Prediction of Hatred Ness in Election Candidates

Submitted in partial fulfillment of the requirements for the degree of

M.Tech Integrated Computer Science and Engineering (In Collaboration with Virtusa)

by

G. Venkata Srikar Gupta (19MIC0007)

N. Ganesh Surya Vamsi (19MIC0027)

K. Arun Kiran (19MIC0036)

B. Venkata Nagaraju (19MIC0089)

Under the guidance of

Prof. / Dr. Vishnu Srinivasa Murthy

SCOPE

VIT, Vellore.

April, 2023

DECLARATION

I hereby declare that the thesis entitled "Prediction of Hatred Ness in Election Candidates" submitted by me, for the award of the mini-project in M.Tech. (Integrated) Computer Science and Engineering to VIT is a record of bonafide work carried out by me under the supervision of Dr Vishnu Srinivasa Murthy.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore
Date : 10-04-2023

Signature of the Candidate

Srikar Gupta

Ganesh Surya Vamsi

Arun Kiran

Nagaraju

CERTIFICATE

This is to certify that the thesis entitled "PREDICTION OF HATREDNESS IN ELECTION CANDIDATES" submitted by our team VIT, for the award of the Mini- project of M.Tech. (Integrated) Computer Science and Engineering, is a record of bonafide work carried out by him / her under my supervision during period, 01. 12. 2018 to 30.04.2019, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date :

Signature of the Guide

Internal Examiner

External Examiner

Head of the Department

(Dr. Anand Kumar)

Programme

(MTech Integrated)

ACKNOWLEDGEMENTS

This is to declare that the project entitled “Prediction of Hatred Ness in Election Candidates” is an original work done by undersigned, in partial fulfillment of the requirements for the degree “Integrated MTech in Computer Science and Engineering” at School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Vellore.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or University.

Student Name

Srikar Gupta

Ganesh Surya Vamsi

Arun Kiran

Nagaraj

Executive Summary

Prediction of Hatred Ness in Election Candidates is useful for finding the Nominated person who is best and useful for society and useful for people. Since considering the two persons from same party you cant keep voting for the people for knowing who is the best. So we are considering based on Hatred Ness of particular person using YouTube comments . We will extract the YouTube Comments of two members and load into the dataset using web scrapping. Considering different ML Approaches and one Deep Learning Model we are choosing or predicting the best one who is useful for developing the society.

CONTENTS

Page No.

1. INTRODUCTION

1.1 Theoretical Background

In present situation of digital world, the data consists of various forms. Almost every data will be available digitally. With respect to Hate comments , Twitter Dataset is available globally. Considering various types of ML Algorithms and Deep Learning Model Classifying the classes into Hate or Non-Hate is simple. Many of that Datasets are just classifying the classes into only two categories – Hatred and Non Hatred.

To implement the Hatred Ness Prediction we are applying various Machine Learning Algorithms and LSTM Model for the datasets we extracted from YouTube. The main idea of our project is to visualize the classes of Hatred Ness and Non Hatred Ness to get the clear information about the particular member of political party.

1.2 Motivation

Predicting the level of hatred ness in election candidates can have significant implications for the electoral process and the overall political climate of a society. By using various ML models, it is possible to analyse and predict the extent to which a candidate's campaign may rely on negative rhetoric or attack their opponents. This information can be used to inform voters, political analysts, and other stakeholders about the potential impact of a particular candidate on the electoral process and the society as a whole. Additionally, such predictions can be used to identify candidates who rely on divisive tactics and promote a more constructive and positive political discourse. Ultimately, by leveraging the power of ML models to predict the level of hatred ness in election candidates, we can promote a more informed and engaged electorate and contribute to building healthier and more resilient democracies.

There is no common project that have considered the data of YouTube Comments taken from particular political members. Using Web Scrapping we will extract the YouTube comments and load into a dataset of giving different classes. So Prediction among two members is done newly with newly taken dataset and the political leader can select the particular member for the party based on people opinion of Hatred Ness.

1.3 Aim of the Proposed Work

The main aim of our project is to find the person who was helpful for people and for developing the Society. So the Government will consider the YouTube comments and consider who had high Hatred ness will not be eligible for a place in the party. We well consider the YouTube data and we will extract the Data using web scrapping.

We will consider different comments and divide into five classes - dislike, support, neutral, hate and Insult . First step ee will do the preprocessing Technique it ll remove the stop word removal and tokenization process. We will consider different ensemble techniques and find the accuracy to select the best result. We will apply

Training and Testing for given data since we can't train for more comments. For each result we got different values of Accuracy, precision and recall. Based on that they ll select the candidate who is useful for future.

1.4 Objectives of Proposed Work

- The main objective of our project is to find the person who have less Hatred Ness among the people and useful for developing the society.
- Predicting the comments into various classes – Hatred or Non Hatred.
- Finding the best ML Algorithm which suits for the project based on Evaluation Score.

2.Literature Survey

2.1 **Survey of the Existing Models/Work**

Our first survey is about the Hate Towards the Political Opponent: A Twitter Corpus Study of the 2020 US Elections on the Basis of Offensive Speech and Stance Detection . The dataset contains the Corpus of 3000 Tweets. It contains the features favorable, against, mixed and neutral instances to tell about the opinion of both the election candidates. The methodology used is Trained BERT Baseline Classifier and Tensor flow for Stance Detection. The limitations for this survey is that the used algorithm is only helpful for finding the support and challenging at finding the against people.

Second survey is about The Datafication of Hate: Expectations and Challenges in Automated Hate Speech Monitoring. The dataset contains the Public social media messages of all candidates which were collected from social media platform APIs and classified using a machine learning system created for the project, and sent to the NDO for manual checking and for potential follow-up procedures. The features of this paper is Based on Hate Speech (Images, Memes) and Approaches(Word-lists, BOW Approaches and ngrams) divided into binary classification n – hate / not hate. The methodology used here is Bag of words+ SVM. The limitations for this survey is that here many APIs are taken sometimes it is difficult to find accuracy which is good/ which is bad and difficult to identify that word is hate / good.

Third survey is about Hate Speech and Image Sharing in the 2020 Singaporean Elections. The dataset contains the Twitter data surrounding the 2020 Singaporean elections using the Twitter REST API. Search terms included social hashtags around the

election like `#sgelections2020' and candidate specific handles like `@wpsg' or `@jamuslimitations'. The complete dataset contained 240K tweets and 42K unique users. Based on this dataset, images were downloaded from tweets if they were present. A total of 52K images were collected. The features of this dataset is Hate speech, category of interest, offensive speech, regular speech to classify the people's opinion. The methodologies used here is Image clustering Euclidean clustering ,Bot detection Bot hunter algorithm ,Hate speech detection Machine Learning Algorithm, Hierarchical Regression Modelling Using the variables produced through the tools of quantified the extend which bot driven image sharing predicted higher levels of hate speech. The limitations for this survey is that sampling Twitter data remains limited by API generalizability issues, suggesting caution in extrapolating findings to wider contexts. The selection of hashtags may not have been comprehensive and some hashtags corresponded to events happening in the same time zone.

Fourth survey is about Automatic Hate Speech Detection in English-Odia Code Mixed Social Media Data Using Machine Learning Techniques. The data is collected from Facebook Posts and comments from public pages from April 2018 – April 2019. Data took from the pages where the followers are more than 50000 and consists of 50 posts. The features of this survey is divided into classes like Hate, Offensive and neither. The feature extraction process uses three methods, Ngram, TDIDF, and word2vec.The methodologies used here is SVM, Naïve Bayes, Random Forest. The limitations of this survey is that it is time Consuming process . Since it got two different results in binary classification and ternary classification difficult to predict which suits the best.

Fifth survey is about selecting and combining complementary feature representations and classifiers for hate speech detection. Data took from the twitter of various posts and tweets available in twitter. They considered four different detection datasets. The features of this survey is Hate, offensive, nonoffensive Zerak wassem – racism, sexism, none TD + ZW – hate, offensive, non offensive Hate Evaluation – hateful, non hateful to categories the hatespeech. The methodologies used here is SVM, Logistic Regression, Naïve Bayes, MLP, KNN, CNN, TFIDF,word2vec.The limitations for this survey is that it Classifying the Algorithm / Model which suits the best is difficult to find since they had done many ML Classifications and used Deep Learning Models.

Sixth survey is about s countering hate speech against journalists on social media. Data is collected from Twitter of various Accounts and Tweets of different users in five different Languages. Hate Speech is classified into two classes : Positive and negative. The methodologies used here is CNN, CNN+GRU, LSTM. Since they had taken 5 different languages to classify it may took more time for training and may not expect perfect training result since training process is difficult to train another Languages.

Seventh survey is about "Deep Explainable Hate Speech Active Learning on Social-Media Data". The datasets consist of hate speech tweets split evenly between women and immigrants. Each split contains 9,100, 1,000, and 2,971 tweets for training, development, and testing, respectively. The features of this dataset is classified into two different classes offensive and non – offensive. The methodologies used here is LSTM, RNN, GRU, LSTM+GRU. The limitations for this project is since they have used Deep Learning Model it is difficult and more time consuming to predict if have more number of hidden layers to calculate the activation function at every stage.

Eighth survey is about ML-based Offensive Tweet Accuracy Detector on Social Media. The dataset consists of ID of the twitter users and ,content of twitter users .The features used here is Hate speech Detection from the twitter dataset. The methodologies used here is machine learning(DT,SV M,RF,NB). The proposed method achieves only accuracy maximum of 79 percent.

Ninth survey is about Automatic Hate Speech Detection using Machine Learning. Here data was considered in publicly available datasets. Methods used here is Bigram, Word2dec Doc2vec, Logistic regression , support vector machine ,Naïve Bayes, K-NN. The limitations of this survey is the proposed ML model is inefficient in terms of real time prediction s accuracy for the data. It only classifies the hate speech message in three different classes and is not capable enough to identify the severity of the message.

Tenth survey is about "Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media". Here, data is taken from the social media(Facebook ,You Tube, instagram, twitter). The features used here is simple surface features, word generalization, sentiment analysis, lexical resources, linguistic features, knowledge based features, metainformation and multimodal information. The methodologies used

here is SVM,NB,CNN, LSTM,RNN. The limitations here is that no experiment was conducted with a given dataset.

Eleventh survey paper is about Detecting Hate Speech and Offensive Language using Machine Learning. The data is taken from twitter dataset. The features of this survey is It extract the ngram(unigram,bigram) and Tf , Idf. The methodologies used here is Support vector machine,linear regression,naivebayes. The limitations of this survey is .results can be further improved ,model does not account for negative words present in a sentence.

2.2 Summary/Gaps identified in the Survey

The summary gaps in the survey is that mainly they used twitter datasets only or any datasets. They didn't extracted any datasets. Also they are using a single datasets and there is no use of comparing different datasets. Also the hate speech is not divided based on the positive words ,that is they have neutral or not. Also they are using all algorithms but not the great one came from the project.

3.Overview of the Proposed System

3.1 Introduction and Related Concepts

The project prediction of hatred ness in election candidates is about predicting the hatred ness against election candidates using the opinions of the people using the YouTube dataset.

Hatred ness meaning- the state or condition of being full of hatred, intense dislike, or animosity towards someone or something. Hatred ness could be a feeling or an attitude of hostility and aversion towards a particular person, group, or ideology. It is generally considered a negative emotion that can have harmful effects on individuals and society as a whole.

This project is doing to examine the hatred ness of the election candidates which are in the same party itself, which removes the dilemma in the election party for making stand of a single leader from the mandal or state for one position.

This is done by extracting the public opinions from the YouTube dataset and classifying them into five categories in which first two are classified into hatred and last three are classified into non-hatred.

Classification of comments:

(1)-Insulting the candidate- Insulting the candidate refers to making disrespectful, offensive or derogatory remarks about a person who is running for a political office or position. This type of behavior is often seen as a form of negative campaigning, which involves attacking one's opponents rather than focusing on their policies or qualifications for the position.

(2)-Disliking the candidate- "Disliking the candidate" means that you have negative feelings towards a particular candidate, either in an election or for a job position, and you do not want to vote for or hire that person. This dislike can stem from a variety of reasons such as disagreements with the candidate's policies, negative personal experiences with the candidate.

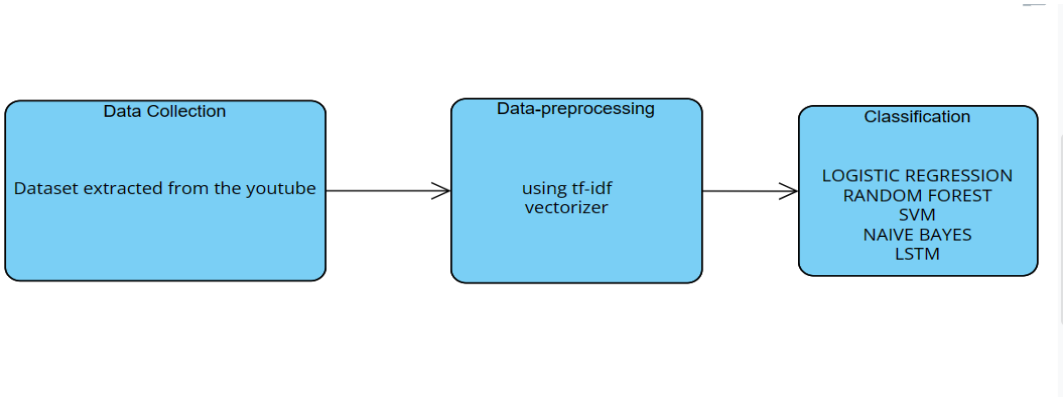
(3)-Neutral opinion- "neutral opinion towards a candidate" would suggest an unbiased assessment or evaluation of the candidate's qualifications, track record, policies, and statements. A neutral opinion would not be influenced by personal biases, emotions, or political affiliations. It would focus solely on providing an objective analysis of the candidate's strengths, weaknesses, and suitability for the position they are seeking.

(4)-Giving support-Support towards the candidate means if any one like him or his ideas or his things which are done in the public want to give support.

(5)-Meaningless comment-Useless comment or unwanted comments.

From this percentages we are going to examine the percentages of the classifications through the visualizations and draws the person who can be eligible to participate from that area.

3.2 Proposed algorithm with Flow chart



PREPROCESSING

TF-IDF Vectorizer is a commonly used tool in natural language processing (NLP) that is used to convert a collection of raw textual data into numerical features that can be used in machine learning algorithms.TF-IDF stands for "Term Frequency - Inverse Document Frequency." It is a statistical

measure that evaluates how relevant a word is to a document in a collection or corpus.

The TF-IDF value increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. This helps to identify words that are most unique and relevant to a particular document while filtering out common and less informative words.

CLASSIFICATION

Logistic regression is a popular supervised learning algorithm used in machine learning for binary classification problems. It is used to predict a binary outcome (e.g., 0 or 1, true or false, yes or no) based on one or more independent variables or predictors.

The goal of logistic regression is to model the probability of a binary outcome given a set of input variables. It does this by fitting a logistic function to the training data. The logistic function is an S-shaped curve that maps any real-valued input to the range [0, 1]. The logistic regression algorithm uses this curve to model the probability of the positive class (i.e., the outcome of interest) given the input variables.

Random Forest is a popular ensemble learning algorithm used in machine learning for both classification and regression problems. It is a combination of multiple decision trees, where each tree is trained on a randomly sampled subset of the training data and features.

Naive Bayes is a popular probabilistic algorithm used in machine learning for classification and regression problems. It is based on Bayes' theorem and the assumption of independence between the features.

In classification, the goal is to predict the class label of an input instance based on its feature values. Naive Bayes works by calculating the probability of each class label given the feature values of the input instance. It does this by applying Bayes' theorem, which states that the probability of a hypothesis (in this case, a class label) given some evidence (in this case, the feature values) is proportional to the probability of the evidence given the hypothesis times the prior probability of the hypothesis.

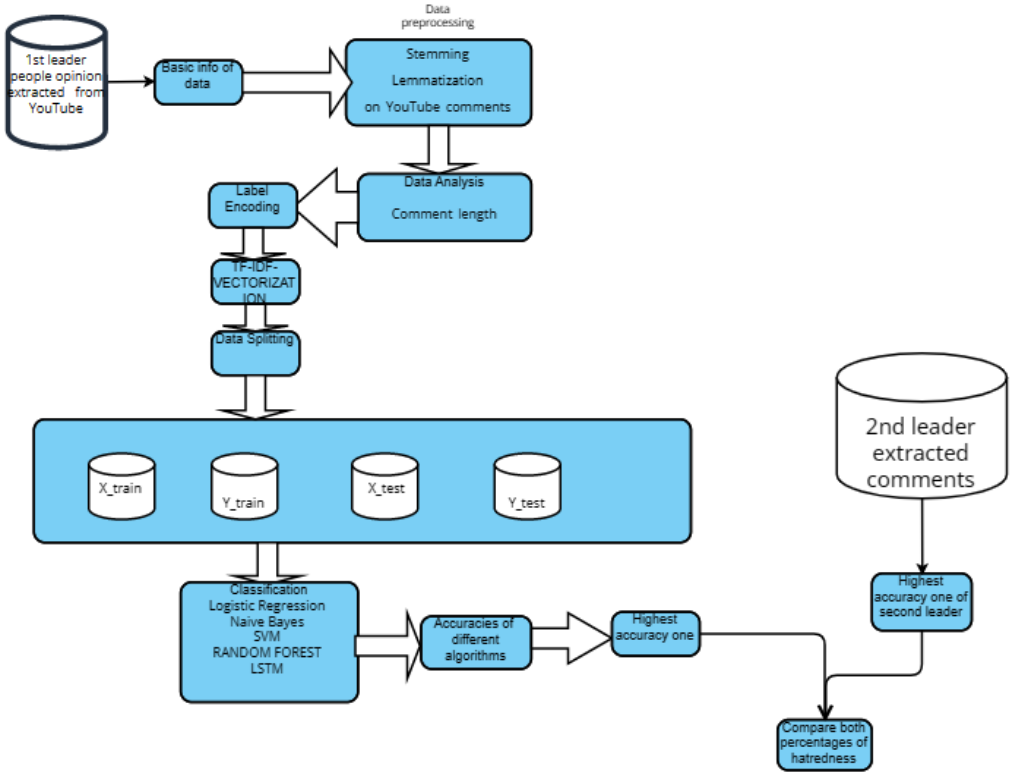
Support Vector Machines (SVMs) are a powerful and widely used algorithm in machine learning for classification, regression, and outlier detection. SVMs find a hyperplane or a set of hyperplanes in a high-dimensional space that best separate the different classes.

The main idea behind SVMs is to find a decision boundary that maximizes the margin between the two classes. The margin is defined as the distance between the decision boundary and the closest data points from each class. The larger the margin, the more robust the classifier will be to new data points.

LSTM stands for Long Short-Term Memory, and it is a type of recurrent neural network (RNN) architecture used in deep learning. Unlike traditional RNNs, which suffer from the problem of vanishing gradients, LSTMs can maintain and propagate information over long sequences of inputs.

LSTMs are designed to handle sequential data, such as time series or natural language text. They have a complex structure that allows them to selectively remember or forget information from previous time steps, making them well-suited for tasks that require long-term memory, such as language translation or speech recognition.

3.3 Detailed description of the proposed system



First of all we have to extract the YouTube comments using the YouTube ids of the respective leaders by using the scripting technology. Then we are going to do preprocessing that is making comments stemming, lemmatization and all. Then making the labels classified with 0's and 1's.Then we have to do vectorization of the comments. Then train the comments according the different classification algorithms. Then based on the accuracies got from the classification algorithms we have to pick the highest one and compare it the highest accuracy of the second leader. From both we have to know which one has higher hatred ness using the visualizations.

4. Proposed System Analysis and Design

4.1 Introduction

We have applied various Machine Learning Algorithms – Logistic regression, Random Forest , Naïve Bayes and Support Vector Machine and Deep Learning Model called LSTM for identifying the highest Hatred Ness percentage.

4.2 Requirement Analysis

4.2.1 Functional Requirements

4.2.1.1 System Perspective – The main perspective of our project is to divide the classes into Hatred Ness and Non Hatred Ness. Based on the visualization we will predict the person who have more Hatred Ness and we will predict him as not suitable for developing the society.

4.2.1.2 System Features

- Initially we will done the all pre processing like Tokenization, Stemming and Stop word removal.
- In Project we will dive the comments into Five Categories – Dislike, Insult, Neutral, Support and Meaning Less.
- From that Five categories we divide into Hatred and Non-Hatred. We will know the how much hatred and how much non hatred for a particular member by visualizing the Data.

4.2.1.3 User characteristics

- The users of the system should have basic knowledge of computer operations and text processing.
- They should be able capture the classes properly and understand the results provided by the system.
- The users should also be able to interpret the classification results accurately.

4.2.1.4 Assumptions and Dependencies

- The system assumes that the input datasets are good accurate data and contain relevant metadata.
- The system depends on the availability of relevant metadata and assumes that the user has provided accurate metadata.

4.2.1.5 Domain Requirements

- The system should be able to handle large datasets of various categories and process them accurately.
- It should be able to classify categories into different class based on relevant metadata.
- The system should also be able to handle different text format like number or word and work on various operating systems

4.2.1.6 User Requirements

- The system should provide accurate and reliable results to the users.
- It should be able to classify comments into different classes accurately and efficiently.
- The system should be fast and provide results in a reasonable amount of time.

4.2.2 Non Functional Requirements

4.2.2.1 Product Requirements

4.2.2.1.1 Efficiency

Predicting the "hatred ness" of election candidates is not a clearly defined task, so it's difficult to determine exactly what efficiency means in this context. However, assuming you're referring to the efficiency of a model in predicting the likelihood of a candidate winning an election based on various factors (such as their policies, past voting records, public opinion polls, etc.), then the efficiency can be evaluated in terms of time and space.

In terms of time efficiency, the model should be able to make accurate predictions in a reasonable amount of time. This means that the model should be optimized for fast prediction, with a low inference time. This can be achieved by using efficient algorithms and models, and by optimizing the hardware used for inference (e.g. using GPUs instead of CPUs).

In terms of space efficiency, the model should be able to operate within the available memory constraints. This means that the model should have a small memory footprint, and use minimal resources during inference. This can be achieved by using compact models, and by reducing the number of parameters and layers in the model. Overall, the efficiency of a model in predicting the "hatred ness" of election candidates depends on the specific requirements and constraints of the task, as well as the available resources and data.

4.2.2.1.2 Reliability

Reliability refers to the consistency and accuracy of a measurement or test. In the context of predicting the readiness of election candidates, reliability would mean that the measurement or test used to assess candidate readiness produces consistent and accurate results over time.

Reliability in prediction of readiness in election candidates would mean that the methods used to predict a candidate's readiness are consistent and accurate in their ability to identify candidates who are actually ready for office. This requires using valid and reliable measures of readiness, such as assessing a candidate's experience, qualifications, policy positions, and track record.

It is also important to evaluate the reliability of the measures used to predict readiness over time, to ensure that they continue to produce accurate results as conditions and circumstances change. This may require updating or modifying the measures used to reflect new developments or changes in the political landscape.

4.2.2.1.3 Portability

Portability refers to the transferability of something from one context to another. In the context of politics and elections, portability can refer to the transfer of support or votes from one candidate or party to another.

Overall, portability can be an important factor in predicting electability and hatred ness in election candidates, as it reflects the ability of a candidate to appeal to a broad range of voters and potentially mitigate the negative effects of polarization.

4.2.2.1.4 Usability

By using this application we can easily choose the which candidate best for the particular seat and best for the people and who is the best to serve the public . it gives clear opinions about the election candidates ,so that people can vote good one. here are some potential benefits .

Helps voters make informed decisions: By using a predictive model to assess candidate readiness, voters can gain more insights into the candidates' strengths and weaknesses, and make more informed decisions at the ballot box.
Provides a consistent and objective assessment: By using a standardized set of criteria and data to evaluate candidates, a predictive model can provide a more consistent and objective assessment of their readiness, which can help reduce biases and favoritism.

4.2.2.2 Organizational Requirements

4.2.2.2.1 Implementation Requirements (in terms of deployment)

Colleting comments and classification : we gathered comments from YouTube that election candidates speeches and meetings. and then we did classification (5 types of classification) on the comments and based on their comment type whether it belongs to supporting, liking, neutral, disliking and finally meaningless likewise we classify the comments and sent to next process.

4.2.2.2.2 Engineering Standard Requirements

Data Quality: The data used to train it must be of high quality. The data quality standard should specify the criteria for data selection, pre-processing, cleaning, and validation.

Model Selection: the criteria for selecting the appropriate machine learning model provide guidance on selecting the model based on accuracy, scalability, interpretability, and other factors.

Validation: the validation process for the model provide guidance on performing cross- validation, testing, and evaluation to ensure the model's accuracy.

Training: The training process for the\model. The standard should provide guidance on selecting the training algorithm, the number of iterations, the learning rate, and other factors.

Model Interpretability: the interpretability requirements for the model. provide guidance on methods to explain the model's predictions and ensure transparency.

4.2.2.2.3 Operational Requirements (Explain the applicability for your work w.r.to the following operational requirement(s))

Economic

There are the costs associated with obtaining the necessary data, which may involve purchasing data sets from third-party providers or conducting surveys and interviews to collect primary data.

There are the costs associated with marketing and promoting the predictive model to potential clients, such as political parties, campaign managers, and other stakeholders in the political process.

Environment

Political climate: The current political climate of the country or region where the election is taking place can have a significant impact on the heatreedeness of candidates. If the public is dissatisfied with the current government or political establishment, there may be a greater appetite for change and new candidates may be more likely to be viewed as viable options.

Social

The prediction model may have social implications as it can influence public opinion and perception of candidates. It is important to ensure that the model is fair and unbiased to avoid any negative impact on social dynamics.

Political

The prediction of hatred ness in the election candidates project is inherently political as it involves predicting the success of candidates in an election. It is important to ensure that the model is not biased towards any particular political party or ideology.

Ethical

The prediction model must adhere to ethical standards such as fairness, transparency, and privacy. The model should not be designed to discriminate against any group of people based on their race, gender, religion, or any other characteristic.

Health and Safety

The prediction of hatred ness in the election candidates project does not have any direct health and safety implications.

Sustainability

The project does not have any direct sustainability implications.

Legality

The prediction model must comply with all legal requirements such as data privacy laws, anti-discrimination laws, and election laws

Inspectability

The prediction model should be transparent and explainable so that it can be inspected by relevant authorities to ensure that it is fair and unbiased. The data used in the model should also be accessible for auditing purposes.

4.2.3 SYSTEM REQUIREMENTS

4.2.3.1 H/W Requirements:

- The system should have a powerful processor, sufficient RAM, and storage to handle large amounts of data.
- Operating System Used – Windows

4.2.3.2 S/W Requirements:

- The system should be compatible with the libraries of the algorithms used for the prediction of hatred ness in election candidates
- It should be compatible for machine learning and deep learning algorithms.
- The system should be compatible with different operating systems and programming languages.

5. Results and Discussion

Code :

```
import pandas as panda

from nltk.tokenize import word_tokenize

from nltk.corpus import stopwords

from nltk.stem.porter import *
```

```
import string
import nltk
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix
import seaborn
from textstat.textstat import *
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score
from sklearn.feature_selection import SelectFromModel
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
import numpy as np
from nltk.sentiment.vader import SentimentIntensityAnalyzer as VS
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
%matplotlib inline
dataset = panda.read_csv('F:/Mini Project/Lizz_Truss_Fin.csv')
df1=dataset.head(699)
df1
df2 = dataset.head(699).fillna(0)
df2
```

```
df2['CommentLength'] = df2['Comment'].apply(len)
print(df2.head())
import seaborn as sns
import matplotlib.pyplot as plt
graph = sns.FacetGrid(data=df2, col='Class')
graph.map(plt.hist, 'CommentLength', bins=50)
sns.boxplot(x='Class', y='CommentLength', data=df2)
comment=df2.Comment
## 1. Removal of punctuation and capitlization
## 2. Tokenizing
## 3. Removal of stopwords
## 4. Stemming
stopwords = nltk.corpus.stopwords.words("english")
#extending the stopwords to include other words used in YouTube such as reply etc.
other_exclusions = ["#ff", "ff", "rt"]
stopwords.extend(other_exclusions)
stemmer = PorterStemmer()
def preprocess(comment):
    # removal of extra spaces
    regex_pat = re.compile(r'\s+')
    comment_space = comment.str.replace(regex_pat, ' ')
    # removal of @name[mention]
    regex_pat = re.compile(r'@[\w\-]+')
    comment_name = comment_space.str.replace(regex_pat, '')
    # removal of links[https://abc.com]
```

```
    giant_url_regex =  re.compile('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|'
          '[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
    comments = comment_name.str.replace(giant_url_regex, '')
    # removal of punctuations and numbers
    punc_remove = comments.str.replace("[^a-zA-Z]", " ")
    # remove whitespace with a single space
    newcomment=punc_remove.str.replace(r'\s+', ' ')
    # remove leading and trailing whitespace
    newcomment=newcomment.str.replace(r'^\s+|\s+?$','')
    # replace normal numbers with numbr
    newcomment=newcomment.str.replace(r'\d+(\.\d+)?','numbr')
    # removal of capitalization
    comment_lower = newcomment.str.lower()
    # tokenizing
    tokenized_comment = comment_lower.apply(lambda x: x.split())
    # removal of stopwords
    tokenized_comment=  tokenized_comment.apply(lambda x: [item for item in x if item not
in stopwords])
    # stemming of the commens
    tokenized_comment = tokenized_comment.apply(lambda x: [stemmer.stem(i) for i in x])
    for i in range(len(tokenized_comment)):
        tokenized_comment[i] = ' '.join(tokenized_comment[i])
        comments_p= tokenized_comment
    return comments_p


processed_comments = preprocess(comment)
```

```
df2['processed_comments'] = processed_comments

print(df2[["Comment","processed_comments"]].head(10))

X1 = tfidf

y1= df2['Class'].astype(int)

X_train_tfidf1, X_test_tfidf1, y_train1, y_test1 = train_test_split(X1, y1, random_state=42,
test_size=0.3)

model1 = LogisticRegression().fit(X_train_tfidf1,y_train1)


y_preds1 = model1.predict(X_test_tfidf1)

report1 = classification_report( y_test1, y_preds1 )

print(report1)

acc=accuracy_score(y_test1,y_preds1)

print("Logistic Regression, Accuracy Score:" , acc)

X_train_tfidf2, X_test_tfidf2, y_train2, y_test2 = train_test_split(X1, y1, test_size=0.3)

rf=RandomForestClassifier()

rf.fit(X_train_tfidf2,y_train2)

y_preds2 = rf.predict(X_test_tfidf2)

acc1=accuracy_score(y_test2,y_preds2)

report2 = classification_report( y_test2, y_preds2 )

print(report2)

print("Random Forest, Accuracy Score:",acc1)

X_train_tfidf3, X_test_tfidf3, y_train3, y_test3 = train_test_split(X1.toarray(), y1, random_state=42,
test_size=0.3)

nb=GaussianNB()

nb.fit(X_train_tfidf3,y_train3)

y_preds3 = nb.predict(X_test_tfidf3)

acc2=accuracy_score(y_test3,y_preds3)

report3 = classification_report( y_test3, y_preds3 )
```

```
print(report3)

print("Naive Bayes, Accuracy Score:",acc2)

support =LinearSVC(random_state=20)

support.fit(X_train_tfidf1,y_train1)

y_preds4 = support.predict(X_test_tfidf1)

acc3=accuracy_score(y_test1,y_preds4)

report4 = classification_report( y_test1, y_preds4 )

print(report4)

print("SVM, Accuracy Score:" , acc3)

from keras.models import Sequential
from keras.layers import Dense, LSTM, Embedding
from keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split
X = df2['Comment'].values
y = df2['Class'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(X_train)
X_train = tokenizer.texts_to_sequences(X_train)
X_test = tokenizer.texts_to_sequences(X_test)
vocab_size = len(tokenizer.word_index) + 1
max_len = 100
X_train = pad_sequences(X_train, padding='post', maxlen=max_len)
X_test = pad_sequences(X_test, padding='post', maxlen=max_len)

model = Sequential()
model.add(Embedding(vocab_size, 32, input_length=max_len))
model.add(LSTM(64, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

model.fit(X_train, y_train, epochs=10, batch_size=32)
```

```
score = model.evaluate(X_test, y_test, verbose=0)
print(f' Test accuracy: {score[1]}')
acc4 = score[1]
```

Results :

```
              precision    recall  f1-score   support

           0       0.65      0.58      0.61        88
           1       0.72      0.77      0.74       122

    accuracy                           0.69       210
   macro avg       0.68      0.68      0.68       210
weighted avg       0.69      0.69      0.69       210

Naive Bayes, Accuracy Score: 0.6904761904761905
```

LSTM :

```
Epoch 1/10
18/18 [==============================] - 5s 96ms/step - loss: 0.6861 - accuracy: 0.5725
Epoch 2/10
18/18 [==============================] - 2s 102ms/step - loss: 0.6774 - accuracy: 0.5850
Epoch 3/10
18/18 [==============================] - 2s 94ms/step - loss: 0.6782 - accuracy: 0.5850
Epoch 4/10
18/18 [==============================] - 2s 88ms/step - loss: 0.6770 - accuracy: 0.5921
Epoch 5/10
18/18 [==============================] - 2s 92ms/step - loss: 0.6764 - accuracy: 0.6011
Epoch 6/10
18/18 [==============================] - 2s 88ms/step - loss: 0.6760 - accuracy: 0.6011
Epoch 7/10
18/18 [==============================] - 2s 97ms/step - loss: 0.6746 - accuracy: 0.6011
Epoch 8/10
18/18 [==============================] - 2s 93ms/step - loss: 0.6739 - accuracy: 0.6011
Epoch 9/10
18/18 [==============================] - 2s 96ms/step - loss: 0.6695 - accuracy: 0.6029
Epoch 10/10
18/18 [==============================] - 2s 107ms/step - loss: 0.6673 - accuracy: 0.6047
 Test accuracy: 0.5785714387893677
```

For Liz Truss Dataset we are getting Naïve Bayes as the Highest Accuracy so based on that we will divide the classes.

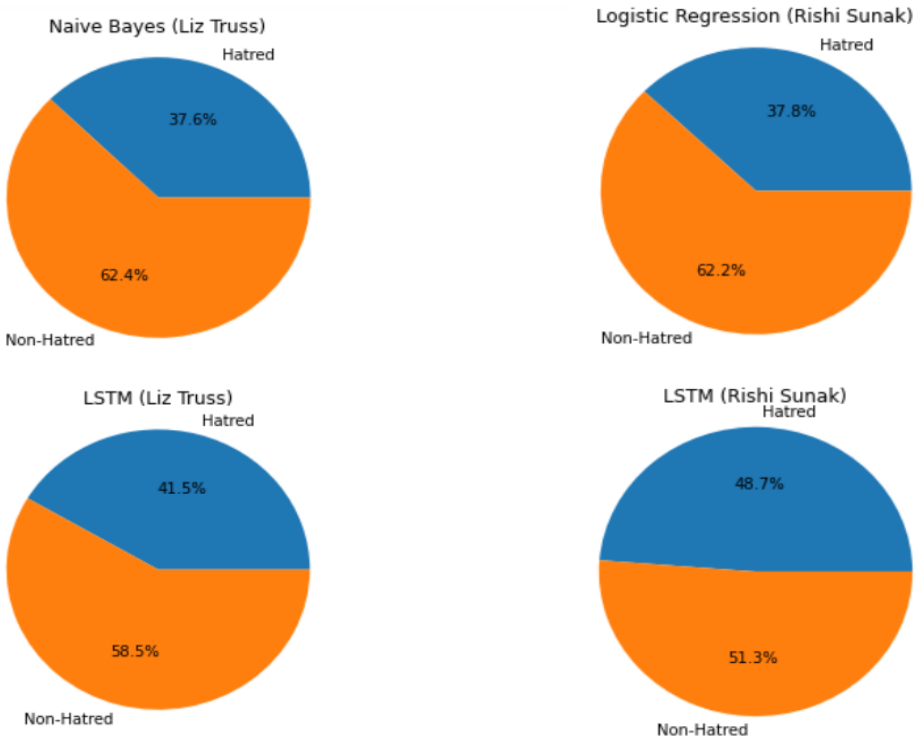
	precision	recall	f1-score	support
0	0.59	0.62	0.61	16
1	0.79	0.76	0.77	29
accuracy			0.71	45
macro avg	0.69	0.69	0.69	45
weighted avg	0.72	0.71	0.71	45

Logistic Regression, Accuracy Score: 0.7111111111111111

LSTM –

```
Epoch 1/10
4/4 [==============================] - 3s 95ms/step - loss: 0.6933 - accuracy: 0.4958
Epoch 2/10
4/4 [==============================] - 0s 89ms/step - loss: 0.6926 - accuracy: 0.5210
Epoch 3/10
4/4 [==============================] - 0s 82ms/step - loss: 0.6932 - accuracy: 0.5210
Epoch 4/10
4/4 [==============================] - 0s 102ms/step - loss: 0.6925 - accuracy: 0.5210
Epoch 5/10
4/4 [==============================] - 0s 87ms/step - loss: 0.6930 - accuracy: 0.5210
Epoch 6/10
4/4 [==============================] - 0s 85ms/step - loss: 0.6921 - accuracy: 0.5210
Epoch 7/10
4/4 [==============================] - 0s 83ms/step - loss: 0.6918 - accuracy: 0.5210
Epoch 8/10
4/4 [==============================] - 0s 87ms/step - loss: 0.6915 - accuracy: 0.5210
Epoch 9/10
4/4 [==============================] - 0s 83ms/step - loss: 0.6917 - accuracy: 0.5210
Epoch 10/10
4/4 [==============================] - 0s 83ms/step - loss: 0.6909 - accuracy: 0.5210
 Test accuracy: 0.6666666865348816
```

For Rishi Sunak Dataset we got the highest accuracy for Logistic Regression so we will classify the hatred and non hatred classes based on this Regression.



From the above pie chart visualization we will say that rishi Sunak has highest percentage of hatredness by the people opinion. So we will say that Lizz Truss can participate in the election.

6. Conclusion and Future work:

In this paper, we have investigated hatred detection on political comments and whether or not supporters of one use more hate speech than supporters of other leader (not significantly). We found that manual annotation is possible with acceptable agreement scores, and that automatic hatredness detection towards political candidates is possible with good performance and the words occur often.

The limitations of this project is that it won't classify or predict the other language comments and also it is not able to classify using hybrid models accurately.

So the future work include that for neural networks it can be better using better hyperparameters and also building the model for multilingual data.

7.References

https://www.mdpi.com/2076-3417/11/18/8575

https://pdfs.semanticscholar.org/0445/07a2f4d0030c05434eceb0230c40f868804d.pdf

https://arxiv.org/pdf/2201.06721.pdf

https://ieeexplore.ieee.org/abstract/document/9455353

https://arxiv.org/abs/1809.08651

https://ieeexplore.ieee.org.egateway.vit.ac.in/stamp/stamp.jsp?tp=&arnumber=9760468

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9455353

https://doi.org/10.31234/osf.io/2rz47

https://doi.org/10.3390/app11188575

https://aclanthology.org/2021.wassa-1.18

https://doi.org/10.47750/pnr.2022.13.S03.051