TOXIC COMMENTS CLASSIFICATION USING MACHINE LEARNING

A PROJECT REPORT

Submitted By

ASHWIN.P	**211614104026**
GANESH.R	**211614104046**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2018

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"TOXIC COMMENTS CLASSIFICATION USING MACHINE LEARNING"** is the bonafide work of **ASHWIN.P [Regd. No. 211614104026] and GANESH.R [Regd. No. 211614104046]** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr. P. Kumar **Professor and Head** Department of Computer Science and Engineering Rajalakshmi Engineering College Chennai – 602 105	**Ms. J. Silviya Nancy** **Supervisor** Department of Computer Science and Engineering Rajalakshmi Engineering College, Chennai - 602 105

Submitted to Project and Viva Examination held on ______________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I extend my gratitude to my Chairman **Mr. S. MEGANATHAN** and Chairperson **Dr. (Mrs.).THANGAM MEGANATHAN** for providing me with all the necessary resources and other facilities towards completion of this project. I am extremely grateful to my principal **Mr. S.N.MURUGESAN** for giving me a valuable support and encouragement throughout the duration of this course.

I wish to thank **Dr. P.KUMAR**, Head of the Department, Department of Computer Science and Engineering, Rajalakshmi Engineering College, for extending all facilities to me to work on this project.

I would like to express my sincere appreciation and gratitude to my guide **Ms. J. SILVIYA NANCY**, Assistant Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College for her guidance, constant encouragement, and support. Her meticulous attention and creative thinking have been a source of inspiration for me throughout this project.

I also extend my sincere thanks to **all faculty members** and supporting staffs for their direct and indirect involvement in successful completion of the project. All endeavors over a long period can be successful only with the advice and support of many well-wishers. I take this opportunity to express my gratitude and appreciation to all of them. Above all, I express my heartfelt thanks to my parents and family members who have dedicated their life to my well-being.

ABSTRACT

Nowadays, the conversations in different styles and expressing one's opinion on certain affairs online has become a major trend. Currently, the debates and group conferences in social media is more prevalent than face-to-face interactions. However, at certain times, it is difficult to discuss about things that we care about. Increase of threats, abuses and harassment in online conversations has stopped many people from expressing themselves and they give upon seeking different opinions due to the fear of being offended and cornered. Current platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments. With the advancements in the field of Machine Learning, the subset of Artificial Intelligence offers platform to possibly classify the comments based on its level of toxicity based on its nature. In this project, we propose an application that classifies the comments based on its nature of toxicity. Here the learning model is trained with datasets containing toxic comments which would help the user in needful. And when such comments are used, the prototype identifies and highlights it. Logistic Regression is the data classifier which is used in the prototype and feature extraction is done using sklearn TfidfVectorizer module . The module is trained using Supervised learning methodology where training data is collected and formed a dataset .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine Learning
SVM	Support Vector Machines
MLE	Maximum Likelihood Estimation
RNN	Recurrent Neural Network
LSTM	Long-Short Term Memory Cell
CNN	Convolutional Neural Network
DBSCAN	Density-based spatial clustering of applications with noise
NPM	Node Package Manager
HTML	Hypertext Markup Language
CSV	Comma separated values
JS	Java Script
UI	User Interface

CHAPTER 1

INTRODUCTION

1.1 Artificial Intelligence

Artificial intelligence is a branch of computer science that aims to create intelligent agents/machines. It has become an essential part of the technology industry. Research associated with artificial intelligence is highly technical and specialized. The core problems of artificial intelligence include programming computers for certain traits such as Knowledge, Reasoning, Problem-solving Learning, etc. Knowledge engineering is a core part of AI research. Machines can often act and react like humans only if they have abundant information relating to the world. Artificial intelligence must have access to objects, categories, properties and relations between all of them to implement knowledge engineering. Initiating common sense, reasoning and problem-solving power in machines is a difficult and tedious approach.

Machine learning is another core part of AI which we will look into in the next section. Learning without any supervision requires an ability to identify patterns in streams of inputs, whereas learning with adequate supervision involves classification and numerical regressions. Classification determines the category an object belongs to and regression deals with obtaining a set of numerical input or output examples, thereby discovering functions enabling the generation of suitable outputs from respective inputs.

Machine perception deals with the capability to use sensory inputs to deduce the different aspects of the world, while computer vision is the power to analyze visual inputs with a few sub-problems such as facial, object and gesture recognition. Robotics is also a major field related to AI. Robots require intelligence to handle tasks such as object manipulation and navigation, along with sub-problems of localization, motion planning, and mapping. All together AI is showing up in every part of life, anywhere from driving, to the cell phones we use, how our data is managed in the world, how our homes are going to be built in the future. So given its ubiquity, it is important to start addressing the strengths and limitations of artificial intelligence.

1.2 Machine Learning

Machine Learning is a branch of AI which enables a machine to learn models without being explicitly programmed every time a problem is given. Machine learning focuses on the development of computer programs that can change by itself when the machine is exposed to new data or problem. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they can independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. It's a science that's not new – but one that's gaining fresh momentum.

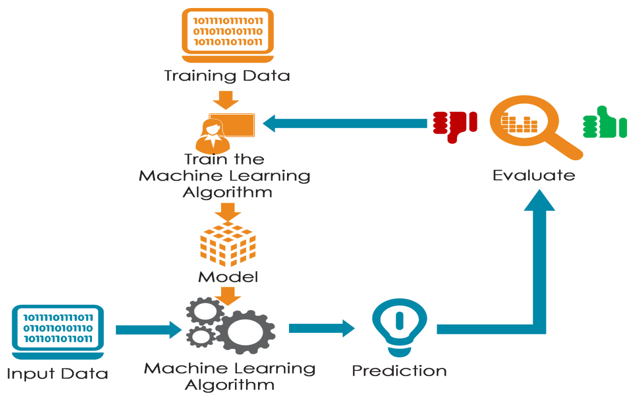

Training Data
Train the Machine Learning Algorithm
Evaluate
Model
Input Data
Machine Learning Algorithm
Prediction

Figure 1.1 Machine Learning Workflow

Machine Learning is broadly classified into two main types:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Semi-supervised Learning

1.2.1 Supervised Learning

Supervised Machine Learning models are the ones who learn and makes predictions based on evidence in the presence of uncertainty. It uses a known dataset (called the training dataset) to make predictions. The training dataset includes input data and response values. From it, the supervised learning algorithm seeks to build a model that can make predictions of the response values for a new dataset. The new data set is usually provided by the user as an Input for which the machine has to give the desired output. A test dataset is often used to validate the model. Using larger training data sets often yield models with higher predictive power that can generalize well for new datasets.

Supervised learning involves two different sets of algorithms. They are Classification, Regression. The most common classification Algorithms include support vector machine, Neural networks, Naïve Bayesian classifier, decision trees, nearest neighbors, etc. on the other hand, the Regression technique includes algorithm such as Linear Regression, Non-Linear Regression, Generalized Linear models, decision trees, etc. Supervised learning is used in financial applications for credit scoring, algorithmic trading, and bond classification; in biological applications for tumor detection and drug discovery; in energy applications for the price and load forecasting; and in pattern recognition applications for speech and images. In our case, we use Supervised Learning to train our machine against known Image Datasets such as Flickr 8K, Flickr30K, and MSCOCOso that a hypothesis is developed to recognize Object in an Image.

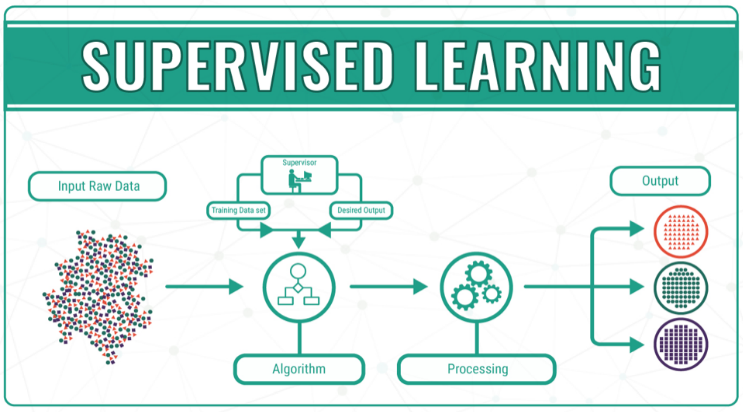


Figure 1.2 Supervised Learning Workflow

1.2.2 Unsupervised Learning

Unsupervised Machine Learning model is another extreme of Machine Learning, where the machine without any experience or data will try to predict the output for an Input. Unsupervised Learning algorithms when provided with Input data sets, will first try to classify and group them based on their similarity and functionality.

Unsupervised learning is important since it is likely to be much more common in the brain than supervised learning. For instance, there are around 106 photoreceptors in each eye whose activities are constantly changing with the visual world and which provide all the information that is available to indicate what objects there are in the world, how they are presented, what the lighting conditions are, etc. Developmental and adult plasticity are critical in animal vision – indeed structural and physiological properties of synapses in the neocortex are known to be substantially influenced by the patterns of activity in sensory neurons that occur. However, essentially none of the information about the contents of scenes is available during learning. This makes unsupervised methods essential, and, equally, allows them to be used as computational models for synaptic adaptation.

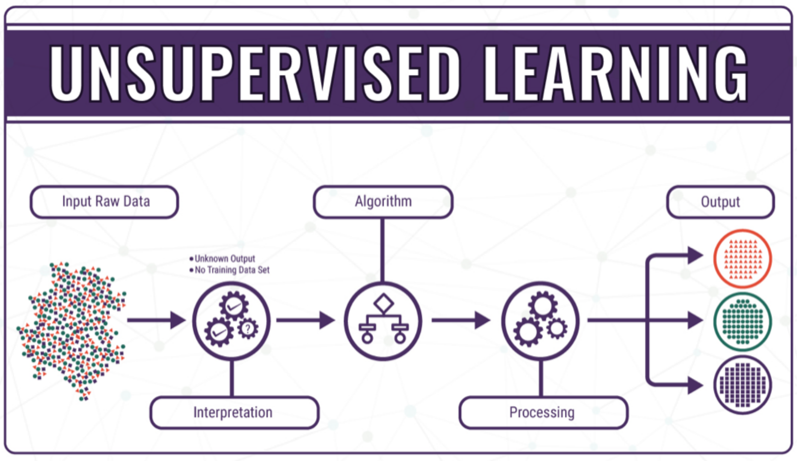


Figure 1.3 Unsupervised Learning Workflow

1.2.3 Reinforcement Learning

Reinforcement Learning is another part of Machine Learning that is gaining a lot of prestige in how it helps the machine learn from its progress. Readers who have studied psychology in college would be able to relate to this concept on a better level.

Reinforcement Learning spurs off from the concept of Unsupervised Learning, and gives a high sphere of control to software agents and machines to determine what the ideal behavior within a context can be. This link is formed to maximize the performance of the machine in a way that helps it to grow. Simple feedback that informs the machine about its progress is required here to help the machine learn its behavior.

Reinforcement Learning is not simple, and is tackled by a plethora of different algorithms. As a matter of fact, in Reinforcement Learning an agent decides the best action based on the current state of the results. The growth in Reinforcement Learning has led to the production of a wide variety of algorithms that help machines learn the outcome of what they are doing.

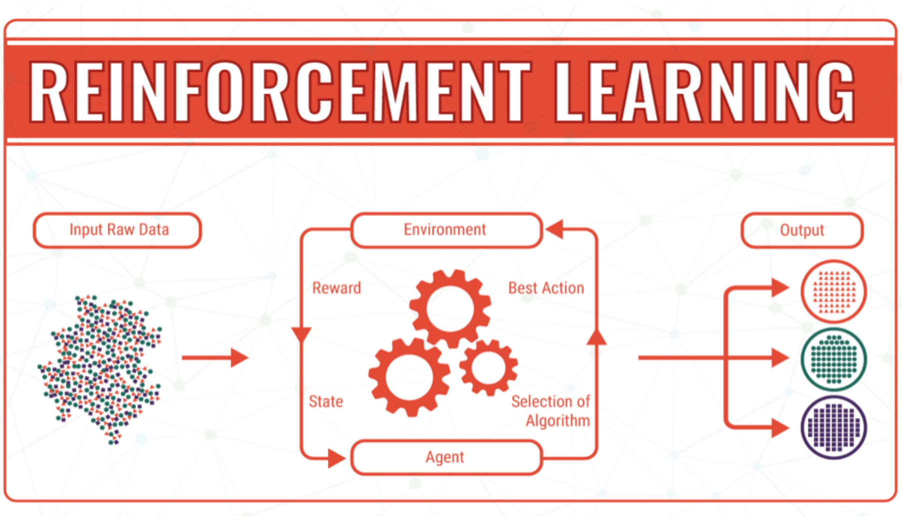


Figure 1.4 Reinforcement Learning Workflow

1.2.4 Semi-supervised Learning

Semi-supervised learning is a class of supervised learning tasks and techniques that also make use of unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy.The acquisition of labeled data for a learning problem often requires a skilled human agent (e.g. to transcribe an audio segment) or a physical experiment (e.g. determining the 3D structure of a protein or determining whether there is oil at a particular location). The cost associated with the labeling process thus may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value. Semi-supervised learning is also of theoretical interest in machine learning and as a model for human learning.

1.2.5 Machine Learning Algorithms

From an algorithmic point of view, there are wide varieties of Machine Learning algorithms which vary from one use case to other. It is required that we've to pick the ones which are suitable for our application. The most widely used Machine Learning Algorithms are **Decision Trees, Naive Bayes Classification, Linear Regression, Logistic Regression** and **Support Vector Machines.**

Decision Trees: A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Take a look at the image to get a sense of how it looks like. From a business decision point of view, a decision tree is the minimum number of yes/no questions that one has to ask, to assess the probability of making a correct decision, most of the time. As a method, it allows

to approach the problem in a structured and systematic way to arrive at a logical conclusion.

Naïve Bayes Classification: Naïve Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. The featured image is the equation — with P(A|B) is posterior probability, P(B|A) is a likelihood, P(A) is class prior probability, and P(B) is predictor prior probability.

Linear Regression If you know statistics, you probably have heard of linear regression before. Least square is a method for performing linear regression. You can think of linear regression as the task of fitting a straight line through a set of points. There are multiple possible strategies to do this, and "ordinary least squares" strategy go like this — You can draw a line, and then for each of the data points, measure the vertical distance between the point and the line, and add these up; the fitted line would be the one where this sum of distances is as small as possible.

Logistic Regression: Logistic regression is a powerful statistical way of modeling a binomial outcome with one or more explanatory variables. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

Support Vector Machines: SVM is binary classification algorithm. Given a set of points of 2 types in N-dimensional places, SVM generates an (N — 1) dimensional hyperplane to separate those points into two groups. Say you have some points of 2 types in a paper which are linearly separable. SVM will find a straight line which separates those points into two types and situated as far as possible from all those points.

1.3 Data Preprocessing

Machine learning algorithms learn from data. It is critical that you feed them the right data for the problem you want to solve. Even if you have good data, you need to make sure that it is in a useful scale, format and even that meaningful features are included.

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Therefore, certain steps are executed to convert the data into a small clean data set. This technique is performed before the execution of Iterative Analysis. The set of steps is known as Data Preprocessing. It includes Data Cleaning, Data Integration, Data Transformation and Data Reduction.

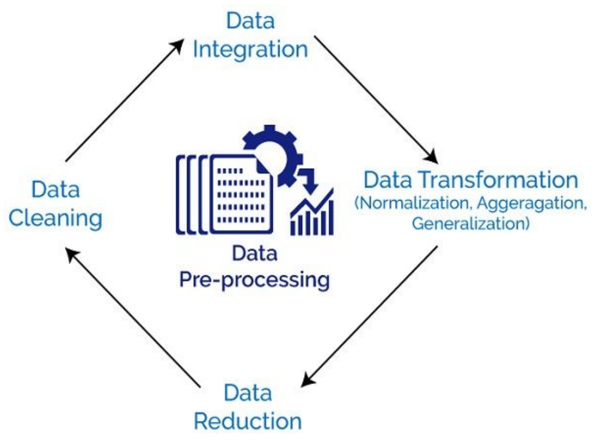

Data Integration
Data Transformation (Normalization, Aggeragation, Generalization)
Data Reduction
Data Cleaning
Data Pre-processing

Figure 1.5 Steps in Data Preprocessing

1.3.1 Need For Data Preprocessing

For achieving better results from the applied model in Machine Learning and Deep Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning and Deep Learning model need information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values has to be managed from the original raw data set.Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

Data Preprocessing is necessary because of the presence of unformatted real world data. Mostly real world data is composed of -

- **Inaccurate data (missing data)** - There are many reasons for missing data such as data is not continuously collected, a mistake in data entry, technical problems with biometrics and much more.
- **The presence of noisy data (erroneous data and outliers)** - The reasons for the existence of noisy data could be a technological problem of gadget that gathers data, a human mistake during data entry and much more.
- **Inconsistent data** - The presence of inconsistencies are due to the reasons such that existence of duplication within data, human data entry, containing mistakes in codes or names, i.e., violation of data constraints and much more.

1.3.2 Data Preparation Process

The more disciplined you are in your handling of data, the more consistent and better results you are like likely to achieve. The process for getting data ready for a machine learning algorithm can be summarized in three steps:

a) Select Data

b) Preprocess Data

c) Transform Data

We followed this process in a linear manner, but it is very likely to be iterative with many loops.

a) Select Data

This step is concerned with selecting the subset of all available data that you will be working with. There is always a strong desire for including all data that is available, that the maxim “more is better” will hold. This may or may not be true.

The data collected should have a clear picture of everything . The data which are not recorded or cannot be recorded can be used to derive or simulate this data. Exclude the data which are not needed and cannot be used for the use case .

b) Preprocess Data

After you have selected the data, you need to consider how you are going to use the data. This preprocessing step is about getting the selected data into a form that you can work.

Three common data preprocessing steps are formatting, cleaning and sampling:

- **Formatting**: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.
- **Cleaning**: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.

- **Sampling**: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

c) Transform Data

The final step is to transform the process data. The specific algorithm you are working with and the knowledge of the problem domain will influence this step and you will very likely have to revisit different transformations of your preprocessed data as you work on your problem.

Three common data transformations are scaling, attribute decompositions and attribute aggregations. This step is also referred to as feature engineering.

- **Scaling**: The preprocessed data may contain attributes with a mixtures of scales for various quantities such as dollars, kilograms and sales volume. Many machine learning methods like data attributes to have the same scale such as between 0 and 1 for the smallest and largest value for a given feature. Consider any feature scaling you may need to perform.
- **Decomposition**: There may be features that represent a complex concept that may be more useful to a machine learning method when split into the constituent parts. An example is a date that may have day and time components that in turn could be split out further. Perhaps only the hour of day is relevant to the problem being solved. consider what feature decompositions you can perform.
- **Aggregation**: There may be features that can be aggregated into a single feature that would be more meaningful to the problem you are trying to

solve. For example, there may be a data instances for each time a customer logged into a system that could be aggregated into a count for the number of logins allowing the additional instances to be discarded. Consider what type of feature aggregations could perform.

1.4 Anaconda Python Distribution

Anaconda is a freemium open source distribution of the Python and R programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment.

Conda , the Anaconda's own **package manager**, is used for updating anaconda and its packages.Conda is a **cross platform package and environment manager**. It provides installing, executing and updating different packages along with their dependencies.It helps in **switching between environments** in our local machine.

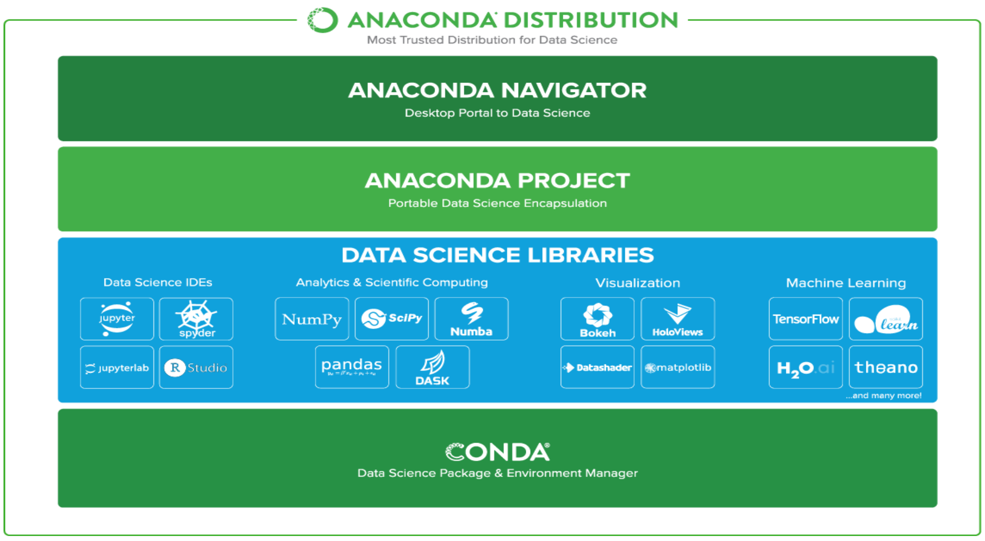


Figure 1.6 Python Anaconda Distribution Architecture

1.5 Anaconda Libraries

Anaconda python provides many libraries for several categories such as **Data Science , Analysis and Scientific Computing , Visualisation** and **Machine Learning .**

1.5.1 Scikit Learn

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines , random forests,gradient boost , k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

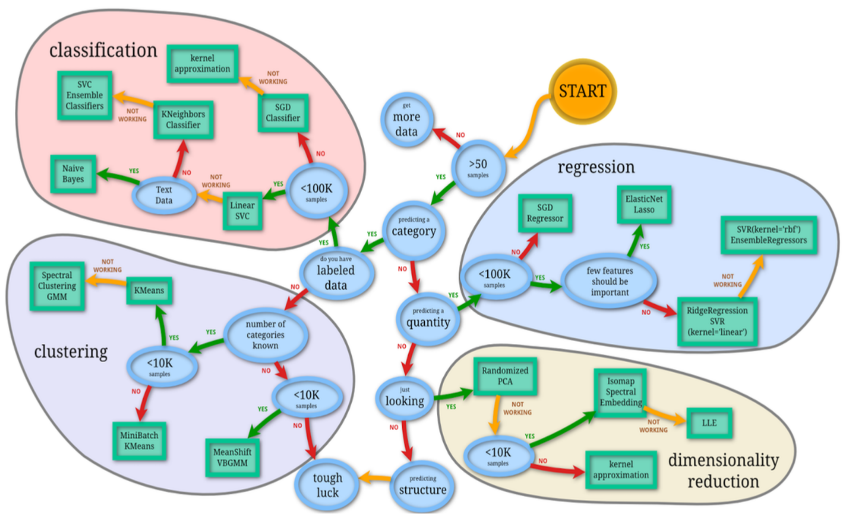


Figure 1.7 Scikit Libraries based on Data Scale

1.5.2 Pandas

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 licence.The name is derived from the term “panel data”, an econometrics term for data sets that include both time-series and cross-sectional data.

Features :

- DataFrame object 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.
- Label-based slicing, fancy indexing, and subsetting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
- The library is highly optimized for performance, with critical code paths written in Cython or C

1.5.3 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hard copy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

1.6 Feature Extraction

In machine learning , pattern recognition and in image processing , feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction.

When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vectors). Determining a subset of the initial features is called *feature selection*.The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

1.7 Logistic Regression

Logistic regression is generally used where the dependent variable is Binary or Dichotomous. That means the dependent variable can take only two possible values such as “Yes or No”, “Default or No Default”, “Living or Dead”, “Responder or Non Responder”, “Yes or No” etc. Independent factors or variables can be categorical or numerical variables.

Even though logistic (logit) regression is frequently used for binary variables (2 classes), it can be used for categorical dependent variables with more than 2 classes. In this case it’s called Multinomial Logistic Regression.The applications of Logistic Regression are as follows

a) Logistic Regression is used for prediction of output which is binary, as stated above. For example, if a credit card company is going to build a model to decide whether to issue a credit card to a customer or not, it will model for whether the customer is going to “Default” or “Not Default” on this credit card. This is called “Default Propensity Modeling” in banking lingo.

b) Similarly an ecommerce company that is sending out costly advertisement / promotional offer mails to customers, will like to know whether a particular customer is likely to respond to the offer or not. In Other words, whether a customer will be “Responder” or “Non Responder”. This is called “Propensity to Respond Modeling”

c) Using insights generated from the logistic regression output, companies may optimize their business strategies to achieve their business goals such as minimize expenses or losses, maximize return on investment (ROI) in marketing campaigns etc.

1.7.1 Underlying Algorithms and Assumptions

The underlying algorithm of Maximum Likelihood Estimation (MLE) determines the regression coefficient for the model that accurately predicts the probability of the binary dependent variable. The algorithm stops when the convergence criterion is met or maximum number of iterations are reached. Since the probability of any event lies between 0 and 1 (or 0% to 100%), when we plot the probability of dependent variable by independent factors, it will demonstrate an ‘S’ shape curve.

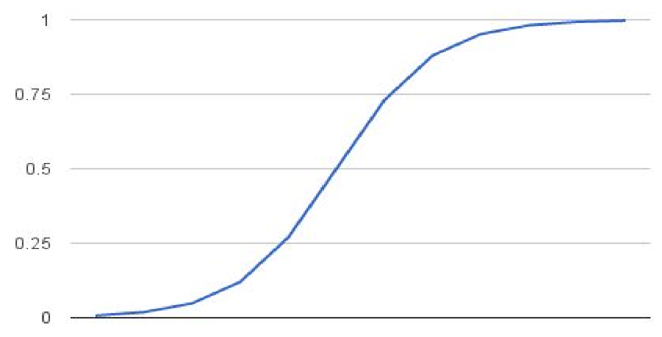


Figure 1.8 Dependent variable versus Independent variable

Logit Transformation is defined as follows-

Logit = Log (p/1-p) = log (probability of event happening/ probability of event not happening) = log (Odds)

1.8 Organisation of the Project

The organization of the project as depicted as follows: Chapter 2 is regarding the literature survey on the domain of the project , which was done to acquire the information of the existing systems and to get ideas about improvising it . Chapter 3 contains the detailed information of the proposed system architecture and it determines the workflow of the model . Chapter 4 deals with the process of implementation of the project and also conveys the results obtained. Chapter 5 is about testing the trained model by providing it with different inputs like non-toxic and toxic comments . Chapter 6 concludes the project with a conclusion note and suggesting some future enhancements that can be done to improvise the project .Chapter 7 has various references links which was used for references while working with the project .

CHAPTER 2

LITERATURE SURVEY

A literature survey determines the various analyses and research made in the field of the project and the results already published, taking into account the various parameters of the project and the extent of the project.

2.1 Survey on Toxic Comments Classification using Machine Learning

1) Comment Abuse Classification With Deep Learning

Authors: Theodora Chu , Kylie Jue , Max Wang

Description:

Social media platforms, online news commenting spaces, and many other public forum of the Internet have become increasingly known for issues of abusive behavior such as cyberbullying and personal attacks. However, determining whether or not a comment or post should be ”flagged” is difficult and time-consuming, and many platforms are still searching for more efficient moderation solutions. Automating the process of identifying abuse in comments would not only save websites time but would also increase user safety and improve discussions online. In this paper, we use Wikipedia talk page data in order to train deep learning models to flag comments. Comments are classified as personal attacks or not personal attacks, and ratings were determined by Crowdflower workers. We tested three models: a recurrent neural network (RNN) with a long-short term memory cell (LSTM) and word embeddings, a convolutional neural network (CNN) with word embeddings, and a CNN with character embeddings. Our models improve upon previous results from non-deep learning machine-learning models, and we find that a CNN with character-level embeddings reaches the highest performance.

2) Deceiving Google's Perspective API Built For Detecting Toxic Comments

Authors: Hossein Hosseini, Sreeram Kannan, Baosen Zhang and Radha Poovendran

Description:

Social media platforms provide an environment where people can freely engage in discussions. Unfortunately, they also enable several problems, such as online harassment. Recently, Google and Jigsaw started a project called Perspective, which uses machine learning to automatically detect toxic language. A demonstration website has been also launched, which allows anyone to type a phrase in the interface and instantaneously see the toxicity score. In this paper, we propose an attack on the Perspective toxic detection system based on the adversarial examples. We show that an adversary can subtly modify a highly toxic phrase in a way that the system assigns significantly lower toxicity score to it. We apply the attack on the sample phrases provided in the Perspective website and show that we can consistently reduce the toxicity scores to the level of the non-toxic phrases. The existence of such adversarial examples is very harmful for toxic detection systems and seriously undermines their usability.

3) Ex Machina: Personal Attacks Seen at Scale

Authors: Ellery Wulczyn , Nithum Thain , Lucas Dixon

Description:

The damage personal attacks cause to online discourse motivates many platforms to try to curb the phenomenon. However, understanding the prevalence and impact of personal attacks in online platforms at scale remains surprisingly difficult. The contribution of this paper is to develop and illustrate a method that combines crowdsourcing and machine learning to analyze personal attacks at scale.

We show an evaluation method for a classifier in terms of the aggregated number of crowd-workers it can approximate. We apply our methodology to English Wikipedia, generating a corpus of over 100k high quality human-labeled comments and 63M machine-labeled ones from a classifier that is as good as the aggregate of 3 crowd-workers, as measured by the area under the ROC curve and Spearman correlation. Using this corpus of machine labeled scores, our methodology allows us to explore some of the open questions about the nature of online personal attacks. This reveals that the majority of personal attacks on Wikipedia are not the result of a few malicious users, nor primarily the consequence of allowing anonymous contributions from unregistered users.

4) Convolutional Neural Networks for Toxic Comment Classification

Authors: Spiros V. Georgakopoulos , Sotiris K. Tasoulis , Aristidis G. Vrahatis , Vassilis P. Plagianakos

Description:

Flood of information is produced in a daily basis through the global internet usage arising from the online interactive communications among users. While this situation contributes significantly to the quality of human life, unfortunately it involves enormous dangers, since online texts with high toxicity can cause personal attacks, online harassment and bullying behaviors. This has triggered both industrial and research community in the last few years while there are several tries to identify an efficient model for online toxic comment prediction. However, these steps are still in their infancy and new approaches and frameworks are required. On parallel, the data explosion that appears constantly, makes the construction of new machine learning computational tools for managing this information, an imperative need. Thankfully advances in hardware, cloud computing and big data management allow the development of Deep Learning approaches appearing very promising performance so far. For text classification in particular the use of Convolutional Neural Networks (CNN) have recently been proposed approaching text analytics in a modern manner emphasizing in the structure of words in a document. In this work, we employ this approach to discover toxic comments in a large pool of documents provided by a current Kaggle's competition regarding Wikipedia's talk page edits. To justify this decision we choose to compare CNNs against the traditional bag-of-words approach for text analysis combined with a selection of algorithms proven to be very effective in text classification. The reported results provide enough evidence that CNN enhance toxic comment classification reinforcing research interest towards this direction.

5) Toxic Comment Tools: A Case Study

Authors: Pooja Parekh , Hetal Patel

Description:

The online web tools enable everyone who can confidently voice their opinions in public sphere. The opinion may be encouragement, blessing or good suggestion i.e. positive, or it may be negative to that extent that it must be restricted at some point. The aim of this paper is to survey the different machine learning techniques employed within the scope of discovering the hateful language on social networking site, their challenges to provide a solution to detect the toxic comment and modify it of the same.

6) Text classification of short messages (Detecting inappropriate comments in online user debates)

Authors: Anton Lundborg

Description:

Almost every large Swedish online newspaper has disabled comments under their articles due to problems with hateful and offensive comments. In this Master's thesis, we explore different ways to detect toxic comments using machine learning. We carry out a comparison of classification algorithms and evaluate a number of different feature sets with the goal of optimizing accuracy for the classification of comments. We carry out the experiment with a manually labeled data set. The best classifier was logistic regression with the f-score of 0.47 and recall of 0.50. We incorporated the classifier into a moderation tool for comments to help streamline the moderation process.

2.2 Scope of the Literature Survey

A conversation AI team are working on tools, which could possibly help to improve the online conversations. It is a research initiative founded by Jigsaw and Google (both are the parts of Alphabet).The main area of focus is the study of negative online behaviours, like toxic comments (i.e. comments that are rude, disrespectful or otherwise likely to make someone leave a discussion).

Perspective API is an API which was developed my Google in their Jigsaw Incubator . The Perspective API offers developers and publishers a tool to help them spot toxicity online in an effort to support better discussions

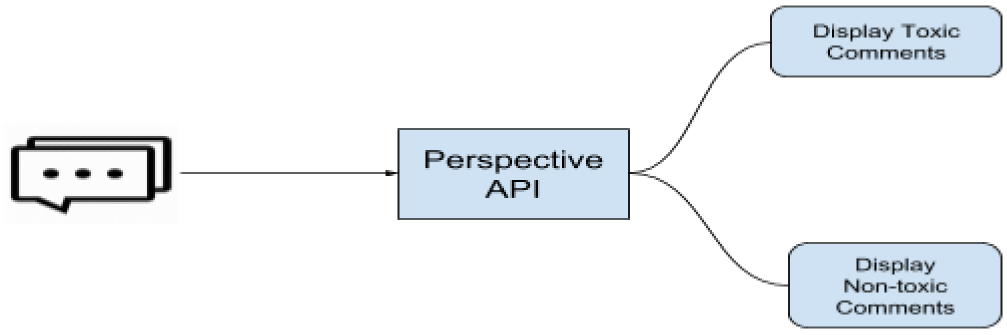


Figure 2.1 Perspective API Architecture

The above architecture diagram depicts the working of Perspective API

In the system Architecture:

- Initially , large amount of online comments are fetched from a discussion forums or social media websites.
- These comments are given as an input to the Perspective API.
- The Perspective API is an API which was developed by Google which is already trained to find the level of toxicity in the comments.
- The comments given as the input are checked for their level of toxicity .
- The comments are filtered into two types :
    - Toxic Comments
    - Non-Toxic Comments
- This API finally displays the comments which are toxic in nature.

2.3 Drawbacks of the System

- So far, the AI group has built certain models, which are publicly served using API. Still the current model makes errors and they are not much efficient.
- In addition, they do not allow users to select the type of toxicity they are interested in.
- Some websites may be fine with insult comments but not the other types of toxic comments.
- Also the model built is not capable of classifying what is the nature of the toxicity of the comment.

CHAPTER 3
SYSTEM ARCHITECTURE

3.1 Proposed System Architecture

The entire application contains a sequence of primary processes. Each of them are considered primary because it is interdependent. The following is the flow of our application which classifies the comments based on the toxicity of the comments

Initially a dataset is created or fetched from the internet sources which consists of lot of comments and their corresponding nature of the toxicity .The fetched dataset is used to create a learning model by extracting the features from the datasets and combining it with one of the data classification algorithms. In our case Logistic Regression is used as the data classification algorithm.Once the learning process ends , the result model which is obtained is the trained model , which is capable of identifying the toxic comments and it’s nature of toxicity or classifying the comments based on it’s toxic nature.In order to implement this feature , it is necessary to create a real world application software which could be capable of parallely executing the model in the backend.In our case , we are creating a Messaging Application using simple socket programming .The trained model is made to run as a script in the application backend and the messages which are sent through the Messaging Application and given as the input to the model.The model processes the obtained comment and checks for the toxicity of the comment . If the comment is toxic then the nature of the toxicity of the comment is identified and the type name is displayed beside the comment. In case if the comment is not toxic then nothing is displayed.

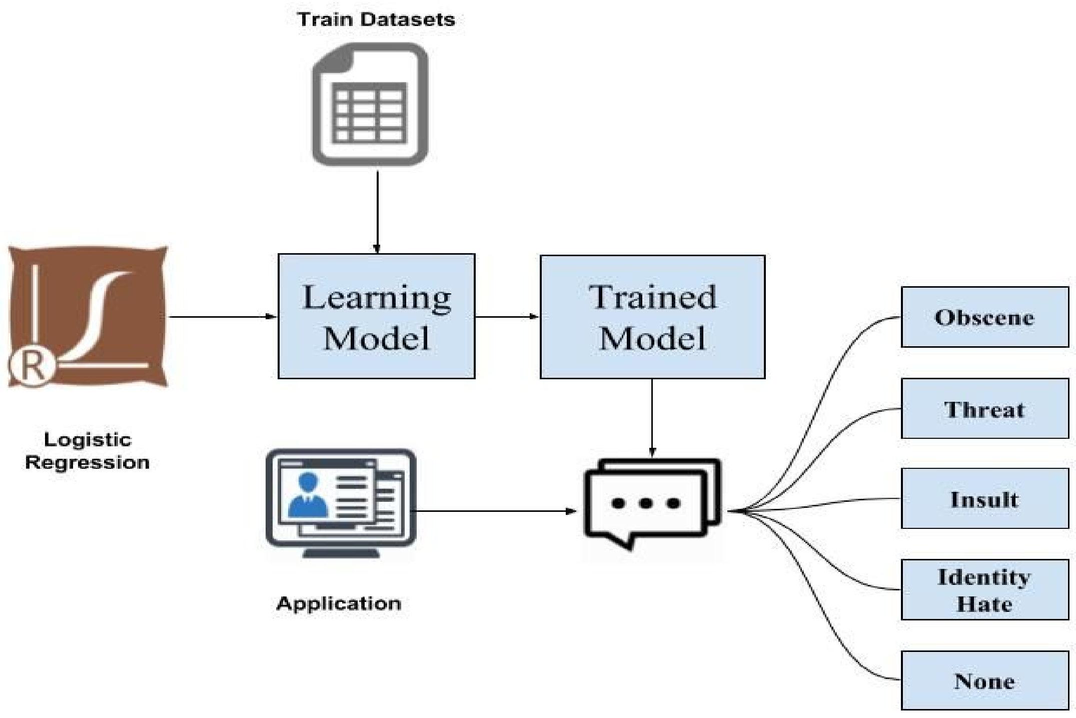


Figure. 3.1 Toxic Comment Classification using Machine Learning Architecture Diagram

3.2 System Requirements

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified software or project. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.

3.2.1 Software Requirements

The required Software packages to run the application are given below:

- Sklearn Kit for Feature Extraction and Data Classifier Libraries
- Anaconda for Machine Learning
- Git for version controlling system

- Python 2.7 or later
- Node JS framework version 6.0 or later for Server in Message Application

3.2.2 Hardware Requirements

The required hardware specifications are given below :

- Processor- 4X 16, GHz CPU
- 4GB RAM
- 80 GB (2 x 120) of SSD Storage
- 32v CPUS’s

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1 Modules

The following are the main modules of the proposed system which is prepared in order to aid in overcoming the problems faced by the existing system.

- Setting up an environment
    - Installing Python
    - Installing Anaconda
- Training the model using comments dataset
- Creating a classifier script
- Creating the Messaging Application
- Integrating the Application with the trained model
- Execution of the Application

Anaconda is an open source distribution of Python which is used for large scale data processing , predictive analysis and scientific computation . Anaconda is a package which consists of various libraries such as numpy , scipy , matplotlib Pandas , Scikit etc .

Pandas is a package provided by python which gives implements flexible, expressive and fast data structures designed to work with labelled or relational data both easy and intuitive. It is a high-level building block which is used for doing practical and real world data analysis in python.

Scikit is a python Machine Learning library . It features various classification, regression and clustering algorithms including support vector machines, random forests ,gradient boosting , k-means and DBSCAN , and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

4.2 Setting up an environment

a) Installing Python

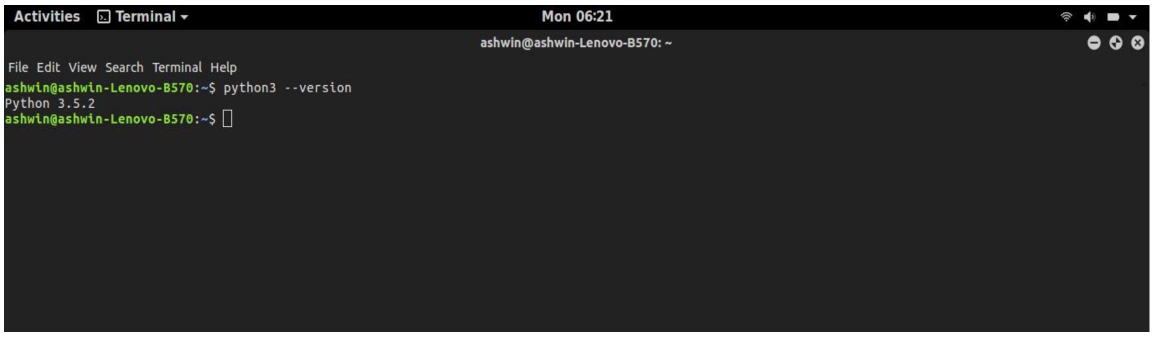
Installing Python 2.7 or later in Ubuntu 14.04 LTS and above.

Command to Install Python:

```
$ sudo apt-get update
$ sudo apt-get install python3.4
```

Command to check the version if already installed:

```
$ python3 --version
```


Activities
Terminal
Mon 06:21
ashwin@ashwin-Lenovo-B570: ~
File Edit View Search Terminal Help

```
ashwin@ashwin-Lenovo-B570:~$ python3 --version
Python 3.5.2
ashwin@ashwin-Lenovo-B570:~$
```

Figure 4.1 Python Installation

4.3 Install Anaconda

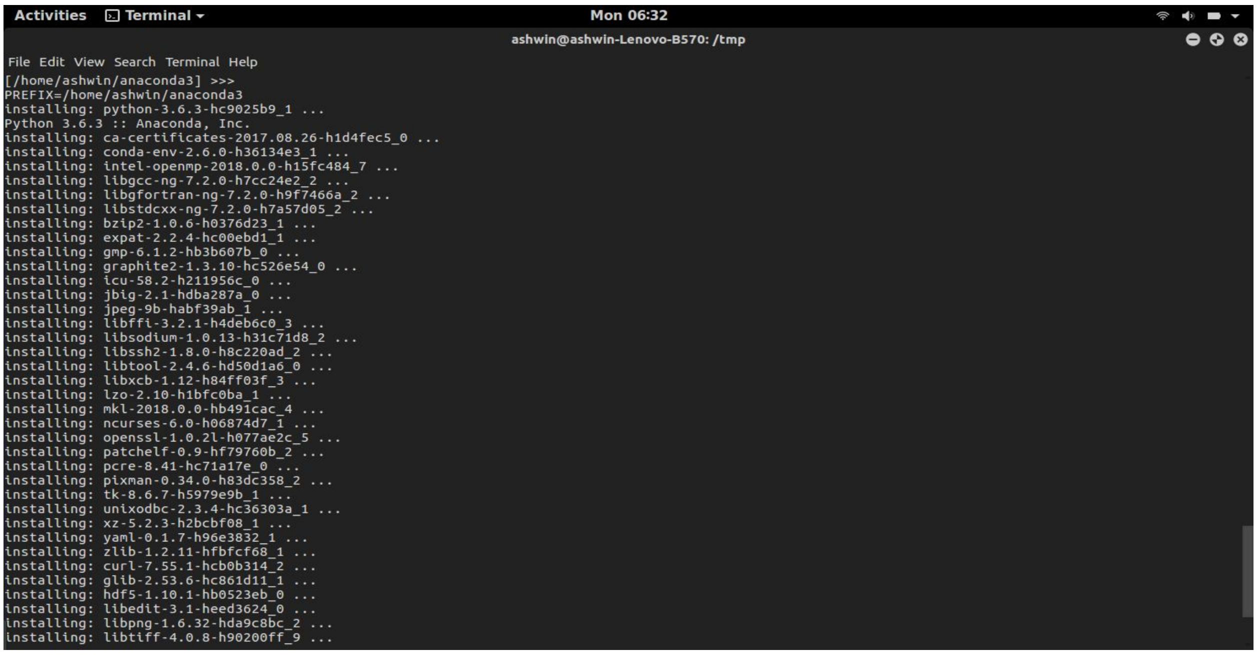
Installing Anaconda which is a package which contains libraries such as Pandas,Scikit , NumPy , SciPy etc.Here we use Pandas and Scikit as main libraries for the project.

Command to download Anaconda :

```
$ cd /tmp
$ curl -O https://repo.continuum.io/archive/Anaconda3-5.0.1-Linux-x86_64.sh
```

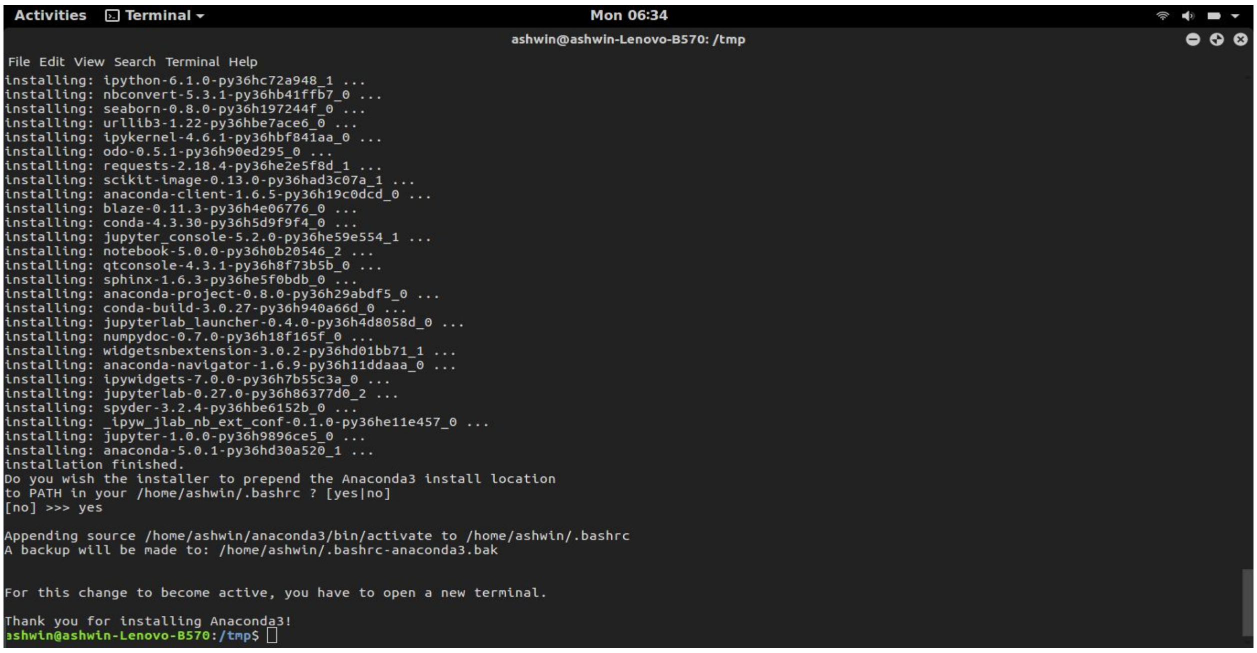
Command to start Anaconda :

```
$ bash Anaconda3-5.0.1-Linux-x86_64.sh
```



Activities
Terminal
Mon 06:32
ashwin@ashwin-Lenovo-B570: /tmp
File Edit View Search Terminal Help

```
[/home/ashwin/anaconda3] >>>
PREFIX=/home/ashwin/anaconda3
installing: python-3.6.3-hc9025b9_1 ...
Python 3.6.3 :: Anaconda, Inc.
installing: ca-certificates-2017.08.26-h1d4fec5_0 ...
installing: conda-env-2.6.0-h36134e3_1 ...
installing: intel-openmp-2018.0.0-h15fc484_7 ...
installing: libgcc-ng-7.2.0-h7cc24e2_2 ...
installing: libgfortran-ng-7.2.0-h9f7466a_2 ...
installing: libstdcxx-ng-7.2.0-h7a57d05_2 ...
installing: bzip2-1.0.6-h0376d23_1 ...
installing: expat-2.2.4-hc00ebd1_1 ...
installing: gmp-6.1.2-hb3b607b_0 ...
installing: graphite2-1.3.10-hc526e54_0 ...
installing: icu-58.2-h211956c_0 ...
installing: jbig-2.1-hdba287a_0 ...
installing: jpeg-9b-habf39ab_1 ...
installing: libffi-3.2.1-h4deb6c0_3 ...
installing: libsodium-1.0.13-h31c71d8_2 ...
installing: libssh2-1.8.0-h8c220ad_2 ...
installing: libtool-2.4.6-hd50d1a6_0 ...
installing: libxcb-1.12-h84ff03f_3 ...
installing: lzo-2.10-h1bfc0ba_1 ...
installing: mkl-2018.0.0-hb491cac_4 ...
installing: ncurses-6.0-h06874d7_1 ...
installing: openssl-1.0.2l-h077ae2c_5 ...
installing: patchelf-0.9-hf79760b_2 ...
installing: pcre-8.41-hc71a17e_0 ...
installing: pixman-0.34.0-h83dc358_2 ...
installing: tk-8.6.7-h5979e9b_1 ...
installing: unixodbc-2.3.4-hc36303a_1 ...
installing: xz-5.2.3-h2bcbf08_1 ...
installing: yaml-0.1.7-h96e3832_1 ...
installing: zlib-1.2.11-hfbfcf68_1 ...
installing: curl-7.55.1-hcb0b314_2 ...
installing: glib-2.53.6-hc861d11_1 ...
installing: hdf5-1.10.1-hb0523eb_0 ...
installing: libedit-3.1-heed3624_0 ...
installing: libpng-1.6.32-hda9c8bc_2 ...
installing: libtiff-4.0.8-h90200ff_9 ...
```



Activities
Terminal
Mon 06:34
ashwin@ashwin-Lenovo-B570: /tmp
File Edit View Search Terminal Help

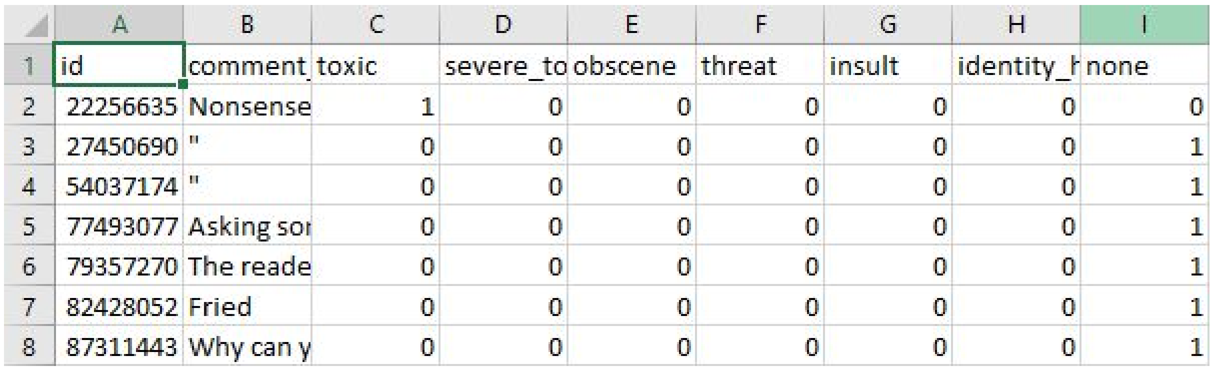
```
installing: ipython-6.1.0-py36hc72a948_1 ...
installing: nbconvert-5.3.1-py36hb41ffb7_0 ...
installing: seaborn-0.8.0-py36h197244f_0 ...
installing: urllib3-1.22-py36hbe7ace6_0 ...
installing: ipykernel-4.6.1-py36hbf841aa_0 ...
installing: odo-0.5.1-py36h90ed295_0 ...
installing: requests-2.18.4-py36he2e5f8d_1 ...
installing: scikit-image-0.13.0-py36had3c07a_1 ...
installing: anaconda-client-1.6.5-py36h19c0dcd_0 ...
installing: blaze-0.11.3-py36h4e06776_0 ...
installing: conda-4.3.30-py36h5d9f9f4_0 ...
installing: jupyter_console-5.2.0-py36he59e554_1 ...
installing: notebook-5.0.0-py36h0b20546_2 ...
installing: qtconsole-4.3.1-py36h8f73b5b_0 ...
installing: sphinx-1.6.3-py36he5f0bdb_0 ...
installing: anaconda-project-0.8.0-py36h29abdf5_0 ...
installing: conda-build-3.0.27-py36h940a66d_0 ...
installing: jupyterlab_launcher-0.4.0-py36h4d8058d_0 ...
installing: numpydoc-0.7.0-py36h18f165f_0 ...
installing: widgetsnbextension-3.0.2-py36hd01bb71_1 ...
installing: anaconda-navigator-1.6.9-py36h11ddaaa_0 ...
installing: ipywidgets-7.0.0-py36h7b55c3a_0 ...
installing: jupyterlab-0.27.0-py36h86377d0_2 ...
installing: spyder-3.2.4-py36hbe6152b_0 ...
installing: _ipyw_jlab_nb_ext_conf-0.1.0-py36he11e457_0 ...
installing: jupyter-1.0.0-py36h9896ce5_0 ...
installing: anaconda-5.0.1-py36hd30a520_1 ...
installation finished.
Do you wish the installer to prepend the Anaconda3 install location
to PATH in your /home/ashwin/.bashrc ? [yes|no]
[no] >>> yes

Appending source /home/ashwin/anaconda3/bin/activate to /home/ashwin/.bashrc
A backup will be made to: /home/ashwin/.bashrc-anaconda3.bak


For this change to become active, you have to open a new terminal.

Thank you for installing Anaconda3!
ashwin@ashwin-Lenovo-B570:/tmp$ 
```

Figure 4.2 Python Anaconda Installation

4.4 Training the Model using Comments Datasets

A trained model is created by providing the model with several datasets and allowing it to learn the behaviour of the datasets . In our project , a dataset is fetched from the internet resources which consists of large set of comments taken from online discussion forums and social media comments.

The below image is a snippet of the representation of comment dataset.



	A	B	C	D	E	F	G	H	I
1	id	comment_	toxic	severe_to	obscene	threat	insult	identity_h	none
2	22256635	Nonsense	1	0	0	0	0	0	0
3	27450690	"	0	0	0	0	0	0	1
4	54037174	"	0	0	0	0	0	0	1
5	77493077	Asking sor	0	0	0	0	0	0	1
6	79357270	The reade	0	0	0	0	0	0	1
7	82428052	Fried	0	0	0	0	0	0	1
8	87311443	Why can y	0	0	0	0	0	0	1

Figure 4.3 Screenshot of Training Data Set

4.4.1 Feature Extraction

Feature Extraction in Machine Learning is defined as process which is used to extract features from the given datasets . The process starts from an initial set of measured data and it uses those data to build the derived values which is also called as **features** which are intended to be non-redundant and informative. It also facilitates the subsequent learning and generalization steps , and in some cases it also leads to better human interpretations. Feature extraction is related to complexity reduction.

In our project , we have used TfidfVectorizer which is a module from Scikit Library . A TfidfVectorizer converts or transforms text into feature vectors which could be in turn used as an input to the estimator. In technical terms it converts a collection of raw documents to a matrix of TF-IDF features.

Module Importing :

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

Creating a TfidfVectorizer Object :

```
tfv = TfidfVectorizer(min_df=3, max_df=0.9, max_features=None,
        strip_accents='unicode', analyzer='word', token_pattern=r'\w{1,}',
        ngram_range=(1, 2), use_idf=1, smooth_idf=1,
        sublinear_tf=1, stop_words='english')
```

Learning vocabulary and idf from training set :

```
tfv.fit(X_train)
```

4.4.2 Saving the extracted feature using Pickle

The feature which was extracted from the datasets have to be saved using suitable file formats for further processing . The extracted feature is saved in .sav format file . This format is used to save statistical datas in it .

Pickle Module

Pickle is a python module which is used for serializing and de-serializing a python object structure . Any type of object in python can be pickled and saved into the disk . The python objects are saved in the form of character streams.

Module Importing :

```
import pickle
```

Syntax of Pickle:

```
pickle.dump(python_object,open(filename,'wb'))
```

In our example :

```
filename = "fit_model.sav"
pickle.dump(tfv,open(filename,'wb'))
```

4.4.3 Creating models using extracted features

Typically , features are the labels which are used to differentiate the behaviour of each and every data from the other datas . In this project the labels are the entities that differentiate the comments from each other .

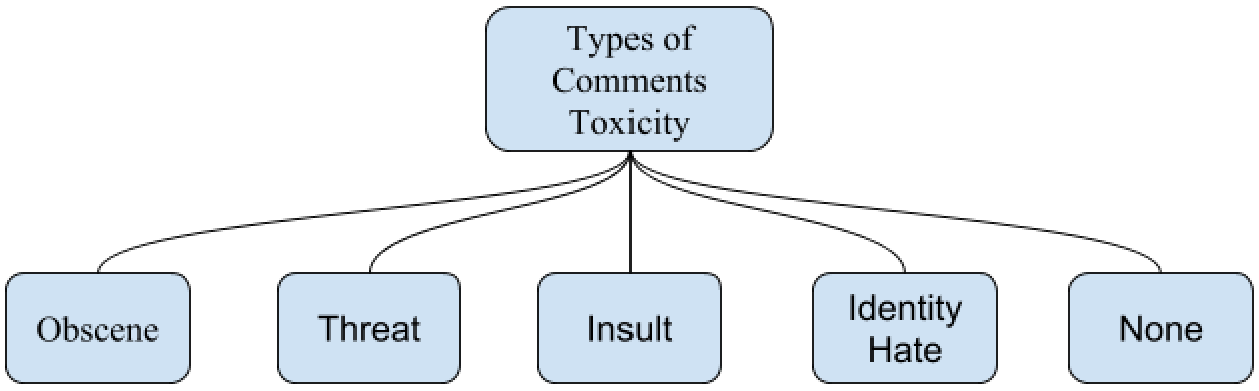


Figure 4.4 Types of Comments Toxicity

a) Logistic Regression

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

In our scenario , Logistic Regression is used because a there are many types of toxicity of comments (independent variables) which determine the type of the comment (outcome) .

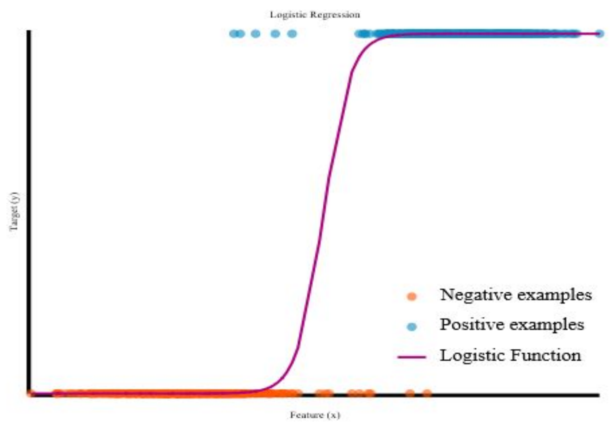


Figure 4.5 Sample representation of Logistic Regression using 2D graph

Module Importing:

```
from sklearn.linear_model import LogisticRegression
```

Creating object for Logistic Regression and training :

```
clf = LogisticRegression(C=4.0, solver='sag')
clf.fit(X_train, Y_train.iloc[:, i])
```

Saving the trained model :

```
filename = 'model_{}.sav'.format(feature)
pickle.dump(clf,open(filename,'wb'))
```

Since there are five types of labels , five different models are created and saved into five different .sav files .

Obscene	:	model_obscene.sav
Threat	:	model_threat.sav
Insult	:	model_insult.sav
Identity Hate	:	model_identity_hate.sav
None	:	model_none.sav

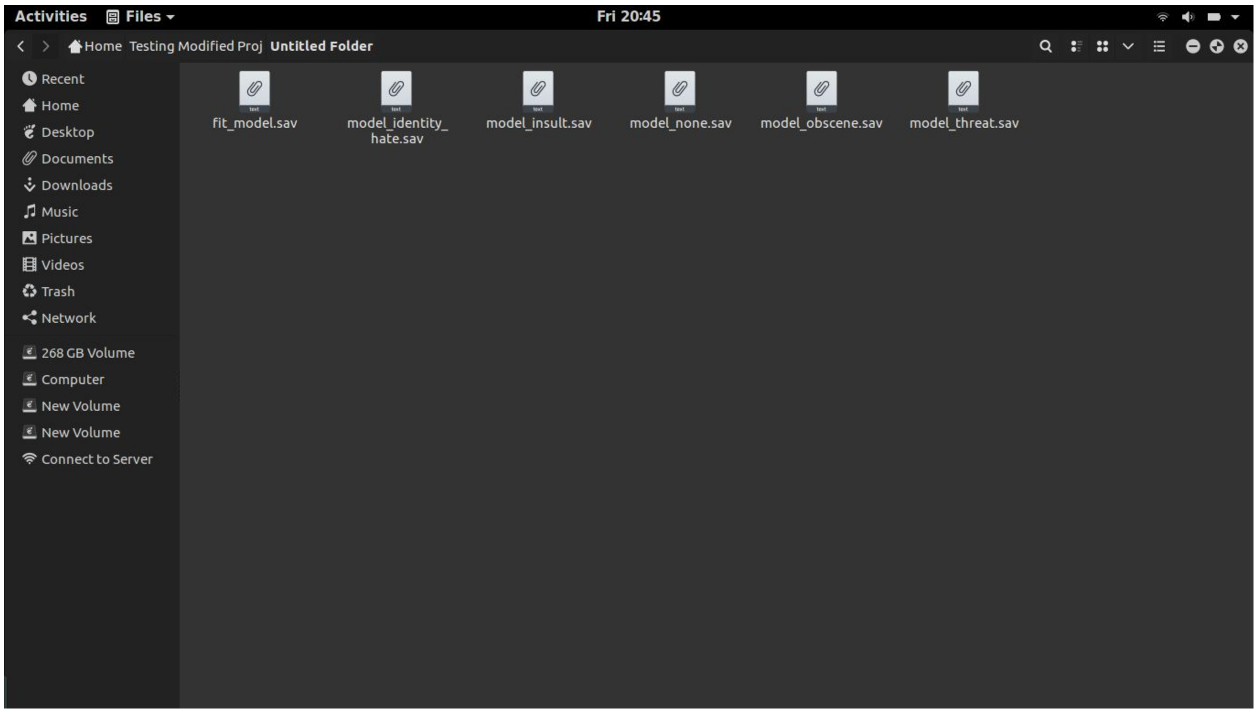


Figure 4.6 Screenshot of .sav files of models

4.5 Creating a Classifier Script

A classifier script (classifier.py) is a python script which is created in order to fetch the comment sent through the application , analyze the comment and find the toxicity nature of the comment and then return back the discovered toxicity label.

The classifier script runs in the background of the application in which the classification feature is integrated .

4.5.1 Transforming the comment into the extracted feature

The extracted feature from the datasets is saved in a .sav file (fit_model.sav) . This file is loaded using pickle.load() and the comment/message is transformed .

Code Snippet:

```
loaded_fit = pickle.load(open('fit_model.sav','rb'))
X_test = loaded_fit.transform(X_test) //X_test is the comment
```

4.5.2 Determining the nature of the comment using trained models

The trained models are saved into six .sav files . These files are loaded using the pickle.load() and are used to determine the probability value for each of the six models with the corresponding comment . The toxic label with highest probability value determines the toxicity nature of the comment . Since the comment has to be compared with six label models , it runs in the form of a for loop.

Code Snippet:

```
for i in range(Y_train.shape[1]):
feature = Y_train.columns[i]
loaded_model = pickle.load(open('model_{}.sav'.format(feature), 'rb'))
```

exec('pred_{}=pd.Series(loaded_model.predict_proba(X_test).flatten()[1::2]
)'.format(feature))

4.5.3 Creating a dataframe and converting it into .csv file

The predicted probability values are converted into pandas dataframe format and further the data frame created is converted into .csv file (result.csv).

Code Snippet:

```
submission = pd.DataFrame({
'obscene': pred_obscene,
'threat': pred_threat,
'insult': pred_insult,
'identity_hate': pred_identity_hate,
'none': pred_none })
submission.to_csv("result.csv", index=False)
```

Before running classifier.py

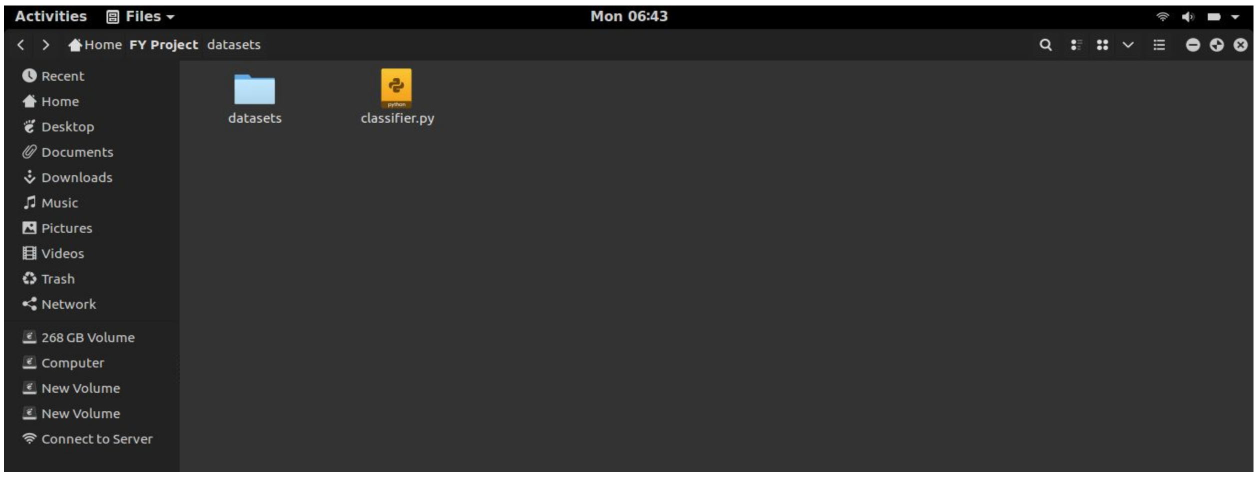


Figure 4.7 Before Running Classifier Script

After running classifier.py

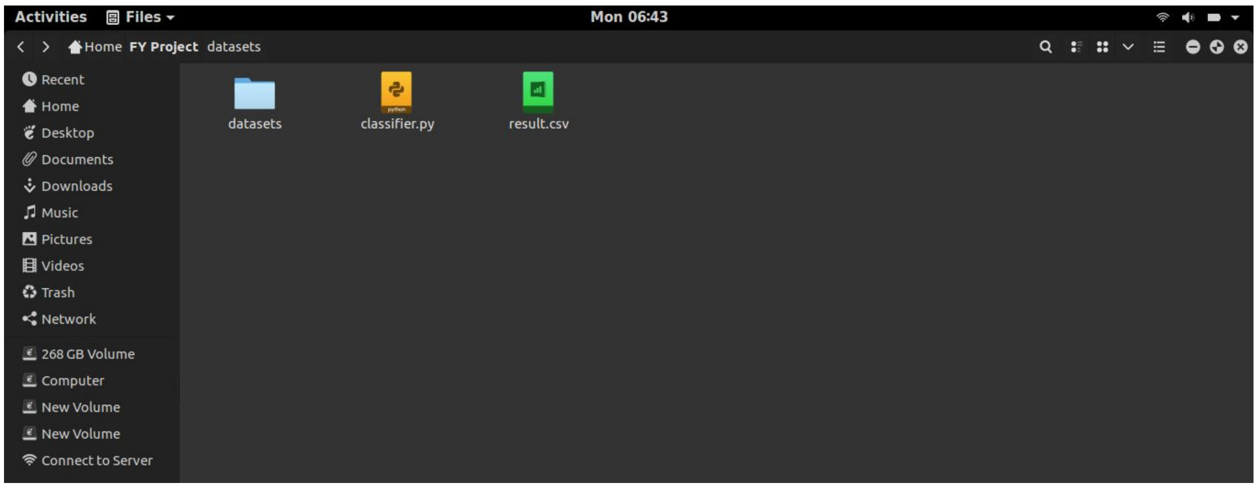


Figure 4.8 After Running Classifier Script

4.5.4 Determining the suitable toxic label

The suitable toxic label for the comment can be identified by finding the maximum probability value from the dataframe created using idxmax() function.

Code Snippet:

```
submission.idxmax(axis=1).get(0)
```

4.6 Creating the Message Application

Writing a chat application with popular web applications stacks like LAMP (PHP) has traditionally been very hard. It involves polling the server for changes, keeping track of timestamps, and it's a lot slower than it should be.

Sockets have traditionally been the solution around which most realtime chat systems are architected, providing a bi-directional communication channel between a client and a server.

This means that the server can push messages to clients. Whenever you write a chat message, the idea is that the server will get it and push it to all other connected clients.

4.6.1 Components used the Application

a) Node JS

Node.js is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML, to be run client-side by a JavaScript engine in the user's web browser. Node.js enables JavaScript to be used for server-side scripting, and runs scripts server-side to produce dynamic web page content *before* the page is sent to the user's web browser. Consequently, Node.js has become one of the foundational elements of the "JavaScript everywhere" paradigm, allowing web application development to unify around a single programming language, rather than rely on a different language for writing server side scripts.

Though .js is the conventional filename extension for JavaScript code, the name "Node.js" does not refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

The Node.js distributed development project, governed by the Node.js Foundation, is facilitated by the Linux Foundation Collaborative Projects program.

b) Node Package Manager (npm)

npm is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website. The package manager and the registry are managed by npm, Inc.

c) Express JS

Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It is in fact the standard server framework for Node.js.The original author, TJ Holowaychuk, described it as a Sinatra-inspired server,meaning that it is relatively minimal with many features available as plugins. Express is the backend part of the MEAN stack, together with MongoDB database and AngularJS frontend framework.

d) Socket.io

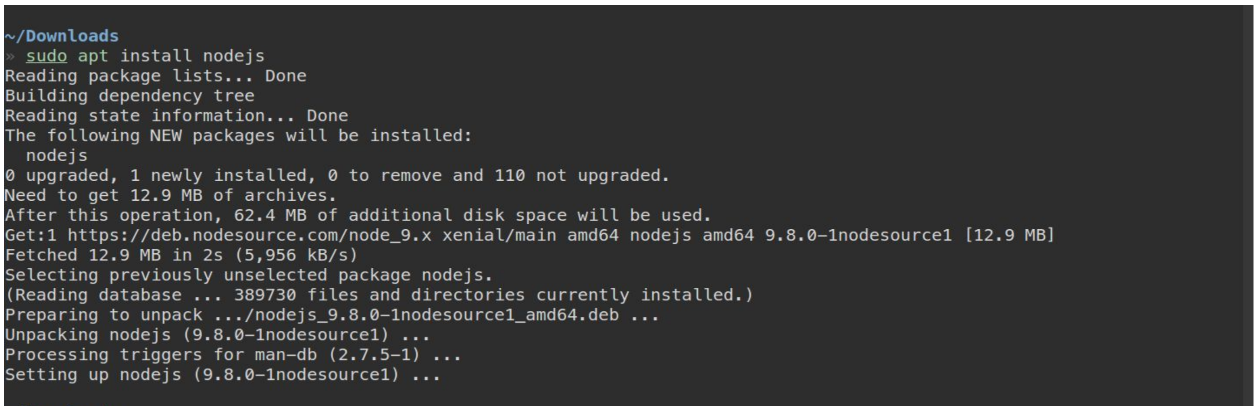
Socket.io is a JavaScript library for realtime web applications. It enables real time, bidirectional communication between web clients and servers. It has two parts: a client-side library that runs in the browser, and a server-side library for Node.js. Both components have a nearly identical API. Like Node.js, it is event-driven.

Socket.io primarily uses the WebSocket protocol with polling as a fallback option,while providing the same interface. Although it can be used as simply a wrapper for WebSocket, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O. It can be installed with the npm tool.

4.6.2 Installing Node JS and Node Package Manager (npm)

Commands :

```
$ sudo apt-get install nodejs
$ sudo apt-get install npm
```



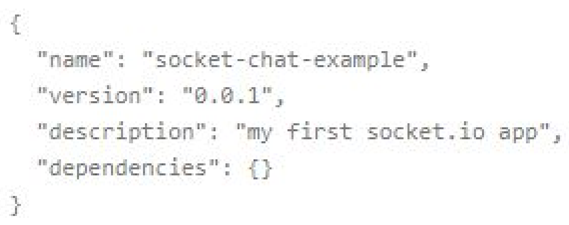
```
~/Downloads
» sudo apt install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 110 not upgraded.
Need to get 12.9 MB of archives.
After this operation, 62.4 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_9.x xenial/main amd64 nodejs amd64 9.8.0-1nodesource1 [12.9 MB]
Fetched 12.9 MB in 2s (5,956 kB/s)
Selecting previously unselected package nodejs.
(Reading database ... 389730 files and directories currently installed.)
Preparing to unpack .../nodejs_9.8.0-1nodesource1_amd64.deb ...
Unpacking nodejs (9.8.0-1nodesource1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up nodejs (9.8.0-1nodesource1) ...
```

Figure 4.9 Installation of Node JS

4.6.3 The web framework

The first goal is to setup a simple HTML webpage that serves out a form and a list of messages. Here we are going to use the Node.JS web framework “express” to this end.

First a package.json manifest file is created that describes the project and it is put in an empty directory.



```
{
  "name": "socket-chat-example",
  "version": "0.0.1",
  "description": "my first socket.io app",
  "dependencies": {}
}
```

Also express js is installed using the command

```
$ node install --save express
```

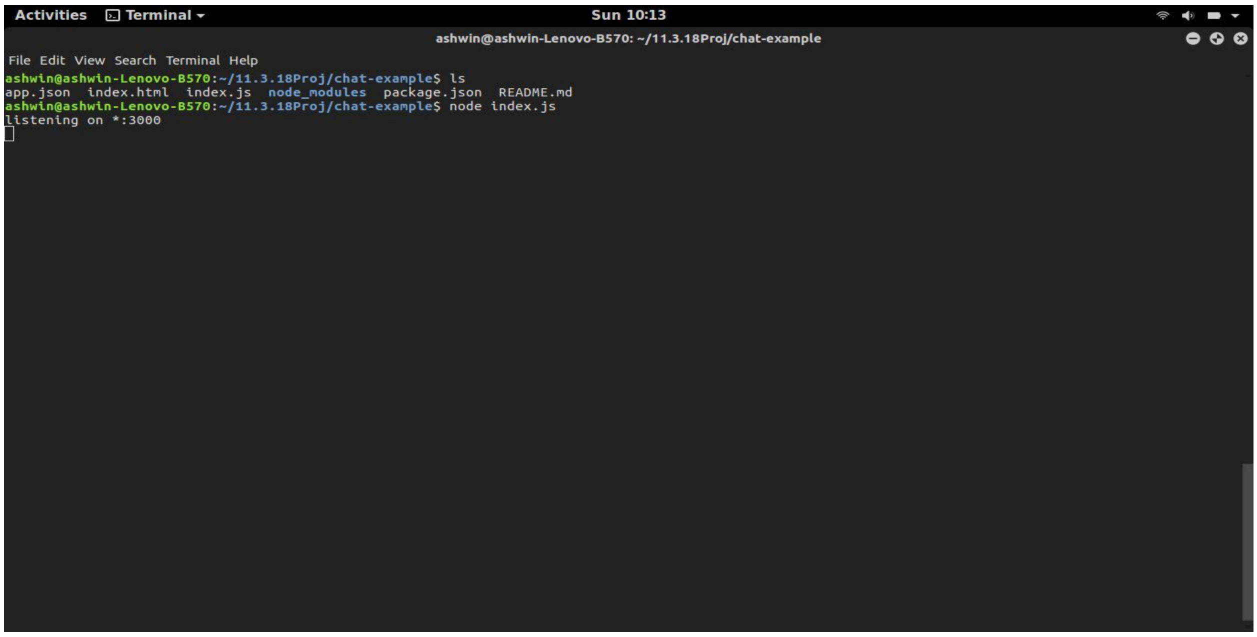
Now that express is installed we can create an index.js file that will setup our application.

```
var app = require('express')();
var http = require('http').Server(app);

app.get('/', function(req, res){
  res.send('<h1>Hello world</h1>');
});

http.listen(3000, function(){
  console.log('listening on *:3000');
});
```

This translates into the following:

1. Express initializes app to be a function handler that you can supply to an HTTP server (as seen in line 2).
2. We define a route handler / that gets called when we hit our website home.
3. We make the http server listen on port 3000.

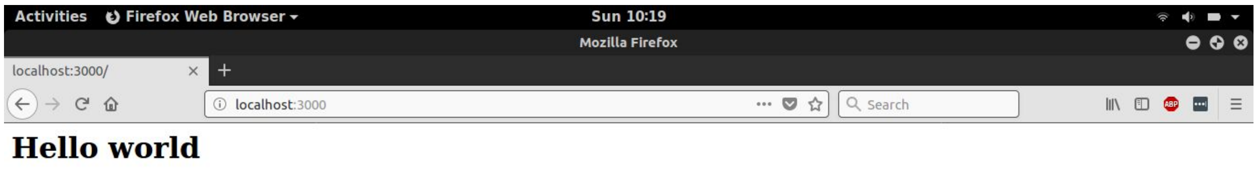
If you run node index.js you should see the following:


Activities
Terminal
Sun 10:13
ashwin@ashwin-Lenovo-B570: ~/11.3.18Proj/chat-example
File Edit View Search Terminal Help

```
ashwin@ashwin-Lenovo-B570:~/11.3.18Proj/chat-example$ ls
app.json  index.html  index.js  node_modules  package.json  README.md
ashwin@ashwin-Lenovo-B570:~/11.3.18Proj/chat-example$ node index.js
listening on *:3000
```

Figure 4.10 Running Node JS Server in Localhost

And if you point your browser to http://localhost:3000:



Hello world

Figure 4.11 Sample Server Running on Localhost Port 3000

4.6.4 Serving HTML

So far in index.js we're calling res.send and pass it a HTML string. Our code would look very confusing if we just placed our entire application's HTML there. Instead, we're going to create a index.html file and serve it.

Let's refactor our route handler to use sendFile instead:

And populate index.html with the following:

```
<!doctype html>
<html>
  <head>
    <title>Socket.IO chat</title>
    <style>
      * { margin: 0; padding: 0; box-sizing: border-box; }
      body { font: 13px Helvetica, Arial; }
      form { background: #000; padding: 3px; position: fixed; bottom: 0; width: 100%; }
      form input { border: 0; padding: 10px; width: 90%; margin-right: .5%; }
      form button { width: 9%; background: rgb(130, 224, 255); border: none; padding: 10px; }
      #messages { list-style-type: none; margin: 0; padding: 0; }
      #messages li { padding: 5px 10px; }
      #messages li:nth-child(odd) { background: #eee; }
    </style>
  </head>
  <body>
    <ul id="messages"></ul>
    <form action="">
      <input id="m" autocomplete="off" /><button>Send</button>
    </form>
  </body>
</html>
```

If you restart the process (by hitting Control+C and running node index again) and refresh the page it should look like this:

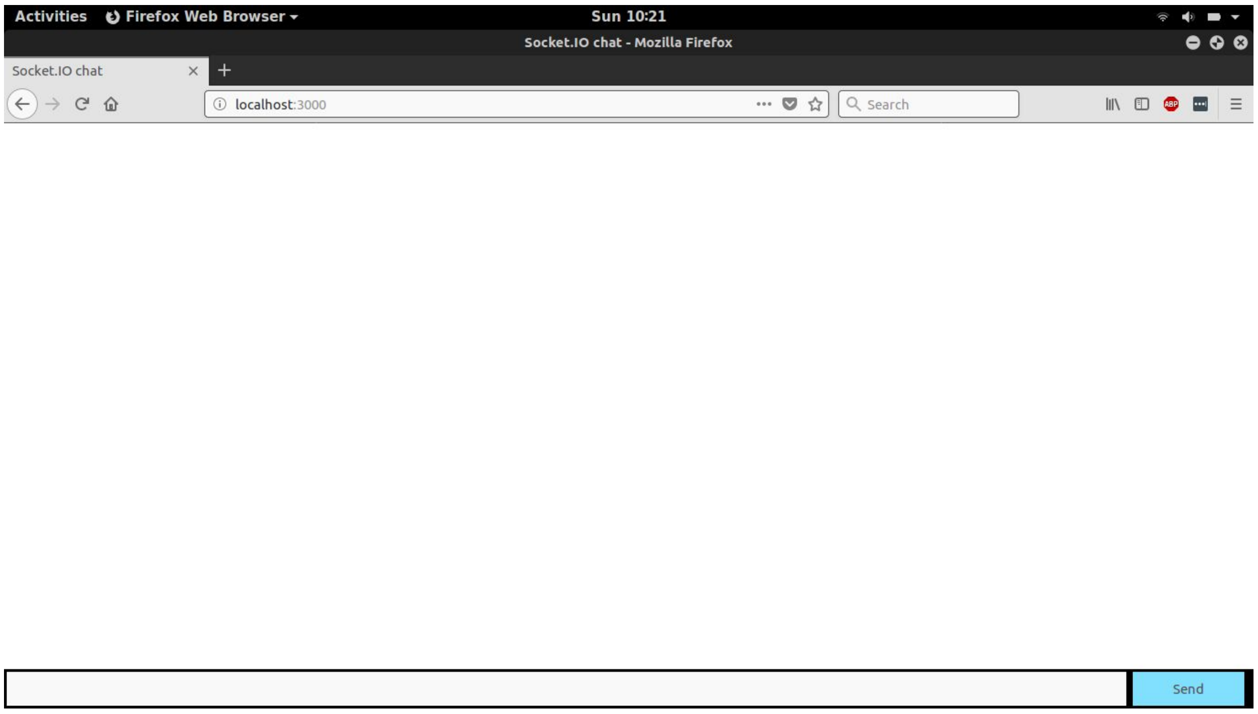


Figure 4.12 Message Application Client User Interface

4.6.5 Integrating Socket.IO

Socket.IO is composed of two parts:

- A server that integrates with (or mounts on) the Node.JS HTTP Server: socket.io
- A client library that loads on the browser side: socket.io-client

During development, socket.io serves the client automatically for us, as we'll see, so for now we only have to install one module:

```
$ npm install --save socket.io
```

That will install the module and add the dependency to package.json. Now let's edit index.js to add it:

```
var app = require('express')();
var http = require('http').Server(app);
var io = require('socket.io')(http);

app.get('/', function(req, res){
  res.sendFile(__dirname + '/index.html');
});

io.on('connection', function(socket){
  console.log('a user connected');
});

http.listen(3000, function(){
  console.log('listening on *:3000');
});
```

Notice that I initialize a new instance of socket.io by passing the http (the HTTP server) object. Then I listen on the connection event for incoming sockets, and I log it to the console.

Now in index.html I add the following snippet before the </body>:

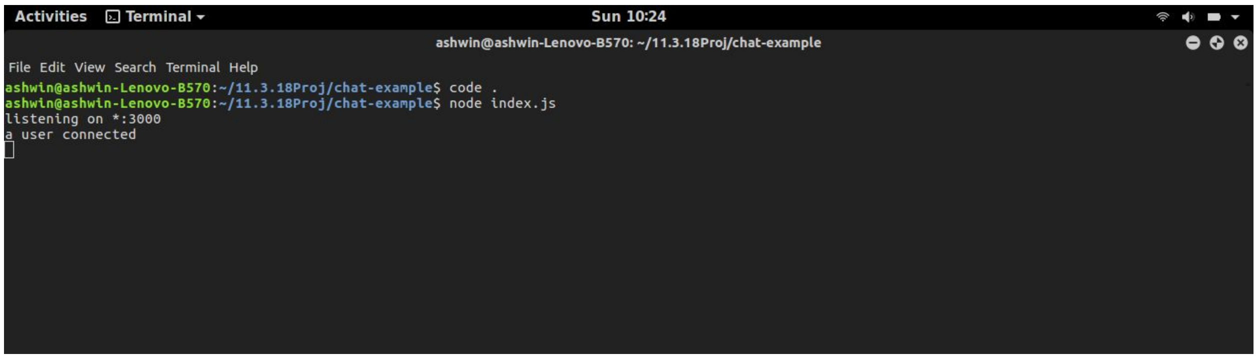
```
<script src="/socket.io/socket.io.js"></script>
<script>
  var socket = io();
</script>
```

That’s all it takes to load the socket.io-client, which exposes a io global, and then connect.

Notice that I’m not specifying any URL when I call io(), since it defaults to trying to connect to the host that serves the page.

If you now reload the server and the website you should see the console print “a user connected”.

Try opening several tabs, and you’ll see several messages:


Activities
Terminal
Sun 10:24
ashwin@ashwin-Lenovo-B570: ~/11.3.18Proj/chat-example
File Edit View Search Terminal Help

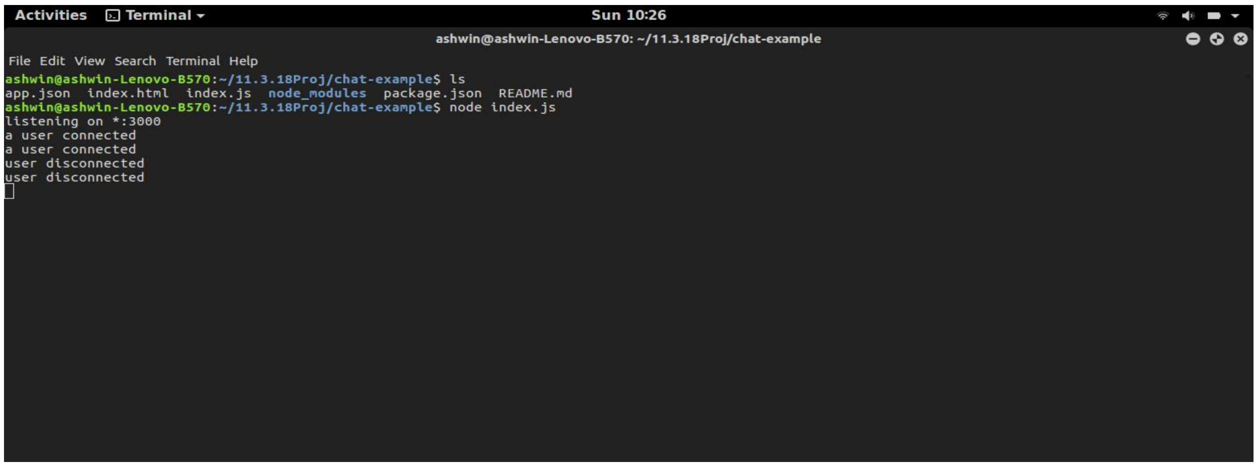
```
ashwin@ashwin-Lenovo-B570:~/11.3.18Proj/chat-example$ code .
ashwin@ashwin-Lenovo-B570:~/11.3.18Proj/chat-example$ node index.js
listening on *:3000
a user connected
```

Figure 4.13 Console Message when Single Client is Connected

Each socket also fires a special disconnect event:

```
io.on('connection', function(socket){
  console.log('a user connected');
  socket.on('disconnect', function(){
    console.log('user disconnected');
  });
});
```

Then if you refresh a tab several times you can see it in action:


Activities
Terminal
Sun 10:26
ashwin@ashwin-Lenovo-B570: ~/11.3.18Proj/chat-example
File Edit View Search Terminal Help

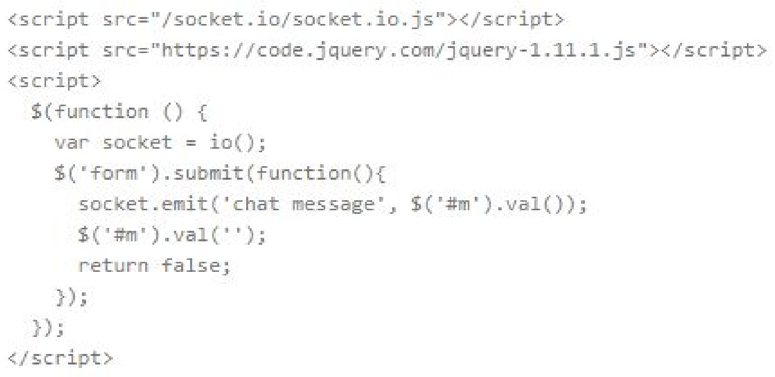
```
ashwin@ashwin-Lenovo-B570:~/11.3.18Proj/chat-example$ ls
app.json  index.html  index.js  node_modules  package.json  README.md
ashwin@ashwin-Lenovo-B570:~/11.3.18Proj/chat-example$ node index.js
listening on *:3000
a user connected
a user connected
user disconnected
user disconnected
```

Figure 4.14 Console Message when Multiple Clients are Connected

4.6.6 Emitting Events

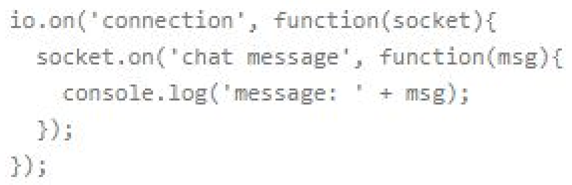
The main idea behind Socket.IO is that you can send and receive any events you want, with any data you want. Any objects that can be encoded as JSON will do, and binary data is supported too.

Let’s make it so that when the user types in a message, the server gets it as a chat message event. The scripts section in index.html should now look as follows:



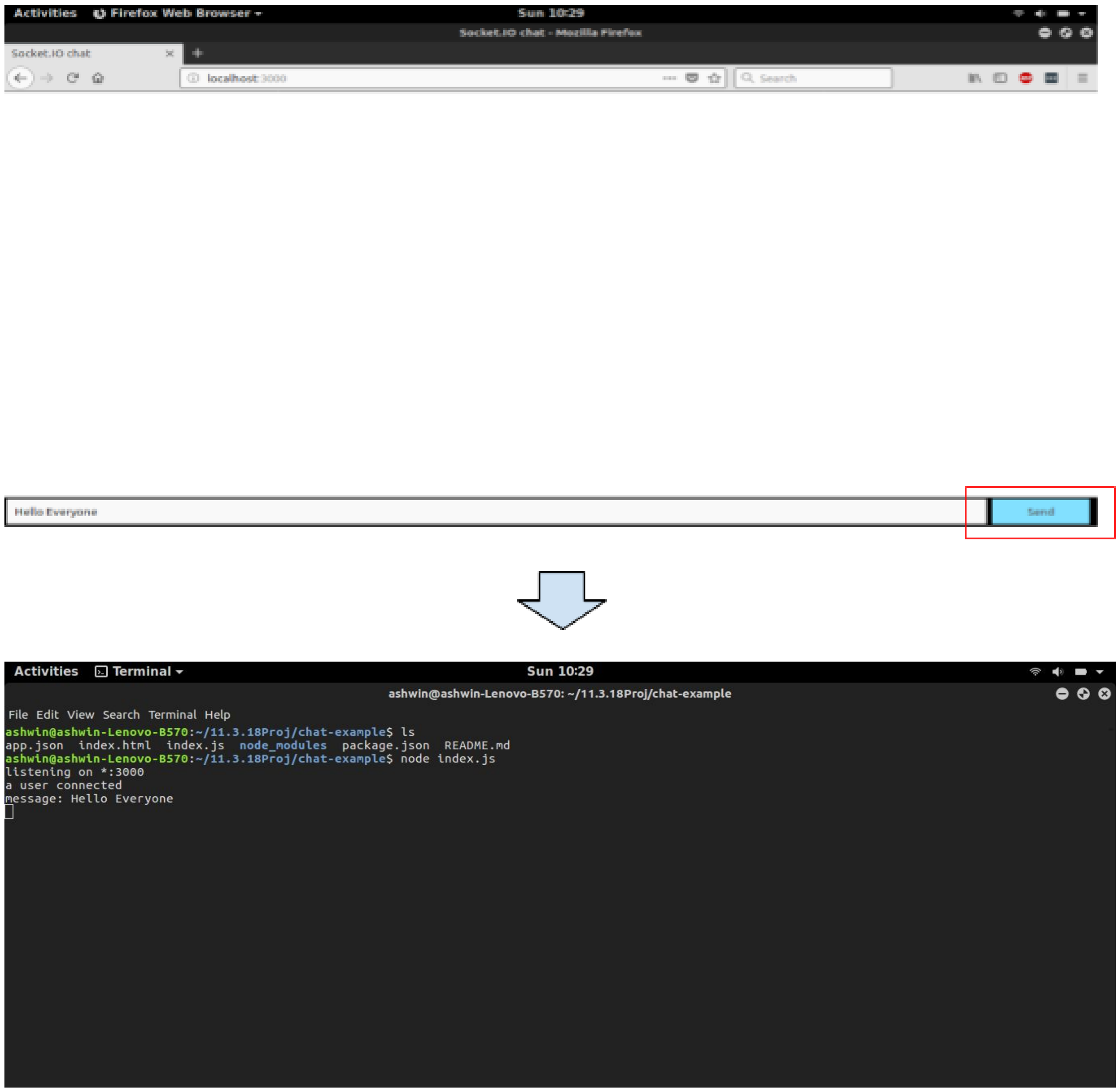
```
<script src="/socket.io/socket.io.js"></script>
<script src="https://code.jquery.com/jquery-1.11.1.js"></script>
<script>
  $(function () {
    var socket = io();
    $('form').submit(function(){
      socket.emit('chat message', $('#m').val());
      $('#m').val('');
      return false;
    });
  });
</script>
```

And in index.js we print out the chat message event:



```
io.on('connection', function(socket){
  socket.on('chat message', function(msg){
    console.log('message: ' + msg);
  });
});
```

The result should be like the following


Activities
Firefox Web Browser
Sun 10:29
Socket.IO chat - Mozilla Firefox
Socket.IO chat
localhost:3000
Search
Hello Everyone
Send
Activities
Terminal
Sun 10:29
ashwin@ashwin-Lenovo-B570: ~/11.3.18Proj/chat-example
File Edit View Search Terminal Help

```
ashwin@ashwin-Lenovo-B570:~/11.3.18Proj/chat-example$ ls
app.json  index.html  index.js  node_modules  package.json  README.md
ashwin@ashwin-Lenovo-B570:~/11.3.18Proj/chat-example$ node index.js
listening on *:3000
a user connected
message: Hello Everyone
```

Figure 4.15 Console Output for a Message sent through App

4.6.7 Broadcasting

The next goal is for us to emit the event from the server to the rest of the users.

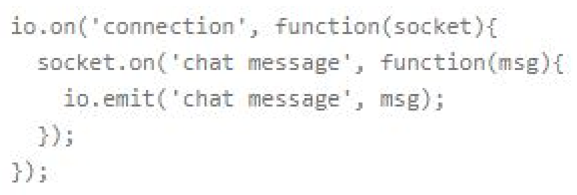
In order to send an event to everyone, Socket.IO gives us the io.emit:

io.emit('some event', { for: 'everyone' });

If you want to send a message to everyone except for a certain socket, we have the broadcast flag:

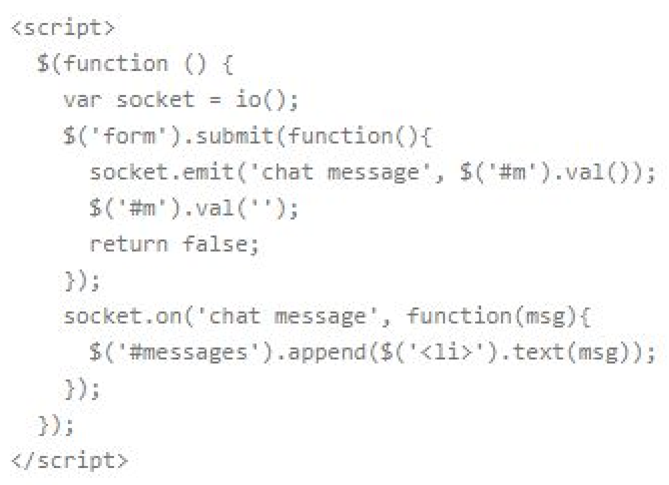
```
io.on('connection', function(socket){
 socket.broadcast.emit('hi');
});
```

In this case, for the sake of simplicity we'll send the message to everyone, including the sender.



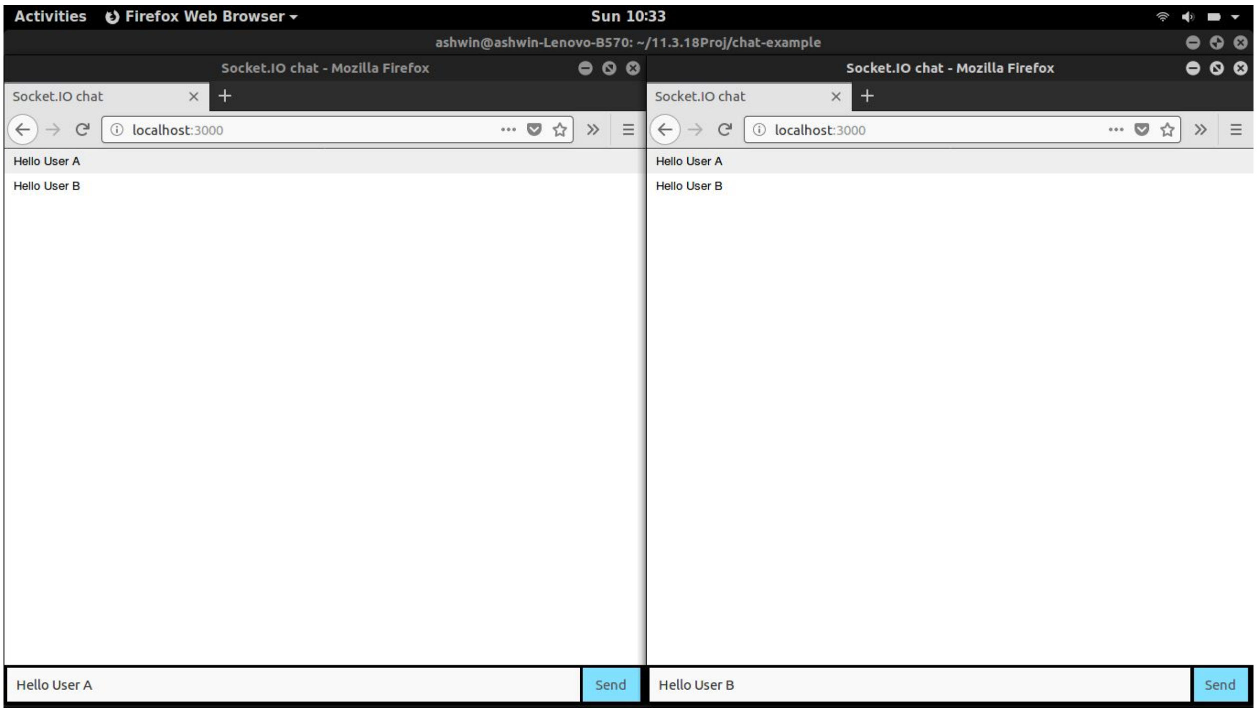
```
io.on('connection', function(socket){
  socket.on('chat message', function(msg){
    io.emit('chat message', msg);
  });
});
```

And on the client side when we capture a chat message event we'll include it in the page. The total client-side JavaScript code now amounts to:



```
<script>
  $(function () {
    var socket = io();
    $('form').submit(function(){
      socket.emit('chat message', $('#m').val());
      $('#m').val('');
      return false;
    });
    socket.on('chat message', function(msg){
      $('#messages').append($('<li>').text(msg));
    });
  });
</script>
```

And that completes our chat application. This is what it looks like:


Activities
Firefox Web Browser
Sun 10:33
ashwin@ashwin-Lenovo-B570: ~/11.3.18Proj/chat-example
Socket.IO chat - Mozilla Firefox
Socket.IO chat
localhost:3000
Socket.IO chat - Mozilla Firefox
Socket.IO chat
localhost:3000
Send
Send

```
Hello User A
Hello User B
```

```
Hello User A
Hello User B
```

Figure 4.16 Screenshot of Conversation of Two Clients using the Message Application

4.7 Integrating Application With Trained Model

Now as a sample messaging application using sockets is ready , the trained machine learning model is integrated with the application for processing the comments / messages sent through the application .

4.7.1 Modifications in index.html file

The message application is designed in such a way that if the comment is toxic (any type of toxicity nature) then the toxicity label is displayed to the right side of the comment , if not then nothing will be displayed .

To add this functionality the following code is added to index.html file

- **To add colour to the label**

```
.result {
        color: red;
        font-size: 20px;
        float: right;
    }
```

- **Functionality**

```
socket.on('chat message', function (data) {let msg = data.msg;
let result = data.result !== 'none' ? `<span class='result'>${data.result}</span>` : ''
        $('#messages').append($('<li>').html(`<p>${msg}  <span class='result'>${result}</span></p>`));
  });
 });
```

4.7.2 Modification in index.js file

While modifying the index.js file we should consider the situation where if a message is sent through the application , then the trained model script (classifier.py) has to be triggered automatically which is similar to an event listener process .

The other challenge here is to run a python script in a nodejs application . To do this we need to install an Python Shell which is an npm package .

Installing python-shell :

```
$ npm install --save python-shell
```

Importing module :

```
$ var PythonShell = require("python-shell");
```

Running python script (classifier.py) in node js

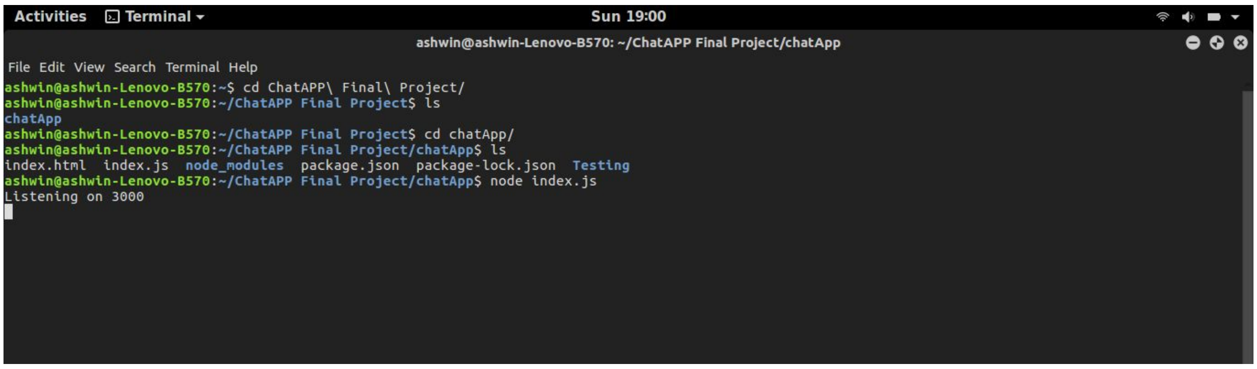
```
io.on("connection", function(socket) {
socket.on("chat message", function(msg) {
        var options = {
                mode: "text",
                pythonPath:"/home/ganesh/anaconda3/bin/python",
                 pythonOptions: ["-u"],
                    scriptPath: "/home/ganesh/Downloads/chatapp/Testing",
                    args: [msg]
                };
            PythonShell.run("classifier.py", options, function(err, results) {
             if (err) throw err;
                console.log(results[0]);
            io.emit("chat message", { msg, result: results[0] });
            });
        });
});
```

4.8 Execution of The Application

To execute the message application , go to the terminal, change the directory to the path where the message application is saved and run index.js file .

Running index.js file :

```
$ cd “path of message application”
$ node index.js
```



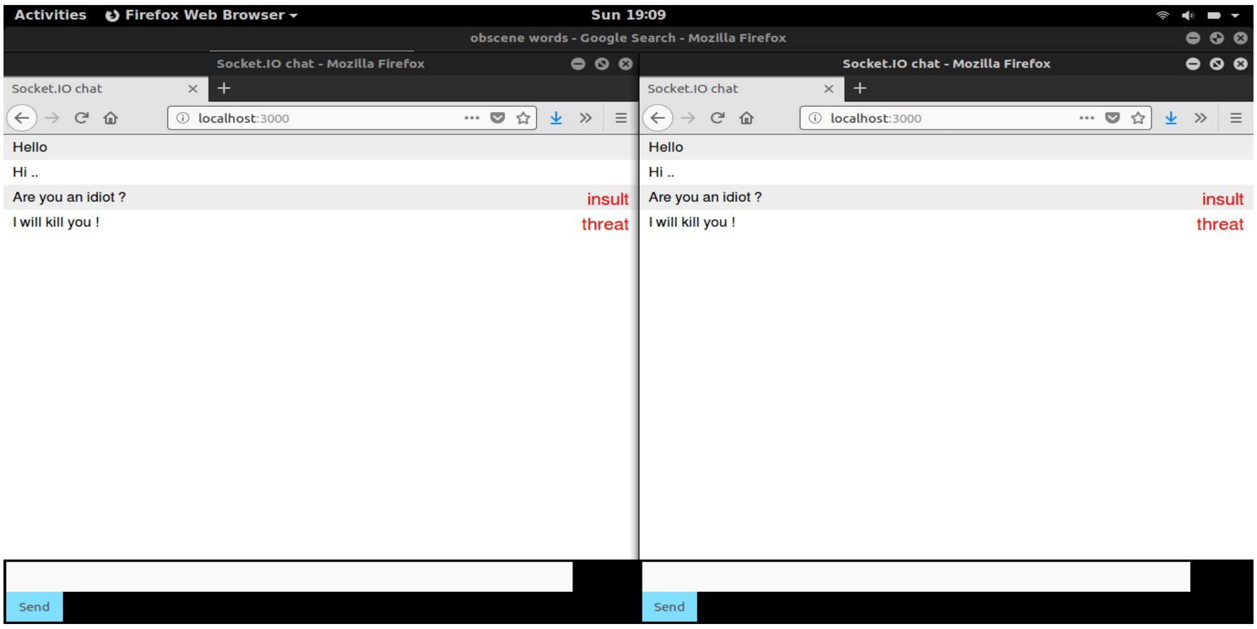
Activities
Terminal
Sun 19:00
ashwin@ashwin-Lenovo-B570: ~/ChatAPP Final Project/chatApp
File Edit View Search Terminal Help

```
ashwin@ashwin-Lenovo-B570:~$ cd ChatAPP\ Final\ Project/
ashwin@ashwin-Lenovo-B570:~/ChatAPP Final Project$ ls
chatApp
ashwin@ashwin-Lenovo-B570:~/ChatAPP Final Project$ cd chatApp/
ashwin@ashwin-Lenovo-B570:~/ChatAPP Final Project/chatApp$ ls
index.html  index.js  node_modules  package.json  package-lock.json  Testing
ashwin@ashwin-Lenovo-B570:~/ChatAPP Final Project/chatApp$ node index.js
Listening on 3000
```

Figure 4.17

Now the app will be running in the http://localhost:3000

Two different clients are opened and messages are exchanged between them.

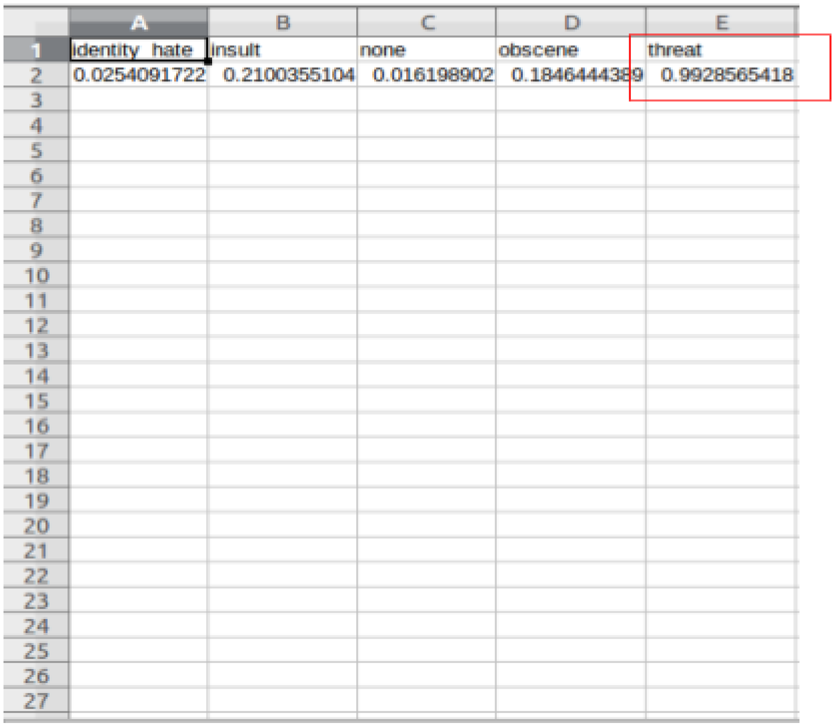

Activities
Firefox Web Browser
Sun 19:09
obscene words - Google Search - Mozilla Firefox
Socket.IO chat - Mozilla Firefox
Socket.IO chat
localhost:3000
Send

```
Hello
Hi ..
Are you an idiot ?                      insult
I will kill you !                       threat
```

Figure 4.18 Message Application Classifying Message based on the Nature of its Toxicity

From the above image we could observe that the comments like “Hi”, “Hello Everyone” doesn’t detect any toxicity in it . Since other two comments are toxic , the nature of toxicity is displayed in red colour to the right side of the comment .

Now let’s observe the **result.csv** file obtained here .



	A	B	C	D	E
1	identity_hate	insult	none	obscene	threat
2	0.0254091722	0.2100355104	0.016198902	0.1846444389	0.9928565418

Figure 4.19 Screenshot of the result.csv file generated for each message

For each and every message the same result.csv file is overwritten. Since our last message was “I will kill you” which is definitely a **threat** message , we could see that the probability value in the threat label column is the maximum compared to other label columns . This is how the suitable toxicity label for each and every comment / message is determined.

4.9 Simple Algorithm for the proposed project

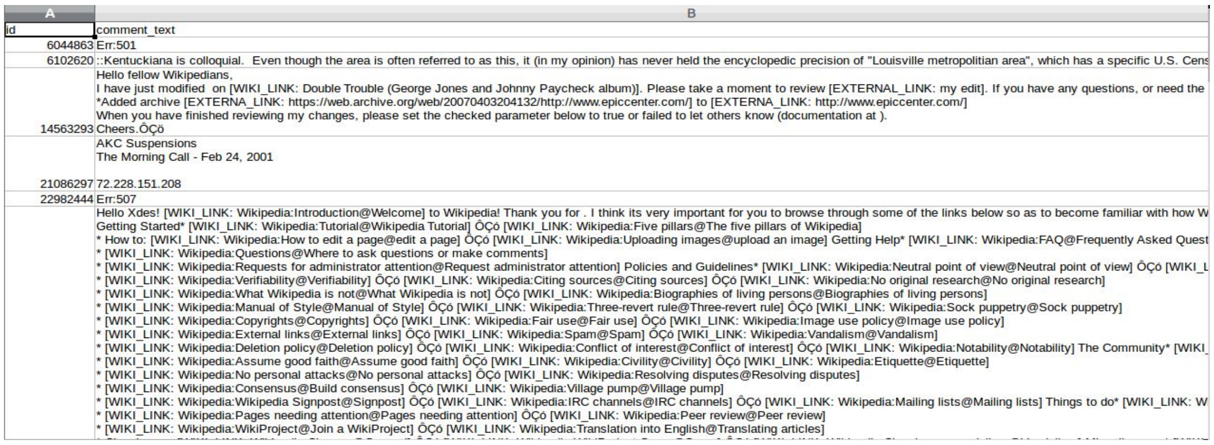
```
trained_model = featureExtraction( train_dataset )
trained_model = logisticRegression( trained_model )
for comments_fetched
    probability_values[] = trained_model( comments_fetched )
    toxicity_type = getName( max ( probability_values ) )
    print(toxicity_type)
```

Initially , a trained model is created by extracting the feature and performing logistic regression . Then , for each and every comments fetched , the model creates a set of probability values for each and every comment toxicity type . Then the correct toxicity type is selected by finding the maximum probability value from the set of probability values and determining its toxicity label . This label is considered as the final result .

CHAPTER 5
SYSTEM TESTING

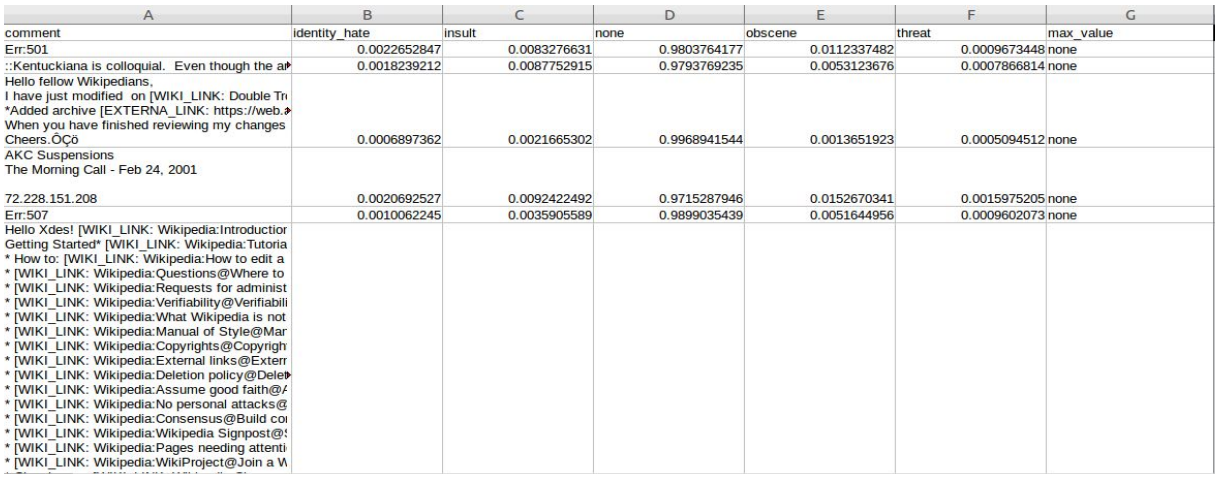
5.1 Testing the model with test dataset

The test dataset contains many number of comments in it . Unlike the training dataset , the test dataset contains only 2 columns , Comment Id and the comments . After running the model with test dataset , a resulting .csv file is produced which holds the probabilistic measure of each and every comment to the corresponding toxic label.



A	B
id	comment_text
6044863	Err:501
6102620	::Kentuckiana is colloquial. Even though the area is often referred to as this, it (in my opinion) has never held the encyclopedic precision of "Louisville metropolitan area", which has a specific U.S. Cens
14563293	Hello fellow Wikipedians, I have just modified on [WIKI_LINK: Double Trouble (George Jones and Johnny Paycheck album)]. Please take a moment to review [EXTERNAL_LINK: my edit]. If you have any questions, or need the *Added archive [EXTERNA_LINK: https://web.archive.org/web/20070403204132/http://www.epiccenter.com/] to [EXTERNA_LINK: http://www.epiccenter.com/] When you have finished reviewing my changes, please set the checked parameter below to true or failed to let others know (documentation at). Cheers.ÔÇö
21086297	AKC Suspensions The Morning Call - Feb 24, 2001 72.228.151.208
22982444	Err:507
	Hello Xdes! [WIKI_LINK: Wikipedia:Introduction@Welcome] to Wikipedia! Thank you for . I think its very important for you to browse through some of the links below so as to become familiar with how W Getting Started* [WIKI_LINK: Wikipedia:Tutorial@Wikipedia Tutorial] ÔÇó [WIKI_LINK: Wikipedia:Five pillars@The five pillars of Wikipedia] * How to: [WIKI_LINK: Wikipedia:How to edit a page@edit a page] ÔÇó [WIKI_LINK: Wikipedia:Uploading images@upload an image] Getting Help* [WIKI_LINK: Wikipedia:FAQ@Frequently Asked Quest * [WIKI_LINK: Wikipedia:Questions@Where to ask questions or make comments] * [WIKI_LINK: Wikipedia:Requests for administrator attention@Request administrator attention] Policies and Guidelines* [WIKI_LINK: Wikipedia:Neutral point of view@Neutral point of view] ÔÇó [WIKI_L * [WIKI_LINK: Wikipedia:Verifiability@Verifiability] ÔÇó [WIKI_LINK: Wikipedia:Citing sources@Citing sources] ÔÇó [WIKI_LINK: Wikipedia:No original research@No original research] * [WIKI_LINK: Wikipedia:What Wikipedia is not@What Wikipedia is not] ÔÇó [WIKI_LINK: Wikipedia:Biographies of living persons@Biographies of living persons] * [WIKI_LINK: Wikipedia:Manual of Style@Manual of Style] ÔÇó [WIKI_LINK: Wikipedia:Three-revert rule@Three-revert rule] ÔÇó [WIKI_LINK: Wikipedia:Sock puppetry@Sock puppetry] * [WIKI_LINK: Wikipedia:Copyrights@Copyrights] ÔÇó [WIKI_LINK: Wikipedia:Fair use@Fair use] ÔÇó [WIKI_LINK: Wikipedia:Image use policy@Image use policy] * [WIKI_LINK: Wikipedia:External links@External links] ÔÇó [WIKI_LINK: Wikipedia:Spam@Spam] ÔÇó [WIKI_LINK: Wikipedia:Vandalism@Vandalism] * [WIKI_LINK: Wikipedia:Deletion policy@Deletion policy] ÔÇó [WIKI_LINK: Wikipedia:Conflict of interest@Conflict of interest] ÔÇó [WIKI_LINK: Wikipedia:Notability@Notability] The Community* [WIKI_ * [WIKI_LINK: Wikipedia:Assume good faith@Assume good faith] ÔÇó [WIKI_LINK: Wikipedia:Civility@Civility] ÔÇó [WIKI_LINK: Wikipedia:Etiquette@Etiquette] * [WIKI_LINK: Wikipedia:No personal attacks@No personal attacks] ÔÇó [WIKI_LINK: Wikipedia:Resolving disputes@Resolving disputes] * [WIKI_LINK: Wikipedia:Consensus@Build consensus] ÔÇó [WIKI_LINK: Wikipedia:Village pump@Village pump] * [WIKI_LINK: Wikipedia:Wikipedia Signpost@Signpost] ÔÇó [WIKI_LINK: Wikipedia:IRC channels@IRC channels] ÔÇó [WIKI_LINK: Wikipedia:Mailing lists@Mailing lists] Things to do* [WIKI_LINK: W * [WIKI_LINK: Wikipedia:Pages needing attention@Pages needing attention] ÔÇó [WIKI_LINK: Wikipedia:Peer review@Peer review] * [WIKI_LINK: Wikipedia:WikiProject@Join a WikiProject] ÔÇó [WIKI_LINK: Wikipedia:Translation into English@Translating articles]

Figure 5.1 Screenshot snippet of Test dataset



A	B	C	D	E	F	G
comment	identity_hate	insult	none	obscene	threat	max_value
Err:501	0.0022652847	0.0083276631	0.9803764177	0.0112337482	0.0009673448	none
::Kentuckiana is colloquial. Even though the ar	0.0018239212	0.0087752915	0.9793769235	0.0053123676	0.0007866814	none
Hello fellow Wikipedians, I have just modified on [WIKI_LINK: Double Tr *Added archive [EXTERNA_LINK: https://web.a When you have finished reviewing my changes Cheers.ÔÇö	0.0006897362	0.0021665302	0.9968941544	0.0013651923	0.0005094512	none
AKC Suspensions The Morning Call - Feb 24, 2001 72.228.151.208	0.0020692527	0.0092422492	0.9715287946	0.0152670341	0.0015975205	none
Err:507	0.0010062245	0.0035905589	0.9899035439	0.0051644956	0.0009602073	none
Hello Xdes! [WIKI_LINK: Wikipedia:Introductior Getting Started* [WIKI_LINK: Wikipedia:Tutoria * How to: [WIKI_LINK: Wikipedia:How to edit a * [WIKI_LINK: Wikipedia:Questions@Where to * [WIKI_LINK: Wikipedia:Requests for administ * [WIKI_LINK: Wikipedia:Verifiability@Verifiabili * [WIKI_LINK: Wikipedia:What Wikipedia is not * [WIKI_LINK: Wikipedia:Manual of Style@Mar * [WIKI_LINK: Wikipedia:Copyrights@Copyrigh * [WIKI_LINK: Wikipedia:External links@Exterr * [WIKI_LINK: Wikipedia:Deletion policy@Delet * [WIKI_LINK: Wikipedia:Assume good faith@/ * [WIKI_LINK: Wikipedia:No personal attacks@ * [WIKI_LINK: Wikipedia:Consensus@Build coi * [WIKI_LINK: Wikipedia:Wikipedia Signpost@ * [WIKI_LINK: Wikipedia:Pages needing attenti * [WIKI_LINK: Wikipedia:WikiProject@Join a W						

Figure 5.2 Screenshot snippet of result file formed after processing test dataset with trained model

5.2 Testing the working of the message application

Here , the message application is tested to check whether the user’s are connected or disconnected successfully or not . Also the messages sent from one interface should be sent to other interface also.

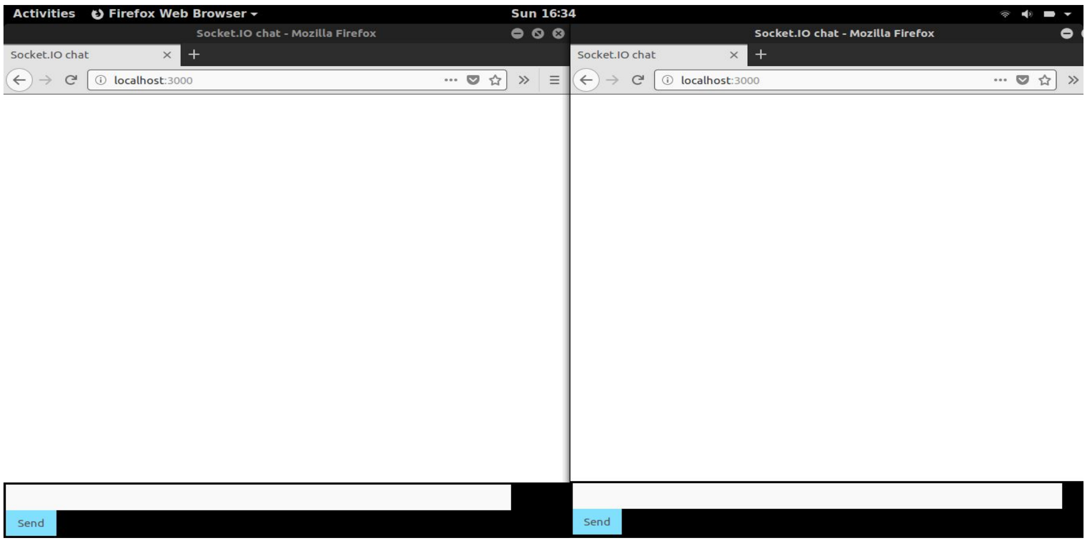
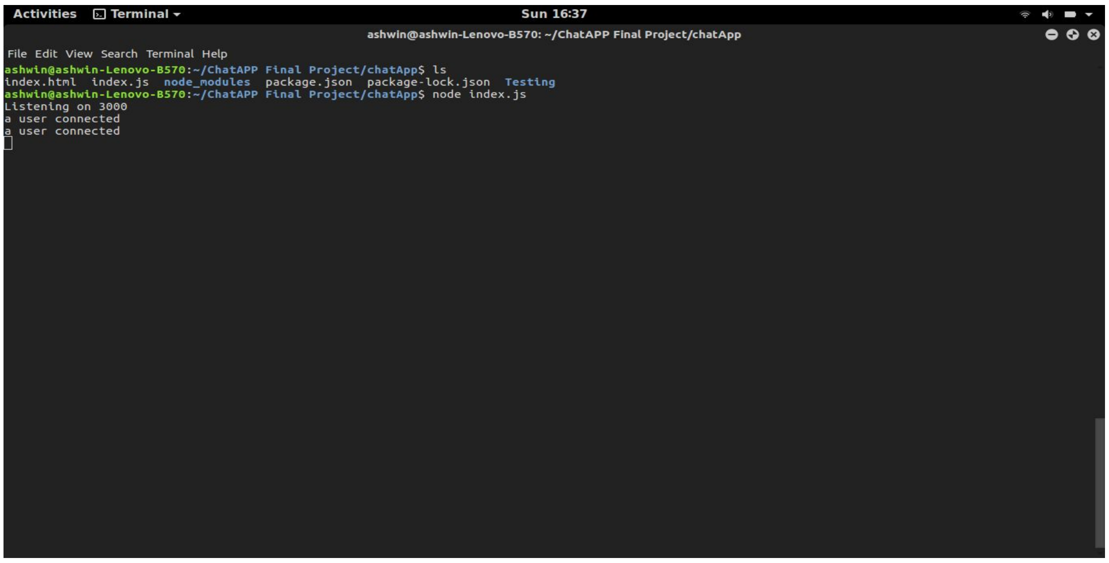


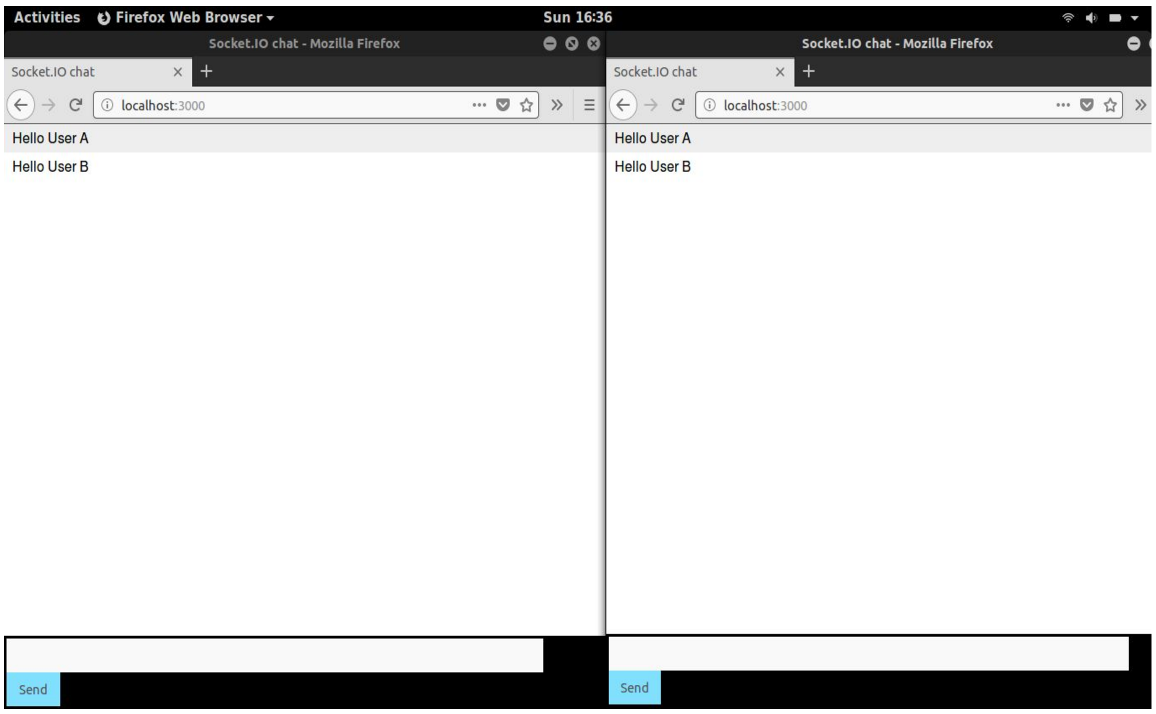
Figure 5.3 Two different Messaging UI is opened



Activities
Terminal
Sun 16:37
ashwin@ashwin-Lenovo-B570: ~/ChatAPP Final Project/chatApp
File Edit View Search Terminal Help

```
ashwin@ashwin-Lenovo-B570:~/ChatAPP Final Project/chatApp$ ls
index.html  index.js  node_modules  package.json  package-lock.json  Testing
ashwin@ashwin-Lenovo-B570:~/ChatAPP Final Project/chatApp$ node index.js
Listening on 3000
a user connected
a user connected
```

Figure 5.4 Console displaying number of users connected

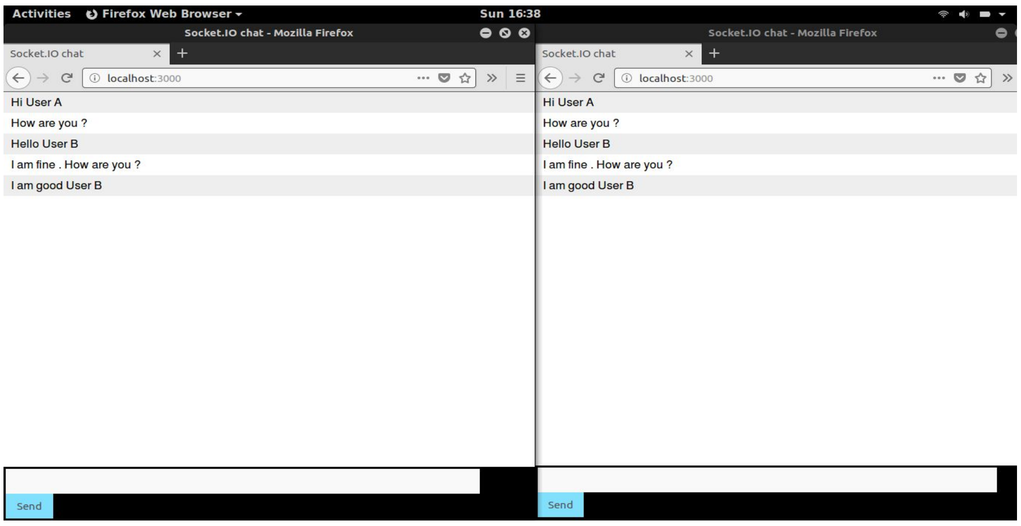


Hello User A
Hello User B

Figure 5.5 Messages transferred between two interfaces

5.3 Testing the application with non toxic messages

The application shows the conversation between two people . The conversation just consists of normal message without any toxic content in it . So the trained model analyses the messages and since it is not toxic , the messages are not labelled with any warning .

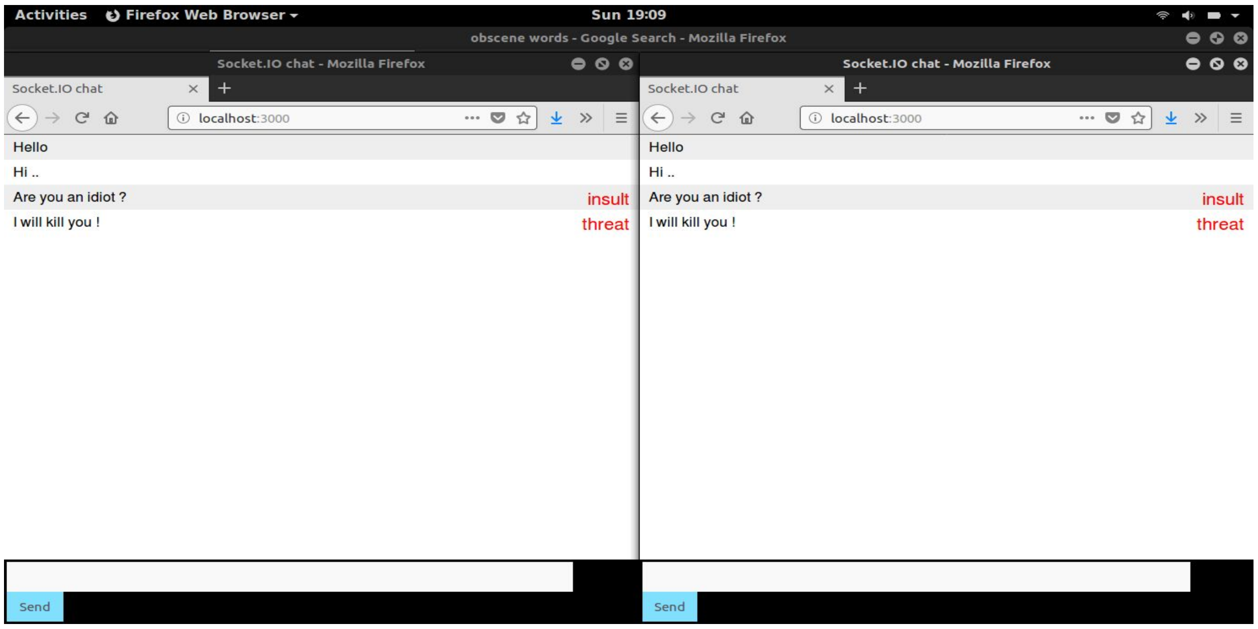


Hi User A
How are you ?
Hello User B
I am fine . How are you ?
I am good User B

Figure 5.6 Conversation with no toxic messages

5.4 Testing the application with both non toxic and toxic messages

Here , the application is tested with both toxic and non toxic messages . When each and every message is sent , the classifier script runs in the background and it classifies whether the message is toxic or not and displays the result to the right of the message. Empty if the message is not toxic or display the nature of the toxic message if the message is toxic


Activities
Firefox Web Browser
Sun 19:09
obscene words - Google Search - Mozilla Firefox
Socket.IO chat - Mozilla Firefox
Socket.IO chat
localhost:3000
Send

Hello
Hi ..
Are you an idiot ? insult
I will kill you ! threat

Hello
Hi ..
Are you an idiot ? insult
I will kill you ! threat

Figure 5.7 Conversation with both non toxic and toxic messages

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

Websites like Reddit , Stackoverflow , Facebook etc. deal with a lot of comments day by day . Since the conversations deal with opinions of different people , it is natural for a conversation to end up in a rude manner . Our project is involved around this issue . It looks to solve this problem by determining the toxic nature of the comment and classify it and remove the comment if needed . This could help a lot in the field data analytics performed on the websites. We have used various technologies like Logistic Regression , Vectorization and Machine Learning to label those comments which are toxic and also determine the nature of toxicity .

Further enhancement of the model could be improvised by taking consideration of classifying the toxic images posted over the internet . Since dealing with images need a more improvised classification processing , it would be more challenging than the classification of textual content and it would be more helpful for the websites which deal with more content in the form of images . Improvisation in the textual format can be done by incorporating the learning model with cross lingual datasets and training the model further to find the toxic comments which include other languages also .

REFERENCES

[1] Hossein Hosseini, Sreeram Kannan, Baosen Zhang and Radha Poovendran. Deceiving Google’s Perspective API Built for Detecting Toxic Comments . Network Security Lab (NSL), Department of Electrical Engineering, University of Washington, Seattle, WA

[2] Theodora Chu,Kylie Jue,Max Wang.Comment Abuse Classification with Deep Learning

[3] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge /data

[4] Mary Anne Franks.Sexual Harassment 2.0.University of Miami School of Law .Maryland Law Review, Vol. 71, 655 (2012). Date Written: 21 Oct 2009

[5] Uwe Bretschneider,Thomas Wöhner,Ralf Peters.Detecting Online Harassment in Social Networks . Completed Research Paper

[6] Duggan, Maeve. ”5 Facts about Online Harassment.” Pew Research Center. Pew, 30 Oct. 2014. Web. 20 Mar. 2017.

[7] Mehdad, Yashar, and Joel Tetreault. ”Do Characters Abuse More Than Words?.” Proceedings of the SIGdial 2016 Conference: The 17th Annual

Meeting of the Special Interest Group on Discourse and Dialogue. 2016.

[8] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification.. In AAAI, Vol. 333. 2267–2273

[9] Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. 69–78.

[10] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Characterlevel convolutional networks for text classification. In Advances in neural information processing systems. 649–657.

[11] Charu C Aggarwal and ChengXiang Zhai. 2012. A survey of text classification algorithms. In Mining text data. Springer, 163–222.

[12] Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., & Edwards, L. (2009). Detection of harassment on web 2.0. Proceedings of the Content Analysis in the WEB, 2, 1-7.

[13] Pooja Parekh and Hetal Patel . Toxic Comment Tools: A Case Study . International Journal of Advanced Research in Computer Science . Volume 8, No. 5, May-June 2017

[14] Razavi, A. H., Inkpen, D., Uritsky, S., &Matwin, S. (2010, May). Offensive language detection using multi-level classification. In Canadian Conference on Artificial Intelligence (pp. 16-27). Springer Berlin

Heidelberg.

[15]] Kansara, K. B., &Shekokar, N. M. (2015). A framework for cyberbullying detection in social network. International Journal of Current Engineering and Technology, 5.

[16] Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., &Bhamidipati, N. (2015, May). Hate speech detection with comment embeddings. In Proceedings of the 24th International Conference on World Wide Web (pp. 29-30). ACM.

[17] Rohan Shetty, Kalyani Nair, Shivani Singh, Shantanu Nakhare, GopalUpadhye (2015), A System To Detect Inappropriate Messages In Online Social Networks, International Journal of Advanced Computational Engineering and Networking, (40-43).