



WEB SCRAPING MODEL FOR HATE SPEECH TRACKING TO ENHANCE HEALTHY CYBERSPACE WITH NATURAL LANGUAGE TOOL KIT USING MACHINE LEARNING ALGORITHM

NWAMINI BARTHOLOMEW TOCHUKWU AND DR. UGAH JOHN OTOZI
DEPARTMENT OF COMPUTER SCIENCE, EBONYI STATE UNIVERSITY,
ABAKALIKI EBONYI STATE. EMAIL: NWAMINIBT@GMAIL.COM

ABSTRACT

Hate speech refers to any speech or expression that attacks an individual or group on the basis of their identity, such as race, religion, gender, sexual orientation, and ethnicity. It can have serious negative effects on individuals and society as a whole, including increased discrimination, violence, and social exclusion. The increasing use of social media and information sharing has given major benefits to humanity. However, this has also given rise to a variety of challenges including the spreading and sharing of hate speech messages. Thus, to solve this emerging issue in social media sites, recent studies employed a variety of feature engineering techniques and machine learning algorithms to automatically detect the hate speech messages on different datasets. Though there have been different algorithms from various authors to solve these challenges of hate speech but needs enhancement in the area of the following which include no study to compare the variety of feature engineering techniques and machine learning algorithms to evaluate which feature engineering technique and machine learning algorithm outperform on a standard publicly available dataset, distinguished abusive language related to racism and sexism together with non-abusive language as well as many of these approaches are largely biased towards detecting content that is non-hate as opposed to detecting and discriminating real hate hateful content, possibly, because the non-hate content may not contain any discriminating features so can't detect hate speech, previous studies showed that a variety of researchers from across the globe are working on hate speech recognition written in different languages such as German, Dutch and English. However, no study provides a comparative study of various features and ML algorithms on the standard dataset that can serve as a baseline study for future researchers in the field of hate speech recognition. This paper introduces a decision tree algorithm to predict hate speech, train and test a model using dataset gotten from tweets from tweeters and clean, validate, and visualize the dataset, as well as removing stop words so as to ensure accurate detection. The model was developed using Python, Jupyter and Anaconda.

KEYWORDS: Hate speech, Machine learning, Decision tree, Nutritional deficiency, Stop words



1.0 INTRODUCTION

In today's ubiquitous society, we experience a situation where digital informing and mediated communication are dominant. The contemporary online media landscape consists of the web forms of the traditional media along with new online native ones and social networks. Content generation and transmission are no longer restricted to large organizations and anyone who wishes may frequently upload information in multiple formats (text, photos, audio, or video) which can be updated just as simple. Especially, regarding social media, which since their emergence have experienced a vast expansion and are registered as an everyday common practice for thousands of people, the ease of use along with the immediacy they present made them extremely popular. In any of their modes, such as microblogging (like Twitter), photos oriented (like Instagram), etc., they are largely accepted as fast forms of communication and news dissemination through a variety of devices. The portability and the multi-modality of the equipment employed (mobile phones, tablets, etc.), enables users to share, fast and effortless, personal or public information, their status, and opinions via the social networks. Thus, communications nodes that serve many people have been created minimizing distances and allowing free speech without borders; since more voices are empowered and shared, this could serve as a privilege to societies. However, in an area that is so wide and easily accessible to large audiences many improper intentions with damaging effects might be met as well, one of which is hate speech.

Social Network Systems are a great way for online users to stay in touch and exchange information regarding their everyday interests and activities, as well as publish and access documents, images, and videos. Unfortunately, these are the prime place for harmful information to spread. While SNSs facilitate communication and information sharing, they are sometimes used to launch problematic campaigns against certain organizations and individuals. Cyber bullying, hate speech to self-harm, and sexual predatory behavior are only a few of the serious consequences of large-scale internet offensives. With the rise of social media and its unfortunate usage for hate speech, automatic hate speech identification has become a critical challenge. The number of hostile actions is rising as a result of the huge rise in user generated web content, particularly on social media networks where anybody may make a comment freely and without any restrictions. People may rapidly express their opinions, including hate speech, via social media technology, which subsequently spreads widely and becomes viral if the issues addressed are 'interesting'. It has the potential to cause conflict amongst social groupings. Obviously, there are many more hate statements on numerous social media platforms. Twitter is a prominent social networking platform in Indonesia. Twitter and other social media and microblogging online services allow users to view and analyse user tweets in near real time. Because Twitter users are more inclined to convey their emotions about an event by publishing a tweet, it provides a natural source of data for hate speech analysis [3]. This research can aid in the early

detection of hate speech, preventing it from spreading widespread. It's also beneficial for content screening and detecting illegal activity early on [5]. The manual method of identifying nasty tweets is inefficient and unsalable. As a result, an automated method for detecting hate speech in tweets is required.

1.1 Statement of the Problem

The increasing use of social media and information sharing has given major benefits to humanity. However, this has also given rise to a variety of challenges including the spreading and sharing of hate speech messages. Thus, to solve this emerging issue in social media sites, recent studies employed a variety of feature engineering techniques and machine learning algorithms to automatically detect the hate speech messages on different datasets. Though there have been different algorithms from various authors to solve these challenges of hate speech but needs enhancement in the area of the following which include:

1. There is no study to compare the variety of feature engineering techniques and machine learning algorithms to evaluate which feature engineering technique and machine learning algorithm outperform on a standard publicly available dataset.
2. Distinguished abusive language related to racism and sexism together with non-abusive language as well as many of these approaches are largely biased towards detecting content that is non-hate as opposed to detecting and discriminating real hate hateful content, possibly, because the non-hate content may not contain any discriminating features so can't detect hate speech.
3. Previous studies showed that a variety of researchers from across the globe are working on hate speech recognition written in different languages such as German, Dutch and English. However, no study provides a comparative study of various features and ML algorithms on the standard dataset that can serve as a baseline study for future researchers in the field of hate speech recognition.

1.2 Research Aim/Specific Objectives

The aim of this paper is to develop a robust web scraping model for hate speech tracking to enhance healthy cyberspace with Natural Language Tool Kit using machine learning algorithm. The following are the specific objectives:

1. Use a decision tree algorithm to predict hate speech
2. Train and test a model using dataset gotten from tweets from tweeters.
3. Clean, validate, and visualize the dataset, as well as removing stop words so as to ensure accurate detection.

1.3 Scope of the Study

This paper hate speech detection model focuses on detection of hate speech which is limited to a negative expression about individuals by using decision tree algorithm. In this paper the input is first preprocessed by removing stop words and stemming the words. Then, it extracts relevant features from the preprocessed text using the Term Frequency-Inverse Document Frequency (TF-IDF) method. Finally, the decision tree algorithm is trained on the extracted features to classify the input text as hate speech or non-hate speech.

1.4 Review of Related Works

These days, hate speech is very common on social media. Therefore, in previous years, some of the researchers have applied a supervised ML-based text classification approach to classify hate speech content. Different researchers have employed different variety of feature representation techniques namely, dictionary-based, Bag-of-wordsbased, N-grams-based, TFIDF-based and Deep-Learning-based [2].

[5] Employed a dictionary-based approach to identify cyber hate on Twitter. In this research, they employed an N-gram feature engineering technique to generate the numeric vectors from the predefined dictionary of hateful words. The authors fed the generated numeric vector to ML classifier namely, SVM and obtained a maximum of 67% F-score. [8] Also used a dictionarybased approach for the automatic detection of racism in Dutch Social Media. In their study, the authors used the distribution of words over three dictionaries as features. They fed the generated features to the SVM classifier. Their experimental results obtained 0.46 F-Score. [5] Used ML-based classifier to classify hate speech in web forums and blogs. The authors employed a dictionary-based approach to generate a master feature vector. The features were based on sentiment expressions using semantic and subjectivity features with an orientation to hate speech. Afterward, the authors fed the masters feature vector to a rule-based classifier. In the experimental settings, the authors evaluated their classifier by using a precision performance metric and obtained 73% precision.

Nonetheless, the combination of dictionary-based and ML approaches showed a good result. However, the major disadvantage of such type of approach is that it requires a dictionary, based on the large corpus to look for domain words. To overcome this drawback, many of the researchers have used a BOW-based approach which is similar to a dictionary-based approach but the word features are obtained from training data and not from the predefined dictionaries. [10] used the supervised ML approach to classify the racist text. To convert the raw text into numeric vectors, the authors employed a bigram feature extraction technique. The authors used bigram features, with the BOW feature representation technique. They used the SVM classifier to perform experimental results. In their results, they achieved 87% accuracy. [1] employed an



ML-based approach to the automatic detection of racism against black in the twitter community. In their research, they employed unigram with the BOW-based technique to generate the numeric vectors. The authors fed the generated numeric vector to the Naïve Bayes classifier. Their experimental results obtained a maximum of 76% accuracy. [2] classified hate speech on twitter. In their research, they employed BOW features. The authors fed the generated numeric vector to the Naïve Bayes classifier. Their experimental results showed a maximum of 73% accuracy. Nevertheless, BOW showed better accuracy in social network text classification. However, the major disadvantage of this technique is, the word-order is ignored and causes misclassification as different words are used in different contexts. To overcome this limitation, researchers have proposed an N-grams-based approach [4].

1.5 Proposed System

The proposed system is a machine learning model using decision tree algorithm to detect hate speech, and takes in text input to detect hate speech. The system takes in input and classified them into hate speech, offensive and non-offensive. Gather a labeled dataset of hate speech and non-hate speech examples. Text Preprocessing, clean and preprocess the text data by removing noise, formatting inconsistencies, and irrelevant information. Perform tasks like tokenization, stemming, and removing stop words. Represent the text data using features like bag-of-words, TF-IDF, or word embedding. Build a decision tree classifier using the labeled dataset and the extracted features. Train the decision tree classifier using the labeled dataset. Evaluate the performance of the decision tree classifier using metrics like accuracy, precision, recall, and F1-score on a testing set. Fine-tune the hyper parameters of the decision tree classifier, such as the maximum depth or minimum samples per leaf, to optimize its performance. Deploy the trained decision tree model for hate speech detection in real-time applications or integrate it into existing moderation systems.

1.6 System Flowchart of the Proposed System

Figure 1 describe the processes involve in the model and the various users to the model and the flow of data.

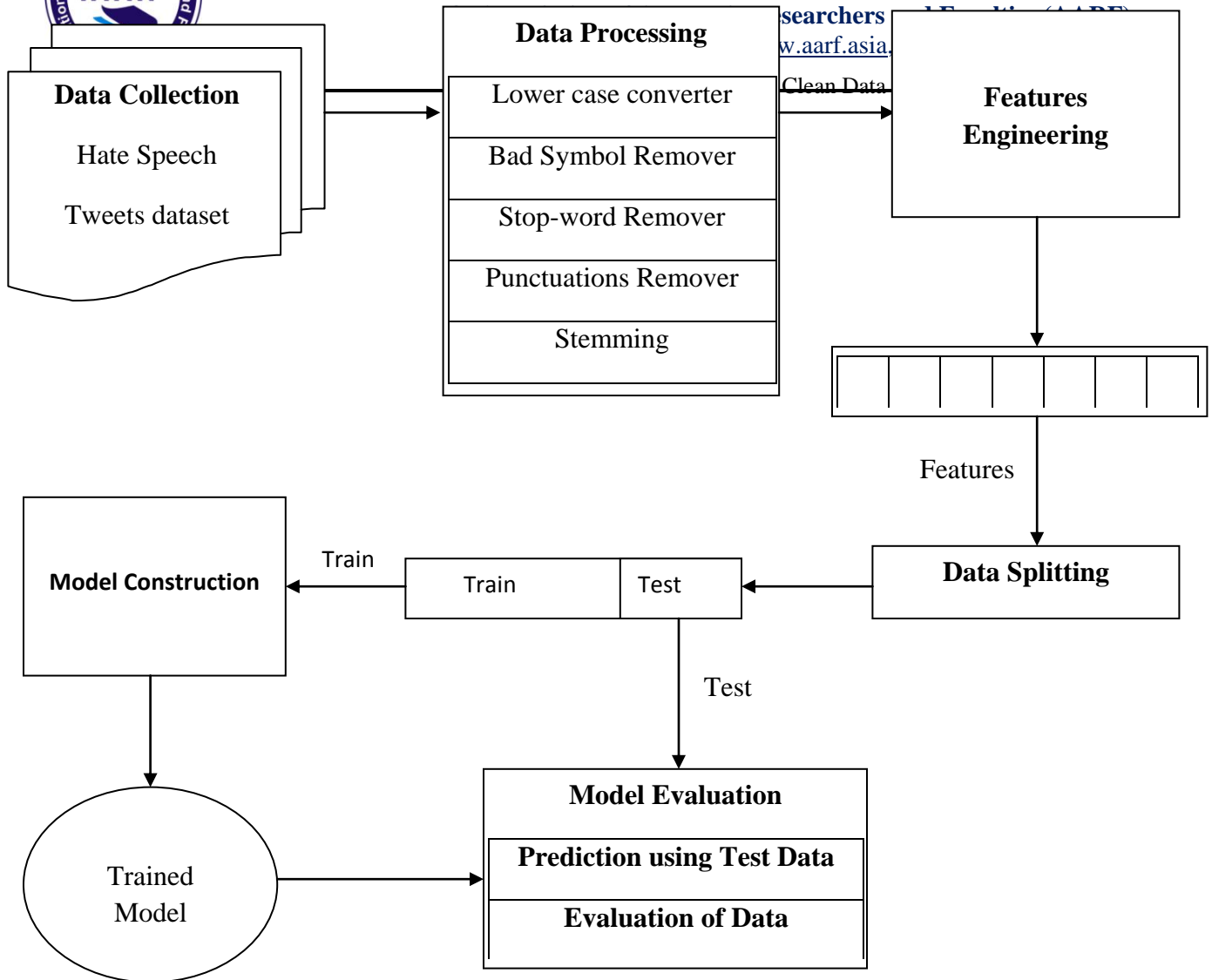


Figure 1: System Flowchart of the Proposed System

Figure 1 explains various sections in the diagram. In data collection section in this research study, dataset containing publicly available hate speech tweets dataset was collected. This dataset is compiled and labeled by CrowdFlower. In this dataset, the tweets are labeled into three distinct classes, namely, hate speech, not offensive, and offensive but not hate speech. This dataset has 24783 numbers of tweets.

1.7 Program Flowchart of the Proposed System

The program flowchart diagram below explained the complete process and function of the proposed system. The diagram is showed below.

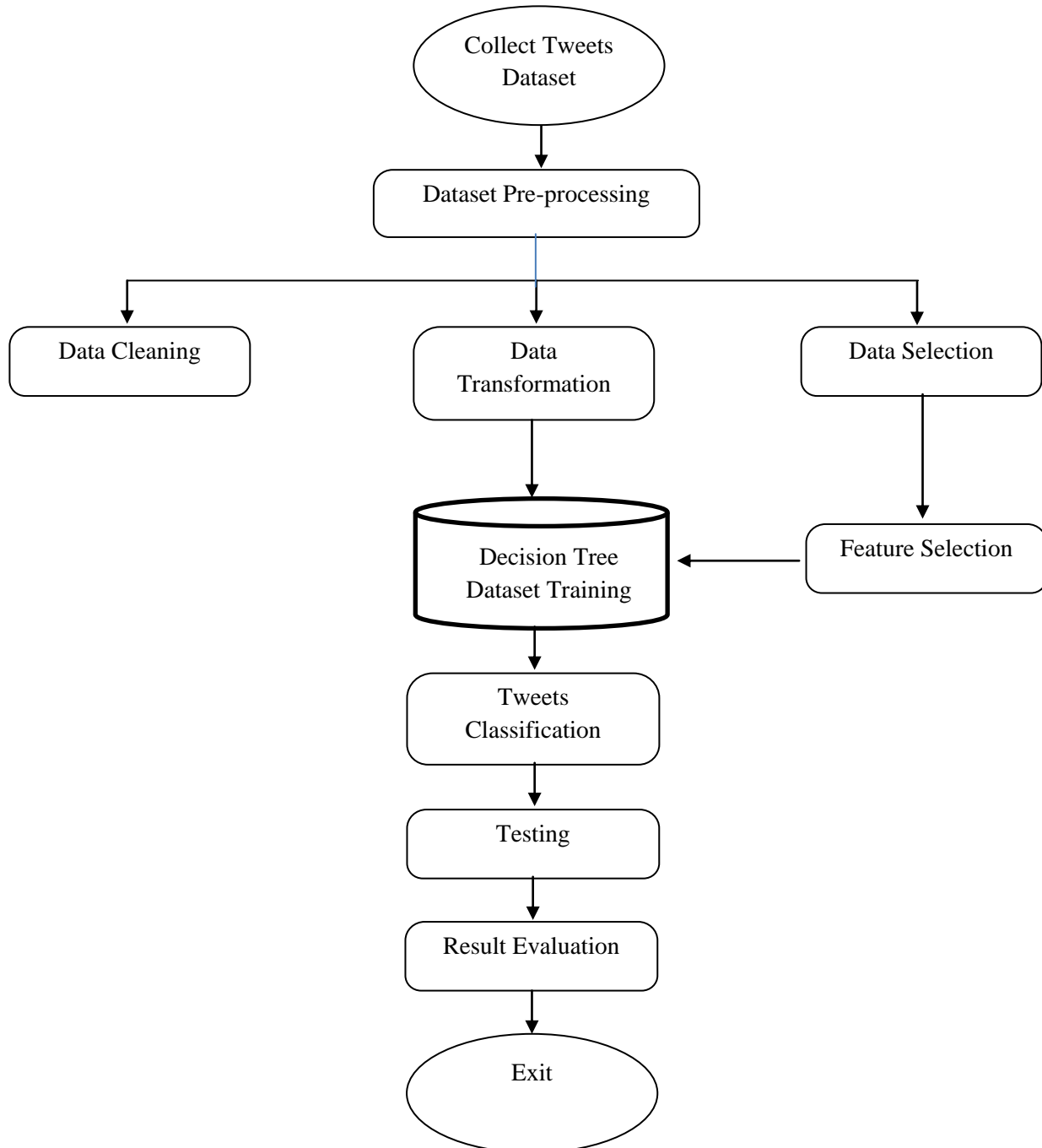


Figure 2: Program Flowchart of the Proposed System

Figure 2 explains the processes involved in training and testing the model. Here dataset is collected after which there will be data pre-processing then followed by three aspect which are data cleaning, data transformation, data selection then the algorithm which is used to train the model then tweet classification, testing and result is evaluated.

1.8 Design Architecture of the proposed system

Architectural design in software engineering is about decomposing the system into interacting components. It is expressed as a block diagram defining an overview of the system structure, features of the components, and how these components communicate with each other to share data. It identifies the components that are necessary for developing a computer-based system and communication between them i.e. relationship between these components. It defines the structure and properties of the components that are involved in the system and also the interrelationships between these components. Figure 2 is the block diagram of the proposed system that explains the processes.

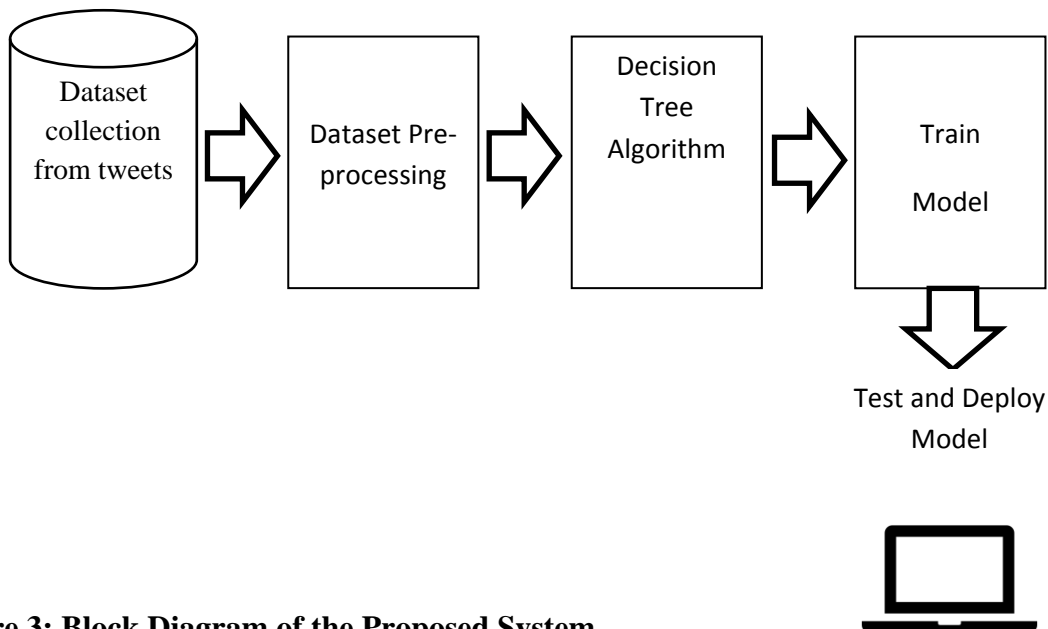
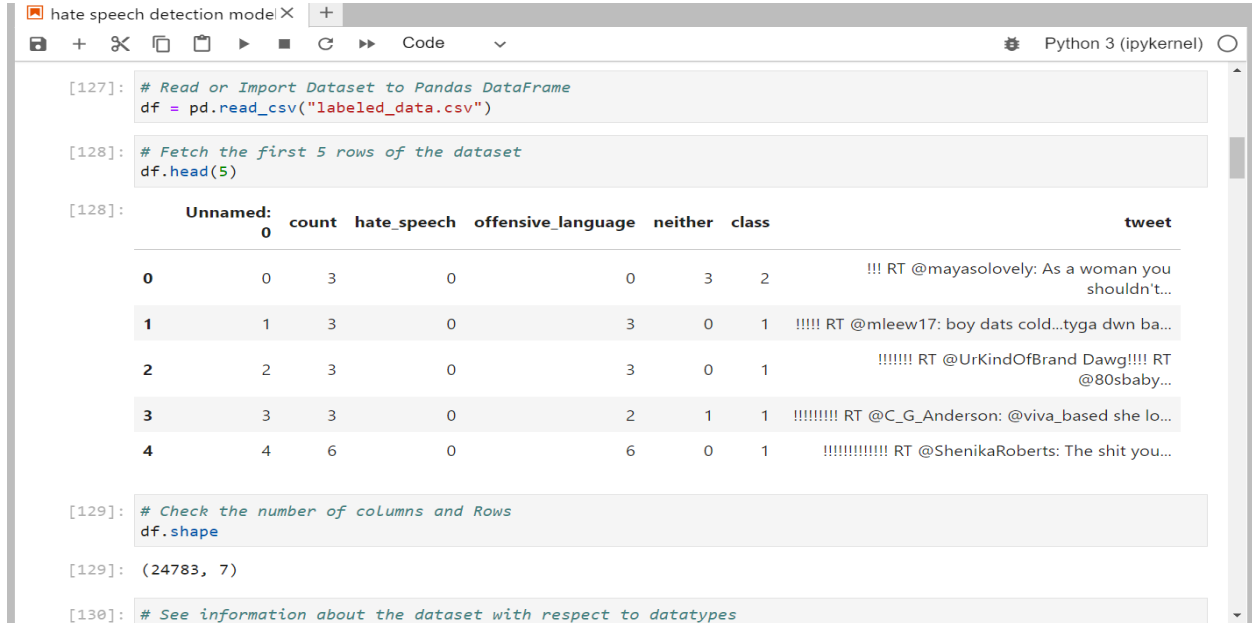


Figure 3: Block Diagram of the Proposed System

Here 24789 dataset is gotten from tweeter containing hate speech, offensive and non-offensive dataset respectively, this data is trained using decision tree algorithm and the model is tested and deployed.

1.9 Implementation Phase

This phase describes how the model was built and implemented and also how the dataset was trained.



```

[127]: # Read or Import Dataset to Pandas DataFrame
df = pd.read_csv("labeled_data.csv")

[128]: # Fetch the first 5 rows of the dataset
df.head(5)

[128]:      Unnamed: 0  count  hate_speech  offensive_language  neither  class  tweet
0              0      3            0                    0         3      2  !!! RT @mayasolovely: As a woman you shouldn't...
1              1      3            0                    3         0      1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2              2      3            0                    3         0      1  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3              3      3            0                    2         1      1  !!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4              4      6            0                    6         0      1  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...

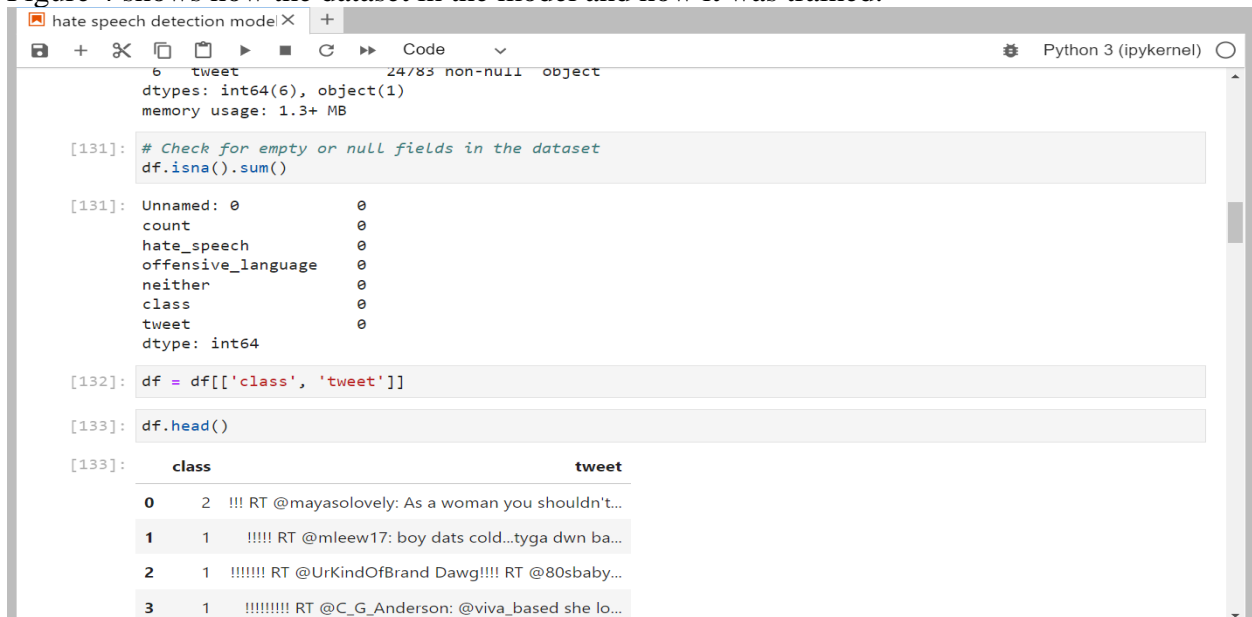
[129]: # Check the number of columns and Rows
df.shape

[129]: (24783, 7)

[130]: # See information about the dataset with respect to datatypes
  
```

Figure 4: Showing dataset imported into Jupyter environment

Figure 4 shows how the dataset in the model and how it was trained.



```

6 tweet 24783 non-null object
dtypes: int64(6), object(1)
memory usage: 1.3+ MB

[131]: # Check for empty or null fields in the dataset
df.isna().sum()

[131]: Unnamed: 0      0
count          0
hate_speech    0
offensive_language 0
neither        0
class          0
tweet         0
dtype: int64

[132]: df = df[['class', 'tweet']]

[133]: df.head()

[133]:      class  tweet
0      2  !!! RT @mayasolovely: As a woman you shouldn't...
1      1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2      1  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3      1  !!!!!!!! RT @C_G_Anderson: @viva_based she lo...
  
```

Figure 5: Showing the shape (number of rows and columns) of the dataset, and the datatype of each of the column.

Figure 5 shows how the dataset was trained in rows and columns.

```

hate speech detection mode X +
Python 3 (ipykernel)
clr = DecisionTreeClassifier()

[149]: # Train Model with dataset
clr.fit(X_train, y_train)

[149]: DecisionTreeClassifier
DecisionTreeClassifier()

[150]: clr.predict(X_test)

[150]: array([1, 1, 1, ..., 1, 2, 1], dtype=int64)

[151]: # Get model testing accuracy
# calculating the accuracies
print("Testing Accuracy :", clr.score(X_train, y_train))

Testing Accuracy : 0.9975213281069863

[152]: # Get model testing accuracy
# calculating the accuracies
print("Testing Accuracy :", clr.score(X_test, y_test))

Testing Accuracy : 0.886079354404842

[153]: # Get model testing accuracy
# calculating the accuracies
print("Testing Accuracy :", clr.score(X_test, y_test))

Testing Accuracy : 0.886079354404842
    
```

Figure 6: Showing Decision tree Model using training dataset

Figure 6 shows the how decision tree algorithm was used for the prediction of hate speech.

```

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)
model.fit(X_train, Y_train)

# predicting the x-test results
y_pred = model.predict(X_test)
y_pred

Out[30]: array([1, 0, 2, ..., 0, 2, 2])

In [31]: # calculating the accuracies
print("Training Accuracy :", model.score(X_train, Y_train))

Training Accuracy : 0.9996986136226642

In [34]: from sklearn.metrics import confusion_matrix

# creating a confusion matrix
cm = confusion_matrix(Y_test, y_pred)

# printing the confusion matrix
print(cm)

[[281 28 44]
 [ 42 80 71]
 [ 14 23 523]]

In [35]: # Save model for Deployment
import pickle as pk
    
```

Figure 7: Showing training accuracy of 100% as well as testing accuracy of 75%



1.10 System Requirement

The system requirements provide an overview of the least criteria the hardware and software must obtain before its usage. It is to provide a detailed overview of the hardware and software product, its parameters and goals to the wide range of users. This describes the project's target audience and its user interface, hardware and software requirements.

1. Hardware Requirement

The hardware requirements are as follows:

- i. A minimum of 150GB hard disk drive.
- ii. At least a Pentium III 800 MHz MMX Intel Processor
- iii. Minimum of 4GB Random Access Memory.
- iv. A CD-ROM Drive.
- v. A super Video Graphic Adapter (SVGA) Monitor.
- vi. A stabilizer and an uninterruptible Power Supply Unit (UPS).
- vii. A keyboard and a mouse.

2. Software Requirement

The minimum software requirements for running this application are:

1. Microsoft Windows (XP, Vista, Windows 7, Windows 8).
2. Anaconda 2.1
3. Jupyter 2.1 and above
4. Python 2.7 and above
5. Browsers (Mozilla Firefox 9.2.1/Google Chrome 21.0.1180.81 or any other browser that supports HTML5)
6. Anti-virus

1.11 Conclusion

The outcomes from this paper hold practical importance because this will be used as a baseline study to compare upcoming researches within different automatic text classification methods for automatic hate speech detection. Furthermore, this study also holds a scientific value because this study presents experimental results in form of more than one scientific measure used for automatic text classification. Our work has two important limitations. First, the proposed ML model is inefficient in terms of real-time predictions accuracy for the data. Finally, it only classifies the hate speech message in three different classes and is not capable enough to identify the severity of the message. Hence, in the future, the objective is to improve the proposed ML model which can be used to predict the severity of the hate speech message as well. Moreover, to improve the proposed model's classification performance two approaches will be used. First, the lexicon-based techniques will be explored and assessed by comparing with other current state-of-the-art results. Secondly, more data instances will be collected, to be used for learning the classification rules efficiently.

REFERENCES

- [1] Burnap P., and Williams M. L., (2015) "Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making," *Policy & Internet*, vol. 7, no. 2, pp. 223–242, 2015.
- [2] Fortuna P., and Nunes S., (2018) "A Survey on Automatic Detection of Hate Speech in Text," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–30, 2018. Protected characteristics," *EPJ Data Sci.*, vol. 5, no. 1, 2016.
- [3] George S., and Joseph K., (2014) "Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature," *IOSR J. Comput. Eng.*, vol. 16, no. 1, pp. 34–38, 2014.
- [4] Ghwanmeh S. H., Kanaan G., Al-Shalabi R., and Rabab'ah S., (2009) "Enhanced Algorithm for Extracting the Root of Arabic Words," 2009 Sixth Int. Conf. Comput. Graph. Imaging Vis., pp. 388–391, 2009.
- [5] Irfan R. (2015) "A survey on text mining in social networks," *Knowl. Eng. Rev.*, vol. 30, no. 2, pp. 157–170, 2015.
- [6] Khoja S., and Garside R., (1999) "Stemming arabic text," Lancaster, UK, Comput. Dep. Lancaster Univ., 1999.



- [7] Pandarachalil S., Sendhilkumar R., and Mahalakshmi G. S., (2015) “Twitter Sentiment Analysis for Large- Scale Data: An Unsupervised Approach,” *Cognit. Comput.*, vol. 7, no. 2, pp. 254–262, 2015.
- [8] Salawu S., He Y., and Lumsden J., (2017) “Approaches to Automated Detection of Cyberbullying: A
- [9] Sun S., Luo C., and Chen J., (2017) A review of natural language processing techniques for opinion mining systems, vol. 36, no. November 2016. 2017.
- [10] Waseem Z., and Hovy D., (2016) “Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter,” *Proc. NAACL Student Res. Work.*, pp. 88–93, 2016.