

A Project Report  
On  
**Sarcasm Detection for English Text**

Submitted in partial fulfilment of the requirement of University of Mumbai  
For the Degree of  
**Bachelor of Engineering**  
*in*  
**COMPUTER ENGINEERING**

*Submitted by*  
**Ms. Riya Das**  
**Ms. Shailey Kadam**  
**Mr. Chetan Kalra**  
**Ms. Vijeta Nayak**

*Supervisor*  
**Dr. Sharvari Govilkar**



**DEPARTMENT OF COMPUTER ENGINEERING**  
**PILLAI COLLEGE OF ENGINEERING**  
**NEW PANVEL - 410206**  
  
**UNIVERSITY OF MUMBAI**  
**Academic Year 2017-18**



PILLAI COLLEGE OF ENGINEERING  
NEW PANVEL - 410206  
DEPARTMENT OF COMPUTER ENGINEERING

## CERTIFICATE

This is to certify that the requirements for the project report entitled "**Sarcasm Detection For English Text**" have been successfully completed by the following project group students:

Group Member	Roll Number
Riya Das	CEA808
Shailey Kadam	CEA815
Chetan Kalra	CEA816
Vijeta Nayak	CEA830

in partial fulfilment of Bachelor of Engineering of Mumbai University in the Department of Computer Engineering, Pillai College of Engineering, New Panvel-410 206 during the Academic Year 2017-2018.

---

**Dr. Sharvari Govilkar**  
Supervisor  
Department of Computer Engineering

---

**Dr. Madhumita Chatterjee**  
Head,  
Department of Computer Engineering

---

**Dr. Sandeep M. Joshi**  
Principal  
PCE, New Panvel



PILLAI COLLEGE OF ENGINEERING  
NEW PANVEL - 410206  
DEPARTMENT OF COMPUTER ENGINEERING

## Project APPROVAL for B.E

This project entitled "Sarcasm Detection for English Text" by Riya Das, Shailey Kadam, Chetan Kalra, and Vijeta Nayak is approved for the degree of B.E. in Computer Engineering.

**Examiners:**

1. \_\_\_\_\_

2. \_\_\_\_\_

**Supervisors:**

1. \_\_\_\_\_

2. \_\_\_\_\_

**Chairman:**

1. \_\_\_\_\_

**Date:**

**Place:**

# Declaration

We declare that this written submission for B.E. project entitled "Sarcasm Detection for English Text" represent our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission have not been taken when needed.

## **Project Group Members:**

Riya Das & Sign : \_\_\_\_\_

Shailey Kadam & Sign : \_\_\_\_\_

Chetan Kalra & Sign : \_\_\_\_\_

Vijeta Nayak & Sign : \_\_\_\_\_

**Date:**

**Place:**

# Contents

<b>Abstract</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview of the project	1
1.2 Background of the project	1
1.3 Motivation	1
1.4 Significance of the project	1
1.5 Objective of the project	2
1.6 Scope of the project	2
1.7 Beneficiaries	2
<b>2 Literature Survey</b>	<b>3</b>
2.1 Literature Summary	6
2.2 Inferences of Literature Review	10
<b>3 Sarcasm Detector And Comparator</b>	<b>11</b>
3.1 Input Documents	12
3.2 Preprocessing Block	13
3.2.1 Filtering and Script Validation	13
3.2.2 Removing URL	13
3.2.3 Removing HTML Tags	14
3.2.4 Converting into Lower Case	14
3.3 Clean Dataset	14
3.4 Training Classifier	15
3.4.1 Classification	15
3.4.2 Supervised Learning	15
3.4.3 TF-IDF	16
3.4.4 Pipeline	16
3.5 Random Forest Classifier	17
3.5.1 Parameters	17
3.5.2 Train the Dataset	17
3.5.3 Random Forest Prediction	18
3.5.4 Advantages	18
3.6 Support Vector Machine (SVM)	19
3.6.1 Support Vectors	19
3.6.2 Hyperplane	19

3.6.3	CountVectorizer	20
3.6.4	SGDClassifier	20
3.6.5	GridSearchCV	20
3.6.6	Algorithm	20
3.6.7	Advantages	21
3.7	Naive Bayes Classifier	22
3.7.1	Multinomial Naive Bayes	22
3.7.2	Algorithm	22
3.7.3	Advantages	24
3.8	Examples	24
3.8.1	With Hashtags	24
3.8.2	With Emoticons	24
3.8.3	With more number of Punctuation Marks	25
3.8.4	With No special feature	25
4	Result and Discussion	26
4.1	Dataset and GUI	26
4.2	Performance Evaluation and Result Analysis	32
4.2.1	Performance Analysis	32
4.2.2	Result Analysis	37
4.2.3	Evaluation	38
5	Applications	40
5.1	Sentiment Analysis	40
5.1.1	Review Summarization	40
5.2	Public media analysis	40
5.3	Business Analytics	40
5.4	Literature Analysis	40
5.4.1	Analysis of Phrases	41
5.4.2	Chat and Email Conversations	41
6	Conclusion and Future Scope	42
	References	43
	Acknowledgement	44
	Publications	ix
	Certificates	xx

# Abstract

In recent years, with the increasing popularity of social media sites, people express themselves by communicating with each other in the form of texts. Without facial expression and vocal sounds it becomes very difficult to extract the exact meaning and intentions of the text. Sentiments, sarcasm and other elements present in spoken language are lost. So understanding the sentiments of the text becomes very important. Hence, one of the basic idea in sentimental analysis is to understand the sarcasm used in the statements.

Our project mainly focuses on the detection along with the comparative analysis of three machine learning algorithms such as Random Forest, Support Vector Machine and Naive Bayes Classifier.

Finally, from the results obtained after applying these algorithms on input data which comprises of emoticons, hashtags, punctuation marks the best approach for sarcasm detection can be selected.

# List of Figures

3.1	Sarcasm Detector.	12
3.2	Random Forest Classifier.	18
3.3	Support Vectors.	19
3.4	Support Vector Machine Classifier.	21
3.5	Naive Bayes Classifier.	23
4.1	Homepage for Sarcasm Detector.	27
4.2	Generation of Random Input String.	27
4.3	Preprocessing Block.	28
4.4	Removing HTML Tags.	28
4.5	Removing Unwanted URLs.	29
4.6	User provides Input.	29
4.7	Prediction of Randomly Generated Sentence.	30
4.8	Prediction of User Input Sentence.	30
4.9	Randomly Generated Sentence from Twitter.	31
4.10	Cleaning of Input Sentence.	31
4.11	Prediction of Input Sentence.	32
4.12	Confusion Matrix.	33
4.13	SVM Confusion Matrix.	34
4.14	Naive Bayes Confusion Matrix.	34
4.15	Random Forest Confusion Matrix.	35
4.16	Function of SVM.	36
4.17	Performance Analysis.	39
4.18	Accuracy Comparison.	39



# List of Tables

2.1	Summary of literature survey	6
4.1	Training Accuracy	36
4.2	Examples	37
4.3	Testing Accuracy	38

# Chapter 1

## Introduction

### 1.1 Overview of the project

Sarcasm is a mode of satirical wit depending for its effect on bitter, caustic, and often ironic language that is usually directed against an individual. Sarcasm is often used for the purpose of criticism and mockery. Sometimes sarcasm is an intended humour. It is not implicitly understood by most individuals.

Sarcasm detection is the need of an hour to understand the context in which a person is expressing his views. It helps to distinguish sentence into sarcastic and non-sarcastic sentences. Major conflicts that are caused due to misunderstanding of a written text can be avoided with help of our project i.e *Sarcasm Detection for English*.

### 1.2 Background of the project

Sarcasm detection is a one of the tasks of opinion mining. It is important for correct identification of user opinions expressed in written text. Understanding sarcastic phrases is sometimes difficult even for human individuals, thus a computational solution for this problem is a challenging task. A common approach for sarcasm identification would be Lexicon analysis but due to the overhead and time complexity we are using three machine learning algorithms.

### 1.3 Motivation

The pitch and the tone of the speaker used in a sentence help to perceive the context of the given text. For example "I had a great time with you in traffic." The sentence wanted to convey the message that, he never enjoyed the traffic but during speech with the help of tone and expression of a person, we can directly say that it is a negative message but in written text it is very difficult to understand whether it is sarcasm or not.

Thus, the premise of a sentence is best understood by speech but a written text creates a lot of misunderstanding and confusion. In order the perceive the actual sentiment behind the written text and avoiding the conflicts amongst people, sarcasm detection is of prior importance.

### 1.4 Significance of the project

With the rapid development of social media and the increasing craze of various TV series, use of Sarcasm has become more common. Besides this, use of Hashtags and emoticons have

rapidly been increasing. It has become a need of an hour for all the product based companies to understand the progress of their products in the market and among their clients. For this, Sarcasm detection plays an important role to judge the exact review obtained from the users and also to understand the hidden meaning in the posts and tweets on various Social media sites.

## **1.5 Objective of the project**

Sentiment analysis is one of the field in Natural Language Processing that deals with people's sentiments, attitude, and emotions from text. It is one of the most widely studied field in text mining. Sentiment analysis have many domains which needs to be analyzed such as consumer product reviews, review of any hotel, movie or social events. A common task in analyzing these domains is to classify the document or statement into positive or negative sentiments. There are many challenges in Sentiment Analysis and one of them is Sarcasm Detection.

Our objective is to use the concept of machine learning in order to train and test various sentences. Social Media is the most budding platform with a great global outreach and an important source for sentiment analysis in social media analytic.

## **1.6 Scope of the project**

The scope of the system is to do a comparative study and analysis of three machine learning algorithm such as Random Forest, Support Vector Machine and Naive Bayes Classifier. For analysis it is necessary to determine whether the given input sentence is sarcastic or not. For the input data, all tweets and reviews from various social media sites are considered which consists of hashtags, punctuation marks, emoticons etc..

The project mainly focuses on English text, therefore the most important process is to remove all other mixed languages present in the given statement. This is done by script validation and filtering of pre-processing block. Before training any dataset first step is to clean the data which is done by preprocessor block by removing all URL's, HTML tags and converting it into lower case. This clean data is then used to train classifier such as Random Forest, SVM and Naive Bayes Classifier. Depending upon the nature of data and classification algorithm used, the major chunk of data can be used for training and remaining for testing.

From the results obtained after comparison, the best approach to detect sarcasm can be selected to improve overall sentiments present in any sentences used for social media analytics.

## **1.7 Beneficiaries**

There is always a mystery whilst encountering any kind of review. People tend to use sarcasm just for the sake of mocking a person's work or criticizing his efforts. But, such review gives best result in the form of speech and becomes difficult to interpret by many of the viewers. Sarcasm detection will help to understand the essence of the sentence in its inherent form.

In many of the microblogging sites as well as chat applications people are accustomed to using emoticons and hashtags which may not be comprehended by many who are unaware about this trend and may misinterpret its meaning entirely. Sarcasm detection would decipher the meanings of the emoticons and also make life easier for the people who do not understand the modern trend of hash tags.

# Chapter 2

## Literature Survey

This chapter consists of the literature survey that we have conducted on the various systems employing understanding of text in order to find the actual sentiments of the sentences as sarcasm can change the actual meaning. The following chapter explains the systems that we surveyed.

Whiting, A. and D. Williams [1], proposed the paper which explored the uses and gratification that the consumer receive from social media. Based on the study of 25 interviews of people using gratifications, 10 uses and gratifications are listed with their usage are as follows:- Social interaction - 88%, Information seeking - 80%, Pass time - 76%, Entertainment - 64%, Relaxation - 60%, Expression of opinions - 56%, Communicatory utility - 56%, Convenience utility - 52%, Information sharing - 40% and Surveillance/knowledge about others - 20%. This application of uses and gratifications theory to social media not only proves to be rich and comprehensive understanding of the reasons of the consumers to utilize social media but it effectively contributes to the business and social media marketing and also helps in communicating with the potential customers by fulfilling their needs .

Hiroshi Shimodaira [10], classified the documents with its contents, and of the words of which they are composed of. Two document models - Bernoulli and Multinomial were used for the classification. The Zero Probability Problem is overcome by Laplace's law of succession or add one smoothing, that adds a count of one to each word type. Naive Bayes approximation can be used for document classification, by constructing distributions over words. The classifiers require a document model to estimate  $P(\text{document} \rightarrow \text{class})$ .

1. Bernoulli document model : a document is represented by a binary feature vector, whose elements indicate absence or presence of corresponding word in the document.

2. Multinomial document model : a document is represented by an integer feature vector, whose elements indicate frequency of corresponding word in the document.

B. Pang and L. Lee [7], stated the General challenges for opinion mining and Sentiment analysis which are: Contrasts with standard fact-based textual analysis, Factors that make opinion mining difficult. The mentioned Key Concepts are sentiment Polarity and degree of positivity, Subjectivity detection and opinion identification, Joint-topic sentiment analysis, Viewpoints and perspectives. Then the various features taken into considerations are Term Presence vs Frequency, Term-based features beyond term unigrams, Parts of speech, Syntax, Negation and Topic-oriented features. Afterwards Impact of labeled data is observed and obtained. Here, the unsupervised approach used are Unsupervised lexicon induction. The classification based on relationship information are: Relationships between sentences and between documents, Rela-

tionships between discourse participants, Relationships between product features, Relationships between classes and Incorporating discourse structure. Special considerations for extraction are: Identifying product features and opinions in reviews and Problems involving opinion holders. Basically, the aim to use all the techniques is achieved, but no conclusion either positive or negative can be made for the algorithms used.

Alec Go, Richa Bhayani and Lei Huang [5], proposed a different approach of Distant Supervision as they removed all emoticon and non-word tokens while training their algorithms. They found that removing the non-word tokens allowed the classifiers to focus on other features like classification. They used tweets ending in positive emoticons like :) :-)) as positive and negative emoticons like :( :-( as negative. They applied Naive Bayes, Maximum Entropy, and Support Vector Machine algorithms to classify Twitter sentiment which resulted to be in the range of 80% accuracy. They concluded that the unigram model outperforms all other models used, specifically bigrams and POS features do not help. Suggestion of using Maximum Entropy classifier was provided to obtain best result of 83% with both Unigrams and Bigrams during classification. They suggested that domain- specific tweets, handling neutral tweets, sentiment analysis in regional language and utilizing emoticon data in the test set must be considered to make proposed algorithm error resistance and with higher accuracy.

Luciano Barbosa and Junlan Feng [3], understood the usage of meta - information along with feature based model which gave description and information regarding hash tag, punctuation and emoticons. From the discussion, and observation an algorithm was designed which featured that for a given word in a tweet, they mapped these words to it's part-of-speech using a part-of-speech dictionary and opinion based messages contains adjectives or interjection. Along with this mapping, the word also mapped with its subjectivity. The algorithm used a two-step classification method : first training a classifier to distinguish between subjective and objective tweets and secondly training another classifier to differentiate between positive and negative sentiment. The accuracy rate was calculated on the popular list of words collected from various websites by comparing between various approaches such as ReviewSA, Unigrams, TwitterSA. These methods, reduced the error rate was from 46% to 23%. The main limitation of this approach was in the cases of sentences that contain antagonistic sentiments and web vocabulary.

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passonneau [2], discussed SVM with Unigram based, feature based and tree kernel based model. They also presented a method in which they used POS specific prior polarity features and explored the use of tree kernel to obviate the need of feature engineering. The designed used unigrams for sentiment classification. They collected data source from various websites and generated their sample sentences. The dataset used was trained with polarity and tweets contained emoticons and noisy labels. For evaluation of data collected they used a 5-fold cross validation model. For binary classification of sentence various techniques were developed which gave result as : For Only unigram classification it had result of 71.35%, Kernel Tree Technique gave an accuracy of 73.98%, Unigram + Senti-feature technique found to be 75.39% accurate, Kernel + senti-feature technique resulted into 74.61%. Same techniques were used for ternary classification which gave 56.58, 56.31, 60.6 and 60.50% accurate results. They therefore concluded that unigram+senti-feature gave result with maximum accuracy for binary classification. They investigated on two kinds of models : tree kernel and feature based models and demonstrate that both these models outperform the unigram baseline. Finally from the analysis it was stated that sentiment analysis for Twitter data is not that different from sentiment analysis for other genres.

Ashwin Rajadesingan and Reza Zafarani Arizona and Huan Liu Arizona [8], identify the traits using the users past tweets. SCUBA framework for Behavioral modeling Approach was being used to analyze the users past tweets and categorise it as: Sarcasm as a contrast of sentiments in which divisions were made based on Contrasting connotations and Contrasting present with the past, Sarcasm as a complex form of expression where readability was considered, Sarcasm as a means of conveying emotion in which emotions like mood, affect and sentiments and frustrations were observed, Sarcasm as a possible function of familiarity depending on familiarity of environment or language were scrutinized or Sarcasm as a form of written expression in which Prosodic variations or structural variations were being observed. Observations are, with no historical information, accuracy of 79.38%, is obtained. considerable gain (+4.14%) in performance is obtained by observing past 30 tweets. But including more past history still gives significant results. Results have derived that SCUBA is effective in detecting sarcastic tweets. SCUBAs main two advantages are considering psychological and behavioral aspects of sarcasm and leveraging users historical information to decide whether tweets are sarcastic or not. In future, SCUBA would detect sarcasm between strangers too.

Dmitry Davidov, Oren Tsur and Ari Rappoport [4], discussed a supervised classification framework that provided a way to utilize tagged data and emoticons for classification. It calculated the contribution of different feature types for sentiment classification and it was shown that the framework successfully identified sentiment types of untagged tweets. This quality was confirmed by human judges. They developed a methodology in which they used four basic feature types for sentiment classification : single word features, n-gram features, pattern features and punctuation features. All these feature types are combined into a single feature vector. They also used surface patterns to classify the words into into high frequency words and content words. For each feature vector construction they developed and used k-nearest neighbours (KNN) strategy for classification and with help of Euclidean distance to matching vectors were calculated. The Amazon Mechanical Turk (AMT) service was used to obtain a list of the most commonly used and unambiguous ASCII smileys to train and generate the data sets. They also discussed about algorithms that would help them to find dependencies and overlapping between different sentiment types represented by all smileys and hashtags.

N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iamnitchi [6], introduced a peer-to-peer service (Prometheus, a P2P service that enables socially-aware applications by providing decentralized, user-controlled social data management). To check its real time based constraints performance, collect or expose sensitive information from social media, mobile application were used in collaboration with Prometheus which ensures data integrity and security of the peers. Relations, labels, weights, and location are the ACPs mentioned. They emulated Prometheus the workload of two socially-aware applications and one social sensor based on previous system characterizations. The social-based mapping of users onto peers leads to significant improvements, especially for the 30 users/peer case. 15% of the invocations finishing faster when compared to the random case (some invocations can finish in half the time). Additionally, the benefits compound as the number of hops increases. The results show that a three-fold improvement in service availability can be achieved with minimum performance degradation, the average RTT is 200-300 msec and creating the trusted peer list can be an expensive operation.

Haruna Isah, Paul Trundle, Daniel Neagu [11], proposed a product safety framework using text mining and sentiment analysis. They utilised the framework to gather and analyse views and experiences of users of drug and cosmetic products. They also demonstrated how to develop

product safety lexicon and training data. Naive Bayes Classifier is used for implementation obtaining 83% of accuracy for twitter. Since, this research is work in progress yet can be used for users, product manufacturers, regulatory and enforcement agencies.

Hassan Saif, Yulan He and Harith Alani [9], proposed a novel approach of adding sentiment at each topic levels along with lexicon based pattern matching algorithm. For each extracted entity from tweets, the algorithm added the semantic concept as an additional feature and measured the correlation between the added concept and negative/positive sentiment. The model used techniques such as Unigram, POS, Sentiment at topic level and semantics. To detect these techniques used by them different data sets were generated. Unigram and POS techniques were used to find whether sentences are positive or negative as these sentences were gathered from Stanford Twitter Sentiment Corpus. These sentences when tested, gave accuracy around 71.5% and 75.53% respectively. Data collected from Health Care Reform were trained and tested using sentiment at topic level which evaluated to be 77.02%. OMD used n-fold cross validation to detect the semantics that evaluated to be 77.18%. All these techniques used were based on binary classification. To implement these techniques three different approaches were incorporating for the analysis; replacement, augmentation, and interpolation.

## 2.1 Literature Summary

A literature review is an objective, critical summary of published research literature relevant to a topic under consideration for research. The summary is presented in Table 2.1

Table 2.1: Summary of literature survey

SN	Year	Title of the paper and author	Research Paper Detail	Remarks
1	2015	Title: Text Classification using Naive Bayes Author: Hiroshi Shimodaira [10]	Techniques: 1. Naive Bayes approach 2. Bernoulli Model 3. Multinomial Model 4. Zero probability problem Datasets: Spam Emails	The classifiers require a document model to estimate $P(\text{document} - \text{class})$ .
2	2015	Title: Social Media Analysis for Product Safety Using Text Mining and Sentiment Analysis Author: Haruna Isah, Daniel Neagu and Paul Trundle [6]	Techniques: 1. Lexicon sentiment analysis 2. Naive Bayes for classification Datasets: Twitter, Avon, Dove and Oral B pages, 3 Brand Y products	83% accuracy for Naive Bayes classification for Twitter. It is used for users, product manufacturers, regulatory and enforcement agencies.

3	2015	Title: Sarcasm Detection on Twitter: A Behavioral Modeling Approach Author: Ashwin Rajadesingan, Reza Zafarani, and Huan Liu [8]	Techniques: SCUBA Framework Datasets: Twitter	Without any past history accuracy is 79.38%. With analysing past 30 tweets accuracy increases by 4.14%
4.	2013	Title: Why people use social media: a uses and gratifications approach Author: Anita Whiting, David Williams [1]	Techniques: 25 interviews of people. Datasets: Social Media (FB)	Uses and gratifications of FB are Social interaction.: - 88%, Information seeking.: - 80%, Pass time.: - 76%, Entertainment.: - 64% Relaxation.: - 60% Expression of opinions.: - 56% Communicatory utility.: - 56% Convenience utility.: - 52% Information sharing.: - 40% Surveillance/ knowledge about others.: - 20%
5.	2011	Title: Semantic Sentiment Analysis of Twitter Author: Hassan Saif, Yulan He and Harith Alani [9]	Techniques: Adding sentiment at each topic levels along with lexicon based pattern matching algorithm. Unigram, POS, Sentiment at topic level and semantics. Datasets: Stanford Twitter Sentiment Corpus, Healthcare Reform and OMD	Three approaches were incorporating for the analysis; replacement, augmentation, and interpolation.
6	2011	Title: Sentiment Analysis of Twitter Data Author: Apoorv Agarwal and Boyi Xie [2]	Techniques: SVM with Unigram based, feature based and tree kernel based model. POS specific prior polarity features. 5-fold cross validation model. Datasets: Twitter data	Unigram+senti-feature gave result with maximum accuracy for binary classification. Sentiment analysis for Twitter data is not that different from sentiment analysis for other genres.



7	2010	<p>Title: Robust Sentiment Detection on Twitter from Biased and Noisy Data</p> <p>Author: Luciano Barbosa and Junlan Feng</p> <p>[3]</p>	<p>Techniques: Two-step classification method : first training a classifier to distinguish between subjective and objective tweets and secondly training another classifier to differentiate between positive and negative sentiment</p> <p>Datasets: Twitter Dataset.</p>	<p>The main limitation of this approach was the sentences that contains antagonistic sentiments and web vocabulary.</p>
8	2010	<p>Title: Prometheus: User-Controlled P2P Social Data Management for Socially-Aware Applications</p> <p>Author: N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea and A. Iamnitchi</p> <p>[6]</p>	<p>Techniques: Prometheus, Mobile applications</p> <p>Datasets:</p> <ol style="list-style-type: none"> <li>1. Restaurants</li> <li>2. Spam emails</li> <li>3. Comments on blogs</li> <li>4. Ratings on user-generated content</li> </ol>	<p>Significant improvements, especially for the 30 users/peer case where 15% of the invocations finished faster.</p> <p>Three-fold improvement in service availability.</p> <p>The average RTT is 200-300 msec - creating the trusted peer list can be an expensive operation.</p>
9	2010	<p>Title: Enhanced Sentiment Learning Using Twitter Hashtags and Smileys</p> <p>Author: Dmitry Davidov, Oren Tsur and Ari Rappoport</p> <p>[4]</p>	<p>Techniques : Feature vector construction with k-nearest neighbours (KNN) strategy for classification and with help of Euclidean distance and matching vectors were calculated</p> <p>Datasets: Twitter, Amazon Mechanical Turk (AMT)</p>	<p>Supervised classification framework was used that provided a way to utilize tagged data and emoticons for classification.</p> <p>Dependencies and overlapping between different sentiment types represented by all smileys and hashtags.</p>

10	2009	<p>Title: Twitter Sentiment Classification using Distant Supervision Author: Alec Go, Richa Bhayani and Lei Huang [5]</p>	<p>Techniques : Naive Bayes, Maximum Entropy and Support Vector Machine algorithms Datasets: Tweets ending in positive emoticons like :) :- ) as positive and negative emoticons like :( :-( as negative</p>	<p>Maximum Entropy classifiers was provided to obtain best result of 83% with both Unigrams and Bigrams during classification.</p>
11	2008	<p>Title: Opinion Mining and Sentiment Analysis Author: B. Pang and L. Lee [7]</p>	<p>Techniques: 1. Sentiment polarity and degrees of positivity 2. Subjectivity detection and opinion identification 3. Joint topic-sentiment analysis 4. Viewpoints and perspectives Unsupervised Approach: 1. Unsupervised lexicon induction 2. Classification based on relationship information 3. Relationships between sentences and between documents 4. Relationships between discourse participants 5. Relationships between product features 6. Relationships between classes 7. Incorporating discourse structure Special considerations for extraction: 1. Identifying product features and opinions in reviews: 2. Problems involving opinion holders Datasets: e-bay, e-books, Yahoos Cornell movie Reviews, Blog06, Amazon, Cnet, TREC Blog</p>	<p>No specific Conclusion can be drawn i.e either positive or negative. The work is still under progress.</p>

## 2.2 Inferences of Literature Review

- In 2008, unsupervised approach was used wherein concepts like lexicon inductions and relationship information which was proposed by B. pang and L. Lee.
- Then in 2009, techniques like Naive Bayes, Maximum Entropy and Support Vector Machine algorithms proposed by Alec Go, Richa Bhayani and Lei Huang were used.
- In 2010, initially, two step classification method was used, then a software - Prometheus was used along with the mobile application which is socially aware and provides security and data integrity too.  
The other paper used supervised classification framework which constructed feature vector with k-nearest neighbors strategy.
- In 2011, Along with feature based approach and machine learning based approach(SVM), tree kernel was used for Polarity and feature selections.
- In 2013, The technique used was interviewing 25 people and then noting their interest and reasons for using social media.
- In 2015 various models were made to get good approximations and accuracy to detect sarcasm not only in twitter datasets but also in other social media. For eg:- Bernoulli model, multinomial, zero probability, SCUBA along with behavioural approach to detect the presence of Contrast in the sentiments, Complex form of expression in the given statements.
- So we can conclude that sarcasm can be determined with a lexicon based approach, but it would take more time for computation. While if we can obtain the features and store it in a file, we can reuse the same featured for determining sarcasm any number of times without actually performing all the processes.  
Therefore, our project mainly focuses on machine learning approach as it is a better way to obtain whether sentences are sarcastic or not in order to increase its result and accuracy.

# Chapter 3

## Sarcasm Detector And Comparator

In this chapter we would be discussing about the system architecture. The input of the system would be reviews or simply some content from various Social Media Sites and tweets from twitter. The first step is to clean the raw input so that a standardized format of content is obtained. From the cleaned data obtained we have constructed our dataset which is used in the training phase to train the various machine learning classifiers.

The system mainly focuses on English text, therefore it is necessary to check whether the content is in English text only. This is done by script validation and filtering of preprocessing block. Next step is to remove all the URLs present in the data, unwanted HTML tags and converting it into lower case.

This cleaned data is then converted into standard format i.e data matrix with reviews and labels. Labels is of two types 0 and 1 indicating the sentence being not sarcastic and sarcastic respectively. Depending upon the nature of data and classification algorithm used, the major chunk of data can be used for training and remaining for testing. Ideally for each classification 70-30% is used for our classifier.

Training data consists of hashtags, emoticons, punctuation marks and too positive and too negative sentences, therefore there is no need to handle them separately.

The system uses three supervised machine learning algorithms, such as **Random Forest**, **Support Vector Machine (SVM)** and **Naive Bayes Classifier** to train and test the dataset. It also uses K-fold Validation to define a data for testing the model in training phase. This helps in minimizing problems like over-fitting. With help of this validation, it gives us the idea to design the system on how the model will generalize any independent dataset.

In training phase the algorithm builds a classifier by analysing the training data and associated label with each class and creates a pickle file which consists of all the features extracted by the model in training phase. From the data model created, a confusion matrix is generated which help us to find true positives, true negative, false positive and false negative obtain during the training phase.

During testing, the system accepts the input from the user and compares with the features stored in the pickle file and predict whether given input sentence is sarcastic or not.

The main aim of the system is to compare these algorithms to find which algorithm can be further used to detect sarcasm during text analytics.

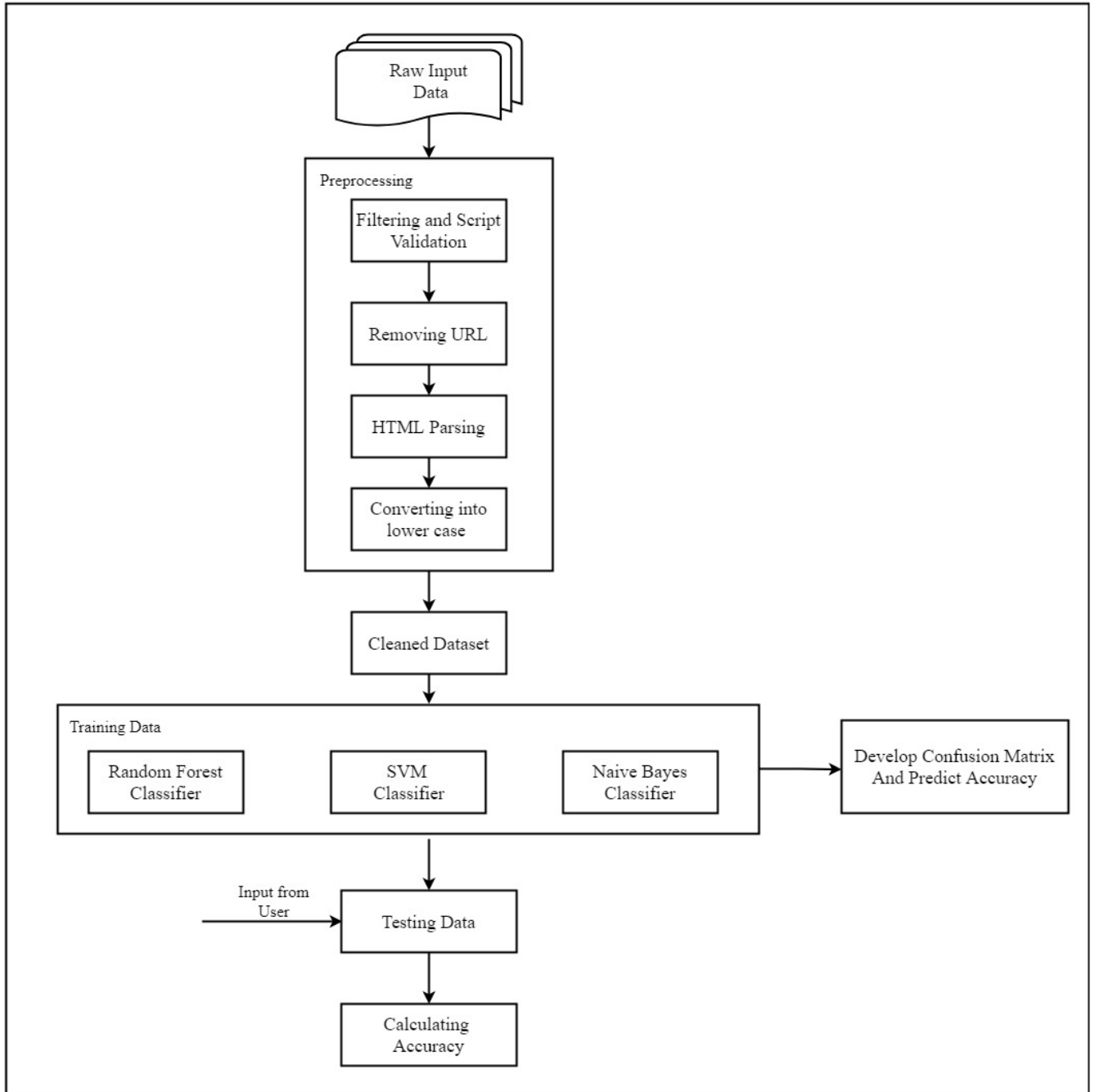


Figure 3.1: Sarcasm Detector.

### 3.1 Input Documents

The text will be in Romanized English format. The content would be collected from different social media domains like Twitter, etc. or from product based websites like Amazon, etc..

## 3.2 Preprocessing Block

### 3.2.1 Filtering and Script Validation

The process of considering only English text by ignoring all the mixed language text so that processing of text can be made easier.

**Algorithm :**

- a. Input : Mixed Language.
- b. Output : Romanized English Language.
- c. Steps :
  - i. START.
  - ii. Use the character set as UTF-8
  - iii. Scan the sentence character by character.
  - iv. Compare each character from scanned input with UTF-8.
  - v. If character is present in the UTF-8, then it does not belong to English Script and display it is not a valid English statement otherwise it is valid.
  - vi. Ignore all the special characters.
  - vii. Repeat the above process for all the sentences.

### 3.2.2 Removing URL

The process of removing all unwanted text such as URL so that more informative data can be stored in the dataset for training.

**Algorithm :**

- a. Input : The sentences only containing English Text and special characters like hashtags, emojis, punctuation marks, etc..
- b. Output : URL present in the sentence are removed.
- c. Steps :
  - i. START.
  - ii. Define a regular expression to identify the presence of `https://www.abc.com`
  - iii. Scan the input document.
  - iv. Check for not End of file.
    1. Read a character from input file.
    2. IF character matches with regular expression then remove it.
    3. Display the text after removing text otherwise go to step 4.
    4. Read the next input sentence.
    5. STOP.

### 3.2.3 Removing HTML Tags

The process of removing all unwanted text such as HTML tags so that more informative data can be stored in the dataset for training.

**Algorithm :**

- a. Input : Sentences with no URLs.
- b. Output : Sentences without any HTML tags
- c. Steps :
  - i. START.
  - ii. Identify all predefined HTML Tags by using predefined packages.
  - iii. If the sentences contain any html Tags then remove it and display it otherwise go to next step.
  - iv. Read the next input sentence.
  - v. Presence of HTML tags can be compared by comparing the input and output string of this block.
  - vi. Repeat the same process until end of document is found.
  - vii. STOP.

### 3.2.4 Converting into Lower Case

This block converts the input string into one standardized format which is in lower case.

**Algorithm :**

- a. Input : Sentences without URL and HTML Tags.
- b. Output : Sentences are in lower case
- c. Steps :
  - i. START.
  - ii. Read the input document character by character.
  - iii. Check for end of document.
    1. Apply lower() function to each character.
    2. Display the content in lower case and go to next step.
    3. Read next character.
    4. Stop.

## 3.3 Clean Dataset

This block contains dataset free from all unwanted URL, HTML tags and converted into Lower Case. Stop words are not removed during pre processing as it might contain some sentiments that would affect its meaning. In this blocks labels are assigned to each sentences and stored into standardized format i.e review and its corresponding label.

- a. Input : Sentences free from URL and HTML Tags.
- b. Output : Dataset formatted into data matrix.
- c. Steps :
  - i. START.
  - ii. Scan the input document.
  - iii. Assign labels to each class. Here we make use of 1 and 0 to indicate whether sentences are sarcastic or not respectively.
  - iv. Repeat the process till end of the document.
  - v. STOP.

## 3.4 Training Classifier

### 3.4.1 Classification

Data Classification is termed as the process that organizes data into categories so that it can be used efficiently and effectively. It basically has two phases :

- i. **Training Phase** : At this phase, the classification algorithm uses the training data for analysing.
- ii. **Testing Phase** : In this phase, testing data are used to estimate the accuracy of the classifier. Testing data is the dataset used for evaluating the model in the training phase.

Based upon the data chunk the dataset is divided for training and testing. Ideally we used 70-30% to train and test data respectively.

### 3.4.2 Supervised Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training dataset. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). In case of our system we have two pairs reviews and labels.

A supervised learning algorithm analyses the training data and from the results it produces an inferred function, which can be used for testing new data samples.

**Steps :**

- i. Determine which kind of domain we are analysing. In our system for sarcasm detection we have considered all reviews, contents from social media websites, tweets from twitter, etc..
- ii. Gather the training data set. We have collected dataset for training from github, Kaggle, etc..
- iii. Determine the input feature for training classifier. Accuracy depends upon the feature extracted during training process.
- iv. Determine the structure of the learned function and corresponding learning algorithm. For example, our system uses Random Forest, SVM and Naive Bayes Classifier.



- v. Complete the training process.
- vi. Evaluate confusion matrix to determine accuracy for particular training algorithm.

### 3.4.3 TF-IDF

The TF (term frequency) of a word is the frequency of a word (i.e. number of times it appears) in a document.

The IDF (inverse document frequency) of a word is the measure of how significant that term is in the whole corpus.

If a word appears frequently in a document, then it can be concluded to be the most important word and we need to give that word a high score. But if a word appears in too many other documents, then it might be not a unique identifier, therefore we must assign a lower score to that word. The math formula for this measure :

$$tfidf(t, d, D) = tf(t, d) * idf(t, D) \quad (3.1)$$

Where t denotes the terms; d denotes each document; D denotes the collection of documents.

#### Algorithm

- i. Assign lower score to the stop words (frequent words).
- ii. Search for words with higher search volumes and lower competition.
- iii. The content should have words which are unique and relevant to the user.

#### Term Frequency (TF)

The first part of the formula  $tf(t,d)$  indicates to calculate the number of times each word appeared in each document.

#### Inverse Document Frequency (IDF)

Inverse document frequency is the factor which is implemented to diminish the weight of terms that occur very frequently in the document set and increase the weight of terms that occur rarely.

$$idf(t, D) = \log \frac{|D|}{1 + |d \in D : t \in d|} \quad (3.2)$$

The numerator : D is referring to our document space. It can also be seen as  $D = d_1, d_2, \dots, d_n$  where n is the number of documents in the collection.

### 3.4.4 Pipeline

The machine learning algorithm usually takes clean (and often tabular) data, and learns some pattern from the data, to make predictions on new data(test data/evaluating data).

So the entire framework of converting the raw data to clean that can be used by machine learning algorithm to train an algorithm, and finally from features extracted using the algorithm we perform some function to solve certain problem is the called as pipeline. It is termed as a pipeline because it is analogous to physical pipelines because it sequentially solves all the features or parameters specified.

## 3.5 Random Forest Classifier

Random forest algorithm is a one of the supervised learning classification algorithm. Random forest classifier creates a set of decision trees from randomly selected subset of training set. In general, the more trees in the forest the more lively the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. It is basically based on rules.

### 3.5.1 Parameters

- i. TfidfVectorizer : Convert a collection of raw documents to a matrix of TF-IDF features.
  - a. lowercase : This parameter converts all characters to lowercase before tokenizing.
  - b. encoding : If bytes or files are given to analyze, this encoding is used to decode.
- ii. n-estimators : This parameter gives the number of trees that has been developed in the forest.
- iii. max-features : This field gives the number of features that must be consider when looking for the best split. Types are : auto, sqrt, log2.
- iv. n-jobs : The number of jobs to run in parallel for both fit and predict. If -1, then the number of jobs is set to the number of cores. For laptop we use 1 but for our system we used 2 cores for parallel processing.
- v. verbose : This field controls the verbosity of the tree building process.

Algorithm for Random Forest can be divided into two phases :

- i. To train the dataset and extract features.
- ii. To perform prediction from the created random forest classifier.

### 3.5.2 Train the Dataset

**Algorithm :**

1. Select each sentences from the document and extract "m" features.
2. Randomly select "i" features from total "m" features.
3. Among the "i" features, calculate the node "d" using the best split point.
4. Split the node into daughter nodes using the best split.
5. Build forest by repeating above for "n" number times to create "n" number of trees.

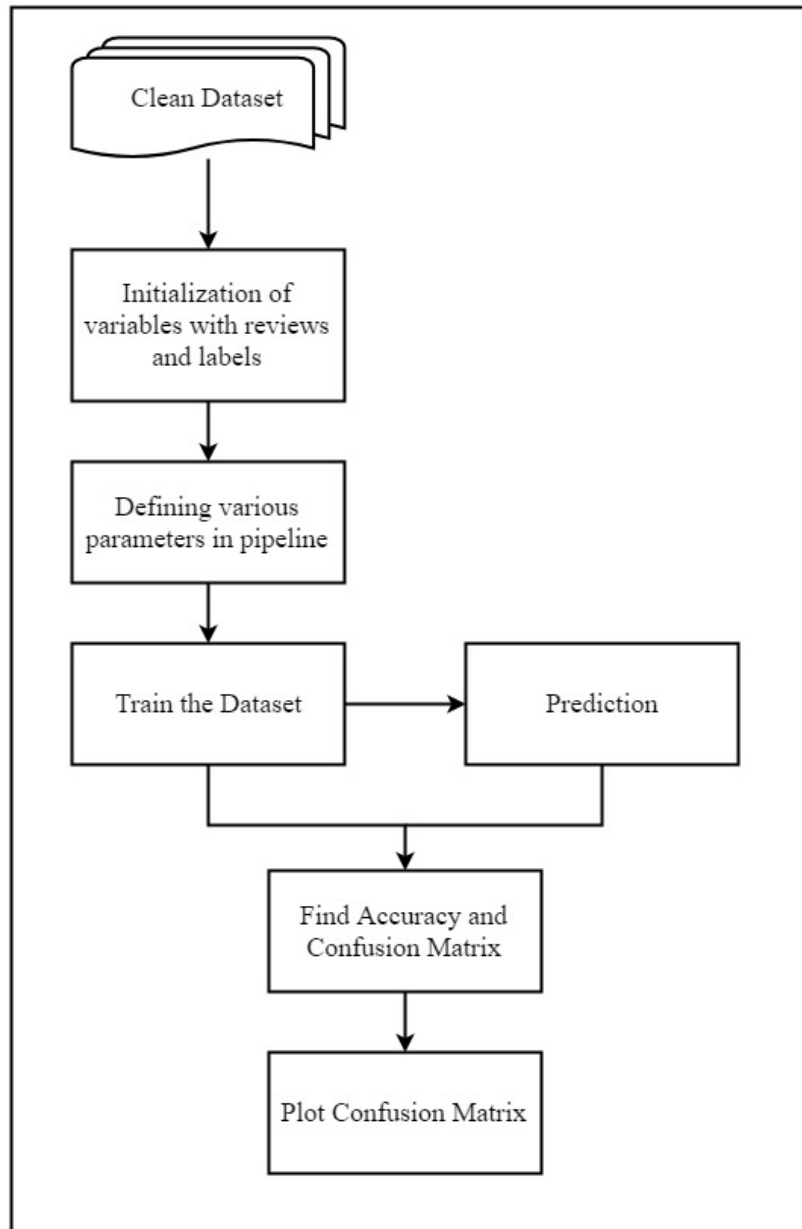


Figure 3.2: Random Forest Classifier.

### 3.5.3 Random Forest Prediction

**Algorithm :**

- i. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)
- ii. Calculate the votes for each predicted target.
- iii. Consider the high voted predicted target as the final prediction from the random forest algorithm.

### 3.5.4 Advantages

- i. The random forest classifier can use for both classification and the regression task.

- ii. It will handle the missing values.
- iii. When we have more number of trees in the forest, this classifier won't over-fit the model.

## 3.6 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is one of the supervised machine learning algorithm that can be used for both classification and regression purposes. It is mainly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features extracted) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that best divides a dataset into two classes, as shown in the image below.

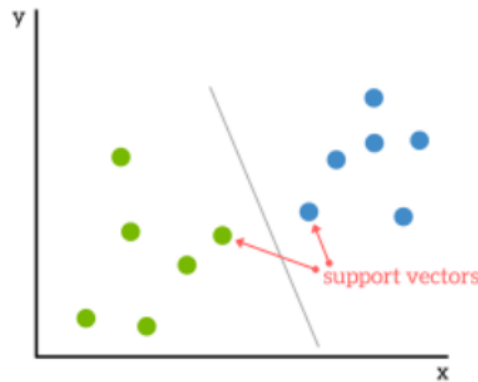


Figure 3.3: Support Vectors.

### 3.6.1 Support Vectors

Support vectors are the data points nearest to the hyperplane. If the points are removed then it will alter the position of the dividing hyperplane. Due to this support vectors are considered to be the critical elements of a data set.

### 3.6.2 Hyperplane

Hyperplane Can be generalized as :

- i. In one dimension, an hyperplane is called a **point**.
- ii. in two dimensions, it is a **line**.
- iii. in three dimensions, it is a **plane**.
- iv. in more dimensions you can call it an **hyperplane**.

If the data point lies further from the hyperplane constructed then it can be classified correctly. Therefore, our main aim is to keep the data point as far as possible from the hyperplane but it should be on correct side of classification. So when new testing data is added, whatever side of the hyperplane it lands, it will decide the class that is assigned to it.

To segregate two classes in the given data efficiently, then the distance between the hyperplane

and the nearest data point from either set known as the margin should be less.

The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, which gives a greater chance of new data being classified correctly.

### 3.6.3 CountVectorizer

It converts a collection of text documents to a matrix of token counts. This implementation produces a sparse representation of the counts.

#### Parameters.

- i. analyzer : Whether the feature should be made of word or character n-grams. If a callable is passed it is used to extract the sequence of features out of the raw, unprocessed input.
- ii. input : File name passed as argument to fit then it is expected to be a list of file names that need reading to fetch the raw content to analyze.

### 3.6.4 SGDClassifier

SGD stands for Stochastic Gradient Descent : The gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule. It is example of Linear model fitted by minimizing a regularized empirical loss.

#### Parameters

- i. loss : The hinge loss is a margin loss used by standard linear SVM models.
- ii. alpha : Constant that multiplies the regularization term. It chooses the best accuracy from the given range and displays it.
- iii. max-iter : This field gives details regarding the maximum number of passes over the training data.
- iv. random-state : This gives information regarding the seed of the pseudo random number generator to use when shuffling the data.

### 3.6.5 GridSearchCV

Using GridSearchCv is not necessary. If it is not used we need to loop the parameters and run all the combination of parameters. For this we need to write the code manually which increases the time requirements. Hence for during training of our system by SVM Classifier we used GridSearchCV.

### 3.6.6 Algorithm

- i. Defining various parameters using SGDClassifier.
- ii. Use GridSearchCV to iterate the parameters automatically.
- iii. Train the classifier based upon parameters defined.
- iv. Make predictions of data from training dataset.

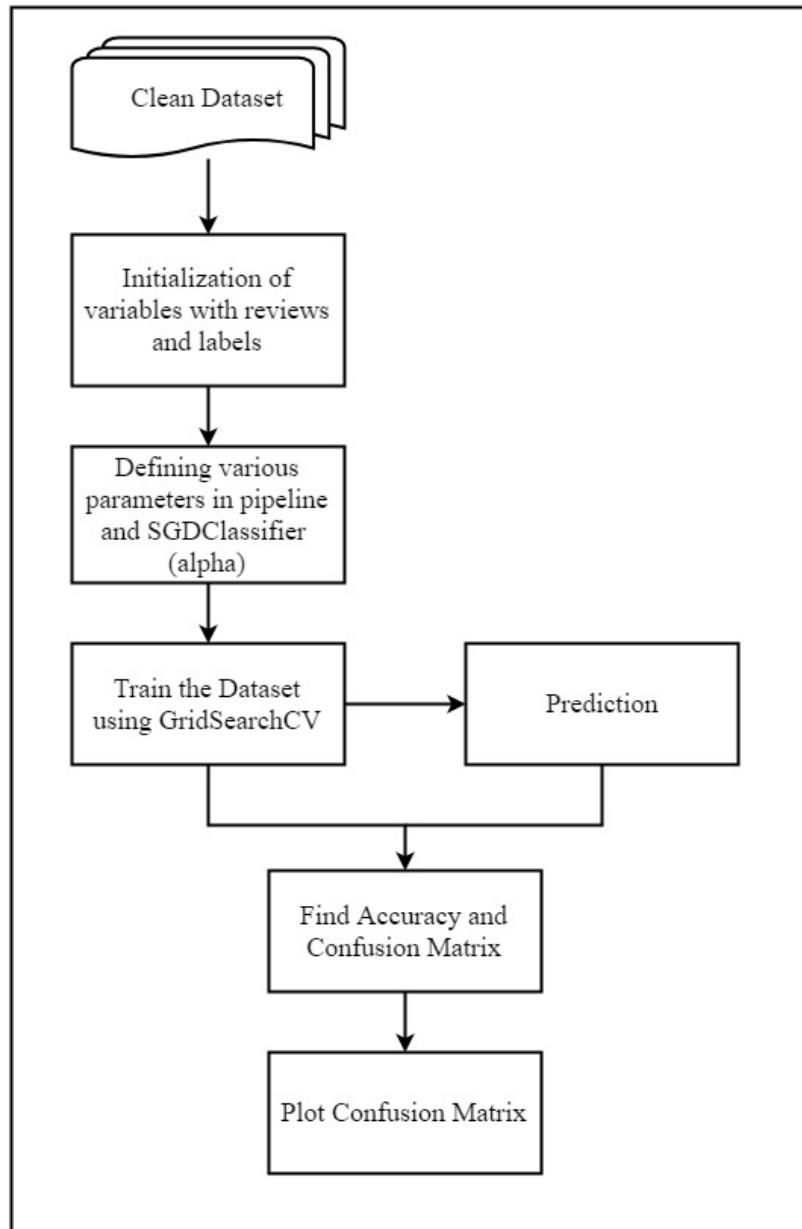


Figure 3.4: Support Vector Machine Classifier.

- v. Find accuracy and confusion matrix for training and testing dataset.
- vi. Plot confusion matrix.

### 3.6.7 Advantages

- i. It works really well with clear margin of separation.
- ii. It is effective in high dimensional spaces.
- iii. It is useful where number of dimensions is greater than the number of samples.
- iv. SVM uses a subset of training points in the decision function (called support vectors), hence it can also be termed as memory efficient.

## 3.7 Naive Bayes Classifier

Naive Bayes Classifier is based on the Bayesian theorem. It is suitable where the dimensionality of the input attributes is high. In this model, parameter estimation is done by using maximum likelihood. It is used to find conditional probabilities.

$P(X|Y)$  is the conditional probability of event X occurring for the event Y which has already been occurred.

$$P(X|Y) = P(X \text{ and } Y) / P(Y) \quad (3.3)$$

Naive Bayes classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

- i.  $P(c|x)$  is the posterior probability of class (target) given predictor (attribute).
- ii.  $P(c)$  is the prior probability of class.
- iii.  $P(x|c)$  is the likelihood which is the probability of predictor given class.
- iv.  $P(x)$  is the prior probability of predictor.

### 3.7.1 Multinomial Naive Bayes

For our system we have implemented MultinomialNB which makes use of the Naive Bayes algorithm for multinomially distributed data. It is used when the multiple occurrences of the words matter a lot in the classification problem. The parameters  $\theta_y$  is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\theta_y = (N_y i + \alpha) / (N_y + \alpha_n) \quad (3.4)$$

where  $N_{yi} = \sum_{x \in T} x_i$  is the number of times feature i appears in a sample of class y in the training set T, and  $N_y = \sum_{i=1}^{|T|} N_{yi}$  is the total count of all features for class y.

#### Parameters

- i. TfidfVectorizer : Convert a collection of raw documents to a matrix of TF-IDF features.
- ii. fit\_prior : Whether to learn class prior probabilities or not. If false, a uniform prior will be used.
- iii. alpha : This parameter is known as a hyper parameter i.e. a parameter that controls the form of the model itself.

### 3.7.2 Algorithm

- i Define parameters using TfidfVectorizer and MultinomialNB.
- ii Training the classifier is as follows :
  - a. D : set of tuples.
  - b. Each tuple is an 'n' dimensional attribute.  $X : (x_1, x_2, x_3, \dots, x_n)$ .

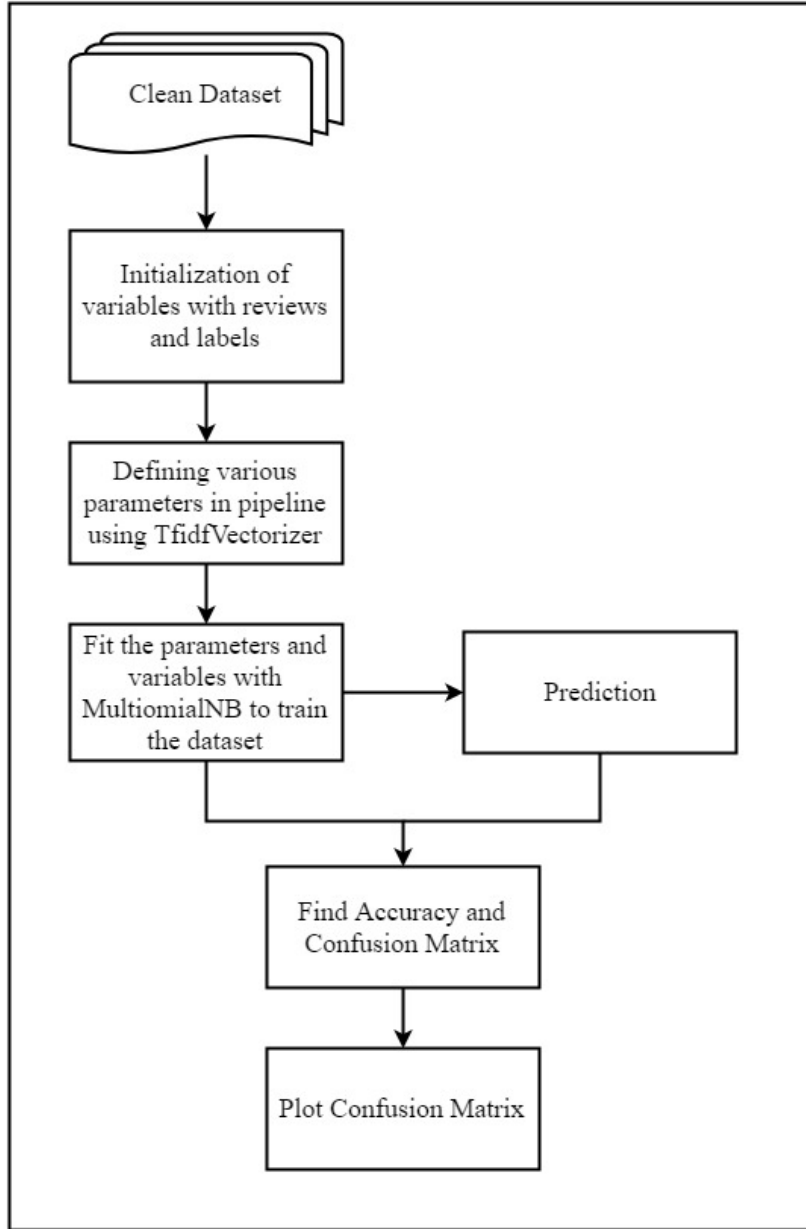


Figure 3.5: Naive Bayes Classifier.

c. Let there be 'm' classes :  $C_1, C_2, C_3, \dots, C_m$ .

d. Naive Bayes classifier predicts X belongs to Class  $C_i$  if :

$$P(C_i/X) > P(C_j/X) \text{ for } 1 \leq j \leq m, j \neq i \quad (3.5)$$

Maximum posteriori hypothesis gives :

$$P(C_i/X) = P(X/C_i) * P(C_i)/P(X) \quad (3.6)$$

e. Maximum  $P(X/C_i)$  as  $P(X)$  is a constant. with many attributes, it is computationally expensive to evaluate  $P(X/C_j)$ .

$$P(X/C_i) = P(x_1/C_i) * P(x_2/C_i) * \dots * P(x_n/C_i) \quad (3.7)$$

The classifier predicts that the class label of tuple X is the class  $C_i$  if and only if:

$$P(X/C_i) * P(C_i) > P(X/C_j) \text{ for } 1 \leq j \leq m \text{ and } j \neq i \quad (3.8)$$



- iii Make predictions of data from training dataset.
- iv Find accuracy and confusion matrix for training and testing dataset.
- v Plot confusion matrix.

### 3.7.3 Advantages

- i. Fast to train and classify.
- ii. Handles each data points accurately.
- iii. Not conscious to irrelevant features.

## 3.8 Examples

### 3.8.1 With Hashtags

Example 1 : Life is so great # sarcasticTweet

#### Steps

- i. Train classifier to extract features.
- ii. The extracted features is stored into pickle file.
- iii. Take the input sentence and extract features from the sentence.
- iv. Compare the features extracted from sentence and the stored feature.
- v. Since, the input sentence contains sarcastic hashtag, So, the output for the given input sentence is sarcastic.

### 3.8.2 With Emoticons

Example 2 : You are so beautiful that I dont want to look at you. :P

#### Steps

- i. Train classifier to extract features.
- ii. The extracted features is stored into pickle file.
- iii. Take the input sentence and extract features from the sentence.
- iv. Compare the features extracted from sentence and the stored feature.
- v. Since, the input sentence contains sarcastic emoji. So, the output for the given input sentence is sarcastic.

### 3.8.3 With more number of Punctuation Marks

Example 3 : What a lovely day it is !!!!

#### Steps

- i. Train classifier to extract features.
- ii. The extracted features is stored into pickle file.
- iii. Take the input sentence and extract features from the sentence.
- iv. Compare the features extracted from sentence and the stored feature.
- v. Since, the input sentence contains lot of punctuation marks which indicates too positive sentence. So, the output for the given input sentence is sarcastic.

### 3.8.4 With No special feature

- a. Sarcasm Present.

Example 4 : Your speech is so interesting that I dont want to hear it.

#### Steps

- i. Train classifier to extract features.
- ii. The extracted features is stored into pickle file.
- iii. Take the input sentence and extract features from the sentence.
- iv. Compare the features extracted from sentence and the stored feature.
- v. Since, the input sentence indicates contrast. So, the output for the given input sentence is sarcastic.

- b. Sarcasm not Present.

Example 5 : You are looking beautiful.

#### Steps

- i. Train classifier to extract features.
- ii. The extracted features is stored into pickle file.
- iii. Take the input sentence and extract features from the sentence.
- iv. Compare the features extracted from sentence and the stored feature.
- v. Since, the input sentence shows no contrast as well as does not match with any features extracted. So, the output for the given input sentence is not sarcastic.

# Chapter 4

## Result and Discussion

### 4.1 Dataset and GUI

Training dataset is generated by cleaning the raw data collected from various social media sites like Amazon, etc. and tweets from twitter. Cleaning of dataset is done by pre-processing block to get data in standardized format which is in the form of reviews and labels. Different classifier models are trained using this cleaned dataset to extract features and store it into a pickle file.

For the evaluation of our system, we have used 10,000 sentences for each type to test the classifier model. The system extracts the features from the input sentence and compare it with the features stored in pickle file to detect whether the given input sentence is sarcastic or not. The accuracy of the system is based upon the confusion matrix generated for each classifier models and the evaluation of test sentences.

The in-depth analysis of the outputs and their performance evaluation has been described in detail in the following sections. Our GUI is basically a web page which consists of all the phases discussed below :

Figure (4.1) depicts the first page of our system. Here, the user can enter the input sentence in the text area or randomly generate a string and can also collect the live tweets from twitter by specifying a keyword to test the system.

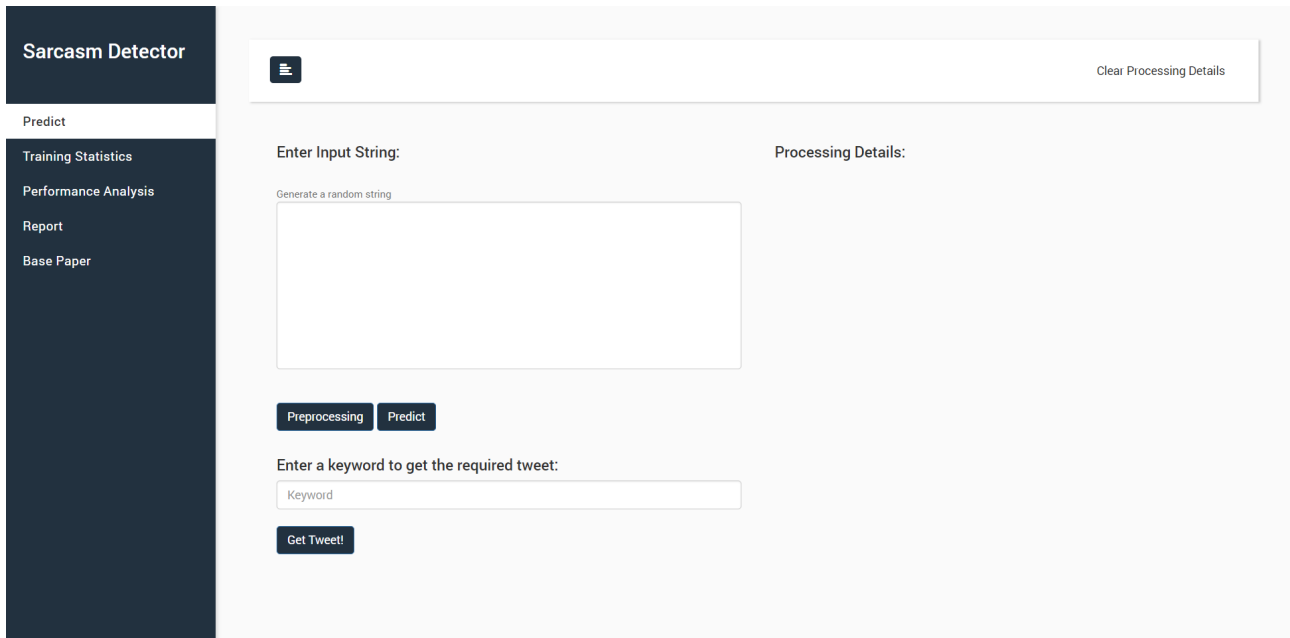


Figure 4.1: Homepage for Sarcasm Detector.

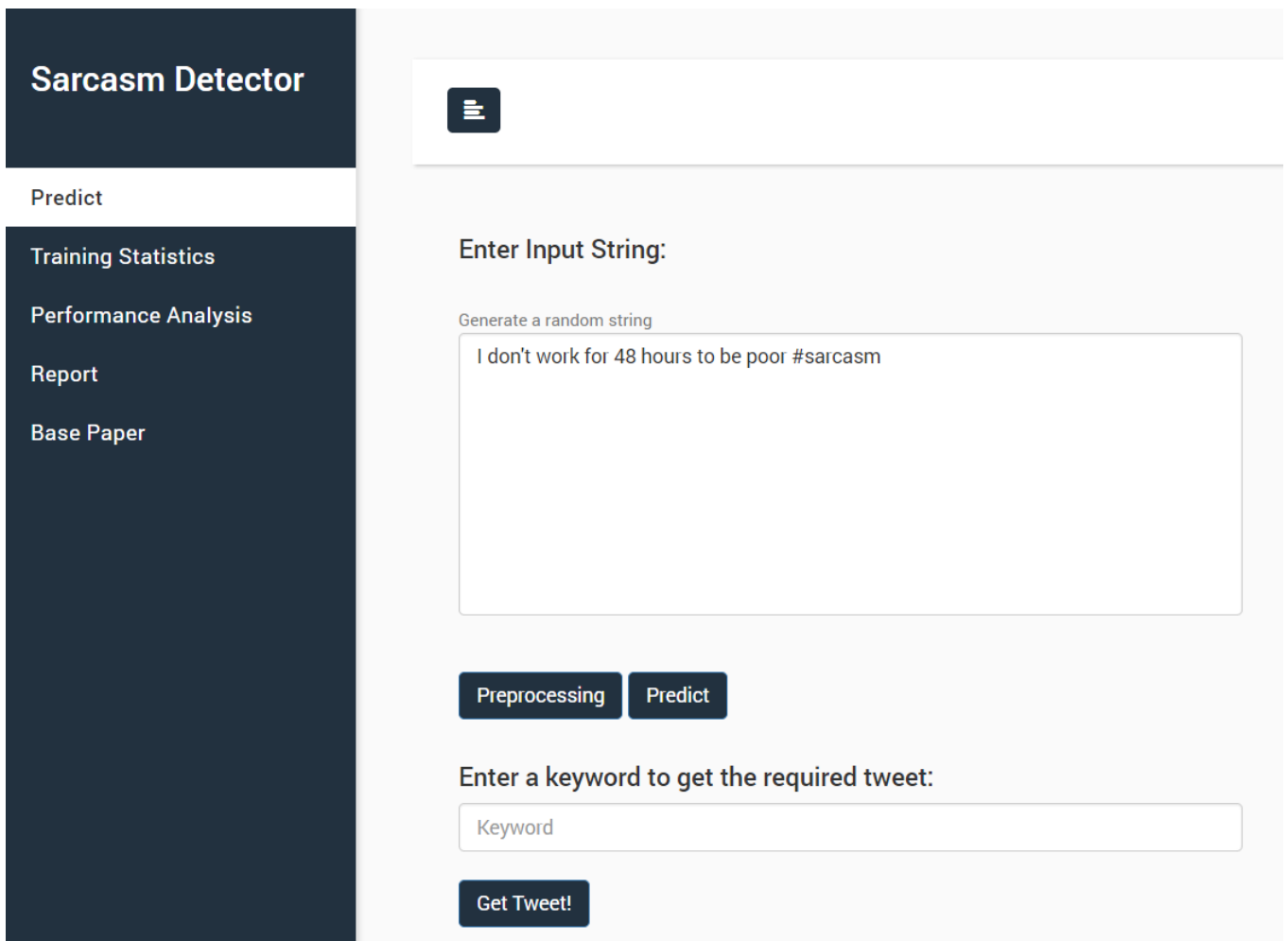


Figure 4.2: Generation of Random Input String.

In above figure (4.2) the user can get a random sentence from the testing file by clicking

on *Generate a Random string*.

The screenshot shows a web interface for text preprocessing. On the left, under 'Enter Input String:', there is a text area containing 'I don't work for 48 hours to be poor #sarcasm' and a button labeled 'Generate a random string'. Below this are 'Preprocessing' and 'Predict' buttons. Further down is a 'Keyword' input field and a 'Get Tweet!' button. On the right, under 'Processing Details:', the status is 'Starting...'. The 'Preprocessing Stage' is active, showing three steps: 1. Removing HTML Tags (No Tag Found), 2. Removing URLs (No Link Found), and 3. Converting string to lowercase (i don't work for 48 hours to be poor #sarcasm).

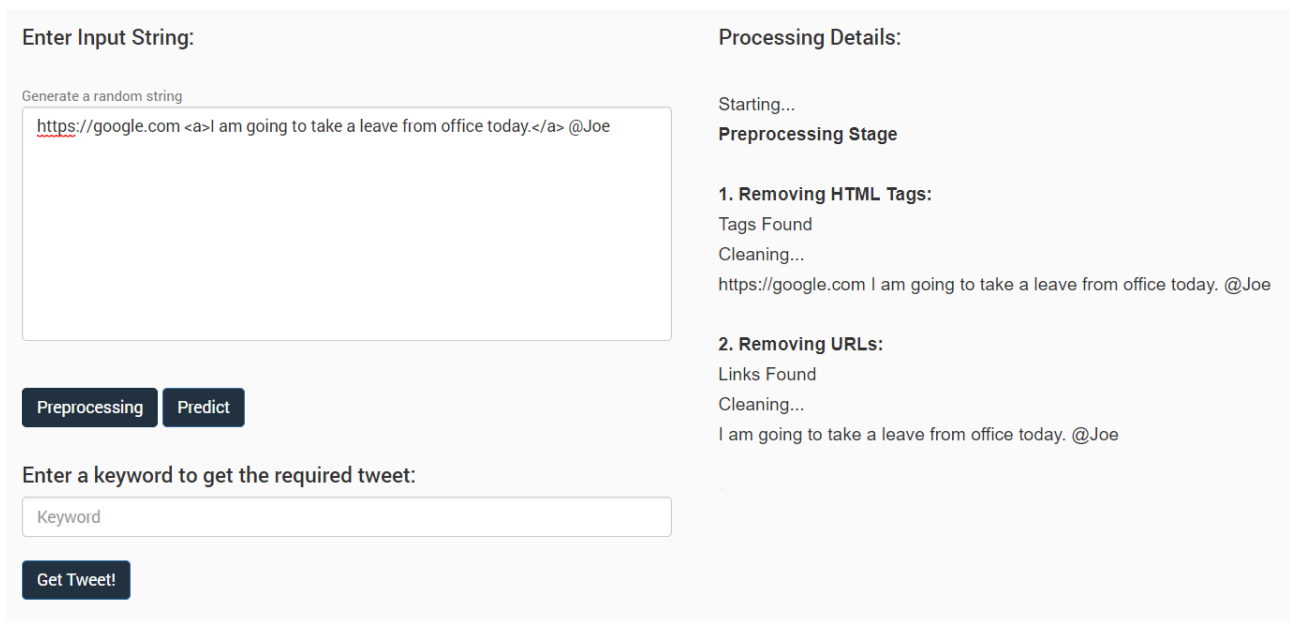
Figure 4.3: Preprocessing Block.

In figure (4.3), when user clicks on *Preproceesing* block, it will perform few processes like removing URLs, unwanted HTML tags and convert it into lower case for each input sentence.

This screenshot shows the same interface as Figure 4.3 but with different input and processing details. The input string is 'https://google.com <a>I am going to take a leave from office today.</a> @Joe'. The 'Preprocessing' button is highlighted. The 'Processing Details' section shows the 'Preprocessing Stage' with step 1, 'Removing HTML Tags', indicating 'Tags Found' and 'Cleaning...'. The resulting string is 'https://google.com I am going to take a leave from office today. @Joe'.

Figure 4.4: Removing HTML Tags.

When user clicks on *Preprocessing* block of figure (4.4), the first step the system implements is to remove the unwanted HTML tags to make input sentence more meaningful.



Enter Input String:

Generate a random string

<https://google.com> <a>I am going to take a leave from office today.</a> @Joe

Preprocessing Predict

Enter a keyword to get the required tweet:

Keyword

Get Tweet!

Processing Details:

Starting...

**Preprocessing Stage**

**1. Removing HTML Tags:**

Tags Found

Cleaning...

<https://google.com> I am going to take a leave from office today. @Joe

**2. Removing URLs:**

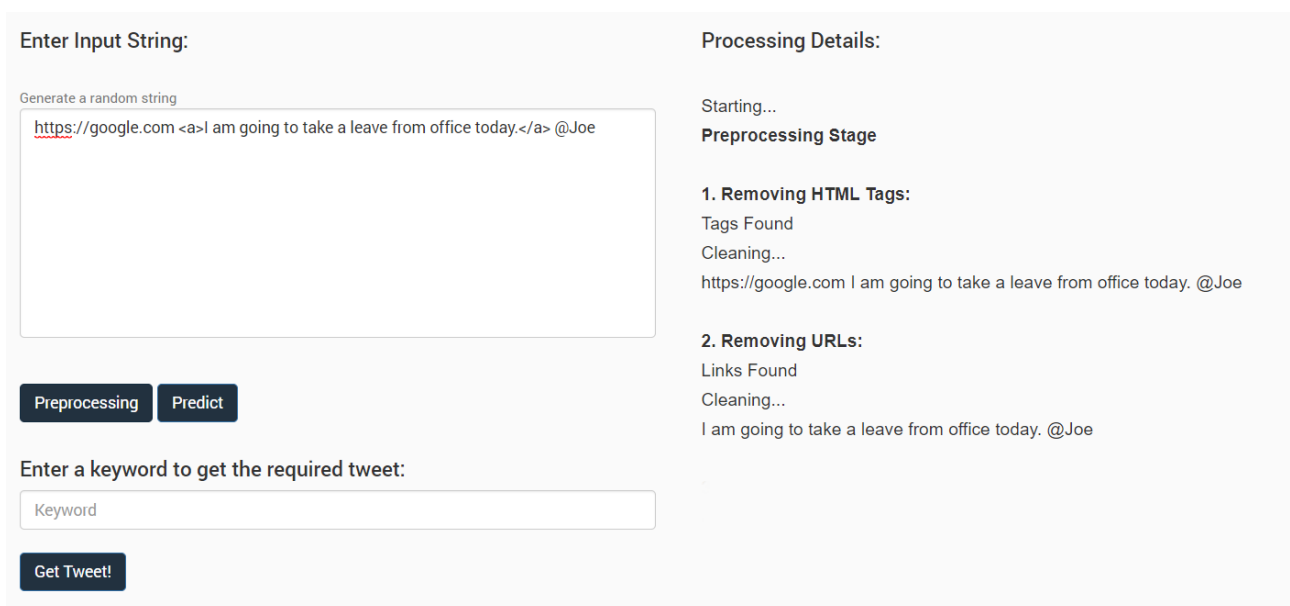
Links Found

Cleaning...

I am going to take a leave from office today. @Joe

Figure 4.5: Removing Unwanted URLs.

When user clicks on *Preprocessing* block of figure (4.5), the second step the system implements is to remove the unwanted URLs to make input sentence readable for prediction.



Enter Input String:

Generate a random string

<https://google.com> <a>I am going to take a leave from office today.</a> @Joe

Preprocessing Predict

Enter a keyword to get the required tweet:

Keyword

Get Tweet!

Processing Details:

Starting...

**Preprocessing Stage**

**1. Removing HTML Tags:**

Tags Found

Cleaning...

<https://google.com> I am going to take a leave from office today. @Joe

**2. Removing URLs:**

Links Found

Cleaning...

I am going to take a leave from office today. @Joe

Figure 4.6: User provides Input.

In figure (4.6), user enters the input sentence for which the system needs to predict the presence of sarcasm. When the user clicks on *Preprocessing* block, it will clean the input sentence based upon unwanted URLs, removing HTML tags and converting it into lower case. The output of preprocessing block will be used during prediction phase.

Enter Input String:

Generate a random string

I don't work for 48 hours to be poor #sarcasm

Preprocessing

Predict

Enter a keyword to get the required tweet:

Keyword

Get Tweet!

Processing Details:

Starting...

Preprocessing Stage

1. Removing HTML Tags:

No Tag Found

I don't work for 48 hours to be poor #sarcasm

2. Removing URLs:

No Link Found

I don't work for 48 hours to be poor #sarcasm

3. Converting string to lowercase:

i don't work for 48 hours to be poor #sarcasm

Predictions:

1. Naive Bayes

Sarcastic

2. SVM

Sarcastic

3. Random Forest

Sarcastic

Figure 4.7: Prediction of Randomly Generated Sentence.

In figure (4.7), when user clicks on *Predict* button, the system will then extract the features from the input sentence and compare with the features stored in the pickle file to give whether the randomly generated sentence by the system is sarcastic or not sarcastic.

Enter Input String:

Generate a random string

<https://google.com> <a>I am leaving now</a> @Joe

Preprocessing

Predict

Enter a keyword to get the required tweet:

Keyword

Get Tweet!

Processing Details:

Starting...

Preprocessing Stage

1. Removing HTML Tags:

Tags Found

Cleaning...

<https://google.com> I am leaving now @Joe

2. Removing URLs:

Links Found

Cleaning...

I am leaving now @Joe

3. Converting string to lowercase:

i am leaving now @joe

Predictions:

1. Naive Bayes

Not Sarcastic

2. SVM

Not Sarcastic

3. Random Forest

Not Sarcastic

Figure 4.8: Prediction of User Input Sentence.

In figure (4.8), when user clicks on *Predict* button, the system will then extract the features from the input sentence and compare with the features stored in the pickle file to give whether the user entered sentence is sarcastic or not sarcastic.

Figure 4.9: Randomly Generated Sentence from Twitter.

In figure (4.9), when user enters a keyword in *Keyword* block and *Get Tweet* button is clicked, the system randomly selects the recent tweet from Twitter and displays it in the text area for which the system needs to perform the prediction.

Figure 4.10: Cleaning of Input Sentence.

In figure (4.10), when user clicks on *Preprocessing* button the system performs few process like removing URLs, unwanted HTML tags and convert it into lower case for input sentence.



Enter Input String:

Generate a random string

"If we do not lead for democracy, for prosperity, and for human rights around the world, who will?" – Secretary-Designate Mike Pompeo This was one of my favorite statements from yesterday's hearing. At we embrace our duty to lead. <https://t.co/bScCxxVqST>

Preprocessing
Predict

Enter a keyword to get the required tweet:

Human

Get Tweet!

the world, who will?" – Secretary-Designate Mike Pompeo This was one of my favorite statements from yesterday's hearing. At we embrace our duty to lead. <https://t.co/bScCxxVqST>

2. Removing URLs:

Links Found
Cleaning...

"If we do not lead for democracy, for prosperity, and for human rights around the world, who will?" – Secretary-Designate Mike Pompeo This was one of my favorite statements from yesterday's hearing. At we embrace our duty to lead.

3. Converting string to lowercase:

"if we do not lead for democracy, for prosperity, and for human rights around the world, who will?" – secretary-designate mike pompeo this was one of my favorite statements from yesterday's hearing. at we embrace our duty to lead.

Predictions:

1. Naive Bayes
Not Sarcastic

2. SVM
Not Sarcastic

3. Random Forest
Not Sarcastic

Figure 4.11: Prediction of Input Sentence.

In figure (4.11), when user clicks on *Predict* button, the system will then extract the features from the input sentence and compares with the features stored in the pickle file to give whether the randomly generated sentence by the system is sarcastic or not sarcastic.

## 4.2 Performance Evaluation and Result Analysis

The quality of our system can be obtained by calculating the efficiency of the system. There is no particular method to calculate such an efficiency. The efficiency of our system is based on the confusion matrix generated after training the classifier and number of correct output given by each classifier for input sentence during testing.

### 4.2.1 Performance Analysis

#### Confusion Matrix

Summarizing the performance of a classification algorithm and placing them together is termed as confusion matrix. Calculating confusion matrix help us to find which classification model is better and what type of errors the classification model is making, which is affecting the overall performance of the system.

In figure (4.12), the matrix which is formed during training phase help us to understand the number of false positives, false negatives, true positive and true negatives.

## Confusion Matrix:

		predicted	
		p'	n'
actual	p	true positives	false negatives ← Type II error
	n	false positives ↑ Type I error	true negatives

Figure 4.12: Confusion Matrix.

### The Training Phase

Our Sarcasm detector detects sarcasm based on three Machine Learning Languages i.e. *Random Forest*, *SVM* and *Naive Bayes* classifier. It not only detects sarcasm but tell us which is the best algorithm to detect it. This detector has given us the following results for Sarcasm Detection. Once the training phase is done, We then find the *Confusion Matrix* for each algorithm. The confusion Matrix gives the number of false positives and false negatives as well as true positives and too negatives. This confusion matrix is the plotted in the form of a graph for a better understanding.

Figure (4.13), gives summarization of SVM classifier model when clicked on *Training Statistics* and *SVM*.

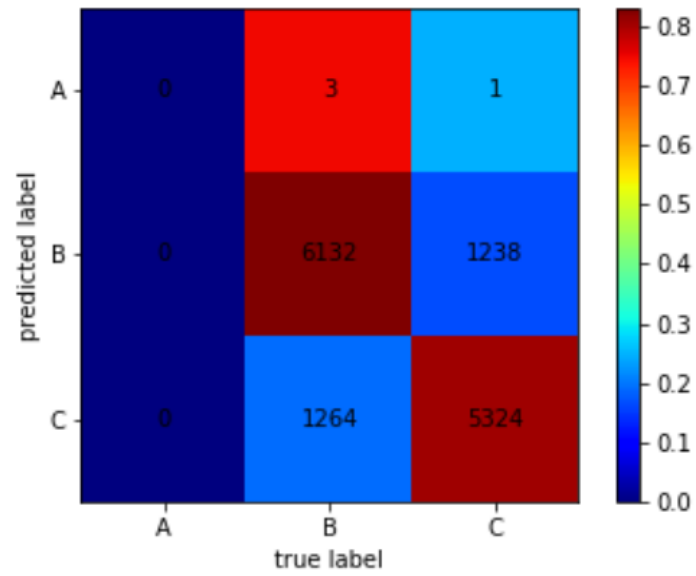


Figure 4.13: SVM Confusion Matrix.

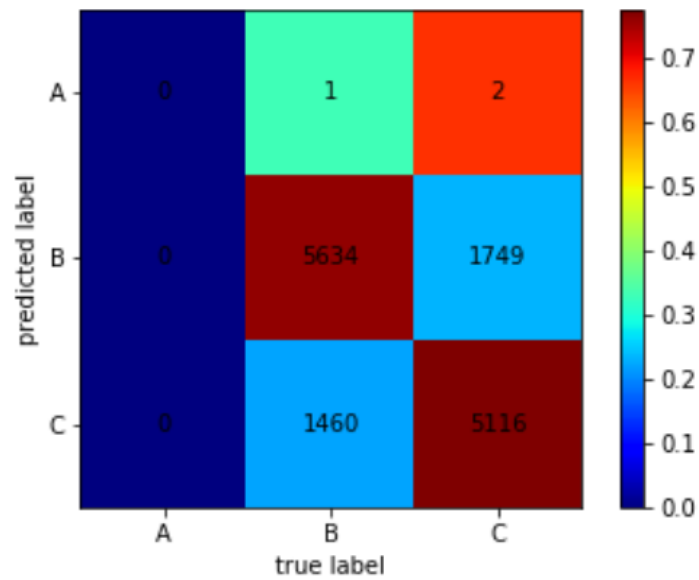


Figure 4.14: Naive Bayes Confusion Matrix.

Figure (4.14), gives summarization of Naive Bayes classifier model when clicked on *Training Statistics* and *Naive Bayes*.

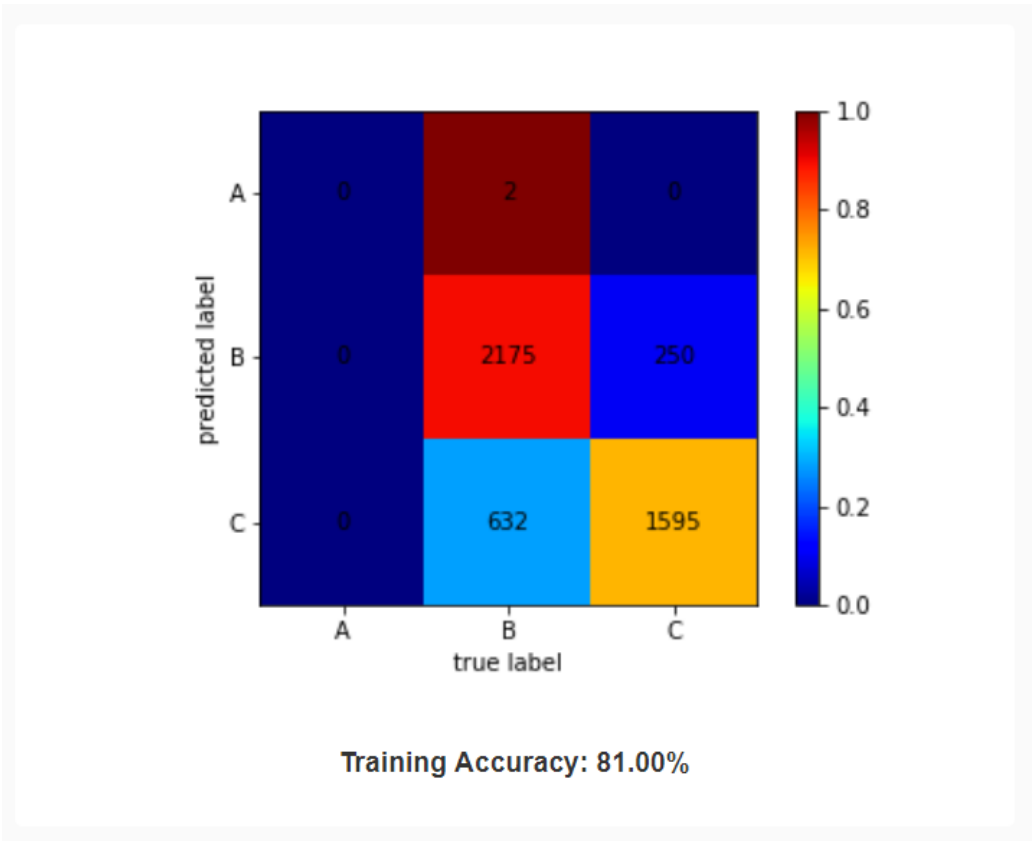


Figure 4.15: Random Forest Confusion Matrix.

Figure (4.15), gives summarization of Random Forest classifier model when clicked on *Training Statistics* and *Random Forest*.

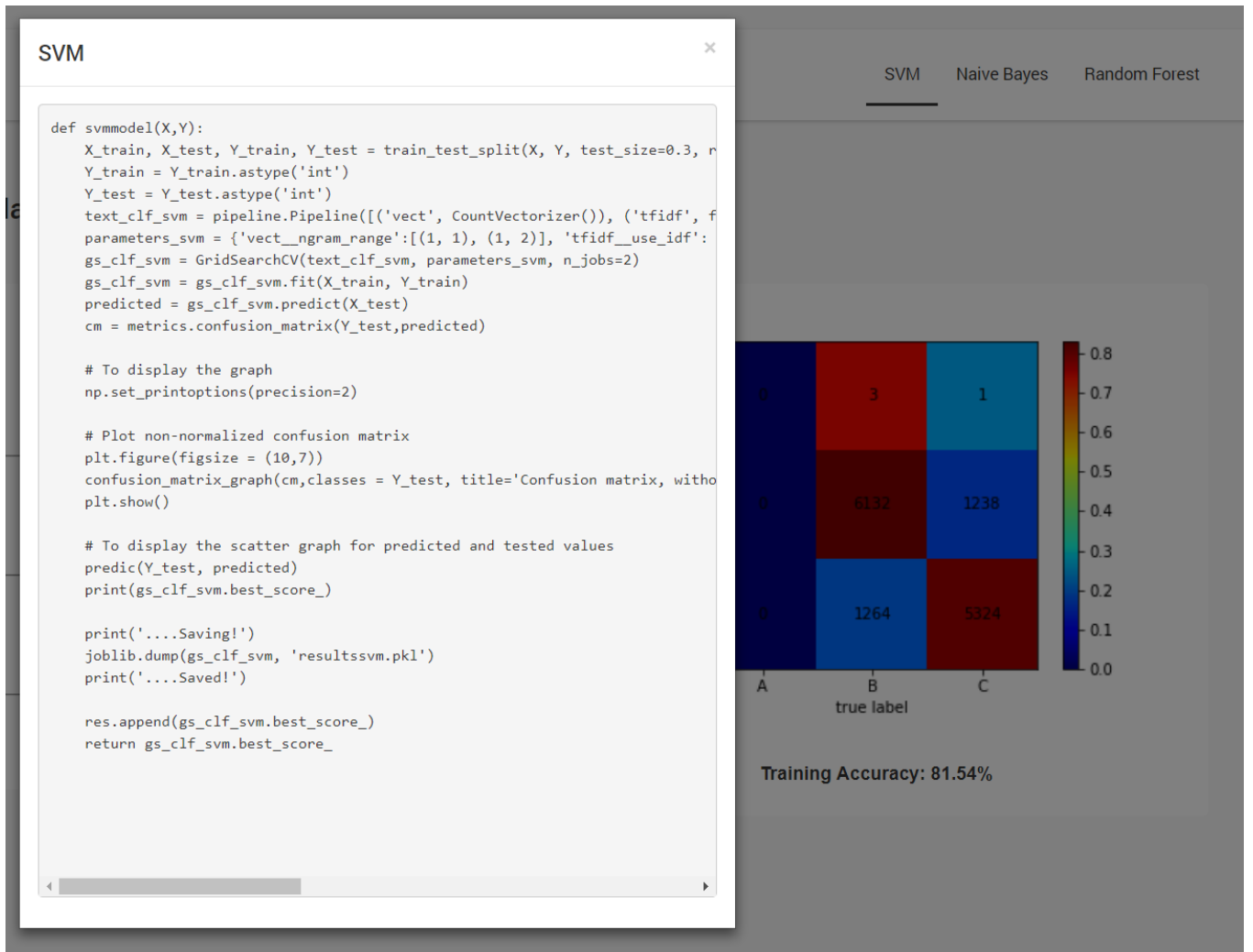


Figure 4.16: Function of SVM.

Figure (4.16), gives the function to display the graph of any classifier model in this case it is *SVM* on graph click.

After finding the confusion matrix, we then find the training accuracy of each algorithm. The training accuracy is depicted in the table given below.

Table 4.1: Training Accuracy

Sr No	Algorithms	Accuracy
1	Random Forest Classifier	81.00%
2	Support Vector Machine	81.54%
3	Naive Bayes Classifier	76.99%

## 4.2.2 Result Analysis

### Testing Phase

In this phase 10,000 Sarcastic sentences and 10,000 Non-sarcastic sentences are evaluated to find the accuracy of these three classifier models. Few of these sentences are shown in the table as :

Table 4.2: Examples

Sr No	Sentences	Random Forest	Support Vector Machine	Naive Bayes	Expected Outcome
1	Im glad to see you're not letting your education get in the way of your ignorance.	Yes	Yes	No	Sarcastic
2	You would never be able to live down to your reputation, but I see youre doing your best.: <i>smiling_imp</i> :	No	Yes	No	Sarcastic
3	When I was little I had a car door slammed shut on my hand. I still remember it quite vividly.	No	Yes	Yes	Non Sarcastic
4	That man is cruelly depriving a village somewhere of an idiot.	No	No	Yes	Sarcastic
5	She did not cheat on the test, for it was not the right thing to do.	Yes	No	Yes	Non Sarcastic
6	Joe made the sugar cookies; Susan decorated them.	No	No	No	Non Sarcastic
7	When I look into your eyes, I see straight through to the back of your head.	Yes	Yes	Yes	Sarcastic
8	When I was little I had a car door slammed shut on my hand. I still remember it quite vividly.	No	Yes	Yes	Non Sarcastic
9	They got there early, and they got really good seats.	No	No	Yes	Non Sarcastic
10	Whatever it is that is eating you, it must be suffering horribly.	No	No	No	Sarcastic
11	I just realized that movie lacked reality # not	Yes	Yes	Yes	Sarcastic
12	I am leaving now!	No	No	No	Non Sarcastic

13	Please contact us so we can consider adding your ideas to make the random sentence generator the best it can be.	No	Yes	No	Non Sarcastic
14	I was horrified watching the horror movie, So I started made me laughing # Lol	No	No	No	Sarcastic
15	Apparently I was not supposed to be happy : <i>unamused_face</i> :	Yes	Yes	Yes	Sarcastic
16	Our goal is to make this tool as useful as possible.	No	No	No	Non Sarcastic
17	I got the cold in here that I am sweating.	No	No	Yes	Sarcastic
18	I am going to take a leave from office today.	No	No	No	Non Sarcastic
19	Love is in the air that is why it is so suffocating # YeahRight : <i>unamused_face</i> : : <i>unamused_face</i> :	Yes	Yes	Yes	Sarcastic
20	Being ignored is Fun <i>sad_face</i>	Yes	Yes	Yes	Sarcastic

Here, Yes indicates that the classifier has predicted that sarcasm is present while No indicates that sarcasm is not present.

Thus we have calculated accuracy as the percentage of correct outputs for all the different datasets considered for testing.

$$Accuracy = \left( \frac{Total\_number\_of\_correct\_results}{Total\_number\_of\_test\_data} \right) * 100 \quad (4.1)$$

### 4.2.3 Evaluation

The table below gives information regarding testing accuracy obtained during evaluation of system.

Table 4.3: Testing Accuracy

Sr No	Algorithms	Accuracy
1	Random Forest Classifier	63.09%
2	Support Vector Machine	66.74%
3	Naive Bayes Classifier	67.81%

Performance Analysis is the graph which indicates the training accuracy as shown in figure(4.17), This graph will help the user to select the best algorithm that will be further used for sarcasm detection in text analytics.

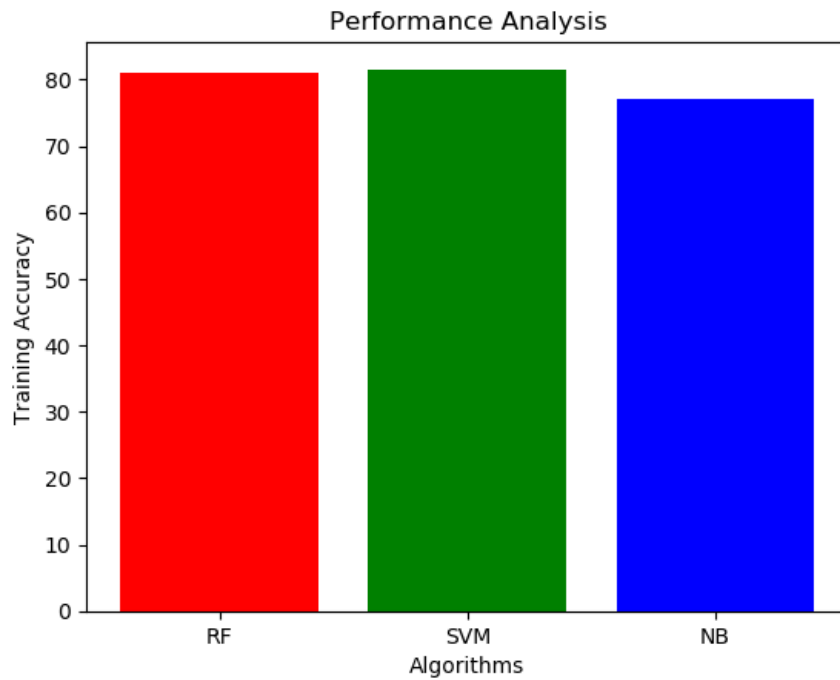


Figure 4.17: Performance Analysis.

The following graph in figure(4.18) helps us to compare which algorithm is best to classify the sentences into sarcastic and non-sarcastic respectively by comparing system training and testing accuracy respectively.

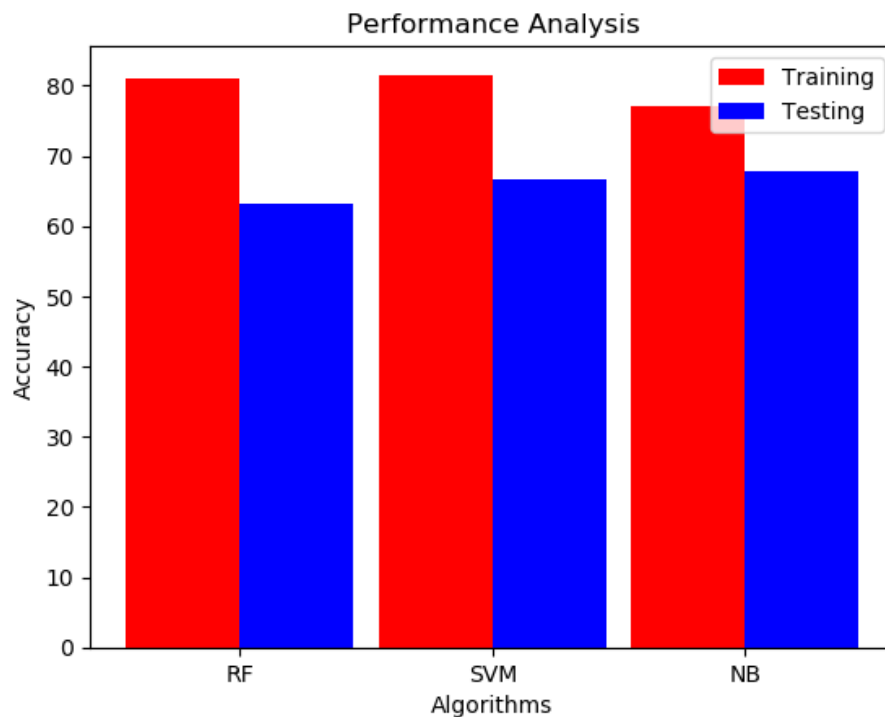


Figure 4.18: Accuracy Comparison.



# Chapter 5

## Applications

There are various applications of this domain system. Some of the applications are listed here.

### 5.1 Sentiment Analysis

Sentiment Analysis well known as opinion mining is used to detect, extract the real sentiments or thoughts of the users. This application deals with sentiments of the people and can be used to find the most appropriate conclusion for the given feedback, review or comment and would classify accordingly as Sarcastic, Non-Sarcastic or undeterminable.

#### 5.1.1 Review Summarization

Review Summarization plays an important role to understand the overall requirement of the market. It helps us to understand the mockery present in the reviews, comments or feedbacks. This would help to understand the real meaning and sentiments used in the feedbacks and would in turn help the market to grow in a more accurate and rapid way.

### 5.2 Public media analysis

It helps to understand the real intentions from the sentences. This would help to understand the comments used on social medias and blogs. It would lead to the better understanding of the post and understand the hidden sarcasm present in that post belonging to any social medias or blogs.

### 5.3 Business Analytics

Analysis of the product is a very important task which is carried out by an organisation to understand the progress of its products. This helps to boost the opinion mining systems by letting the respective organisation know the value of its products in the market among the users through the reviews and feedbacks obtained from various sites.

### 5.4 Literature Analysis

There are avid book readers all over the world. Sarcasm plays a very important role in finding the mockery present in e-books. This helps the person for a better understanding of the

messages conveyed in the books.

### **5.4.1 Analysis of Phrases**

People use phrases during their text conversation to convey their messages, so it is necessary to detect whether any sarcasm is present in these sentences as it can alter the important sentiment present in the message. It becomes very important to interpret the message property to analyse if the sentence is sarcastic or not.

### **5.4.2 Chat and Email Conversations**

People nowadays use lot of sarcastic text during their conversations and many of the times it goes undetected as it is very difficult for people to understand the actual message conveyed in the text. Thus it becomes very important to understand the current trends used by the people in their day to day conversations.

# Chapter 6

## Conclusion and Future Scope

Sarcasm plays an important role in our everyday life and can be observed in many application domains so, it becomes extremely important to detect the presence of sarcasm in it.

As, we know that every algorithm has its own advantages and completely different process to identify patterns. The training accuracy obtained after training the three classifier is as : Random Forest : 81%, SVM : 81.54% and Naive Bayes Classifier : 76.99%. While testing accuracy obtained after evaluating 10,000 dataset of each is as : Random Forest : 63.09%, SVM : 66.74% and Naive Bayes Classifier : 67.81% respectively.

The Naive Bayes algorithm performed better than the other two algorithm performed for identifying similarities between non sarcastic and sarcastic sentences respectively whereas by using Support Vector Machine, system has a slight edge for extracting sarcastic patterns.

Our System compares the best machine learning algorithm from the three algorithms viz. Naive Bayes, Random Forest and SVM to detect sarcasm present in the given text. It gives us the desired output from the features obtained during the training phase. But due to false positives and false negatives obtained while training, sometimes, this system can give us the wrong output. But this can be further improved by using deep learning techniques like Keras and Tensor-flow. Classifiers can be made more powerful by training more amount of dataset with emoticons which might increase the accuracy of the classifier.

# References

- [1] Whiting A and D Williams. Why people use social media: a uses and gratifications approach. *Qualitative Market Research: An International Journal*, 2013.
- [2] Apoorv Agarwal, Ilia Vovsha Boyi Xie, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. *In Proceedings of the ACL 2011 Workshop on Languages in Social Media*, pages 30–38, 2011.
- [3] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. *In Proceedings of COLING*, pages 36–44, 2010.
- [4] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. 2010.
- [5] R. Go, A. Bhayani and L Huang. Twitter sentiment classification using distant supervision. Technical report, CS224N Project Report, Stanford, 2009.
- [6] N. Kourtellis, P. Anderson J. Finnis, C. Borcea J. Blackburn, and A. Iamnitchi. Prometheus user-controlled p2p social data management for socially aware applications. 2010.
- [7] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2008.
- [8] Ashwin Rajadesingan, Reza Zafarani Arizona, and Huan Liu Arizona. Sarcasm detection on twitter: A behavioral modeling approach. 2015.
- [9] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. 2011.
- [10] Hiroshi Shimodaira. Text classification using naive bayes. 2015.
- [11] Haruna Isah Paul Trundle and Daniel Neagu. Social media analysis for product safety using text mining and sentiment analysis. 2015.

# Acknowledgment

I remain immensely obliged to my project guide Dr. Sharvari Govilkar, for her valuable guidance, patience, keen interest and constant encouragement and for her invaluable support.

I would like to thank Dr. Madhumita Chatterjee, Head of Computer Engineering Department for the invaluable support.

I would also like to thank Dr. Sandeep M. Joshi, Principal, PCE, New Panvel for his invaluable support and for providing an outstanding academic environment.

I would also like to thank Prof. Dhiraj Amin and all other staff members of the Department of Computer Engineering for their critical advice and guidance without which this project would not have been possible.

Last but not the least I would also like to acknowledge with much appreciation the crucial role of my family members and my friends who have been a constant source of inspiration during this project work. The completion of this project would not have been possible without them. I would like to say that it has indeed been a fulfilling experience working on this project.

Riya Das  
Shailey Kadam  
Chetan Kalra  
Vijeta Nayak

# Publications

1. *Conceptual Framework For Sarcasm Detection For English Text* by Riya Das, Shailey Kadam, Chetan Kalra, Vijeta Nayak and Dr. Sharvari Govilkar from Department of Computer Engineering, Mumbai University, PCE, New Panvel, India.
2. *Sarcasm Detection For English Text* by Riya Das, Shailey Kadam, Chetan Kalra, Vijeta Nayak and Dr. Sharvari Govilkar from Department of Computer Engineering, Mumbai University, PCE, New Panvel, India.

# CONCEPTUAL FRAMEWORK FOR SARCASM DETECTION FOR ENGLISH TEXT

Riya Das, Shailey Kadam, Chetan Kalra, Vijeta Nayak and Dr. Sharvari Govilkar  
Department of Computer Engineering, Mumbai University, PCE, New Panvel, India

**Abstract—** In recent years, with the increasing popularity of social media sites, people express themselves by communicating with each other in the form of texts. Without facial expression and vocal sounds it becomes very difficult to extract the exact meaning and intentions of the text. Sentiments, sarcasm and other elements present in spoken language are lost. So understanding the sentiments of the text becomes very important. Hence, one of the basic idea in sentimental analysis is to understand the sarcasm used in the statements. Our project mainly focuses on the detection and comparative analysis of three machine learning algorithms such as Random Forest, Support Vector Machine and Naive Bayes Classifier. Finally, from the results obtained after applying these algorithms on input data which comprises of emoticons, hashtags, punctuation marks the best approach for sarcasm detection can be selected.

**Index Terms —** Hashtags, Punctuation Marks, Emoticons, Random Forest Classifier, Support Vector Machine (SVM), Naive Bayes Classifier

## 1. INTRODUCTION

Sentiment analysis is one of the field in Natural Language Processing that deals with people's sentiments, attitude, and emotions from text. It is one of the most widely studied field in text mining. Sentiment analysis have many domains which needs to be analyzed such as consumer product reviews, review of any hotel, movie or social events. A common task in analyzing these domains is to classify the document or statement into positive or negative sentiments. There are many challenges in Sentiment Analysis and one of them is Sarcasm Detection.

Sarcasm is a mode of satirical wit depending for its effect on bitter, caustic, and often ironic language that is usually directed against an individual. It is often used for the purpose of criticism and mockery. Sometimes sarcasm is an intended humour. It is not implicitly understood by most individuals. Therefore, sarcasm detection is the need of an hour to understand the context in which a person is expressing his views. It helps to distinguish sentence into sarcastic and non-sarcastic sentences. Major conflicts which occurs are due to misunderstanding of a written text that can be avoided through sarcasm detection system.

The pitch and the tone of the speaker used in a sentence help to perceive the context of the given text. For example "I had a great time with you in traffic." The sentence

wanted to convey the message that he never enjoyed the traffic but during speech with the help of tone and expression of a person we can directly say that it is a negative message but in written text it is very difficult to understand whether it is sarcasm or not. Thus, the premise of a sentence is best understood by speech but a written text creates a lot of misunderstanding and confusion. In order the perceive the actual sentiment behind the written text and avoiding the conflicts among people. Therefore, analyzing any statement becomes our first priority.

Social Media websites, review on any product, chat applications are the best sources from which statements can be analyzed. Its anonymity provides the perfect ground to detect the presence of sarcasm in it. However, extensive usage of slang words, abbreviations, mixed language, number of punctuation marks, hashtags and emotions affect the meaning of the sentence. Hence a system is designed to analyze these given statements based upon the features like handling hashtags, punctuation marks, emoticons, etc. To enhance the result of sentiment analysis.

Our objective is to use the concept of machine learning in order to train and test various sentences. Social Media is the most budding platform with a great global outreach and an important source for sentiment analysis in social media analytic. Hence, this paper presents a method for detecting sarcasm in given text. Since, this project mainly focuses on English text, the most important process is to remove all other mixed languages present in the given statement. This is done by script validation and filtering of pre-processing block. Before training any dataset first step is to clean the data which is done by preprocessor block by removing stop words and HTML tags. This clean data is then used to train classifier such as Random Forest, SVM and Naive Bayes Classifier. The dataset is divided into approximately 70-30% in order to train and test data to get a confusion matrix which provides us with an estimate result about the training of our dataset. This paper also deals with comparing the result obtained from the above mentioned machine learning algorithms to find out which classifier gives a better result and accuracy so that the best classifier can be used in social media analytics in order to improve the understanding of the overall sentiment present in the statements.

The scope of the system would be to find the Sarcasm present in English Language Only. The contents are taken from Social media sites like twitter, Product based sites like Amazon, etc. This project deals with machine learning approach. There is always a mystery whilst encountering any kind of review. People tend to use sarcasm just for the sake of mocking a person's work or criticizing his efforts. Therefore, this system plays a vital role in sarcasm detection so that the essence of the sentence could be understood more effectively.

The recipients of the system would be organizations which use social media monitoring such as public opinion, reviews and rating of the product which provide valuable information about emerging trends and what consumers and clients think about specific topics, brands or products. and also with the rapid development of craze TV series, use of sarcasm in daily life has become more common and prominent. Besides this, use of Hashtags and emoticons have been rapidly increasing. Therefore, it has become a need of an hour for all these companies to understand the progress of their products in the market and among their clients.

## 2. LITERATURE SURVEY

Whiting, A. and D. Williams [1], proposed the paper that explored the uses and gratification of the consumer that they receive from social media. Based on the study of 25 interviews of people using gratifications, 10 uses and gratifications are listed with their usage are as follows:- Social interaction - 88%, Information seeking - 80%, Pass time - 76%, Entertainment - 64%, Relaxation - 60%, Expression of opinions - 56%, Communicatory utility - 56%, Convenience utility - 52%, Information sharing - 40% and Surveillance/knowledge about others - 20%. This application of uses and gratifications theory to social media not only proves to be rich and comprehensive understanding of the reasons of the consumers to utilize social media but it effectively contributes to the business and social media marketing and also helps in communicating with the potential customers by fulfilling their needs .

Hiroshi Shimodaira [11], classified the documents with its contents, and of the words of which they are composed of. Two document models - Bernoulli and Multinomial were used for the classification. The Zero Probability Problem is overcome by Laplace's law of succession or add one smoothing, that adds a count of one to each word type. Naive Bayes approximation can be used for document classification, by constructing distributions over words. The classifiers require a document model to estimate  $P(\text{document} \mid \text{class})$ . 1. Bernoulli document model : a document is represented by a binary feature

vector, whose elements indicate absence or presence of corresponding word in the document. 2. Multinomial document model : a document is represented by an integer feature vector, whose elements indicate frequency of corresponding word in the document.

B. pang and L. Lee [9], stated the General challenges for opinion mining and Sentiment analysis which are: Contrasts with standard fact-based textual analysis, Factors that make opinion mining difficult. The mentioned Key Concepts are sentiment Polarity and degree of positivity, Subjectivity detection and opinion identification, Joint-topic sentiment analysis, Viewpoints and perspectives. Then the various features taken into considerations are Term Presence vs Frequency, Term-based features beyond term unigrams, Parts of speech, Syntax, Negation and Topic-oriented features. Afterwards Impact of labeled data is observed and obtained. Here, the unsupervised approach used are Unsupervised lexicon induction. The classification based on relationship information are: Relationships between sentences and between documents, Relationships between discourse participants, Relationships between product features, Relationships between classes and Incorporating discourse structure. Special considerations for extraction are: Identifying product features and opinions in reviews and Problems involving opinion holders. Basically, the aim to use all the techniques is achieved, but no conclusion either positive or negative can be made for the algorithms used.

Alec Go, Richa Bhayani and Lei Huang [5], proposed a different approach of Distant Supervision as they removed all emoticon and non-word tokens while training their algorithms. They found that removing the non-word tokens allowed the classifiers to focus on other features like classification. They used tweets ending in positive emoticons like “:)” “:-)” as positive and negative emoticons like “:(” “:-)” as negative. They applied Naive Bayes, Maximum Entropy, and Support Vector Machine algorithms to classify Twitter sentiment which resulted to be in the range of 80% accuracy. They concluded that the unigram model outperforms all other models used, specifically bigrams and POS features do not help. Suggestion of using Maximum Entropy classifier was provided to obtain best result of 83% with both Unigrams and Bigrams during classification. They suggested that domain- specific tweets, handling neutral tweets, sentiment analysis in regional language and utilizing emoticon data in the test set must be considered to make proposed algorithm error resistance and with higher accuracy.

Luciano Barbosa and Junlan Feng [3], understood the usage of meta - information along with feature based



model which gave description and information regarding hash tag, punctuation and emoticons. From the discussion, and observation an algorithm was designed which featured that for a given word in a tweet, they mapped these words to its part-of-speech using a part-of-speech dictionary and opinion based messages contains adjectives or interjection. Along with this mapping, the word also mapped with its subjectivity. The algorithm used a two-step classification method : first training a classifier to distinguish between subjective and objective tweets and secondly training another classifier to differentiate between positive and negative sentiment. The accuracy rate was calculated on the popular list of words collected from various websites by comparing between various approaches such as ReviewSA, Unigrams, TwitterSA. These methods, reduced the error rate was from 46% to 23%. The main limitation of this approach was in the cases of sentences that contain antagonistic sentiments and web vocabulary.

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passonneau [2], discussed SVM with Unigram based, feature based and tree kernel based model. They collected data source from various websites and generated their sample sentences. The dataset used was trained with polarity and tweets contained emoticons and noisy labels. For evaluation of data collected they used a 5-fold cross validation model. For binary classification of sentence various techniques were developed which gave result as : For Only unigram classification it had result of 71.35%, Kernel Tree Technique gave an accuracy of 73.98%, Unigram + Senti-feature technique found to be 75.39% accurate, Kernel + senti-feature technique resulted into 74.61%. Same techniques were used for ternary classification which gave 56.58, 56.31, 60.6 and 60.50% accurate results. They therefore concluded that unigram+senti-feature gave result with maximum accuracy for binary classification. Finally from the analysis it was stated that sentiment analysis for Twitter data is not that different from sentiment analysis for other genres.

Ashwin Rajadesingan and Reza Zafarani Arizona and Huan Liu Arizona [10], identify the traits using the user's past tweets. SCUBA framework for Behavioral modeling Approach was being used to analyze the user's past tweets and categorize it as: Sarcasm as a contrast of sentiments in which divisions were made based on Contrasting connotations and Contrasting present with the past, Sarcasm as a complex form of expression where readability was considered. Observations are, with no historical information, accuracy of 79.38%, is obtained. considerable gain (+4.14%) in performance is obtained by observing past 30 tweets. But including more past history still gives significant results. Results have derived that

SCUBA is effective in detecting sarcastic tweets.

Dmitry Davidov, Oren Tsur and Ari Rappoport [4], discussed a supervised classification framework that provided a way to utilize tagged data and emoticons for classification. It calculated the contribution of different feature types for sentiment classification and it was shown that the framework successfully identified sentiment types of untagged tweets. This quality was confirmed by human judges. They developed a methodology in which they used four basic feature types for sentiment classification : single word features, n-gram features, pattern features and punctuation features. All these feature types are combined into a single feature vector. They also used surface patterns to classify the words into high frequency words and content words. For each feature vector construction they developed and used k-nearest neighbours (KNN) strategy for classification and with help of Euclidean distance to matching vectors were calculated. The Amazon Mechanical Turk (AMT) service was used to obtain a list of the most commonly used and unambiguous ASCII smileys to train and generate the data sets. They also discussed about algorithms that would help them to find dependencies and overlapping between different sentiment types represented by all smileys and hashtags.

N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iammitchi [8], introduced a peer-to-peer service (Prometheus, a P2P service that enables socially-aware applications by providing decentralized, user-controlled social data management). They emulated Prometheus the workload of two socially-aware applications and one social sensor based on previous system characterizations. The social-based mapping of users onto peers leads to significant improvements, especially for the 30 users/peer case. 15% of the invocations finishing faster when compared to the random case (some invocations can finish in half the time).

Haruna Isah, Paul Trundle, Daniel Neagu [6], proposed a product safety framework using text mining and sentiment analysis. They utilized the framework to gather and analyse views and experiences of users of drug and cosmetic products. They also demonstrated how to develop product safety lexicon and training data. Naive Bayes Classifier is used for implementation obtaining 83% of accuracy for twitter. Since, this research is work in progress yet can be used for users, product manufacturers, regulatory and enforcement agencies.

Hassan Saif, Yulan He and Harith Alani [7], proposed a novel approach of adding sentiment at each topic levels along with lexicon based pattern matching algorithm. For each extracted entity from tweets, the algorithm added the

semantic concept as an additional feature and measured the correlation between the concept and negative/positive sentiment. The model used techniques such as Unigram, POS, Sentiment at topic level and semantics. Unigram and POS techniques were used to find whether sentences are positive or negative as these sentences were gathered from Stanford Twitter Sentiment Corpus. These sentences when tested, gave accuracy around 71.5% and 75.53% respectively. Data collected from Health Care Reform were trained and tested using sentiment at topic level which evaluated to be 77.02%. OMD used n-fold cross validation to detect the semantics that evaluated to be 77.18%. All these techniques used were based on binary classification. To implement these techniques three different approaches were incorporating for the analysis; replacement, augmentation, and interpolation.

So we can conclude that the approach for the Sarcasm Detection started with Lexicon based approach. Then Machine Learning was used in the mid years. As machine learning approach requires more amount of time to train the dataset, so a hybrid approach consisting of both lexical and machine learning approach was implemented which gave results in optimized amount of time.

The main inference obtained was, Twitter is not different from other social medias and therefore same approach can be used for analyzing the sarcasm present in the data from other sources as well. Therefore, this project mainly focuses on machine learning approach as it is better way to obtain whether sentences are sarcastic or not to increase its result and accuracy.

### 3. PROPOSED ARCHITECTURE

In this, we would be discussing about the proposed system architecture. The input of the system would be reviews or simply some content from various Social Media Sites Amazon, Zomato, etc. and tweets from twitter, etc..The first step is to clean the raw input so that a standardized format of content is obtained. From this clean data obtained we have constructed our dataset which is used in training phase to train the various machine learning classifier.

The system proposed will mainly focuses on English text, therefore the important part before cleaning the content to get dataset, it is necessary to check whether the content is in English text only. This is done by script validation and filtering of pre-processing block. Next step is to clean the content by removing all URLs present in the data, unwanted HTML tags and converting the whole data in the dataset into lower case. This cleaned data is then converted into standard format to get our cleaned data set which would be in the form of data matrix with two columns as review and labels.

The proposed system would use three machine learning algorithms to train our classifier such as Random Forest, Support Vector Machine (SVM) and Naive Bayes Classifier. The system during training phase, builds a classifier by analyzing the training data and associated label with each class and develops a pickle file which consists of all the features extracted by the model in training phase.

During testing the system accepts the input from the user and compares with the features stored in pickle and predict whether given input sentence is sarcastic or not.

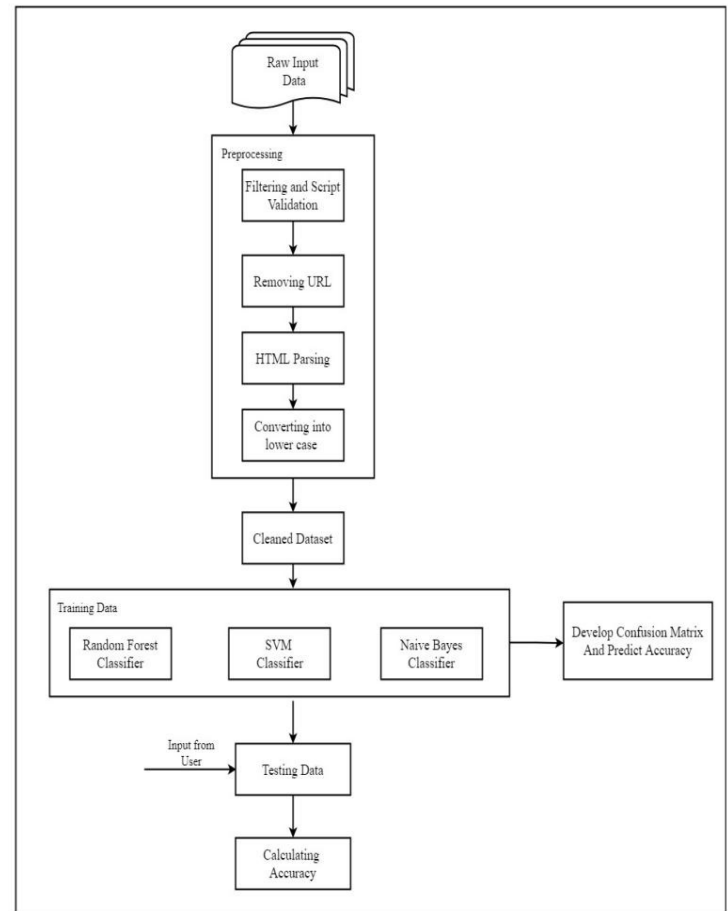


Figure 1 : Proposed Architecture

The main aim of our system is to compare these machine learning algorithm to find which algorithm can be further used to detect sarcasm during text analytic.

### 4. CONCLUSION

Thus, the system will be built in order to detect the presence of sarcasm not only in twitter tweets but also on various contents collected from various social media sites. As our system is implemented only for English text script validation and filtering is done with help of UTF-8. Before generating the dataset for training and testing it is

necessary to clean all raw inputs. This would be implemented by using pre-processing block. It also uses three supervised machine learning algorithm such as Random Forest, Support Vector Machine and Naive Bayes Classifier to train the classifier on the given dataset collected from social media sites in order to find certain features and storing it separately in a file which would be used later for prediction during evaluation process. Depending upon the nature of data and classification algorithm used, the system will then use the major chunk of data for training and remaining for testing.

#### ACKNOWLEDGEMENT

We would like to thank Dr. Madhumita Chatterjee, Head of Computer Engineering Department for the invaluable support. We would also like to show our gratitude towards Dr. Sandeep M. Joshi, Principal, PCE, New Panvel for his invaluable support and for providing an outstanding academic environment. Last but not the least we would like to thank Prof. Dhiraj Amin and all other staff members of the Computer Engineering Department for their critical advice and guidance without which this project would not have been possible. Also, we would like to say that it has indeed been a fulfilling experience for working out this project topic.

#### REFERENCES

- [1] Whiting A and D Williams. Why people use social media: a uses and gratifications approach. *Qualitative Market Research: An International Journal*, 2013
- [2] Ilia Vovsha Owen Rambow Apoorv Agarwal, Boyi Xie and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the ACL 2011 Workshop on Languages in Social Media*, pages 30-38, 2011.
- [3] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of COLING*, pages 36-44, 2010.
- [4] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. 2010.
- [5] Bhyani R. Go, A. and L Huang. Twitter sentiment classification using distant supervision. Technical report, CS224N Project Report, Stanford, 2009.
- [6] Daniel Neagu Haruna Isah, Paul Trundle. Social media analysis for product safety using text mining and sentiment analysis. 2015.
- [7] Yulan He Hassan Saif and Harith Alani. Semantic sentiment analysis of twitter. 2011.
- [8] P. Anderson J. Blackburn C. Borcea N. Kourtellis, J. Finnis and A. Iamnitchi. Prometheus user-controlled p2p social data management for socially aware applications. 2010.
- [9] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*,

2008.

[10] Ashwin Rajadesingan, Reza Zafarani Arizona, and Huan Liu Arizona. Sarcasm detection on twitter: A behavioral modeling approach. 2015.

[11] Hiroshi Shimodaira. Text classification using naive bayes. 2015.

## Paper 2

# SARCASM DETECTION FOR ENGLISH TEXT

Riya Das, Shailey Kadam, Chetan Kalra, Vijeta Nayak and Dr. Sharvari Govilkar  
Department of Computer Engineering, Mumbai University, PCE, New Panvel, India

***Abstract**— Sarcasm determines the mockery or irony used by that person to express his emotions. With the increase in the use of social medias which is mostly in the form of text, it becomes important to detect the sarcasm present in the sentences. So understanding the sentiments of the text becomes very important. In our previous paper [14] we proposed a conceptual framework for Sarcasm detection using three machine learning algorithms Viz. Random forest, Naive Bayes, SVM. Our training consists of Twitter dataset with emoticons, punctuation's, hashtags and other dataset from different sites. This paper describes the processing steps and the actual work flow and compares the best algorithm among the three algorithms for future work purposes.*

**Index Terms** — Hashtags, Punctuation Marks, Emoticons, Random Forest Classifier, SVM, Naive Bayes Classifier

### 1. INTRODUCTION

Our objective is to use the concept of machine learning in order to train and test various sentences. Hence, this paper presents a method for detecting sarcasm in given text. Our dataset is a collection of tweets and various reviews with 46,000+ sentences.

Since our project mainly focuses on English text, the most important process is to remove all other mixed languages present in the given statement. This is done by script validation and filtering of pre-processing block. Before training any dataset first step is to clean the noise present in the dataset, which is done by preprocessor block by removing stop words and HTML tags. This cleaned data is then used to train classifiers such as Random Forest, SVM and Naive Bayes. The dataset is divided approximately into 70-30% in order to train and test data to get the desired result. A confusion matrix is then formed which helps us to understand the number of false positives and false negatives during the training part. This paper also deals with comparing these result to find out which classifier gives a better result and accuracy so that the best classifier can be used in social media analytics in order to improve the overall sentiment of these statements. The scope of the system would be to find the Sarcasm present in English Language Only.

The recipients of the system would be organizations which use social media monitoring such as public opinion, reviews and rating of the product which provide valuable information about emerging trends and what consumers and clients think about specific topics, brands or products. and also with the rapid development of craze TV series, use of sarcasm in daily life has become more

common and prominent. Besides this, use of Hashtags and emoticons have rapidly been increasing. Therefore, it has become a need of an hour for all these companies to understand the progress of their products in the market and among their clients.

### 2. LITERATURE SURVEY

As discussed in our previous paper [14] we can conclude that though sarcasm can be determined with a lexicon based approach, but it would take more time for computation. While if we can obtain the features and store it in a file, we can reuse the same featured for determining sarcasm any number of times without actually performing all the processes. Therefore, our project mainly focuses on supervised machine learning approach as it is better to train and store the features, and use them for testing other sentences.

### 3. SARCASM DETECTOR

In this, we would be discussing about the system architecture. The input of the system would be reviews or simply some content from various Social Media Sites and tweets from twitter, etc.. The first step is to clean the raw input so that a standardized format of content is obtained. From the cleaned data, we have constructed our dataset which is used in training phase to train the various machine learning classifiers.

Few preprocessing of data is done like script validation, removal of URLs and HTML tags. This cleaned data is then converted into standard format i.e data matrix with reviews and labels. labels is of two types 0 and 1 indicating the sentence being not sarcastic and sarcastic respectively.

Training data consists of hashtags, emoticons, punctuation marks and too positive and negative sentences, therefore there is no need to handle them separately. The system uses three supervised machine learning algorithm, such as **Random Forest, Support Vector Machine (SVM) and Naive Bayes Classifier** to train and test the dataset.

In training phase the algorithm builds a classifier by analyzing the training data and associated label with each class and creates a pickle file which consists of all the features extracted by the model in the training phase.

From the data model created, a confusion matrix is generated which help us to find the number of true positives, true negatives, false positives and false negatives during the training phase to understand how accurately the data is being trained by each classifier. During the testing phase, the system accepts the input from the user and compares with the features stored in the pickle file and predicts whether the given input sentence is sarcastic or not.

The main aim of the system is to compare these algorithms to find which algorithm can be further used to detect sarcasm during text analytics.

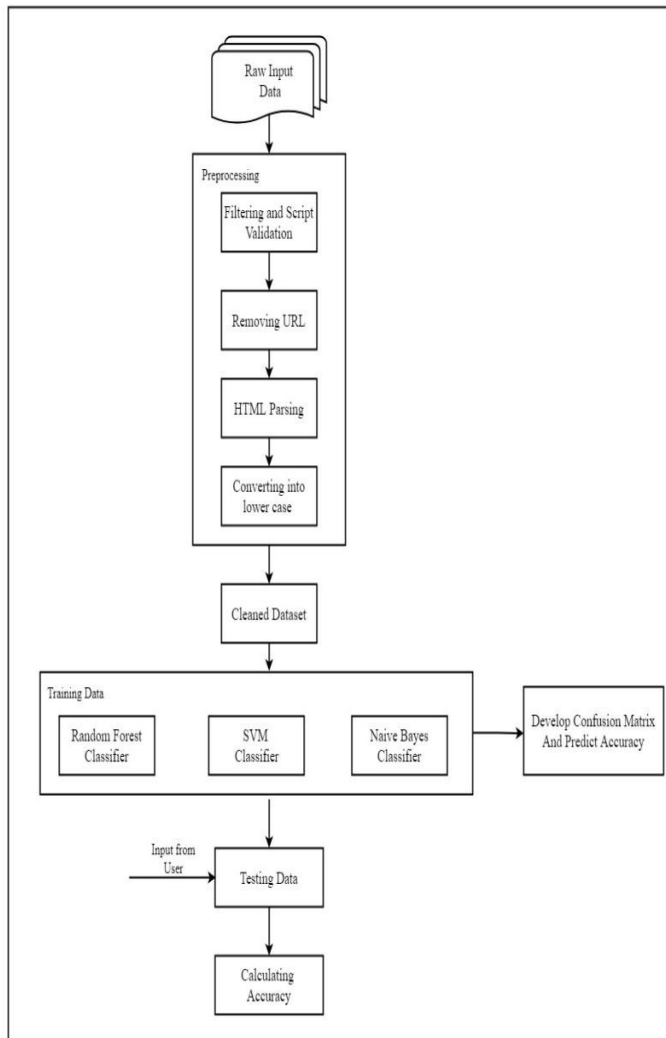


Figure 1 : Sarcasm Detector

### 3.1 Input Documents

The text will be in Romanized English format. The content would be collected from different social media domains like Twitter or from product based websites like Amazon, etc.

### 3.2 Preprocessing Block

The process of converting raw input data collected from various social media sites and twitter into standardized format of data matrix i.e label and review.

#### 3.2.1 Filtering and Script Validation

The process of considering only English text by ignoring all the mixed language text so that processing of text can be made easier.

In this step, the given sentence is scanned character by character and compared with UTF-8. If character is present in the given list, then it does not belong to English Script and hence can be ignored.

#### 3.2.2 Removing URLs

The process of removing all unwanted text such as URL so that more informative data can be stored in the dataset for training.

Algorithm :

- a. Input : The sentences only containing English Text and special characters like hashtags, emojis, punctuation marks, etc.
- b. Output : URL present in the sentence are removed.
- c. Steps :
  - i. START.
  - ii. Define a regular expression to identify the presence of `https://www.abc.com`
  - iii. Scan the input document.
  - iv. Check for not End of file.
    1. Read a character from input file.
    2. IF character matches with regular expression then remove it.
    3. Display the text after removing text otherwise go to step 4.
    4. Read the next input sentence.
    5. STOP.

#### 3.2.3 Removing HTML Tags

The process of removing all unwanted text such as HTML tags so that more informative data can be stored in the dataset for training.

Algorithm :

- a. Input : Sentences with no URLs.
- b. Output : Sentences without any HTML tags.
- c. Steps :
  - i. START.
  - ii. Identify all predefined HTML Tags by using predefined packages.
  - iii. If the sentences contain any html Tags then

remove it and display it otherwise go to next step.

- iv. Read the next input sentence.
- v. Presence of HTML tags can be compared by comparing the input and output string of this block.
- vi. Repeat the same process until end of document is found.
- vii. STOP.

### 3.2.4 Converting into Lower Case

This block converts the input string into one standardized format which is in lower case.

### 3.2.5 Clean Dataset

This block contains dataset free from all unwanted URL, HTML tags and converted into Lower Case.

Stop words are not removed during pre processing as it might contain some sentiments that would affect its meaning. In this blocks labels are assigned to each sentences and are stored into standardized format i.e review and its corresponding label. Labels are in form of 1 and 0 which represent sentences are sarcastic or non - sarcastic respectively.

## 3.3 Training Classifier

Data Classification is termed as the process that organizes data into categories so that it can be used efficiently and effectively. It basically has two phases :

- a. **Training Phase :** At this phase, the classification algorithm uses the training data for analysing.
- b. **Testing Phase :** In this phase, testing data are used to estimate the accuracy of the classifier. Testing data is the dataset used for evaluating the model in the training phase.

Based upon the data chunk the dataset is divided for training and testing. Ideally we used 70-30% to train and test data respectively.

### 3.3.1 Tf-idf

The TF (term frequency) of a word is the frequency of a word (i.e. number of times it appears) in a document.

The IDF (inverse document frequency) of a word is the measure of its importance in the whole corpus.

The formula for to measure Tf-idf is :

$$tfidf(t,d,D)=tf(t,d)*idf(t,D).....(3.1)$$

Where t denotes the terms; d denotes each document; D denotes the collection of documents.

### 3.3.2 Random Forest Classifier

Random forest algorithm is one of the supervised learning classification algorithm. This classifier generates large number of decision trees and randomly selects the best node from which features can be extracted and stored.

With increased number of trees for predication will automatically gives higher accuracy results. Hence, of our system we have generated maximum number of trees which help us to extract features for the classifier.

Algorithm for Random Forest can be divided into two phases :

- i. Train the Dataset
- ii. Random Forest Prediction

Algorithm :

- i. Define parameters using TfidfVectorizer.
- ii. Train the classifier with the parameters defined.
- iii. Make predictions of data from training dataset.
- iv. Find accuracy and confusion matrix for training and testing dataset.
- v. Plot confusion matrix.

### 3.3.3 Support Vector Machine

A Support Vector Machine (SVM) is also one of the supervised machine learning algorithm that can be used for both classification and regression purposes. It is mainly used in classification problems.

In this algorithm, each data item is plotted against hyper plane in space with its feature extracted as the value od data item. The data points which are nearest to the defined hyper plane is called as support vectors.

- i. CountVectorizer : It converts a collection of text documents to a matrix of token counts. This implementation produces a sparse representation of the counts.
- ii. SGDClassifier : SGD stands for Stochastic Gradient Descent where the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule.
- iii. GridSearchCV : If it is not used we need to loop the parameters and run all the combination of parameters. For this we need to write the code manually which increases the time requirements.

Hence, for our system we have used GridSearchCV.

Algorithm :

- i. Defining various parameters using SGDClassifier.
- ii. Use GridSearchCV to iterate the parameters automatically.
- iii. Train the classifier based upon parameters defined.

- iv. Make predictions of data from training dataset.
- v. Find accuracy and confusion matrix for training and testing dataset.
- vi. Plot confusion matrix.

### 3.3.4 Naive Bayes Classifier

Naive Bayes Classifier is based on the Bayesian theorem. It is suitable where the dimensionality of the input attributes is high. In this model, parameter estimation is done by using maximum likelihood. It is used to find conditional probabilities.

$P(X|Y)$  is the conditional probability of event  $X$  occurring for the event  $Y$  which has already been occurred.

$$P(X|Y) = P(X \text{ and } Y) / P(Y) \dots \dots \dots (3.2)$$

- a. MultinomialNB Classifier : For our system we have implemented MultinomialNB which makes use of the Naive Bayes algorithm for multinomially distributed data. The parameters is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting.

Algorithm :

- i. Define parameters using TfidfVectorizer and MultinomialNB.
- ii. Training the classifier with the parameters defined.
- iii. Make predictions of data from training dataset.
- iv. Find accuracy and confusion matrix for training and testing dataset.
- v. Plot confusion matrix.

## 4. RESULT ANALYSIS

Training dataset is generated by cleaning the raw data collected from various social media sites like Amazon, Facebook, etc. and tweets from twitter. For the evaluation of our system, we have used 10,000 sentences of each type for each classifier model. The system extracts the features from the input sentence and compare it with the features stored in pickle file to detect whether the given input sentence is sarcastic or not.

Example 1 : Apparently I was not supposed to be happy :unamused\_face:  
Random Forest : Yes  
SVM : Yes  
Naive Bayes Classifier : Yes  
Expected Outcome : Sarcastic

Example 2 : I am going to take a leave from office today.  
Random Forest : No  
SVM : No  
Naive Bayes Classifier : No  
Expected Outcome : Non - Sarcastic

Example 3 : Whatever it is that is eating you, it must be suffering horribly.

Random Forest : No

SVM : No

Naive Bayes Classifier : No

Expected Outcome : Sarcastic

The efficiency of our system is based on the confusion matrix generated after training the classifier and number of correct output given by each classifier for input sentence during testing.

The following graph shows the accuracy obtained by the system during training phase.

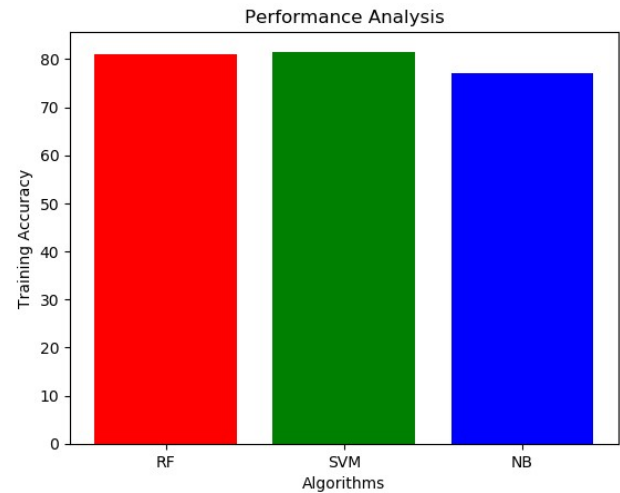


Figure 2 : Performance Analysis for Training Phase

Therefore the graph below helps us to compare which algorithm is best to classify the sentences into sarcastic and non-sarcastic respectively.

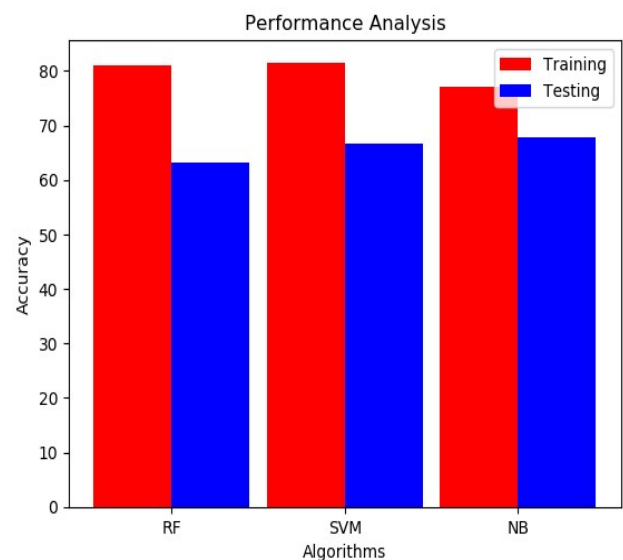


Figure 3 : Comparison in accuracy

## 5. CONCLUSION

Every algorithm has its own advantages and completely different process to identify patterns. The training accuracy obtained after training the three classifier is as :

Table 1 : Training Accuracy

Algorithms	Accuracy
Random Forest	81%
SVM	81.54%
Naive Bayes	76.99%

While testing accuracy obtained after evaluating 10,000 dataset of each is as :

Table 2 : Testing Accuracy

Algorithms	Accuracy
Random Forest	63.09%
SVM	66.74%
Naive Bayes	67.81%

The Naive Bayes algorithm performed better than the other two algorithm performed for identifying similarities between non sarcastic and sarcastic sentences respectively whereas by using Support Vector Machine, system has a slight edge for extracting sarcastic patterns.

Our System compares the best machine learning algorithm from the three algorithms viz. Naive Bayes, Random Forest and SVM to detect sarcasm present in the given text. It gives us the desired output from the features obtained during the training phase. But due to false positives and false negatives obtained while training, sometimes, this system predicts a wrong output. But this can be further improved by using deep learning techniques like Keras and Tensor-flow. Classifiers can be made more powerful by training more amount of dataset with emoticons which might increase the accuracy of the classifier.

## Acknowledgment

We would like to thank Dr. Madhumita Chatterjee, Head of Computer Engineering Department for the invaluable support. We would also like to show our gratitude towards Dr. Sandeep M. Joshi, Principal, PCE, New Panvel for his invaluable support and for providing an

outstanding academic environment. Last but not the least we would like to thank Prof. Dhiraj Amin and all other staff members of the Department of Computer Engineering for their critical advice and guidance without which this project would not have been possible. Also, we would like to say that it has indeed been a fulfilling experience for working out this project topic.

## REFERENCES

- [1] Whiting A and D Williams. Why people use social media: a uses and gratifications approach. Qualitative Market Research: An International Journal, 2013
- [2] Ilia Vovsha Owen Rambow Apoorv Agarwal, Boyi Xie and Rebecca Passonneau. Sentiment analysis of twitter data. In Proceedings of the ACL 2011 Workshop on Languages in Social Media, pages 30-38, 2011.
- [3] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In Proceedings of COLING, pages 36-44, 2010.
- [4] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. 2010.
- [5] Bhyani R. Go, A. and L Huang. Twitter sentiment classification using distant supervision. Technical report, CS224N Project Report, Stanford, 2009.
- [6] Daniel Neagu Haruna Isah, Paul Trundle. Social media analysis for product safety using text mining and sentiment analysis. 2015.
- [7] Yulan He Hassan Saif and Harith Alani. Semantic sentiment analysis of twitter. 2011.
- [8] P. Anderson J. Blackburn C. Borcea N. Kourtellis, J. Finnis and A. Iamnitchi. Prometheus user-controlled p2p social data management for socially aware applications. 2010.
- [9] B. Pang and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2008.
- [10] Ashwin Rajadesingan, Reza Zafarani Arizona, and Huan Liu Arizona. Sarcasm detection on twitter: A behavioral modeling approach. 2015.
- [11] Hiroshi Shimodaira. Text classification using naive bayes. 2015.
- [12] <https://github.com/AniSkywalker/-Dataset>
- [13] <http://scikit-learn.org/stable/>
- [14] Conceptual Framework For Sarcasm Detection for English - Riya Das, Shailey Kadam, Chetan Kalra and Vijeta Nayak (Department of Computer Engineering, Mumbai University, PCE, New Panvel, India).



# Certificates

1. We participated in the 1<sup>st</sup> Inter-College Project Competition organized by the Computer Engineering Department of *St. Francis Institute Of Technology* on 7<sup>th</sup> April ,2018.



# St. Francis Institute of Technology

(A.I.C.T.E. Approved, Affiliated to University of Mumbai. All the UG programs NBA Accredited & ISO 9000:2008 Certified)  
Mt. Painsur, S.V.P Road, Borivli (W), Mumbai - 400103. Phone: 022 2892 8585, 2890 8585. Email: [sfedu@sftengg.org](mailto:sfedu@sftengg.org)

## Department of Computer Engineering

# 1<sup>st</sup> Inter-College Project Competition 2017-18

## CERTIFICATE

This is to certify that RIYA BIKASH DAS has presented the  
project titled SARCASM DETECTION IN ENGLISH TEXT

in the 1<sup>st</sup> Inter-College Project Competition organized by Department of Computer Engineering,

St. Francis Institute of Technology on 7<sup>th</sup> April 2018.

Dr. Kavita Sonawane

Convenor/Head of Department

Dr. Siney George

Principal

Bro. Jose Thurruthy

Director



# St. Francis Institute of Technology

(A.I.C.T.E. Approved, Affiliated to University of Mumbai. All the UG programs NBA Accredited & ISO 9000:2008 Certified)  
Mt. Painsur, S.V.P Road, Borivli (W), Mumbai – 400103. Phone: 022 2892 8585, 2890 8585. Email: [stedu@stfengg.org](mailto:stedu@stfengg.org)

Department of Computer Engineering

## 1<sup>st</sup> Inter-College Project Competition 2017-18

### CERTIFICATE

This is to certify that SHAILEY KADAM has presented the  
project titled SARCASM DETECTION IN ENGLISH TEXT

in the 1<sup>st</sup> Inter-College Project Competition organized by Department of Computer Engineering,

St. Francis Institute of Technology on 7<sup>th</sup> April 2018.

Dr. Kavita Sonawane

Convenor/Head of Department

Dr. Sincy George

Principal

Bro. Jose Thuruthiyil

Director





# St. Francis Institute of Technology

(A.I.C.T.E. Approved, Affiliated to University of Mumbai. All the UG programs NBA Accredited & ISO 9000:2008 Certified)  
Mt. Painsur, S.V.P Road, Borivili (W), Mumbai - 400103. Phone: 022 2892 8585, 2890 8585. Email: [stedu@sftengg.org](mailto:stedu@sftengg.org)

Department of Computer Engineering

## 1<sup>st</sup> Inter-College Project Competition 2017-18 CERTIFICATE

This is to certify that CHE TAN KALRA has presented the  
project titled SARCASM DETECTION IN ENGLISH TEXT

in the 1<sup>st</sup> Inter-College Project Competition organized by Department of Computer Engineering,

St. Francis Institute of Technology on 7<sup>th</sup> April 2018.

Dr. Kavya Sonawane

Convenor/Head of Department

Dr. Sincy George

Principal

Bro. Jose Thuruthiyil

Director



# St. Francis Institute of Technology

(A.I.C.T.E. Approved, Affiliated to University of Mumbai. All the UG programs NBA Accredited & ISO 9000:2008 Certified)  
Mt. Painsur, S.V.P Road, Borivali (W), Mumbai – 400103. Phone: 022 2892 8585, 2890 8585. Email: [stedu@sftengg.org](mailto:stedu@sftengg.org)

Department of Computer Engineering

## 1<sup>st</sup> Inter-College Project Competition 2017-18

### CERTIFICATE

This is to certify that VIJETA NAYAK has presented the  
project titled SARCASM DETECTION IN ENGLISH TEXT

in the 1<sup>st</sup> Inter-College Project Competition organized by Department of Computer Engineering,

St. Francis Institute of Technology on 7<sup>th</sup> April 2018.

Dr. Kavita Sonawane

Convenor/Head of Department

Dr. Sincy George

Principal

Bro. Jose Thuruthiyil

Director