1. INTRODUCTION

With the rapid development of social networks and microblogging websites. Microblogging websites have become one of the largest web destinations for people to express their thoughts, opinions, and attitudes about different topics Twitter is a widely used microblogging platform and social networking service that generates a vast amount of information. In recent years, researchers preferably made the use of social data for the sentiment analysis of people's opinions on a product, topic, or event. Sentiment analysis, also known as opinion mining, is an important natural language processing task. This process determines the sentiment orientation of a text as positive, negative, or neutral Twitter sentiment analysis is currently a popular topic for research. Such analysis is useful because it gathers and classifies public opinion by analyzing big social data. However, Twitter data have certain characteristics that cause difficulty in conducting sentiment analysis in contrast to analyzing other types of data. Tweets are restricted to 140 characters, written in informal English, contain irregular expressions, and contain several abbreviations and slang words. To address these problems, researchers have conducted studies focusing on sentiment analysis of tweets, Twitter sentiment analysis approaches can be generally categorized into two main approaches, the machine learning approach, and a lexicon-based approach. In this study, we use machine learning techniques to tackle twitter sentiment analysis.

1.1 PROJECT SCOPE

Sentiment analysis is the task of finding the opinions and affinity of people towards specific topics of interest. Be it a product or a movie, opinions of people matter, and it affects the decision-making process of people. The first thing a person does when he or she wants to buy a product on-line, is to see the kind of reviews and opinions that people have written. Social media such as Facebook, blogs, twitter have become a place where people post their opinions on certain topics. The sentiment of the tweets of a particular subject has multiple usage, including stock market analysis of a company, movie reviews, in psychology to analyze the mood of people that has a variety of applications.

1.2 PROJECT PURPOSE

The idea of this project came into existence because of the increase in theaccuracy in tweet analysis and more the objectives of this project are:

1. To Make the Sentiment analysis of the tweets.
2. To help Businesses about product opinions.
3. Provide More categories In the Analysis.

1.3 PROJECT FEATURE

The main feature of this project is to automate our analysis based on the reviews or comments in the social media by the people, for a sentimental analysis. Sentimental Analysis (SA) is introduced to the world to tell us the information is correct or wrong in each scenario using the social media tags. Thus, we can know about how world or people are reacting to every aspect currently going in the world.

2.LITERATURE SURVEY

From tweets to polls: Linking text sentiment to public opinion time series
AUTHORS: B. O'Connor, R. Balasubramanian, B. R. Routledge, and N. A. Smith,

We connect measures of public opinion measured from polls with sentiment measured from text. We analyze several surveys on consumer confidence and political opinion over the 2008 to 2009 period, and find they correlate to sentiment word frequencies in contemporaneous Twitter messages. While our results vary across datasets, in several cases the correlations are as high as 80%, and capture important large-scale trends. The results highlight the potential of text streams as a substitute and supplement for traditional polling.

Using appraisal groups for sentiment analysis
AUTHORS: C. Whitelaw, N. Garg, and S. Argamon

Little work to date in sentiment analysis (classifying texts by `positive' or `negative' orientation) has attempted to use fine-grained semantic distinctions in features used for classification. We present a new method for sentiment classification based on extracting and analyzing appraisal groups such as "very good" or "not terribly funny". An appraisal group is represented as a set of attribute values in several taskindependent semantic taxonomies, based on Appraisal Theory. Semi-automated methods were used to build a lexicon of appraising adjectives and their modifiers. We classify movie reviews using features based upon these taxonomies combined with standard "bag-of-words" features, and report state-of-the-art accuracy of 90.2%. In addition, we find that some types of appraisals appear to be more significant for sentiment classification.

Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold AUTHORS: H. Saif, M. FernÆndez, Y. He, and H. Alani

Sentiment analysis over Twitter offers organisations and individuals a fast and effective way to monitor the publics' feelings towards them and their competitors. To assess the performance of sentiment analysis methods over Twitter a small set of evaluation datasets have been released in the last few years. In this paper we present an overview of eight publicly available and manually annotated evaluation datasets for Twitter sentiment analysis. Based on this review, we show that a common limitation of most of these datasets, when assessing sentiment analysis at target (entity) level, is the lack of distinctive sentiment annotations among the tweets and the entities contained in them. For example, the tweet "I love iPhone, but I hate iPad" can be annotated with a mixed sentiment label, but the entity iPhone within this tweet should be anno-tated with a positive sentiment label. Aiming to overcome this limitation, and to complement current evaluation datasets, we present STS-Gold, a new evaluation dataset where tweets and targets (entities) are annotated individually and therefore may present different sentiment labels. This paper also provides a comparative study of the various datasets along several dimensions including: total number of tweets, vocabulary size and sparsity. We also investigate the pair-wise correlation among these dimensions as well as their correlations to the sentiment classification performance on different datasets.

Optimizing N-gram based text feature selection in sentiment analysis for commercial products in Twitter through polarity lexicons
AUTHORS: M. A. Cabalist and K. J. Espinosa

This study aims to optimize N-gram based text feature selection in sentiment analysis for commercial products in twitter through polarity lexicons. This can be done by merging dictionary-based weighing with naïve-Bayes classification of sentiments. The study is still ongoing but partial results show potential.

Determining the sentiment of opinions
AUTHORS: S.-M. Kim and E. Hovey

Identifying sentiments (the affective parts of opinions) is a challenging problem. We present a system that, given a topic, automatically finds the people who hold opinions about that topic and the sentiment of each opinion. The system contains a module for determining word sentiment and another for combining sentiments within a sentence. We experiment with various models of classifying and combining sentiment at word and sentence levels, with promising result.

3.SYSTEM ANALYSIS

3.1 INTRODUCTION

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

3.2 PROBLEM DEFINITION

A Twitter sentiment analysis identifies negative, positive, or neutral emotions within the text of a tweet. It is a text analysis using natural language processing (NLP) and machine learning. It identifies and extracts subjective information from original data, providing a company with a better understanding of the social sentiment of its brand, product, or service. At the same time, analyze the online conversations of customers. It is frequently used to analyze customer feedback, survey responses, and product reviews.

3.3 EXISTING SYSTEM

Most classification algorithms are focused on predicting nominal class data labels. However, a rule for predicting categories or labels on an ordinal scale involves many pattern recognition issues. This type of problem, known as ordinal classification or ordinal regression. Recently, ordinal regression has received considerable attention. Ordinal regression issues in many fields of research are very common and have often been regarded as standard nominal problems that can lead to non-optimal solutions.

3.3.1 DISADVANTAGES OF EXISTING SYSTEM

1. lead to non-optimal solutions.

3.4 PROPOSED SYSTEM

The current study mainly focuses on the sentiment analysis of Twitter data (tweets) using different machine learning algorithms to deal with ordinal regression problems. In this paper, we propose an approach including pre-processing tweets, feature extraction methods, and constructing a scoring and balancing system, then using different techniques of machine learning to classify tweets under several classes.

3.4.1 ADVANTAGES OF PROPOSED SYSTEM

1. Experimental findings reveal that the proposed approach can detect ordinal regression using machine learning methods with good accuracy.
2. Moreover, results indicate that Decision Trees obtains the best results outperforming all the other algorithms.

3.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

3.5.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.5.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.5.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.6 HARDWARE & SOFTWARE REQUIREMENTS

3.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Dual Core I5 and above
- Hard disk : 8GB and above
- RAM : 8GB and above
- Input devices : Keyboard, mouse

3.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements:

- Operating system : Windows 8 and above
- Front end : HTML, JavaScript, CSS, Bootstrap
- Languages : Python
- Editor : Vs code
- Software : Nodejs (Version 12.3.1)

3.7 ALGORITHMS

3.7.1 DECISION TREE ALGORITHM

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable.

3.7.2 SUPPORT VECTOR MACHINE NEURAL

Support Vector Machine: Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

3.7.3 RANDOM FOREST

A Random Forest Algorithm is a supervised machine learning algorithm which is extremely popular and is used for Classification and Regression problems in Machine Learning. We know that a forest comprises numerous trees, and the more trees more it will be robust.

3.8 SOFTWARE REQUIREMENT SPECIFICATION

3.8.1 FUNCTIONAL REQUIREMENTS

1.Data Collection
2.Data Preprocessing
3.Training and Testing
4.Modiling
5.Predicting

3.8.2 NON FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of non-functional requirement, “how fast does the website load?”

Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Serviceability requirement
- Manageability requirement
- Recover ability requirement
- Security requirement
- Data Integrity requirement

4.ARCHITECTURE

4.1 PROJECT ARCHITECTURE

This project architecture shows the procedure for classifying the sentimental analysis of tweets.

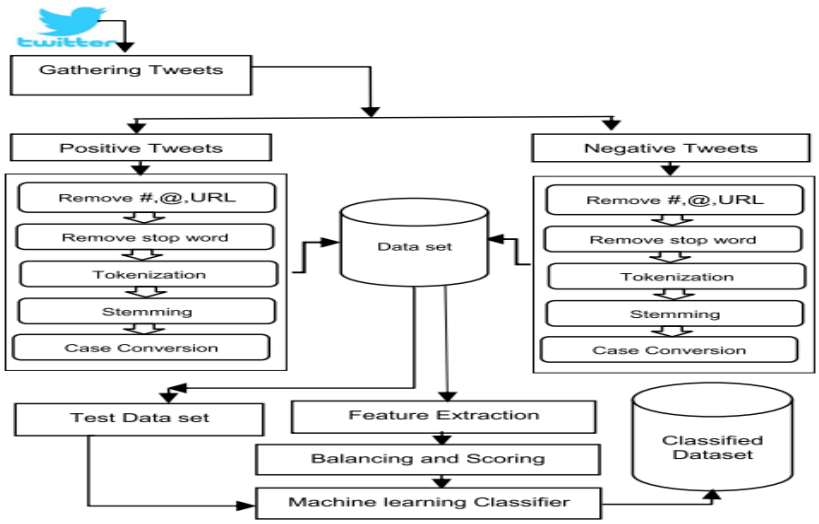


Figure 4.1 : System Architecture .

4.2 PROJECT DESCRIPTION

- Load NLTK Tweets: Using this module we will load twitter sentiment corpora dataset from NLTK library.
- Read NLTK Tweets: Using this module we will read tweets from NLTK and then clean tweets by removing special symbols, stop words and then perform stemming (stemming means removing from words for example ORGANIZATION word will become ORGANIZE after applying stem) on each words. Then we will calculate TFIDF vector.
- Run SVR Algorithm: In this module we will give TFIDF vector as input to train SVR algorithm. This algorithm will take 80% vector for train and 20% vector as test. Then algorithm applied 80% trained model on 20% test data to calculate prediction accuracy.

- Similarly, we will build model for Random Forest and Decision tree to calculate their accuracy.
- Detect Sentiment Type: Using this module we will upload test tweets and then application will apply train model on those test tweets to predict sentiment of that tweet.
- Accuracy Graph: Using this module we will display accuracy graph between all algorithms.

4.3 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

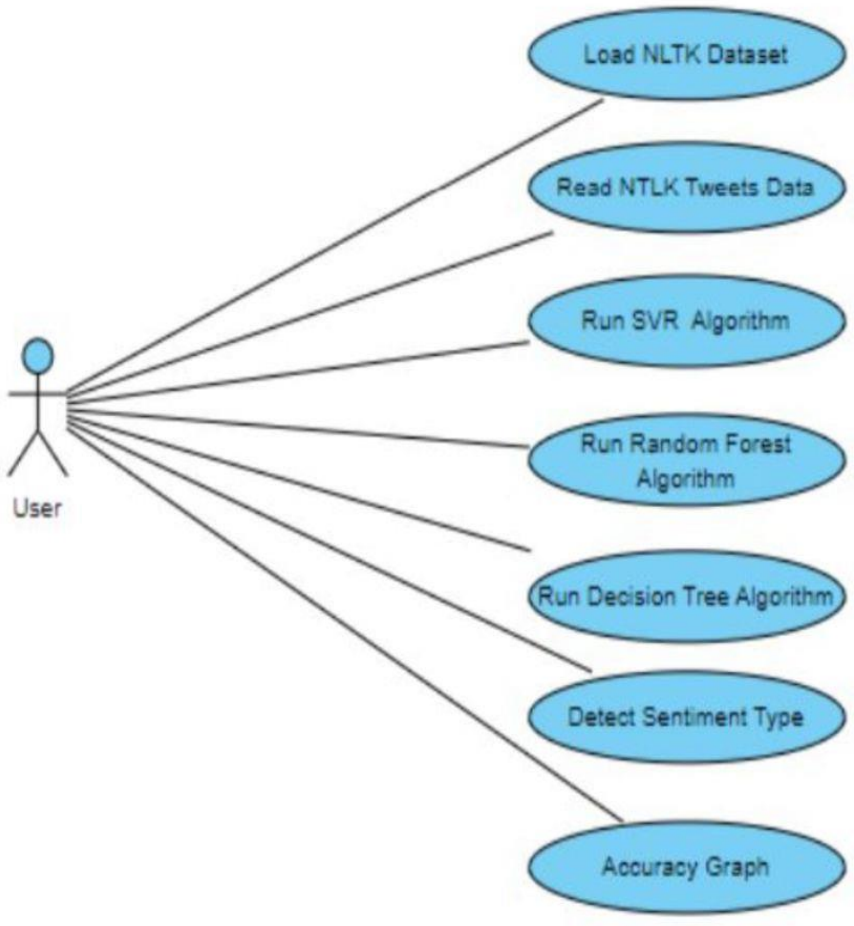


Figure 4.3: Use Case Diagram

4.4 CLASS DIAGRAM

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

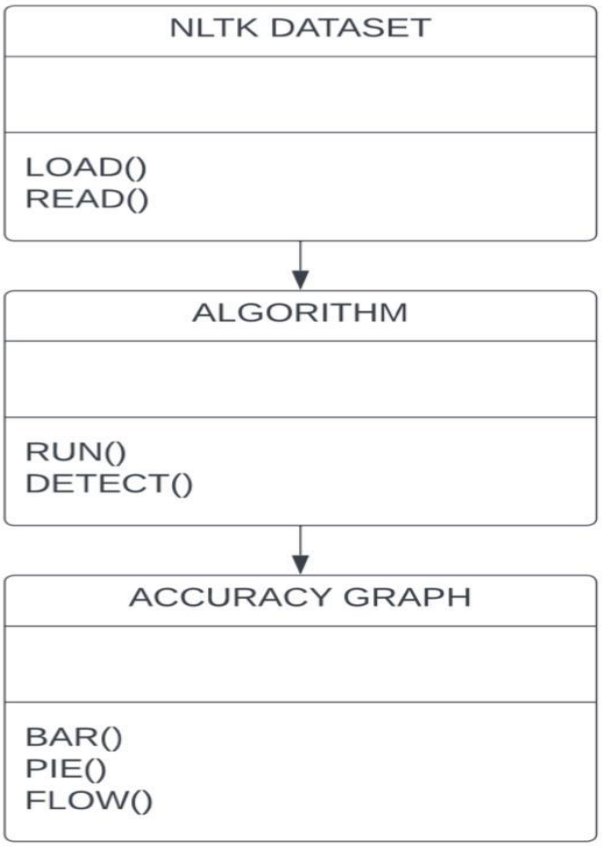


Figure 4.4: Class Diagram

4.5 SEQUENCE DIAGRAM

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

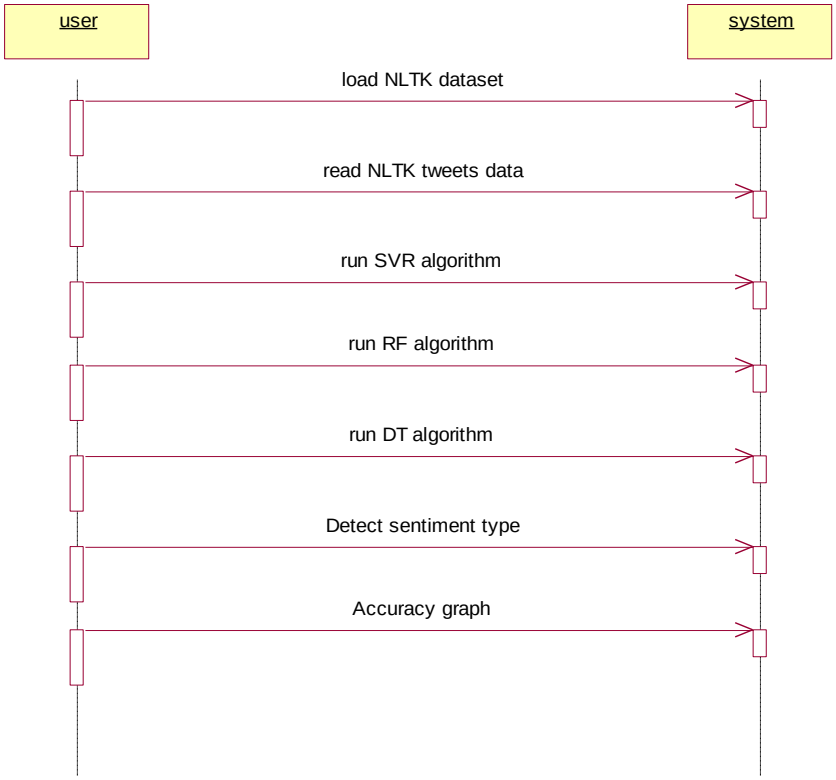


Figure 4.5: Sequence Diagram

4.6 ACTIVITY DIAGRAM

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

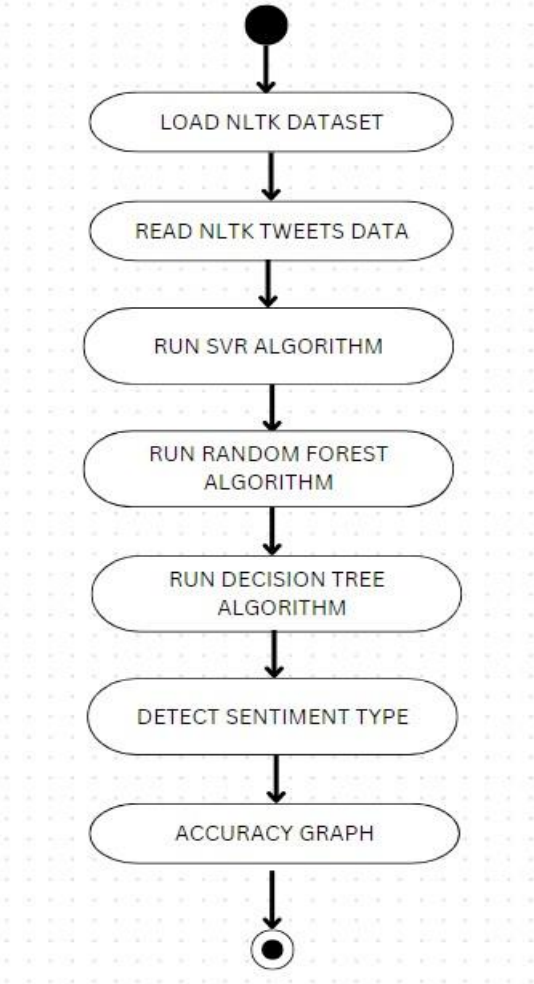


Figure 4.6: Activity Diagram

5. IMPLEMENTATION

5.1 SAMPLE CODE

Class PorterStemmer:

```
def isCons(self, letter):
if letter == 'a' or letter == 'e' or letter == 'i' or letter == 'o' or letter == 'u':
return False
else:
return True
def isConsonant(self, word, i):
letter = word[i]
if self.isCons(letter):
if letter == 'y' and abs(i-1)<=len(word) and self.isCons(word[i-1]):
return False
else:
return True
else:
return False
def isVowel(self, word, i):
return not(self.isConsonant(word, i))
# *S
def endsWith(self, stem, letter):
if stem.endswith(letter):
return True
else:
return False
# *v*
def containsVowel(self, stem):
for i in stem:
if not self.isCons(i):
return True
return False
# *d
def doubleCons(self, stem):
if len(stem) >= 2:
if self.isConsonant(stem, -1) and self.isConsonant(stem, -2):
return True
else:
return False
```

```
else:
return False
def getForm(self, word):

form = []
formStr = ''

for i in range(len(word)):
if self.isConsonant(word, i):
if i != 0:
prev = form[-1]
if prev != 'C':
form.append('C')
else:
form.append('C')
else:
if i != 0:
prev = form[-1]
if prev != 'V':
form.append('V')
else:
form.append('V')
for j in form:
formStr += j
return formStr
def getM(self, word):
form = self.getForm(word)
m = form.count('VC')
return m
# *o
def cvc(self, word):
if len(word) >= 3:
f = -3
s = -2
t = -1
third = word[t]
if self.isConsonant(word, f) and self.isVowel(word, s) and
self.isConsonant(word, t):
if third != 'w' and third != 'x' and third != 'y':
return True
else:
return False
```

```
else:
return False
else:
return False
def replace(self, orig, rem, rep):
result = orig.rfind(rem)
base = orig[:result]
replaced = base + rep
return replaced

def replaceM0(self, orig, rem, rep):

result = orig.rfind(rem)
base = orig[:result]

if self.getM(base) > 0:

replaced = base + rep
return replaced
else:
return orig
def replaceM1(self, orig, rem, rep):
result = orig.rfind(rem)
base = orig[:result]
if self.getM(base) > 1:
replaced = base + rep
return replaced
else:
return orig
def step1a(self, word):
if word.endswith('sses'):
word = self.replace(word, 'sses', 'ss')
elif word.endswith('ies'):
word = self.replace(word, 'ies', 'i')
elif word.endswith('ss'):
word = self.replace(word, 'ss', 'ss')
elif word.endswith('s'):
word = self.replace(word, 's', '')
else:
pass
return word
def step1b(self, word):
flag = False
```

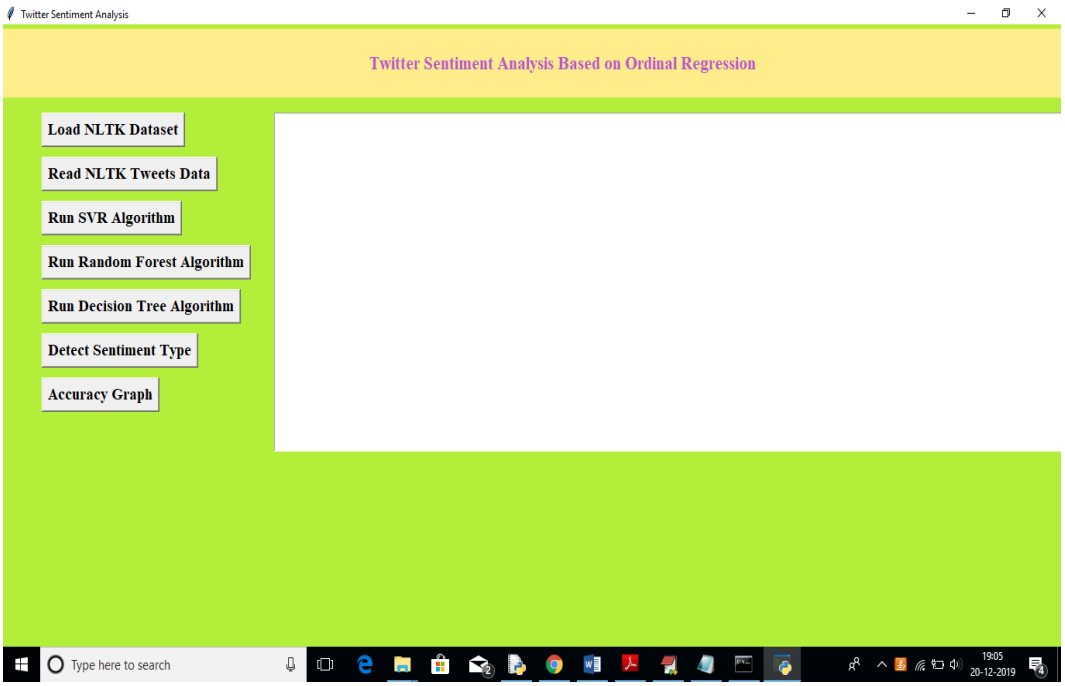
```
if word.endswith('eed'):
result = word.rfind('eed')
base = word[:result]
if self.getM(base) > 0:
word = base
word += 'ee'
elif word.endswith('ed'):
result = word.rfind('ed')
base = word[:result]
if self.containsVowel(base):
word = base
flag = True
elif word.endswith('ing'):
result = word.rfind('ing')

base = word[:result]
if self.containsVowel(base):

word = base
flag = True
if flag:

if word.endswith('at') or word.endswith('bl') or word.endswith('iz'):
word += 'e'
elif self.doubleCons(word) and not self.endsWith(word, 'l') \

and not self.endsWith(word, 's') and not self.endsWith(word, 'z'):
word = word[:-1]
elif self.getM(word) == 1 and self.cvc(word):
word += 'e'
else:
pass
else:
pass
return word
def step1c(self, word):
if word.endswith('y'):
result = word.rfind('y')
base = word[:result]
if self.containsVowel(base):
word = base
word += 'i'
return word
```

```
def step2(self, word):
if word.endswith('ational'):
word = self.replaceM0(word, 'ational', 'ate')
elif word.endswith('tional'):
word = self.replaceM0(word, 'tional', 'tion')
elif word.endswith('enci'):
word = self.replaceM0(word, 'enci', 'ence')
elif word.endswith('anci'):
word = self.replaceM0(word, 'anci', 'ance')
elif word.endswith('izer'):
word = self.replaceM0(word, 'izer', 'ize')
elif word.endswith('abli'):
word = self.replaceM0(word, 'abli', 'able')
elif word.endswith('alli'):
word = self.replaceM0(word, 'alli', 'al')
elif word.endswith('entli'):
word = self.replaceM0(word, 'entli', 'ent')
elif word.endswith('eli'):
word = self.replaceM0(word, 'eli', 'e')

elif word.endswith('ousli'):
word = self.replaceM0(word, 'ousli', 'ous')
elif word.endswith('ization'):

word = self.replaceM0(word, 'ization', 'ize')
elif word.endswith('ation'):
word = self.replaceM0(word, 'ation', 'ate')
elif word.endswith('ator'):

word = self.replaceM0(word, 'ator', 'ate')
elif word.endswith('alism'):
word = self.replaceM0(word, 'alism', 'al')
elif word.endswith('iveness'):
word = self.replaceM0(word, 'iveness', 'ive')

elif word.endswith('fulness'):
word = self.replaceM0(word, 'fulness', 'ful')
elif word.endswith('ousness'):
word = self.replaceM0(word, 'ousness', 'ous')
elif word.endswith('aliti'):
word = self.replaceM0(word, 'aliti', 'al')
```

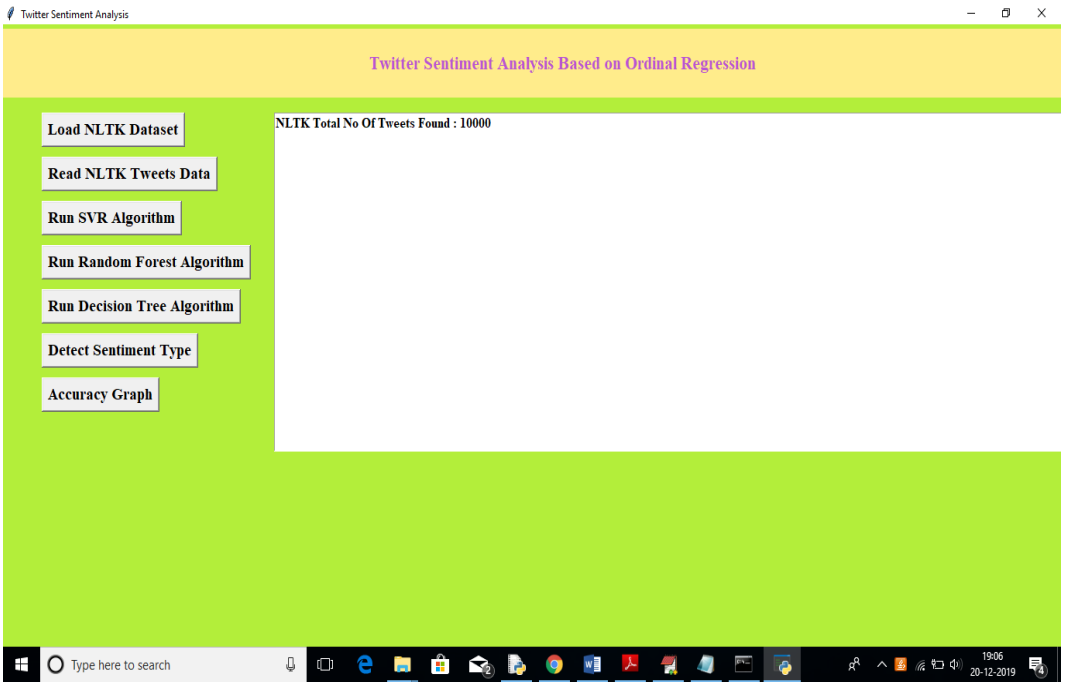
```
elif word.endswith('iviti'):
word = self.replaceM0(word, 'iviti', 'ive')
elif word.endswith('biliti'):
word = self.replaceM0(word, 'biliti', 'ble')
return word
def step3(self, word):
if word.endswith('icate'):
word = self.replaceM0(word, 'icate', 'ic')
elif word.endswith('ative'):
word = self.replaceM0(word, 'ative', '')
elif word.endswith('alize'):
word = self.replaceM0(word, 'alize', 'al')
elif word.endswith('iciti'):
word = self.replaceM0(word, 'iciti', 'ic')
elif word.endswith('ful'):
word = self.replaceM0(word, 'ful', '')
elif word.endswith('ness'):
word = self.replaceM0(word, 'ness', '')
return word
def step4(self, word):
if word.endswith('al'):
word = self.replaceM1(word, 'al', '')
elif word.endswith('ance'):
word = self.replaceM1(word, 'ance', '')
elif word.endswith('ence'):
word = self.replaceM1(word, 'ence', '')

elif word.endswith('er'):
word = self.replaceM1(word, 'er', '')
elif word.endswith('ic'):
word = self.replaceM1(word, 'ic', '')
elif word.endswith('able'):

word = self.replaceM1(word, 'able', '')
elif word.endswith('ible'):
word = self.replaceM1(word, 'ible', '')
elif word.endswith('ant'):

word = self.replaceM1(word, 'ant', '')
elif word.endswith('ement'):
word = self.replaceM1(word, 'ement', '')
elif word.endswith('ment'):
word = self.replaceM1(word, 'ment', '')
elif word.endswith('ent'):
word = self.replaceM1(word, 'ent', '')
elif word.endswith('ou'):
```

```
word = self.replaceM1(word, 'ou', '')
elif word.endswith('ism'):
word = self.replaceM1(word, 'ism', '')
elif word.endswith('ate'):
word = self.replaceM1(word, 'ate', '')
elif word.endswith('iti'):
word = self.replaceM1(word, 'iti', '')
elif word.endswith('ous'):
word = self.replaceM1(word, 'ous', '')
elif word.endswith('ive'):
word = self.replaceM1(word, 'ive', '')
elif word.endswith('ize'):
word = self.replaceM1(word, 'ize', '')
elif word.endswith('ion'):
result = word.rfind('ion')
base = word[:result]
if self.getM(base) > 1 and (self.endsWith(base, 's') or
self.endsWith(base, 't')):
word = base
word = self.replaceM1(word, '', '')
return word
def step5a(self, word):
if word.endswith('e'):
base = word[:-1]
if self.getM(base) > 1:
word = base
elif self.getM(base) == 1 and not self.cvc(base):
word = base
return word
def step5b(self, word):
if self.getM(word) > 1 and self.doubleCons(word) and
self.endsWith(word, 'l'):
word = word[:-1]
return word
def stem(self, word):
word = self.step1a(word)
word = self.step1b(word)
word = self.step1c(word)
word = self.step2(word)
word = self.step3(word)
word = self.step4(word)


word = self.step5a(word)
word = self.step5b(word)
return
```

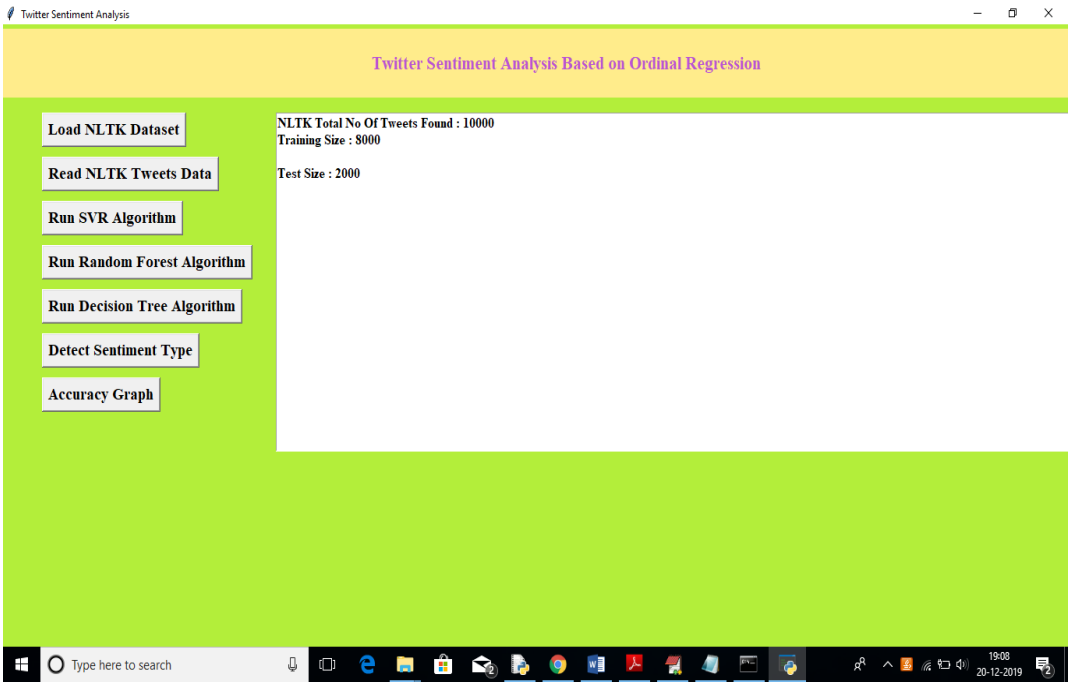
6.RESULTS



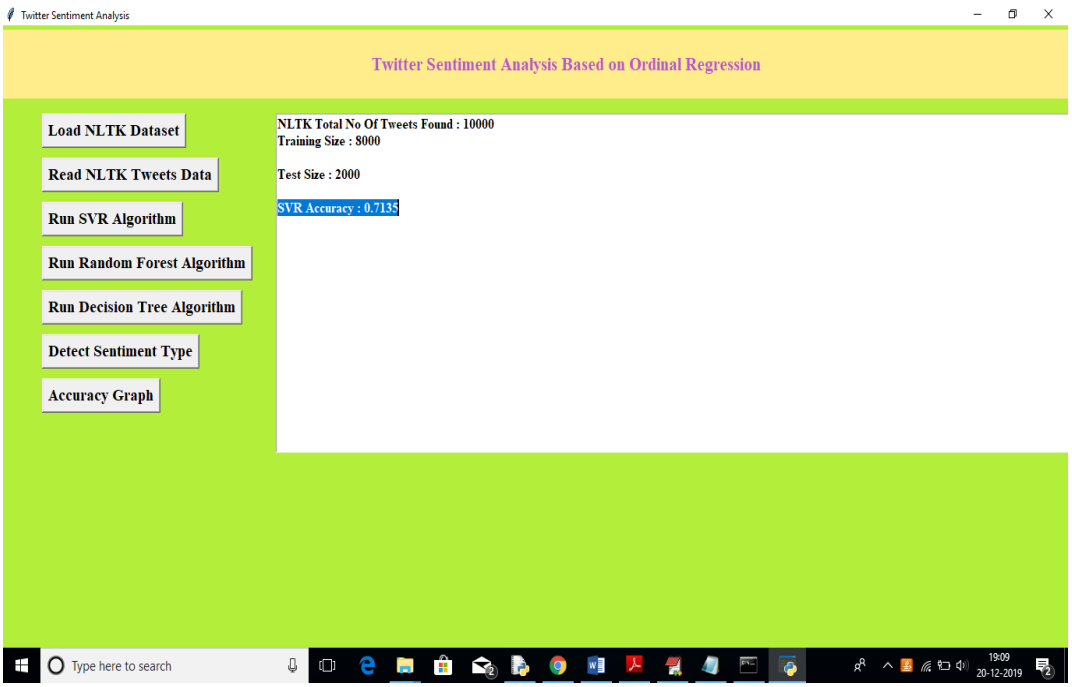
6.1 Load NLTK Dataset



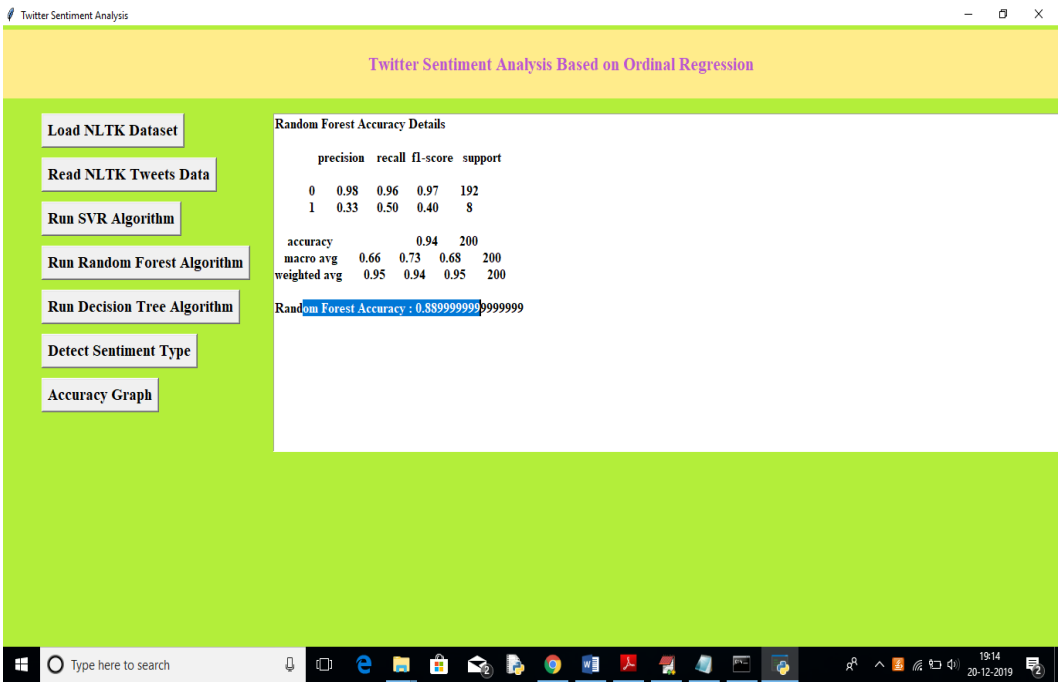
6.2 Read NLTK Tweets Data



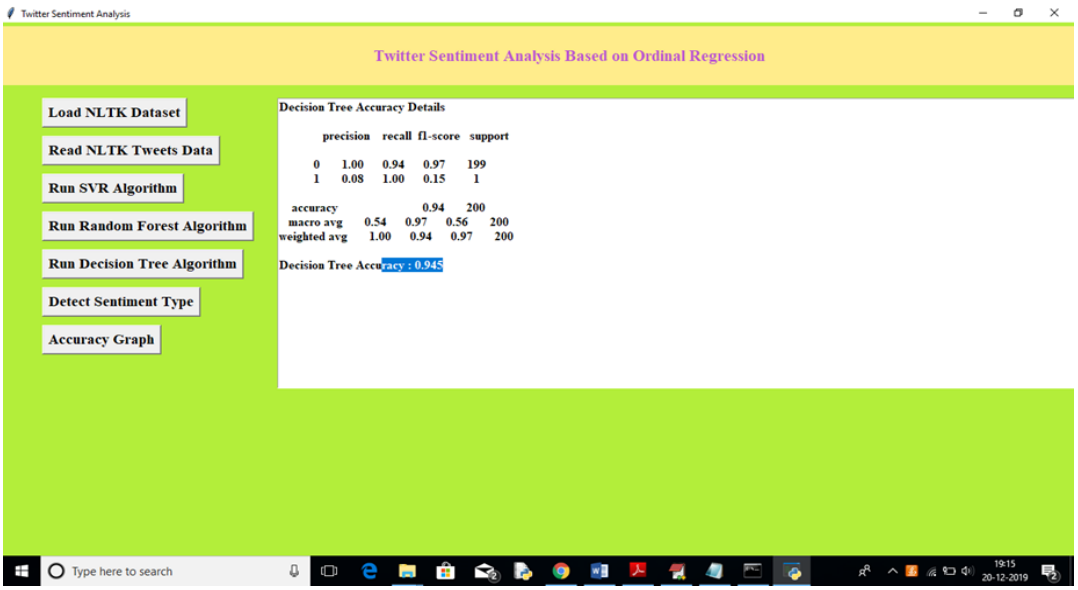
6.3 Run SVR Algorithm



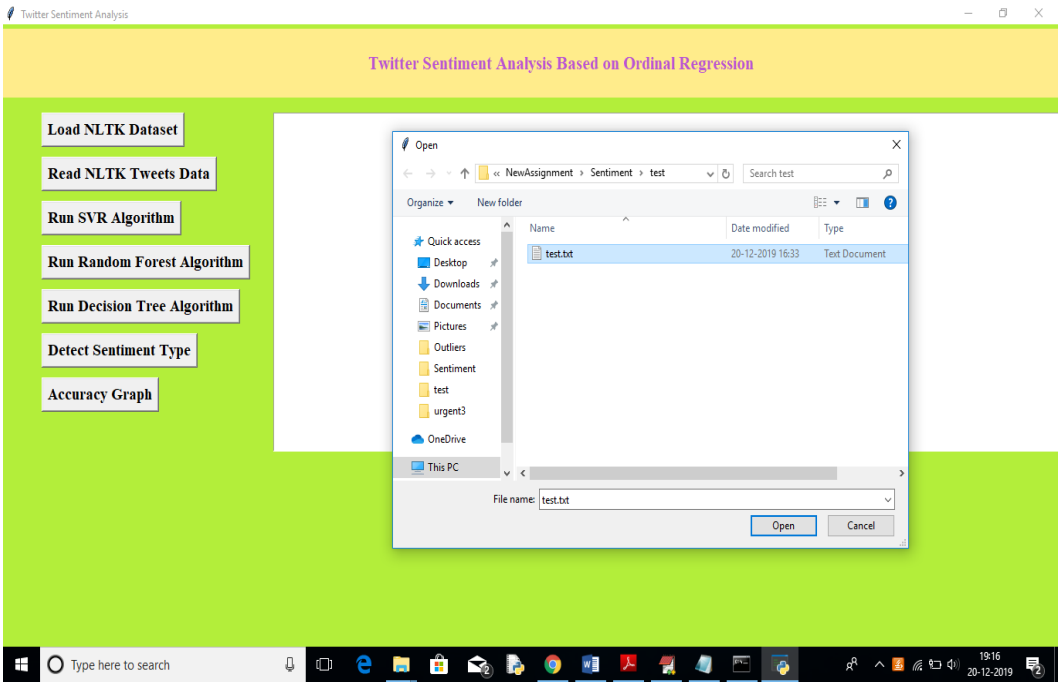
6.4 Run Random Forest Algorithm



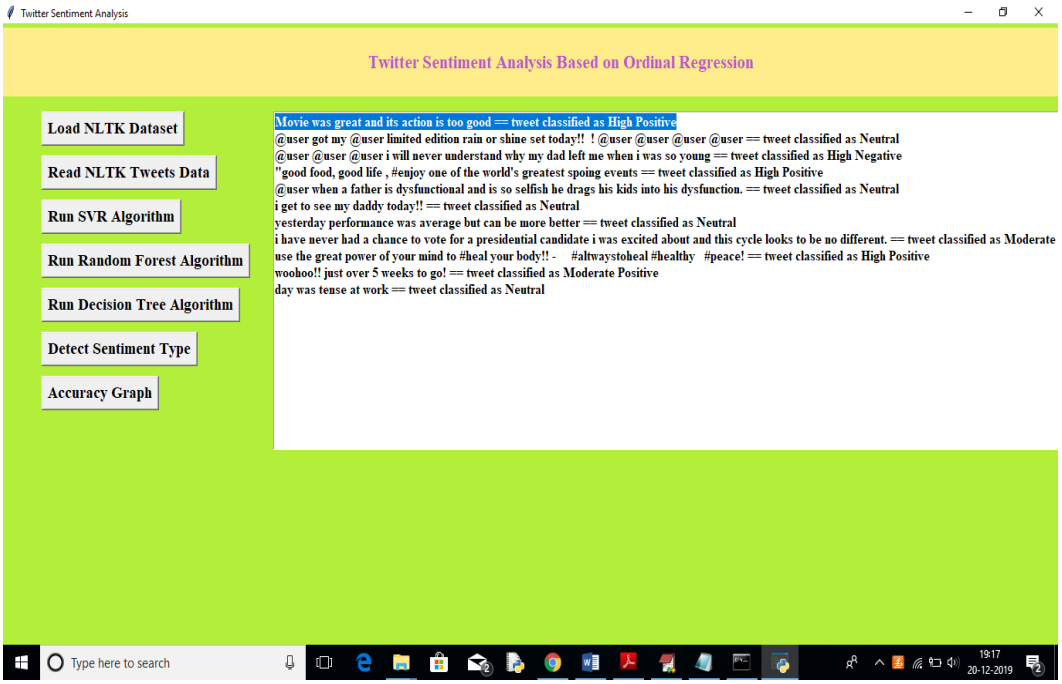
6.5 Run Decision Tree Algorithm



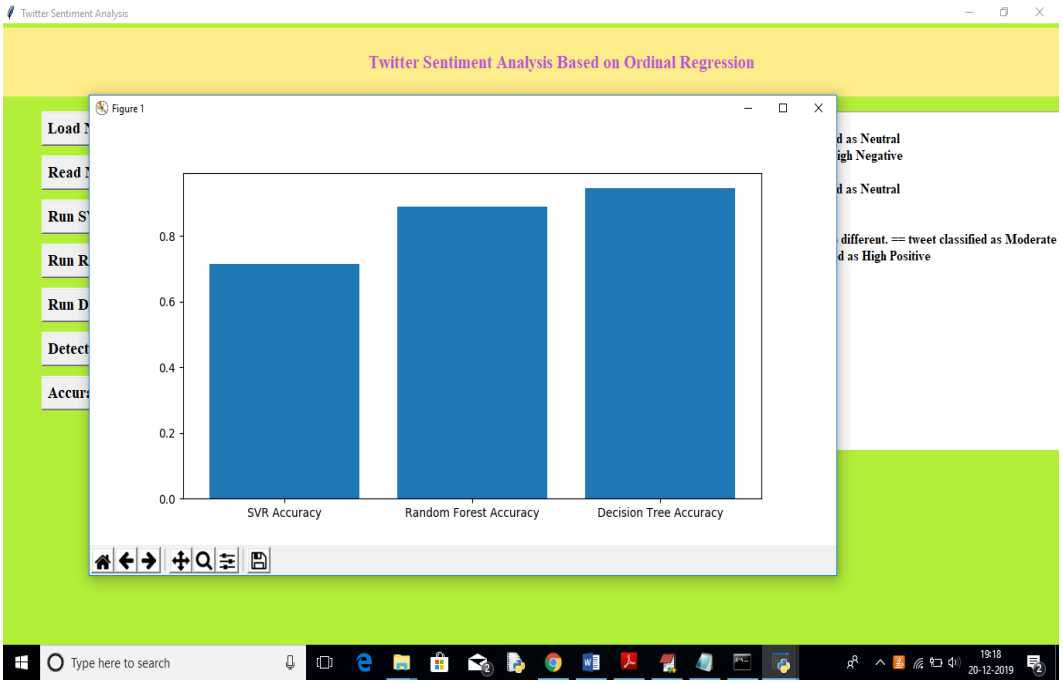
6.6 Detect Sentiment Type



6.7 Screen uploading test tweets file



6.8 Prediction results



6.9 Accuracy graph

7.TESTING

7.1 INTRODUCTION TO TESTING

The purpose of trying out is to discover errors. testing is the manner of trying to find out every achievable fault or weak spot in a work product. It offers a way to test the capability of components, sub-assemblies, assemblies and/or a finished product it is the system of exercising software with the rationale of ensuring that the software program system meets its requirements and consumer expectancies and does now not fail in an unacceptable way. there are numerous types of tests. every check type addresses a selected testing requirement.

7.2 TYPES OF TESTING

7.2.1 UNIT TESTING

Unit trying out entails the layout of test instances that validate that the inner application logic is functioning nicely, and that software inputs produce valid outputs. All selection branches and inner code glide need to be verified. it is the trying out of man or woman software gadgets of the utility .it is carried out after the completion of an man or woman unit earlier than integration. that is a structural trying out, that is predicated on information of its creation and is invasive. Unit exams perform fundamental exams at aspect level and take a look at a selected enterprise method, utility, and/or gadget configuration. Unit tests ensure that every particular direction of a enterprise procedure plays as it should be to the documented specs and contains really described inputs and predicted outcomes.

7.2.2 INTEGRATION TESTING

Integration tests are designed to check included software additives to determine in the event that they surely run as one application. testing is occasion. pushed and is extra involved with the fundamental final results of screens or fields.

Integration tests show that even though the additives have been in my opinion delight, as proven by using successfully unit checking out, the aggregate of components is correct and constant. Integration checking out is mainly geared toward exposing the troubles that get up from the mixture of components.

7.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is cantered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

7.3 TEST CASES

7.3.1 CLASSIFICATION

S.NO	INPUT	If available	If not available
1	Load NLTK Tweets	will load twitter sentiment corpora dataset from NLTK library	There is no process
2	Read NLTK Tweets	will calculate TFIDF vector	There is no process
3	Run SVR Algorithm	calculate prediction accuracy	There is no process
4	Run RF Algorithm	calculate prediction accuracy	There is no process
5	Run DT Algorithm	calculate prediction accuracy	There is no process
6	Detect Sentiment Type	predict sentiment of that tweet	There is no process
7	Accuracy Graph	display accuracy graph algorithms	There is no process

8. CONCLUSION & FUTURE SCOPE

8.1 PROJECT CONCLUSION

This study aims to explain sentiment analysis of twitter data regarding ordinal regression using several machine learning techniques. In the context of this work, we present an approach that aims to extract Twitter sentiment analysis by building a balancing and scoring model, afterward, classifying tweets into several ordinal classes using machine learning classifiers. Classifiers, such as Multinomial logistic regression, Support vector regression, Decision Trees, and Random Forest, are used in this study. This approach is optimized using Twitter data set that is publicly available in the NLTK corpora resources. Experimental results indicate that Support Vector Regression and Random Forest have an almost similar accuracy, which is better than that of the Multinomial logistic regression classifier. However, the Decision Tree gives the highest accuracy at 91.81%. Experimental results concluded that the proposed model can detect ordinal regression in Twitter using machine learning methods with a good accuracy result. The performance of the model is measured using accuracy, Mean Absolute Error, and Mean Squared Error.

8.2 FUTURE SCOPE

In the future, we plan to improve our approach by attempting to use bigrams and trigrams. Furthermore, we intend to investigate different machine learning techniques and deep learning techniques, such as Deep Neural Networks, Convolutional Neural Networks, and Recurrent Neural Networks.

9. BIBLIOGRAPHY

9.1 REFERENCES

- [1] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, ''From tweets to polls: Linking text sentiment to public opinion time series.,'' in Proc. ICWSM, 2010, vol. 11, nos. 122–129, pp. 1–2.
- [2] M. A. Cabanlit and K. J. Espinosa, ''Optimizing N-gram based text feature selection in sentiment analysis for commercial products in Twitter through polarity lexicons,'' in Proc. 5th Int. Conf. Inf., Intell., Syst. Appl. (IISA), Jul. 2014, pp. 94–97.
- [3] S.-M. Kim and E. Hovy, ''Determining the sentiment of opinions,'' in Proc. 20th Int. Conf. Comput. Linguistics, Aug. 2004, p. 1367.
- [4] C. Whitelaw, N. Garg, and S. Argamon, ''Using appraisal groups for sentiment analysis,'' in Proc. 14th ACM Int. Conf. Inf. Knowl. Manage., Oct./Nov. 2005, pp. 625–631.
- [5] H. Saif, M. FernÆndez, Y. He, and H. Alani, ''Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold,'' in Proc. 1st Interantional Workshop Emotion Sentiment Social Expressive Media, Approaches Perspect. AI (ESSEM), Turin, Italy, Dec. 2013.
- [6] A. P. Jain and P. Dandannavar, ''Application of machine learning techniques to sentiment analysis,'' in Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT), Jul. 2016, pp. 628–632.
- [7] A. Go, R. Bhayani, and L. Huang, ''Twitter sentiment classification using distant supervision,'' Processing, vol. 150, no. 12, pp. 1–6, 2009.
- [8] M. Bouazizi and T. Ohtsuki, ''A pattern-based approach for multi-class sentiment analysis in Twitter,'' IEEE Access, vol. 5, pp. 20617–20639, 2017.

- [9] R. Sara, R. Alan, N. Preslav, and S. Veselin, ‘‘SemEval-2016 task 4: Sentiment analysis in Twitter,’’ in Proc. 8th Int. Workshop Semantic Eval., 2014, pp. 1–18.
- [10] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, and V. Stoyanov, ‘‘Semeval-2015 task 10: Sentiment analysis in Twitter,’’ in Proc. 9th Int. Workshop Semantic Eval. (SemEval), Jun. 2015, pp. 451–463.

9.2 GITHUB LINK

https://github.com/ahmed-19-593/Tweets-Major-.git

IJARST

International Journal For Advanced Research In Science & Technology

A peer reviewed international journal www.ijarst.in

ISSN: 2457-0362

Tweets Categorization and Comparison of Results Using Machine Learning Models

1 MD.VAHEED, Department of CSE, CMR Technical Campus, Telangana, India,197r1a0591@cmrtc.ac.in

2 MUDASSAR AHMED KHAN, Department of CSE, CMR Technical Campus, Telangana, India,player9basketball@gmail.com

3 MD ABDUL RAHMAN, Department of CSE, CMR Technical Campus, Telangana, India, abdulrahaman16490@gmail.com

4 K. PRAVEEN KUMAR, (Assistant Professor),Department of CSE, CMR Technical Campus, Telangana, India, praveenkumar.cse@cmrtc.ac.in

ABSTRACT: As of late, there has been a huge expansion in the utilization of Twitter opinion examination, which utilizations Twitter information (tweets) to decide client perspectives in regards to a subject. The utilization of ML strategies for such investigations is liked by numerous scholastics. Utilizing ML and ordinal relapse, this work intends to lead a profound feeling examination of tweets. Pre-handling tweets is the most vital phase in the proposed technique, and afterward a compelling component is acquired through highlight extraction. Then, at that point, these qualities are separated into various gatherings for scoring and adjusting. Multinomial logistic regression (SoftMax), support vector regression (SVR), decision trees (DTs), and random forest (RF) are among the opinion examination order procedures used in the proposed framework. The genuine development of this framework is made conceivable by using a Twitter dataset that has been made accessible to people in general through the NLTK corpus assets. The exploratory results exhibit the way that the proposed framework can perceive ordinal backslide with high accuracy using ML moves close. Also, the outcomes show that Choice Trees outflank any remaining methodologies.

Keywords *– Twitter, the technique of machine learning, sentiment analysis, and ordinal regression.*

1. INTRODUCTION

In view of the quick extension of microblogging administrations and informal communities. One of the most generally involved web-based stages for people to offer their viewpoints, thoughts, and considerations on a large number of subjects are microblogging sites. 1], [2]. Twitter is a notable long range informal communication administration and famous microblogging stage that produces a great deal of information. Social information has as of late been inclined toward by scholastics for opinion investigation of individuals' viewpoints in regards to an item, issue, or occasion. Natural language processing depends intensely on feeling examination, otherwise called assessment mining. This technique decides if a message has a positive, negative, or impartial opinion direction [3, 4]. Right now, Twitter opinion investigation is a huge area of exploration.

IJARST
International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

Overwhelmingly of social information, this sort of study gathers and arranges popular assessment. In any case, opinion examination is more challenging for Twitter information than for different kinds of information because of various attributes. Tweets must be 140 characters in length, are written in easygoing English, contain different abbreviations and shoptalk terms, and are restricted to 140 characters. Scholastics have led tests zeroing in on tweet opinion examination to resolve these issues [5].

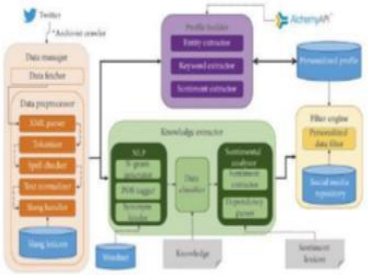


Fig.1: Example figure

T here are fundamentally two sorts of strategies for Twitter feeling examination: approaches in light of dictionaries and ML In this review, we break down Twitter opinion utilizing ML procedures. Most of characterization calculations are made to expect information marks for ostensible classes. Then again, there are various issues with design acknowledgment while utilizing an ordinal scale to foresee marks or classifications. Ordinal classification or ordinal relapse are terms for this. As of late, a ton of consideration has been paid to customary relapse. Issues including ordinal relapse are normal across a great many scholastic fields. They are oftentimes viewed as customary ostensible issues that can bring about lacking arrangements. For sure, arrangement and relapse issues can be applied to ordinal relapse issues with specific equals and contrasts. Clinical examination, age gauges, mind PC interface, face acknowledgment, facial engaging quality rating, picture characterization, sociologies, and text order are only a couple of the numerous uses of ordinal relapse. Some assessment propose using ML ways of managing tackle backslide issues to further develop the assessment examination portrayal execution of Twitter data and anticipate new disclosures. Improved results are the essential advantage of this methodology.

2. LITERATURE REVIEW

From tweets to polls: Linking text sentiment to public opinion time series:

Message based feeling measurements and prominent attitude estimations from surveys are connected. We take a gander at different reviews on customer sureness and political evaluation from 2008 to 2009 and find that they contrast and feeling word frequencies in contemporaneous Twitter posts. In spite of the fact that our outcomes vary from dataset to dataset, critical enormous scope designs are featured by connections as high as 80% in specific conditions. Text streams can possibly supplement and supplant ordinary surveys, as the discoveries accentuate.

Determining the sentiment of opinions:

It is challenging to distinguish opinions, which are perspectives' personal parts. We show a strategy that, given a subject, consequently decides individuals

IJARST
International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

who have those perspectives and how they feel about them. The framework's still up in the air by one module, and the blend of feelings in a not set in stone by another. We test various models for requesting and integrating assessment at the word and sentence levels, and the results are engaging.

Using appraisal groups for sentiment analysis:

Fine-grained semantic differences in classification features have never been used in sentiment analysis, which is used to categorize texts according to their "positive" or "negative" orientation. A novel method for sentiment categorization is presented here, and its foundation is the extraction and evaluation of assessment groups like "very good" or "not terribly funny." Various undertaking free semantic scientific categorizations portray an examination bunch as an assortment of characteristic qualities in light of Evaluation Hypothesis. Utilizing semi-computerized techniques, a word reference of surveying descriptors and their modifiers was made. With a accuracy of 90.2%, we order film audits utilizing highlights in light of these scientific classifications and conventional "pack of-words" highlights. Also, we find that for feeling order, a few evaluations have all the earmarks of being more significant than others.

Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold:

Twitter feeling investigation is a fast and compelling instrument for associations and people to screen public impression of them and their opponents. As of late, few evaluation datasets have been made to look at Twitter's exhibition of feeling examination calculations. In this review, we give a synopsis of eight physically clarified and freely open assessment datasets for Twitter opinion examination. We show that the shortfall of particular feeling explanations across tweets and the elements that are remembered for them is a typical shortcoming of most of these datasets while performing opinion investigation at the objective (substance) level. For example, the tweet "I love iPhone yet can't stand iPad" might be marked with blended feeling, yet the iPhone in this tweet should be named with good opinion. To address this impediment and supplement existing evaluation datasets, we give STS-Gold, a unique appraisal dataset in which tweets and targets (substances) are named freely and subsequently could show elective inclination marks. Likewise, this study looks at the different datasets in view of various qualities, including the complete number of tweets, jargon size, and sparsity. Moreover, we examine how feeling arrangement execution on different datasets is connected with the pair-wise connections that exist between these factors.

Application of machine learning techniques to sentiment analysis:

The "information age" is setting down deep roots. Organizations that set forth some part of energy to screen client criticism and remarks about their items presently approach a great many new open doors because of the fast development in the volume of client produced information via virtual entertainment stages like Twitter. Twitter is a monstrous microblogging long range informal communication webpage with a quick development rate where clients can voice their viewpoints on various themes,

IJARST
International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

including governmental issues, items, sports, and that's just the beginning. Organizations, state run administrations, and people all advantage according to these points of view. Tweets may thusly be a significant device for social occasion popular assessment. The strategy for deciding if client created message portrays a positive, negative, or nonpartisan assessment of a specific element is known as opinion investigation. e.g., individuals, item, occasion, and so on.). This work plans to give ML put together directions to opinion examination with respect to Twitter information. Also, the proposed technique for opinion examination is totally depicted in this review. Since it is worked with Apache Flash, the message examination system introduced in this paper for Twitter information is more versatile, speedy, and adaptable. Opinion examination is done utilizing the ML strategies Nave Bayes and Decision trees in the proposed framework.

3. METHODOLOGY

The majority of classification algorithms are made to anticipate data labels for nominal classes. On the other hand, there are a number of problems with pattern recognition when using an ordinal scale to predict labels or categories. Ordinal categorization or ordinal regression are terms for this. Recently, a lot of attention has been paid to ordinary regression. Problems involving ordinal regression are common across a wide range of academic fields. They are frequently regarded as ordinary nominal problems that can result in inadequate solutions.

Disadvantages:

1. result in inadequate solutions Issues with design acknowledgment

The difficulties presented by ordinal relapse are the essential focal point of this work, which centers around opinion examination of Twitter information (tweets) utilizing an assortment of ML techniques. Prior to characterizing tweets utilizing an assortment of ML draws near, we propose a technique that consolidates highlight extraction, pre-handling, and building a score and adjusting framework in this review.

Benefits:

1. Utilizing ML procedures, the trial results show the way that the proposed system can distinguish ordinal relapse with high exactness.

2. Furthermore, the information show that Decision Trees perform better compared to some other calculation.

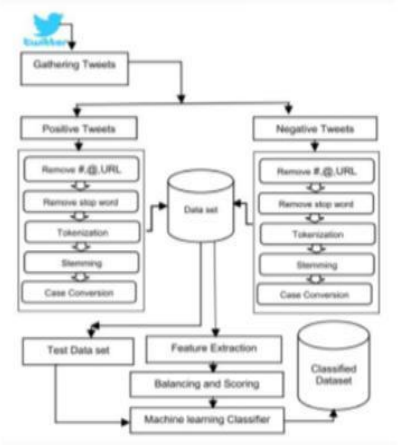


Fig.2: System architecture

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
IJARST
ISSN: 2457-0362

MODULES:

- For this project, we created the following modules.
- NLTK tweets to load: The Twitter sentiment corpus dataset from the NLTK library will be loaded using this module.
- Read Tweets by NLTK: We will read NLTK tweets using this module, clean them by removing special symbols, stopping words, and stemming each word (for example, ORGANIZATION will become ORGANIZE after applying stem). After that, the TFIDF vector will be calculated.
- Apply the SVR Algorithm: The TFIDF vector will be used as an input to train the SVR algorithm in this module. Eighty percent of the vector will be used for training in this method, and twenty percent will be used for testing. The framework then, at that point, determined forecast exactness by utilizing a 80% prepared model on 20% test information.
- In a similar vein, in order to determine whether or not they are correct, we will develop models for Decision Tree and Random Forest.
- Type of Sentiment: We will upload test tweets and use a train model to predict sentiment with the help of this module.
- Graph of Accuracy: An accuracy graph will be provided for each method by this module.

4. IMPLEMENTATION

Decision Tree: While choosing whether to isolate a hub into at least two sub-hubs, choice trees utilize various methodologies. The homogeneity of recently framed subnodes is reinforced by the advancement of subnodes. To put it another way, the hub's immaculateness expansions corresponding to the objective variable.

A choice tree is a strategy for non-parametric directed discovering that can be utilized for both characterization and relapse. A root hub, branches, inner hubs, and leaf hubs make up its various leveled tree structure.

SVM: Support Vector Machine (SVM) is a regulated technique for ML that can be utilized for both relapse and grouping. They are the most ideal for characterization, despite the fact that we allude to them as relapse issues. In a N-layered space, the goal of the SVM calculation is to find a hyperplane that plainly orders the info focuses.

SVMs are used in handwriting recognition, face recognition, intrusion detection, email classification, gene classification, and page generation. This is why SVMs are utilized in machine learning. It can handle regression and classification on linear and non-linear data.

In any case, most of its utilization is in ML for issues with characterization. The objective of the SVM calculation is to find the best line or choice limit for n-layered space arrangement so we can undoubtedly put new data of interest in the right class from now on.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
IJARST
ISSN: 2457-0362

Random Forest:: A well known regulated ML procedure for Characterization and Relapse issues is the Random Forest: technique. We know that there are a great deal of trees in a timberland, and the more trees there are, the more grounded the backwoods is. Random forest is utilized by data scientists on the job in a number of industries, including e-commerce, finance, stock trading, and medicine. It is used to forecast customer behavior, patient history, and safety, all of which contribute to the smooth operation of these businesses.

It is able to carry out both classification and regression tasks. Forecasts that are accurate and easy to comprehend are produced by a random forest. It is able to handle large datasets with ease. In terms of prediction accuracy, the random forest method performs better than the decision tree algorithm. The random forest method prevents overfitting by employing a large number of trees. The discoveries are inaccurate. As a result, the outcomes are precise. Because decision trees require less computation, their implementation time and accuracy are reduced.

5. EXPERIMENTAL RESULTS

Fig.3: Output

Fig.4: Output

Fig.5: Output

Fig.6: Output

IJARST
International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

Fig.7: Output

Fig.8: Output

Fig.9: Output

6. CONCLUSION

The aftereffects of the trial demonstrate the way that the proposed model can precisely recognize ordinal relapse in Twitter utilizing ML procedures. The precision, mean outright mistake, and mean squared blunder are utilized to survey the model's presentation.

7. FUTURE SCOPE

We expect to attempt to consolidate bigrams and trigrams in the future to work on our methodology. Deep Neural Networks, Convolutional Neural Networks, and Recurrent Neural Networks are only a couple of the ML and deep learning procedures we need to explore.

REFERENCES

[1] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series.," in Proc. ICWSM, 2010, vol. 11, nos. 122–129, pp. 1–2.

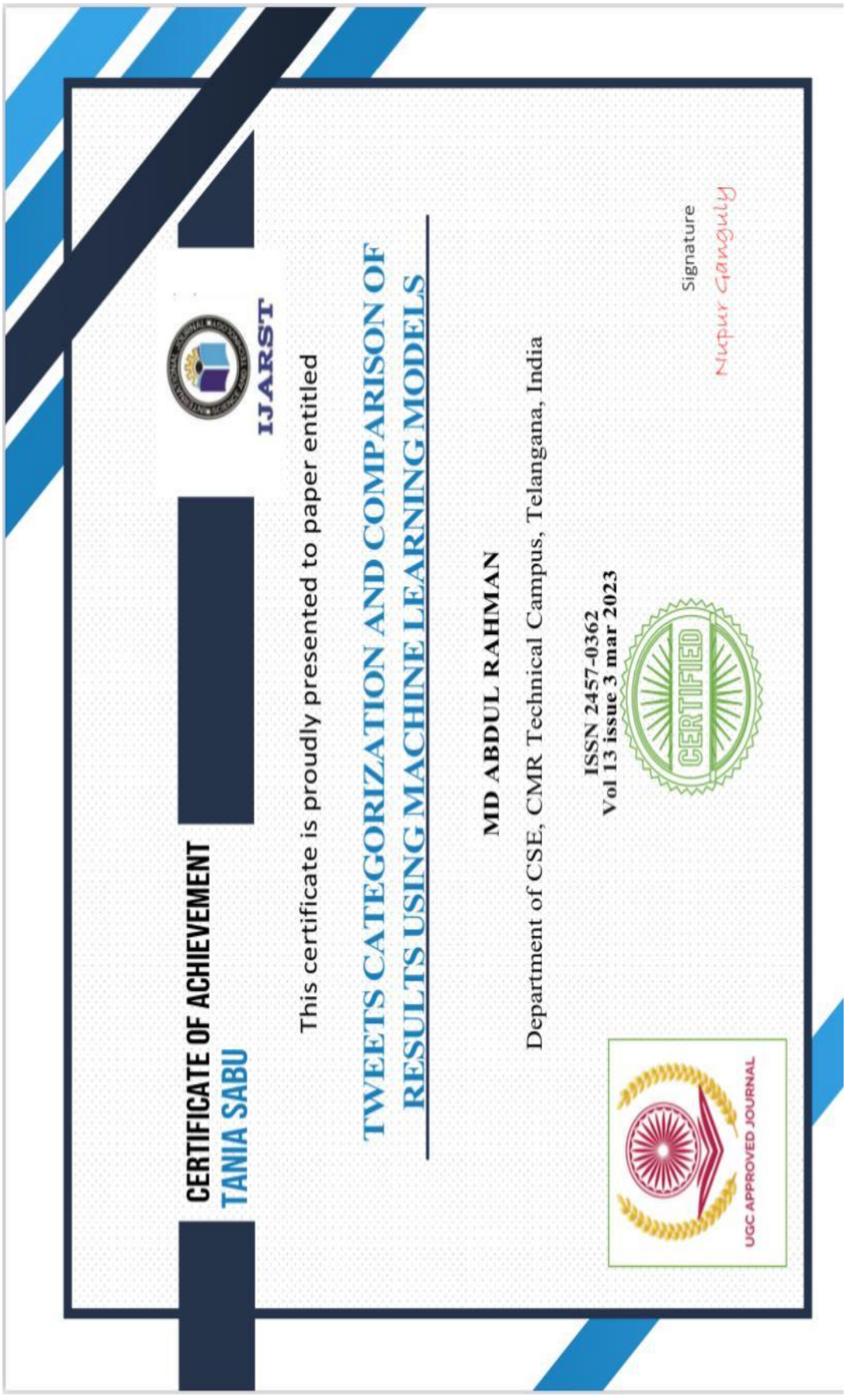
[2] M. A. Cabanlit and K. J. Espinosa, "Optimizing N-gram based text feature selection in sentiment analysis for commercial products in Twitter through polarity lexicons," in Proc. 5th Int. Conf. Inf., Intell., Syst. Appl. (IISA), Jul. 2014, pp. 94–97.

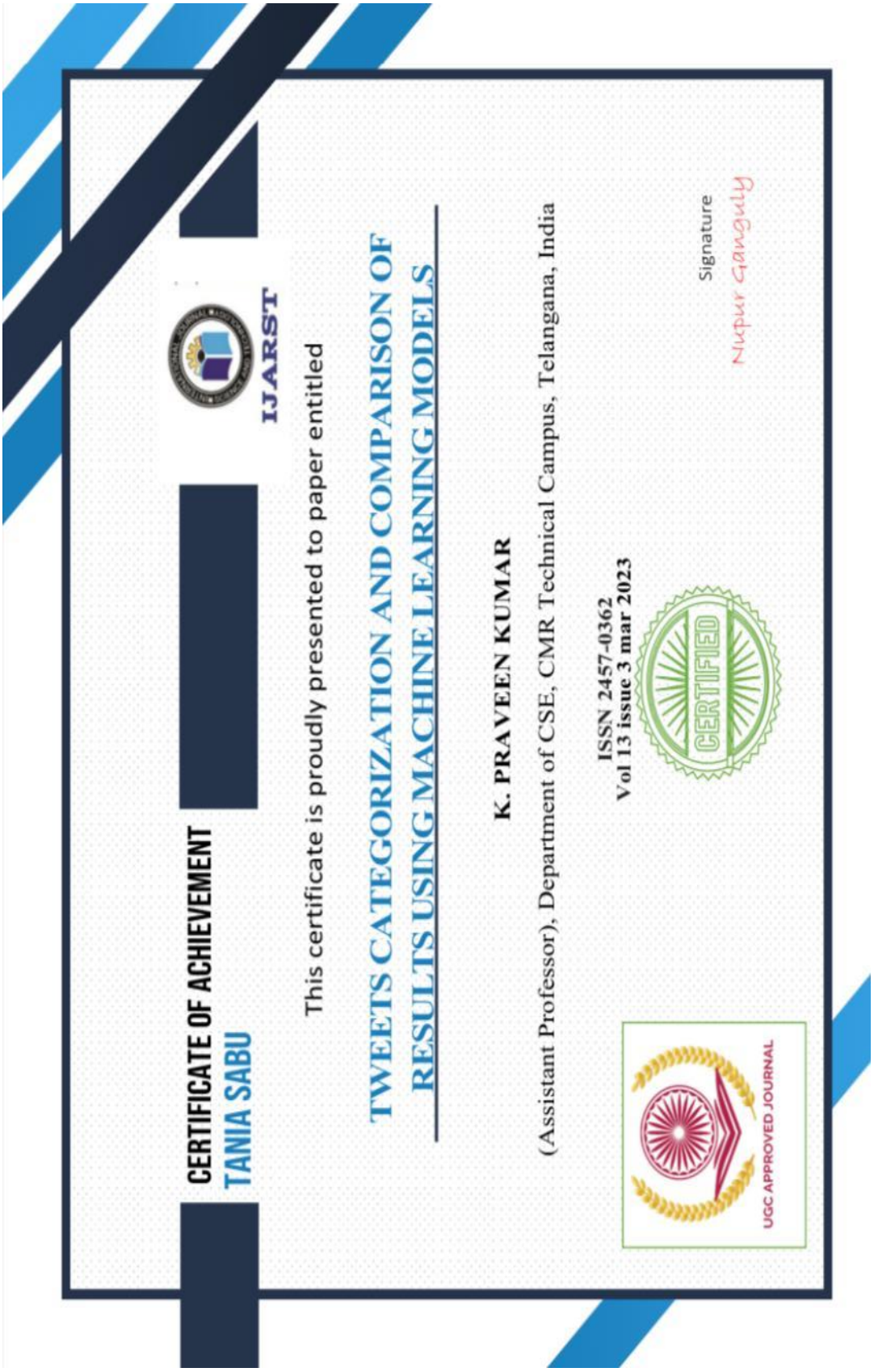
[3] S.-M. Kim and E. Hovy, "Determining the sentiment of opinions," in Proc. 20th Int. Conf. Comput. Linguistics, Aug. 2004, p. 1367.

[4] C. Whitelaw, N. Garg, and S. Argamon, "Using appraisal groups for sentiment analysis," in Proc. 14th ACM Int. Conf. Inf. Knowl. Manage., Oct./Nov. 2005, pp. 625–631.


CERTIFICATE OF ACHIEVEMENT
TANIA SABU
IJARST
This certificate is proudly presented to paper entitled
TWEETS CATEGORIZATION AND COMPARISON OF RESULTS USING MACHINE LEARNING MODELS
MUDASSAR AHMED KHAN
Department of CSE, CMR Technical Campus, Telangana, India
ISSN 2457-0362
Vol 13 issue 3 mar 2023
CERTIFIED
Signature
Nupur Ganguly
UGC APPROVED JOURNAL


CERTIFICATE OF ACHIEVEMENT
TANIA SABU
IJARST
This certificate is proudly presented to paper entitled
TWEETS CATEGORIZATION AND COMPARISON OF RESULTS USING MACHINE LEARNING MODELS
MD.VAHEED
Department of CSE, CMR Technical Campus, Telangana, India
ISSN 2457-0362
Vol 13 issue 3 mar 2023
CERTIFIED
Signature
Nupur Ganguly
UGC APPROVED JOURNAL


CERTIFICATE OF ACHIEVEMENT
TANIA SABU
IJARST
This certificate is proudly presented to paper entitled
TWEETS CATEGORIZATION AND COMPARISON OF RESULTS USING MACHINE LEARNING MODELS
MD ABDUL RAHMAN
Department of CSE, CMR Technical Campus, Telangana, India
ISSN 2457-0362
Vol 13 issue 3 mar 2023
CERTIFIED
Signature
Nupur Ganguly
UGC APPROVED JOURNAL


CERTIFICATE OF ACHIEVEMENT
TANIA SABU
IJARST
This certificate is proudly presented to paper entitled
TWEETS CATEGORIZATION AND COMPARISON OF RESULTS USING MACHINE LEARNING MODELS
K. PRAVEEN KUMAR
(Assistant Professor), Department of CSE, CMR Technical Campus, Telangana, India
ISSN 2457-0362
Vol 13 issue 3 mar 2023
CERTIFIED
Signature
Nupur Ganguly
UGC APPROVED JOURNAL