**Abstract –** Sarcasm means expressing views that could hurt someone’s emotions and criticize in a humorous manner.Sarcasm text identification is an important task in Natural Language Processing (NLP) applications like sentimental analysis. The paper proposes a novel approach using Bayesian classifier that identifies the sarcasm automatically from a given text. Sarcasm detection has been done for languages like English, Spanish and Mexican. Existing methods have used cue words, use of expressions to classify sarcastic and non-sarcastic texts. The proposed approach uses Naive Bayes classifier and makes use of bi-grams to detect positive and negative sentiments to detect sarcasm. A data set of 42836 tweets has been used to train the classifier. The proposed approach gives an accuracy of 81%.

Sarcasm Detection Model for Text Using Naive Bayes Classifier

Thamilthaen S \* Subalalitha C.N \* Department of Computer Science and Engineering \* SRM IST, Kattankulathur, Chennai.

E-mail: [thamil0802@gmail.com] [subalalitha@gmail.com] SRM IST, KTR CSE.

Keywords: Sarcasm Detection, Text Processing, Machine Learning, Twitter, Training dataset, Testing dataset, Sentiment Analysis, Natural Language Processing (NLP).

**1. INTRODUCTION**

Sarcasm implies specifying opposite of reality. It is also termed as verbal incongruity. To Detect Sarcasm it has been a difficult assignment for humans. Sarcasm identification becomes an essential component in sentiment to understand the people emotions and thoughts. Detecting sarcasm can be one of the methodologies that could confirm whether the assessment communicated by the client is humorous or not. In Sentiment Analysis (SA) to detect sarcasm is too difficult if dialects are considered. Today, a web-based social networking like Twitter, Facebook have enabled clients to express their feeling through messages. Sarcasm is a type of dialect that is present in their messages in the inverse way. Sarcasm while talking could be less demanding to recognize on the grounds that location ends up straighter forward through the tone, pitch, articulation the discovery of the equivalent ends up harder. This paper proposes a Sarcasm detection approach using Naive Bayes Classifier and sentimental analysis as the predominant features.

The existing works have used Twitter data set for Sarcasm detection. Linguistic Inquiry and Word Count (LIWC) is the metric that is used for detecting sarcasm. Psycholinguistics features of the texts used for sarcasm detection. Psycholinguistics features such as human language, expression, speech, nature of human. Machine Learning classification algorithms available in WEKA tool have been used. In text classification SMO, classifier perform better and C4.5 decision tree (J48), Bayes Network learning algorithm (Bayes Net) classifiers is used for building the classifier model.

Natural Language Processing (NLP) enables machines to peruse and understand human dialects spoken automatically. NLP also deals with semantics and pragmatics. Semantics is one of the levels of NLP which gives importance to meaning of words and expressions. Sarcasm detection also involves semantic level of processing the texts. Various levels of features such as, lexical, syntactic and semantic are used. A Semantic features incorporates a group of features that boost the words meaning and make it more intense. The proposed approach makes use of bi-grams to detect the sentiment which in turn detects sarcasm more precisely. Bi-gram is a two word sequence which predominantly occur aiding in understanding the context. . Example 1 shows the example of bigrams.

**Example 1:** ‘It felt so good to be home’ ‘it felt,’ ‘felt so,’ ‘so good,’ ‘good to’, ‘to be’, ‘be home’, are bi-gram features.

The rest of the paper is sorted out as following sections. Section 2 includes the existing work related to the proposed approach. Section 3 portrays the proposed sarcasm detection model. Section 4 demonstrates the experimental outcome of the proposed approach. Section 5 concludes the paper and presents the future research works that could be built on top of the proposed model.

1. **RELATED WORK**

Ren, Yafeng, Donghong Ji, and Han Ren have introduced sarcasm detection and neural network models [1]. Sarcasm is a task to detect text classification problem. The Neural network models is to verify the dataset effectiveness of contextual information.

Raghav, Shalini, and Ela Kumar, have proposed display feature representations of sentences, expressions and documents from distribute word representation [2]. They have used Convolution Neural Networks (CNN) model.

del Pilar Salas-Zarate, Maria, et al, have proposed to Detect Sarcasm that can be formulated as the text classiﬁcation. To ﬁnd out the sarcasm detection problem whether the sentences within the text are sarcasm or non-sarcasm [3]. The algorithm that will extract out the words having multiple meaning to deal with sarcasm detection. They have proposed feature extraction approach some bag of-words model to demonstrate while others investigate increasingly modern highlights, for example, semantic and full of feeling highlights or punctuation marks. A lot of satirical tweets from Mexico and another from Spain so as to decide if there is a huge diverse between the examples of highlights used to detect satire.

Jain, Tanya, et al. have proposed a pragmatic classifier which goes for distinguishing contextualized sarcasm which are commonly worried about the discussion between two people [4] predictive methods to classify pragmatic for both sentiment and polarity.

Prasad, Anukarsh G., et al. have proposed different slang and emoji dictionaries for classification which may result in more accurate, which can be seen by the outcomes displayed in this paper[6].

Erik F. and Niklas W., is the most recent advancement of the sarcasm identification and irony identification on twitter sentiment analysis [13]. It uses existent algorithms such as SVM and Decision tree classifiers. It performs Tokenization, Stemming and Lemmatization, POS-tagging, Feature selection and then model evaluation.

Bouazizi, Mondher, and Tomoaki Otsuki Ohtsuki aim to classify the sarcasm tweets by extracting a set of features, to refer a training set and machine learning algorithms to perform the classification [7]. The features are extracted in a way that makes use of different components of the tweet, and covers different types of sarcasm.

Mukherjee, Shubhadeep, and Pradip Kumar Bala, The effective identification of sarcasm, both the substance just as composing style of the creator assumes a significant job [8]. Through the classification algorithms, to identify a set of features that capture authorial style. An accuracy of more than 70 percent with the features they have proposed in study, which is on the higher side in sarcasm detection.

Joshi, Aditya, P. B. and Carman, M. J, perform sarcasm detection using sentiment analysis. The method rule based, deep learning is used to detect sarcasm [9]. The multiple datasets is to classify the sarcasm labelled datasets and manual datasets. The experimental dataset is split up into two categories: training dataset and testing dataset. Logistic regression and SVM is to result the accuracy level classified sarcasm.

Dave et.al, have detected sarcastic sentences using sentiment analysis to classify difficult task. The different techniques is used to find sarcastic sentences in machine learning. TF-IDF is to calculate the score in feature extraction [10]. In feature selection it approach lexicon and statistical approaches. SVM classifier is to trained Bag-Of-Words and TF-IDF to measure frequency. The accuracy level of sarcasm detection is 50%.

Forslid, Erik, and Niklas Wikén has fetched various features from different irony text patterns thereby proposed the training classifier. The data were collected from amazon and twitter datasets. Text classification implement by supervised learning, unsupervised learning and semi-supervised learning. The results of classification had an accuracy over our proposed baseline on 50%. The datasets could find sarcastic texts. SVM algorithm classification accuracy approach is conducted for dataset classification.

Ashwin Rajadesingan et.al introduced a sarcasm identification using supervised learning algorithm to predict sarcastic sentences [15]. It precisely predicts the sarcasm identification in sites like Twitter. The research [19] suggests user uses sarcasm with twitter. A behavioural model to detect non-literal forms of language to predict humour sense. Using machine learning algorithms such features are being evaluated.

The proposed approach makes use of 42836 tweets as a data set and uses a Naive Bayes classifier to build the sarcasm detection model. The proposed approach gives a higher accuracy than that of the existing techniques that uses Naive Bayes classifier by the finding the sentiment using the bi-grams as the features.

The next section explains about the proposed approach in detail.

**3. PROPOSED WORK**

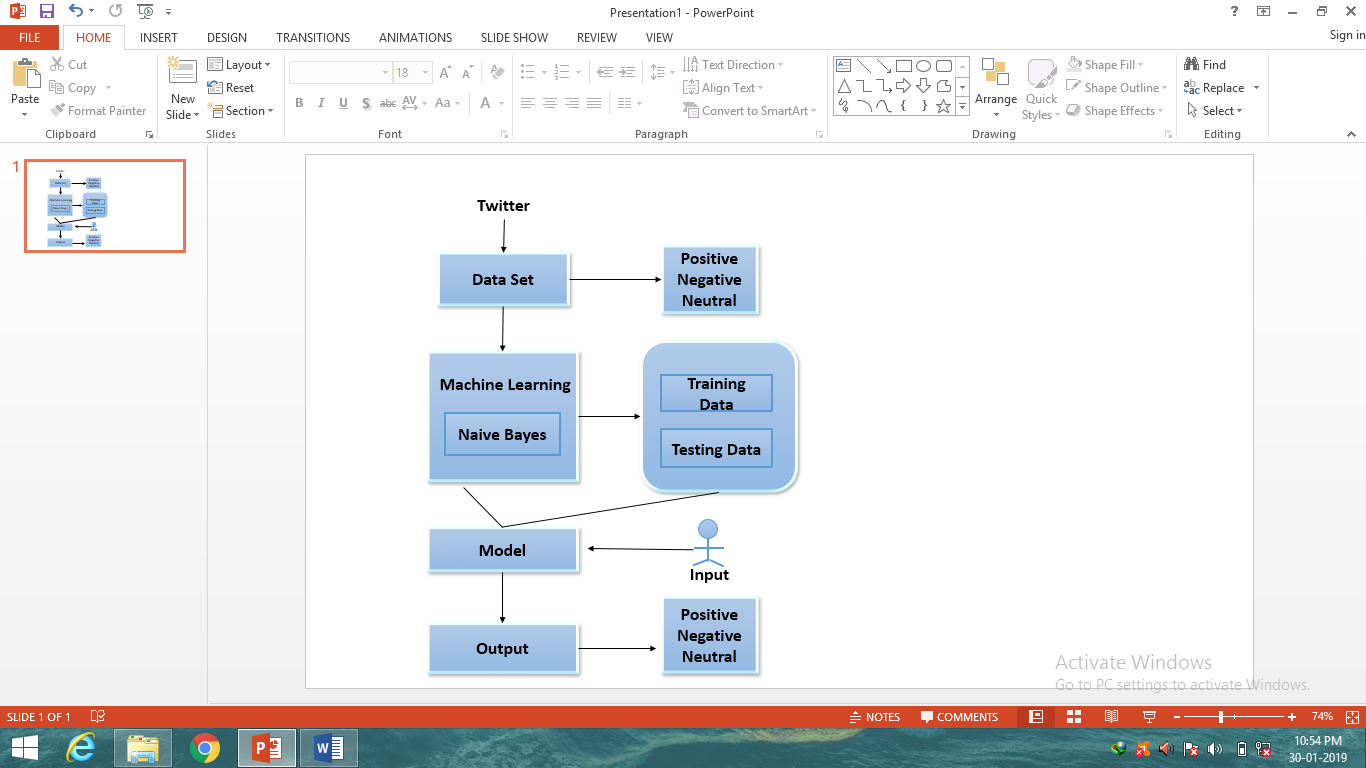
The proposed work architecture is shown in Figure 1. The data set is collected, pre-processed, features are extracted and the training model is built.

The detailed description of each of these steps are given in this section.

The data collected from twitter. The training sentences were tagged as positive, negative and neutral sentences.

From the datasets we have gained we are trying to train our model so that test live streaming tweets from Twitter API to detect weather they are sarcastic or not.

To classify the sarcasm detection of given text using Naive Bayes classifier. Naive Bayes classifier trains the labelled data sets and tests the model using test data. The test outcome reveals that the dataset successfully yields in higher classification performance accuracy.



**Figure.1 Proposed Overall Architecture**

**3.1 Dataset Collection**

Training Dataset: has been downloaded from GitHub. It contains 42836 tweets for training. The link is given below:

1. https://github.com/surajr/SarcasmDetection/Dataset
2. <https://github.com/Mandar_Kulkarni/SarcasmDetection/Dataset>

**3.2 Pre-processing**

The collected tweets are cleaned by tokenization and by running a Parts -Of-Speech Tagger to identify the Parts of speech [13]. NLTK tagger has been used. Stemming is done by using the Porter Stemmer Algorithm. Data pre-processing consider under Tokenization and Normalization.

**Tokenization**

Tokenization is a process of streaming text data which splits longer strings into smaller pieces. A text can be tokenized into sentences, sentence can be tokenized into words.

**Normalization**

Normalization is a process to convert text to upper or lower case, removing punctuation, replace numbers to words, stop words, stemming and lemmatization. Stemming is the process eliminating suffix, prefixes and infixes. It is the process of reducing words to their root forms such as mapping a group of words to the same stem word.

The training data is tokenized and normalized using stemming. Then the features are extracted which is explained in the next section. It includes an example 2,

**Example 2:** The words “waiting”, “waited”, “waits” are stemmed as “wait”.

**3.3 Feature Extraction**

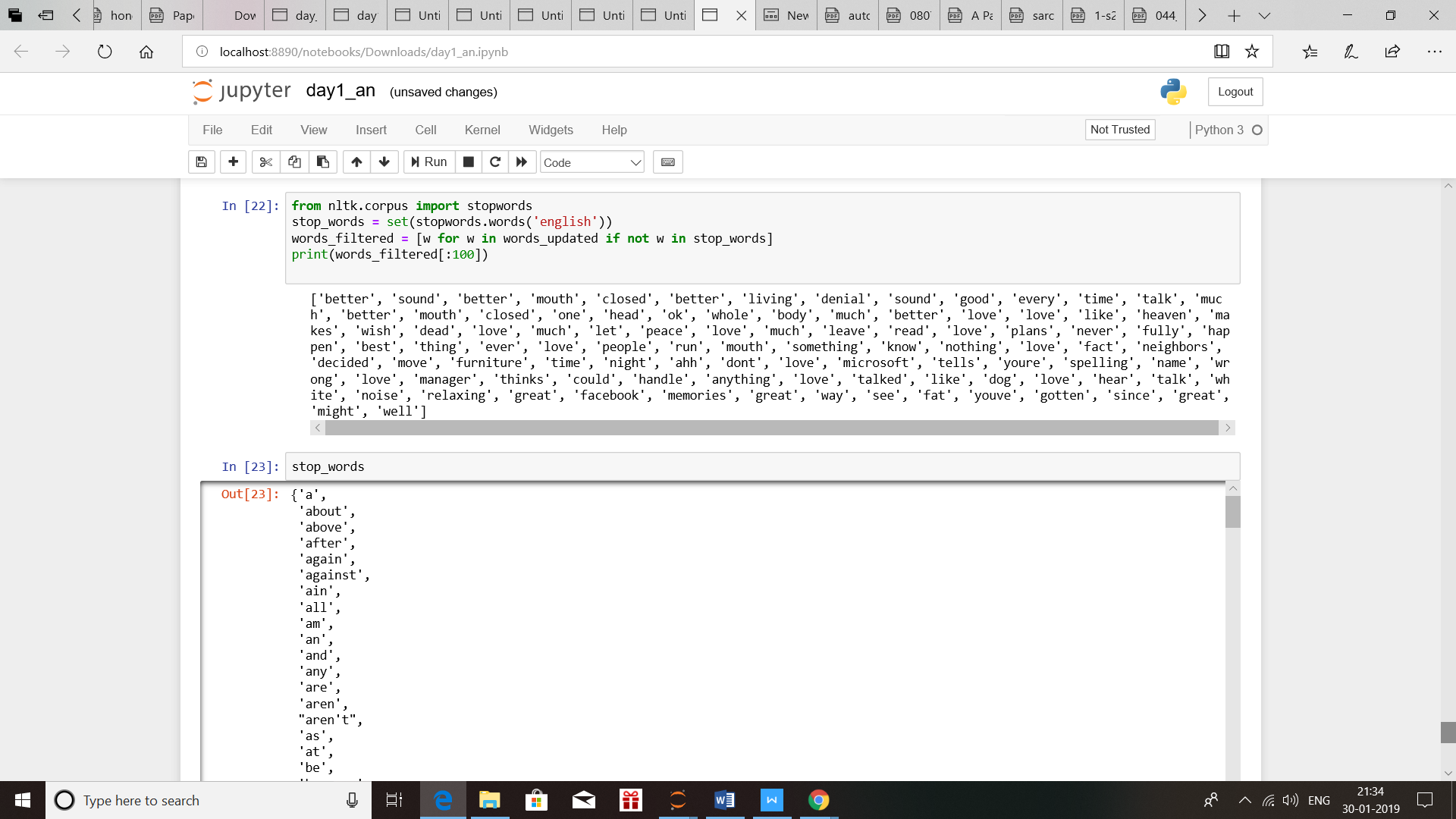
**N-Grams**

Individual tokens uni-grams and bi-grams are placed into a binary feature dictionary. Bi-grams are extracted using the same library and are deﬁned as pairs of words that typically go together. Bi-gram Models is to order the words by analysing the context of each word by pairs. Bi-grams are highlights comprising of two sets of words in the text it's appeared in the example 3,

**Example 3**, in the sentence ‘It felt so happy’ ‘it felt,’ ‘felt so,’ ‘so happy,’ are bi-gram features.

**Stop words**

Stop words are those words which are sifted through before further handling of content, since there words contribute little and their removal will reduce the dimension it’s shown in the figure.2.



**Figure.2 Feature Extraction for a stop words**

**Bag of Words**

The bag-of-words is a model, a content (for example a sentence or a record) is represented as the bag of words, and even word order yet keeping multiplicity. The bag-of-words is a model in Information Retrieval (IR) and Natural Language Processing (NLP). In the proposed approach, the bag of words are used to label the sentiments of the training data [13]. An example of the bag of words used as features for each label of sentiment like positive, negative and neutral is shown in example 4,

**Example 4:**

**"**Swetha likes to watch movies. Harini likes movies too**"**.

**"**Swetha also likes to watch basketball games**"**.

**Swetha-2; likes-3; to-2; watch-2; movies-2;**

**Harini-1; too-1; also-1; basketball-1; games-1**

**Feature list:**

Positive - better, great, fun, glad, love, terrific, happy.

Negative - hate, sad, not.

Neutral - famous, the, actors, did, outstanding.

**3.4 Feature Selection**

Naive Bayes (NB) classifier is used to select the predominant features from the bag of words initially used for each sentiment. NB speaks to each class with a probabilistic run down and ﬁnds the no doubt class for every model it is approached to group. A few analysts have stressed the issue of excess characteristics, just as the upsides of highlight determination for the Naive Bayesian classiﬁer. This technique for highlight choice prompts improved execution of the Naive Bayesian classiﬁer

**Naive Bayes Classifier Algorithm**

Naive Bayes classiﬁer has been used to train the data.

The prior probabilities are calculated for each bag of words across their label using the equation (1), conditional probabilities for training and testing using in the equation (2) and calculate the posterior probability in equation (3)

**1. Calculation of prior probabilities**



**2. Calculation of conditional probabilities**





1. **Calculation of posterior probability**





**Sarcasm Detection:**

The sentiment analysis done as clarified in the past section analyses the polarity of the sentence in terms of its emotion. Sarcasm comes into picture when there exist a mixture of polarities, while one polarity may dominate over the other. Also there is a chance of sarcasm existing when the two polarities are equally present. This is calculated by finding the percentage of each sentiment class in each test sentence. It includes an example 5,

**Example 5:**

Love is like Heaven. It makes me wish I was dead.

You sound better with your mouth closed.

The silent treatment hurts so much that I rather argue to have you just speak to me.

I'm so fed up with everything to the bud man’s house as we speak

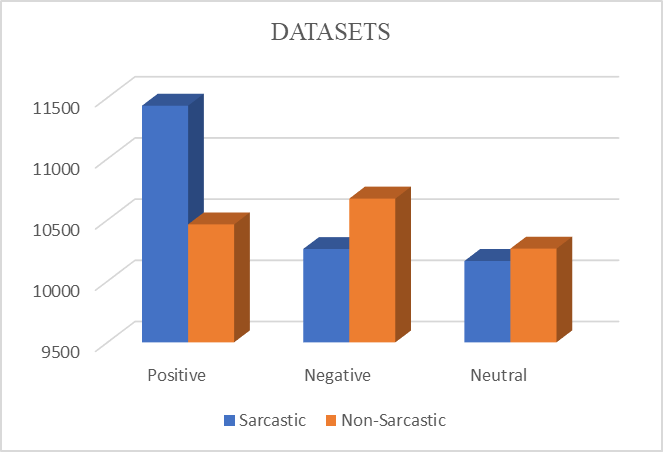
A sarcasm in the tweets arises from a positive sentiment word (e.g., better, love, enjoyed) and negative sentiment words are (e.g., fed up, hurts, hate, being ignored).

**4. MODEL EVALUATION AND EXPERIMENTAL RESULTS**

The data set of 42836 tweets are divided into training and test set. 70% of the tweets are taken as training set and the remaining as test set. The training set are labelled and the statistics are shown in Table 1

**Table 1: No of sarcastic sentence in the data set**

|  |  |  |
| --- | --- | --- |
|  | **Sarcastic** | **Non-Sarcastic** |
| **Positive** | 11434 | 10464 |
| **Negative** | 10263 | 10675 |
| **Neutral** | 10165 | 10266 |

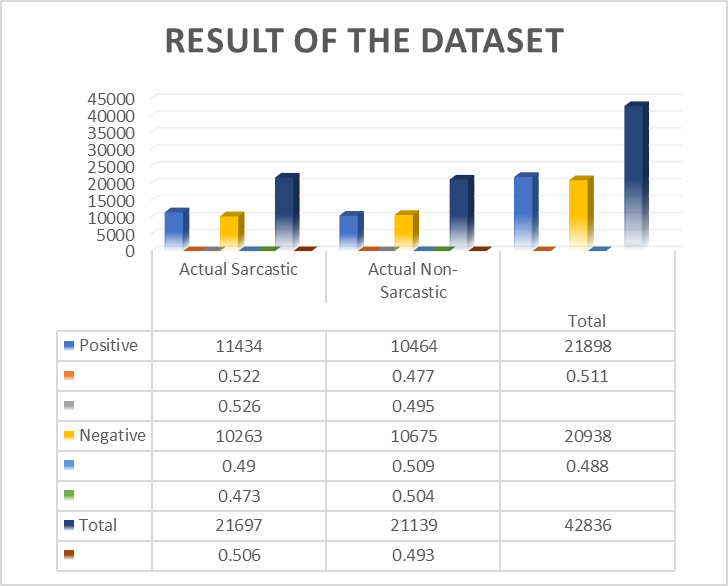


**Figure 3: Data sets**

**Table 2: Result on the data set with Naive Bayes Classifier**

|  |  |  |  |
| --- | --- | --- | --- |
| **Predicted** | **Actual**  **Sarcastic** | **Actual**  **Non-Sarcastic** | **Total** |
| **Positive** | **11434**  0.522  0.526 | **10464**  0.477  0.495 | **21898**  0.511 |
| **Negative** | **10263**  0.490  0.473 | **10675**  0.509  0.504 | **20938**  0.488 |
| **Total** | **21697**  0.506 | **21139**  0.493 | **42836** |

Table 2 shows, the 21697 patterns that are sarcastic, 11434 were Positive Sarcastic 10464 were incorrectly predict as Non-Sarcastic. Similarly, the 10263 were Negative Sarcastic 10675 were incorrectly predict as Non-Sarcastic.

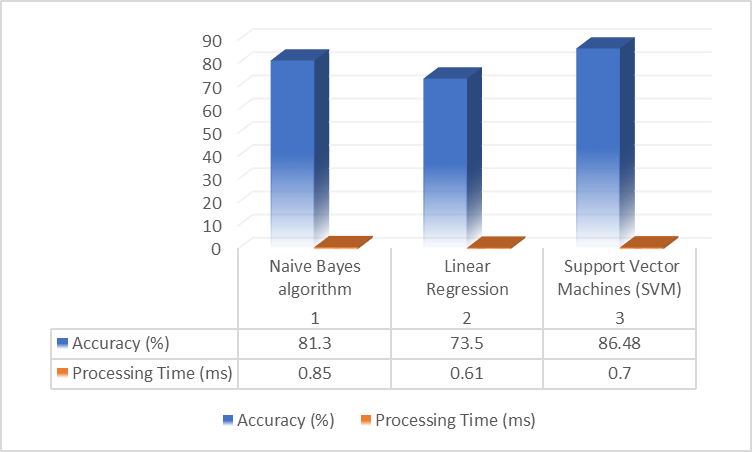


**Figure 4: Performance of Techniques based Feature selection**

Figure.4 mentioned above the tweets which are classiﬁed as sarcastic or non-sarcastic sentence. It shows the scoring of positive and negative results.

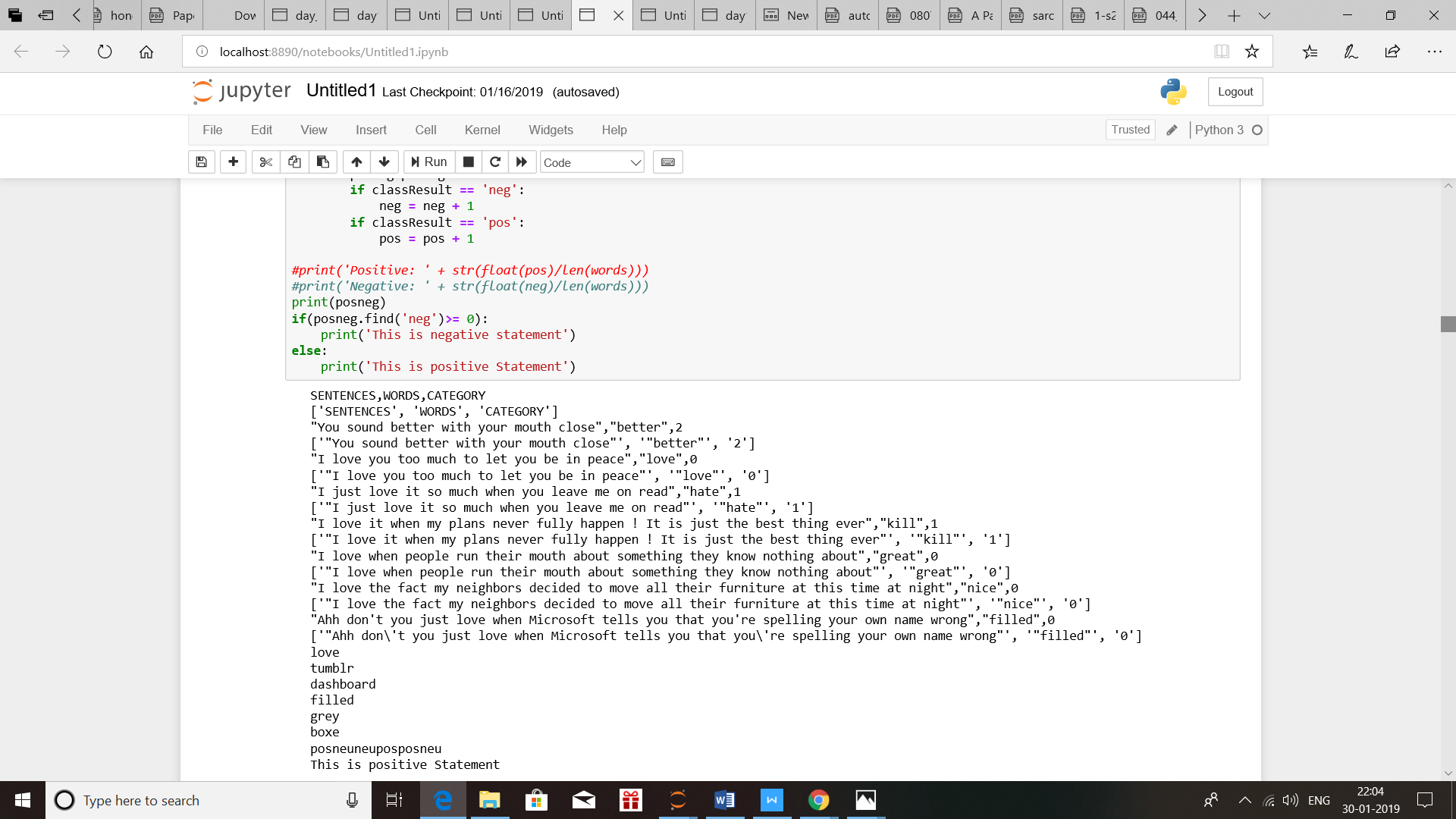
**Table 3: Performance of classification techniques**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **No. of Techniques** | **Accuracy (%)** | **Processing Time (ms)** |
| 1 | Naive Bayes algorithm | 81.3 | 0.85 |
| 2 | Linear Regression | 73.5 | 0.61 |
| 3 | Support Vector Machines (SVM) | 86.48 | 0.7 |



**Figure 5: Performing of Techniques based Classification**

Figure.5 mentioned above, to identify sarcasm detection to implement in Naïve Bayesian algorithm comparing with different techniques like Support Vector Machine (SVM) algorithm, Machine Learning (ML) algorithm, Sentiment Analysis. The results are high and effective accuracy performance compared to rest of the techniques. This is due to the features considered and the factors taken up in polarity analysis for sarcasm detection.



**Figure 6: Features selection from training data set**

Figure.6 shows the result of training data set. All highlights were separated from each case, paying little to its area. The classiﬁcation estimation to pick how to best exploit this supplemental data to use Naive Bayes, for earlier work. The performance and system stability metric can be evaluated and analysed using below mentioned parameters. The techniques proposed for computing performance are based upon feature selection and Naive Bayesian classification.

**5. CONCLUSION**

The paper proposes a Sarcasm detection model from texts. Sarcasm recognition is the undertaking of marking the content as positive, negative or neutral. Our essential objective is to break down a tweet and classify as sarcasm or not. The datasets have gained to prepare our model so test live gushing tweets from Twitter API to distinguish weather sarcastic or not. This will bring about getting result in obtaining accurate sentiment analysis and opinion mining. It assesses assorted element types for sentiment extraction including feelings, words and n-grams, confirming that each feature type adds to the notion of grouping structure in text classification. In this task we have utilized a dataset containing around 80,000 tweets to prepare our model, we have pre-processed the raw data information utilizing NLP. To create a dictionary of the words, we have done it for word, where we trained out Tf-idf. The dataset for testing contained around 2000 tweets. On the testing dataset, after running it in our model. A classifier based on the Naive Bayes algorithm using Natural Language Toolkit. The most common words in a sentence is removing using stop words and adding a PyDictionary. PyDictionary is a Dictionary Module for Python to get a meaning, synonyms words and Antonyms of words. It utilizes for getting meaning, Google for translations, and thesaurus.com for getting synonyms words and antonyms. So as to discover the likelihood for a name data's, this calculation first uses the Bayes standard to express probabilities. The proposed approach can further be extended using many other high level semantic features like context features which are not necessarily in sequence captured by a neural network model like Recurrent Neural Networks (RNN).

**REFERENCE**

1. Ren, Yafeng, Donghong Ji, and Han Ren. "Context-augmented convolutional neural networks for twitter sarcasm detection." Neurocomputing (2018).
2. Raghav, Shalini, and Ela Kumar. "Review of automatic sarcasm detection." Telecommunication and Networks (TEL-NET), 2017 2nd International Conference on. IEEE, 2017.
3. del Pilar Salas-Zárate, María, et al. "Automatic detection of satire in Twitter: A psycholinguistic-based approach." Knowledge-Based Systems 128 (2017): 20-33.
4. Jain, Tanya, et al. "Sarcasm detection of tweets: A comparative study." Contemporary Computing (IC3), 2017 Tenth International Conference on. IEEE, 2017.
5. Chaudhari, Pranali, and Chaitali Chandankhede. "Literature survey of sarcasm detection." Wireless Communications, Signal Processing and Networking (WiSPNET), 2017 International Conference on. IEEE, 2017.
6. Prasad, Anukarsh G., et al. "Sentiment analysis for sarcasm detection on streaming short text data." Knowledge Engineering and Applications (ICKEA), 2017 2nd International Conference on. IEEE, 2017.
7. Bouazizi, Mondher, and Tomoaki Otsuki Ohtsuki. "A pattern-based approach for sarcasm detection on Twitter." IEEE Access 4 (2016): 5477-5488.
8. Mukherjee, Shubhadeep, and Pradip Kumar Bala. "Detecting sarcasm in customer tweets: an NLP based approach." Industrial Management & Data Systems 117.6 (2017): 1109-1126.
9. Joshi, Aditya, Pushpak Bhattacharyya, and Mark J. Carman. "Automatic sarcasm detection: A survey." ACM Computing Surveys (CSUR) 50.5 (2017): 73.
10. Dave, Anandkumar D., and Nikita P. Desai. "A comprehensive study of classification techniques for sarcasm detection on textual data." Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on. IEEE, 2016.
11. Suhaimin, Mohd Suhairi Md, et al. "Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts." Information Technology (ICIT), 2017 8th International Conference on. IEEE, 2017.
12. Bamman, David, and Noah A. Smith. "Contextualized Sarcasm Detection on Twitter." ICWSM 2 (2015): 15.
13. Forslid, Erik, and Niklas Wikén. "Automatic irony-and sarcasm detection in Social media." (2015).
14. Agarwal, Basant, et al. "Concept-level sentiment analysis with dependency-based semantic parsing: a novel approach." Cognitive Computation 7.4 (2015): 487-499.
15. Rajadesingan, Ashwin, Reza Zafarani, and Huan Liu. "Sarcasm detection on twitter: A behavioral modeling approach." Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. ACM, 2015.
16. Walaa Medhat, Ahmed Hassan, Hoda Korashy, "Sentiment analysis algorithms and applications: A survey", Ain Shams Engineering Journal, vol. 5, no. 4, pp. 1093-1113, 2014.
17. Abercrombie, Gavin, and Dirk Hovy. "Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations." Proceedings of the ACL 2016 Student Research Workshop. 2016.
18. Bouazizi, Mondher, and Tomoaki Ohtsuki. "Sentiment analysis: From binary to multi-class classification: A pattern-based approach for multi-class sentiment analysis in Twitter." Communications (ICC), 2016 IEEE International Conference on. IEEE, 2016.
19. Van Hee, Cynthia, Els Lefever, and Véronique Hoste. "Exploring the fine-grained analysis and automatic detection of irony on Twitter." Language Resources and Evaluation (2018): 1-25.



**Thamilthaen S**

**M.Tech**

**Department of Computer Science and Engineering**

**SRM Institute of Science and Technology**

**Kattankulathur Campus**

**Chennai**



**Subalalitha C.N**

**Associate Professor**

**Department of Computer Science and Engineering**

**SRM Institute of Science and Technology**

**Kattankulathur Campus**

**Chennai**