remove_punc.sub('', text)

    # Removing stopwords
    text = " ".join([word for word in str(text).split() if word not in STOPWORDS])

    # Taking care of emojis
    emoji = demoji.findall(text)
    for emot in emoji:
        text = re.sub(r"(%s)" % (emot), "_".join(emoji[emot].split()), text)

    return text
```

Figure 1 data cleaning

Figure 1 shows code spinet of a function which dealt with removing punctuation on post to maintain consistency for every word used, removing characters and transforming every letter to lowercase, removed emojis and stop words as part of data cleaning.

Text preprocessing

On this sage the author prepared the data for model fitting. This was archived through developing a function which removed punctuation, removing words written combined with numbers, lowered the case of characters and text normalization. This helps the machine learning algorithms to understand the datasets by ensuring there is consistency with the wording structure and character presentation. For instance, the same word, one in lower case and the other in uppercase are treated differently by the machine although used to imply the same logic.

Words identified as stop words which are commonly occurring words which in-turn do not add value to building an accurate model were removed. Text stemming and lemmatization, which is crucial, were implemented. Stemming which deals with shortening a word to its stem word to help normalize the text and the author used Poter Stemmer for its efficiency.

While lemmatization reduced words into their lamma thus groping different forms of the same word which is crucial in maintaining accuracy and recognizing the same word can have several meaning and be used to imply different logic as evident in everyday conversations.

Transformation

The author converted dataset into a more understandable and compatible state with the model using sklearn TfidfVectorizer.

```
In [32]: from sklearn.preprocessing import StandardScaler
         scaler = StandardScaler()
         tfidf_array_train = X_train_tfidf.toarray()    # Converting the sparse matrix to a numpy array (dense matrix)
         tfidf_array_test = X_test_tfidf.toarray()      # Converting the sparse matrix to a numpy array (dense matrix)
         scaled_X_train = scaler.fit_transform(tfidf_array_train)  # Fitting on only training data to avoid data leakage from test data
         scaled_X_test = scaler.transform(tfidf_array_test) # and then tranforming both training and testing data
```

Figure 2 Transformation

```
In [25]: from sklearn.model_selection import train_test_split
         from sklearn.feature_extraction.text import TfidfVectorizer

In [26]: X = df['cleaned_text']  # Feature (raw data)
         y = df['cyberbullying_type']  # Target Label

In [27]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.1, random_state = 42)
         # Performing the train|test split. This test set is essentially a hold out test set as we'll be performing Cross Validation
         # using Grid Search which will split our training data into a training and validation split

In [28]: tfidf = TfidfVectorizer(max_features = 5000)  # Using the TF - IDF Vectorizer to extract top 5000 most important features
         # from the text data

In [29]: # Feature Extraction
         X_train_tfidf = tfidf.fit_transform(X_train)  # Creating the vocabulary only from the training set to avoid data leakage from
         X_test_tfidf = tfidf.transform(X_test)        # the test set.

In [30]: X_train_tfidf  # Sparse Matrix is created to save memory since many values are close to 0

Out[30]: <35043x5000 sparse matrix of type '<class 'numpy.float64'>'
                 with 403374 stored elements in Compressed Sparse Row format>
```

Figure 3 splitting the dataset.

Model Design

Different ML machine learning algorithms were experimented on the data.

Logical Regression

logistic regression model is a statistical approach for binary classification that may be applied to multiclass classification. Scikit-learn includes a highly efficient logistic regression implementation that can handle multiclass classification tasks. Two versions of logical regression were used, the first one was basic logical regression which were exposed to a reduced 90% of the data. The second improved logical regression which were trained and evaluated to meet the demands of the project used SK Learn HalvingGridSearchCV.

Evaluation

Upon evaluation of the model based on the results after experimenting, the model developed addressed and was able to output the desired results. The model distinguished bully hate speech from non-bully speech.

Model Training and Performance Evaluation

The first version of the logical regression was exposed and fitted to the test data with the ratio of 1:4 as testing data and training data respectively. The model was able to study them and find patterns existing in data as it is evident with the decent results from the experiment. The results are displayed in the figure below.

```
In [42]: # LOGISTIC REGRESSION with the the 90% variance data
from sklearn.linear_model import LogisticRegression
log_model_pca = LogisticRegression()
log_model_pca.fit(reduced_90, y_train)
preds_log_model_pca = log_model_pca.predict(reduced_90_test)
print(classification_report(y_test, preds_log_model_pca))
plot_confusion_matrix(log_model_pca, reduced_90_test, y_test)
```

```
                   precision    recall  f1-score   support

              age       0.86      0.83      0.85       766
        ethnicity       0.90      0.85      0.87       801
           gender       0.77      0.79      0.78       788
not_cyberbullying       0.65      0.67      0.66       783
         religion       0.84      0.86      0.85       756

         accuracy                           0.80      3894
        macro avg       0.80      0.80      0.80      3894
     weighted avg       0.80      0.80      0.80      3894
```

Figure 4 Model Evaluation Matrix

As displayed in figure 4 the results from the first version were good with an accuracy of 80%. Although this means the model has the capability to differentiate between hate and non-hate speech, this is not good enough to be used on real life situation as there is a lot of room for error.

Model Enhancements

The results from the first version of logical regression showed that there is room for improvement, there further model optimization was crucial to address cyber bullying hence now the model was exposed to 100% of data. The model incorporated SK Learn HalvingGridSearchCV which features searching over specific parameters with successive halving. It evaluates all the candidates with specific small amounts of resources and interactively selects the most suitable candidate using more and more resources. There are other approaches which could have been implemented as opposed to using the parameter search HalvingGridSearchCV which includes the Hyper-parameter which deals with model optimization however the model needed precision and accuracy improvement which HalvingGridSearchCV proved to be effective on.

```
In [39]: # LOGISTIC REGRESSION with the complete data
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import HalvingGridSearchCV
log_model = LogisticRegression(solver = 'saga')
param_grid = {'C': np.logspace(0, 10, 5)}
grid_log_model = HalvingGridSearchCV(log_model, param_grid = param_grid, n_jobs = -1, min_resources = 'exhaust', factor = 3)
grid_log_model.fit(X_train_tfidf, y_train)
preds_grid_log_model = grid_log_model.predict(X_test_tfidf)
print(classification_report(y_test, preds_grid_log_model))
plot_confusion_matrix(grid_log_model, X_test_tfidf, y_test)
```

```
              precision    recall  f1-score   support
```

Figure 5 Enhanced Model

Results Analysis

Figure 6 shows the results of the experiment after implementing the SK Learn HalvingGridSearchCV to the model. That approach proved to be fruitful as the accuracy of the model improved by 12% from a total accuracy of 80% to 92% therefore a step closer to solving the problem at hand. With these results this means the model can be tested for more assurance than if successfully deployed.

In [39]:
```
# LOGISTIC REGRESSION with the complete data
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import HalvingGridSearchCV
log_model = LogisticRegression(solver = 'saga')
param_grid = {'C': np.logspace(0, 10, 5)}
grid_log_model = HalvingGridSearchCV(log_model, param_grid = param_grid, n_jobs = -1, min_resources = 'exhaust', factor = 3)
grid_log_model.fit(X_train_tfidf, y_train)
preds_grid_log_model = grid_log_model.predict(X_test_tfidf)
print(classification_report(y_test, preds_grid_log_model))
plot_confusion_matrix(grid_log_model, X_test_tfidf, y_test)
```

```
                   precision    recall  f1-score   support

              age       0.96      0.97      0.96       766
        ethnicity       0.98      0.98      0.98       801
           gender       0.92      0.84      0.88       788
not_cyberbullying       0.80      0.85      0.82       783
         religion       0.94      0.96      0.95       756

         accuracy                           0.92      3894
        macro avg       0.92      0.92      0.92      3894
     weighted avg       0.92      0.92      0.92      3894
```

Figure 6 Results

Figure 7 Model Evaluation Matrix

Other Machine Learning Classifiers used.

Naive Bayes

This classifier used a multinomialNB on which probable classes were set out by its feature vector. The Naive Bayer model did good although evident that there is room for improvement the model had an accuracy of 84% when means it is capable of classifying hate speech from non-hate speech but compared to other models on the experiment, the Naive Bayes is the least performing model.

```
In [49]: # NAIVE - BAYES
from sklearn.naive_bayes import MultinomialNB
nb_model = MultinomialNB()
nb_model.fit(X_train_tfidf, y_train)
preds_nb_model = nb_model.predict(X_test_tfidf)
print(classification_report(y_test, preds_nb_model))
plot_confusion_matrix(nb_model, X_test_tfidf, y_test)
```

```
                   precision    recall  f1-score   support

              age       0.81      0.97      0.88       766
        ethnicity       0.88      0.91      0.90       801
           gender       0.86      0.80      0.83       788
not_cyberbullying       0.83      0.55      0.66       783
         religion       0.81      0.96      0.88       756

         accuracy                           0.84      3894
        macro avg       0.84      0.84      0.83      3894
     weighted avg       0.84      0.84      0.83      3894
```

Support Vector machine

SVG plots values in an N-dimensional space matching each value distinctly with a unique coordinate. With SVG, the model developed was precise and accurate with 92 percent, making it second best model developed. This model can be used and tested in a real-life environment.

```
In [45]: # SUPPORT VECTOR MACHINES
from sklearn.svm import LinearSVC
svm_model = LinearSVC()
C = [1e-5, 1e-4, 1e-2, 1e-1, 1]
param_grid = {'C': C}
grid_svm_model = HalvingGridSearchCV(svm_model, param_grid = param_grid, n_jobs = -1, min_resources = 'exhaust', factor = 3)
grid_svm_model.fit(X_train_tfidf, y_train)
preds_grid_svm_model = grid_svm_model.predict(X_test_tfidf)
print(classification_report(y_test, preds_grid_svm_model))
plot_confusion_matrix(grid_svm_model, X_test_tfidf, y_test)
```

```
                   precision    recall  f1-score   support

              age       0.94      0.98      0.96       766
        ethnicity       0.97      0.98      0.98       801
           gender       0.94      0.81      0.87       788
not_cyberbullying       0.79      0.85      0.82       783
         religion       0.95      0.96      0.96       756

         accuracy                           0.92      3894
        macro avg       0.92      0.92      0.92      3894
     weighted avg       0.92      0.92      0.92      3894
```

Classifier's results Evaluation

Logical Regression is has proved to be the most accurate model in this experiment, with an accuracy of 92 however the SVG model did indeed have the same total accuracy of 92%, the decision to choose logical regression over SVG was influenced by the model precision over none hate speech parameter on which SVG had 79% while logical regression had 80%. The results were close. The naive Bayer was the least performing model of all the 4 models.

Validity of the Experiment

The development of this project observed and upheld pure work ethics and science ethics to provide an equal and just environment for all machine learning environment. The author ensured the best optimal setting for the best performance in solving hate speech. All the classifiers were exposed to the data that were processed and transformed in the same manner. All machine learning classifier's experiments were conducted on the same computer with the same technical specification. All the measures taken were to eliminate bias and giving one classifier an advantage over the other therefore ensuring the results obtained on this experiment just and fair which can be seen, analyzed and evaluated because every procedure taken and results have been recorded for future evaluation.

Conclusion

This paper highlights the journey taken to solve hate speech on social platforms through machine learning. It explains the need to investigate this area of research and the benefit it would bring if the problem is addressed, how the problem was addressed using Natural language processing and the evaluation of the solution.

During development of the model, the first approach was to use basic simple regression but due to poor results obtained during the experiment, Logical regression with SK Learn HalvingGridSearchCV was implemented and yielded the desired output. The initial problem with the first logical regression was because the model could not train and predict the parameters corrected therefore the SK Learn HalvingGridSearchCV focuses on searching over specific parameters with successive halving.

The most challenging part of the project was addressing errors and ensuring that all classifies used the same data transformation because having to transform data for each model increases code redundancy hence the need to do research in which all models could use the same transformation without sacrificing the accuracy of every model.

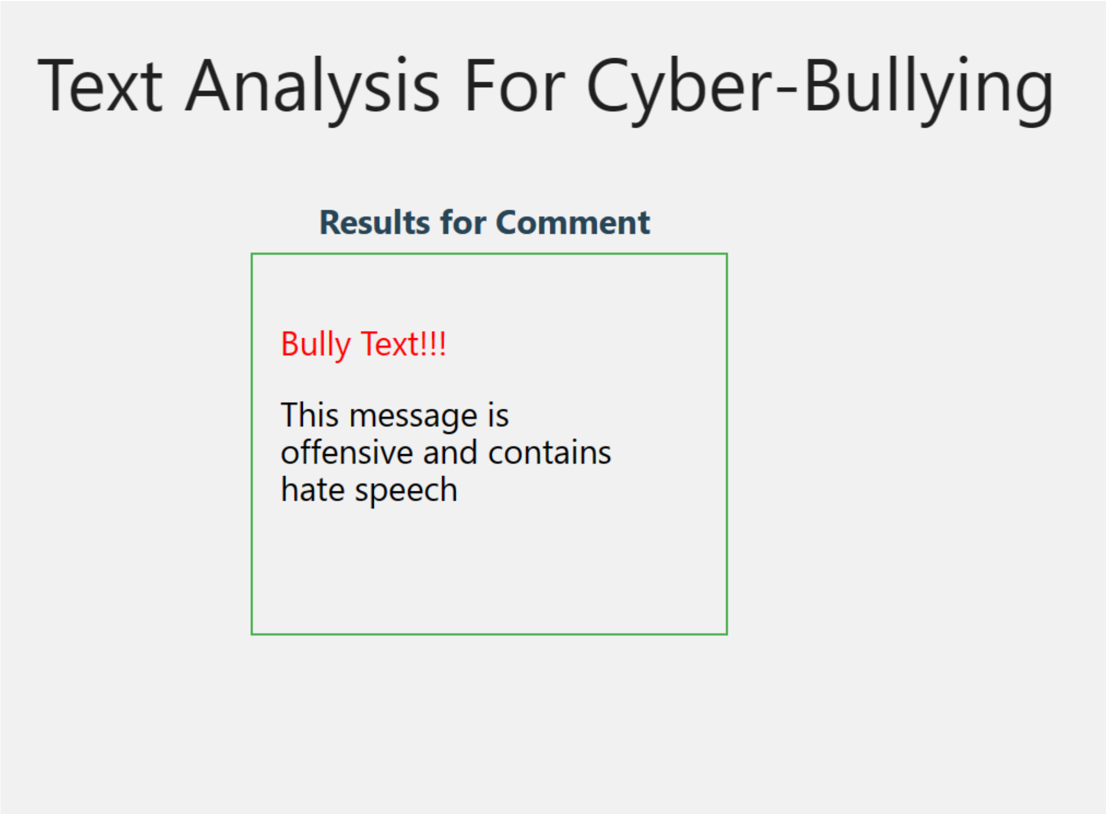
Due to the level of research done during the development of this project the author gained a priceless amount of knowledge which could later be used on different machine learning projects. The literature review proved to be engaging as the author read through several articles about cyber bullying, how currently it is being addressed and research on natural language processing hence the great full appreciation to all the research materials that made this project a success.

Index

Data Exploration: checking for missing values, duplicates and white spaces.

```
In [17]: df.isnull().sum()  # Checking for missing values

Out[17]: tweet_text            0
         cyberbullying_type    0
         cleaned_text          0
         dtype: int64

In [18]: df['cleaned_text'].duplicated().sum() # Checking for duplicate values

Out[18]: 2887

In [19]: df.drop_duplicates("cleaned_text", inplace = True)

In [20]: df['cleaned_text'].str.isspace().sum()     # Checking for tweets with only whitespaces

Out[20]: 0
```

Model's GUI



References

Dalvi, R.R., Baliram Chavan, S. and Halbe, A. (2020) 'Detecting Twitter Cyberbullying Using Machine Learning', Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020, (Iciccs), pp. 297–301. doi:10.1109/ICICCS48265.2020.9120893.

Kwanya, T. et al. (2021) 'Cyber-bullying research in Kenya: a meta-analysis', Global Knowledge, Memory, and Communication [Preprint]. doi:10.1108/GKMC-08-2020-0124.

Online, H.T. et al. (2020) 'Multi-Authority Attribute Based Data Access Control in Fog Computing', IEEE Access, 8(May), pp. 83986–83999. doi:10.1109/ACCESS.2020.2992460.

Watanabe, H., Bouazizi, M. and Ohtsuki, T. (2018) 'Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection', IEEE Access, 6, pp. 13825–13835. doi:10.1109/ACCESS.2018.2806394.

Win, Y. (2019) 'Classification using Support Vector Machine to Detect Cyberbullying in Social Media for Myanmar Language', 2019 IEEE International Conference on Consumer Electronics - Asia, ICCE-Asia 2019, pp. 122–125. doi:10.1109/ICCE-Asia46551.2019.8942212.

Zois, D.S. et al. (2018) 'Optimal Online Cyberbullying Detection', ICASSP, IEEE International Conference on Acoustics, Speech, and Signal Processing - Proceedings, 2018-April, pp. 2017–2021. doi:10.1109/ICASSP.2018.8462092.

E-Portfolio Link:

Below is a link to my e-portfolio which contains my practical work, tutorials and exercises available on canvas.

https://canvas.sunderland.ac.uk/eportfolios/9467?verifier=2ZmkcSA8gRsU8Tooltvl7iPsUbKMSwf5Sxv8bhCJ