
Sarcasm Detection From Twitter Database Using Text Mining Algorithms**Tamanna Siddiqui¹, Abdullah Yahya Abdullah Amer²**¹Associate professor, Department of Computer Science, Aligarh Muslim University, Aligarh, Aligarh, Uttar Pradesh, India²Research Scholar, Department of Computer Science, Aligarh Muslim University, Aligarh, Aligarh, Uttar Pradesh, India**Article History:** Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract

Sarcasm is well-defined as a cutting, frequently sarcastic remark intended to fast ridicule or dislike. Irony detection is the assignment of fittingly labeling the text as 'Sarcasm' or 'non-Sarcasm.' There is a challenging task owing to the deficiency of facial expressions and intonation in the text. Social media and micro-blogging websites are extensively explored for getting the information to extract the opinion of the target because a huge of text data existence is put out into the open field into social media like Twitter. Such large, openly available text data could be utilized for a variety of researches. Here we applied text data set for classifying Sarcasm and experiments have been made from the textual data extracted from the Twitter data set. Text data set downloaded from Kaggle, including 1984 tweets that collected from Twitter. These data already have labels here. In this paper, we apply these data to train our model Classifiers for different algorithms to see the ability of model machine learning to recognize sarcasm and non-sarcasm through a set of the process start by text pre-processing feature extraction (TF-IDF) and apply different classification algorithms, such as Decision Tree classifier, Multinomial Naïve Bayes Classifier, Support vector machines, and Logistic Regression classifier. Then tuning a model fitting the best results, we get in (TF-IDF) we achieve 0.94% in Multinomial NB, Decision Tree Classifier we achieve 0.93%, Logistic Regression we achieve 0.97%, and Support vector machines (SVM) we achieve 0.42%. All these result models were improved, except the SVM model has the lowest accuracy. The results were extracted, and the evaluation of the results has been proved above to be good in accuracy for identifying sarcastic impressions of people.

Keywords: Classification; Sarcasm; Social Media; Data Analysis; Twitter; Machine Learning Algorithms; detection.

1. Introduction

Platforms as Microblog are easy tools for a person to express his or her opinions, thoughts, while, sarcasm is a sophisticated and informative human nature to express sentimentally refined viewpoints in an implicit way. The irony is phenomenally expressed in various online textual expressions. Here the Sarcasm is widely considered to be the linguistic phenomenon to express significant sentimental tasks predominantly found in online content writings. Sarcasm is predominantly found in social media platforms like Twitter, Facebook, Instagram, and others. Sarcasm is popularly known as the eloquent process of denoting the expressions or words for uttering an opposite suggestion from the words they used in accordance with the context. Sarcasm is used predominantly to express two-faced quality illustrations, i.e., comic and mean. Here sentimental expression is widely used on twitter, a social expression platform, and more attention from the researchers. Twitter is growing with more popularity to reveal the manifestations of the common men to leaders. The twitter is rich with sarcasm expressions to show their genuine testimonials with an expression of negative feelings using positive words to criticize the people (Hayat et al., 2019).

Sarcasm is rich with features related to sentiment, punctuation, syntactic, and various expressional values. These features are the most useful elements in the training data. These features are predominantly used for the classification of different algorithms (Siddiqui & Alam, 2009). Automatic sarcasm detection is widely used by researchers by using feature selection(Siddiqui et al., 2019). The researchers have used various applications for detecting Sarcasm applying the data collected from social media. Detection of Sarcasm is possible from the online postings with capital letters, emoticons, and preceded by exclamation marks, and it is predominantly used for sentiment analysis (Liu et al., 2014). During an automatic detection of Sarcasm through extract features, the data process plays a predominant role. The identifying of Sarcasm can be made with the presence of hashtags within the sentence. Once the words preceded by the hashtags are identified, these words are taken as input data for processing the data analysis (Ahmad et al., 2018). The input data can be taken from the data extracted from different social media platforms(Siddiqui et al., 2019). These platforms are, namely, the Twitter data set. The sample data can be made from this data set, and the analysis is usually done from this data set by performing the sarcastic sentences and non-sarcastic sentences (Rajeswari & Shanthibala, 2018).

While doing the classification of data from the data sets, the word level of polarity is predominant. It can be performed with the polarity judgment dictionary. The sentimental summarization can be done with the negative meaning and positive meaning of the word distinguished from the sample data set, preceded by the hashtags. The polarity detector is going to play a vital role in determining the word as negative by mistake that text has distinguished as a positive meaning. The sarcastic texts are influencing the classification accuracy of the polarity detector(Yahya et al., 2020). The polarity detector also identifies the motions placed before the proposed phrases

based on the number of words with the emotion included in the terms. The main goal of the classification and bifurcation is to identify the words with negative expression and express the positive meaning or expression in the sentences available in the sample data taken for the process (A. Joshi et al., 2017). The rest of the paper is prepared as follows: Section two displays Literature Survey. Section Three presents supervised classification Machine Learning Techniques. Section four the design and modeling of our Proposed Methodology equations to measure the performance text data sets, pre-processing, feature extraction, Training, Test Data Set. Section four displays equations to measure the performance of the proposed model and Algorithm Algorithms Twitter Extract Sentiment. In Section five, we discussion of the results of the experiments, and the confusion Matrix. Finally the conclusion and future research in In Section six.

2. Literature Survey

The research work done by Peng Liu et al. has focused on the imbalanced classification to perform the detection of Sarcasm from social media information. In this research work, a novel multi-strategy ensemble learning method is implemented to handle imbalance problems in finding the sarcasm words from the targeted data. In this work, the word from English Language and Chinese language data sets have been evaluated, and the results have been extracted (Liu et al., 2014).

Bala Durga Dharmavarapu et al. have developed a research concept to detect the Sarcasm from the twitter data by implementing sentiment analysis. In this paper, the sarcasm detection is done by using sentiment analysis, naïve Bayes Classification, and AdaBoost algorithms. In this process, the tweets have been categorized into sarcastic and non-sarcastic words. This process has successfully demonstrated the sarcasm words from targeted twitter database information (Dharmavarapu & Bayana, 2019).

The sarcasm detection method is proposed to improve review analysis by Shota Suzuki et al. This work has focused on a sequential approach to detect Sarcasm. This approach is starting by applying dependency parsing to the data. Then the method has classified expressions in the sentences into the proposed word based on the structure of parts of speech. Then the analysis is done with sentiment phrases to determine it as Sarcasm. This has given successful results in finding the sarcasm words from the given data (Suzuki et al., 2017).

The IEEE research work has been predominant in sarcasm detection presented by Le Hoang Son et al. In this work, sarcasm detection is incorporated with the help of soft attention based bidirectional log short term memory model within the convolution Network. This paper is predominantly concentrated to detect Sarcasm with the help of a deep learning model that has been developed on the basis of a hybrid of soft attention-based bidirectional long short-term memory. This model is incorporated a convolution neural network, then applying global vectors for word illustration for building semantic word embedding. This work has exemplified the concept with the help of maps for punctuation-based auxiliary features (Son & Nayyar, 2019).

Yitao Cai et al. has developed a prominent manner to detect Sarcasm from Twitter with the help of the Hierarchical Fusion Model. This paper has focused on Multi-model sarcasm detection by treating text image features and image attributes as three modalities and examined with multi-model hierarchical fusion (Cai et al., 2019).

2.1. Supervised classification Machine Learning Techniques

Supervised learning techniques can be used to sufficient labeled training text corpora are accessible. Sarcastic detection query can be made as given training text document set $TD = \{TD_1, TD_2, TD_3, \dots, TD_n\}$, with each text document, selected one of the binary class labels, Sarcasm' or 'non- Sarcasm. A classification model x relates a feature set of documents to the class label. After that, it was given the new text document 'td'; a model x is used in order to predict the class label for new text documents. There are some of the main methods explained as follows.

2.1.1. Multinomial Naïve Bayes (MNB)

It is a Machine Learning algorithm associated with the supervised classification. A multinomial distribution is helpful to our model feature extraction vectors, where each value expresses the number of the appearance of a term or its relative rate. When feature vector has n elements where each of them, we can assume k , other values with likelihood p_k .

$$x_1 = x_1 \cap x_2 = x_2 \cap \dots \cap x_k = x_k \quad (1)$$

The conditional probabilities $P(x_i|y)$ here are counted with a frequency computed (corresponds to implementing a maximum probability approach) (Alami & Elbeqqali, 2015)

2.1.2. Logistic Regression (LR)

It is a Machine Learning algorithm within the supervised- classification. Logistic Regression is a predictive analysis based on the thought of probability. The hypothesis of logistic regression head for it is to limit the detriment function between 0 and 1. Below is an example logistic regression equation: As shown in the figure below (Ali Zia & Khan, 2017)

$$y = e^{(b_0 + b_1 * x)} / (1 + e^{(b_0 + b_1 * x)}) \quad (2)$$

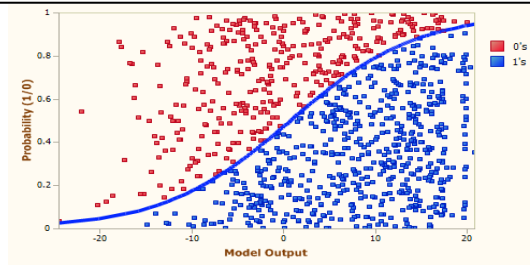


Fig 3: Logistic Regression (Ali Zia & Khan, 2017)

2.1.3. Support vector machines (SVM)

It is a Machine Learning algorithm with the category of supervised- classification – Regression. Support vector machines (SVM) are supervised learning models with learning algorithms that analyze data used for regression and classification analysis As shown in the figure below (A. Joshi et al., 2017) .

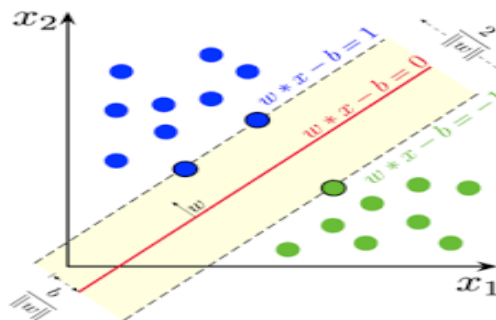


Fig 3: Support vector machines (SVM) (A. Joshi et al., 2017)

2.1.4. Decision Tree (DT)

It is a Machine Learning algorithm with the category of supervised- classification – Regression. The decision tree may be defined as the predictive modeling methods used in data mining, statistics, and machine learning. Decision trees are envisioned by an algorithmic method that knows ways to divide a data set based on various conditions. It is one of the common algorithms, widely and efficient methods used for supervised learning as it is shown in the decision tree in the figure below (Kharde, 2016)

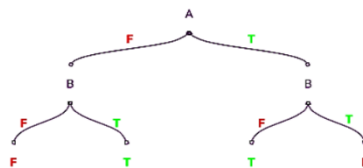


Fig 3: Decision tree (Kharde, 2016)

3. Proposed Methodology

The proposed research methodology is taken from the previous research papers. The main goal of our research is to classify the primary data and use the kaggle obtaining the secondary data for the classification of Sarcasm from the social media twitter platforms (Bhanap & Kawthekar, n.d.).

3.1. Dataset

Data set are downloaded from Kaggle contains two columns, one tweet, and the other one label. The first tweet contains a tweet that brings from API, and the next column label contains binary 0 or 1, extracted manually in CSV file contains 1984rows of tweets. It consists of 1024 Sarcasm tweets and 962 as non- Sarcasm tweets were taken to train a model.

3.2. Text Preprocessing

Pre-processing and feature extraction are critical steps for text classification applications. In this division, we introduce procedures for cleaning the text data set, thus noise removal implicit, and allowing for useful information. Pre-processing is enriched with text cleaning with mentions identification, URL’s classification, and symbols of points, hashtags, commas, and other kinds of symbols identification, Identification of white spaces, Identification of lower case words, fixing words, Tokenization, stop words, and Text vectorization. In the preprocessing stage,

the noise would be cleaning. Basically, sarcasm detection is done with three essential elements. These are Lexical, Hyperbole, and Pragmatic. Lexical classification can be done with unigram, Bigram, and N-gram. The Hyperbole classification can be incorporated with the interjection, Punctuation Marks, Quotes, and intensifier (Hayat et al., 2019).

3.3. Feature Extraction and Sarcasm Detection

After finished from the preprocessing stage, the data must be prepared and made ready for the next step. There are five phases, (1) Words tokenization, (2) Parts-Of-Speech (POS), (3) lemmatization, (4) Feature extraction, and (5) New Representation. The first stage is the word Tokenization. Tokenization is implemented on tweets to reveal them to words slipping into perfect expressive modules from a decision as follows :{ “After” “four,” “sleeping,” “hours,” “for”}. (Chen & Yan, 2018) (Analysis, 2017) Next, the tweets are tokenized, the POS tag is achieved. The next word in a tweet and their POS plays a role in the analysis. By that into mind, this proposed model preprocess steps started with clean text data and remove specials characters. After the clean is finished, Lemmatization and Stemming are completed. Lemmatization is an NLP process that replaces the suffix of a word with a different one (Li & Li, 2013), (Rui et al., 2013), (Rathore et al., 2017). Next, Identification specific weights are assigned based on the purpose. That is complete so that if the initial part of the sentence is positive Sarcasm and the second one is negative non-sarcasm, is expected to be detected. Based on these, a weight among 0 and 1 is selected as the feature extraction weight. Then, the features are recognized including weighted; a representation is given with utilizing the features extractions and their respective weights (Upadhyay & Fujii, 2016), (S. Joshi, 2018). That presentation is fed as input to classify the model for the training set and testing set. In the process of sarcasm detection, the data set provided by twitter, rich with 1984 rows, has been taken into consideration. The data has to be sent for the preprocessing stage to eliminate the general word and Identification of sarcasm words with the help of symbols and hashtags. The machine learning concepts are needed to be implemented in the processing stage (Sukanya & Biruntha, 2012), (Salloum et al., 2018b).

3.4. Training, Test Data Set

We use a dataset to train our model by 70%, the classifier that contains 1984 tweets, out of which 1024 are sarcasm tweets, and the residual is non- sarcasm tweets. Test this set includes 30% tweets after those results are compared with a label that has been created manually in order to predict the performances of our proposed classification (Kharde, 2016) (Salloum et al., 2018).

4. Proposed model and Algorithm

Sarcasm in pure text data is being detected by following practically step by step approaches as arranged in figure 4 below.

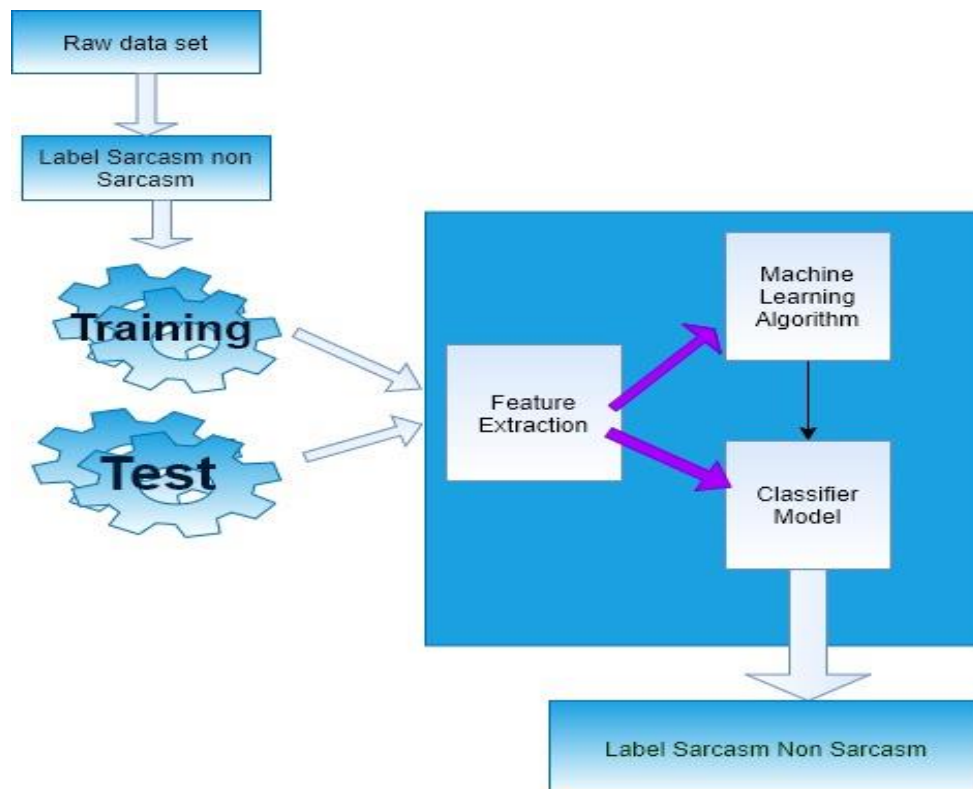


Fig. 4. Algorithmic building of Sarcasm Non-sarcasm detection.

In figure (4) shown above, the initial step is to import a library related to machine learning and Natural language processing (NLP) related tasks. The next step is to load CSV file as data set, and preprocess data set with some

steps such as the Filtration of stop words, slang, and abbreviation, noise removal, HTML tags, non-letters, Stemming each word. The next step is to train the model data set, bag of word, Feature Extraction the data (TF-IDF) vectorizer, then split data into train and test, after that chose classification model such as Multinomial NB, and fit model, and confusion matrix, classification report last setups predict accuracy

Algorithm 1. Twitter Extract Sentiment

Begin

Collected Datasets from microblogging websites twitter
 Filter Tweets English Language
 Remove all the secluded Twitter symbols (#topic, @user name, retweet (RT))
 Remove noise removal, Emoticons, numbers, and punctuations
 Remove URL (“HTTP:// HTML”)
 Remove slang, and abbreviation, non-letters
 Removal of Stop Words
 Remove stemming each word
 Tokenization
 Unigrams
 End Procedure
 Procedure Feature Extraction (text tweet):
 Extract Features Appropriately to algorithms machine learning
 End Procedure
 Procedures Sentiment analysis Classification (Feature):
 Choose a Classify tweet by applying machine learning techniques: LR, Multinomial NB, (SVM) & DT.
 Fitting the data (training the data)
 End Procedure
 Procedures evaluate the model using (confusion matrix)
 Predications, detecting results
 End Procedure
 End Until

End

5. Experimentation and Discussion Result

The experimental section consists of a windows system, and the anaconda consists of Jupyter Notebook. The code can be run by a python virtual environment. Jupyter Notebook can be installed with the help of pip or conda library executing the code to build a Machine learning model. The code can be divided into four stages. The first stage is to import the library; the second one is to load and preprocess the data set, the third stage is to train the model data set & bag of word classifier train, and the last stage is to predict evaluation. These four mentioned steps above can import all libraries related to machine learning and Natural language processing (NLP) related tasks.

Another step is to load CSV files as data set and preprocess data set with some steps such as the Filtration of stop words, slang, and abbreviation, noise removal, HTML tags, non-letters, Stemming each word. Then, to train model data set bag of word, Feature Extraction, the data (TF-IDF) vectorizer then split data into train and test. After that, a classification model is to be chosen, such as Multinomial NB, Logistic Regression, Decision Tree Classifier, and SVM. Another step is to fit the model, confusion matrix, and classification report; at last, there is a prediction of Accuracy for results that are presented below in Table 1.

algorithms	accuracy	Avg. precision	Avg. Recall	Avg. f1-score
Logistic Regression	0.97	0.97	0.97	0.97
Decision Tree Classifier	0.93	0.93	0.93	0.93
SVM	0.42	0.18	0.43	0.26
Multinomial NB	0.94	0.94	0.94	0.94

Table 1. Results Achieved by Using Different Classifiers models

Algorithms	accuracy
Logistic Regression	0.97
Decision Tree Classifier	0.93
SVM	0.42
MultinomialNB	0.94

Table 2. Accuracy results Using Different Classifiers models

Table 2 above shows the results of classifier models. Here, the highest accuracy of Logistic Regression, Decision Tree Classifier, and Multinomial NB is shown where the Logistic Regression has the highest frequency with 0.97 percent—followed by MultinomialNB with 0.94 percent and 0.93 in Decision Tree Classifier. The lowest highest frequency was shown in SVM, with only 0.42 percent. In Support vector machines (SVM), accuracy is around 0.42. Thus, we can say that SVM does not fit with these data set, as indicated below in fig 2.

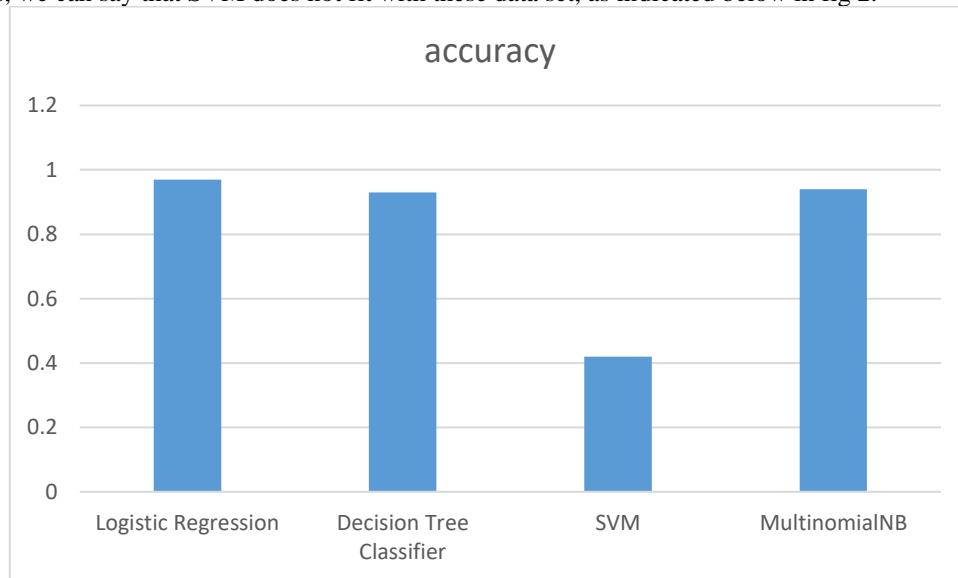


Fig. 5. Accuracy Classifiers models

As showed in the figure 5 above that Logistic Regression shows better results and higher improve accuracy with data training from other model classification, then demonstrated in Support vector machines (SVM) low accuracy. Next, we have compared all the results obtained from our model in clouded Avg. Precision, Avg. Recall, Avg. f1-score and support, as shown below in fig 6.

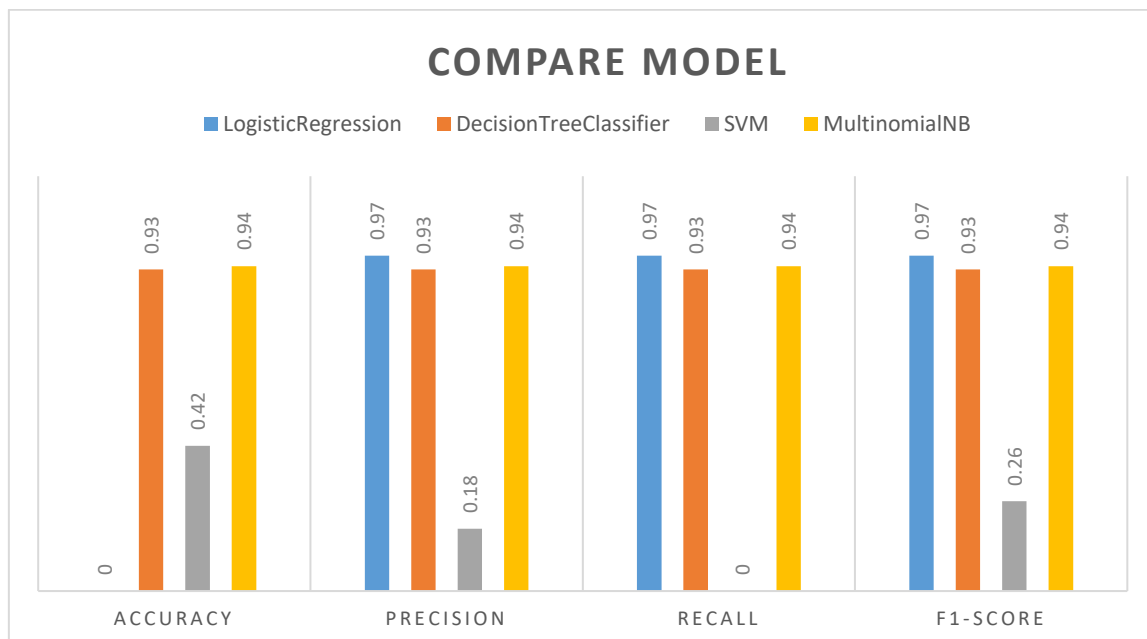


Fig. 6. Comparison Classifiers models

This figure shows all the results of the classification model, where the comparative results of all the four classifiers used are presented. Also, the accuracy of the results takes a Sarcasm detection of data reflected for training.

5.1. Confusion Matrix

Confusion Matrix, It is applied by classification issues where in the output performance measures to evaluate our model algorithms. Hence, in fact, such rules may just exist to some ways. The main difficulty when evaluate text data classification method is the need for standard text data. Accuracy measures the division of right prediction. Within our case, it is the quantity of terms the main kind appeared in the head three predicted classes distributed

through that whole number of classification assignments (Saito, 2020)(Build Your First Text Classifier in Python with Logistic Regression | Kavita Ganesan, n.d.). Accuracy estimates the division of correct predictions. In our state, it is the number of times the Main category appeared in the head 3 predicted categories distributed by the whole number of categorization assignments.

Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Recall	$\frac{TP}{TP+FN}$
Precision	$\frac{TP}{TP+FP}$
F1-Measure	$\frac{2(P*R)}{(P+R)}$

Table 3. Confusion Matrix

Comparative Accuracy and AVG of F1-score

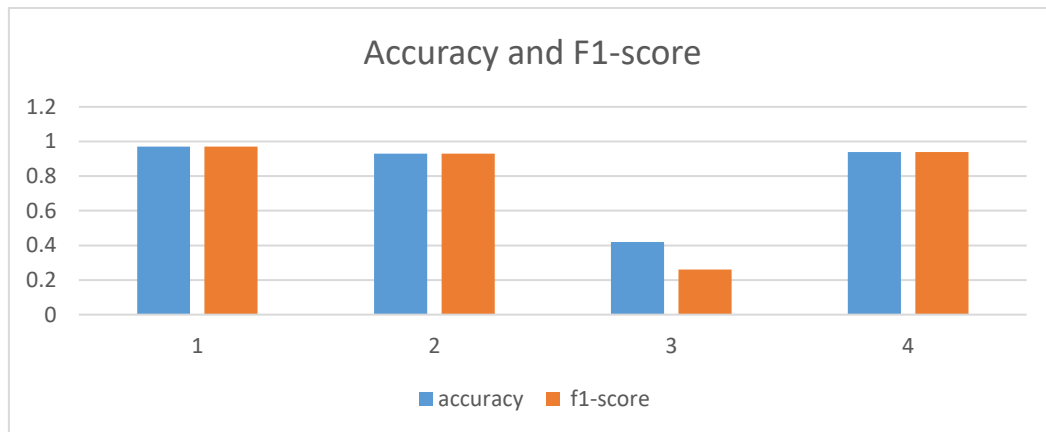


Fig. 7. Compare between Accuracy and Avg. F1-scoreClassifiers models

This figure shows the results of the imbalance between Accuracy and Avg. F1-score in Logistic Regression, Decision Tree Classifier, but in SVM shown differently. Accuracy and Avg. F1-score differs as indicated below in fig 7.

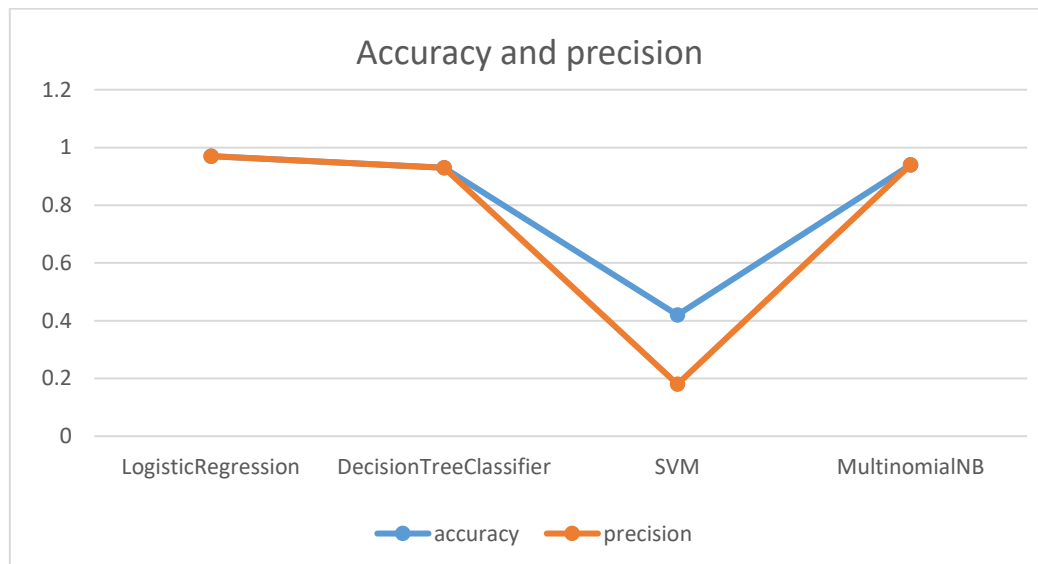


Fig. 8. Compare Accuracy and Avg. precision Classifiers models

These results imbalance between accuracy and precision in Logistic Regression, Decision Tree Classifier, but in SVM has shown different results. Though SVM seems so poor model with the given data, all procedures are parallel. Here it can be seen Logistic Regression shows better accuracy in classifying models than SVM. This is often called feature extraction, which affords better results. The same technique, which is inattentive in SVM, leads to low accuracy. The Decision Tree Classifier and Multinomial NB are achieved by carrying weights in training and modifying them through one only line repetition. The SVM fails in this classification. The dataset though, when expanded, can be used Count-Vectorizer to obtain better results.

6. Conclusion and Future Scope of the Study

Sarcasm is a complex form of irony that was widely observed on social media such as the Twitter platform. Detecting offensive tweets is an essential matter in textual classification also thus has several implications. This current paper successfully demonstrated the sarcasm detection classification by including enhanced preprocessing, feature extraction, and text mining techniques, which is valuable in Machine learning for investigating the social media opinion on a specific organization. The data supplied by Twitter is analyzed and extracted the sarcasm words from the data set to get scrams or non-scrams sentences. In our model applied some classification algorithms such as the Decision Tree Classifier, Multinomial Naïve Bayes, Logistic Regression, and Support vector machines (SVM) Algorithm. We achieved good accuracy results compare with label data, and our model showed improved accuracy with Logistic Regression by 0.97%, Decision Tree Classifier 0.93%, and Multinomial Naïve Bayes 0.94%. It frailer with Support vector machines (SVM) shown 0.42 % only. This paper has examined several approaches that can extract the Sarcasm from the targeted social media. Among all processes, machine learning, especially deep machine learning algorithms enriched with Multinomial Naïve Bayes, is implemented to extract the results. The proposed Algorithm is proved to be useful to obtain the sarcasm words from the social media documents obtained from a twitter data set. This study focused on the detection of Sarcasm by plain text with the highest accuracy. In the future, it is recommended to pay attention and do more researches related to a comprehensive hybrid classification method to detect Sarcasm in deep machine learning in massive data. We suggest applying another classification model with CountVectorizer.

References

- [1] Ahmad, A., Siddiqui, T., & Khan, N. A. (2018). A Detailed Phasewise Study on Software Metrics : A Systematic Literature Review. 3(3), 1696–1705.
- [2] Alami, S., & Elbeqqali, O. (2015). Cybercrime profiling: Text mining techniques to detect and predict criminal activities in microblog posts. 2015 10th International Conference on Intelligent Systems: Theories and Applications, SITA 2015. <https://doi.org/10.1109/SITA.2015.7358435>
- [3] Ali Zia, U., & Khan, N. (2017). Predicting Diabetes in Medical Datasets Using Machine Learning Techniques. In International Journal of Scientific & Engineering Research (Vol. 8, Issue 5). <http://www.ijser.org>
- [4] Analysis, S. (2017). Sarcasm detection using combinational Logic and Naïve Bayes Algorithm. 5, 546–551.
- [5] Bhanap, S., & Kawthekar, S. (n.d.). Twitter Sentiment Polarity Classification & Feature Extraction. 1–3.
- [6] Build Your First Text Classifier in Python with Logistic Regression | Kavita Ganesan. (n.d.). Retrieved October 18, 2020, from <https://kavita-ganesan.com/news-classifier-with-logistic-regression-in-python/#.X4xyddD7Q7e>
- [7] Cai, Y., Cai, H., & Wan, X. (2019). Multi-Modal Sarcasm Detection in Twitter with Hierarchical Fusion Model. 2506–2515.
- [8] Chen, J., & Yan, S. (2018). Verbal aggression detection on Twitter comments : convolutional neural network for short-text sentiment analysis. Neural Computing and Applications, 0. <https://doi.org/10.1007/s00521-018-3442-0>
- [9] Dharmavarapu, B. D., & Bayana, J. (2019). Sarcasm Detection in Twitter using Sentiment Analysis. 1, 642–644.
- [10] Hayat, M. K., Daud, A., Alshdadi, A. A., Banjar, A., Abbasi, R. A., Bao, Y., & Dawood, H. (2019). Towards Deep Learning Prospects : Insights for Social Media Analytics. IEEE Access, 7(May), 36958–36979. <https://doi.org/10.1109/ACCESS.2019.2905101>
- [11] Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). Automatic Sarcasm Detection : A Survey. 0(0).
- [12] Joshi, S. (2018). Twitter Sentiment Analysis System. 180(47), 35–39.
- [13] Kharde, V. A. (2016). Sentiment Analysis of Twitter Data : A Survey of Techniques. 139(11), 5–15.
- [14] Li, Y., & Li, T. (2013). Deriving market intelligence from microblogs. Decision Support Systems, 55(1), 206–217. <https://doi.org/10.1016/j.dss.2013.01.023>
- [15] Liu, P., Chen, W., Ou, G., Wang, T., Yang, D., & Lei, K. (2014). Sarcasm Detection in Social Media Based on Imbalanced Classification. 459–460.
- [16] Rajeswari, K., & Shanthibala, P. (2018). SARCASM DETECTION USING MACHINE LEARNING TECHNIQUES. 9(L), 26368–26372. <https://doi.org/10.24327/IJRSR>
- [17] Rathore, A., Chauhan, S., & Gujral, S. (2017). Detecting and Predicting Diabetes Using Supervised Learning: An Approach towards Better Healthcare for Women. In International Journal of Advanced Research in Computer Science (Vol. 8, Issue 5). <http://www.ijarcs.info/index.php/Ijarcs/article/viewFile/3674/3504>
- [18] Rui, H., Liu, Y., & Whinston, A. (2013). Whose and what chatter matters? The effect of tweets on movie sales ☆. Decision Support Systems, 55(4), 863–870. <https://doi.org/10.1016/j.dss.2012.12.022>
- [19] Saito, T. (2020). Basic evaluation measures from the confusion matrix. Classifier Evaluation with Imbalanced Datasets. <https://classeval.wordpress.com/introduction/basic-evaluation-measures/>
- [20] Salloum, S. A., Al-Emran, M., Monem, A. A., & Shaalan, K. (2018a). Intelligent Natural Language Processing: Trends and Applications (K. Shaalan, A. E. Hassanien, & F. Tolba (Eds.); Vol. 740, Issue January). Springer International Publishing. <https://doi.org/10.1007/978-3-319-67056-0>
- [21] Salloum, S. A., Al-Emran, M., Monem, A. A., & Shaalan, K. (2018b). Using text mining techniques for extracting information from research articles. In Studies in Computational Intelligence (Vol. 740, pp. 373–397). Springer, Cham. https://doi.org/10.1007/978-3-319-67056-0_18

-
- [22] Siddiqui, T., & Alam, M. A. (2009). Discovery of Fuzzy Censored Production Rules from Large Set of Discovered Fuzzy if then Rules. 3(1), 158–161.
- [23] Siddiqui, T., Amer, A. Y. A., & Khan, N. A. (2019). Criminal Activity Detection in Social Network by Text Mining: Comprehensive Analysis. 2019 4th International Conference on Information Systems and Computer Networks, ISCON 2019, 224–229. <https://doi.org/10.1109/ISCON47742.2019.9036157>
- [24] Son, L. E. H., & Nayyar, A. (2019). Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network. *IEEE Access*, 7, 23319–23328. <https://doi.org/10.1109/ACCESS.2019.2899260>
- [25] Sukanya, M., & Biruntha, S. (2012). Techniques on text mining. Proceedings of 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies, ICACCCT 2012, 978, 269–271. <https://doi.org/10.1109/ICACCCT.2012.6320784>
- [26] Suzuki, S., Orihara, R., Sei, Y., Tahara, Y., & Ohsuga, A. (2017). Sarcasm Detection Method to Improve Review Analysis. *Icaart*, 519–526. <https://doi.org/10.5220/0006192805190526>
- [27] Upadhyay, R., & Fujii, A. (2016). Semantic Knowledge Extraction from Research Documents. 8, 439–445. <https://doi.org/10.15439/2016F221>
- [28] Yahya, A., Amer, A., & Siddiqui, T. (2020). Detection of Covid-19 Fake News text data using Random Forest and Decision tree Classifiers Abstract : 18(12), 88–100.