International Journal on Emerging Technologies **11**(2): 25-30(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

A Sentiment Classification on Indian Government Schemes Using PySpark

E. Sujatha[1] and R. Radha[2]
[1]Research Scholar, Research Department of Computer Science,
SDNBV College for Women, University of Madras, Chrompet 600044 (Chennai), India.
[2]Associate Professor, Research Department of Computer Science,
SDNBV College for Women, Chrompet 600044 (Chennai), India.

(Corresponding author: E. Sujatha)
(Received 14 November 2019, Revised 04 January 2020, Accepted 08 January 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT : With the rapid growth of social media, millions of users share their opinions everyday about different issues such as services, products, persons, politics, events etc. By analysing these social user's opinions the most valuable data can be obtained which is used for decision making in various domains. Sentiment Analysis in social media plays a vital role in monitoring of public opinion behind certain topics. Actually 92% of marketing professionals thought that social media has intense effect on their business. By doing an accurate sentiment analysis, winning might not be an impossible thing. Success does not depend only in the number of likes, comments or followers. It also depends in how much positive discussions have been held among the users of social media. The objective of this paper is to build a model that performs multi-class classification of Big data with improved accuracy. The proposed model uses a hybrid of Lexicon-based approach and Machine Learning based approach. In this paper, we have performed the sentiment analysis on the three Central Government Schemes namely Digital India, Swachh Bharat and Make-in India. The Apache Spark's Machine Learning library, MLlib has been used in order to achieve better results in sentiment analysis for the Big Data. The AFINN dictionary was used to label the tweets and bigrams feature set was used. The Hashing TF-IDF method was used to extract the feature vectors from the raw feature set. These vectors were classified by Random Forest classifier to determine positive, negative and neutral sentiments of tweets. The result from this model was tested by using the various testing metrics like accuracy, precision, recall and f-score. The higher accuracy of 89.27% was obtained.

Keywords: Sentimental Analysis, Twitter Data, Make-in India, Digital India, Swachh Bharat, PySpark, Random Forest classifier.

I. INTRODUCTION

Sentiment analysis of twitter data can be done by using various techniques such as corpus-based, dictionary-based and machine learning algorithms [4]. The authors reviewed some papers of sentiment analysis of twitter data. They have limited this paper to that of machine learning models and show the comparison of these models. It was found that almost 85% - 90% of accuracy was reached by using these models [3].

From the experimental results it shows that machine learning algorithms are very efficient and performs better in terms of time and accuracy. These techniques can be useful in various areas such as purchasing product/service, improving product/service, recommendation systems, decision making etc [14]. Day by Day millions and millions of tweets were generated in the twitter. The Challenging task is that analysing the sentiments and its classification based on the polarity. There are lots of work has been done on sentiment analysis of twitter data and lots need to be done [12].

The authors analysed the mindset of famous persons in every situation when they used to tweet. They collected the tweets about Narendra Modi and Rahul Gandhi. Both the personalities were compared by using number of likes, re-tweets, average length of all the tweets and polarity of tweets. The polarity of tweets has been found by using TextBlob [13]. The tweets about the popularity of iPhone6 were analysed. The authors extracted tweets from the seven major cities of USA. Totally 940 tweets were collected out of which 410 were from female users and 530 were from male users. The Stanford Natural Language Processing tool was used to pre-process the data. They used POS-tagger and Senti-WordNet for Sentiment Analysis [15].

The authors proposed a predictive model for sentiment analysis. They used Linear Regression with the parameters such as Customer's Age, Gender as dependent variables and prediction of future sale as independent variable. They used 75% of tweets as training set and 25% of tweets as testing set from 14,000 tweets. They analysed opinions about election status between Hillary and Trump. They achieved 85.23% of accuracy [1]. In this paper, the authors proposed a system of analysing sentiments by using SVM classifier. The Weka tool was used to analyse the performance of SVM. The tweets about self-driving cars and tweets about apple products were the two types of data sets used in this study. The accuracy of 59.91% and 71.20% has been obtained for self-driving cars and apple data sets respectively [8]. In this paper, the authors proposed the system of analysing the twitter data by using k-nearest neighbour (KNN) and support vector machine (SVM). ROC (Receiver Operating Characteristics) graph technique was used to select the classifiers which depend on their performance. From the graph it shows that the KNN always performs better than SVM. The accuracy of 80.80% has been obtained by using KNN [7]. A hybrid of KNN & SVM classifiers was used to analyse the sentiments in tweets. From the result it shows that the machine learning approaches in hybrid manner improves the accuracy of 76.17% [2].

The authors extracted 2000 tweets and classified as positive, negative and average sentiment.
For their research work they have taken tweets about electronics products. They found the 6 best products for the year 2014–2015. They classified the tweets by using NB, SVM and maximum entropy.
In this paper, the authors proposed a system of multidimensional sentiment analysis. They classified the emotions in five different categories namely, happiness, fear, anger, surprise and sorrow. They classified the tweets using Convolution Neural Networks with n-gram feature sets. Python and NLTK libraries were used along with Senti-WordNet lexicon dictionary [5].
Sentiment was analysed in Turkish language. This dataset includes 9100 reviews of a product. The word2vec algorithm was used to generate word models. The authors used bag-of-words and n-gram models as feature sets and classified by using Random Forest algorithm. They achieved the accuracy of 84.23%.
The tweets were analysed in four steps such as extraction of tweets, pre-processing, feature extraction and classification [10]. The authors classified the tweets with SVM with n-gram feature set. And they applied the pattern based technique for the feature extraction in order to increase the accuracy [6]. Totally the authors were collected 500 tweets about “Digital India” by using twitter API. They analysed the tweets in three classes namely positive, negative and neutral. They used dictionary based approach for classifying the tweets. They found that out of 500 tweets 250 as positive, 150 as neutral and 100 as negative sentiments [11].
The adaptable sentiment analysis was found to extract the social zusers’ opinions [16]. The tweets were processed through three steps: constructing dictionaries, then classifying and balancing set of words before prediction. To validate the approach, the 2016 US election tweets were classified. The performance was evaluated only by means of accuracy getting 90.21%.
The fake account detection and sentiment analysis systems were trained and being tested by Naïve Bayes classifier from Apache Spark Framework [17]. The performance of the model was evaluated by means of accuracy, for offline and real-time modes are 86.77% and 80.93%, respectively [18]. The sentiment analysis was performed on Lithuanian Internet comment dataset. The authors compare the traditional machine learning approaches such as Naïve Bayes Multinomial and Support Vector Machine with deep learning approaches such as Long Short-Term Memory and Convolutional Neural Network. The NBM reaches the best accuracy of 73.5% and CNN reaches 70.6%.
Although there are several various methods to classify the sentiments in the existing literature, still it is an open problem to have optimizations. The proposed model was not a domain specific and it was designed to process the Big Data in the Apache Spark framework with the improved accuracy. The Table 1 compares the various machine learning approaches for Twitter sentiment analysis and outcomes of algorithms performance and [Fig. 1] shows the architecture diagram.

Table 1: The supervised machine learning approach for twitter sentiment analysis.

Papers	Total no. of tweets	Classifier	Accuracy	Year published
[1]	14000	Linear Regression	85.23%	2018
[8]	—	SVM	59.91%	2017
		SVM	71.20%	
[7]	—	KNN	80.80%	2017
[2]	—	KNN + SVM	76.17%	2017
[10]	9100	Random Forest	84.23%	2017
[17]	—	Naïve Bayes	86.77%	2019
			80.93%	
[18]	—	Naïve Bayes Multinomial	73.5%	2019
		CNN	70.6%	

II. ARCHITECTURE DIAGRAM

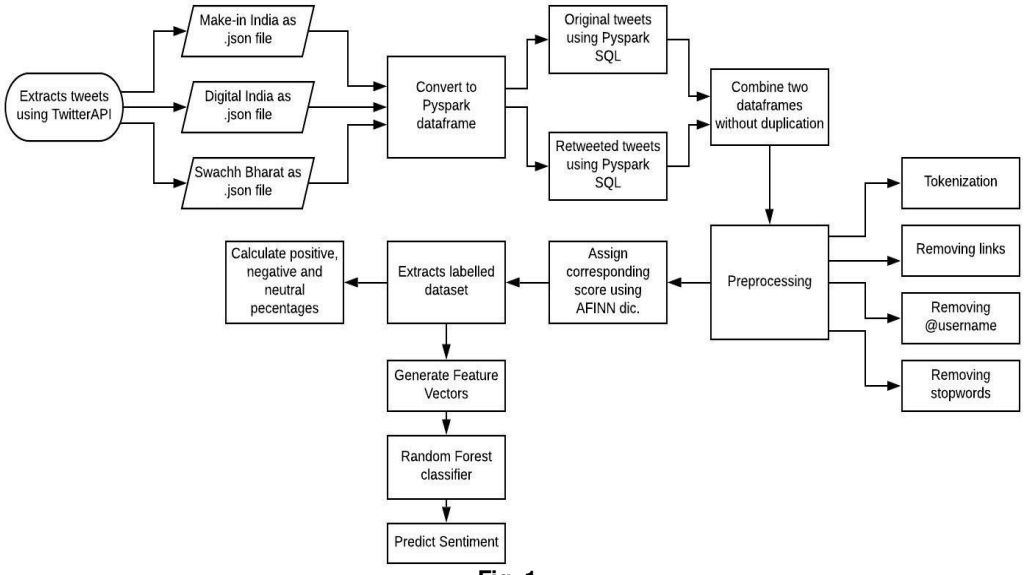

Fig. 1.

III. PROPOSED MODEL

The proposed model has been described in two phases.

A. Phase 1

The sentiment analysis technique consists of four steps. They are:

– **Extraction of tweets:** In the first step, the tweets were extracted by using Twitter API. Totally 7500 tweets were collected regarding three Central Government Schemes namely Make-in India, Digital India and Swachh Bharat. These tweets were saved in a three separate json files. In the Table 2, the total number of tweets for each scheme was given. The Table 2 shows some of the sample data.

– **Pre-processing:** In this step, the extracted tweets have been pre-processed by using pyspark package. This step includes tokenization, removal of stop words, links and unwanted characters.

– **Evaluation of tokens**: After pre-processing the tokens were evaluated by using AFINN Dictionary. This dictionary consists of list of English words which are manually rated with an integer between -5(for negative) and +5(for positive). By using this dictionary the corresponding ratings for each token were assigned and label the tweets according to their scores. The labelled dataset was shown in Table 4. Calculate positive, negative and neutral percentages for each scheme.

Table 5 shows the percentages of all the three schemes.

– **Feature Extraction**: The feature vectors are extracted by using Hashing TF-IDF (Term Frequency - Inverse Document Frequency) method which is widely being used for text mining to reflect the importance of a term to a document in the corpus. Spark MLLIb has a Hashing TF which is a Transformer that takes sets of terms and converts those sets into fixed-length feature vectors. The IDF Model takes feature vectors which are created from HashingTF to rescale it; this generally improves performance when using text as features. Intuitively, it down-weights columns which appear frequently in a corpus. Our feature vectors could then be passed to a classifier.

– **Classification:** First generate the suitable training samples for the classification. We used 5133 training and 2203 test samples for total of 7336 samples. This experiment makes use of bi-grams as a feature selection modal. The subset of the vectors with labels as positive, negative and neutral sentiments are utilized by a Random Forest classifier in training. Then the model generated is used to classify the sentiments in testing. Algorithm1 given below was used to extract and classify the tweets.

Algorithm 1:

Extracted tweets in json files.
Output: Classifying the tweets as negative, positive and neutral.

1. Convert the tweets from json files to Pyspark dataframe by using Pyspark SQL.
2. Split the dataframes into two such as original tweets and re-tweeted tweets.
3. Remove the duplicate records from both the dataframes by using User_id.
4. Combine the two dataframes by using Pyspark Union function.
5. Pre-process the tweets that are extracted from the dataframe by using pyspark package.
6. Assign the corresponding label from the scores by using AFINN dictionary.
7. Calculate the positive, negative and neutral percentages for each scheme by using,

$$\text{Positive percentage} = \frac{\text{no.of positive tweets}}{\text{total no.of tweets}} \times 100$$

$$\text{Negative percentage} = \frac{\text{no.of negative tweets}}{\text{total no.of tweets}} \times 100$$

$$\text{Neutral percentage} = \frac{\text{no.of neutral tweets}}{\text{total no.of tweets}} \times 100$$

8. Extract the feature vector by using Hashing TF and IDF method.
9. Taking the 70% of data as train set and 30% as test set.
10. Fit the model by using Random Forest algorithm.

Table 2, describes that, in case of re-tweeted tweet, the tweet was split into original tweet and re-tweeted tweet in order to get the count of sentiments. And then the tweets were filtered by removing the duplicated tweets by using the User_id.

In Table 3, the extracted tweets were converted to dataframes with the columns such as User_id, User_name, Screen_name, Text and Full_text. One problem is that if the tweets exceed 140 chars then the message in text attribute gets truncated. Since we are using the Streaming API, tweet_mode=extended has no effect in the code. So full_text attribute was used in order to extract the longer text. A new column “Txt_msg” was created in order to get the text message if full_text message has null values.

In Table 4, the tweets were labelled based on their corresponding scores. If the score value exceeds zero then the tweet was labelled as 1 for positive, if the value is less than zero then the tweet was labelled as 0 for negative and if the value is equal to zero then the tweet was labelled as 2 for neutral.

[Table 5], shows the number of positive, negative and neutral tweets which were calculated from the labelled datasets. From the count we have calculated the percentages for each scheme.

Table 2: Total number of tweets.

Schemes	Total no. of tweets	No. of original tweets without duplication	No. of re-tweeted tweets without duplication	No. of tweets	Total no. of tweets without null values
Make-in India	5060	387	4534	4921	4920
Digital India	1560	208	1228	1436	1435
Swachh Bharat	1020	110	872	982	981

Table 3: Sample Data.

User_id	User_name	Screen_name	Text	Full_text	Txt_msg
36922582	Amit Jaiswal	amitjaiswal9	RT @amitsinghap:..	Looks like author...	Looks like author...
1123326827248750592	Kamal Singh	KamalSi18214540	@narendramodi @PM...	null	@narendramodi @PM...
349093636	Nandan Kelkar	nskelkar	RT @narendramodi:...	The @NITIAayogre...	The @NITIAayogre...
3270153152	Gayatri Borpatrag...	GayatriBGohain	For all who thoug...	null	For all who thoug...
266465729	samar slathia	samarslathia	RT @madhukishwar:.	"PM Modi wants to...	"PM Modi wants to...
2169698258	KADARUVEERABRAHMAM	vbkadaru	@vbkadaru @swachh...	null	@vbkadaru @swachh...

Table 4: Labelled Dataset.

Id	Token	Token_clean	Score	Sentiment	Label
940853290732085248	[remarkable, succ...	[remarkable, succ...	4	Positive	1
818377022690848770	[bringing, man, s...	[bringing, stage,...	-4	Negative	0
205146235	[celebration, ama...	[celebration, ama...	5	Positive	1
604188262	[india, survive, ...	[india, survive, ...	0	Neutral	2
468291494	[one, milestone, ...	[one, milestone,...	3	Positive	1
79696673	[shit, lyrics, go...	[shit, lyrics, di...	-6	Negative	0
3060444101	[india, takes, to...	[india, takes, po...	0	Neutral	2

Table 5: Percentages of all the three Schemes.

Schemes	Total no. of tweets	No. of Positive tweets	No. of Negative tweets	No. of Neutral tweets	Positive %	Negative %	Neutral %
Make-in India	4920	1671	1292	1957	33.96 %	26.26 %	39.78 %
Digital India	1435	52	397	510	36.79 %	27.67 %	35.54 %
Swachh Bharat	981	644	49	288	65.65 %	4.99 %	29.36 %
Total	7336	2843	1738	2755	38.75 %	23.69 %	37.55 %

B. Phase 2

As shown in the Table 6, the tweets were analysed based on their location. In this phase, we are separating the tweets based on the location of the users from the labelled dataset. Then count the number of positive, negative and neutral sentiments for each location. Since most of the tweet attributes such as coordinates, place, location etc., have null values, the locations were extracted from the user profiles by using Web Scraping technique. The algorithm2 given below was used to extract the location of each user by using their screen names.

Table 6 shows that, in all regions the number of positive tweets are more than the negative tweets for all three schemes. Here, the four regions of India (North, South, East and West) were alone compared since the other locations such as Not Specified, India and Foreign were not able to find out the particular region. By comparing the average positive percentages from the Table 7, East India has the highest percentage where as the South India has the lowest percentage.

Table 6: Analysis of tweets based on the location.

Schemes		Make-in India	Digital India	Swachh Bharat
Total no. of tweets		4920	1435	981
Not Specified	Total tweets	2262	549	365
	Positive	770	196	230
	Negative	566	163	19
	Neutral	926	190	116
North India	Total tweets	573	190	176
	Positive	191	73	125
	Negative	174	43	8
	Neutral	208	74	43
South India	Total tweets	426	165	94
	Positive	141	56	62
	Negative	125	47	5
	Neutral	160	62	27
East India	Total tweets	228	40	58
	Positive	85	21	39
	Negative	69	14	0

	Neutral	74	5	19
West India	Total tweets	464	164	139
	Positive	163	64	94
	Negative	113	49	6
	Neutral	188	51	39
India	Total tweets	524	145	116
	Positive	177	52	76
	Negative	122	47	9
	Neutral	225	46	31
Foreign	Total tweets	443	182	33
	Positive	144	66	18
	Negative	123	34	2
	Neutral	176	82	13

Table 7: Positive percentages for all the four regions of India.

Regions	Make-in India	Digital India	Swachh Bharat	Average Percentage
North	33%	38%	71%	47%
South	33%	33%	65%	43%
East	37%	52%	67%	52%
West	35%	39%	67%	47%

Algorithm 2

Input: Screen_name of all users
Output: Location of all users
1. Save the screen-names of all the users in a list j
2. For each item i in the list j,
(a) Append the screen-name i with the twitter link as "https://twitter.com/" + i
(b) By using a python library Beautiful soup, pull the data from the above html page.
(c) Search for the div tag which specifies the location.
(d) If the div tag is not none then,
– Find the specific span tag that actually contains the location and then assign the location.
– If the span tag contains smileys or any other images other than the text, then assign the location as "not specified".
– If the span tag contains text in any other languages other than English, then also the location is assigned as "not specified".
(e) Else
– If the div tag is none then location is assigned as "not specified".
– In case of any suspended account, then also location is assigned as "not specified".

IV. RESULTS AND ANALYSIS

In this experiment we have extracted 7500 tweets regarding all the three schemes. Before pre-processing the duplicated tweets were removed by using User_id attribute. Finally 7336 tweets were classified by getting the accuracy of 89.27%. The Fig. 2 shows the percentages of positive, negative and neutral sentiments for all the three schemes.

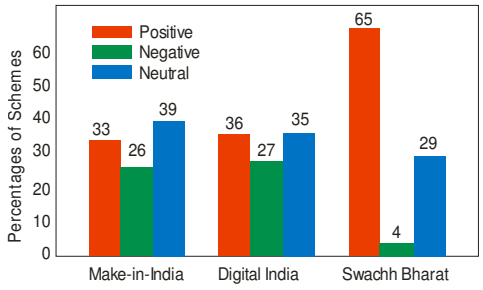

Fig. 2.

With the help of confusion matrix the performance of classifier was evaluated. Fig. 3 shows the relation between correctly and wrongly predicted sentiments. The table of confusion matrix formation is shown in Table 8. From this confusion matrix, different performance evaluation parameter like precision, recall, F-measure and accuracy are calculated. Here the column represents the Predicted values where as the row represents the Actual values.

The main diagonal (474, 745, 746) gives the correct predictions, this is because the actual and predicted values are same. Since the exactness and completeness are more important than the high accuracy, the analytical evaluation of the proposed model will be analyzed using the four parameters, namely accuracy, F-measure, precision, and recall. These results are arranged in Table 9.

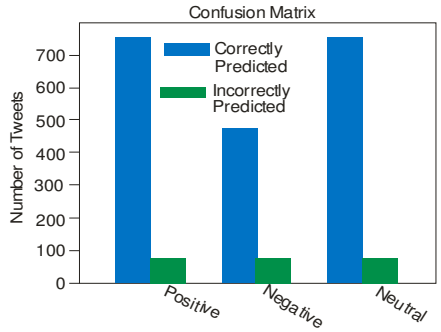

Fig. 3.

Table 8: Confusion Matrix.

Class	Negative	Positive	Neutral
Negative	474	39	36
Positive	13	745	66
Neutral	9	73	746

Table 9: Classification Report.

Class	Precision	Recall	F1-Score	Support
Negative	0.96	0.86	0.91	549
Positive	0.87	0.90	0.89	824
Neutral	0.88	0.90	0.89	828
Total/avg	0.89	0.89	0.89	2201

V. CONCLUSION

Every day high volume of user data is shared on social media sites. Analysing these data would be the tedious one but it's the most valuable thing to develop any businesses. Currently there exist several models to analyse the people's opinions but still need some more optimization process. In this paper a hybrid of lexicon based approach and machine learning approach has been implemented in the Apache Spark framework for Big data. The sentiment analysis has been performed by using AFINN dictionary and Random Forest algorithm. This approach was implemented by using three Central Government Schemes such as Digital India, Swachh Bharat and Make-in India. The Random Forest algorithm performs the multi-class classification which classifies the tweets in three classes such as positive, negative and neutral. Finally the performance of the classifier was analysed by using confusion matrix. According to results, the average Precision, Recall and F-Measure achieves 89%. Thus the above results show that our approach works better both in terms of accuracy and f-measure.

VI. FUTURE SCOPE

As the future work, the analysis should be improved since the neutral sentiments are significantly high. The neutral sentiments are another challenge to perform accurate sentiment analysis. Need to use more evaluation metrics (such as Log-loss, ROC-AUC, etc.,) to improve the performance of the model and thereafter analyse the sentiments of Tamil tweets in big data environment. Even though Random Forest works well in high dimensional data, some dimensionality reduction technique required significantly for the Big data.

ACKNOWLEDGMENTS

The authors delightfully acknowledge the reviewers and editorial board for their valuable comments that leds to the improvement of this manuscript.

Conflict of Interest. No.

REFERENCES

[1]. Sulthana, A. R., Jaithunbi, A. K., & Ramesh, L. S. (2018). Sentiment analysis in twitter data using data analytic techniques for predictive modelling. In *Journal of Physics: Conference Series*, 1-7.

[2]. Gupta, A., Pruthi, J., & Sahu, N. (2017). Sentiment analysis of tweets using machine learning approach. *International Journal of Computer Science and Mobile Computing*, *6*(4), 444-458.

[3]. Gupta, B., Negi, M., Vishwakarma, K., Rawatand, G., & Badhani, P. (2017). Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python. *International Journal of Computer Applications*, *165*(9), 29–34.

[4]. Arora C., & Rachna, Dr. (2017). Sentiment Analysis on Twitter Data, *International Research Journal of Engineering and Technology (IRJET), 4*(6), 31-36.

[5]. Balika, Dr., Chelliah, J., Lathia, D., Yadav, S., Trivedi, M., & Soni, S. S. (2018). Sentiment Analysis of Twitter Data using CNN, International *Journal of Emerging Technologies in Engineering Research (IJETER)*, *6*(4), 97-103.

[6]. Huda, K., Nafis, M. T., & Shaukat, N. K. (2017). Classification Technique for Sentiment Analysis of Twitter Data, *International Journal of Advanced Research in Computer Science*, *8*(5), 2551–2555.

[7]. Huq, M. R., Ali, A., & Rahman, A. (2017). Sentiment analysis on Twitter data using KNN and SVM. *IJACSA) International Journal of Advanced Computer Science and Applications*, *8*(6), 19-25.

[8]. Ahmad, M., Aftab, S., & Ali, I. (2017). Sentiment analysis of tweets using svm. , *International Journal of Computer Applications, 177*(5), 25-29.

[9]. Upadhyay, N., & Singh, A. (2016). Sentiment analysis on twitter by using machine learning technique. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *4*(5), 488-494.

[10]. Pervan, N., & Keleş, H. Y. (2017). Sentiment analysis using a random forest classifier on turkish web comments. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, *59*(2), 69-79.

[11]. Mishra, P., Rajnish, R., & Kumar, P. (2016). Sentiment analysis of Twitter data: Case study on digital India. In *2016 International Conference on Information Technology (InCITe)-The Next Generation IT Summit on the Theme-Internet of Things: Connect your Worlds* (pp. 148-153). IEEE.

[12]. Desai, R. (2018). Sentiment Analysis of Twitter Data : A Survey, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 3*(1), 464-470.

[13]. Shobana, G., Vigneshwara, B. & Maniraj Sai, A. (2018). Twitter Sentimental Analysis, *International Journal of Recent Technology and Engineering (IJRTE)*, *7*(4), 343–346.

[14]. Siddharth, S., Darsini, R., & Sujithra, M. (2018). Sentiment Analysis on Twitter Data Using Machine Learning Algorithms in Python, *International Journal of Engineering Research in Computer Science and Engineering, 5*(2), 285–291.

[15]. Hridoy, S. A. A., Ekram, T. M., Islam, M. S., Ahmed, F., & Rahman, R. M. (2015). Localized twitter opinion mining using sentiment analysis, *Springer, Heidelberg*, *2*(1), 1-19.

[16]. Imane El Alaoui, Youssef Gahi, Rochdi Messoussi, Youness Chaabi, Alexis Todoskof and Abdessamad Kobi, (2018). A novel adaptable approach for sentiment analysis on big social data. *Journal of Big Data*, *5*(12), 1-18.

[17]. Kılınç, D. (2019). A spark-based big data analysis framework for real-time sentiment prediction on streaming data. *Software: Practice and Experience*, *49*(9), 1352-1364.

[18]. Kapočiūtė-Dzikienė, J., Damaševičius, R., & Woźniak, M. (2019). Sentiment analysis of Lithuanian texts using traditional and deep learning approaches. *Computers, 8*(1), 1-16.

How to cite this article: Sujatha, E. and Radha, R. (2020). A Sentiment Classification on Indian Government Schemes Using PySpark. *International Journal on Emerging Technologies*, *11*(2): 25–30.