A Project report on

CLASSIFICATION FOR DETECTING INSULTING AND ABUSIVE CONTENT

Project submitted to the Jawaharlal Nehru Technological University in
Partial fulfillment of the requirements for the award of the Degree of Bachelor of
Technology in Computer Science and Engineering

Submitted By

M. Swetha
(16891A0532)

K. Mahesh
(16891A0529)

K.V.S.Dheeraj
(16891A0522)

Under the Guidance of
Mr. B. Srinu
Associate Professor

Department of Computer Science and Engineering,

VIGNAN INSTITUTE OF TECHNOLOGY AND SCIENCE, Deshmukhi

Affiliated to Jawaharlal Nehru Technological University. Hyderabad

2019-2020

VIGNAN INSTITUTE OF TECHNOLOGY AND SCIENCE

(Programs accredited by NBA)

Deshmukhi (V), Pochampally (M), Yadadri-Bhuvanagiri District, Telangana, 508284.

Department of Computer Science & Engineering

Vision

To emerge as a premier center for education and research in computer science and engineering and in transforming students into innovative professionals of contemporary and future technologies to cater the global needs of human recourses for IT and ITES companies .

Mission

- To produce excellent computer science professionals by imparting quality training, hands-on-experience and value based education.
- To strengthen links with industry through collaborative partnerships in research & product development and student internships.
- To promote research based projects and activities among the students in the emerging areas of technology.
- To explore opportunities for skill development in the application of computer science among rural and under privileged population.

Program Educational Objectives

- To create and sustain a community of learning in which students acquire knowledge and apply in their concerned fields with due consideration for ethical, ecological, and economic issues.
- To provide knowledge based services so as to meet the needs of the society and industry.
- To make the students understand, design and implement the concepts in multiple arenas.
- To educate the students in disseminating the research findings with good soft skills so as to become successful entrepreneurs.

VIGNAN INSTITUTE OF TECHNOLOGY AND SCIENCE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report titled **Classification for detecting insulting and abusive content** is being submitted by **M. Swetha**, bearing 16891A0532**, K. Mahesh Reddy**, bearing 16891A0529 **, K.V.S. Dheeraj** , bearing 16891A0522 in IV BTech I semester *Computer Science & Engineering* is a record bonafide work carried out by them. The results embodied in this report have not been submitted to any other University for the award of anydegree.

Mr. B. Srinu
Associate Professor

Mr. G. Raja Vikram
Head of the Department

PROJECT EVALUATION CERTIFICATE

This is to certify that the Project work entitled **" CLASIFICATION FOR DETECTING INSULTING AND ABUSIVE CONTENT "** submitted by **M. Swetha** (16-532), **K. Mahesh** (16-529),**K.V.S. Dheeraj**(16-522) has been examined and adjudged as sufficient for the partial fulfilment of the requirement of the degree of Bachelor of Technology in **Computer Science and Engineering** of Jawaharlal Nehru Technological University, Hyderabad.

External Examiner : ______________________________

(Signature with Date)

Internal Examiner : ______________________________

(Signature with Date)

Head of the Department : ______________________________

(Signature with Date)

ACKNOWLEDGEMENT

Success will be crowned to people who made it reality but the people whose constant guidance and encouragement made it possible will be crowned first on the eve of success.

This acknowledgement transcends the reality of formality when we would like to express deep gratitude and respect to all those people behind the screen who guided, inspired and helped us for the completion of our project work.

We would like to thank respected Mr. G. Raja Vikram, Head of the Department for giving us such a wonderful opportunity to expand our knowledge. It helped us a lot to realize of what we study for. Secondly, we would like to thank our guide Mrs. B. Srinu , Associate Professor for guiding us at every step and helping us know how to complete a project. Thirdly, we would like to thank our parents who patiently helped us as we went through our work and helped to modify and eliminate some of the irrelevant or un-necessary stuffs. Next, we would thank Microsoft for developing such a wonderful tool like MS Word. It helped my work a lot to remain error-free.

We also extend our thanks to all the staff of the department of Computer Science and Engineering, VITS for their co-operation and support during our course work. And, we would like to thank our friends too who helped us to make our work more organized and well- stacked till the end.

Last but clearly not the least, we would thank the Almighty for giving us strength to complete our work on time.

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

M. SWETHA (16-532)
K. MAHESH (16-529)
K.V.SAI DHEERAJ (16-522)

CONTENTS

ABSTRACT

The sheer simplicity with which abusive and insulting comments can be made on the web – normally from the solace of your home and the absence of any quick negative repercussions – utilizing the present computerized correspondence innovations (particularly web based life), is liable for their noteworthy increment and worldwide universality. Natural Language Processing advancements can help in tending to the negative impacts of this improvement. In this project, we assess a lot of classification calculations on two sorts of client created online substance in two English. The various arrangements of information we deal with were grouped towards aspects, for example, prejudice, sexism, hate speech, animosity and individual assaults. While recognizing issues with between annotator understanding for grouping errands utilizing these names, the focal point of this project is on arranging the information as per the commented on qualities utilizing a few Text Classification Algorithms finally detecting which one is more suitable for the problem.

LIST OF FIGURES

LIST OF SCREENS

1. INTRODUCTION

Hateful conduct, abusive language and verbal aggression are by no means new phenomena. Comments and statements of this type seriously hamper a constructive privat discussion or publi debate. The sheer ease with which hateful utterances can be made – typicall from the comfort o your home and the lack of any immediate negative repercussions – using today's digital communication technologies, is responsible for their significant increase and global ubiquity. I recent years, the topic has received an increasing amount of attention fro multiple stakeholders. Among these are social scientists who want to analyze this phenomenon and reasons for abusive online behaviour and politicians who realize that major parts of public debates and social discourse are carried out online. In addition, we have seen that not only such online discussions but also the perception of concepts, politicians, elections and civil rights movements can be influenced using highly targeted social media marketing campaigns. We live in a time in which online media, including online news and online communication, have an unprecedented level of social, political and also economic relevance. This situation creates a plethora of challenges with regard to the key question how best to address the importance and relevance of online media and online content with technological means while at the same time no putting in place a centralized infrastructure that ca be misused for the purpose of censorship o surveillance. One challenge is to separate high qualit content from offensive, hateful, abusive or massively biased content. While these tasks have been mostly in the realm of journalism, they are getting more and more transferred to the end user of online content, i.e., the analysis, curation an assessment of information is no longer carried out b professional news editors or journalist exclusively – the burden of fact checking is more and mor left to the reader.

1.1. MOTIVATION

A central challenge for automatic abusive content detection on social media is the separation of hate speech from other instances of offensive language. There are important qualitative differences between different types of potentially abusive language that need to be considered. For example a tweet quoting rap lyrics that contain potentially racist or sexist terms should not be regarded in the same way as a tweet that directs racist slurs at another user.

Existing work often does not make this distinction and many types of abusive language are often conflated.

1.2. PROBLEM DEFINITION

The sheer simplicity with which abusive and insulting comments can be made on the web normally from the solace of your home and the absence of any quick negative repercussions – utilizing the present computerized correspondence innovations (particularly web based life), is liable for their noteworthy increment and worldwide universality. Natural Language Processing advancements can help in tending to the negative impacts of this improvement. In this project, we assess a lot of classification calculations on two sorts of client created online substance in two English. The various arrangements of information we deal with were grouped towards aspects, for example, prejudice, sexism, hate speech, animosity and individual assaults. While recognizing issues with between annotator understanding for grouping errands utilizing these names, the focal point of this project is on arranging the information as per the commented on qualities utilizing a few Text Classification Algorithms finally detecting which one is more suitable for the problem.

1.3. OBJECTIVE

The main objective of the project is to make a NLP model which can take text for classifying the text whether it is abusive or not based on the past data. Based on the main factors that affect model, it removes stop words, symbols which are not required we identify the independent and dependent variables. A Deep learning algorithm prediction model in which the actual delay time corresponded to the dependent variable is established via Python3.6. Finally, the prediction accuracy of the random forest model and artificial neural network model is compared. The results indicate that the using neural networks the model outperforms other models.

2. LITERATURE SURVEY

2.1. EXISTING SYSTEM

Classification for Detecting insulting and Abusive Content has been implemented using several algorithms in many different technologies. Although, there is no clarity regarding which one to be used in what scenarios. Hence, a comparative analysis needs to be done**.**

2.1.1. DISADVANTAGES

- Many different algorithms has been implemented with no clarity on which one to use in whattype of scenarios.
- Older systems usually need huge amount of trainingdata

2.2. PROPOSED SYSTEM

In our purposed system, Python based Deep learning algorithms are being implemented for classifying the Abusive and insulting content. Further, these algorithms will be compared altogether to understand the most suitable algorithm for the task.

2.2.1. ADVANTAGES

- Usage of Machine Learning Algorithm makes the system more reliable andaccurate.
- Higher accuracy can be used even with less amount of trainingdata.

2.3. MODULES USED

2.3.1. NUMPY

- NumPy is a library for the Python programming language, adding support for large,multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originallycreated by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Num array into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. The Python programming language was not initially designed for numericalcomputing.
- Attracted the attention of the scientific and engineering community early on, so that aspecial interest group called matrix-sig was founded in 1995 with the aim of defining an array computing package. Among its members was Python designer and maintainer Guido van Rossum, who implemented extensions to Python's syntax (in particular the indexing syntax) to make array computingeasier.

2.3.2. PANDAS

- Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high- level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis manipulation tool available in any language. It is already well on its way toward this goal.
- Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three clause BSD license. The name is derived from the term "panel data", an economic termfor data sets that include observations over multiple time periods for the sameindividuals.

Features

- Data Frame objects for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.

2.3.3. SEABORN

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

- A data set- oriented API for examining relationships between multiple variables.
- Specialized support for using categorical variables to show observations or aggregate statistics.
- Options for visualizing univariate or bi variate distributions and for comparing them between subsets of data.
- Automatic estimation and plotting of linear regression models for different kinds dependent variables.
- Convenient views onto the overall structure of complex datasets.
- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations.
- Seaborn aims to make visualization a central part of exploring and understanding data. Its data set- oriented plotting functions operate on data frames and arrays containing whole data sets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

2.3.4. NATURAL LANGUAGE TOOLKIT(NLTK)

- The Natural Language Toolkit is an open source library for the Python programming language originally written by Steven Bird, Edward Loper and Ewan Klein for use in development and education.It comes with a hands-on guide that introduces topics in computational linguistics as well as programming fundamentals for Python which makes it suitable for linguists who have no deep knowledge in programming, engineers and researchers that need to delve into computational linguistics, students and educators.
- NLTK includes more than 50 corpora and lexical sources such as the Penn Tree bank Corpus, Open Multilingual Word net, Problem Report Corpus, and Lin's Dependency Thesaurus.

2.3.5. NLTK - STOP WORDS

- Natural Language Processing with Python Natural language processing (NLP) is a research field that presents many challenges such as natural language understanding.
- Text may contain stop words like 'the', 'is', 'are'. Stop words can be filtered from the text to be processed. There is no universal list of stop words in NLP research, however the NLTK module contains a list of stop words.

2.3.6. SCIKIT - LEARN

- Scikit - learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.
- Scikit-learn is largely written in Python, and uses numpy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible. It integrates well with many other Python libraries, such as matplotlib and plotly for plotting, numpy for array vectorization, pandas dataframes, scipy, and many more.

2.3.7. KERAS

- Keras is TensorFlow's implementation of the Keras API specification. This is a high-level API to build and train models that includes first-class support for TensorFlow - specific functionality, such as eager execution, tf.data pipelines, and Estimators. tf.keras makes TensorFlow easier to use without sacrificing flexibility and performance.
- Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or Plaid ML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.
- Keras has a simple, consistent interface optimized for common use cases. It provides clear and actionable feedback for user errors.
- Keras models are made by connecting configurable building blocks together, with few restrictions.
- Easy to extend.

3. ANALYSIS

3.1. INTRODUCTION

This software application is a python script that uses various packages that is provided under the Python IDE. The various packages that used are numpy for reading a particular file, NLP for recording the pre - processing , tkinter for providing the GUI interface, sklearn to apply various algorithms, likewise pandas, scipy, statistics etc., and packages are installed.

3.2. TECHNOLOGY REQUIREMENTS

- A data set to train the model which contains the train schedule, delay information etc.
- Machine learning algorithms like Random Forest, RNN etc.

3.2.1. SOFTWARE REQUIREMENTS

The software requirements in this project include:

- Languages used : Python
- Software : Python IDE, jupyter
- User Interface Design : Python

3.2.2. HARDWARE REQUIREMENTS

The Hardware requirements in this project include:

- RAM : 4 GB
- Operating System : Windows 7,8,9
- Processor : Pentium 4 or above

3.3 CONTENT DIAGRAM OF PROJECT

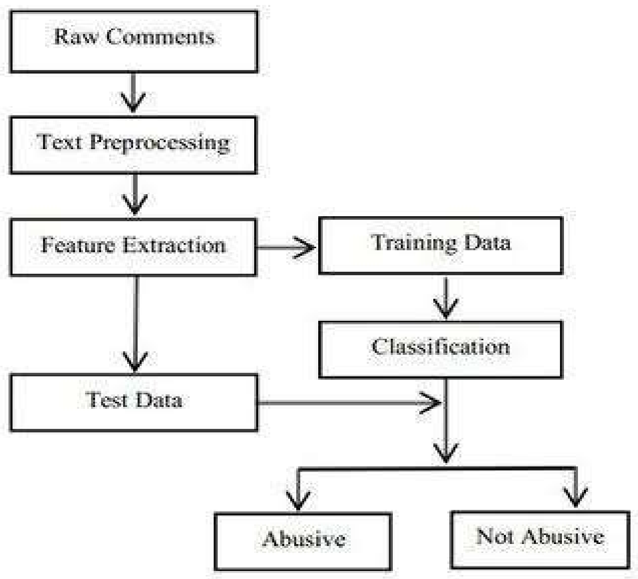


Fig. 3.1 Content Diagram

3.4 SYSTEM ARCHITECTURE OF PROJECT

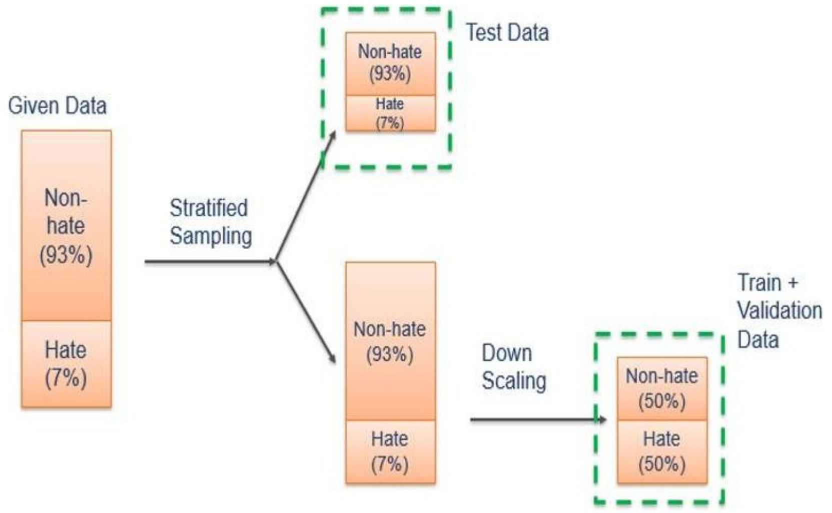


Fig 3.2 System Architecture

3.5 SYSTEM FLOWCHART

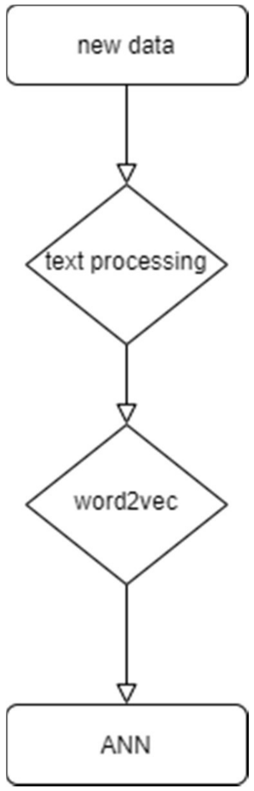


Fig. 3.3 System Flowchart

This flow charts depicts the flow of process which we have followed for our project. First going through collection of data then pre processing it using various NLP techniques, which can used with the help of NLTK library and its sub packages.Further vectorization is used to form vector for each word and then formation of model.

4. DESIGN

4.1 INTRODUCTION

System study is the operation performed by a system and their relationship inside and outside. The boundaries of the system should be defined and also analyzed whether the candidate system considers the other related system. During system study form collected and the available file or documents decision point and transaction handled by the present available system according to the requirements.

4.2 DIAGRAMS

The unified modeling language is a standard language for specifying, Visualizing, Constructing and documenting the software system and its components. It is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain and control Information about the systems.

4.2.1. DFD DIAGRAM

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations.
- It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

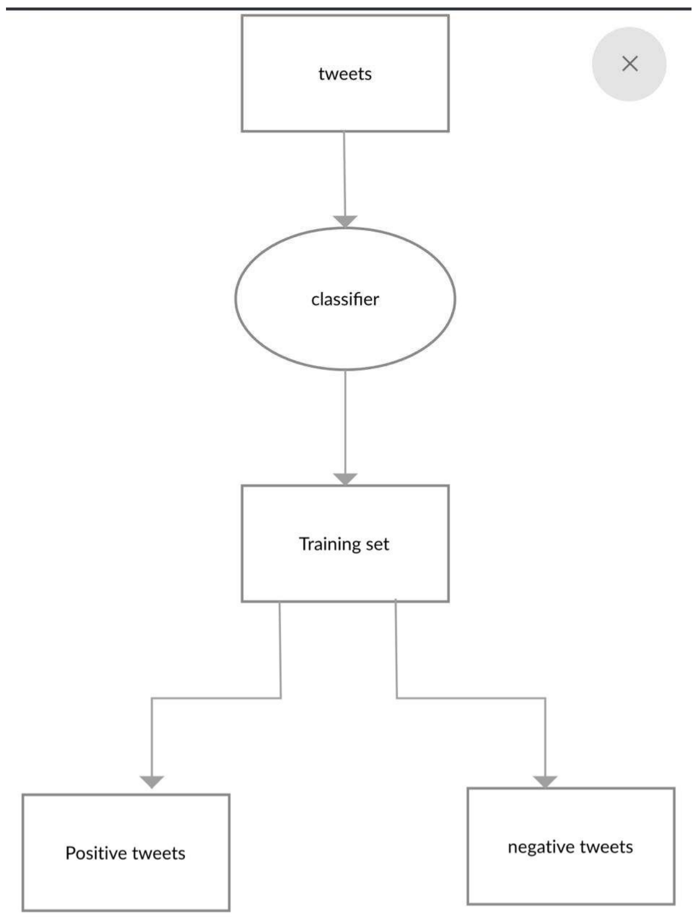


Fig. 4.2 DFD Diagram

4.2.2. UML DIAGRAMS

- UML stands for Unified Modeling Language. UML is a standardized general purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

- The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

- The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- higher level development concepts such as collaborations, frameworks, patterns and components.

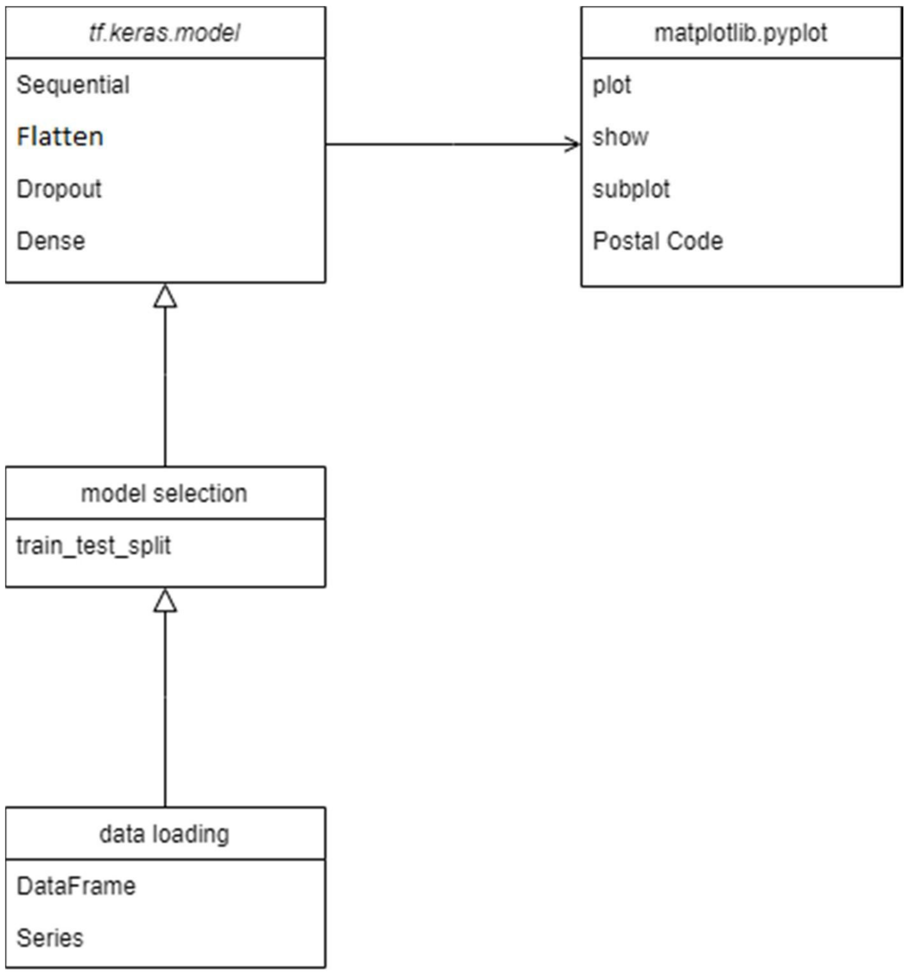


Fig. 4.2 UML Diagram

4.2.3. USE CASE DIAGRAM

A use case diagram acts as a focus for the description of user requirements. It describes the relationships between requirements, users, and the major components. It defines the interactions between external actors and the system under consideration to accomplish a goal.

Actors must be able to make decisions, but need not be human: "An actor might be a person, a company or organization, a computer program, or a computer system-hardware, software, or both."

Actors are always stakeholders, but not all stakeholders are actors, since they "never interact directly with the system, even though they have the right to care how the system behaves."

For example, "the owners of the system, the company's board of directors, and regulatory bodies such as the Internal Revenue Service and the Department of Insurance" could all be stakeholders but are unlikely to be actors. Actors are often working on behalf of someone else.

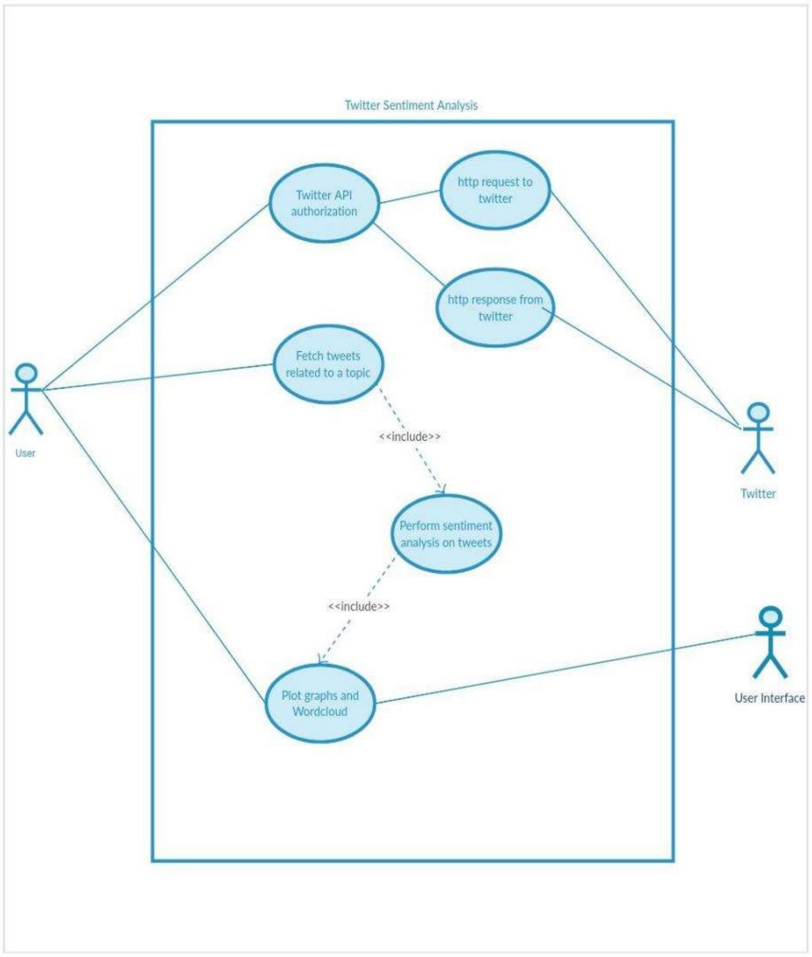

Twitter Sentiment Analysis
Twitter API authorization
http request to twitter
http response from twitter
Fetch tweets related to a topic
<<include>>
Perform sentiment analysis on tweets
<<include>>
Plot graphs and Wordcloud
User
Twitter
User Interface

Fig.4.3 Use case Diagram

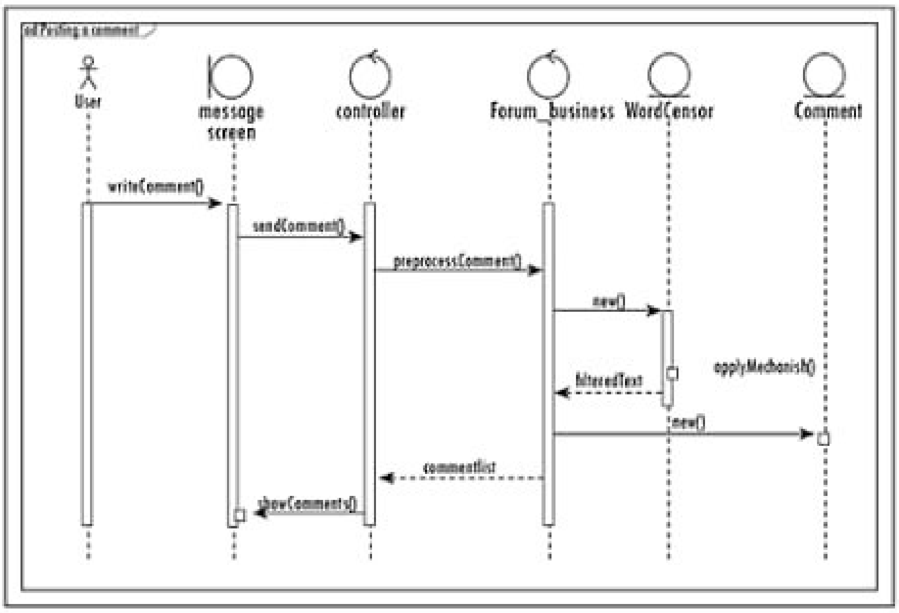
4.2.4. SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple run time scenarios in a graphical manner.

Messages, written with horizontal arrows with the message name written above them, display interaction.Solid arrow heads represent synchronous calls, open arrow heads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine. If a caller sends an asynchronous message, it can continue processing and doesn't have to wait for a response.

Figure 9. *Posting a comment* sequence diagram



Source: authors' own elaboration

Fig. 4.4 Sequence Diagram

4.2.5. ACTIVITY DIAGRAM

- Activity diagrams are graphical representations of work flows of stepwise activities and actions with support for choice, iteration and concurrency.
- In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by step work flows of components in a system.
- An activity diagram shows the overall flow of control.

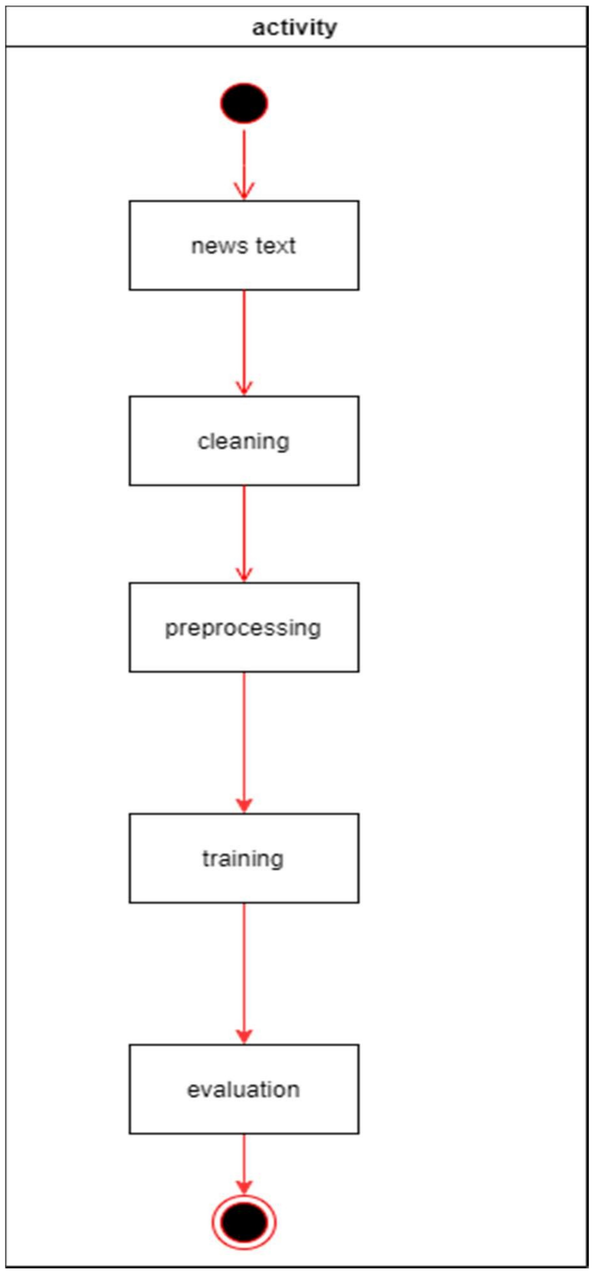


Fig. 4.5 Activity Diagram

5. METHODOLOGY

5.1 INTRODUCTION

The project uses various machine learning algorithms like Random forest , Logistic Regression fine tuned and Neural networks.

5.2. Machine learning

It is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering, and computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

5.3 Random forests

The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.
Random forest is a supervised learning algorithm which is used for both classification as well as regression. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting

5.4 Decision Tree

There are many types of machine learning methods. One of them is decision tree. A decision tree is similar looking to a tree diagram. A tree diagram begins with a single node and from that node, branches will reach out to new nodes that represent mutually exclusive decisions or events, meaning that they cannot occur simultaneously and is not influenced by the other.

The diagram starts by the first node and there after decisions and events will bring it onto the next node. So in other words, it is a sequence of events and this is useful in probability since it can record all the possible outcomes by adding more branches. If it is used to calculate probabilities, the probability is then put onto the separate branches and the outcome is the next-coming node.

A decision tree is a set of questions connected in a tree, so that answering one question leads to another and eventually to a final answer. It can be used for either regression or classification problems. When it is used to solve regression problems it is called regression tree. A decision tree is a directed tree and has one root node without any incoming edges. Other nodes have exactly one incoming edge and they are grown by a conditional split that divides the result into two new branches where the tree can continue to grow. There can be consecutive questions with different conditions by each new branch, creating additional branches that form the tree in a downward matter, or else the branch will end there. The ends of a branch are called a leaf or a terminal node and it displays the observation that falls into that specific branch.

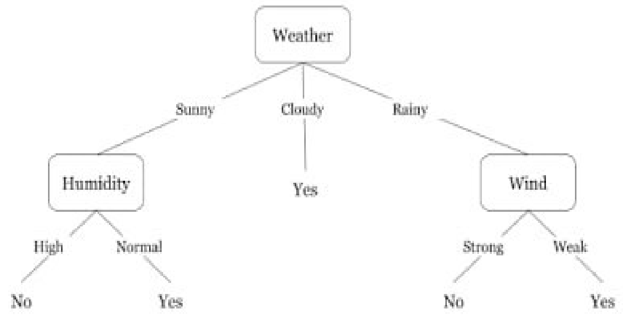


Fig. 5.1 Decision tree

The splits are done with a greedy top-down approach; the best split is done in each step even if another split could have created a better tree in a future step.

The predictor space (the set of possible input values) is split into the regions $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ where a predictor X_j and cut point s are selected such that the sum of squared errors is reduced the most.

5.5 K-Nearest Neighbor

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.

In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation.

Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of $1/d$, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

5.6 Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.

In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary).

Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Logistic regression is a statistical technique used to predict probability of binary response based on one or more independent variables. It means that, given a certain factors, logistic regression is used to predict an outcome which has two values such as 0 or 1, pass or fail, yes or no etc.

5.7. Neural Networks

Biological Neurons (also called nerve cells) or simply neurons are the fundamental units of the brain and nervous system, the cells responsible for receiving sensory input from the external world via dendrites, process it and gives the output through Axons.

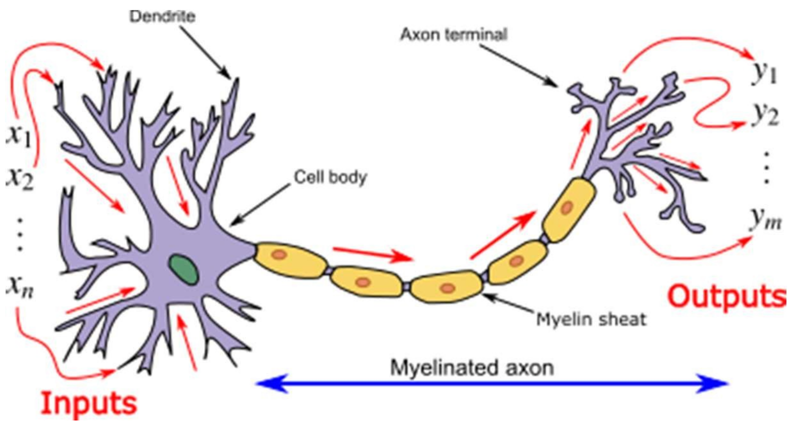


Fig. 5.2 A biological Neuron

- **Cell body (Soma):** The body of the neuron cell contains the nucleus and carries out biochemical transformation necessary to the life of neurons.

- **Dendrites:** Each neuron has fine, hair-like tubular structures (extensions) around it. They branch out into a tree around the cell body. They accept incoming signals.

- **Axon:** It is a long, thin, tubular structure that works like a transmission line.
- **Synapse:** Neurons are connected to one another in a complex spatial arrangement. When axon reaches its final destination it branches again called terminal arborization. At the end of the axon are highly complex and specialized structures called synapses.

The connection between two neurons takes place at these synapses. Dendrites receive input through the synapses of other neurons. The soma processes these incoming signals over time and converts that processed value into an output, which is sent out to other neurons through the axon and the synapses.

How does the Neural network work?

Let us take the example of the price of a property and to start with we have different factors assembled in a single row of data: Area, Bedrooms, Distance to city and Age.

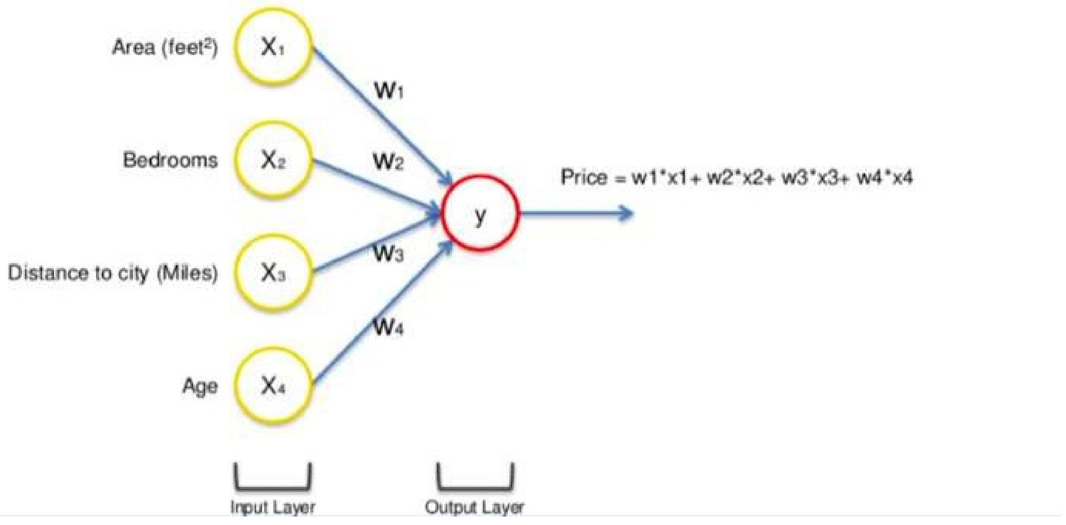


Fig. 5.3 Neural networks

1) The input values go through the weighted synapses straight over to the output layer.

2) All four will be analyzed, an activation function will be applied, and the results will be produced.

3) This is simple enough but there is a way to amplify the power of the Neural Network and increase its accuracy by the addition of a hidden layer that sits between the input and output layers.

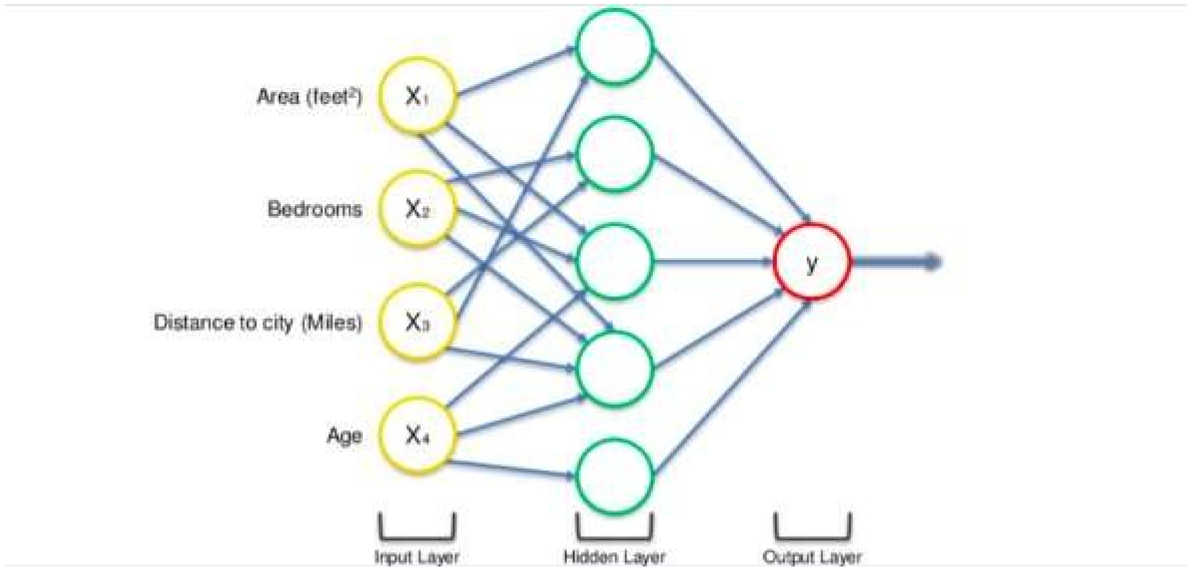


Fig. 5.4 Hidden layer

- A neural network with a hidden layer(only showing non-0 values).

- Now in the above figure, all 4 variables are connected to neurons via a synapse. However, not all of the synapses are weighted. they will either have a 0 value or non-0 value.
  here, the non-0 value → indicates the importance
  0 value → They will be discarded.

- Let's take the example of Area and Distance to City are non-zero for the first neuron, which means they are weighted and matter to the first neuron. The other two variables, Bedrooms and Age aren’t weighted and so are not considered by the first neuron. You may wonder why that first neuron is only considering two of the four variables. In this case, it is common on the property market that larger homes become cheaper the further they are from the city.

- That’s a basic fact. So what this neuron may be doing is looking specifically for properties that are large but are not so far from the city.

- Now, this is where the power of neural networks comes from. There are many of these neurons, each doing similar calculations with different combinations of these variables.

- Once this criterion has been met, the neuron applies the activation function and do its calculations.
- The next neuron down may have weighted synapses of Distance to the city and, Bedrooms.
- This way the neurons work and interact in a very flexible way allowing it to look for specific things and therefore make a comprehensive search for whatever it is trained for.

6. IMPLEMENTATION & RESULT

This chapter describes how the thesis was practically conducted. It begins with explaining the structure and main process. There were a few methods that were used to reach the conclusion. One of them was a literature study that was done to choose the machine learning methods. Data was collected from two different sources; it was combined and used in the performed tests to find the most accurate method.

Then the obtained test-results were analyzed and compared. A prototype was also made in order to apply the obtained information into an environment. The chapter ends with information about the tools that were used.

6.1 DATA COLLECTION

(Twitter Data Collection)

We'll create functions to collect:

- **Tweets:** this also includes retweets, and replies collected as Tweet objects.
- **Followers:** all follower information collected as User objects.
- **Following:** information of all accounts I'm following *(a.k.a. friends)* collected as User objects.
- **Today's Stats:** the followers and following count that day.
- Also, we'll create two helper functions to make our job easier
- **Save JSON:** to save the collected data in a json file on Google Drive
- **Rate Limit Handling:** to manage the Twitter API limits that come with the free version, mainly the number of API calls permitted in a 15-minute period.

Now, let’s breakdown what’s going on in the while-loop.

There are three variables in play: all tweets is a list to store all the collected tweets, new_tweets is a list to store the latest batch of collected tweets since we can only retrieve 200 tweets at a time, and oldest stores the ID of the oldest tweet we retrieved so far, so the next batch of retrieved tweets come before it.

The variables are initialized before the loop starts. Note that if the specified user doesn’t have any tweets,new_tweets will be empty, and the loop won’t execute.

In each iteration, a new list of 200 tweets that were posted before oldest is retrieved and added to all tweets. The while-loop will keep iterating until no tweets are found before oldest or the limit of 3,200 tweets is reached.

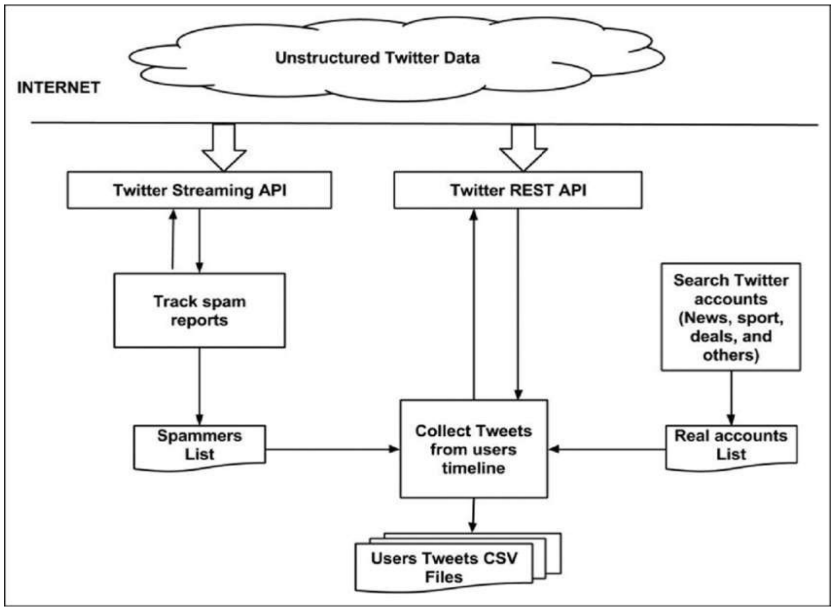


Fig.6.1 Data Collection architecture

6.2 DATA PREPROCESSING

6.2.1 Text Cleaning

Most text data will likely need some processing in order for the chosen machine learning algorithm to perform well. In this case each text document is a tweet and therefore will contain lots of characters that will not be meaningful to any machine learning algorithm. You can see below from just viewing the first few rows of the data that the tweets contain characters such as #, @ and punctuation marks.

In order to remove these I am using the Python re library, this provides regular expression matching operations. The following function successfully cleans up most of these characters. Additionally this function makes everything lower case.

This is generally a good idea as many text classification tools rely on counting the occurrences of words. If both upper and lower case versions of the same word are found in the text then the algorithm will count them as different words even though the meaning is the same. Of course this does mean that where the capitalized versions of a word exists, that does have a different meaning. For example the company Apple vs the fruit apple. This could result in poorer performance for some data sets. This is one area of NLP where you may try different methods to see how they affect the overall performance of the model.

The following work flow is what I was taught to use and like using, but the steps are just general suggestions to get you started. Usually I have to modify and/or expand depending on the text format.

6.2.2 Removing HTML

It is a step I did not do this time, however, if data is coming from a web scrape, it is a good idea to start with that. This is the function I would have used.Pretty much every step going forward includes creating a function and then applying it to a series. Be prepared, lambda functions will very shortly be your new best friend! You could also build a function to do all of these in one go, but I wanted to show the break down and make them easier to customize.

6.2.3 Remove punctuation

One way of doing this is by looping through the Series with list comprehension and keeping everything that is not in string.punctuation, a list of all punctuation we imported at the beginning with import string.

6.2.4 Tokenize

This breaks up the strings into a list of words or pieces based on a specified pattern using Regular Expressions aka Reg Ex. The pattern I chose to use this time (r'\w') also removes punctuation and is a better option for this data in particular. We can also add.lower() in the lambda function to make everything lowercase.

Some other examples of Reg Ex are:

‘\w+|\$[\d\.]+|\S+’ = splits up by spaces or by periods that are not attached to a digit
‘\s+’, gaps=True = grabs everything except spaces as a token
‘[A-Z]\w+’ = only words that begin with a capital letter.

6.2.5 Remove stop words

We imported a list of the most frequently used words from the NL Toolkit at the beginning with from nltk.corpus import stopwords. You can run stopwords.word(insert language) to get a full list for every language.

There are 179 English words, including ‘i’, ‘me’, ‘my’, ‘myself’, ‘we’, ‘you’, ‘he’, ‘his’, for example. We usually want to remove these because they have low predictive power. There are occasions when you may want to keep them though. Such as, if your corpus is very small and removing stop words would decrease the total number of words by a large percent.

6.2.6 Stemming & Lemmatizing

Both tools shorten words back to their root form. Stemming is a little more aggressive. It cuts off prefixes and/or endings of words based on common ones. It can sometimes be helpful, but not always because often times the new word is so much a root that it loses its actual meaning. Lemmatizing, on the other hand, maps common words into one base. Unlike stemming though, it always still returns a proper word that can be found in the dictionary. I like to compare the two to see which one works better for what I need. I usually prefer Lemmatizer, but surprisingly, this time, Stemming seemed to have more of an affect.
You see more of a difference with Stemmer so I will keep that one in place. Since this is the final step, I

added " ".join() to the function to join the lists of words back together.

6.3. DATA ARCHITECTURE

As discussed earlier, the data we have is imbalanced. We have a total of 31935 rows out of which the hate tweets comprise only 7% of the total tweets. Modelling on an imbalanced dataset is not ideal since the model won't be able to learn what pieces of text or information causes the tweet to be classified as a hate tweet.

If we train a model on imbalanced data, the results will be misleading. In such datasets, due to lack of learning, the model would simply predict each tweet as a good tweet. In this case, in spite of having a high accuracy, it is not actually doing a good job of classification since it is unable to classify the hate tweets accurately.

In simpler words, even when the model predicts all tweets as non-hate, it will still be accurate 93 percent of the time.

Therefore, we perform strategic sampling and separate the data into a temporary set and test set. Note that since we have performed strategic sampling, the ratio of good tweets to hate tweets is 93:7 for both the temporary and test datasets.

On the temporary data, we first tried to perform up sampling of hate tweets using SMOTE (Synthetic Minority Oversampling Technique). Since the SMOTE packages don't work directly for textual data, we wrote our own code for it.

The process is as follows:

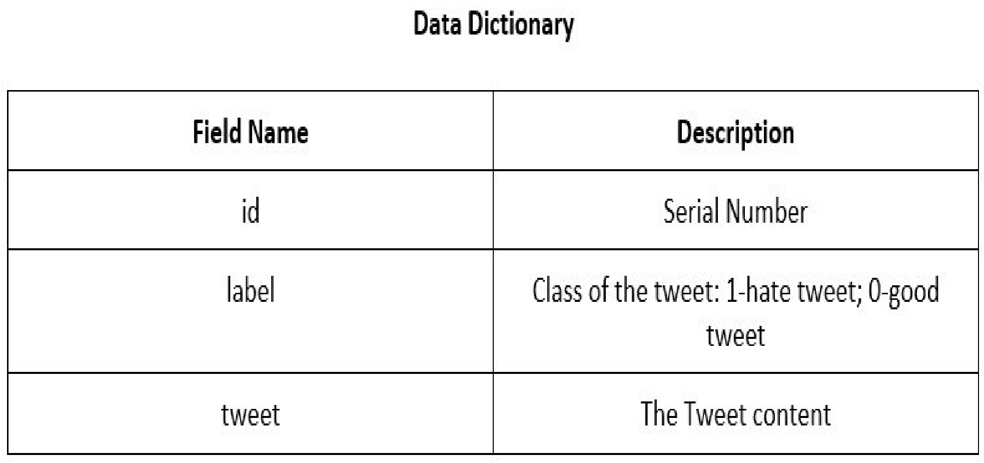
We created a corpus of all the unique words present in hate tweets of the temporary data set. Once we had a matrix containing all possible words in hate tweets, we created a blank new data set and started filling it with new hate tweets. These new tweets were synthesized by selecting words at random from the corpus. The lengths of these new tweets were determined on the basis of the lengths of the tweets from which the corpus was formed.

We then repeated this process multiple times until the number of hate tweets in this synthetic data was equal to the number of non-hate tweets we had in our temporary data.

However, when we employed the Bag of Words approach for feature generation, the number of features went up to 100,000. Due to an extremely high number of features, we faced hardware and processing power limitation and hence had to discard the SMOTE oversampling method.

As it was not possible to up-sample hate tweets to balance the data, we decided to down-sample non-hate tweets to make it even. We took a subset of only the non-hate tweets from the temporary data set. From this subset, we selected n random tweets, where n is the number of hate tweets in the temporary data. We then joined this with the subset of hate tweets in the temporary data. This data set is now the training data that we use for our feature generation and modelling purposes.

The test data is still in a 93:7 ratio of good tweets to hate tweets as we did not perform any sampling on it. Sampling was not performed as real world data comes in this ratio.



Data Dictionary

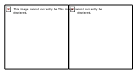
Field Name	Description
id	Serial Number
label	Class of the tweet: 1-hate tweet; 0-good tweet
tweet	The Tweet content

Fig. 6.2 Data Dictionary

6.4 DATA VISUALIZATION :

These are some visualizations from our data:

We got different insights from the collected data using matplotlib and seaborn libraries. Using them formed univariate and bivariate analysis to get more understanding of data , in this way we have formed bar plots of categories of tweets and formed other plots seperately for positive tweets as well as for negative tweets and get to know the skewness of data to handle the imbalance in data.



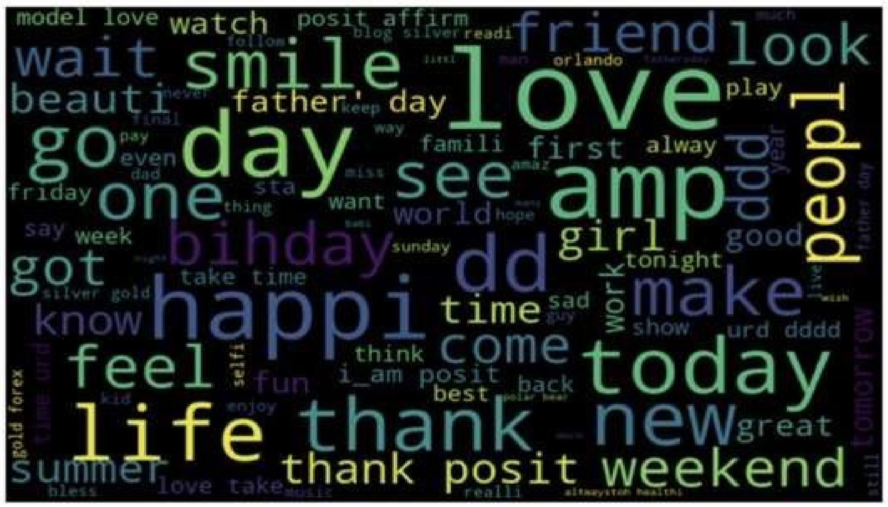


Fig. 6.3 Word cloud for positive tweets

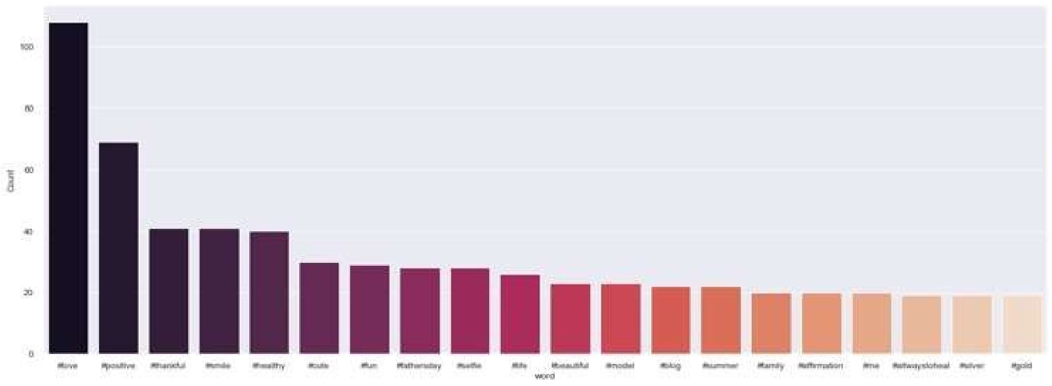


Fig.6.4 Bar plot positive tweets

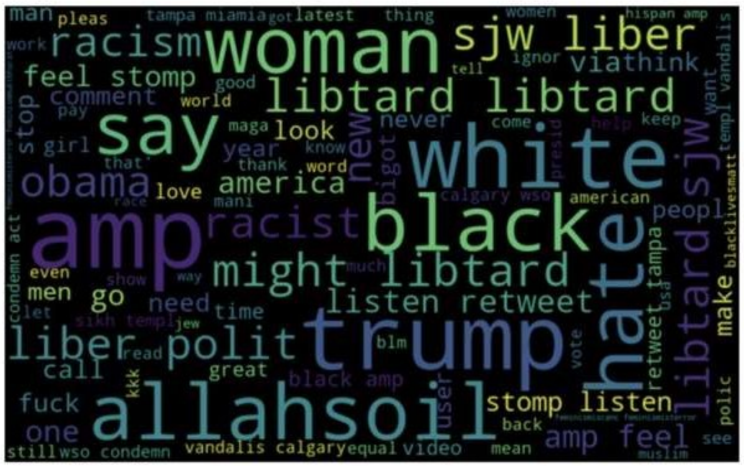


Fig.6.5. Word cloud for negative tweets

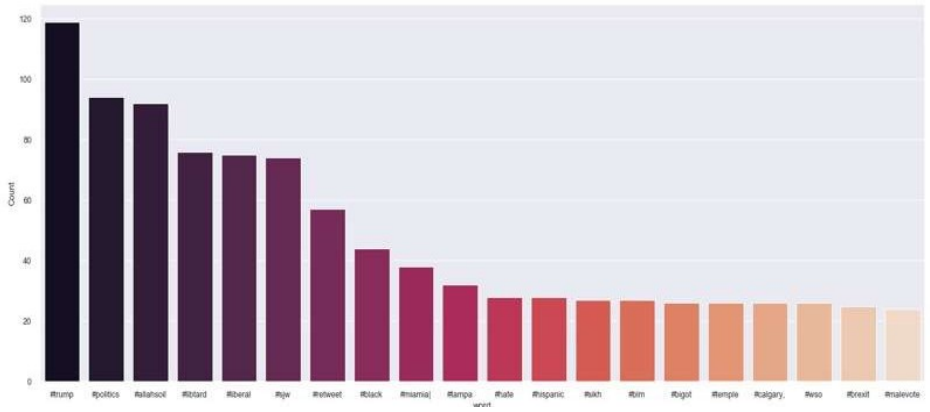


Fig. 6.6 Bar plot negative tweets

6.5 DATA MODELING

For both the Bag of words and TFIDF, we run 5 classification algorithms, namely Logistic Regression, Naive Bayes, Decision Tree, Random Forest and Gradient Boosting. Furthermore, we also perform dimensionality reduction in both the approaches and again run these 5 algorithms.

6.5.1. Bag of Words

We implement the Bag of Words for feature generation, on our training data set. In this approach, we try to find the probability that a tweet is a hate tweet or not, given the presence of certain words and their frequency in the tweet.

Our training data is a total of 4480 tweets which generates a corpus of 12566 unique words. After implementing the Bag of Words feature engineering, the result is a data frame containing 4480 rows and 12566 columns, where each column corresponds to a unique word. We then concatenate the label as a column.

We now debate over the reasoning of having accuracy and recall as our metrics and how they'll help us choose our champion model. At a first glance, gradient boosting seems to be the best model due to its highest accuracy. But as discussed in data architecture, our test data contains a 93:7 ratio for good tweets to hate tweets.

Thus, having high accuracy may not be the best metric. We hence take into consideration recall, which is the proportion of hate tweets we are able to correctly identify.

If this metric is high, it would correspond to a high number of hate tweets being classified as hate tweets and thus achieving our objective. If we look exclusively at recall, Naive Bayes seems to be the best model. However, another factor to be considered is that all the tweets we capture as hate tweets will be forwarded to an operations team which would review each of these tweets individually.

Therefore, having a large number of false positives will mean that the analysts will have to unnecessarily scrutinize a higher number of tweets which are actually non hate tweets. This will lead to inefficiencies in the utilization of time, resources and personnel.

Our recommendation is to have a high recall while at the same time ensuring that the number of false positives are kept to a minimum. It would be ideal to move forward with models which are giving high scores for both accuracy and recall. Thus, we select logistic regression and random forest as the champion models.

6.5.2. Frequency Inverse Document Frequency (TFIDF)

There are certain words which are considered as slang or abusive words, but are so common that people use them a lot in their daily lives. So, if there is a tweet containing these words, it may not actually be intended as a hate tweet.

To draw the line between whether a slang/abusive word is intended as hate or not, we tried the TFIDF approach since it allocates a low weightage to such words. So, instead of the Bag of Words model where we have a count of how many times that word occurred in the tweet, this will have a TFIDF weight instead. After running the algorithms, we obtain the following results.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Fig. 6.7
TF-IDF

6.6 TRAINING AND TESTING

Machine learning is about learning some properties of a data set and then testing those properties against another data set. A common practice in machine learning is to evaluate an algorithm by splitting a data set into two. We call one of those sets the **training set**, on which we learn some properties; we call the other set the **testing set**, on which we test the learned properties.

6.6.1 LOADING AN EXAMPLE DATASET

Scikit-learn comes with a few standard data sets, for instance the iris and digits data sets for classification and the boston house prices data set for regression.

In the following, we start a Python interpreter from our shell and then load the iris and digits data sets. Our notational convention is that $ denotes the shell prompt while >>> denotes the Python interpreter prompt:

```
$ python
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
```

A dataset is a dictionary-like object that holds all the data and some metadata about the data. This data is stored in the .data member, which is a n_samples, n_features array. In the case of supervised problem, one or more response variables are stored in the .target member. More details on the different datasets can be found in the dedicated section.

For instance, in the case of the digits dataset, digits.data gives access to the features that can be used to classify the digits samples:

```
>>> print(digits.data)

[[ 0.   0.   5. ...   0.   0.   0.]
 [ 0.   0.   0. ...  10.   0.   0.]
 [ 0.   0.   0. ...  16.   9.   0.]
 ...
 [ 0.   0.   1. ...   6.   0.   0.]
 [ 0.   0.   2. ...  12.   0.   0.]
 [ 0.   0.  10. ...  12.   1.   0.]]
```

and digits.target gives the ground truth for the digit dataset, that is the number corresponding to each digit image that we are trying to learn:

```
>>>
```

```
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
```

Shape of the data arrays

The data is always a 2D array, shape (n_samples, n_features), although the original data may have had a different shape. In the case of the digits, each original sample is an image of shape (8, 8) and can be accessed using:

```
>>> digits.images[0]
array([[ 0., 0., 5., 13., 9., 1., 0.,  0.],
       [  0.,   0.,  13.,  15.,  10., 15., 5.,  0.],
       [  0.,   3.,  15.,   2.,   0., 11., 8.,  0.],
       [  0.,   4.,  12.,   0.,   0., 8., 8.,  0.],
       [  0.,   5.,   8.,   0.,   0., 9., 8.,  0.],
       [  0.,   4.,  11.,   0.,   1., 12., 7.,  0.],
       [  0.,   2.,  14.,   5.,  10., 12., 0.,  0.],
       [  0.,   0.,   6.,  13.,  10., 0., 0.,  0.]])
```

The simple example on this dataset illustrates how starting from the original problem one can shape the data for consumption in scikit-learn.

6.6.2 LEARNING AND PREDICTION

In the case of the digits dataset, the task is to predict, given an image, which digit it represents. We are given samples of each of the 10 possible classes (the digits zero through nine) on which we *fit* an estimator to be able to *predict* the classes to which unseen samples belong.

In scikit-learn, an estimator for classification is a Python object that implements the methods fit(X, y) and predict(T).

An example of an estimator is the class sklearn.svm.SVC, which implements support vector classification. The estimator's constructor takes as arguments the model's parameters.

For now, we will consider the estimator as a black box:

```
>>>
>>> from sklearn import svm
>>> clf = svm.SVC(gamma=0.001, C=100.)
```

Choosing the parameters of the model

In this example, we set the value of gamma manually. To find good values for these parameters, we can use tools such as grid search and cross validation.

The clf (for classifier) estimator instance is first fitted to the model; that is, it must *learn* from the model. This is done by passing our training set to the fit method. For the training set, we'll use all the images from our dataset, except for the last image, which we'll reserve for our predicting. We select the training set with the [:-1] Python syntax, which produces a new array that contains all but the last item from digits.data:

```
>>> clf.fit(digits.data[:-1], digits.target[:-1])
SVC(C=100.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma=0.001, kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

Now you can *predict* new values. In this case, you'll predict using the last image from digits.data. By predicting, you'll determine the image from the training set that best matches the last image.

```
>>>
```

```
>>> clf.predict(digits.data[-1:])
array([8])
```

The corresponding image is:

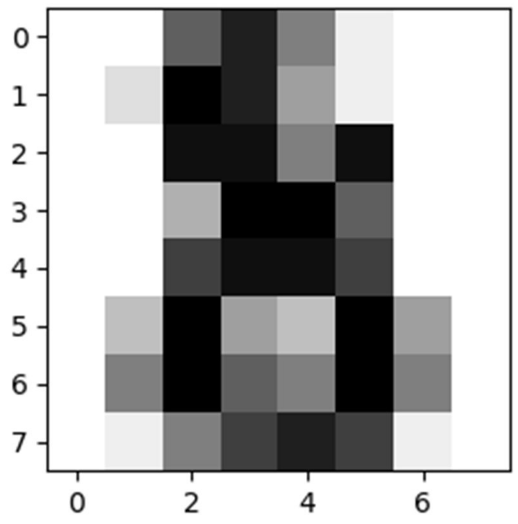
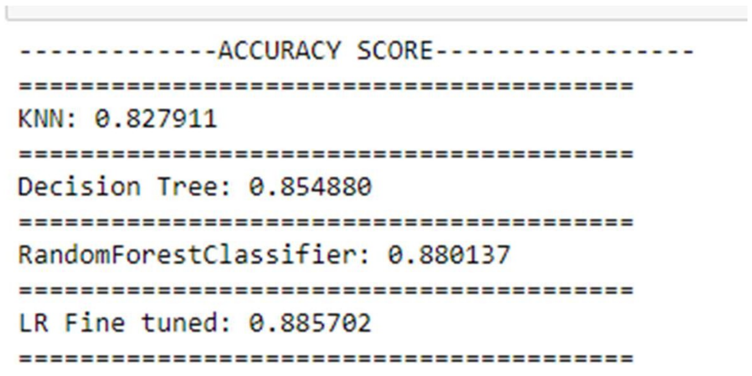


Fig. 6.8 Accuracy levels

6.7. RESULTS

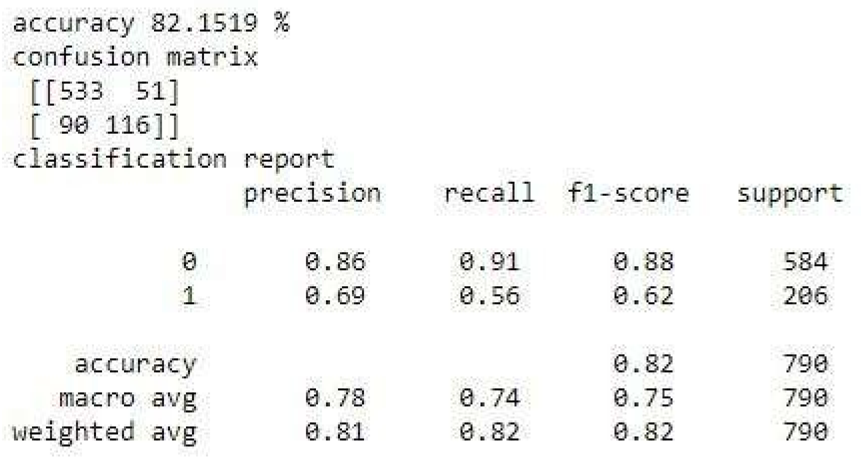
This chapter presents the information that was obtained from the performed tests. It begins with individual tests performed on each machine learning method in order to find the settings that give the most accurate result. We test Logistic regression fine tuned and also Random forest. Thereafter the neural network was tested with various settings.

These are the accuracy levels we have achieved using shallow machine learning algorithms.Here we have done with K nearest neighbors , Decision Tree , Random Forest , and also using fine tuned logistics regression , i.e. We have done hyper parameter tuning for logistic regression using Grid search CV and fiitted it using 5 fits.



```
-------------ACCURACY SCORE----------------
========================================
KNN: 0.827911
========================================
Decision Tree: 0.854880
========================================
RandomForestClassifier: 0.880137
========================================
LR Fine tuned: 0.885702
========================================
```

And this is the classification report of------ algorithm since it is performing in a decent manner on our validation set.We have also formed Confusion matrix to get know how many are mis classified.



```
accuracy 82.1519 %
confusion matrix
 [[533  51]
 [ 90 116]]
classification report
               precision    recall  f1-score   support

           0       0.86      0.91      0.88       584
           1       0.69      0.56      0.62       206

    accuracy                           0.82       790
   macro avg       0.78      0.74      0.75       790
weighted avg       0.81      0.82      0.82       790
```

Fig . 6.9 Classification report

We have also represented confusion matrix for Fine tuned Logistic regression since it is also having a decent performance level.

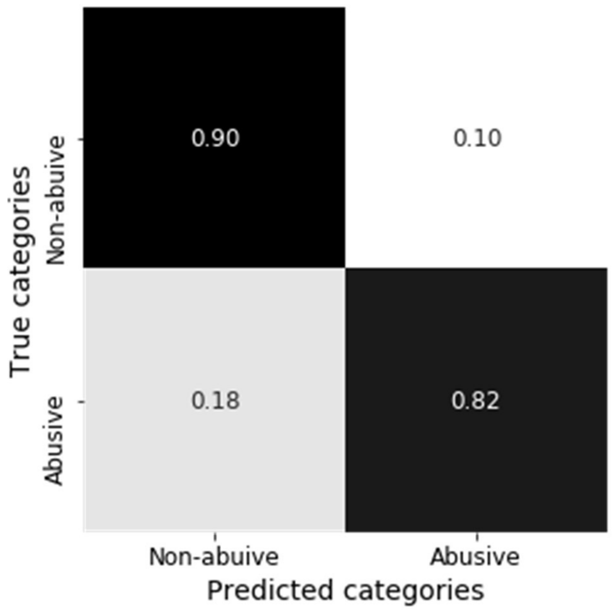
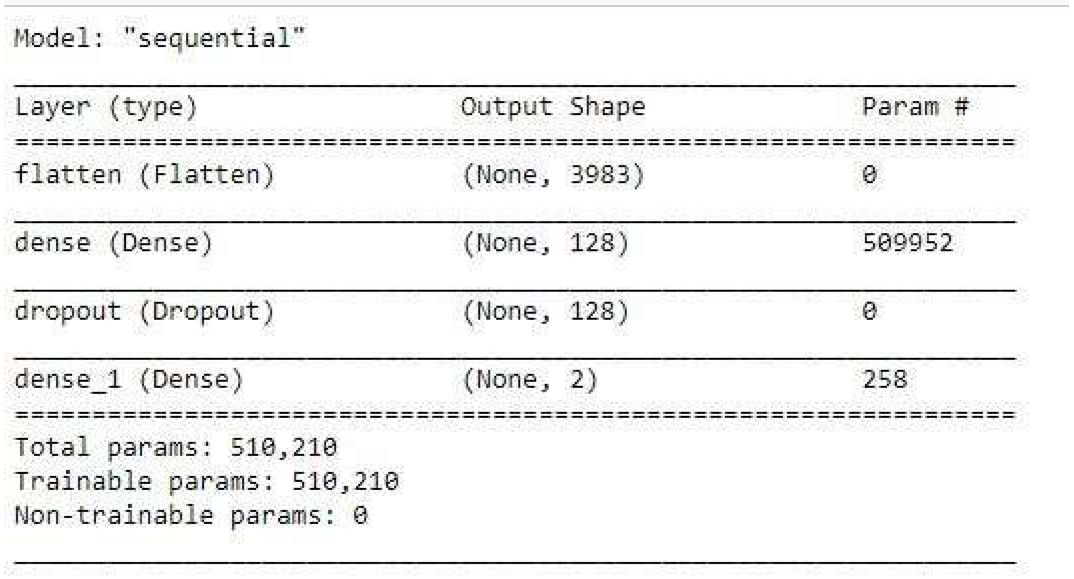


Fig . 6.10 Confusion matrix

Here we have formed a neural network model of multi layer perceptron using keras sequential libraries.Model has been formed using these flatten , drop out layers.Here we can see the summary:

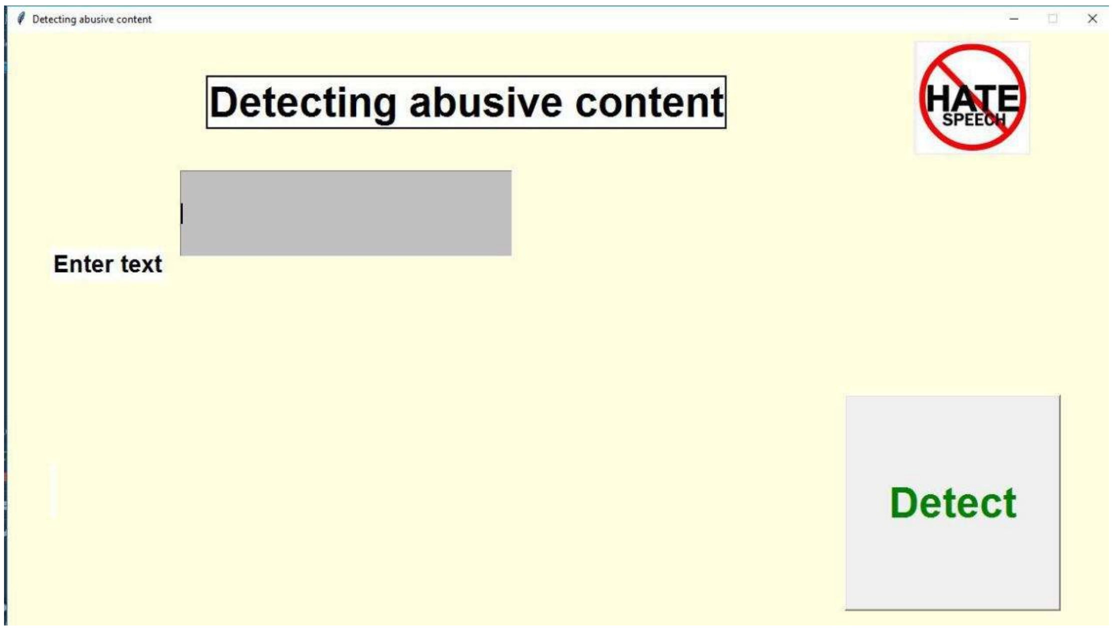


```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 3983)              0
_________________________________________________________________
dense (Dense)                (None, 128)               509952
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 258
=================================================================
Total params: 510,210
Trainable params: 510,210
Non-trainable params: 0
_________________________________________________________________
```

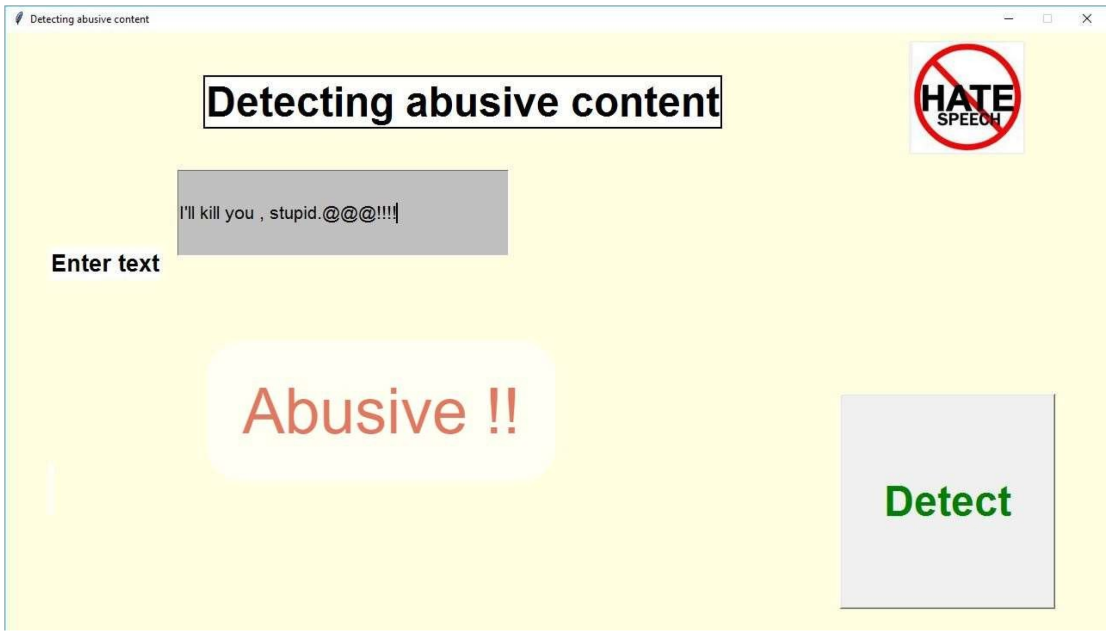
Fig . 6.11 Neural network model

OUT PUT SCREENS:

This is the basic output window , where we need to enter any text here , and model detects whether the given content is abusive or not.

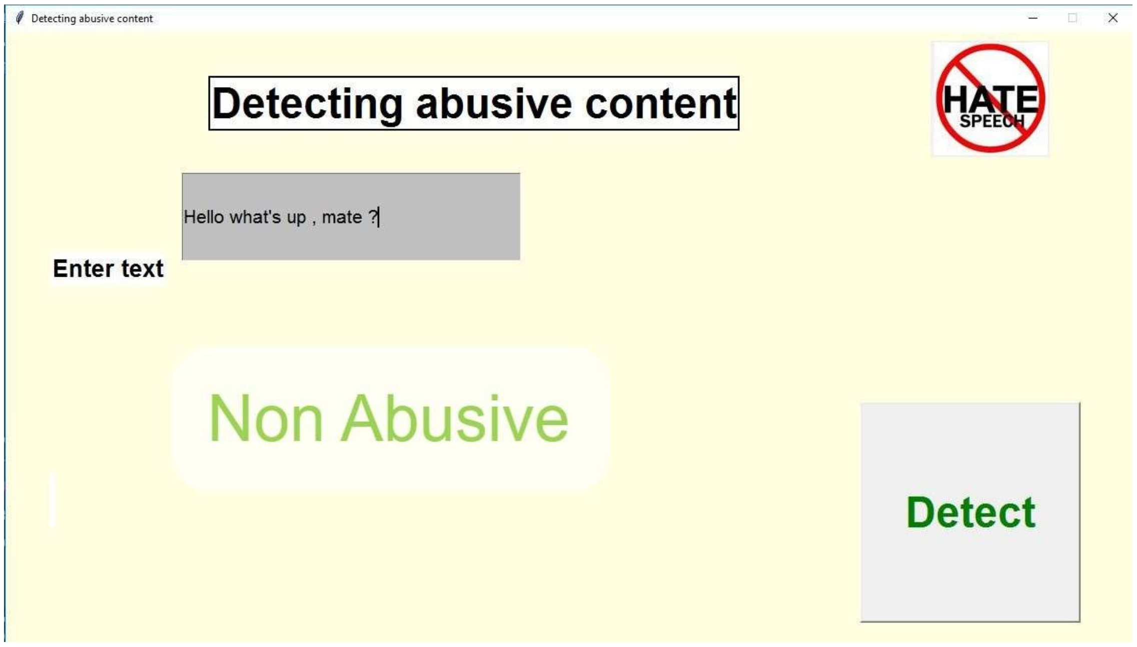


Here we have given some text , and we can see the model has detected it as abusive content, since it has learned from the data which we have feeded it to the model during training.



I'll kill you , stupid.@@@!!!!

This is an output result of non abusive content , that the model has detected as:



Hello what's up , mate ?

Non Abusive

In this way we can give any kind and form of text which it can take , before giving it for model for predictions , our text will go through all pre processing phases.

7. TESTING & VALIDATION

7.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 DESIGN AND TEST CASE SCENARIOS

Functional and Non-Functional Testing

7.2.1 Functional Testing

L TESTING is a type of software testing whereby the system is tested against the functional requirements/specifications. Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. It simulates actual system usage but does not make any system structure assumptions.

Functional testing involves the following steps:

- Identify functions that the software is expected to perform.
- Create input data based on the function’s specifications.
- Determine the output based on the function’s specifications.
- Execute the test case.
- Compare the actual and expected outputs

Functional testing is more effective when the test conditions are created directly from user/business requirements. When test conditions are created from the system documentation (system requirements/ design documents), the defects in that documentation will not be detected through testing and this may be the cause of end-users wrath when they finally use the software.

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.

A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.It finds problems early in the development cycle. This includes both bugs in the programmer's implementation and flaws or missing parts of the specification for the unit. The cost of finding a bug before coding begins or when the code is first written is considerably lower than the cost of detecting, identifying, and correcting the bug later.

Bugs in released code may also cause costly problems for the end-users of the software. Code can be impossible or difficult to unit test if poorly written, thus unit testing can force developers to structure functions and objects in better ways. Allows the programmer to re factor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing).

Integration testing

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

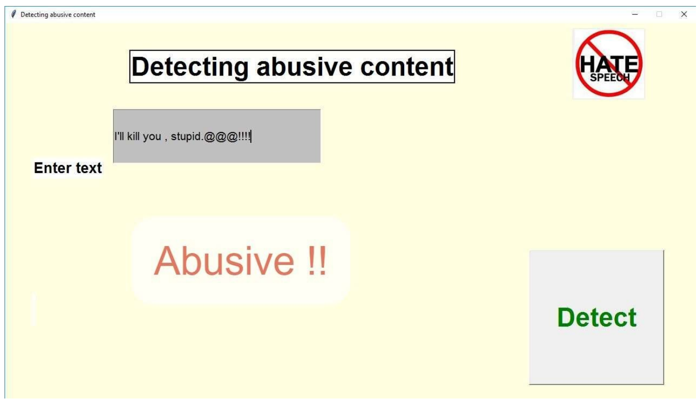
It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.Some different types of integration testing are big-bang, mixed (sandwich), risky hardest, top-down, and bottom-up. Other Integration Patterns are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration

7.2.2 Non Functional testing

Non-functional testing is defined as a type of Software testing to check non functional aspects (performance, usability, reliability, etc.) of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing. An excellent example of non-functional test would be to check how many people can simultaneously login into software.

Non-functional testing is equally important as functional testing and affects client satisfaction.Objectives of Non-functional testing Non-functional testing should increase usability, efficiency, maintainability, and portability of the product. Helps to reduce production risk and cost associated with non-functional aspects of the product.Optimize the way product is installed, setup, executes, managed and monitored. Collect and produce measurements, and metrics for internal research and development. Improve and enhance knowledge of the product behavior and technologies in use.

7.2.3 TESTING SCENARIOS



I'll kill you , stupid.@@@!!!

Abusive !!

Fig. 7.1 Testing abusive content

8. CONCLUSION

In this we analyzed a machine learning algorithm for automatic identification of abusive as well as insulting comments which indicates that more advancement in digital media may make require repercussion for utilizing current innovations.Natural language processing helps us to get trending innovations in improvement of impact of recent innovation. In this project, we assess a lot of classification calculations on two sorts of client created online substance in two English. The various arrangements of information we deal with were grouped towards aspects, for example, prejudice, sexism, hate speech, animosity and individual assaults. While recognizing issues with between annotator understanding for grouping errands utilizing these names, the focal point of this project is on arranging the information as per the commented on qualities utilizing a few Text Classification Algorithms finally detecting which one is more suitable for the problem.

8.1 FUTURE SCOPE

The same technology should be included in video to get audio abusive or in proper content to be removed from audio or video or it may get skip.Depending on the abusing content we can retrive the person information from database to take necessary action.Deep learning algorithms can be used for further details and highly accurate analysis.

9. REFERENCES

[1] Agrawal, S., & Awekar, A. (2018). Deep learning for detecting cyberbullying across multiple social media platforms. In ECIR:Advances in Information Retrieval (pp. 141–153). https://doi.org/10.1007/978-3-319-76941-7_11

[2]. Allen, C. (2011). Islamophobia. Surrey: Ashgate. Amichai-hamburger,Y., & McKenna, K. (2006). The Contact Hypothesis Reconsidered: Interacting via the Internet. Journal of Computer-Mediated Communication, 11(1), 825–843. https://doi.org/10.1111/j.1083-6101.2006.00037.

[3]x Anzovino, M., Fersini, E., & Rosso, P. (2018). Automatic identification and classification of misogynistic language on Twitter. In NLDB (pp. 57–64). https://doi.org/10.1007/978-3-319-91947-8_6

[4] Badjatiya, P., Gupta, M., & Varma, V. (2019). Stereotypical Bias Removal for Hate Speech Detection Task using Knowledgebased Generalizations. In World Wide Web (pp. 49–59).

[5] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep Learning for Hate Speech Detection in Tweets. In World Wide
Web (pp. 759–760). https://doi.org/10.1145/3041021.3054223 Benesch, S. (2012). Dangerous Speech: A Proposal to Prevent Group Violence. New York.

[6] Binns, R., Veale, M., Van Kleek, M., & Shadbolt, N. (2017). Like trainer, like bot? Inheritance of bias in algorithmic content moderation. In Lecture Notes in Computer Science (pp. 1–12). https://doi.org/10.1007/978-3-319-67256-4_32

[7] Buchanan, E. (2017). Considering the ethics of big data research: A case of Twitter and ISIS / ISIL. PLoS ONE, 12(12), 1–6.

[8] Bucy, E., & Holbert, L. (2013). Sourcebook for political communication research. London: Routledge. Burnap, P., & Williams, M. (2016). Us and Them: Identifying Cyber Hate on Twitter across
Multiple Protected Characteristics. EPJ Data Science, 5(1), 1–15.
https://doi.org/10.1140/epjds/s13688-016-0072-6

[9] Caiani, M., & Wagemann, C. (2009). Online networks of the Italian and German Extreme Right. Information, Communication & Society, 66–109. https://doi.org/10.1080/13691180802158482

[10] Chandrasekharan, E., Samory, M., Srinivasan, A., & Gilbert, E. (2017). The Bag of Communities. In CHI (pp. 3175–3187).
https://doi.org/10.1145/3025453.3026018

[11] Crawford, K., & Gillespie, T. (2016). What is a flag for? Social media reporting tools and the vocabulary of complaint. New Media and Society, 18(3), 410–428.
https://doi.org/10.1177/1461444814543163

[12] Daniels, J. (2013). Race and racism in Internet Studies: A review and critique. New Media and Society, 15(5), 695–719.
https://doi.org/10.1177/1461444812462849

[13] Davidson, T., Warmsley, D., Macy, M.,& Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. In ICWSM (pp. 1–4).

[14] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805v2, 1–16. Retrieved from
http://arxiv.org/abs/1810.04805

[15] Dixon, L., Li, J., Sorensen, J., Thain, N., & Vasserman, L. (2018). Measuring and Mitigating Unintended Bias in Text Classification. In AAAI/ACM Conference on AI, Ethics, and Society (pp. 67–73).
https://doi.org/10.1145/3278721.

Classification for Detecting Insulting and Abusive Content

B.Srinu, Assistant Professor Dept. of Computer Science & Engineering from Vignan Institute of Technology and Science.
M.Swetha, student Dept. of Computer Science & Engineering from Vignan Institute of Technology and Science.
K.Dheeraj Pranav, student Dept. of Computer Science & Engineering from Vignan Institute of Technology and Science.
K.Mahesh Reddy, student Dept. of Computer Science & Engineering from Vignan Institute of Technology and Science.

Abstract

The sheer simplicity with which abusive and insulting comments can be made on the web – normally from the solace of your home and the absence of any quick negative repercussions – utilizing the present computerized correspondence innovations (particularly web based life), is liable for their noteworthy increment and worldwide universality. Natural Language processing advancements can help in tending to the negative impacts of this improvement. In this project, we assess a lot of classification calculations on two sorts of client created online substance in two English. The various arrangements of information we deal with were grouped towards aspects, for example, prejudice, sexism, hate speech, animosity and individual assaults. While recognizing issues with between annotator understanding for grouping errands utilizing these names, the focal point of this project is on arranging the information as per the commented on qualities utilizing a few Text Classification Algorithms finally detecting which one is more suitable for the problem.

Keywords: Abusive and Insulting Comments, NLP (Natural Language Processing), Text Classification Algorithm.

I. Introduction

Its common phenomena of commenting abusive content, sometimes verbal aggression as well as hateful content. Statements given for debate or comments on it may hamper seriously for private or public arranged debates. Today's improvement takes care of all the hateful utterances present in digital communication or on government as well as private organizations. There is many stakeholders took an increasing attention for this type of content. Many debates that may be public or social takes place online nowadays which indicates that our social scientists must analyze the content which is present at online.

Today's social media marketing also includes some abusive content which can't be tolerated by the civil rights, politicians as well as some perception of the content. Recently online news channels anc communication Medias are giving unpredictable information for economic, political as well as social information.

For avoiding the surveillance issues as well as censorship misuses for online content as well as online media the technological infrastructure is designed which indicates that the information should address the importance of the address and not the irrelevant data which is not much useful for the online submission and which causes the extra efforts for the people to get the main content. Today's challenge is to separate out the main and high quality information from the abusive, hateful and offensive information.

The manual fact checking by professionals as well as journalism may create burden to check and cure information for online users. The end user may get more correct and satisfactory information so the necessary algorithm must be designed to solve it. So the proposed work is designed to solve the all issues present in manual or existing techniques.

II. Literature Survey

There are many algorithms which are specially designed for detection of abusive or insulting content with different software applications. There is no clarity in their uses under different scenarios hence there is need of comparative study for discussion of such algorithms which can solve many the problem of finding abusive and insulting content from data. Older and existing algorithms need much amount of data to train the classifier as well as one of the major drawback is in what scenario which algorithm we can use there is no proper clarity.

III. Proposed Work

In proposed work we designed system of deep learning algorithm which uses python software for implementing classification of abusing as well as insulting content. Further these all algorithms are compared to understand the best suitable algorithms among all. This algorithms which are machine learning algorithms provides more accurate and reliable results than existing state of art techniques. This algorithm is more reliable because with less training data set also it provides higher accuracy.

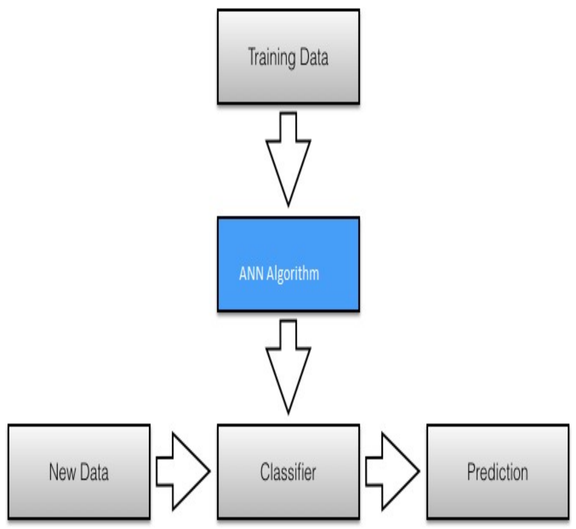


Fig.1 System architecture for proposed work

Above figure represents the system architecture for proposed work. The classifier used is deep learning based classifier. It shows the systematic approach for proposed work.

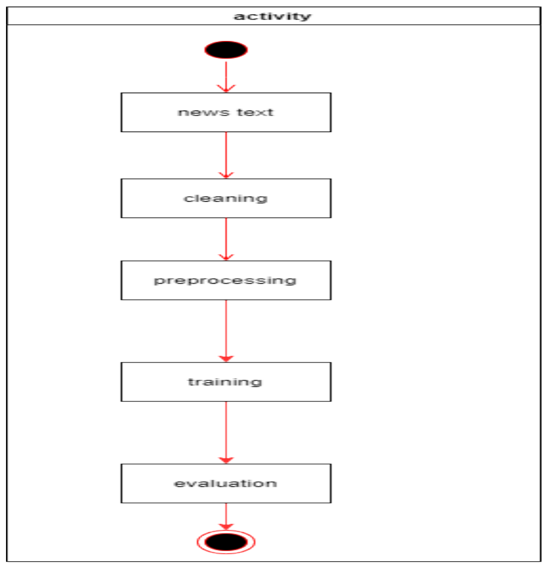


Fig.2 Activity diagram for proposed work

Above figure represents the activity diagram for detecting an abusive and insulting comment identification which helps to make it automatic.

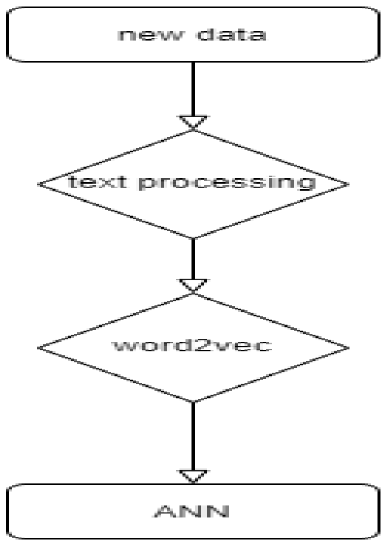


Fig.3 Flow chart of proposed work

Above figure represents the flow wise representation of proposed work. It indicates that the step included in proposed work may have major 4 steps as shown above in flowchart.

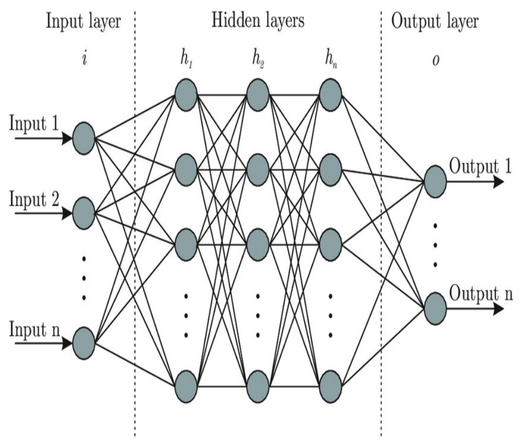


Fig.4 Algorithms Architecture for neural network

This represents the neural network which has major three layers input layer, hidden layer and output layer.

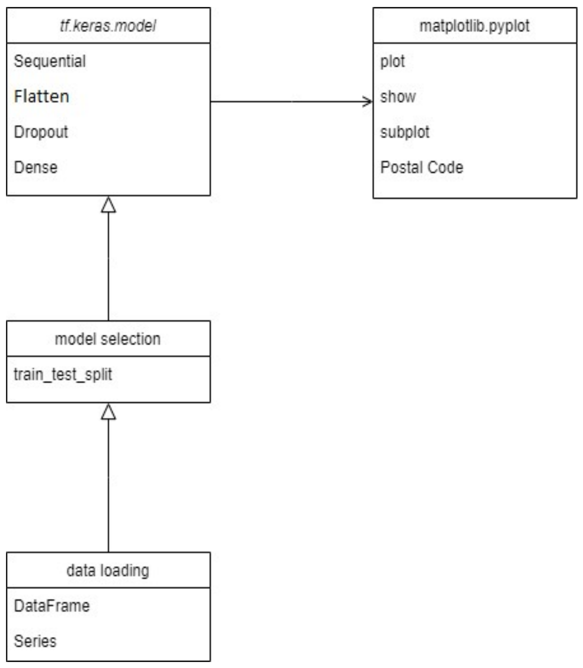


Fig.Class Diagram for Proposed Work

In class diagram also there are 4 major steps which displayed in above.

IV. Result and Analysis

Datasets

Placeholders may work for given simple examples, but tf.data is important step in streaming. To run the program tf.Tensor available at dataset should convert into tf.data.Iterator, and go through the Iterator's tf.data.Iterator.get_next method.

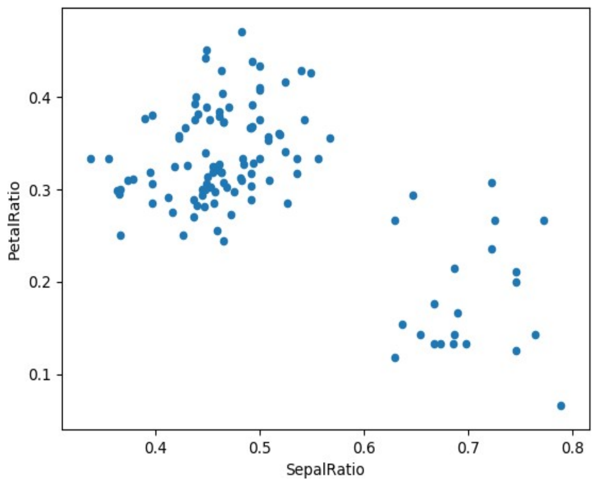


Fig.SepalRatio vs PetalRatio representation of given data

Training Set and Testing Set

Machine learning concept includes learning some information or we can say properties from dataset and testing properties of another data can be classified. Mostly we are splitting data in two categories as training set and testing set. Training set is the set of learning properties from given data. Testing set is to check properties of the data given for query which may include some different properties. Depending on Training set with different classifier we may get different results with different accuracy. The classifier of type deep learning which is having highest classification accuracy can be considered for further analysis.

V. Conclusion

In this we analyzed a machine learning algorithm for automatic identification of abusive as well as insulting comments which indicates that more advancement in digital media may make require repercussion for utilizing current innovations.Natural language processing helps us to get trending innovations in improvement of impact of recent innovation. In this project, we assess a lot of classification calculations on two sorts of client created online substance in two English. The various arrangements of information we deal with were grouped towards aspects, for example, prejudice, sexism, hate speech, animosity and individual assaults. While recognizing issues with between annotator understanding for grouping errands utilizing these names, the focal point of this project is on arranging the information as per the commented on qualities utilizing a few Text Classification Algorithms finally detecting which one is more suitable for the problem.

Future Scope

The same technology should be included in video to get audio abusive or inproper content to be removed from audio or video or it may get skip.Depending on the abusing content we can retrive the person information from database to take necessary action.Deep learning algorithms can be used for further details and highly accurate analysis.

References

[1] Agrawal, S., & Awekar, A. (2018). Deep learning for detecting cyberbullying across multiple social media platforms. In ECIR: Advances in Information Retrieval (pp. 141–153). https://doi.org/10.1007/978-3- 319-76941-7_11

[2]. Allen, C. (2011). Islamophobia. Surrey: Ashgate. Amichai-hamburger, Y., & McKenna, K. (2006). The Contact Hypothesis Reconsidered: Interacting via the Internet. Journal of Computer-Mediated Communication, 11(1), 825–843. https://doi.org/10.1111/j.1083-6101.2006.00037.

[3]x Anzovino, M., Fersini, E., & Rosso, P. (2018). Automatic identification and classification of misogynistic language on Twitter. In NLDB (pp. 57–64). https://doi.org/10.1007/978-3-319-91947-8_6

[4] Badjatiya, P., Gupta, M., & Varma, V. (2019). Stereotypical Bias Removal for Hate Speech Detection Task using Knowledgebased Generalizations. In World Wide Web (pp. 49–59).

[5] Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep Learning for Hate Speech Detection in Tweets. In World Wide Web (pp. 759–760). https://doi.org/10.1145/3041021.3054223 Benesch, S. (2012). Dangerous Speech: A Proposal to Prevent Group Violence. New York.

[6] Binns, R., Veale, M., Van Kleek, M., & Shadbolt, N. (2017). Like trainer, like bot? Inheritance of bias in algorithmic content moderation. In Lecture Notes in Computer

Science (pp. 1–12). https://doi.org/10.1007/978-3-319-67256-4_32

[7] Buchanan, E. (2017). Considering the ethics of big data research: A case of Twitter and ISIS / ISIL. PLoS ONE, 12(12), 1–6.

[8] Bucy, E., & Holbert, L. (2013). Sourcebook for politicalcommunication research. London: Routledge. Burnap, P., & Williams, M. (2016). Us and Them: Identifying Cyber Hate on Twitter across Multiple Protected Characteristics. EPJ Data Science, 5(1), 1–15. https://doi.org/10.1140/epjds/s13688-016-0072-6

[9] Caiani, M., & Wagemann, C. (2009). Online networks of the Italian and German Extreme Right. Information, Communication & Society, 66–109. https://doi.org/10.1080/13691180802158482

[10] Chandrasekharan, E., Samory, M., Srinivasan, A., & Gilbert, E. (2017). The Bag of Communities. In CHI (pp. 3175–3187). https://doi.org/10.1145/3025453.3026018

[11] Crawford, K., & Gillespie, T. (2016). What is a flag for? Social media reporting tools and the vocabulary of complaint. New Media and Society, 18(3), 410–428. https://doi.org/10.1177/1461444814543163

[12] Daniels, J. (2013). Race and racism in Internet Studies: A review and critique. New Media and Society, 15(5), 695–719. https://doi.org/10.1177/1461444812462849

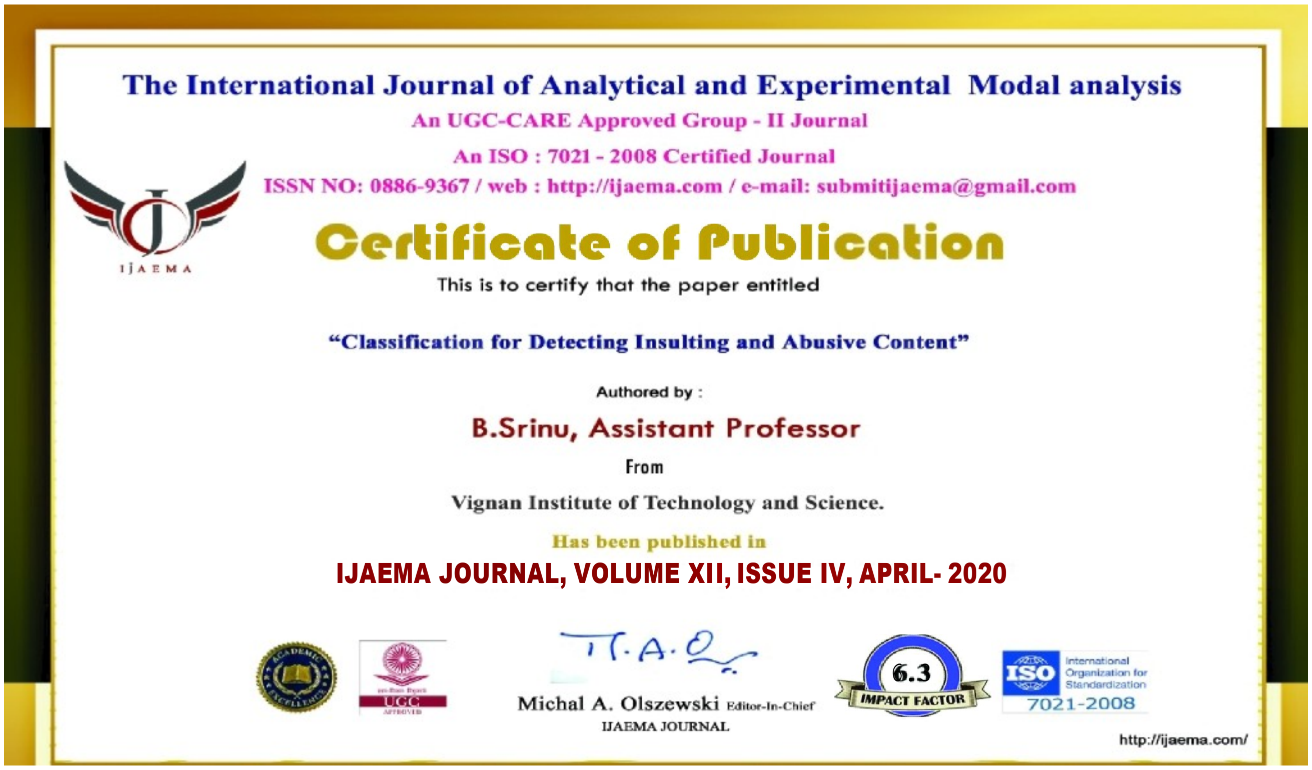
[13] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. In ICWSM (pp. 1–4).

[14] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805v2, 1–16. Retrieved from http://arxiv.org/abs/1810.04805

[15] Dixon, L., Li, J., Sorensen, J., Thain, N., & Vasserman, L. (2018). Measuring and Mitigating Unintended Bias in Text Classification. In AAAI/ACM Conference on AI, Ethics, and Society (pp. 67–73). https://doi.org/10.1145/3278721.3278729

[16] Eatwell, R. (2006). Community Cohesion and Cumulative Extremism in Contemporary Britain. Political Quarterly, 77(2), 204– 216. https://doi.org/10.1111/j.1467-923X.2006.00763.

[17] x Eger, S., Şahin, G. G., Rücklé, A., Lee, J.-U., Schulz, C., Mesgar, M., … Gurevych, I. (2019). Text Processing Like Humans Do: Visually Attacking and Shielding NLP Systems. ArXiv:1903.11508v1, 1–14.



The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

IJAEMA

Certificate of Publication

This is to certify that the paper entitled

"Classification for Detecting Insulting and Abusive Content"

Authored by :

B.Srinu, Assistant Professor

From

Vignan Institute of Technology and Science.

Has been published in

IJAEMA JOURNAL, VOLUME XII, ISSUE IV, APRIL- 2020

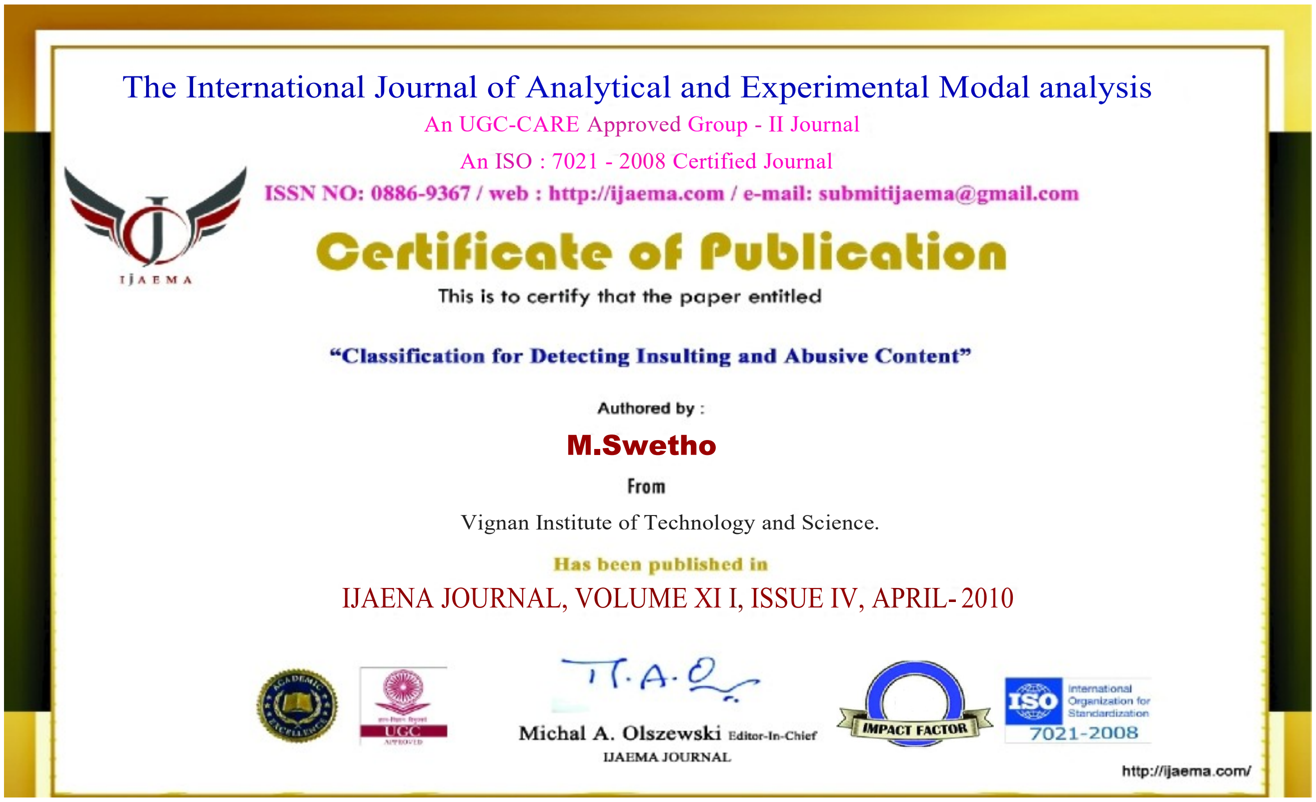
Michal A. Olszewski Editor-In-Chief

IJAEMA JOURNAL

6.3 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

http://ijaema.com/



The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

IJAEMA

Certificate of Publication

This is to certify that the paper entitled

"Classification for Detecting Insulting and Abusive Content"

Authored by :

M.Swetho

From

Vignan Institute of Technology and Science.

Has been published in

IJAENA JOURNAL, VOLUME XI I, ISSUE IV, APRIL- 2010

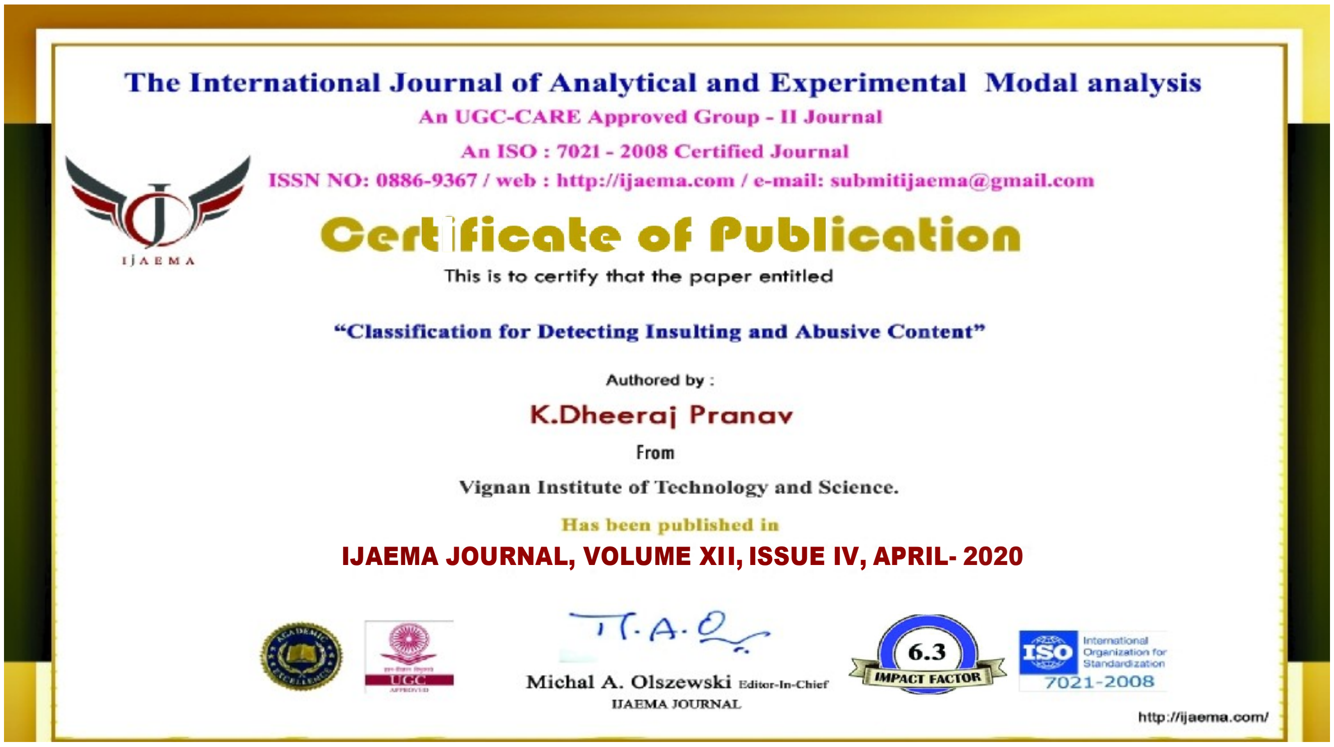
Michal A. Olszewski Editor-In-Chief

IJAEMA JOURNAL

IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

http://ijaema.com/


The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

IJAEMA

Certificate of Publication

This is to certify that the paper entitled

"Classification for Detecting Insulting and Abusive Content"

Authored by :

K.Dheeraj Pranav

From

Vignan Institute of Technology and Science.

Has been published in

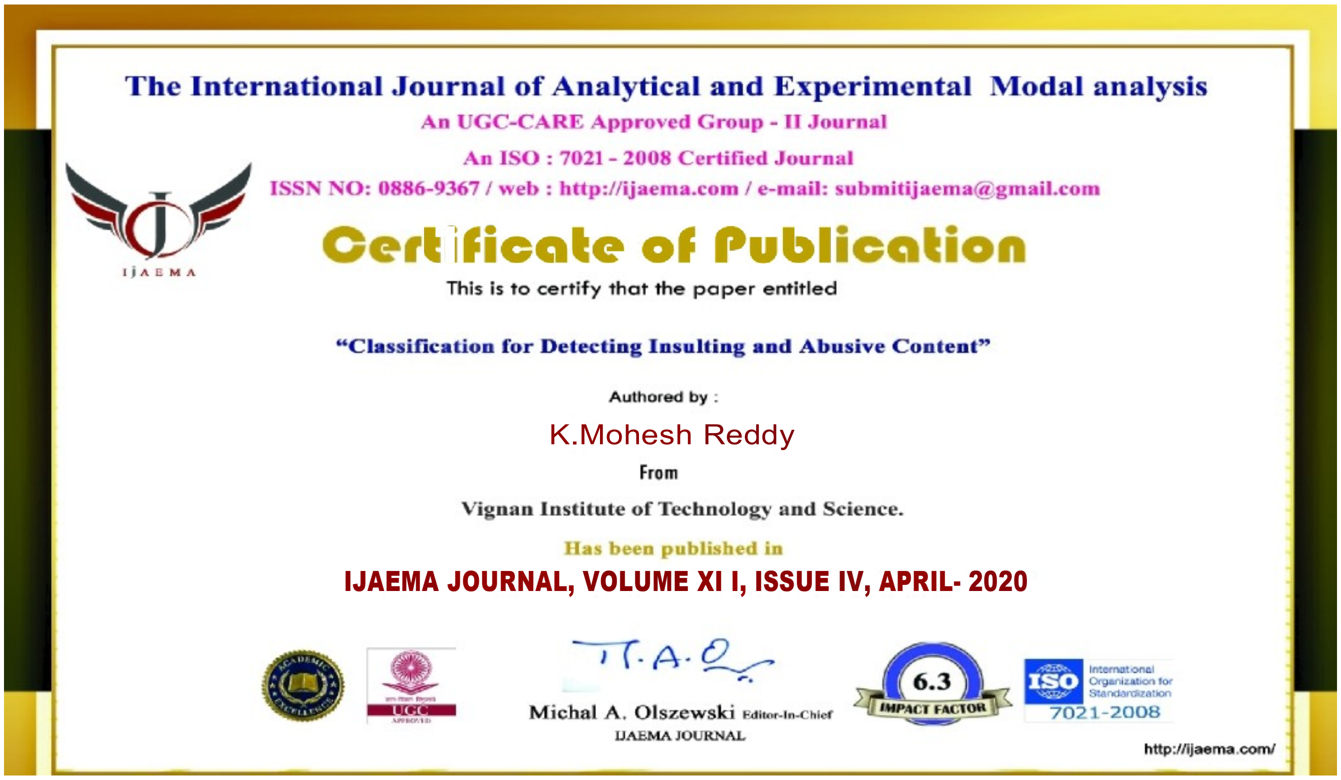
IJAEMA JOURNAL, VOLUME XII, ISSUE IV, APRIL- 2020

Michal A. Olszewski Editor-In-Chief
IJAEMA JOURNAL

6.3 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

http://ijaema.com/


The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

IJAEMA

Certificate of Publication

This is to certify that the paper entitled

"Classification for Detecting Insulting and Abusive Content"

Authored by :

K.Mohesh Reddy

From

Vignan Institute of Technology and Science.

Has been published in

IJAEMA JOURNAL, VOLUME XI I, ISSUE IV, APRIL- 2020

Michal A. Olszewski Editor-In-Chief
IJAEMA JOURNAL

6.3 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

http://ijaema.com/