A CROSS DOMAIN APPROACH TO ANALYSE SENTIMENTS FROM OPINIONS ON SOCIAL MEDIA

[1]Sobia Shafi, [2]Vandana Pushe,[3]Manmeen

[1]Reseach Scholar,[2]Assistant Professor,[3]Assistant Professor,
[1] Department of Computer Science,
[1]Swami Vivekanand institute of Engineering and Technology, Banur, India.

Abstract : Data is generated from versatile sources and in large quantities. The rate of data generation is increasing with time. This data is related to a large range of fields. Also this data contains hidden information which can be extracted. This information can prove beneficial to multiple organizations and institutions. Data from Social Networks mostly contains opinions of general public. Twitter is one such application where people express their opinions regarding every issue, development, product etc. In this paper we have tried toTo store and organize unstructured datasets from twitter into HDFS of Hadoop using Flume and extract data from HDFS through Apache Hive. The design and implement hybrid classification algorithm comprising of Naïve bayes based on probability and Decision Tree based on decision rules is done and along with it analysis and comparison of performance of the proposed approach based on accuracy with the existing naïve bayes.

***IndexTerms* – HDFS,SVM,NLP,POS,NB,API.**

1. Introduction

Data is produced from multiple sources like automobiles, banks, sensors, day-to-day human activities, social media etc. But the volume of data has grown beyond the computing power of traditional approaches of processing. Data is generated in large quantities on social media. Previously, information used to spread into little circles. Nowadays, people use social media to express their opinions about everything. They post about things and share images. Social network have received an upward surge and data generated from them is attaining higher values day by day. According to a survey, in one minute lakhs of tweets are sent, thousands of images are shared on Facebook and lakhs of videos are watched on YouTube [3]. Twitter is among the most used social networks. Users are able to tweet using a limited number of words so that it is read by everybody. There are tweets regarding many things including business establishments, movies, political parties, educational institutions, scientific projects etc. These tweets reflect sentiment of the people regarding various topics, products, movies. Every person has his own opinion regarding a topic or product. This sentiment gets reflected when the person tweets about that topic. One of the approaches is to do Sentiment Analysis. We get data from a social network and store it into Hadoop. Then we implement a classification algorithm using Map Reduce framework which will classify the data we have gathered based on what sentiments are hidden in it.

1.1 Data Classification

Text can be classified according to what information is present in the text. Text classification can be done in two phases [6]. In first phase, a model is built by training it with data we know what class it belongs to. First, we need to generate the dataset along with labels, then this data needs to be processed. Then model is trained after the data is vectorised. In the second phase, we test the model by making it to classify previously unseen data.

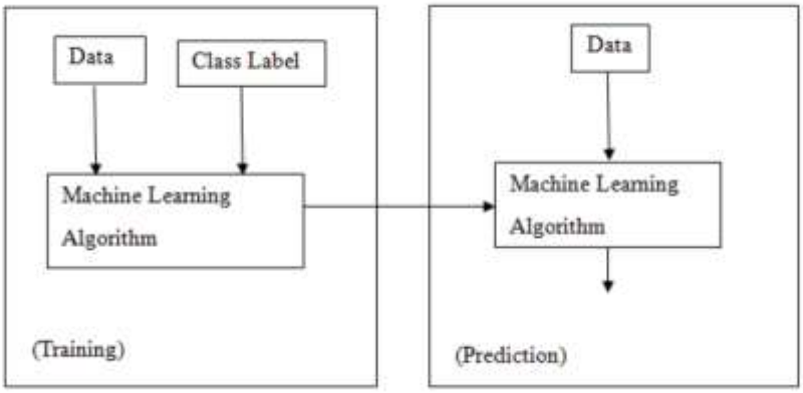


Fig :1 Representation of data classification

Algorithm of Data Classification

Naive Bayes

In naïve bayes, we will have the number of groups. Also we need to choose a number of data points irrespective of their class. We need to determine prior probability of every group. This depends upon the data in the class and also the whole data. Prior probability defines the probability of a data item to belong to this group. Then we use the previously chosen number of data points and use it when a new data item comes. For each data item, we have to compute the likelihood. This determines the probability of the new data item to belong to either of the groups. It is also computed for new data item corresponding to each group. Then from the prior probability and the likelihood, we determine the Posterior Probability of data item corresponding to each class. The highest value indicates that the new data item belongs to that class.

SVM

SVM tries to determine a line in the plane of the data which will be able to isolate the data of different groups. There can be a linear line, or plane or even a hyperplane which will do this isolation. The separator is selected based on distance. The separator with largest normal distance from the data is chosen.

Decision Table

Decision table is a compact way to represent information in a concise manner and predict class labels for the same. It is particularly useful when there are large number of features considered. Many features are simultaneously considered and a label is predicted. It becomes easy when there are a lot of features and when we need to remove some combinations of features.

1.2 SENTIMENT ANALYSIS

Sentiment analysis means identifying the opinions of a person about anything. Sentiment analysis is of use in social media monitoring as we can gain an understanding of what the general public feels about something. Sentiment analysis can be used in various ways. There are two main approaches to sentiment analysis.

- Using Data mining Techniques.
- Using Natural language processing Techniques.

Using Data Mining Techniques: In this approach, first the unstructured text data is converted into a structured form having numerical data. Then, the traditional data mining classifiers can be trained and tested for revealing sentiments. Creation of the document-by-term frequency matrix is the first step in this approach. This is achieved by parsing each document into its individual words. After this, the documents are represented in the form of vectors. As the matrix created during the first stage of this process is sparse, a dimensional reduction process is undertaken to represent each document on a reduced dimensional space containing the features that are significant for the purpose of classification. This is done by using the techniques like Singular Value Decomposition (SVD). During this step, the dimension of each document is reduced to just 50-100 features.

Using Natural language processing Techniques: This technique aims at automatic detection of sentiments from the data. It makes use of Part-Of-Speech Tagging. It tries to develop a lexicon and relies on analysing patterns from data using that lexicon and POS Tags. It is really a tough task to teach a machine to understand and think like a human. The rule-based NLP methods use the syntactic patterns in the text along with certain entities to understand the meaning of the given text. A combination of parts of speech, linguistic dictionaries, and noun phrases with a range of operators is used for the purpose of extracting the meaning.

1.3 BIG DATA HADOOP CLUSTER

Hadoop is a part of the apache project of the Apache Software Foundation. It is a framework based on java which is capable of processing very huge datasets. Hadoop runs in a distributed environment and it can process unstructured, heterogeneous data coming at high volume. A hadoop cluster is composed of racks. The number of nodes in one rack is not fixed and varies from 20 to 40. These racks are connected by a single network connecting device. There are master nodes and slave nodes in the hadoop cluster. Mater node is associated with managing the Hadoop Distributed File System. It also has the knowledge about which slave nodes contain the data of a particular file. The slave node is associated with carrying out the task. When a job is to be performed, the slave nodes are tasked with retrieval and processing of the data. Upon starting of a job, the slave node will find the location of data from the master node and retrieve the data and then process it.

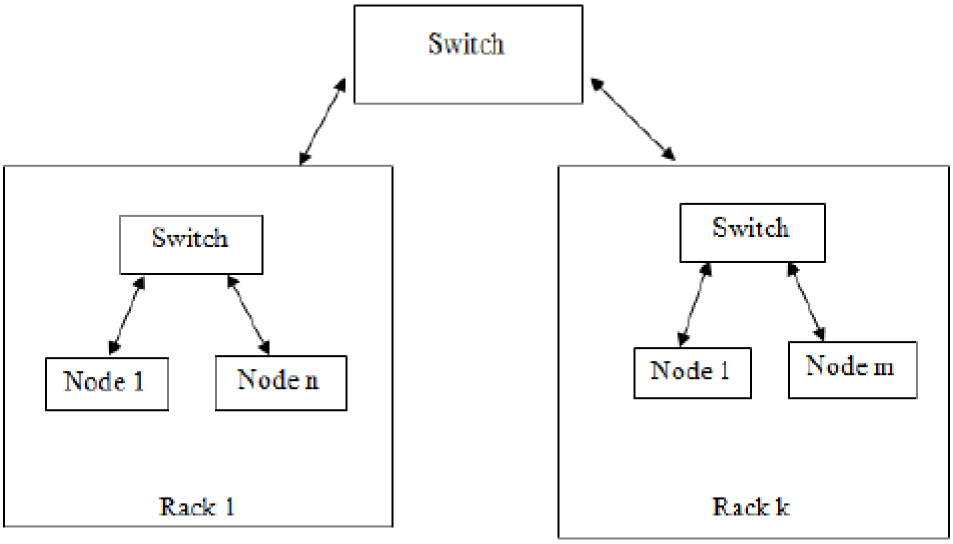


Fig 2: Overview of Hadoop Cluster

1.4 HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

HDFS (Hadoop Distributed File System) is the filesystem for the hadoop framework. It is distributed and scalable. It contains two types of nodes- Name Node and Data Node. Name node is responsible for management of the filesystem and data node is responsible for storage and retrieval of data. Also, HDFS stores the data in terms of blocks that have a fixed size and stores them all over the cluster. Since Hadoop utilises commodity hardware, there can be failures. So, the data is stored on multiple nodes redundantly to ensure high availability. By default, each data item will be stored thrice [8]. Hadoop uses rack awareness to store the replicas. Rack Awareness means that the closest data node is chosen by the name node. A record of rack IDs of datanodes is kept by namenode and that helps in determining the nearest datanode. One replica is stored on the local rack. The second replica is stored on another datanode. The third replica is stored in a different rack.

1.5 MAP REDUCE

MapReduce is a software framework which is associated with processing of data over multiple nodes belonging to a cluster. There are two types of nodes in MapReduce- JobTracker and TaskTracker. They also run on master-slave model. JobTracker is the master node and TaskTracker is the slave node. There are two functions in MapReduce; Map function and Reduce function. The Map divides the task in between the worker nodes. The output of Map serves as the input to the reduce function. Every map and reduce functions are independent of each other. The processing occurs on different nodes, the function is executed on the node where the data required by that part of job is located. When a job needs to be executed, it is taken by the JobTracker. The JobTracker then assigns tasks of the job to the TaskTrackers. The concept of data locality is used.

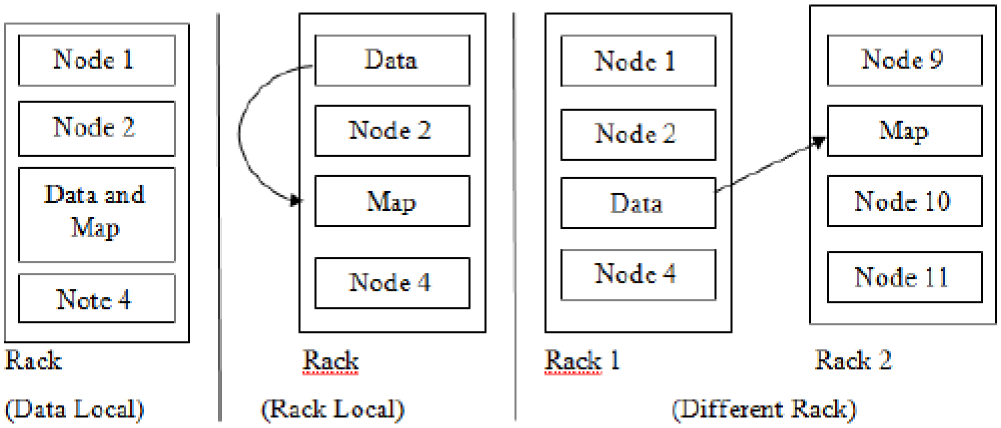


Fig 3 Data locality scenario in Map Reduce

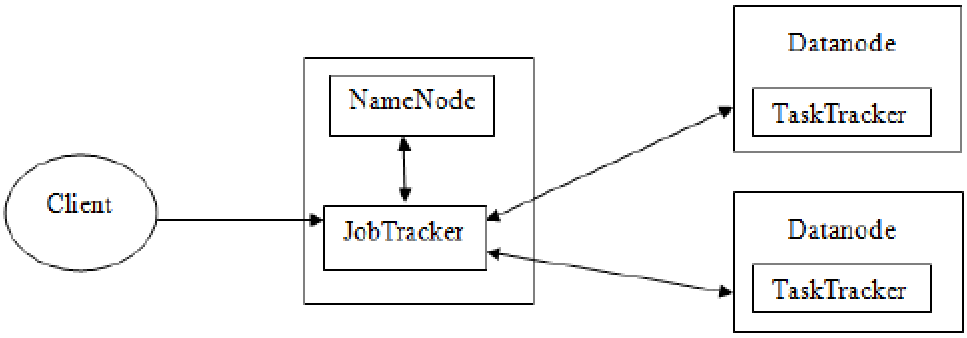


Fig 4 Components of Hadoop Cluster

2. RELATED WORK

Meylan Wongkar et.al. used Naïve Bayes classifier on twitter data to perform sentiment analysis[1]. They studied opinion of public towards presidential candidates of Indonesia. They collected data using Python Libraries and applied various algorithms on it for classification. They also carried a comparison study and found out that Naïve Bayes performs better than SVM and KNN when used for sentiment analysis on twitter data.

Kavya Supalla et.al. proposed a sentiment analysis methodology which utilises Naïve Bayes Algorithm[2]. They collected data from Twitter and performed binary classification –Positive and Negative Classes- using Naïve Bayes classifier to distinguish between positive and negative tweets to measure consumer opinion regarding a particular subject. They built the model using NLTK on tweet dataset.

Rachmawan Adi Laksono et.al. performed sentiment analysis of customer reviews by collecting data from Trip Advisor[3]. They used Naïve Bayes implemented in WEKA for their study. They used WebHarvey Tools to collect data from Trip Advisor. Their study found that Naïve Bayes method performs better than TextBlob sentiment analysis. Their study also had only two classes-positive and negative.

Fatima Zohra ENNAJI et al. propose a multi-agent framework for reading extracted evaluations from social media [4]. In their framework, multiple agents are present for multiple jobs like data extraction, refinement and analysis. A structure based on MapReduce performs data processing and mines the data.

Cui et al. [5], proposed an algorithm which associates a sentiment with the data and tries to associate a strength value to the data also. The algorithm is lexicon-based and makes use of various lists of words. However, the performance of this algorithm was not well for positive sentiment when it was tested with data collected from MySpace.

Abhinandan P Shirahatti et.al. proposed an approach of using twitter data for sentiment analysis using Hadoop [6]. Flume is used to get data from twitter after creating a twitter 20 application. This data gets stored in HDFS. Then Hive is used to analyse the various sentiments and inherent patterns of this data.

Soo-Min Kum et al. [7], proposed an algorithm for sentiment analysis. It is divided into four steps. First, the sentiment is recognised. It uses an algorithm based on WordNet dictionary to assign sentiment to a word. They proposed that synonyms of a word will have same sentiment as the word itself. This resulted in a corpus of data. For their presented approach, more the words in corpus, more will be the recall.

According to Hardi Rajnikant Thakor, various classification algorithms can be used for sentiment analysis [8]. Decision trees have fast fitting speed and fast prediction speed, but have low accuracy. Naïve bayes has high accuracy but has slow prediction speeds and consumes much time in training.

Ankur Goel et al. had proposed an implementation of Naive Bayes on twitter data [9]. It makes use of SentiWordNet. SentiWordNet in combination Naïve Bayes can improve accuracy of tweet classification, by awarding sentiment score based on positivity or negativity of words in a tweet. For implementation of this system python with NLTK and python-twitter APIs are used

Ravichandran et al. [10], introduced a comprehensive study on the challenge of detecting sentiment of words. It considered words to be either positive or negative. Three languages were targeted by them; English, French and Hindi. But little work has been done in developing lexicons for languages other than English.

3. METHODOLOGY

3.1 Collection of data: Data can be collected from Twitter using Twitter API. Twitter has millions of users who tweet billions of times in a week. One more imperative cause of using tweet data is that tweets are mostly in text, while on others, there are usually images, videos etc. Flume is used to get Twitter Data from and store it in HDFS. Later, Hive is used to get this data in textual form from HDFS. Then we manually edit the data to make it usable for our classifiers.

3.2 Pre-Processing and Filtering: The collected dataset from Twitter is in the form of raw data that needs to be filtered in order to do classification on the data. For the Filtration of the raw data, two filters from Weka are applied. They are StringToWord Filter and NominalToBinary Filter. It makes the data ready for use in classification algorithms.

3.3 Classification using Map Reduce Platform: A hybridised classification technique is then implemented. The algorithms used are Naïve Bayes and Decision Tree. Naïve Bayes makes use of conditional probability and bayes theorem and it assumes that features are independent. The overall class probability is estimated by combining the estimated probability. Decision Tree makes use of decision rules. We will use the Weka implementations of these algorithms in hadoop library by using hadoop JARs.

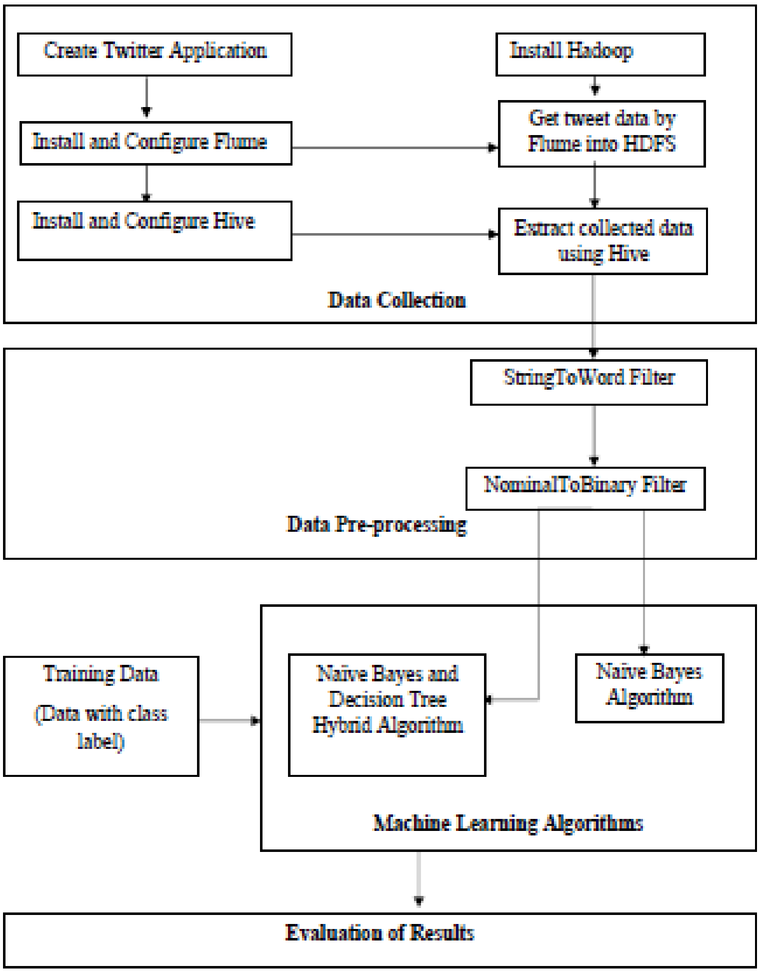
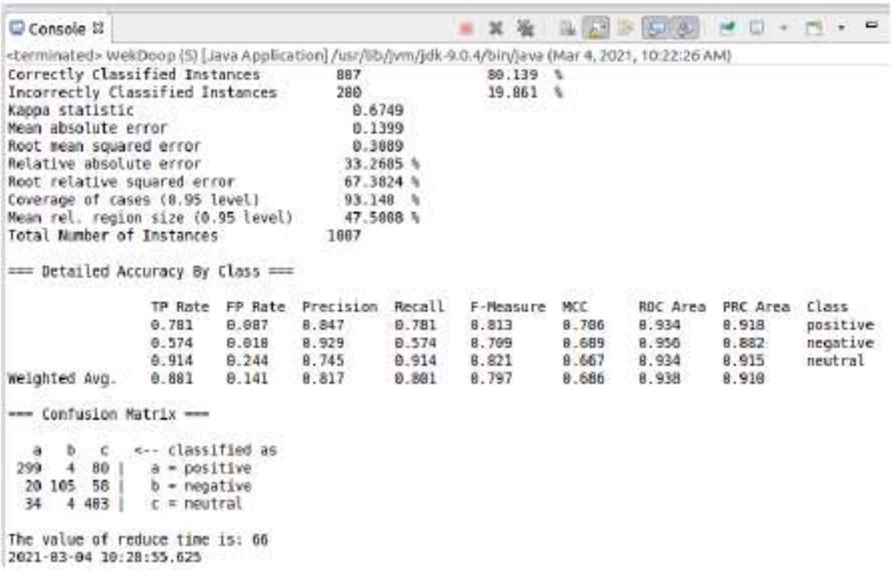

Create Twitter Application
Install Hadoop
Install and Configure Flume
Get tweet data by Flume into HDFS
Install and Configure Hive
Extract collected data using Hive
Data Collection
StringToWord Filter
NominalToBinary Filter
Data Pre-processing
Training Data (Data with class label)
Naïve Bayes and Decision Tree Hybrid Algorithm
Naïve Bayes Algorithm
Machine Learning Algorithms
Evaluation of Results

Fig 5 Flow Chart of Methodology

4. RESULTS

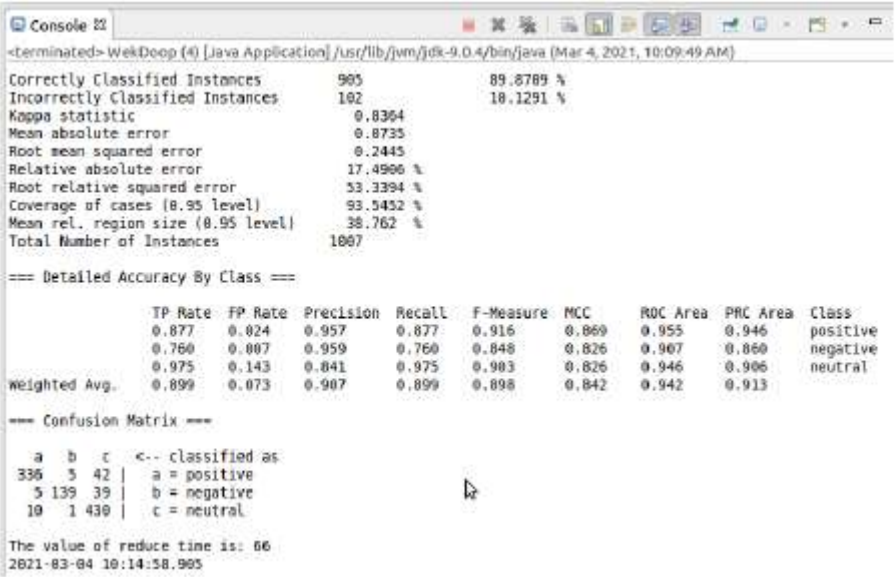
This section shows results obtained during experimentation. The experimentation was done twice on two data sets. One was small dataset with 1000+ instances, other one was a big dataset with 4000+ instances.

Results obtained with smaller dataset having 1007 instances are shown below.



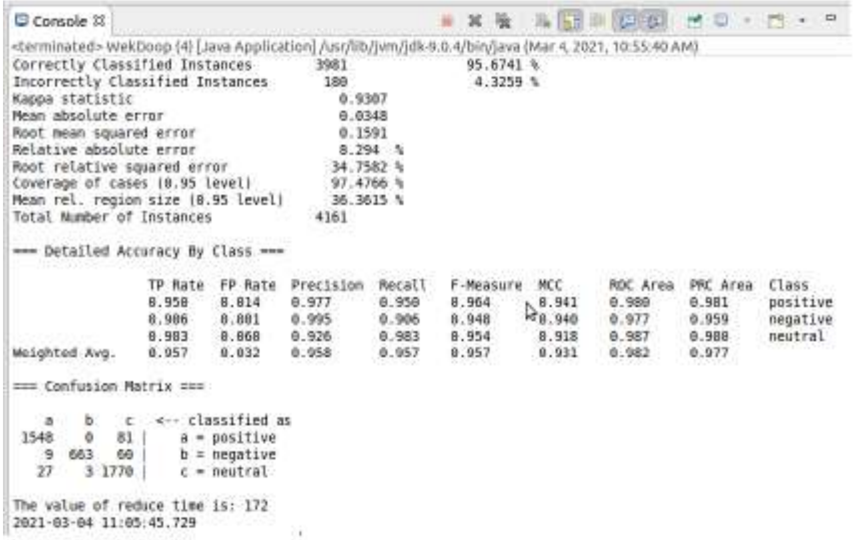
```
<terminated> WekDoop (5) [Java Application] /usr/lib/jvm/jdk-9.0.4/bin/java (Mar 4, 2021, 10:22:26 AM)
Correctly Classified Instances         807               80.139  %
Incorrectly Classified Instances       200               19.861  %
Kappa statistic                          0.6749
Mean absolute error                      0.1399
Root mean squared error                  0.3089
Relative absolute error                 33.2685 %
Root relative squared error             67.3824 %
Coverage of cases (0.95 level)          93.148  %
Mean rel. region size (0.95 level)      47.5008 %
Total Number of Instances             1007

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.781    0.087    0.847      0.781    0.813      0.706    0.934     0.918     positive
                 0.574    0.018    0.929      0.574    0.709      0.689    0.956     0.882     negative
                 0.914    0.244    0.745      0.914    0.821      0.667    0.934     0.915     neutral
Weighted Avg.    0.801    0.141    0.817      0.801    0.797      0.686    0.938     0.910

=== Confusion Matrix ===

   a   b   c   <-- classified as
 299   4  80 |   a = positive
  20 105  58 |   b = negative
  34   4 483 |   c = neutral

The value of reduce time is: 66
2021-03-04 10:28:55.625
```

Fig 6 Results of the existing technique on smaller dataset



```
<terminated> WekDoop (4) [Java Application] /usr/lib/jvm/jdk-9.0.4/bin/java (Mar 4, 2021, 10:09:49 AM)

Correctly Classified Instances         905               89.8709 %
Incorrectly Classified Instances       102               10.1291 %
Kappa statistic                          0.8364
Mean absolute error                      0.0735
Root mean squared error                  0.2445
Relative absolute error                 17.4906 %
Root relative squared error             53.3394 %
Coverage of cases (0.95 level)          93.5452 %
Mean rel. region size (0.95 level)      38.762  %
Total Number of Instances             1007

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.877    0.024    0.957      0.877    0.916      0.869    0.955     0.946     positive
                 0.760    0.007    0.959      0.760    0.848      0.826    0.907     0.860     negative
                 0.975    0.143    0.841      0.975    0.903      0.826    0.946     0.906     neutral
Weighted Avg.    0.899    0.073    0.907      0.899    0.898      0.842    0.942     0.913

=== Confusion Matrix ===

   a   b   c   <-- classified as
 336   5  42 |   a = positive
   5 139  39 |   b = negative
  10   1 430 |   c = neutral

The value of reduce time is: 66
2021-03-04 10:14:58.905
```

Fig 7 Results of the proposed technique on smaller dataset



```
<terminated> WekDoop (4) [Java Application] /usr/lib/jvm/jdk-9.0.4/bin/java (Mar 4, 2021, 10:55:40 AM)
Correctly Classified Instances        3981               95.6741 %
Incorrectly Classified Instances       180                4.3259 %
Kappa statistic                          0.9307
Mean absolute error                      0.0348
Root mean squared error                  0.1591
Relative absolute error                  8.294  %
Root relative squared error             34.7582 %
Coverage of cases (0.95 level)          97.4766 %
Mean rel. region size (0.95 level)      36.3615 %
Total Number of Instances             4161

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.958    0.014    0.977      0.958    0.964      0.941    0.980     0.981     positive
                 0.906    0.001    0.995      0.906    0.948      0.940    0.977     0.959     negative
                 0.983    0.068    0.926      0.983    0.954      0.918    0.987     0.988     neutral
Weighted Avg.    0.957    0.032    0.958      0.957    0.957      0.931    0.982     0.977

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1548    0   81 |    a = positive
    9  663   60 |    b = negative
   27    3 1770 |    c = neutral

The value of reduce time is: 172
2021-03-04 11:05:45.729
```

Fig 8 Results of the proposed technique on larger dataset

To conclude, Previously, the work of Sentiment Analysis was implemented with Naïve Bayes which assumes that features selected are independent of each other and its accuracy of classifying entities is low and also is more error-prone. In this proposed and implemented work, I tried to implement a hybrid-classifier based on Naïve Bayes and Decision Tree. The hybrid-classifier has improved accuracy and has fewer errors. Also the proposed system works better as the volume of data increases. So proposed system has more scalability than existing system.

REFERENCES

[1]. M. Wongkar and A. Angdresey, "Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter," 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 2019, pp. 1-5. IEEE 2019

[2]. Kavya Suppala and Narasinga Rao. "Sentiment Analysis using Naïve Bayes Classifier", International Journal of Innovative Technology and Exploring Engineering, June 2019, Volume 8, Issue 8, pp 264-269.

[3]. R. A. Laksono, K. R. Sungkono, R. Sarno and C. S. Wahyuni, "Sentiment Analysis of Restaurant Customer Reviews on TripAdvisor using Naïve Bayes," 2019 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 2019, pp. 49-54. IEEE 2019

[4]. Fatima Zohra ENNAJI, Abdelaziz EL FAZZIKI, Hasna EL ALAOUI EL ABDALLAOUI, Abderahmane SADIQ, Mohamed SADGAL, Djamal BENSLIMANE, "Multi-Agent Framework for Social CRM:Extracting and Analyzing Opinions", IEEE 2016.

[5]. Cui, Zhang, Liu, Ma, "Emotion tokens: Bridging the gap among multilingual twitter sentiment analysis", Information Retrieval Technology, 2011, pp. 238–249.

[6]. Abhinandan P Shirahatti, Neha Patil, Durgappa Kubasad and Arif Mujawar, "Sentiment Analysis on Twitter Data using Hadoop", International Journal of Emerging Technology in Computer Science and Electronics, April 2015, Volume 14, Issue 2, pp. 831–837.

[7]. Kim, Hovy, "Identifying and analysing judgment opinions", Proceedings of HLT/NAACL, 2006, pp. 200–207.

[8]. Hardi Rajnikant Thakor, "A Survey Paper on Classification Algorithms in Big Data", International Journal of Research Culture Society, Volume 1, Issue 3, May 2017, pp. 21 -27.

[9]. Ankur Goel, Jyoti Gautam, Sitesh Kumar, "Real Time Sentiment Analysis of Tweets Using Naive Bayes", IEEE 2nd International Conference on Next Generation Computing Technologies (NGCT-2016) Dehradun, India 14-16 October 2016, pp. 257-261.

[10].Rao, Ravichandran, "Semi-supervised polarity lexicon induction", Conference of the European Chapter of the Association for Computational Linguistics, 2009, 675–682.