UOW MALAYSIA KDU
PART OF THE UNIVERSITY OF WOLLONGONG AUSTRALIA GLOBAL NETWORK

SCHOOL OF COMPUTING & CREATIVE MEDIA

Please tick ✓ or click if using MS WORD

☐FOUNDATION ☐DIPLOMA ☒ DEGREE ☐ MASTER

Assignment Coversheet

Please complete all details required clearly. For softcopy submissions, please ensure this cover sheet is included at the start of your document or in the file folder.

Assignment & Course Details:

Subject Code: *(e.g. XCAT1234)*	**Subject Name** *(e.g. Fundamentals of Computing)***:**
XBDS 2034	Data Science Toolbox

Course *(e.g. Bachelor in Computing)* :

Bachelor of Computer Science (Hons)

Lecturer Name:

Dr Law Foong Li

Assessment Due Date: *(dd/mm/yy)*	07/04/21	**Assessment Title:**	Group Assignment

I/We declare that:

- *This assignment is my/our own original work, except where I/we have appropriately cited the original source.*
- *This assignment or parts of it has not previously been submitted for assessment in this or any other subject.*
- *I/We allow the assessor of this assignment to test any work submitted by me/us, using text comparison software for plagiarism.*
  (For more information, Please read the Academic Integrity Guidelines)

Name : Lim Xin Yee **Student ID:** 0124305 **Email :** 0124305@kdu-online.com **Mobile No:** 0149682393 **Signature:** **Date:** 05/04/2021	**Name :** Thean Jun Chao **Student ID:** 0127122 **Email :** 0127122@kdu-online.com **Mobile No:** 0173479830 **Signature:** **Date:** 05/04/2021	**Name :** **Student ID:** **Email :** **Mobile No:** **Signature:** **Date:**
Name : **Student ID:** **Email :** **Mobile No:** **Signature:** **Date:**	**Name :** **Student ID:** **Email :** **Mobile No:** **Signature:** **Date:**	**Name :** **Student ID:** **Email :** **Mobile No:** **Signature:** **Date:**

For office use only – Lecturer comments (if applicable)

Marks Breakdown

Table of Contents

1.0 Introduction

1.1 Problem Statement

Social media platforms have become one of the most convenient and widely used platforms for individuals to share their ideas. These platforms also facilitate interaction and communication, allowing users to share thoughts on posts or as posts. It is indeed a good way of knowledge exchange, keeping people up to date on the current events and issues. However, some people take misuse these platforms by posting hate comments, establishing their hate towards a specific group of people. Hate speech refers to a speech that is abusive and threatening towards certain individuals or groups, it can also be defined as expressing hate or encouraging violence towards a person of a targeted group like race, religion, sex or sexual orientation. Hate speech is dangerous as could lead to hate crimes. For instance, Asians are being attacked due to the false belief of covid-19 originated from China (Norwood, 2021). In recent years, data scientists have been researching building a reliable automated hate speech detection system.

1.2 Limitations of Existing Solutions

Social media platforms have established rules to restrict hate speech. However, enforcing these rules is labour intensive and time-consuming. An automated system capable of classifying hate speech is desirable (MacAvaney et al., 2020). Hence, several approaches to automated hate speech classification have been used to automate the process. However, each of them has its limitations which will be discussed below.

1.2.1 Keyword-based Approach

The keyword-based approach identifies hateful keywords by using a dictionary. Hatebase.org is an example of a well-maintained database that stores hateful terms across 95 languages. Although keyword approaches are fast and easy to understand, it does not identify hateful speech that does not use the words in the dictionary. Besides, the system may not be able to understand the semantics of the word used, which could result in misclassification. Therefore, keyword-based approaches could turn out to have high precision but low recall (MacAvaney et al., 2020).

1.2.2 Source Metadata

The characteristics of a post can be further understood by collecting additional information from social media such as the demographics and the social engagement of the user, and the location and timestamp when the post is being posted. However, these data are considered sensitive and collecting them could raise privacy issues. Therefore, researchers often do not have access to these data. Besides, the system could be biased against users whose posts are always flagged, resulting in flagging their non-hateful posts (MacAvaney et al., 2020).

1.2.3 Machine Learning Classifiers

Machine learning models are models that learn from the training data provided to them. Supervise learning learns from labelled data to predict the probability of future events or given scenarios while unsupervised learning clusters the data into clusters based on their characteristics. Researchers have been investigating different machine learning algorithms to discover the best algorithm to classify hate speech. The most commonly used algorithms for hate speech classification are supervised learning such as logistic regression, Naïve Bayes, random forest, support vector machine, and k-means.

Figure 1 shows four tables generated by Abro et al. (2020) on their study of the reliability of eight machine learning algorithms and three feature engineering techniques in classifying hate speech. The Twitter dataset with three different classes, namely hate speech, offensive language, and neither hate nor offensive, were used to conduct their study. The eight algorithms include logistic regression, Naïve Bayes, random forest, support vector machine, k nearest neighbour, decision tree, and adaptive boosting. The three feature engineering techniques include bigram with TF-IDF, Word2Vec, and Doc2Vec. The study has found that the support vector machine algorithm with the bigram feature engineering technique outperformed other models and techniques.

TABLE III. PRECISION OF ALL 24 ANALYSIS

Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.72	0.71	0.73	**0.77**	0.61	0.71	**0.75**	**0.58**
Word2vec	0.69	0.66	0.66	0.70	0.64	**0.62**	0.65	0.69
Doc2vec	0.70	0.65	0.65	0.70	0.69	**0.61**	0.66	0.71

The bold marked values represented are the higher and lower result values.

TABLE IV. RECALL OF ALL 24 ANALYSIS

Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.75	0.73	0.75	**0.79**	**0.57**	0.73	**0.78**	0.70
Word2vec	0.72	0.67	0.68	0.73	**0.61**	0.63	0.68	0.71
Doc2vec	0.72	**0.62**	0.67	0.72	0.65	0.63	0.67	0.71

TABLE V. F-MEASURE OF ALL 24 ANALYSIS

Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.72	0.68	0.74	**0.77**	**0.47**	0.71	0.73	0.63
Word2vec	0.69	0.66	0.66	0.70	0.61	**0.60**	0.65	0.65
Doc2vec	0.70	0.63	0.66	0.72	0.65	**0.61**	0.66	0.66

The bold marked values represented are the higher and lower result values.

TABLE VI. ACCURACY OF ALL 24 ANALYSIS

Features	LR	NB	RF	SVM	KNN	DT	AdaBoost	MLP
Bigram	0.75	0.73	0.75	**0.79**	**0.57**	0.73	0.78	0.70
Word2vec	0.72	0.67	0.68	0.73	**0.61**	0.63	0.68	0.71
Doc2vec	0.72	**0.62**	0.67	0.72	0.65	0.63	0.67	0.71

Figure 1: The results of the study conducted by Abro et al. (2020)

The study conducted by Abro et al. (2020) has demonstrated that the support vector machine (SVM) outperformed other machine learning algorithms in classifying the hate speech from the Twitter dataset with three classes. However, the aim is to produce a model that classifies hate speech and non-hate speech, classifying tweets into three classes can be

redundant, resource-consuming, and potentially lowers the performance of the model in classifying the hate speech from the Twitter dataset.

1.3 Proposed Solution and Approach

Since the main research is to accurately classify hate speech from non-hate speech, we propose to use the binary approach in classifying hate speech. We will be using the SVM algorithm and the TF-IDF bigram feature engineering as our algorithm and technique to train our models. SVM and TF-IDF bigram combined have been proven to produce the best result. We will classify the Twitter dataset only into two classes (the binary approach), namely hate and non-hate. We believe that this approach would produce a model that can better classify hate speech in the Twitter dataset as compared to the three classes approach (the trinary approach).

Furthermore, to ensure that our system or model is trustworthy, we have planned to implement interpretable machine learning in our study. This is to enable the system to justify the prediction made. Abro et al. (2020) state that one of their weaknesses is the inability to determine the severity of the hate speech. With interpretation, graphic insights of the prediction and the likelihood of the text to be hate or offensive or neither are available. Making the system more reliable.

1.4 Hypothesis & Research Question

This paper aims to prove that the binary approach of training the Twitter dataset will outperform the trinary approach that is using the same algorithm and feature engineering technique in accurately classifying the hate speech in the dataset.

2.0 Methodology (Tools, Techniques & algorithms used)

2.1 Data Collection

The Twitter dataset was obtained from kaggle.com. It comes as a Microsoft excel .csv file. The dataset consists of seven columns, namely index, counts, hate_speech, offensive_language, neither, class, and tweet. Table 1 summarizes the indication of each column.

Column Name	Description
index	The number for each record, starting from zero.
counts	The number of CrowdFlower users who coded each tweet (min is 3)
hate_speech	The number of CrowdFlower users who labelled the tweet as a hate speech
offensive_language	The number of CrowdFlower users who labelled the tweet as an offensive language
neither	The number of CrowdFlower users who labelled the tweet as neither hate nor offensive
class	Class label for each tweet based on the majority CrowdFlower users - 0 indicates the tweet is a hate speech - 1 indicates the tweet is an offensive tweet - 2 indicates the tweet is neither hate nor offensive
tweet	The uncleaned tweet text content

Table 1: The details of the dataset columns

2.2 Data Preparation

The pandas library is used for storing the dataset into dataframe. The data cleaning process begins with dropping the unnecessary columns for model training. Since the tweet and the class are the only columns needed for model training, other columns have been dropped.

Each tweet in the dataset has gone through the text processing stages which includes removing punctuations, HTML tags, and other symbols such as emoji. We used 2 libraries to perform this function, which is the tweet-preprocessor and the regular expression library. The tweet-preprocessor library provides the removal of URLs, hashtags, mentions, reserved words such as RT and FAV, emojis, and smileys. The regular expression library is used to remove other special characters that the tweet-preprocessor does not remove, such as punctuations and HTML tags, as well as converting all tweets into lower cases.

The Natural Language Toolkit (nltk) library is used to tokenize text, the process of splitting sentences into words; remove stop words; lemmatize each word for feature extraction. The reason for using lemmatization instead of stemming is because stemming only removes the last few characters from a word and could lead to the incorrect meaning of the word. Lemmatization on the other hand considers the context of the text and convert it to its base form. Therefore, lemmatization could better preserve the meaning of words in a sentence.

After cleaning the tweets, we defined a new column called “clean_tweet” to store them. We have also defined a new column called “binary_class” and grouped classes 1 (offensive) and 2 (neither) from the original “class” column into 1, which indicates the non-hate speech. The 0 (offensive) class in the original “class” remained as 0 (offensive). To distinguish these labelling columns, we named our original class column as “trinary_class”.

2.3 Exploratory Data Analysis (EDA)

One of the tools that we used for EDA is the matplotlib.pyplot library. It has helped us to discover the data insights such as the number of tweets in each class, namely hate, offensive, and neither. Other EDA functions were performed using the pandas dataframe methods. However, we have changed the tool to perform EDA, which will be discussed in section 4.

Figure 2 shows the class distribution for the binary_class, where zero (0) indicates hate speech; one (1) indicates non-hate speech.

Figure 2: Details of the binary_class variable

Figure 3 shows the class distribution for the trinary_class, where zero (0) indicates hate speech; one (1) indicates offensive language; two (2) indicates neither hate nor offensive.

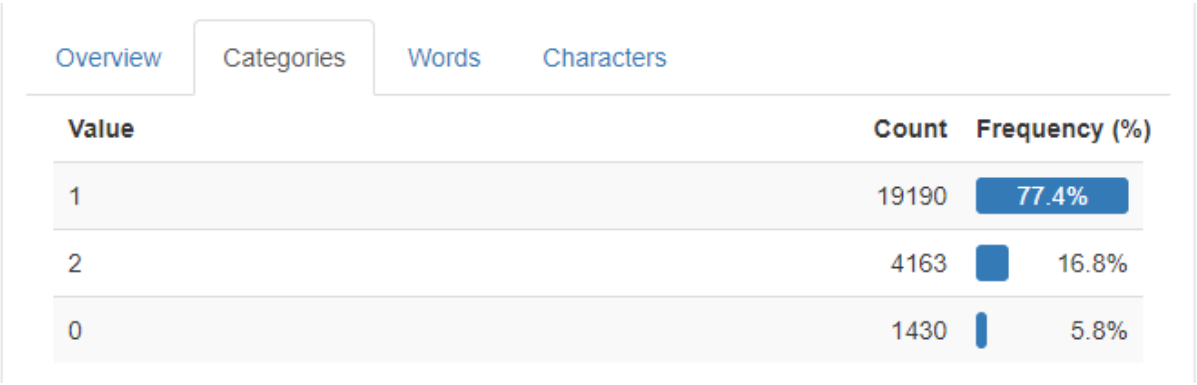


Figure 3: Details of the trinary_class variable

2.3 Feature Engineering

Bigram with TF-IDF feature engineering technique was used as it has been proven to be the best technique by Abro et al. (2020). According to Shah (2020), TF-IDF has been proved effective as it can show the importance of a word by balancing the frequency on the number of the word appear in a certain document with the sequence of the word appears in certain data set in the document. Besides, it reduces the importance of a word if it is repeated

multiple times in a document so that the unique and meaningful words can be spotted easily, awarded with a higher score, and used to train the model.

2.4 Data Splitting

We have decided to follow the study conducted by Abro et al. (2020) by splitting the dataset into the training set and the testing set in the ratio of 80:20. This implies that 80% of the dataset will be used to train the model while 20% of the dataset will be reserved for classifier evaluation. Since we are testing two models with different class labels, we have declared 2 y-variables as the dependent variable for each model, namely y1 and y2. Therefore, x, y1, and y2 are all being split into an 80:20 ratio. We imported train_test_split component from the skilearn.model_selection library to perform data splitting.

2.5 Machine Learning Algorithm

The support vector machine algorithm is used as it has been proven to be the best machine learning algorithm in the study conducted by Abro et al. (2020). We implemented the algorithm using the skilearn library svm component. Since comparing different machine learning algorithm is not part of our study, we only use the SVM algorithm.

2.6 Classifier Evaluation

The library sklearn.matrics is imported to perform classifier evaluation. The classification report provides a table summary for accuracy, precision, recall, and f1-score just with one line of code. Pandas crosstab together with seaborn's heat map is used to generate the confusion matrix.

2.7 Model Interpretation

The lime library is used for model interpretation. Lime provides graphical interpretation that is easily understood by non-data science professionals. With the ease of understanding the interpretation of the model comes with a higher chance of the model being accepted by the non-data science professionals. Some samples are available in section 3.3.

2.8 Platform

Jupyter notebook is used to code and train our model. It is a free, open-source and interactive notebook that allows us to code and write notes with adjustable heading and font sizes. It also allows us to run code block by block so that we know which part of the code contains the error, making the debugging process easier.

3.0 Results

3.1 Classifier Evaluation Findings

Figure 4 is a heat map for the binary model, whereby 0 represents hate speech, 1 represents non-hate speech. It shows that there are 23 true positives for hate speech, approximately 2700 false positives for hate speech, approximately 4600 true positives for non-hate speech, and 23 false positives for non-hate speech.

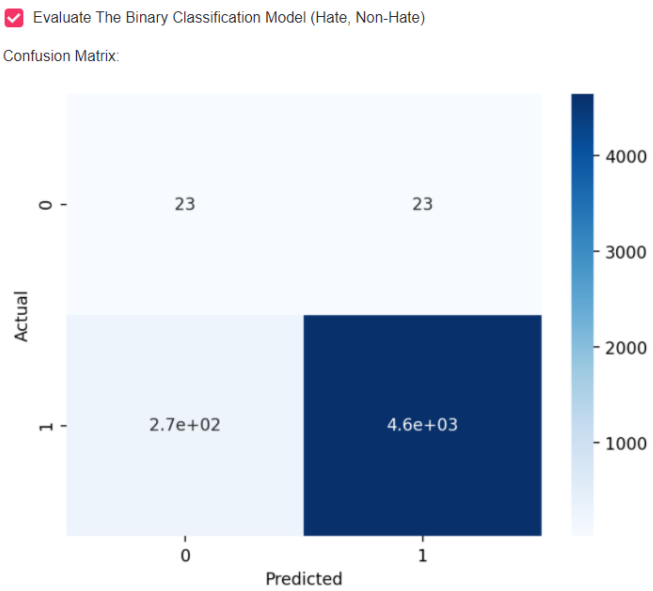


Figure 4: The confusion matrix for the binary classification model

Figure 5 is a model report for the binary model. From figure 4 and 5, it is evident that the model tends to misclassify many of the non-hate speech as hate speech. It has misclassified approximately 2700 non-hate speech as hate speech and only classifies 23 hate speech correctly. This gives it a very low precision of 0.08. It is only able to classify half of the actual hate speech correctly, giving it 0.5 recall. Lastly, the overall f1-score for classifying hate speech is very low. On the positive side, it has a very good ability in classifying non-hate speech, with 100% precision 95% recall, and 97% in the f1-score.

Model Report:

	precision	recall	f1-score	support
0	0.08	0.50	0.14	46
1	1.00	0.95	0.97	4911
accuracy			0.94	4957
macro avg	0.54	0.72	0.55	4957
weighted avg	0.99	0.94	0.96	4957

Figure 5: The model report for the binary classification model.

Figure 6 is a heat map for the trinary model, whereby 0 represents hate speech, 1 represents offensive language, and 2 represents neither hate nor offensive. It shows that there are 23 true positives for hate speech, and approximately 2200 + 32 false positives for hate speech, whereby approximately 2200 tweets are actually offensive language and 32 are neither. For offensive language, there are approximately 3700 true positives, and 29 + approximately 1200 false positives, whereby 29 are actually offensive speech and 1200 are actually neither. For neither hate nor offensive, there are approximately 7400 true positives, and 6 + 92 of false positive, whereby 6 actually belongs to hate speech and 92 actually belongs to offensive speech.

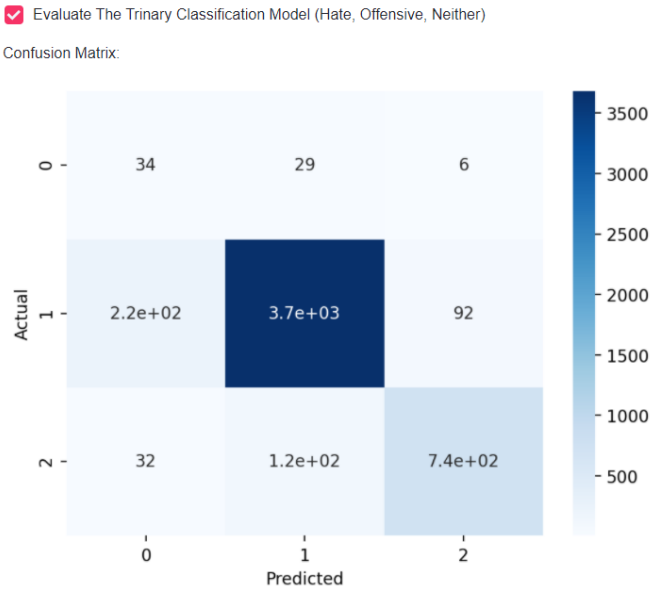


Figure 6: The confusion matrix for the trinary classification model

Figure 7 is a model report for the binary model. From figure 6 and 7, it is evident that the model also tends to misclassify many of the non-hate speech as hate speech. However, it has slightly higher precision and f1-score than the binary model. It has slightly lower precision in classifying neither hate nor offensive speech, which is still acceptable because the accuracy score still high. However, it has relatively high scores in classifying offensive speech due to a large amount of offensive dataset available in the dataset. Since the objective of the study is to identify hate speech, we can conclude that the trinary classification model is better at classifying hate speech.

Model Report:

	precision	recall	f1-score	support
0	0.12	0.49	0.19	69
1	0.96	0.92	0.94	3995
2	0.88	0.83	0.85	893
accuracy			0.90	4957
macro avg	0.65	0.75	0.66	4957
weighted avg	0.93	0.90	0.91	4957

Figure 7: The model report for the trinary classification model

3.3 Samples Screenshot of the Application

3.3.1 The Profile Reporting Page

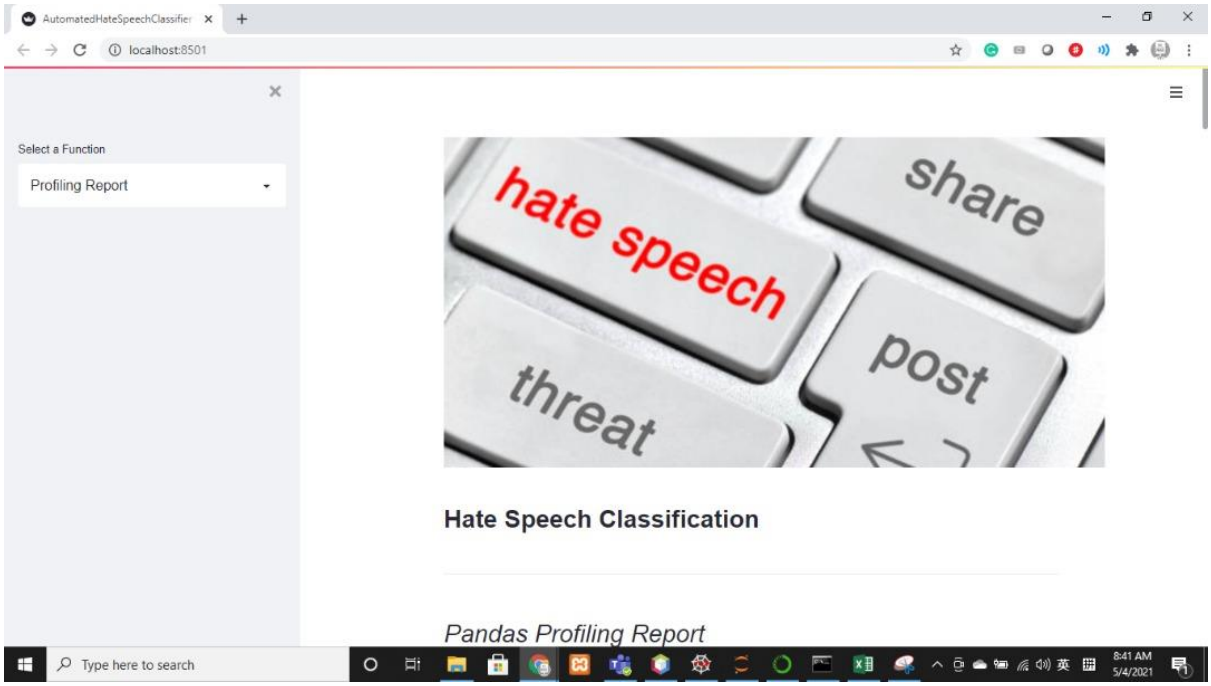


Figure 8: The main page of the application

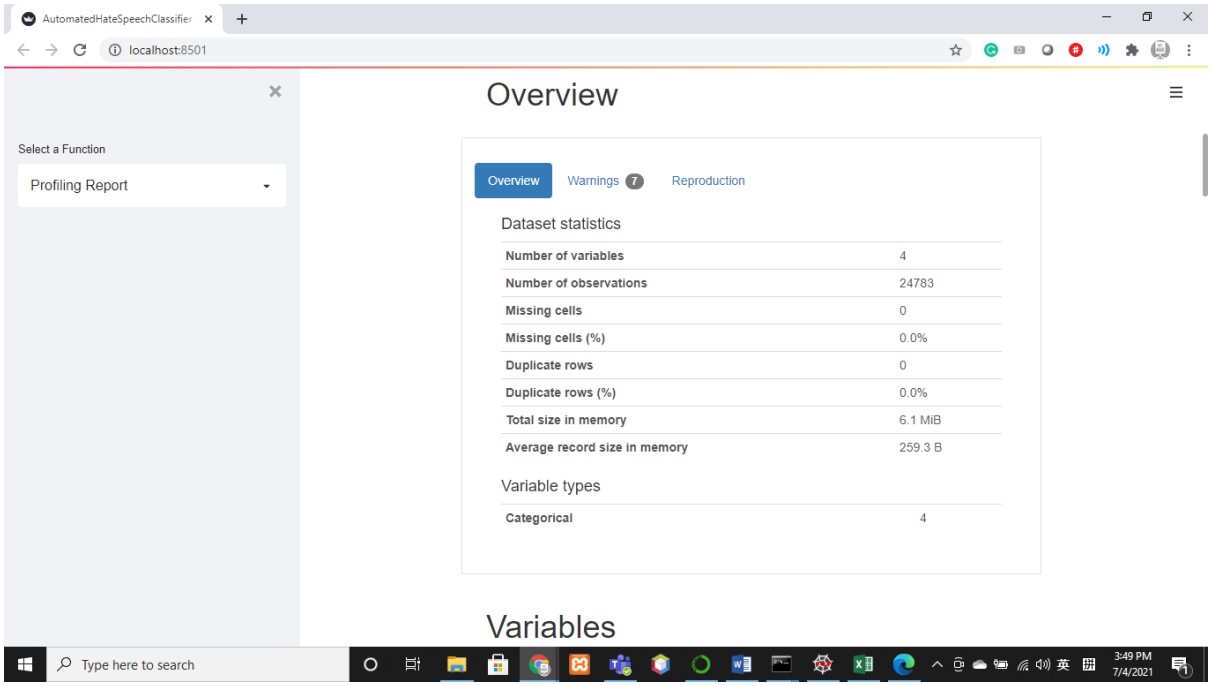


Figure 9: An overview of the dataset

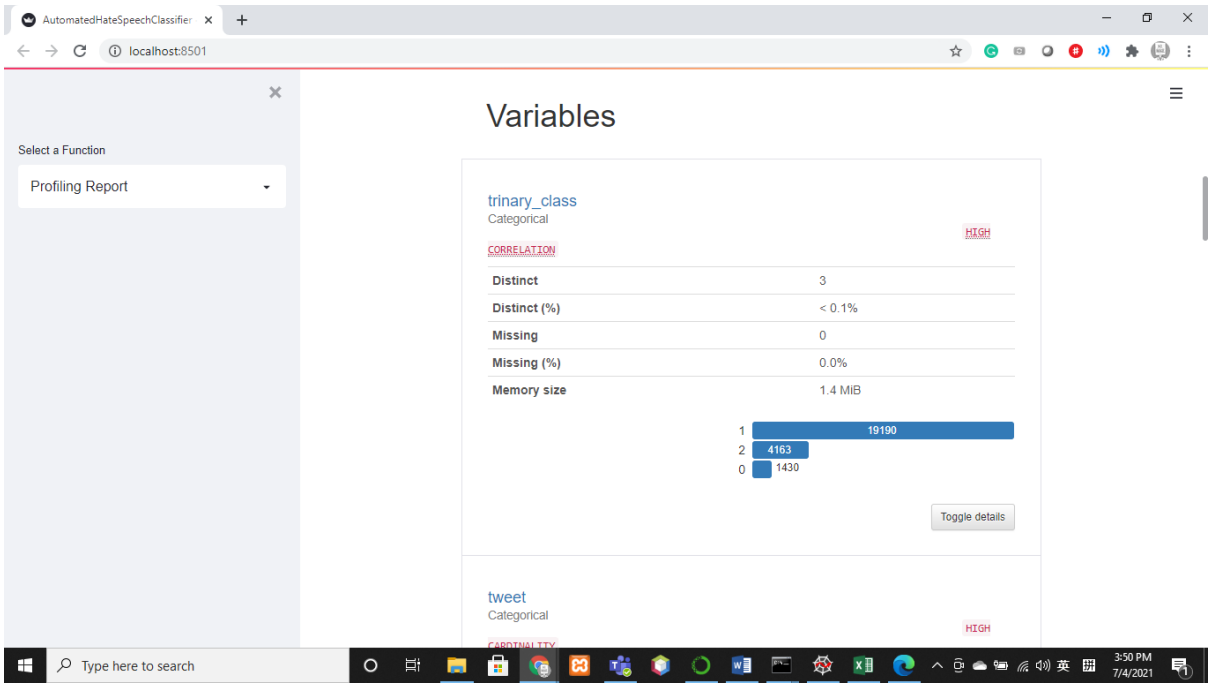


Figure 10: Partial overview of the trinary_class variable

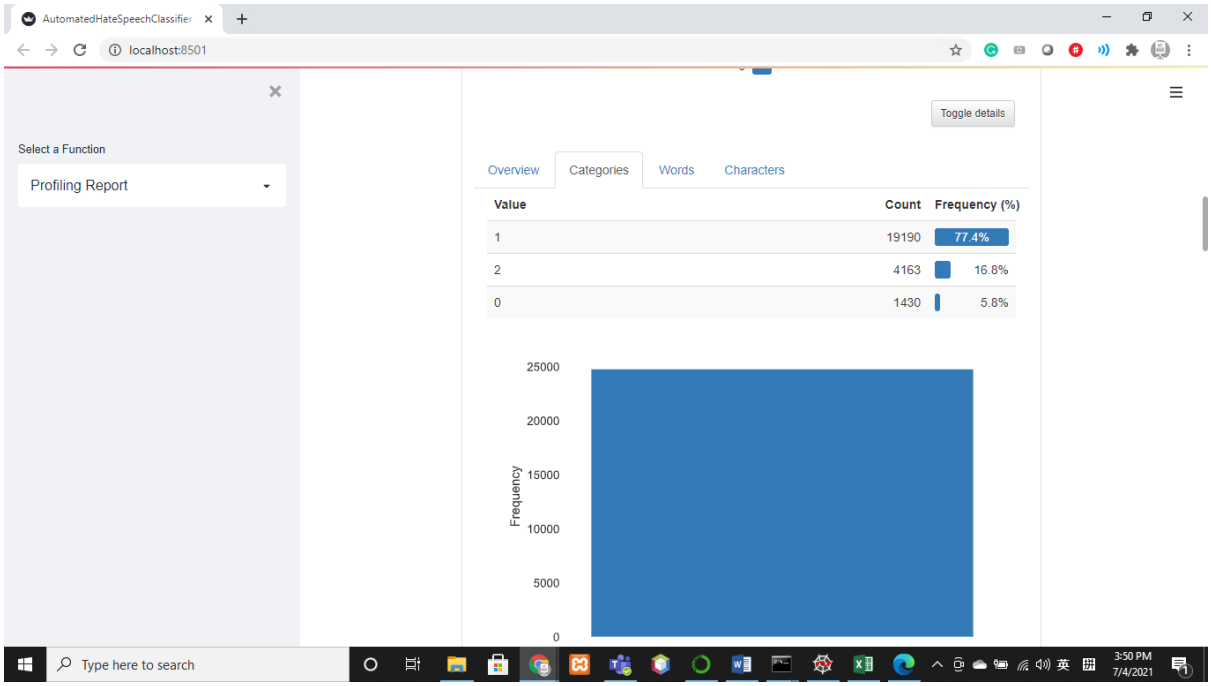


Figure 11: Bar chart class distribution of the trinary_class. 1 – Offensive; 2 - Neither offensive nor hate; 0 – hate.

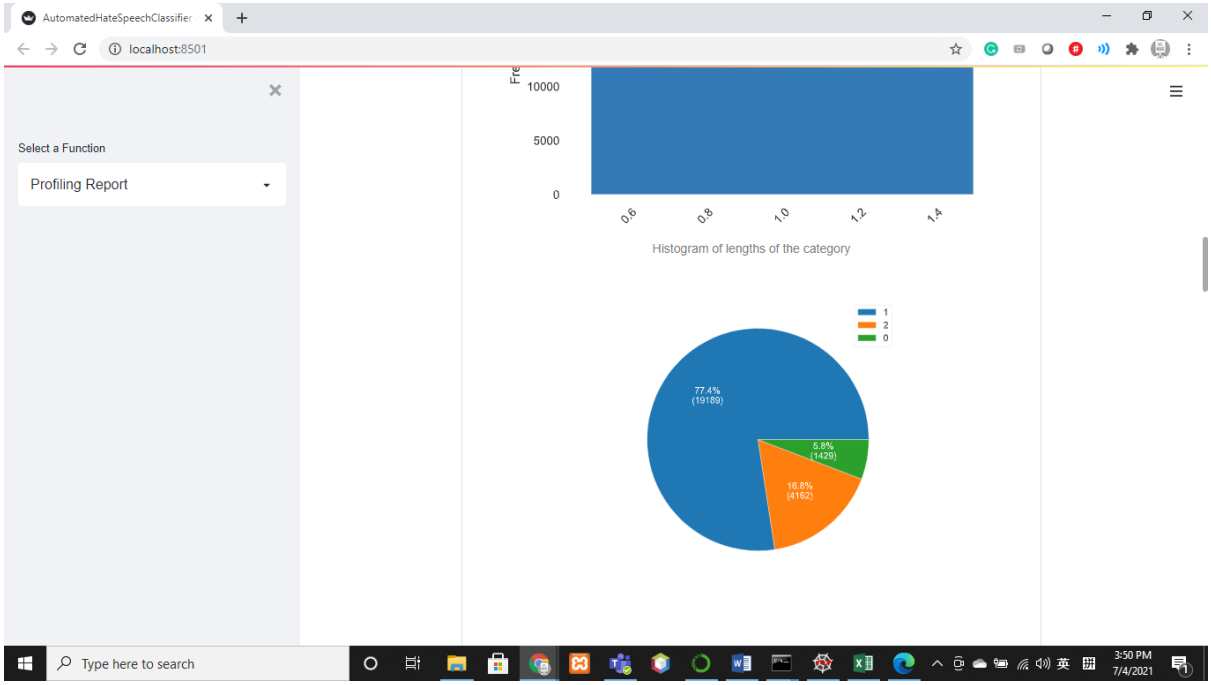


Figure 12: Pie chart class distribution of the trinary_class. 1 – Offensive; 2 - Neither offensive nor hate; 0 – hate.

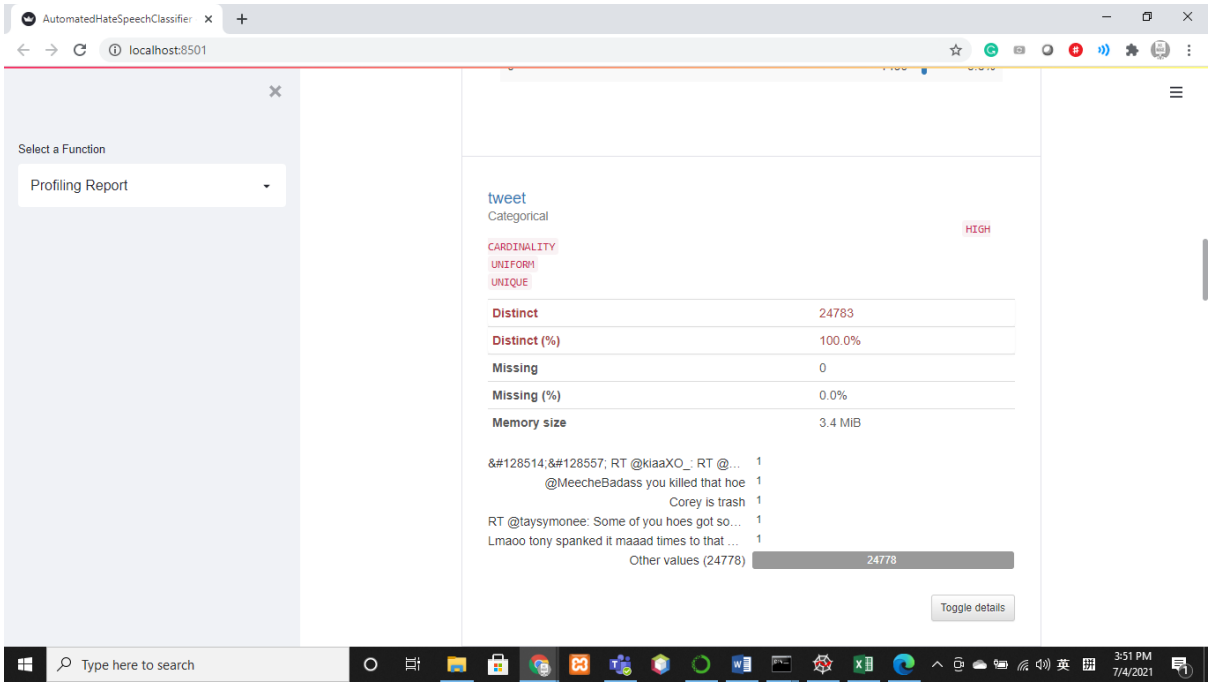


Figure 13: An overview of the tweet column

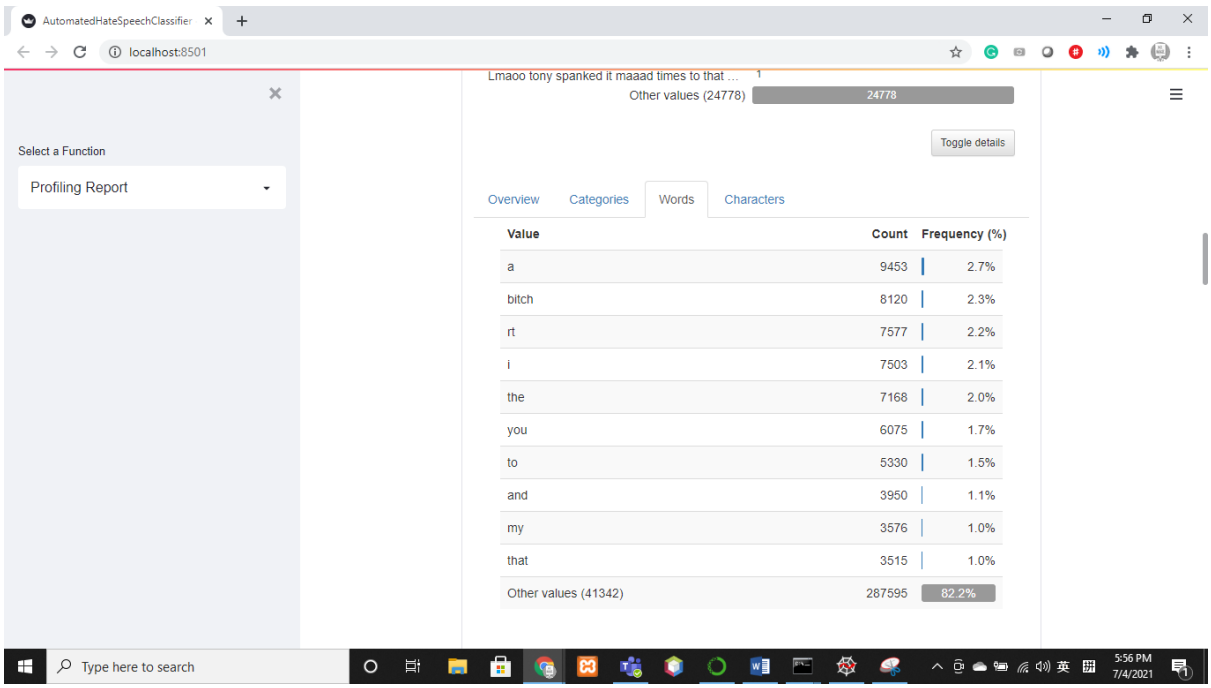


Figure 14: Frequent words appearing in the tweet column

.

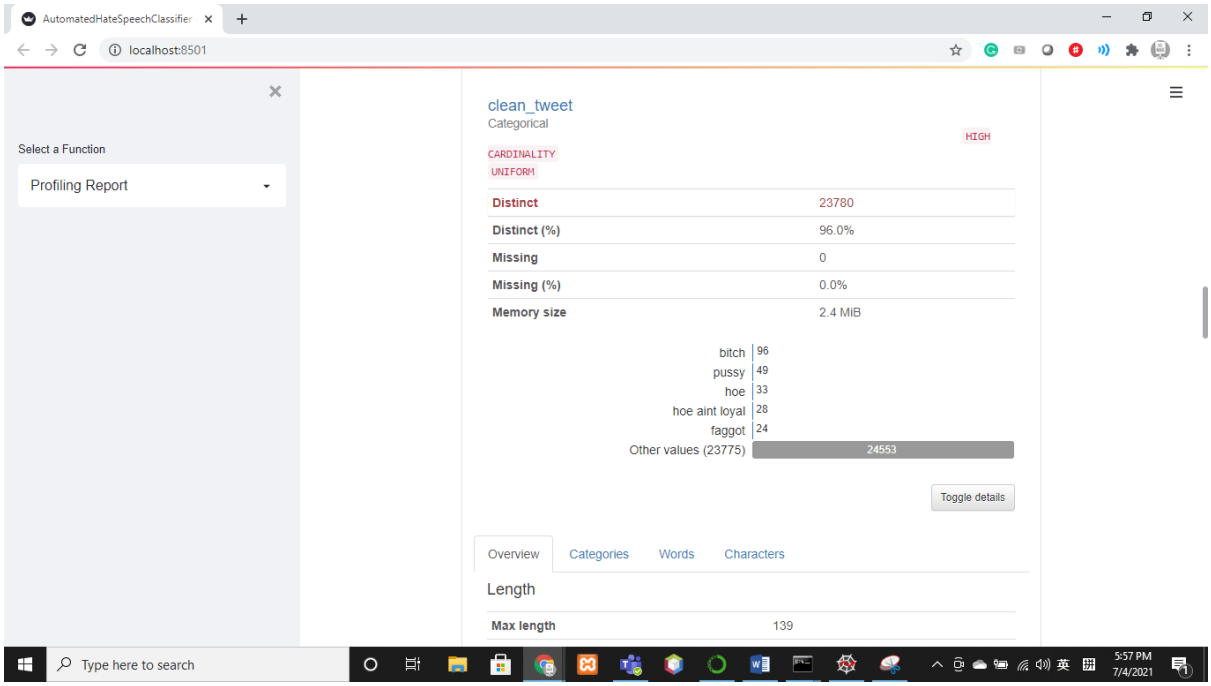


Figure 15: Overview of the clean_tweet column

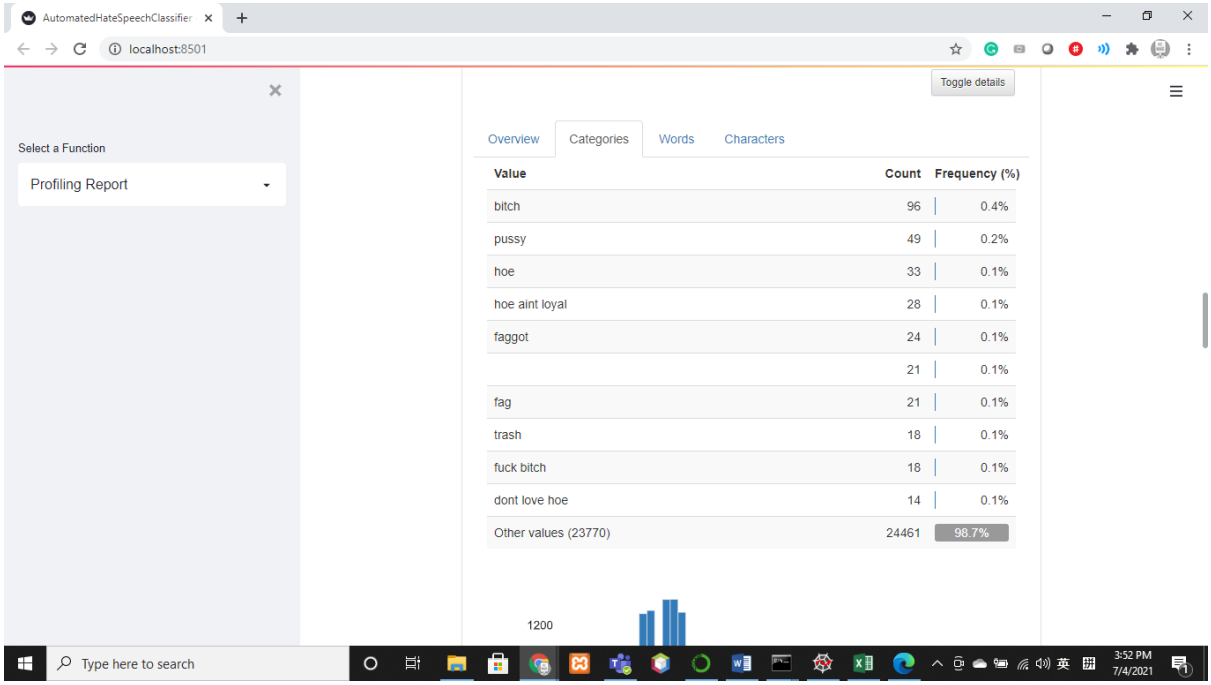


Figure 16: Frequent words appearing in the clean_tweet column

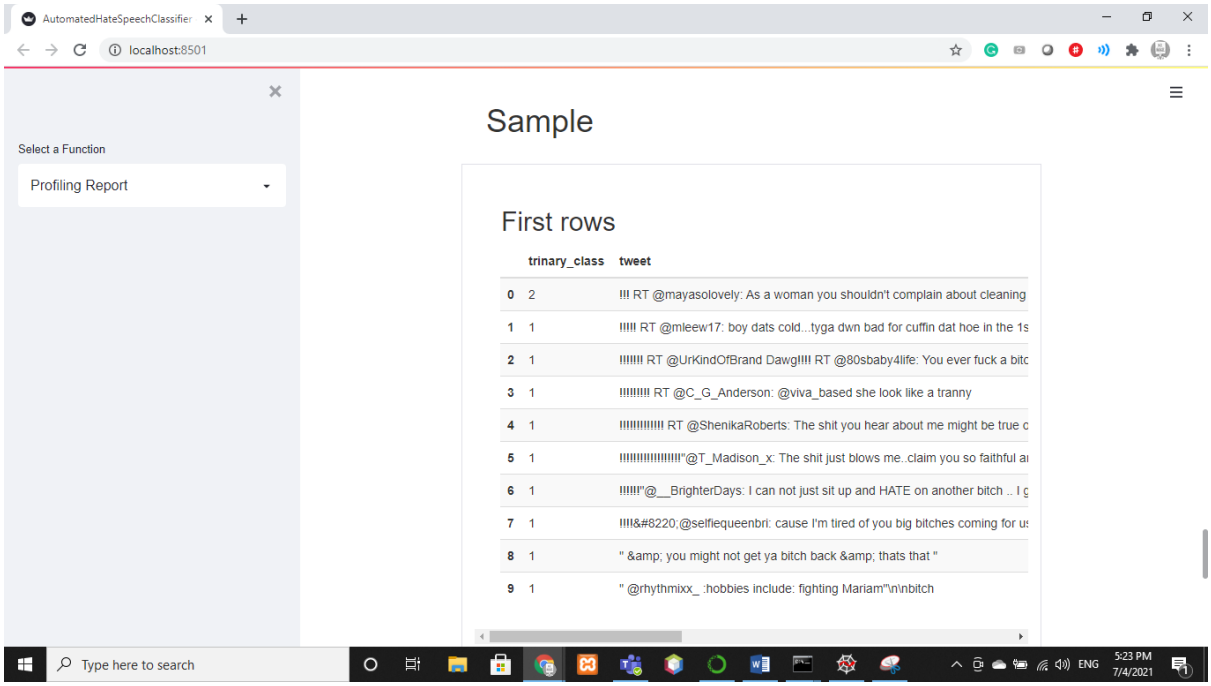


Figure 17: A display of the top 10 records, with clean_tweet and binary_class columns.

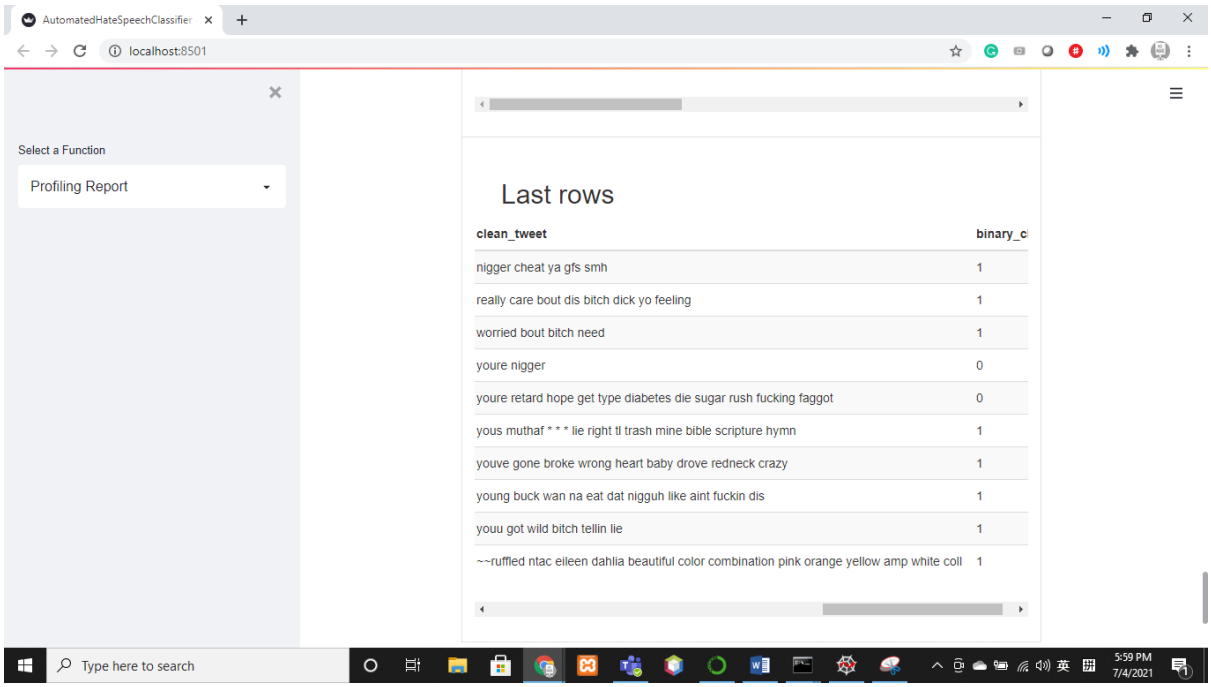


Figure 18: A display of the bottom 10 records, with clean_tweet and binary_class columns.

3.3.2 The Speech Classification Models Page

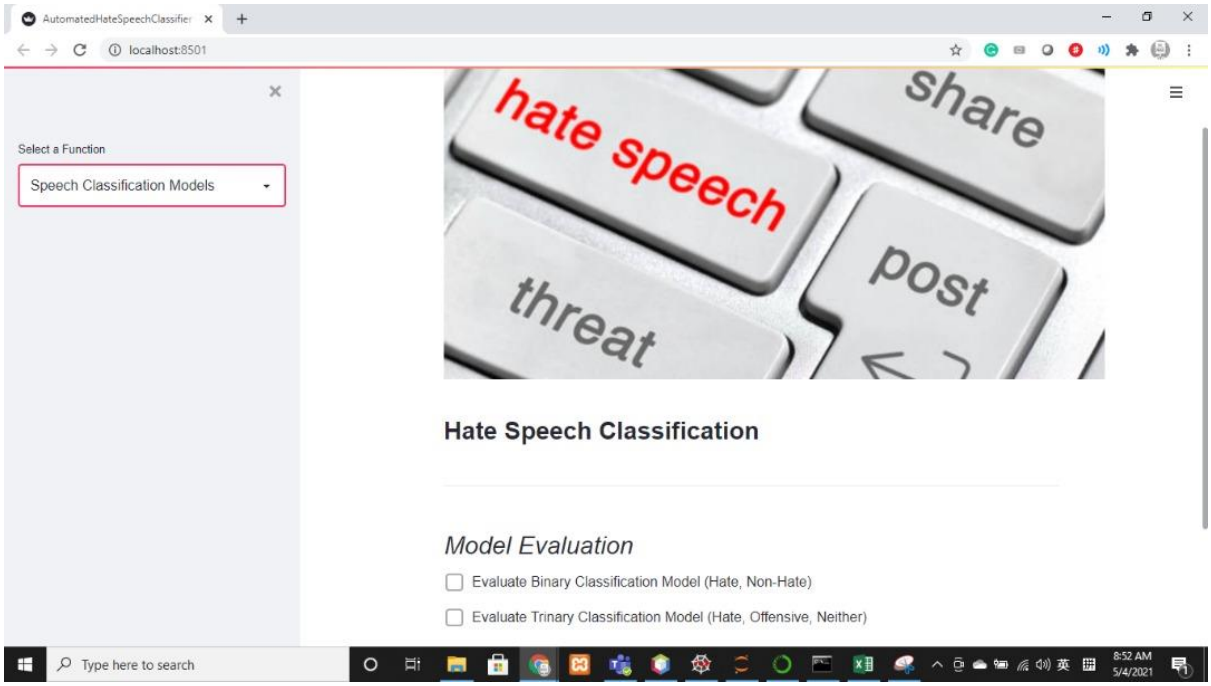


Figure 19: The model evaluation page

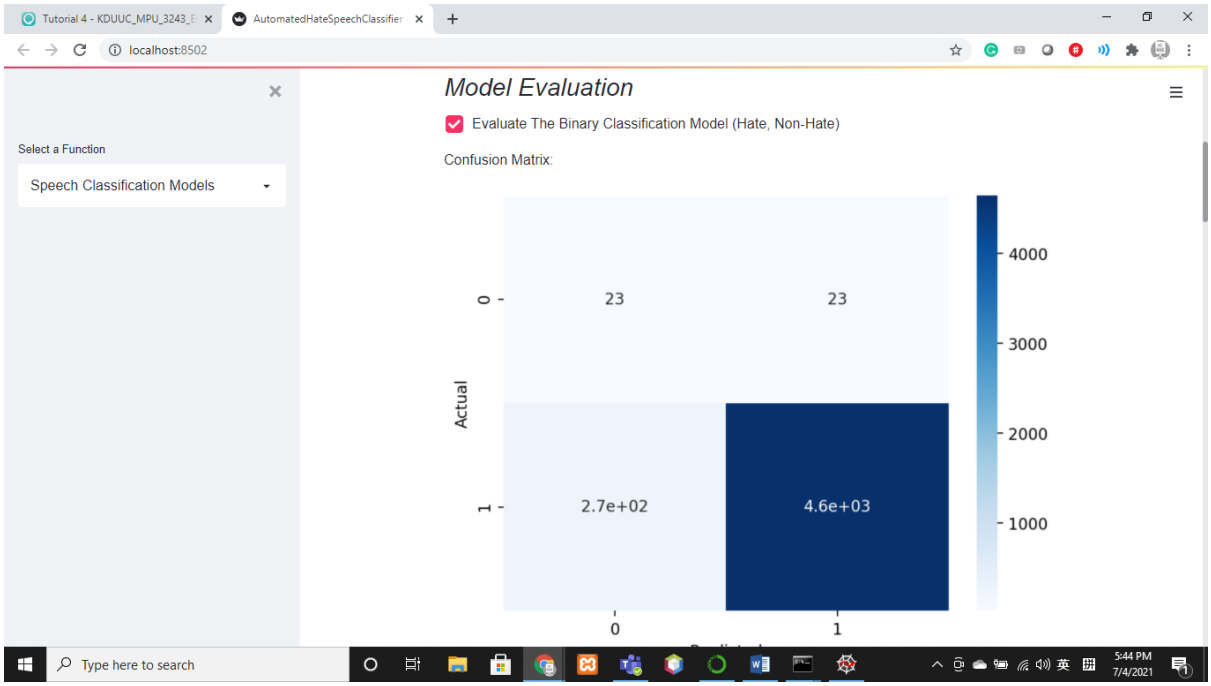


Figure 20: The confusion matrix of the binary model

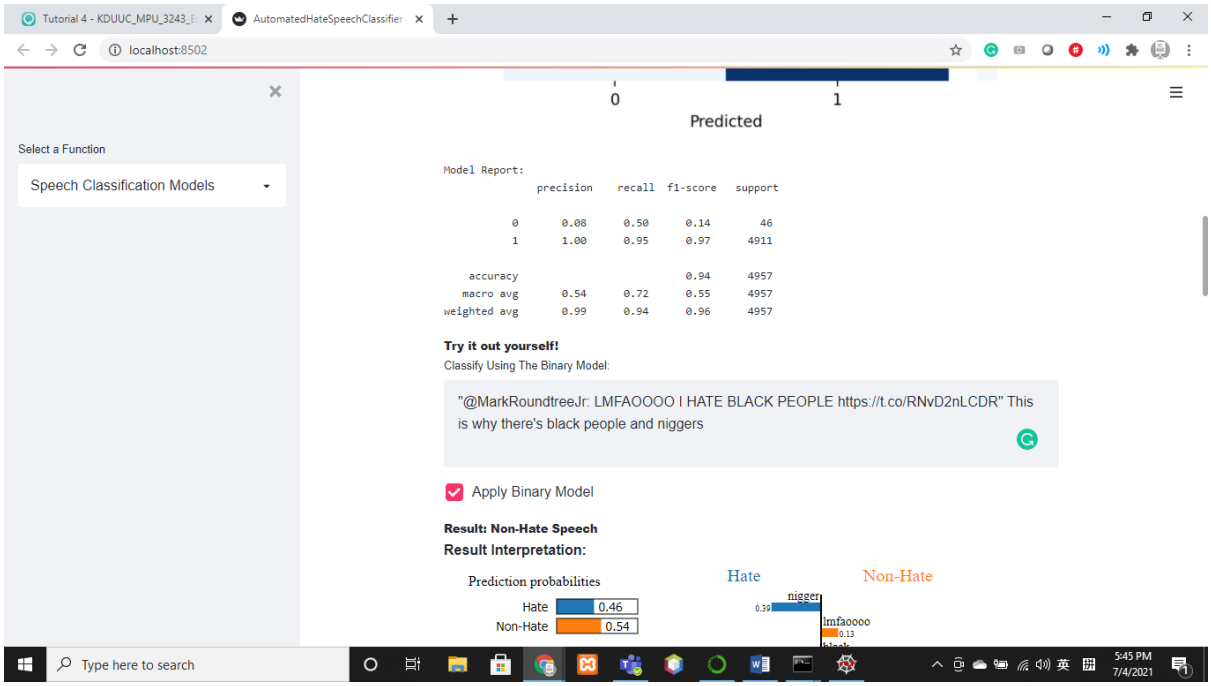


Figure 21: The model report of the binary model

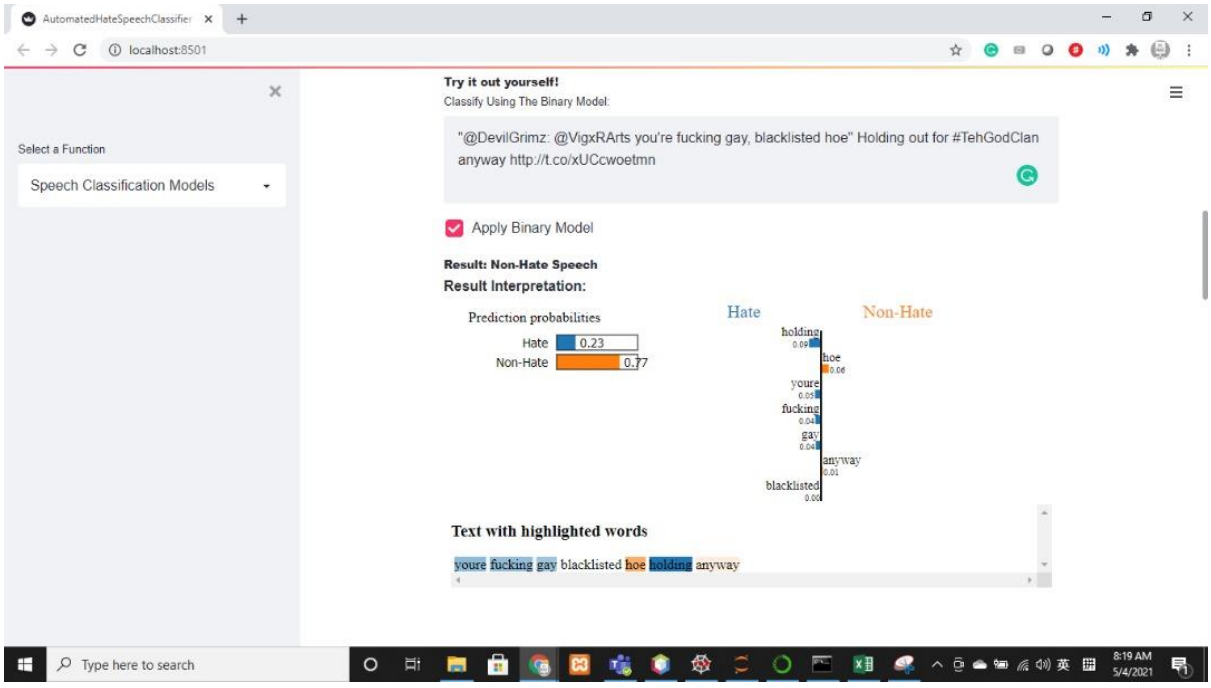


Figure 22: Sample model interpretation of the binary classification model

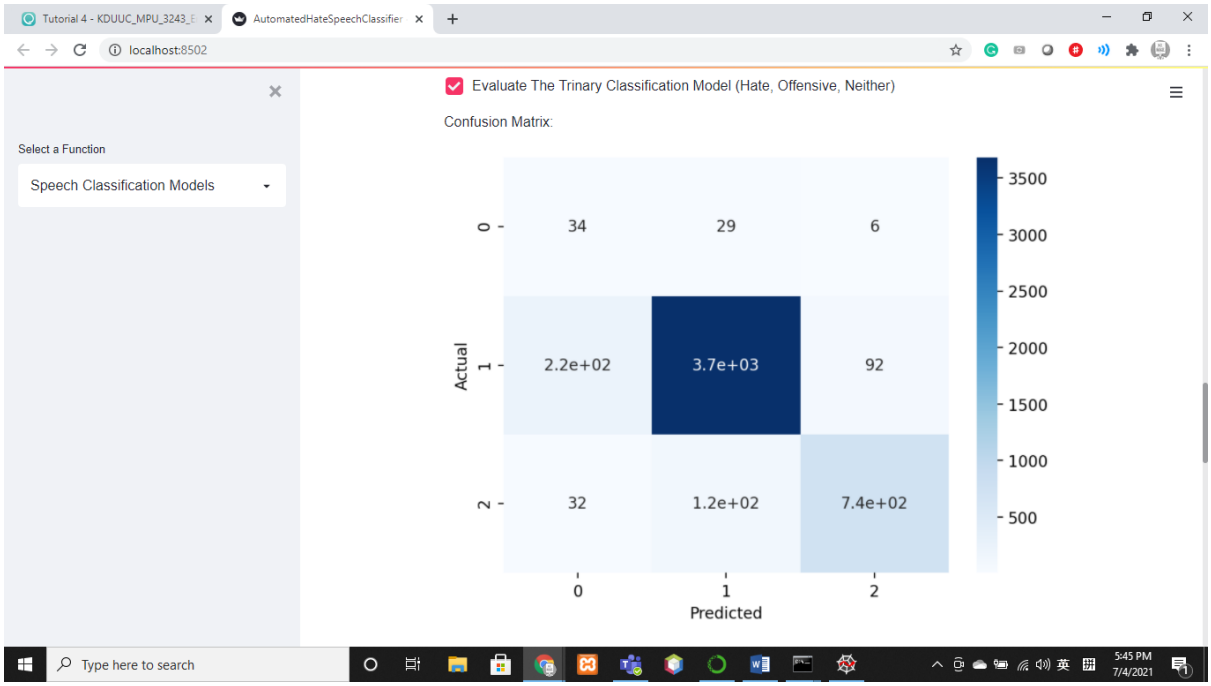


Figure 23: The confusion matrix of the trinary model

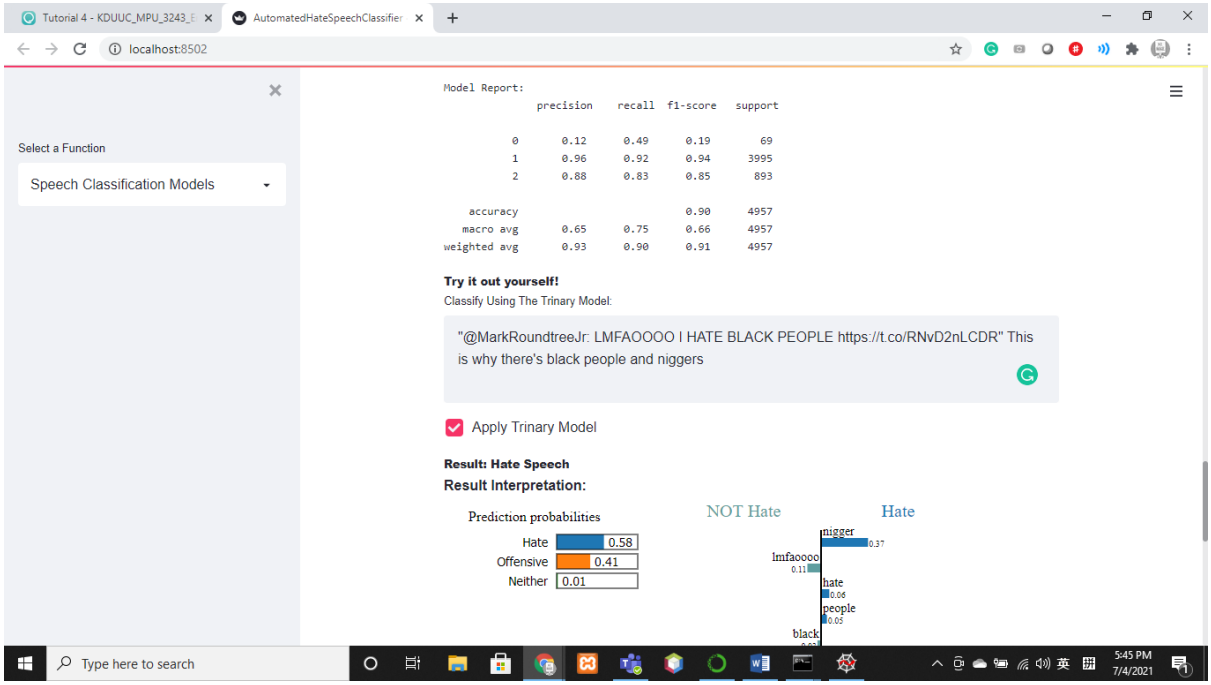


Figure 24: The model report of the trinary model

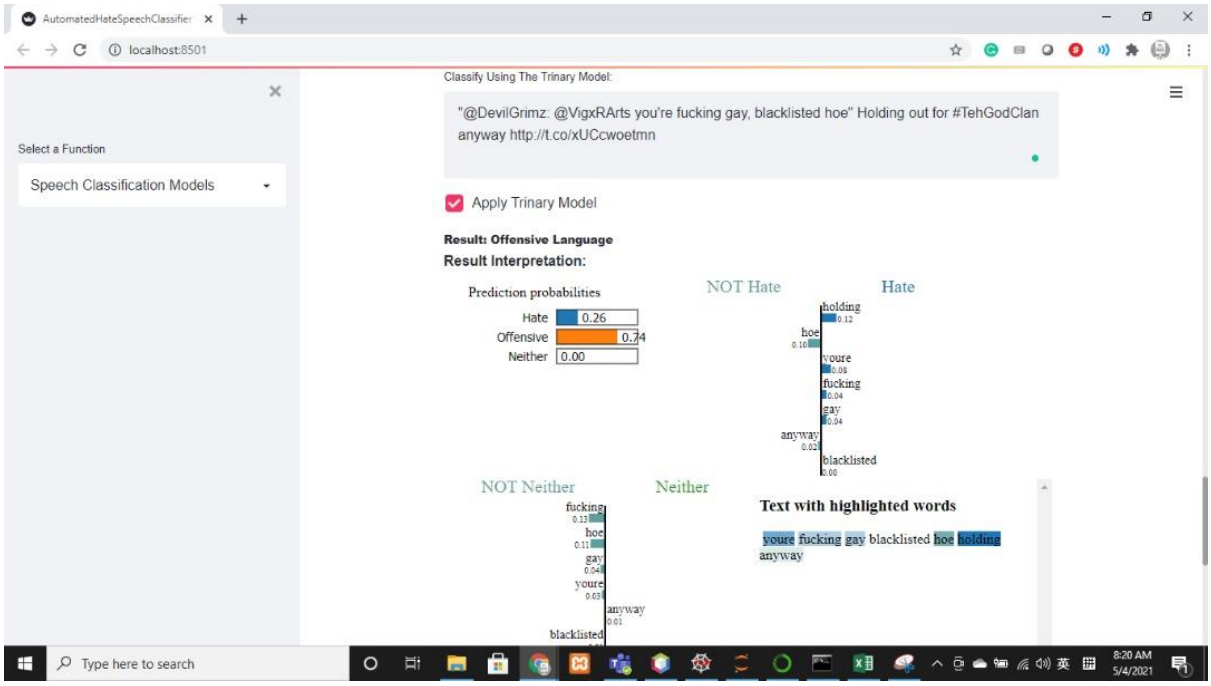


Figure 25: Sample model interpretation of the trinary classification model

4.0 Discussion

4.1 Changes in Tools and Methods Used

Changes in tools and methods have been made throughout the development process. The first change is switching the platform from Jupyter notebook to Spyder. This is because spyder provides an easier way for us to make our data application using the streamlit library. The tool to conduct EDA has also been changed. We have discovered the pandas_profiling library which displays comprehensive visual and graphical information about our dataset with just a few lines of codes. We also wanted to use stemming for data cleaning, however, we found out that lemmatizing does a better job in transforming words into their root words.

4.2 Answer to the Research Question

Although both the binary and trinary models did not do well in classifying hate speech in section 3.1, it is evident that the trinary model performs better than the binary model in classifying hate speech. Comparing figure 5 and 7, we notice that the trinary model has a higher precision score (0.12) compared to the binary model (0.08) in hate speech (0) classification. The trinary model also has a higher f1-score (0.19) compared to the binary model (0.14) in hate speech classification. This shows that the trinary model classifies hate speech better than the binary model. Therefore, we have to reject our null hypothesis claiming that the binary approach of training the Twitter dataset will outperform the trinary approach that is using the same algorithm and feature engineering technique in accurately classifying the hate speech in the dataset.

4.3 Reflection

As observed in figure 5 and figure 7, we suspect that the possible reason leading to the misclassification of hate speech is due to the low support of the hate speech in the test set, which also means low support in the training set. We have reflected on the mistake made and we think we should have used another dataset or add in more samples of hate speech from other datasets. Overall the model tends wrongly classifies hate speech and only a few hate speech are classified correctly. Since both of our models did well in classifying non-hate speech, we believe that our model would work if a well-balanced dataset is used.

5.0 Conclusion

5.1 Implications of the Findings to Machine Learning Area

The findings of our study imply that the more classes or features used in training a model, the more accurate prediction the model can make. However, the increased number of classes would be resource-consuming and time-consuming. Hence, it is important to access the effect of the features on the dependent variable during feature selection to prevent wasting computation resources.

5.2 Limitations & Future Enhancement

One of the limitations of our study is that we did not find a dataset with a well-balanced number of hate samples to train our model. Further studies using a more balanced dataset is needed to further evaluate the effectiveness of our model in classifying hate speech. Researchers may also conduct studies that compare the performance of a well-balanced dataset versus an imbalanced dataset. Besides, our model does not allow users to contribute so far as we think people may be biased towards certain content. Also, our application is built solely for demonstration purposes. It would be useful if the model is implemented into a social media platform to detect suspected hate speech and asks volunteers on the social media platform to vote if it is hate speech. The data can then be kept to train the model.

---Total word count: 3066 (including picture captions)---

References

Abro, S., Shaikh, S., Ali, Z., Khan, S., Mujtaba, G. and Khand, Z,H. (2020). *Automatic Hate Speech Detection using Machine Learning: A Comparative Study.* International Journal of Advanced Computer Science and Applications, 11(8), 1-8. Retrieved 5 April 2021, from https://thesai.org/Downloads/Volume11No8/Paper_61-Automatic_Hate_Speech_Detection.pdf

Basic Tweet Preprocessing in Python. (2020). Retrieved 5 April 2021, from https://towardsdatascience.com/basic-tweet-preprocessing-in-python-efd8360d529e

MacAvaney, S., Yao, H. R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). *Hate speech detection: Challenges and solutions.* PLOS ONE, 14(8), 6-7. Retrieved 5 April 2021, from https://doi.org/10.1371/journal.pone.0221152

Navlani, A. (2019, December 14). *Text Analytics for Beginners using NLTK. Retrieved* 5 April 2021, *from https://www.mendeley.com/guides/apa-citation-guide*

Norwood, C. (2021, April 1). *What advocates and lawmakers are doing to address growing anti-Asian hate crimes.* Retrieved 5 April 2021, from https://www.pbs.org/newshour/politics/what-advocates-and-lawmakers-are-doing-to-address-growing-anti-asian-hate-crimes

Samoshyn, A. (2020) *Hate Speech and Offensive Language Dataset. (2021)*. Retrieved 5 April 2021, from https://www.kaggle.com/mrmorj/hate-speech-and-offensive-language-dataset

Shah, P. (2020). *Why TF-IDF transformation works better than Count Vectorizer in Machine Learning? - Quora.* (2021). Retrieved 5 April 2021, from https://www.quora.com/Why-TF-IDF-transformation-works-better-than-Count-Vectorizer-in-Machine-Learning

ASSESSMENT RUBRIC

CRITERIA	MARKS					
	16-20	**13-15**	**10-12**	**8-9**	**0-7**	**Comments**
Methodology (30%)	Excellent in documenting the methodology. • Generates complete, clear and unambiguous requirements specification. • Identifies ambiguity in givens and states necessary assumptions. • Uses appropriate diagrams to describe implementation clearly including the design decisions. Overall contents comprehensively articulates all relevant and pertinent issues	Good in documenting the methodology. • Generates requirements specification with minor residual ambiguity. • Identifies ambiguity in givens however necessary assumptions are not fully stated. • Uses appropriate diagrams to describe implementation however contain a small number of errors, omissions or additions.	Satisfactory in documenting the methodology. • Generates requirements specification with some residual ambiguity. • Omits ambiguity in givens and states necessary assumptions ambiguously. • Uses appropriate diagrams to describe implementation however contain a number of errors, omissions or additions.	Weak in documenting the methodology. • Generates requirements specification with substantial ambiguity. • Omits ambiguity in given and necessary assumptions. • Uses inappropriate diagrams to describe implementation and contain large number of errors, omissions or additions.	Unsatisfactory in documenting the methodology. • Generates requirements specification with substantial ambiguity. • Omits ambiguity in given and necessary assumptions. • No diagrams to describe software architecture. It is possible that the methodology is weak or above in some areas and unsatisfactory in others. Unsatisfactory in	

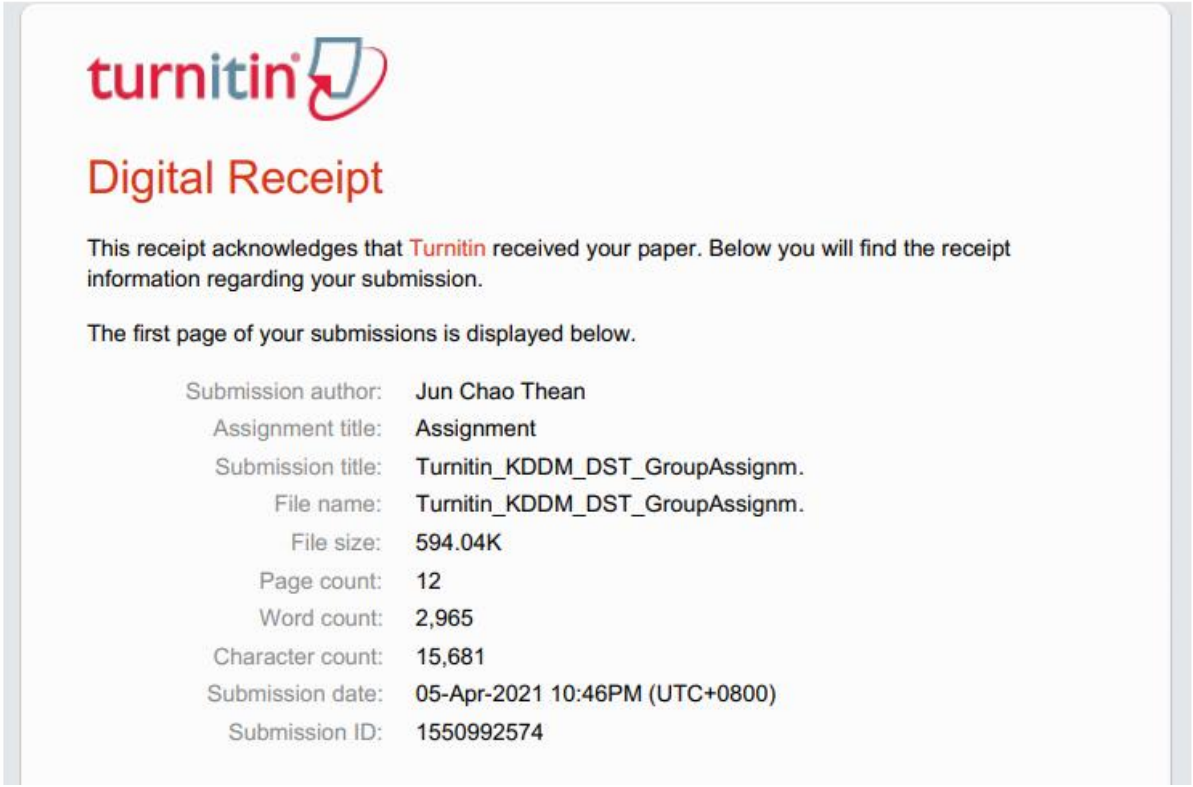
	related to the overall solution. All areas are at least good. May be outstanding is some areas and good in others and hence is on balance excellent. Good or above in all areas. Likely to contain minor errors, omissions or additions which prevent the methodology from being outstanding. Overall an excellent methodology.	It is possible that the methodology is outstanding or excellent in some areas and satisfactory in others but on balance is good. Satisfactory or above in all areas. Likely to contain a small number of errors, omissions or additions which prevent the methodology from being excellent. Overall a good methodology.	It is possible that the methodology is good or above in some areas and satisfactory in others. Weak in no more than two areas. Likely to contain a number of errors, omissions or additions which prevent the methodology from being good. Overall satisfactory.	It is possible that the methodology is satisfactory or above in some areas and unsatisfactory in others. Likely to be weak in more than three areas. It might be unsatisfactory in one area but no more. Likely to contain errors, omissions, additions, or misunderstandings which prevent the methodology from being satisfactory. Overall poor.	two or more areas. Likely to contain errors, omissions, additions, or misunderstandings which prevent the design model from being weak. May not be recognisable as a methodology, might have majors errors in content or a combination of the two.	
	16-20	**13-15**	**10-12**	**8-9**	**0-7**	
Implementation (30%)	Excellent in the implementation, comments, indentation, consistency and syntax. Correct following of object-oriented concepts.	Good in the implementation, comments, indentation, consistency and syntax. It is possible that the application is	Satisfactory in the implementation, comments, indentation, consistency and syntax. It is possible that the application is good	Weak in the implementation, comments, indentation, consistency and syntax. It is possible that the application is	Unsatisfactory in the implementation, comments, indentation, consistency and syntax. It is possible that the	

	The work shows particular insight or originality in its approach. Excellent functionalities which identifies the underlying principles behind the problem. All areas are at least good. May be outstanding is some areas and good in others and hence is on balance excellent. Good or above in all areas. Likely to contain minor errors, omissions or additions which prevent the implementation from being outstanding. Overall an excellent implementation.	outstanding or excellent in some areas and satisfactory in others but on balance is good. Satisfactory or above in all areas. Likely to contain a small number of errors, omissions or additions which prevent the implementation from being excellent. Overall a good implementation.	or above in some areas and satisfactory in others. Likely to be weak in no more than two areas. Likely to contain a number of errors, omissions or additions which prevent the implementation from being good. Overall a satisfactory implementation.	satisfactory or above in some areas and unsatisfactory in others. Likely to be weak in more than three areas. It might be unsatisfactory in one area but no more. Likely to contain errors, omissions, additions, or misunderstandings which prevent the implementation from being satisfactory. Still recognisable as an object-oriented application of the problem in focus. Overall poor but satisfactory.	application is weak or above in some areas and unsatisfactory in others. Unsatisfactory in two or more areas. Likely to contain errors, omissions, additions, or misunderstandings which prevent the implementation from being weak. May not be recognisable as an object-oriented application, might have majors errors in implementation or a combination of the two.	
Results (20%)	**16-20**	**13-15**	**10-12**	**8-9**	**0-7**	
	Excellent in the results discussion. The explanation and justification of how it meets specified requirements shows	Good in the results discussion. The explanation and justification of how it meets specified requirements	Satisfactory in the results discussion. The explanation and justification of how it meets specified	Weak in the areas of the results discussion. The explanation and justification of how it meets specified	Unsatisfactory in the areas of the results discussion. It conveys little understanding of solution of the	

	outstanding insight into the issues involved and alternatives available. The presentation clearly and concisely demonstrates a deep understanding of the project implementation. May be outstanding is some areas and good in others and hence is on balance excellent. Good or above in all areas. Likely to contain minor errors, omissions or additions which prevent the results discussion from being outstanding. Overall an excellent results discussion.	shows good understanding into the issues involved and alternatives available. The presentation demonstrates a good understanding of the project implementation. It is possible that the presentation is outstanding or excellent in some areas and satisfactory in others but on balance is good. Satisfactory or above in all areas. Likely to contain a small number of errors, omissions or additions which prevent the results discussion from being excellent. Overall a good results discussion.	requirements covers relevant aspects into the issues involved and alternatives available, but is not outstanding in any respect. It is possible that the presentation is good or above in some areas and satisfactory in others. Likely to be weak in no more than two areas. Likely to contain a number of errors, omissions or additions which prevent the results discussion from being good. Overall a satisfactory results discussion.	required is inadequate. It is possible that the presentation is satisfactory or above in some areas and unsatisfactory in others. Likely to be weak in more than three areas. It might be unsatisfactory in one area but no more. Likely to contain errors, omissions, additions, or misunderstandings which prevent the results discussion from being satisfactory. Overall poor but satisfactory.	problem or no useful explanation and justification of how it meets specified requirements. It is possible that the presentation is weak or above in some areas and unsatisfactory in others. Unsatisfactory in two or more areas. Likely to contain errors, omissions, additions, or misunderstandings which prevent the results discussion from being weak.	
Communication (20%)						

	16-20	13-15	10-12	8-9	0-7	Comments
	Excellent, well-directed presentation, logically and coherently structured. It is free or almost free grammatical errors. The format is clear and consistent with appropriate use of headings and paragraphs. English usage is easily understandable. References and quotations are utilized appropriately to indicate sources.	Good presentation, logically structured. There are occasional spelling and grammatical errors, but the reader does not struggle to interpret the writer's intended meaning. The writing would benefit from the use of organizational tools (e.g. headings, paragraphs) and more consistent use of references to sources.	Satisfactory presentation, well structured. It contains number of spelling and grammatical errors, but the reader does not struggle to interpret the writer's intended meaning. It is possible that the use of organizational tools (e.g. headings and paragraphs) are good or above in some areas and satisfactory in others.	Weak presentation and structure. Spelling and grammatical errors force the reader to struggle to determine the intended meaning. Organizational tools such as headings, paragraphs are used inconsistently. References are not used properly to indicate the sources of material.	Unsatisfactory presentation and structure. Numerous spelling and grammatical errors and a lack of clear consistent organization interfere with the writer's ability to communicate to the key points. The reader frequently cannot determine the intended meaning. There are no references to indicate material taken from other sources.	

Turnitin Report



turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Jun Chao Thean
Assignment title:	Assignment
Submission title:	Turnitin_KDDM_DST_GroupAssignm.
File name:	Turnitin_KDDM_DST_GroupAssignm.
File size:	594.04K
Page count:	12
Word count:	2,965
Character count:	15,681
Submission date:	05-Apr-2021 10:46PM (UTC+0800)
Submission ID:	1550992574

Turnitin_KDDM_DST_GroupAssignment_0124305_0127122

ORIGINALITY REPORT

3%	1%	2%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to American University of Beirut Student Paper	<1%
2	Gaurav Jariwala, Harshit Agarwal, Vrai Jadhav. "Sentimental Analysis of News Headlines for Stock Market", 2020 IEEE International Conference for Innovation in Technology (INOCON), 2020 Publication	<1%