PROJECT REPORT

on

CLASSIFYING TWEETS BASED ON CLIMATE CHANGE STANCE

Submitted in partial fulfillment of the
requirement for the award of the degree of

BACHELORS OF TECHNOLOGY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr. S. Srinivasan
Professor

Submitted By

Harsha Varshney (18021011381)
Pratyush Shishodiya (18021011532)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING / DEPARTMENT OF COMPUTERAPPLICATION
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **"CLASSIFYING TWEETS BASED ON CLIMATE CHANGE STANCE"** in partial fulfillment of the requirements for the award of the **BACHELORS OF TECHNOLOGY** submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, July 2021 to December 2021, under the supervision of Dr. S.Srinivasan (Professor), Department of Computer Science and Engineering, Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Harsha Varshney, 18021011381

Pratyush Shishodiya, 18021011532

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. S. Srinivasan

Professor

CERTIFICATE

The Final Project Viva-Voce examination of Harsha Varshney (18021011381) and Pratyush Shishodiya (18021011532) has been held on _____ and his/her work is recommended for the award of Bachelors of Technology.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our guide Dr. S. Srinivasan, as well as our esteemed university who gave us the golden opportunity to do this wonderful project on the topic Classifying tweets based on climate change stance which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.

Secondly, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

Harsha Varshney (18021011381)

Pratyush Shishodiya (18021011532)

Abstract

In Today’s world changing climate is becoming a more divisive topic in thepolitical and sociocultural media. Detecting the categories that the world supports in unison is a major requirement in order to understand how people are globally affected by a particular cause or a particular event.

Here, we want to use Twitter to create a classification model that can tell the difference between text that proves belief as well as text that demonstrates disbelief in human-caused changing climates. We've implemented Logistic Regression into three different categories that are positive, negative, and neutral. The report's purpose was to forecast a dataset using machine learning and natural language processing methods. The factors are used to collect more details, analysis univariate and bivariate multivariate analysis, incomplete data treatment and evaluate validation rules, cleansing or readiness information, and a data set of information in visual representation will be given in the evaluation of a set of data offered by various machine learning and natural processing language management techniques. Our general sense in the framework of the research of the forecast of the performance evaluation to determine the best sense of the accuracy of the

calculation is to have a full analysis of the study. We set out to categorize tweets according to their views on climate change. We experimented with Logistic Regression to enhance the outcomes of our classification challenge. We plan to develop our algorithms in a variety of ways in the future.

TABLE OF CONTENTS

CHAPTER – 01

INTRODUCTION

1.1 INTRODUCTION

Despite the fact that climate change is real, it is becoming a divisive issue. Most people, including us, think that terminology used to define the problem is to blame. We believe that this new language is needed for climate change, one which substitutes expressions that inadvertently generate opposition or misunderstanding with words that elicit support and solidarity. To construct this language, we must first be capable of recognizing and categorizing material based on whether it expresses a belief in human-caused changing climates.

1.2 CHALLENGE

The lack of a sufficiently large and good dataset to train a classifier is the most significant obstacle we have in accomplishing such. Our effort addresses this issue by performing classification on tweets using both labeled and unlabeled data, with the goal of employing this approach for further study of climate change language and natural language processing in the building of the new terminology. For our classification problem, we mostly rely on data from Twitter. Twitter is an

appropriate data source since it includes a broad community with various stances on climate change and includes information that is widely available to the researcher.

1.3 METHOD USED

We used Logistic Regression techniques to classify the tweets. We provide this method and examine our findings in this study. In future we will use different machine learning models to examine our findings.

CHAPTER – 02

LITERATURE SURVEY

2.1 SYSTEM REVIEW

The technology and procedures employed in this investigation are detailed in this section. It also goes over the Algorithms and their parameters that have been employed in recent studies. According to the survey It is considered to be one of the most hotly debated themes of the decade. The purpose was to use sentiment analysis on Twitter data to examine how people's attitudes have changed over the last decade. The purpose of this article is to use I gathered tweets containing phrases like "Climate Change" and so on, then used sentiment analysis to identify them as good, negative, or neutral.

2.2 TECHNOLOGY USED

- **PYTHON -** Python is a high-level, interpreted programming language that may be used for a variety of tasks. Python's design concept promotes readability of code, especially shown by its frequent use of spacing. Its language elements and object-oriented approach are designed to assist developers in writing clear, logical code for both major and minor projects. Python is dynamically coded and enables procedural, object-oriented, and

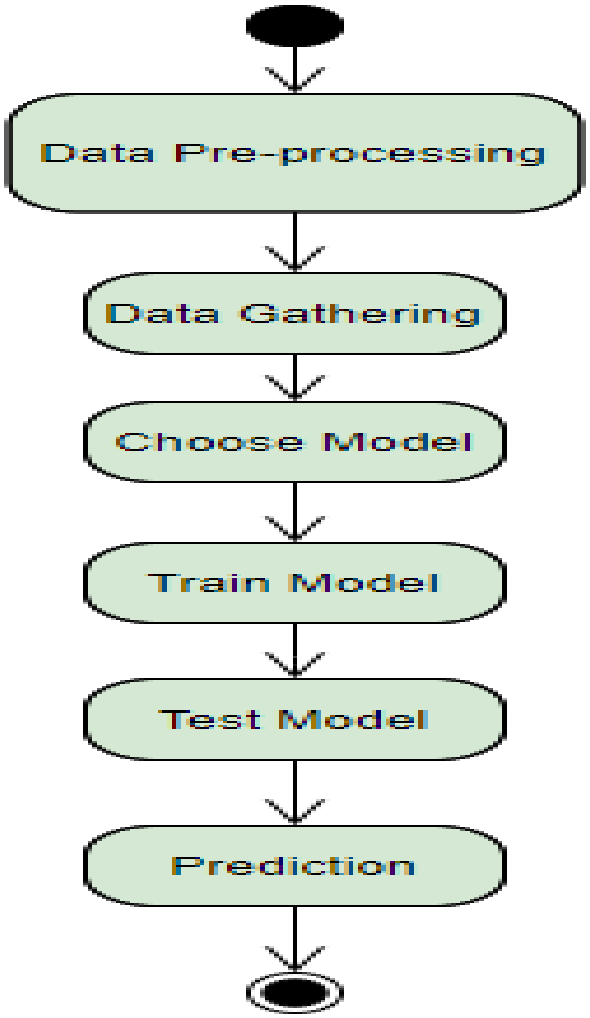
structured programming techniques.

- **MACHINE LEARNING -** Machine learning is the scientific study of algorithms and statistical models that computer systems use to successfully complete the given task without requiring detailed instructions and instead relies on patterns and inferences. Artificial intelligence is regarded as a subset of it. In order to make decisions or judgments without even being explicitly taught, machine learning techniques create a computational formula based on sample data, known as "training data."
- **NATURAL LANGUAGE PROCESSING (NLP) -** NLP is a subject of computer science specifically, a field of artificial intelligence (AI) concerning the abilities of computers to understand text and spoken words inthe same manner that humans can.

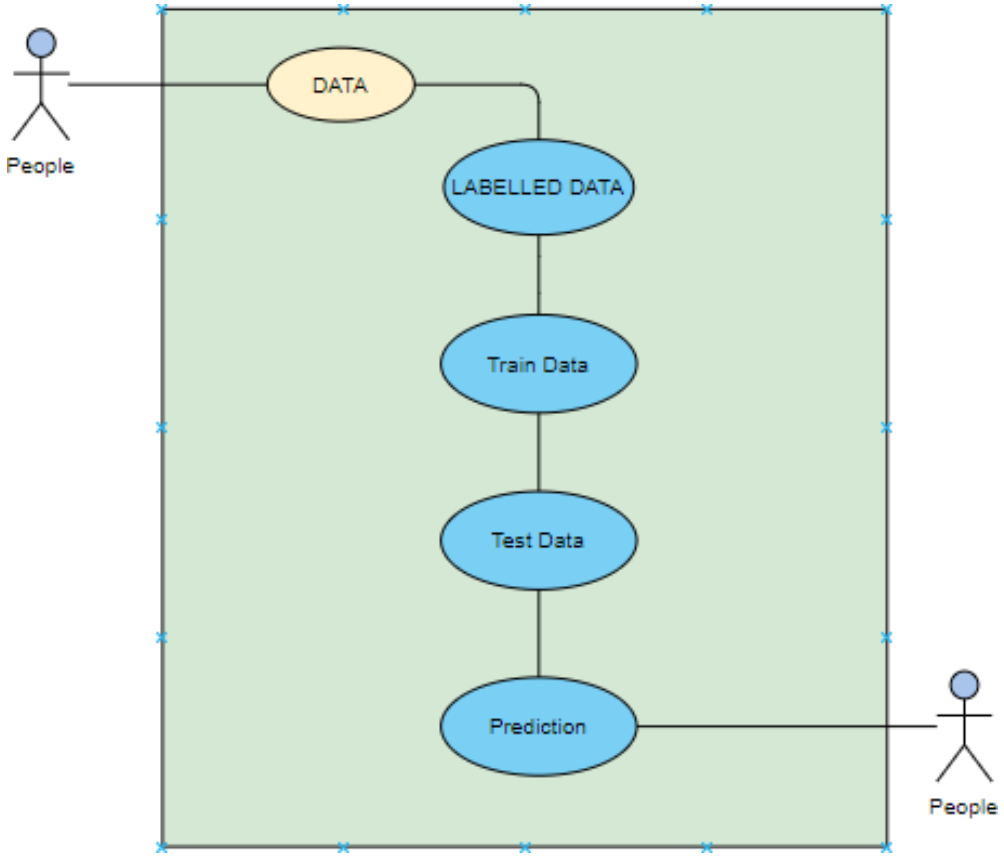
CHAPTER – 03

SYSTEM DESIGN

3.1 ACTIVITY DIAGRAM



3.1 USECASE DIAGRAM



CHAPTER – 04

EXECUTIVE SUMMARY

The dataset "**TWITTER CLIMATE CHANGE SENTIMENT DATASET**" chosen for this project regarding "**CLASSIFYING TWEETS BASED ON CLIMATE CHANGE STANCE**" was downloaded from Kaggle. The Dataset contains texts, comments, replays extracted from the social networking platform Twitter. The dataset contains three variables/features/columns and around 50 thousand messages/comments/replays/rows/tweets. The dataset does not contain any missing values. Using Feature Engineering two new variables were created "Word_Count" and "Text Length" containing the number of words per sentence and number of characters per sentence.

CHAPTER – 05

EXPLORATORY DATA ANALYSIS

In this phase, we are importing different modules that will be needed, and then we load the data and transform it into a structure that we will then use for our model.

IMPORTING THE REQUIRED LIBRARIES

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.stem import PorterStemmer
import regex as re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

STATISTICAL VALUES

```
Data = pd.read_csv("twitter_sentiment_data.csv")
Data['word_counts'] = Data['message'].str.split().str.len()
Data["Text Length"] = Data["message"].str.len()
Data.groupby('sentiment')['word_counts'].mean()

# Exploratory analysis
Data.describe()
```

	sentiment	tweetid	word_counts	Text Length
count	43943.000000	4.394300e+04	43943.000000	43943.000000
mean	0.853924	8.367966e+17	17.400792	122.823954
std	0.853543	8.568506e+16	4.621521	24.720780
min	-1.000000	5.926334e+17	1.000000	7.000000
25%	0.000000	7.970376e+17	14.000000	111.000000
50%	1.000000	8.402301e+17	18.000000	133.000000
75%	1.000000	9.020003e+17	21.000000	140.000000
max	2.000000	9.667024e+17	97.000000	623.000000

PRINTING THE COLUMNS

```
print(Data.columns)

Index(['sentiment', 'message', 'tweetid', 'word_counts', 'Text
Length'], dtype='object')
```

We have observed that there is only one feature that contains non-numerical data.

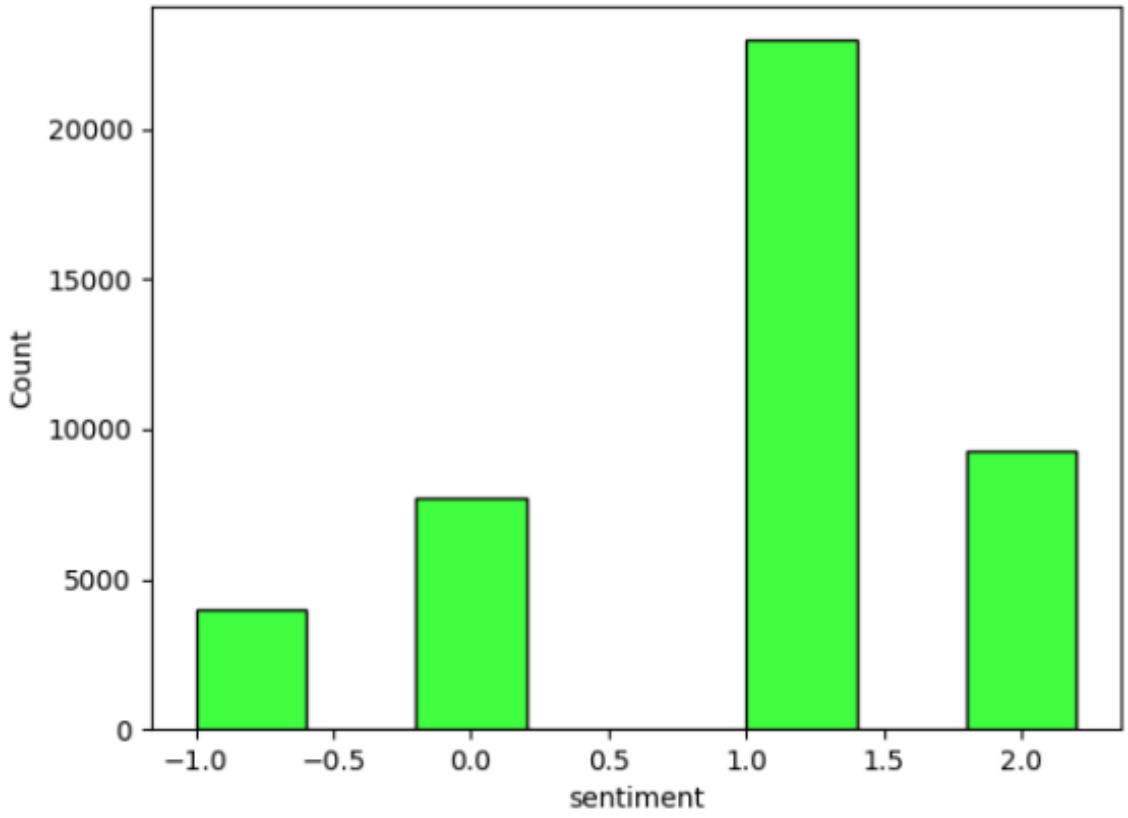
```
print(Data["message"])

0        @tiniebeany climate change is an interesting h...
1        RT @NatGeoChannel: Watch #BeforeTheFlood right...
2        Fabulous! Leonardo #DiCaprio's film on #climat...
3        RT @Mick_Fanning: Just watched this amazing do...
4        RT @cnalive: Pranita Biswasi, a Lutheran from ...
                              ...
43938    Dear @realDonaldTrump,\nYeah right. Human Medi...
```

CHAPTER – 06

DATA VISUALIZATION

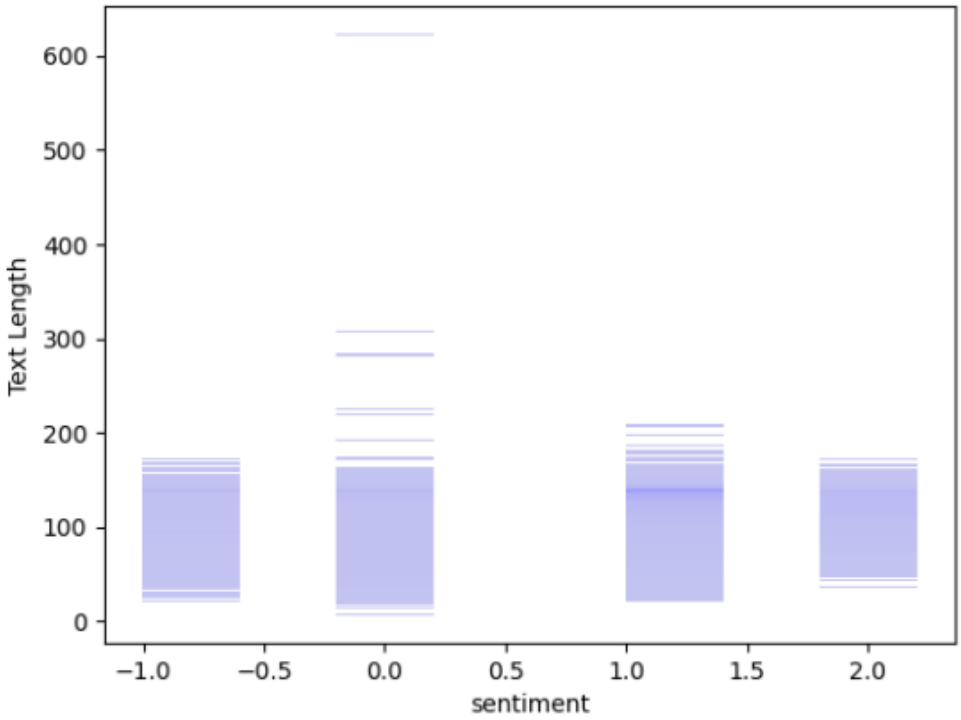
Here we have gained information related by reading the data visually.

Plotting seaborn histplot (histogram) to check the distribution of data.



The histogram reveals that the majority of tweets are pro-climate change, with only a few anti-climate change tweets. This demonstrates that people are concerned about climate change.

To study the relationship between text length and sentiment, a seaborn histplot (histogram) was created.



The histogram reveals that the text length of different groups does not differ significantly. The most noticeable difference is in the neutral class, where people prefer to write more, with some tweets exceeding 600 characters.

CHAPTER – 07

NATURAL PROCESSING LANGUAGE

Processing the main feature of the dataset (message). The message feature is a character variable, so we'll have to clean it up first because it contains a lot of unneeded characters that don't add anything to the sentence's content.

```
print(Data["message"])
0         @tiniebeany climate change is an interesting h...
1         RT @NatGeoChannel: Watch #BeforeTheFlood right...
2         Fabulous! Leonardo #DiCaprio's film on #climat...
3         RT @Mick_Fanning: Just watched this amazing do...
4         RT @cnalive: Pranita Biswasi, a Lutheran from ...
                                ...
43938     Dear @realDonaldTrump,\nYeah right. Human Medi...
43939     What will your respective parties do to preven...
43940     RT @MikkiL: UN Poll Shows Climate Change Is th...
43941     RT @taehbeingextra: i still can$q$t believe th...
43942     @Likeabat77 @zachhaller \n\nThe wealthy + foss...
```

We are using regular expressions to clean the variable message. In the function msg_cleaning, we get rid of different types of content that we don’t need, such as mentions, hashtags, retweets, hyperlinks, numbers, and Chinese words. We have also changed the messages into lowercase for better functioning.

```
def msg_cleaning(msg):
    # Removing @abc12
    msg = re.sub(r'@[A-Za-z0-9]+', '', msg)
    # Removing Hashtags
    msg = re.sub(r'#', '', msg)
    # Removing Chines
    msg = re.sub(r'[^\x00-\x7F]+', '', msg)
    # Removing Retweets
    msg = re.sub(r'RT[\s]+', '', msg)
    # Removing HyperLinks
    msg = re.sub(r'https?:\/\/\s+', '', msg)
    # Removing numeric values
    msg = re.sub(r'\d+', '', msg)
    msg = re.sub(r'aa[A-Za-z0-9]+', '', msg)
    msg = re.sub(r'zz[A-Za-z0-9]+', '', msg)
    return msg
```

```
Data["message"] = Data["message"].str.lower()

0         climate change is an interesting hustle as it...
1        : watch beforetheflood right here, as  travels...
2        fabulous! leonardo dicaprio's film on climate ...
3        _fanning: just watched this amazing documentar...
4        : pranita biswasi, a lutheran from odisha, giv...
                              ...
43938    dear ,\nyeah right. human mediated climate cha...
43939    what will your respective parties do to preven...
43940    : un poll shows climate change is the lowest o...
43941    : i still can$q$t believe this gif of taehyung...
43942      \n\nthe wealthy + fossil fuel industry know ...
Name: message, Length: 43943, dtype: object
```

We have also taken other steps such as tokenization, stemming the words, and stopping words, so that the message feature can help in making the functionality better and understandable.

TOKENIZATION

Tokenization is the process of breaking down a phrase, sentence, paragraph, or even an entire text document into smaller components like individual terms and phrases. Tokens are the names given to each of these smaller units.

```
def identify_tokens(row):
    ide_words = row["message"]
    tokens = word_tokenize(ide_words)

    token_words = [w for w in tokens if w.isalpha()]
    return token_words


Data["message"] = Data.apply(identify_tokens, axis=1)
print(Data['message'])
0         [climate, change, is, an, interesting, hustle,...
1         [watch, beforetheflood, right, here, as, trave...
2         [fabulous, leonardo, dicaprio, film, on, clima...
3         [just, watched, this, amazing, documentary, by...
4         [pranita, biswasi, a, lutheran, from, odisha, ...
                              ...
43938     [dear, yeah, right, human, mediated, climate, ...
43939     [what, will, your, respective, parties, do, to...
43940     [un, poll, shows, climate, change, is, the, lo...
43941     [i, still, can, q, t, believe, this, gif, of, ...
43942     [the, wealthy, fossil, fuel, industry, know, c...
```

STEMMING THE WORDS

The process of reducing a word to its word stem, which affixes to suffixes and prefixes or to the roots of words known as a lemma, is known as stemming. Natural language understanding (NLU) and natural language processing (NLP) both benefit from stemming (NLP).

```
stemming = PorterStemmer()


def stem_list(row):
    my_list = row["message"]
    stemmed_list = [stemming.stem(word) for word in my_list]
    return (stemmed_list)


Data["message"] = Data.apply(stem_list, axis=1)
print(Data["message"])
0         [climat, chang, is, an, interest, hustl, as, i...
1         [watch, beforetheflood, right, here, as, trave...
2         [fabul, leonardo, dicaprio, film, on, climat, ...
3         [just, watch, thi, amaz, documentari, by, leon...
4         [pranita, biswasi, a, lutheran, from, odisha, ...
                                ...
43938     [dear, yeah, right, human, mediat, climat, cha...
43939     [what, will, your, respect, parti, do, to, pre...
43940     [un, poll, show, climat, chang, is, the, lowes...
43941     [i, still, can, q, t, believ, thi, gif, of, ta...
```

STOP WORDS

Stop words are a group of words that are frequently employed in a language. Stop words in English include "a," "the," "is," "are," and others. Stop words are frequently used in Text Mining and Natural Language Processing (NLP) to exclude terms that are so widely used that they contain little meaningful information.

```
stops = set(stopwords.words("english"))
stops.update(["aa", "aaa", "aaaa", "aaaaa", "aaaaaa", "aaaaaaa",
"aaaaaaaa", "aaaaaaaaa", "aaaaaaaaaaaaaaaaaaaah"])


def remove_stops(row):
    my_list = row["message"]
    meningful_words = [w for w in my_list if not w in stops]
    return(meningful_words)

Data["message"] = Data.apply(remove_stops, axis=1)
print(Data["message"])

0         [climat, chang, interest, hustl, wa, global, w...
1         [watch, beforetheflood, right, travel, world, ...
2         [fabul, leonardo, dicaprio, film, climat, chan...
3         [watch, thi, amaz, documentari, leonardodicapr...
4         [pranita, biswasi, lutheran, odisha, give, tes...
                                ...
43938     [dear, yeah, right, human, mediat, climat, cha...
43939     [respect, parti, prevent, climat, chang, globa...
43940     [un, poll, show, climat, chang, lowest, global...
43941     [still, q, believ, thi, gif, taehyung, save, h...
43942     [wealthi, fossil, fuel, industri, know, climat...
Name: message, Length: 43943, dtype: object
```

CHAPTER – 08

METHOD DESCRIPTION

We use a set of auxiliary functions to build and test our model, which accomplish the following tasks. The test and train split are one of them. We divide our data into two portions, test and train data, using the test and train split function. were Train data is used to train the data so that a model can predict the values.

```
# Splitting the data
X_train, X_test, y_train, y_test = train_test_split(X, Y,
test_size=0.25, random_state=1103)

# Splitting the data into validation
X_test, x_val, y_test, y_val = train_test_split(X_test, y_test,
test_size=0.5, random_state=1103)
```

TF – IDF VECTORIZER

TF-IDF are word frequency scores that attempt to identify more interesting words, such as those that are prevalent in a document but not across documents. The TF- IDF Vectorizer tokenizes documents, learns the vocabulary, and inverses document frequency weightings, as well as allowing you to encode new ones.

N – GRAMS

An n-gram is a consecutive sequence of n elements from a particular sample of textor speech in the areas of machine learning and probabilities.

```
tfidf = TfidfVectorizer(ngram_range=(1, 3), max_features=12000,
use_idf=True)
tfidf.fit_transform(X_train)
tfidf.fit_transform(x_val)
```

Using feature engineering, test the model without the features.The

Logistic Regression algorithm was employed.

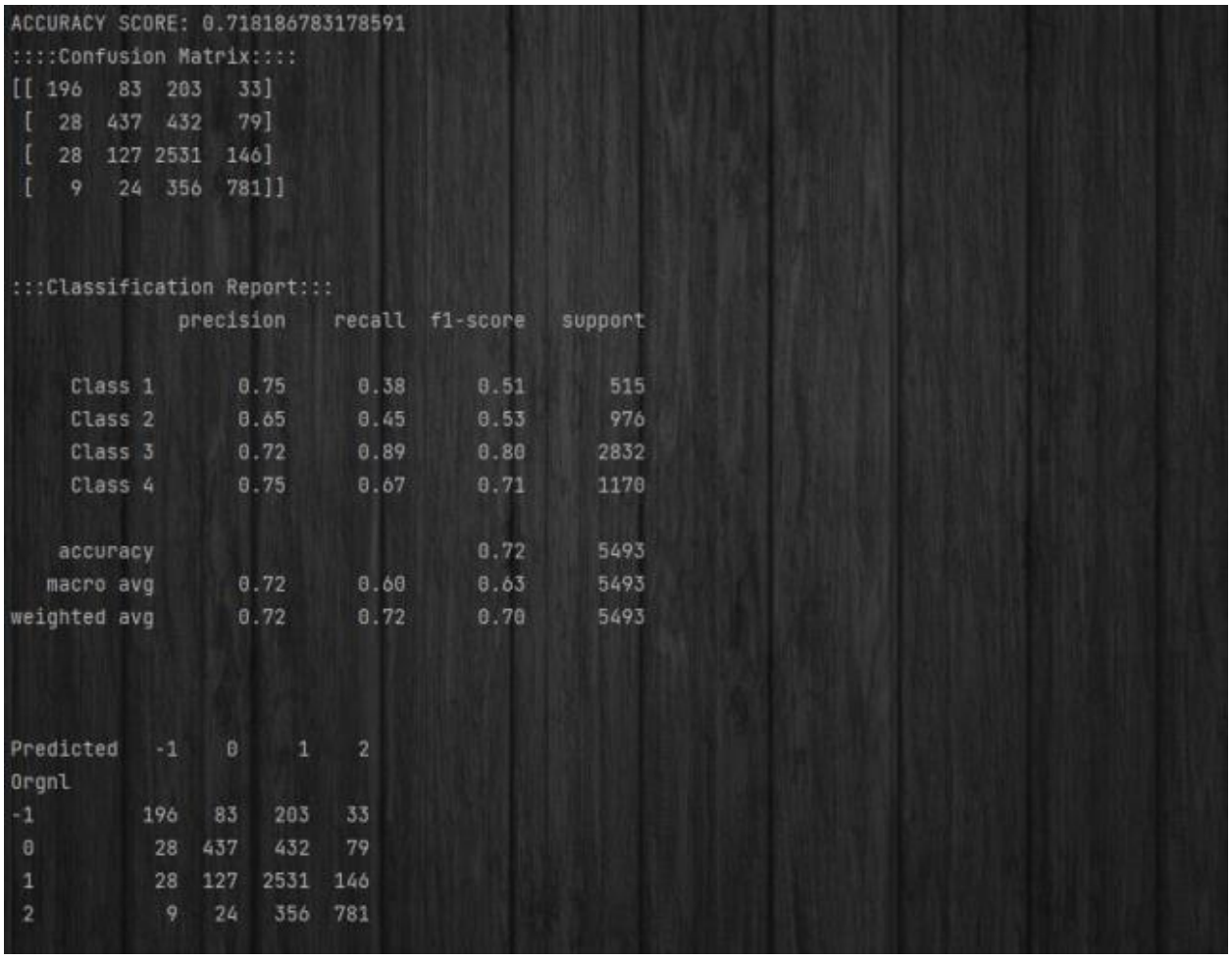
```
grid = {"C": np.logspace(-1, -3, 3, 7, 9), "penalty": ["none",
"l2"]}# l1 lasso l2 ridge
logreg = LogisticRegression(n_jobs=6, max_iter=2000, verbose=True)
#logreg_cv = GridSearchCV(logreg, grid, cv=10, verbose=True)
# X_train["word_count"] = Data["word_counts"]
# X_test["word_count"] = Data["word_counts"]
# X_train["Text Length"] = Data["Text Length"]
# X_test["Text Length"] = Data["Text Length"]
logreg.fit(tfidf.transform(X_train), y_train)

rfc_predict = logreg.predict(tfidf.transform(x_val))
```

```
print("ACCURACY SCORE:", metrics.accuracy_score(y_val, rfc_predict))
print("::::Confusion Matrix::::")
print(confusion_matrix(y_val, rfc_predict))
print("\n")

print(":::Classification Report:::")
print(classification_report(y_val, rfc_predict, target_names=['Class
1', 'Class 2', 'Class 3', 'Class 4']))
print("\n")

print(pd.crosstab(y_val, rfc_predict, rownames=["Orgnl"],
colnames=['Predicted']))
```

CHAPTER – 09

RESULT

We used accuracy on the test set as our assessment metric because our goal was toreliably predict which class a tweet belongs to. We have experimented with Logistic Regression with N-gram and TF-IDF Vectorizer. Using the combination,we have gained an accuracy of 71%.



```
ACCURACY SCORE: 0.7181867831785 91
::::Confusion Matrix::::
[[ 196   83  203   33]
 [  28  437  432   79]
 [  28  127 2531  146]
 [   9   24  356  781]]


:::Classification Report:::
              precision    recall  f1-score   support

     Class 1       0.75      0.38      0.51       515
     Class 2       0.65      0.45      0.53       976
     Class 3       0.72      0.89      0.80      2832
     Class 4       0.75      0.67      0.71      1170

    accuracy                           0.72      5493
   macro avg       0.72      0.60      0.63      5493
weighted avg       0.72      0.72      0.70      5493



Predicted   -1    0     1    2
Orgnl
-1         196   83   203   33
 0          28  437   432   79
 1          28  127  2531  146
 2           9   24   356  781
```

There may be systemic changes in how terminology connected to "climate change"and "global warming" has altered over time because the labelled and unlabeled tweets are over 3 years apart in age. These variations may have resulted in the model performing worse than when it was evaluated on the validation and test sets.Also, there is a lot of

overlap between the most commonly used words for each class. We believe this makes modelling the decision boundary between the groups extremely difficult, which explains why the model is unable to categories tweets with high accuracy.

CHAPTER – 10

CONCLUSION AND FUTURE WORK

We set out to categories tweets according to their views on climate change. However, the strategies we tried yielded worse results, which we assume is due to the fact that the unlabeled data differed systematically from the labelled data.

The decision boundaries between the classes are difficult to model, and there areno labelled training samples. In the future, we'll experiment with other models and aim to improve models in a variety of ways.

CHAPTER – 11

REFERENCES

[Moh+16] Saif Mohammad et al. “SemEval-2016 Task 6: Detecting Stance in Tweets”. In: SemEval@NAACL-HLT. 2016.

[Oga+13] Kohei Ogawa et al. “Infinitesimal Annealing for Training Semi-Supervised Support Vector Machines”. In: Proceedings of the 30th International Conference on Machine Learning. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR,17–19 Jun 2013, pp. 897–905.

URL: http://proceedings.mlr.press/v28/ogawa13a.html.

[Gie+12] Fabian Gieseke et al. “Sparse Quasi-Newton Optimization for Semi-Supervised Support Vector Machines”. In: vol. 1. Jan. 2012, pp. 45–54.

[TT03] Yoshimasa Tsuruoka and Jun’ichi Tsujii. “Training a Naive Bayes Classifier via the EM Algorithm with a Class Distribution Constraint”. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT- NAACL 2003 - Volume 4. CONLL ’03. Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 127–134. DOI: 10.3115/1119176.1119193. URL: https://doi.org/10.3115/1119176.1119193. [Nig+00] Kamal Nigam et al. “Text Classification from Labeled and Unlabeled Documents using EM”. In: Machine Learning 39.2 (May 2000), pp. 103–134. ISSN: 1573-0565. DOI:

10.1023/A:1007692713085.
URL: https://doi.org/10.1023/A:1007692713085.

[Qia19] Edward C Qian. “edwardcqian/climate_change_sentiment”. In: GitHub (Mar. 2019).

URL: https:// github.com/edwardcqian/climate_change_sentiment.

[19] “Tweepy”. In: Tweepy Documentation - tweepy 3.7.0 documentation (2019).

URL: http://docs.tweepy. org/en/latest/.

[PHV17] Neetu Pathak, Michael J Henry, and Svitlana Volkova. “Understanding Social Media’s Take on Climate Change through Large-Scale Analysis of Targeted Opinions and Emotions”. In: 2017 AAAI Spring Symposium Series. 2017. [Maa+15] Lars Maaløe et al. “Improving Semi-Supervised Learning withAuxiliary Deep Generative Models”. In: NIPS 2015. 2015.