A Project Report

on

**Sarcastic Comments Classification**

carried out as part of Natural Language Processing

towards the partial fulfilment for the Award of the Degree of

**BACHELORS OF TECHNOLOGY**

In Computers Science and Engineering

2019-2023

By

Bajjuri Sumanth

-199301168



Under the guidance of

Ms. Shikha Mundra

**Department of Computer Science and Engineering**

**School of Computing and Information Technology**

**Manipal University Jaipur**

**Jaipur, Rajasthan**

Table of Contents

1. Abstract 2

2. Introduction 3

3. Related Work 5

4. Methodology 7

5. Results 15

6. References 17

***Font Type : Aerial and Font size 12 (For all content)***

***Title in BOLD face.***

***Min. required pages are : 15***

***Plagiarism below 20%***

1. **ABSTRACT**

Recent years have seen a huge growth in the use of social media platforms by the people to voice their opinion about a variety of topics, which may lead to comments being ambiguous. For example, sometimes some comments might be misunderstood and may lead to conflict between the reader and the writer. A similar kind of pattern has been seen when it comes to Newspaper headlines or any comment or post on social media. We come across a lot of writing materials throughout the day and while some of them are very straightforward, some can be sarcastic. This may seem offensive to certain groups of readers and result in misunderstanding the subtle nature of humour added to the headlines. Therefore, there is a need to separate the headlines or comments or any writing in general, based on their nature as sarcastic or non-sarcastic. This Project work helps to address this problem by predicting Comments to be sarcastic or not. We applied different Machine learning models and performed a comparative analysis between them. our best performing model achieves a testing accuracy of 68%

Due to increasing technologies in the interactive web applications, there has been a lot of development in E commerce and online social networking activities. The comments or the post always plays a vital role in understanding of the attitude towards a particular topic, product of the online users. Most of the times these comments or posts help the other users to understand the scenario and to take the right decision on the web platform. Machine learning plays a vital role to understand and to estimate the accurate semantics of these posts and comments. Natural language processing is widely used for this, Most of the times natural language processing does not yield much expected results in the classification of these comments due to the complexity in the narration. These complexities generally arise either due to poor narration of the comments or highly sarcastic contents in the comments.

Sarcasm is often used to express a negative opinion using positive or intensified positive words in social media. This intentional ambiguity makes sarcasm detection, an important task of sentiment analysis. Sarcasm detection is considered a binary classification problem wherein both feature-rich traditional models and deep learning models have been successfully built to predict sarcastic comments. In previous research works, models have been built using lexical, semantic and pragmatic features.

1. **INTRODUCTION ( min. 2 pages)**

Humans have a social nature. Social nature means that we interact with each other in positive, friendly ways, and it also means that we know how to manipulate others in a very negative way. Sarcasm, which is both positively funny and negatively nasty, plays an important part in human social interaction. But sometimes it is difficult to detect whether someone is making fun of us with some irony. So to make it easy we built something which helps you in detecting sarcastic text, but before getting into much more detail, let’s define sarcasm:- Sarcasm is “a sharp, bitter, or cutting expression or remark”. The use of irony to mock or convey contempt.

Sarcasm may employ ambivalence, although sarcasm is not necessarily ironic. Most noticeable in spoken word, sarcasm is mainly distinguished by the inflection with which it is spoken and is largely context-dependent. People tend to reply sarcastically for most of the things these days. Sarcasm detection can help in better understanding of the comment and then reply to that comment accordingly. Example if a user has made a sarcastic comment on a product on any e-commerce website then understanding it will help in taking further actions based on that comment. Understanding sarcasm helps us understanding the natural language better. Taking a use case of identifying sarcasm in product reviews on an e-commerce website, understanding sarcasm can help us find the credibility of the review

Sarcasm can actually depend a lot on background information which cannot be very obvious from the raw text itself. The sentence ‘That’s just what I need today’ is a sarcastic phrase given something bad happens at the time when comment was made. Just the phrase it not sarcastic but the context in which it was said makes it a sarcasm. Similarly, the comment on an e-commerce website ‘Must buy! so cheap!!’ is only sarcastic when the product is very heavily priced.

Sarcasm is one of the most difficult and almost impossible to detect by the traditional form of Natural Language Processing applications. Sarcasm detection is a very daunting task for a computer, due to the fact that not most humans can still detect sarcasm effectively. Sarcasm in a language is something that is used to express a bitter taunt or a gibe aimed at something or someone. It is highly challenging to ascertain the sarcasm in normal speech as it can seem to pass by undetected. Most of the sarcasm tends to be contradictory in nature and is usually of a polarizing nature, either objectively negative or objectively positive. This is usually very difficult to detect.

Another characteristic of the sarcastic comments that are very complicated is that most of the sarcastic statements refer to various different events at the same time. This taps into the general knowledge, logical reasoning and anaphoras resolution to fully grasp the underlying meaning of the statement.

Sarcasm analysis is one of the most essential concepts to develop an accurate model for the Natural Language processing and the eventual classification of the comments and limiting the amount of spam that occurs online, especially on the social media websites. Humans tend to be highly hateful and such poisonous acts need to be nipped in the bud and an automated system capable of achieving this is very necessary.

Social media is one of the biggest platforms for people to express their opinion and share information. Many governments and corporate organizations use this data to understand the sentiment of the people towards products, movies, and political events. Sarcasm is a way to convey negative opinions using positive or intensified positive words. On social media people often use sarcasm to express their views, and is inherently difficult to analyze not only for a machine but even for a human. The presence of sarcastic comments has an important effect on sentiment analysis tasks. For example, ‘‘It is a wonderful feeling to carry an expensive phone with short battery-life.’’ is a sarcastic sentence expressing negative sentiment about battery life using positive opinion words like ‘‘wonderful feeling’’. Therefore, sarcasm detection is important to improve the performance of sentiment analysis tasks. Sarcasm detection can be modelled as a binary classification task to predict the given sentence(s) as sarcastic or non-sarcastic. The previous research works in predicting the sarcastic sentences have mainly focused on rule-based and statistical approaches using (a) lexical and pragmatic features (b) presence of punctuations, interjections, sentiment shifts, etc

The primary goal of this project is to create a model that can accurately classify whether a social media comment is sarcastic or not. When given data containing a comment, parent comment, and topic, the model should be able to use this information in order to determine whether the comment sentiment is sarcastic. Such a model could be incredibly important in the field of sentiment analysis as current NLP being used to understand subjective opinions doesn’t account for the fact that people may be sarcastic in their thoughts. This would entail that sentiment analysis cannot account for the intentions of people’s comments and needs to be more contextual. An application of this model would be for natural language bots/virtual assistants that may generate their own sarcastic comments to seem more human-like. The overall impact of this on NLP and sentiment analysis would be very helpful to understanding further conversational, contextual analysis done by AI.

The dataset being used for this project work was taken from open source platform Kaggle. The dataset has 10 columns. The columns namely: ID, comment, date, down, parent comment, score, top, topic, user, label. Here the label indicate whether the comment is sarcastic or not.

.

1. **Related Work**

The sarcasm detection task is a relatively new area of research in natural language processing and it has become a popular area of research in recent years. The main approaches for sarcasm detection have focused on finding lexical and pragmatic features to detect sarcasm in a given sentence. Several approaches have been implemented in literature which exhibit promising ways to discover interesting signals to identify sarcasm. Tepperman et al. perform experiments to detect sarcasm in spoken language, specifically in the expression ‘‘yeah right’’, using spectral, contextual and prosodic cues. Eisterhold et al.It also find that sarcasm can be recognized based on the statement preceding and following the sarcastic statement. Kreuz and Caucci study lexical features and find that the presence of interjections and punctuation play an effective role in identifying sarcasm in the given corpus. Amir et al. Gonzalez-Ibanez et al. It also show that oral or gestural expressions represented by emoticons and special keyboard characters are useful indicators of sarcasm. Davidov et al. D. Davidov use syntactic and pattern-based features to train a sarcasm classifier. Liebrecht et al. consider n-grams along with other signals like intensifiers, exclamations as features and Buschmeier et al. K. Buschmeier, P. Cimiano, and R. Klinger together propose features such as hyperbole, quotation marks and ellipsis as features for this task. Riloff et al.A research demonstrate that the presence of positive sentiment in a negative situation provides an effective clue about sarcasm. Joshi et al. An other paper use, multiple features, including lexical, pragmatic, implicit and explicit incongruity. Bouazizi and Ohtsuki come up with a pattern-based approach to identify sarcasm on Twitter. Rajadesingan et al. An other researcher investigate the psychology aspect behind sarcasm. They present behavioral modelling for sarcasm detection using Twitter data. They recognize different forms of sarcasm and demonstrate the importance of historical tweets provides contextual information and helps in sarcasm detection.

In the recent past, the deep neural network-based approach has gained popularity for the sarcasm detection task. One important reason behind the success of neural networks is the ability to learn latent features automatically. This ability makes neural networks very powerful and enables to explore implicit semantic patterns that are difficult to capture using manually extracted features, such latent features help in the detection of sarcasm. Ghosh and Veale proposes a deep neural network-based sarcasm detection system by stacking a convolutional neural network on top of Long ShortTerm Memory (LSTM). Amir et al. [4] build a deep neural network to focus on contextual features rather than lexical and syntactic patterns. Their network learns user embeddings and helps in improving the context-based sarcasm detection. Zhang et al. M. Zhang, Y. Zhang, and G. Fu together develop a network by combining a bi-directional Gated Recurrent Unit (GRU) with a pooling neural network to detect sarcasm. Poria et al. An other paper come up with a CNN based approach for sarcasm detection. They use a pre-trained CNN for extracting sentiment, emotion and personality features for sarcasm detection. Sulis et al. An other paper explore tweets to understand the difference between sarcasm and irony. They come up with a combination of sentiment, structural and psycholinguistic features to differentiate between irony and sarcasm. Hazarika et al. A paper present a fusion approach, they extract contextual information from the discourse section of a discussion thread, also they use user embeddings to encode stylometric and personality features of users. Their sarcasm detection model shows promising results on a large Reddit corpus.

Machine learning methods and deep neural networks, such as convolutional , recursive, recurrent, and memory networks, have shown good performance for sentiment detection. Knowledge-based methods explore syntactic patterns and employ sarcasm detection resources. However, sarcasm detection currently focuses on extracting features, such as syntactic, surface pattern-based or personality-based features, as well as contextual incongruity. Mishra et al. extracted multimodal cognitive features for sarcasm detection, without modelling the two tasks in a single system. However, recently multitask learning has been successfully applied in many natural language processing tasks, such as implicit discourse relationship identification and key-phrase boundary classification.

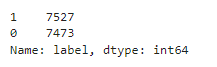
1. **Methodology**

**4.1 Exploratory Data Analysis(EDA) and Data Visualization**

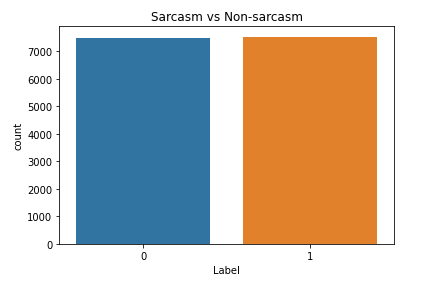
The dataset was taken from Kaggle and it has 10 columns namely ID, comment, date, down, parent comment, score, top, topic, user, label. The total entries are 15000 and no null values are there. There are almost equal number of non-sarcastic and sarcastic comments. The column label in the dataset denoted weather the comment is sarcastic or not.

The value counts for sarcastic comments and non-sarcastic comments are 7527 and 7473

The image below describes the same



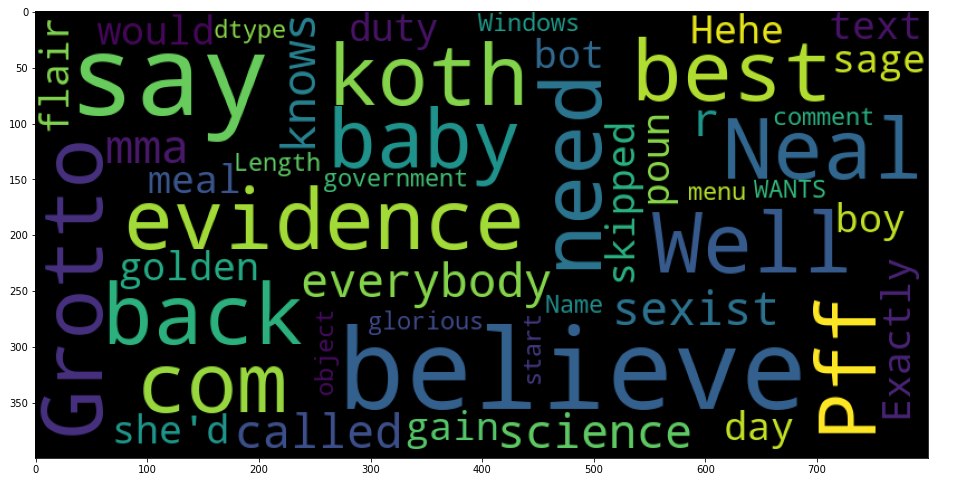
The bar graph below represents the label count of sarcastic and non sarcastic comments.



*Fig 1. Number of sarcastic and non sarcastic comments*

In order to view the most common words in spam and ham messages we made a word cloud of them.

The picture below represents the most occurred words in the comment column.



*Fig 2. Comment Word Cloud*

As observed, words like evidence, believe, say etc. are often used in the comments column.

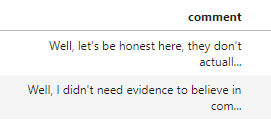
**4.2 Data Preprocessing and Data Cleaning**

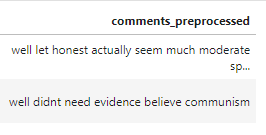
The very first step I followed is to drop all the unnecessary columns in the data set. The dropped columns are namely ID, name, down, score, top. We dropped these columns because these columns have no significance with respect to the project objective. The remaining columns after dropping the unwanted columns are comments, parent comment, topic , user, label.

Then we performed the text cleaning.The removal of punctuation marks and conversion of each letter to lower case has been done using regular expression. Each word was tokenized. Stemming was performed on the corpus for logistic regression. Stem is the root form of a word which does not necessarily belong to the language.

After the cleaning and preprocessing the comments we added a new column to the dataframe. We named the new column as comment\_preprocessed.

Below we can actually find the difference between the actual comment present in the data and the preprocessed comment.





**4.3 TF-IDF Vectorizer**

TF-IDF is an information retrieval and information extraction subtask which aims to express the importance of a word to a document which is part of a colection of documents which we usually name a corpus. It is usually used by some search engines to help them obtain better results which are more relevant to a specific query. we can implement it in Python by using the Scikit-Learn library.

TF-IDF stands for Term Frequency — Inverse Document Frequency and is a statistic that aims to better define how important a word is for a document, while also taking into account the relation to other documents from the same corpus.

This is performed by looking at how many times a word appears into a document while also paying attention to how many times the same word appears in other documents in the corpus.

TF-IDF is an efficient algorithm to find the relevant words related to a particular inquiry or search term. It consists of two part-1)Term frequency part as stated above. 2)Inverse Document Frequency- It is a way to measure the importance of a word or term and is computed by calculating the logarithmic value of the total number of documents to number of documents with that word or term. The TF-IDF weight of every word in a document is computed by the formula-

TF-IDF=(1+log(TF(t)))(log(N/TF(t))

1. **Classification Algorithms**

**5.1 Logistic Regression**

Logistic regression is a type of supervised machine learning algorithm which uses the sigmoid function(logistic function) as its base. The sigmoid function can be written as-

P=1/(1+e^-y)

The equation for logistic regression is stated as below-

y = e^(b0 + b1\*x) / (1 + e^(b0 + b1\*x))

Here,

x= input value

y= output value(between 0 and 1)

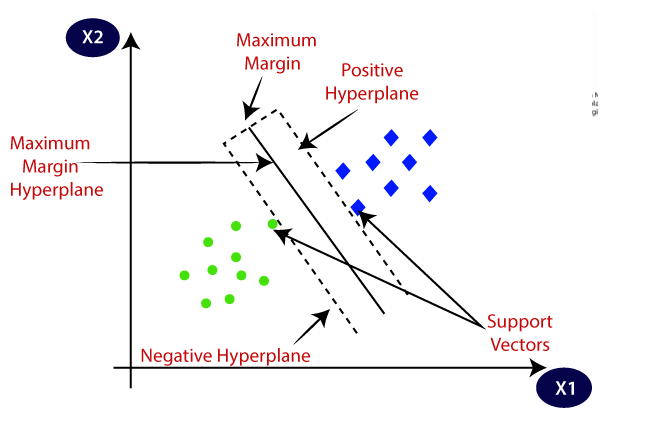
b0=bias term or intercept value

b1=single input value’s coefficient

**5.2 SVM**

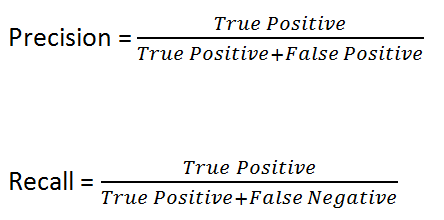
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

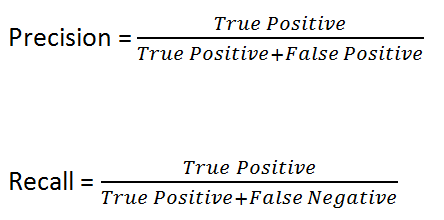
SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

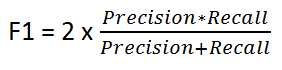


**6.Results**

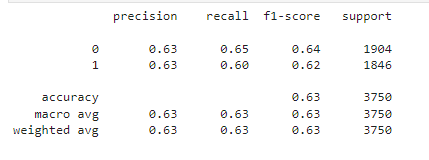
For each of the 2 algorithms used in methodology the accuracy, precision, recall and F1 score have been calculated. The formula for precision ,recall and F1 score are stated below-



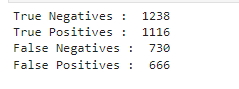




6.1 Logstic Regression

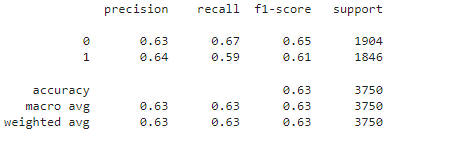


The table represents the precision, recall, f1-score, support for logistic regression method.

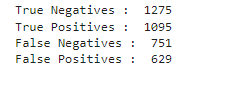


The table represents the True Negatives, positives and False Negatives, positives for logistic regression.

6.2 SVM



The table represents the precision, recall, f1-score, support for SVM model.



The table represents the True Negatives, positives and False Negatives, positives for SVM model

1. **Conclusion and Future scope : { 1 page }**

Our experiment proves that we almost got the same accuracy score for both Logistic Regression model and SVM model but when we compare the precision for label 1 in SVM is 0.01 more than that of Logistic Regression model. On a slight note we can also conclude SVM outperforms logistic regression in our model.

**7. References**

[1] Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM -AVINASH KUMAR , VISHNU TEJA NARAPAREDDY , VEERUBHOTLA ADITYA SRIKANTH ,ARUNA MALAPATI , AND LALITA BHANU MURTHY NETI Birla Institute of Technology and Science at Pilani, Hyderabad 500078, India Corresponding author: Avinash Kumar (p20150507@hyderabad.bits-pilani.ac.

[2] Modified framework for sarcasm detection and classification in sentiment analysis - Mohd Suhairi Md Suhaimin1, Mohd Hanafi Ahmad Hijazi2, Rayner Alfred3, Frans Coenen4 1,2,3Faculty of Computing and Informatics, Universiti Malaysia Sabah, Malaysia 1Kuching Community College, Ministry of Education, Malaysia 4Department of Computer Science, University of Liverpool, United Kingdom

[3] Sentiment and Sarcasm Classification With Multitask Learning - Navonil Majumder Instituto Politecnico Nacional, Soujanya Poria NTU, Haiyun Peng NTU, Niyati Chhaya Adobe Research, Erik Cambria NTU, Alexander Gelbukh Instituto Politecnico Nacional