ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 11(24), DOI: 10.17485/ijst/2018/v11i24/104299, June 2018*

Sentiment Analysis for Odd-Even Scheme in Delhi

Sudhir Kumar Sharma[1]* and Ximi Hoque[2]

[1]Institute of Information Technology & Management, Institutional Area, Janakpuri, New Delhi – 110058, Delhi, India; sudhir_sharma99@yahoo.com
[2]KIIT College of Engineering, Gurgaon – 122102, Haryana, India; hoque.ximi@gmail.com

Abstract

Objectives: This paper analyzes odd-even traffic scheme using tweets posted on Twitter from December 2015 to August 2016. Twitter is a social network where users post their feelings, opinions and sentiments for any event using hashtags and mentions. The tweets posted publicly can be viewed by anyone interested. This paper transforms the unstructured tweets into structured information using open source libraries. Further objective is to build a model using machine learning classification techniques to classify unseen tweets on the same context. **Methods/Analysis**: This paper collects tweets on this event under hashtags. This study explores Dandelion Application Programming Interface for annotation of tweets for academic research. This paper uses machine learning classifications approaches for sentiment analysis and opinion mining. This paper presents empirical comparison of three supervised classification algorithms namely, Multinomial Naïve Bayes, Support Vector Machines (SVM) and Multiclass Logistic Regression. The performances of these classifiers are evaluated through standard evaluation metrics. **Findings**: The experimental results reveal that SVM classifier outperforms the other two classification algorithms. This study may help in decision making of this event to some extent. **Application**: A large number of applications of sentiment and opinion mining can be designed using packages and freely open resources within a time frame now a days.

Keywords: Hashtag, Odd-even Scheme, Opinion Mining, Sentiment Analysis

1. Introduction

Delhi, the capital city of India has more than 25 million populations and more than 9 million registered vehicles. The road traffic in Delhi has grown to a critical level leading to a lot of pollution. The government of Delhi implemented Odd-Even experiment for trial – run basis in two phases of 15 days intervals from 8 A.M to 8 P.M with the objective to reduce air pollution in Delhi. These phases were from 1st – 15th January 2016 and 15th -30th April 2016. The odd-even rule was applied to non-transport four wheeled vehicles (motor cars etc.). This rule would define which car is allowed to play on roads. On the even dates, only cars with registration number ending with an even number were allowed and on the odd dates, cars with registration number ending with an odd number were allowed on the city roads. The public transport buses, trucks, CNG operated passenger / private cars, two wheelers and three wheelers were exempted from the rule. In addition, selective number of VIP and emergency vehicles and cars driven by women were also exempted from this rule[1].

Many studies have analyzed the impact on pollution level in Delhi and traffic conditions in terms of conges-

*Author for correspondence

tion and commuting time. In the first phase there was a 21% reduction in cars and 18% increase in speed. In the second phase, there was a 17% decrease in car numbers and 13% increase in speed. This study concluded that "marginal" reductions (4-7%) of PM 2.5 pollutants during both phases as private cars made a limited contribution to the fine particles in air pollution[2–6]. These studies revealed that traffic density and congestion have been reduced significantly. There is a debate on why the pollution is not reduced[7]. A middle class family who possess one car for commuting daily is more worried about the outcome of the trials. In general, citizens of India and Delhi are keeping the hope of an official declaration of failure / success of the experiment and what would be the next step, either one more trial or implementing the rule permanently or close the pilot project permanently. This paper analyses the opinions, thoughts, feelings, attitude, views and sentiments of citizens about the experiment. This study analyses about what are the citizens talking about this pilot project in social media. Social media are Internet based applications. The well-known social media are Twitter, Facebook, LinkedIn, Stack Overflow and Quora etc. The users of these platforms are increasing day by day due to advancement in Internet and mobile Technologies[8]. It is very easy to connect to these social media sites through mobiles for sharing opinions, feelings, and comments on any topic as per interest. This paper analyses the opinion and sentiment expressed in text messages on Twitter in order to understand the attitude and their feelings towards the odd-even scheme.

Twitter is a micro-blogging service that allows users to communicate with almost 140-characters text messages corresponds to thoughts, opinion and ideas. More than 1 billion people are registered with over 100 millions of them actively engaging their curiosity on a regular monthly basis. Twitter's asymmetric following model exploits a fundamental aspect of human curiosity[9]. A user can follow any other user according to his/her interest and share his/her opinion and ideas on any topics like government's new policy, events, sports, political election, natural hazards, celebrities and public figures. There is no need to be a real person as a Twitter user. A company, organization, an inanimate object and imaginary person are also registered as a user. Twitter allows users to post short status update called tweet in a form of short text message. Tweet text message comprises with hashtag (#OddEven, #Delhi), user mentions (@narendramodi), URLs (http://twitter.com) and places (Delhi).

Nowadays twitter has emerged as one of the most popular platforms for expressing opinions, feelings and thoughts on Internet. It is very useful and obvious to be analyzed for developing many applications. These applications can be utilized as a decision making in marketing for business, political parties and other curious bodies. Governments can also utilize public opinions before or after applying a policy to gauge its effectiveness and acceptance[10].

A team of students in the Computer Science department started study and analysis of sentiment and opinion mining on a live example of odd-even scheme. The objective was to build a model for classifying tweets into positive, negative and neutral/objective according to sentiments they possess using freely available open resources while getting familiar with tools and techniques of this research area. This research paper is an outcome of this project aiming to analyze and develop an efficient system in a fixed time frame. This research work is inspired from the work presented at SemEval-2016 Task 4. SemEval is an international workshop on sentiment and opinion mining on twitter datasets[11]. The objective of this study can be summarized as follows:

Subtask 1: Preparing corpus of relevant tweets in sufficient number on a given context.

Subtask 2: Given corpus of tweets, estimate the distribution of the tweets in the Positive, Negative and Neutral/Objective classes using open source API.

Subtask 3: Given a tweet message, decide whether it has a positive, negative or neutral / objective sentiment.

In the last few years, numerous research papers and studies have focused on sentiment analysis and opinion mining for Twitter. These studies developed many appli-

cations for detecting and identifying sentiment from twitter data[10]. In general, these applications can be divided into three major categories named Machine Learning Approach (ML), Lexicon Based Approach (LB) and the Hybrid Approach. The Machine Learning Approach (ML) uses text features and applies well known ML algorithms. The Lexicon based methods are driven by opinion lexicons, which are a collection of pre-compiled opinion terms and phrases. It is mainly divided into two main approaches namely dictionary based and corpus based. The dictionary based approach uses an opinionated lexicon for calculating the overall sentiment. Corpus based approach uses semantic and statistical methods using an opinionated dictionary. The hybrid Approach combines the ML and LB approaches[12–14].

They categorized and briefly described more than 50 articles on Twitter Sentiment Analysis (TSA). This survey paper discussed current trends, open research challenges and future research direction on TSA[15]. He presented a thorough comparison of twenty-four sentiment analysis methods on eighteen data sets for two tasks: binary classification (positive, negative) and three class classification (positive, negative, and neutral)[16]. His work considers the usefulness of different feature sets, including unigram, bigram, unigram + bigram, and parts of speech tags[17].

The past studies of sentiment classification using ML approaches are not very conclusive about which features and supervised classification algorithms are good for designing accurate and efficient sentiment classification system.

The remainder of the paper is organized in the following manner: Section 2 highlights the experimental setup of the process. Section 3 describes data extraction and lexical diversity of corpus. Section 4 discusses on the automatic annotation of the tweets. Section 5 describes about the Machine learning approach. Experimental results and discussions are given in Section 6. Paper is concluded in Section 7.

2. Experimental Set-up

The experimental setup of the approach presents the research methodology employed, the tools, and libraries used to analyze the even – odd scheme. This section is further categorized into two sub-sections.

We used a laptop of HP, i7, 2.60 GHz with 8GB DDR3 RAM. In this study, open source libraries, packages, APIs are extensively used.

2.1 System Architecture

This subsection discusses the overall architecture of system. This system can be divided in the following three modules as per the three sub tasks discussed in introduction.

2.1.1 Data Extraction

This module is implemented through two sub modules namely Data Extraction and Lexical Diversity. Data extraction sub module is implemented for preparing a corpus of relevant tweets on a topic. Lexical Diversity module is written for measuring the characteristic of the corpus.

2.1.2 Automatic Annotation of Tweets

This module has two sub modules. Pre-processing sub module is implemented prior to annotation sub module. Annotation sub-module is implemented to manage the call for *Dandelion API* for automatic annotation of tweets into positive, negative or neutral classes.

2.1.3 Developing Machine Learning Classifiers

The objective of this module is to empirically compare the three well known classifiers. This module returns a best trained classifier to be used to predict the class of unseen tweets based on positive, negative or neutral sentiment. This module is implemented through many sub modules like feature extraction, feature selection, performance measurement sub modules.

2.2 Tool Used

This subsection discusses the programming language, open source libraries and APIs used in the brief for analysis and developing the system for classification of tweets into three classes.

- Python Programming Language: We used Python 2.7.9 on Windows 10 operating system in this paper. Python is a very powerful object-oriented, high-level programming language. It is an interpreted programming language. Now a days Python is being used for text analysis and text mining.

- Numerical Python (NumPy): Python offers array data structures for numerical data and vectorized operation through NumPy. It is very efficient as compare to other data structures like lists and dictionaries. NumPy produces more compact and readable code through operations with vectors.

- Natural language toolkit (NLTK): It is a well-known package for Natural Language Processing. It provides a friendly interface for many of the common NLP tasks. We used it for removing English stop words from the corpus.

- Pandas: Pandas offers data structures like Series, DataFrame. It provides tools to read and write data between different formats such as CSV, text files, MS Excel spreadsheets and SQL data structures. This paper used pandas for storing tweets in DataFrame format as a convenient database.

- Java Script Object Notation (JSON): We used JSON library of python to parse the json response. This data format has become extremely popular due to its wide use as a way to exchange data between client and server in a web application.

- Scikit –learn: It is main package for machine learning. It is a collection of machine learning algorithms, feature extraction and feature selection methods. It is used for scientific computation by researcher nowadays[18].

- Twitter REST API: It is used for the collection of tweets[19].

3. Data Extraction and Lexical Diversity

This section is divided into two sub-sections.

3.1 Data Extraction

In general, users use Hashtags (#topic) for sharing feelings, opinion etc. on specific and trending topics. Hashtag can be used as a filter to retrieve the tweets on specific topic. A corpus of tweets on the same topic can be created using hashtags. Twitter offers a number of Application

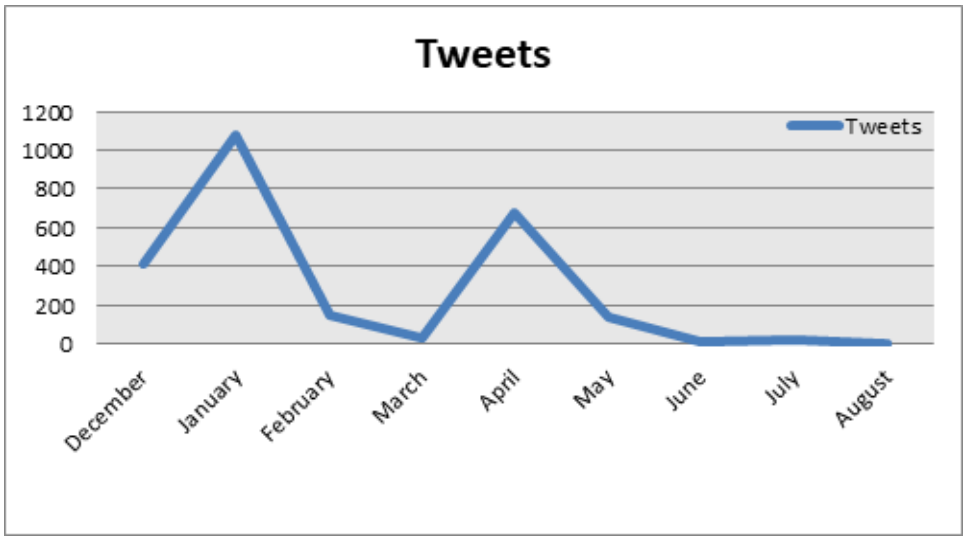

Figure 1. Tweet distribution month wise.

Table 1. Ten samples from the corpus having retweets count more than 100

Sr. No.	DATE	TEXT	RETWEETS
185	4/30/2016 18:27	Should #Delhi have any more #OddEven days this year?	185
186	4/30/2016 18:26	Did #OddEvenphase 2 in #Delhi deliver results?	239
703	4/15/2016 8:25	Rise & Shine #Delhi . Good luck @ ArvindKejriwal& team for Chapter Two #oddeven Do what it takes	418
935	2/11/2016 18:12	World's most polluted city, #Delhi plans new limits on car use http:// reut. rs/1Ta7psj #oddeven	121
1152	1/18/2016 11:25	When odd meets even. BRING BACK #OddEven #Delhi https:// twitter.com/ Joshi_Uncle/status/688948352608673792 â€¦	147
1226	1/16/2016 15:02	Massive #TrafficJams on #Delhi Roads... Want #OddEven days again.	195
1303	1/14/2016 11:25	SC refuses urgent hearing of #OddEven petition, terms plea as "publicity stunt" #Delhi http://www. abplive.in/india-news/sc- refuses-urgent-hearing-of-odd-even-petition-terms-plea-as-publicity-stunt-274259 â€¦ pic.twitter.com/ ziIegMWOND	142
1396	1/9/2016 21:00	Never mind air quality. #OddEven has changed something bigger - #Delhi 's mindset. http:// goo.gl/4BLvVh pic.twitter. com/pccNL5ACoP	644
1481	1/7/2016 15:57	#OddEven Impact Petrol, diesel sales down 25% in #Delhi since odd-even scheme kicked in http:// ecoti.in/JjXaga	237
2125	12/31/2015 19:50	#Delhi 's #OddEven car rule to curb smog comes into effect on Friday - http:// bbc. in/ration #BBCShorts pic.twitter.com/ c4LwnhovV0	139

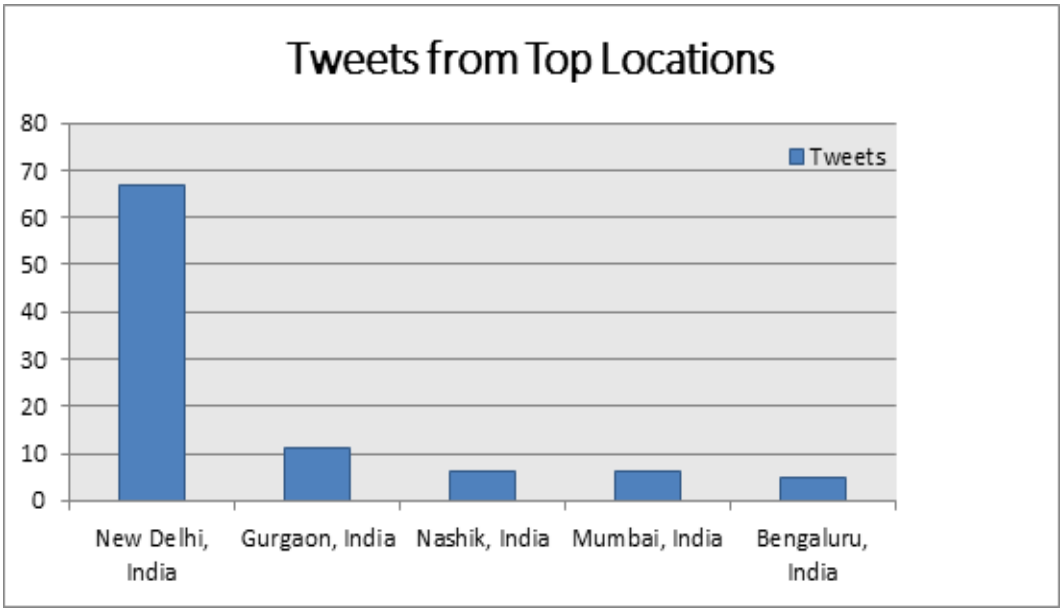

Figure 2. Tweets distribution from top five locations.

Programming Interfaces (APIs), which can be used for automatically extracting data on any event using any of the provided parameters[19]. We collected tweets from December 2015 to July 2016 using hashtags #Oddeven #Delhi. The resulting dataset consists of 2529 tweets. The ten samples of the corpus which have retweets count of more than 100 are given in the following Table 1.

Column 1 shows serial number in corpus. Column 2 shows date and time of the tweet. Column 3 shows the actual raw tweet text. Column 4 depicts retweet count. These retweets are not repeated in corpus.

Figure 1 depicts the tweet distribution month wise. The numbers of tweets are represented on the y-axis and x-axis represents the months from December 2015 to August 2016.

We have only 161 tweets that have non-empty location field in corpus. These tweets are shown in bar chart in Figure 2. Top five locations are shown on x-axis and numbers of tweets are given on y-axis.

3.2 Lexical Diversity

In general, lexical diversity is used as a metric of lexical richness of the corpus. Lexical diversity is defined as the ratio of the number of unique tokens and total number of tokens in the corpus. A corpus of tweets can be characterized in terms of lexical diversity for words, screen names, hashtags and statuses[9]. A python script lexical_diversity.py has been written to provide the lexical diversity measurement of the corpus. These are as follows:

Diversity of Tokens = Number of unique Tokens / Total Tokens = 10406 / 78551 = 0.13

Diversity of screen- names = Unique screen-names / Total screen-names = 1477 / 2529 =0.58

Diversity of hashtags = Unique hashtags / Total hashtag = 1149 / 7236 = 0.16

Average number of tokens per tweet = Total Tokens /Size of corpus =26988 / 2529 =10.67

Average number of hashtags per tweet = Total hashtags /Size of corpus = 7236 / 2529 =2.861

4. Automatic Annotation of Tweets

This section is further categorized into two sub-sections for the sake of explanation and clarity.

4.1 Pre-processing

Pre-processing is an integral part of text analysis and mining applications. The pre-processing phase is further divided into two steps namely, tokenization and removal of English stop words of each tweet in corpus. The task of breaking text message into a list of individual units / tokens is called word tokenization. In general, word_ tokenize () function from NLTK toolkit is used for this task. The NLTK toolkit separately tokenizes the '#' from hashtags and '@' from mentions and URL is not tokenized as a single unit. Tweet_Tokenizer can be used as an alternate for Twitter content[8]. To overcome some problems in tokenization process of previous methods, this paper implemented regular expression for the same. This method tokenized the words as a single unit of '#hashtags', '@-mentions', emoticons, URLs. This method groups 'HTML tags', '@mentions', '#hashtags', 'URLs', 'numbers', 'words with – and', 'other words or anything else' as a single token[20].

The text message may contain information which are not required for designing the system e.g. URLs, mentions and stop words. In general stop words have least discriminative power in determining overall sentiment of a tweet. A list of English stop words is available in NLTK corpus. NLTK toolkit is used for removing these stop words. We applied Unicode filtering for replacing the Unicode characters to null. The following tokens have been removed in this step.

- Unicode, URL, stop words, punctuation and single digit number etc.
- Miscellaneous tokens such as bit, ly, via, com, twitter, instagram, facebook etc.

The hashtags and emoticons may be composed of opinion words sometimes. They have not been removed.

4.2 Dandelion API for Annotation

Dandelion API was used for subtask 1. Dandelion was agreed to give limited service free of cost for academic research. Dandelion platform[21] estimated the distribution of the tweets in the Positive, Negative and Neutral/ Objective classes. This API also assigned label of positive, negative and neutral to each tweet along with their respective scores. We send the tweets one by one after pre-processing using GET request. This platform returns the response in json format of our query. Dandelion system has made a dictionary automatically using opinionated content drawn from the web e.g., product reviews, user comments, etc. The system has extracted features like unigrams, bigrams from the text along with the opinionated dictionary. The punctuation marks, emoji or emoticons are managed separately. The pseudo code of this module is presented in the following Figure 3.

```
Input: Sending Tweets one by one through GET request
Output: Estimated the distribution of tweets into three classes and also
assignslabel with numerical score.
a. First you have to register yourself at Dandelion official site and get
the access token.
b. SENTIMENT_URL = 'https://api.dandelion.eu/datatxt/sent/v1'
c. RESPONSE = request.get(SENTIMENT_URL, params = payload)
d. payload = {
   'token': token,
   'text': text,
    'lang': 'en' //English language
   'social.hashtag': True,
'social.mention' : True}
e. query = text[line]
f. response = get_sentiment(token, query)
g. score.append(response['sentiment']['score'])
h. sentiment.append(response['sentiment']['type'])
```

Figure 3. Pseudo code for accessing Dandelion API.

5. Machine Learning Approach

This paper explores the machine learning classifiers and feature selection strategy for solving the subtask 3 defined in introduction. This section is further divided into three subsections namely feature extraction and feature selection, used classifiers and evaluation measures for classifiers. This paper empirically compares the per-

formance of three classifiers using standard evaluation measurement. This module returns a best trained classifier that can be used to label the unseen tweet into any one class namely positive, negative or neutral/objective.

5.1 Feature Extraction and Feature Selection

Now each tweet is represented as a bag-of-words. The word / token as a single unit is used as a feature of each tweet. This method is called Unigram, that is a special case of n-grams where n=1. Now each tweet text can be represented by a vector with one component corresponding to each term in vector space. Many machine learning algorithms have not accepted text terms as an input. We need a method that assigns weight to each token. This transformation is called weighting scheme. This paper employs TF-IDF approach for assigning real number to each term. This method returns a real number that is a product of Term Frequency (TF) and Inverse Document Frequency (IDF). TF represents how often a word appears in a tweet. IDF represents how rare a word is across a collection of tweets[18]. This paper calculates the weightage of i^{th} term in j^{th} tweet as follows:

$$W_{i,j} = TF_{i,j} \times IDF_i \quad (1)$$

where, $TF_{i,j} = \log^{(f_{i,j})} + 1$ where $f_{i,j}$ is the number of occurrence of i^{th} term in j^{th} tweet text.

$$IDF_i = (1 + log(N|n_i)) \quad (2)$$

where, N = number of tweets in corpus and number of tweets in which i^{th} term occur.

This paper only uses terms that have minimum frequency of occurrence twice in the set of tweets. Those terms have been removed from the feature vectors that have maximum frequency of occurrence more than 70 % in the set of tweets.

This paper also employs *SelectPercentile* method of feature selection strategy of *scikit-learn*. This method selects the best univariate feature selection strategy with hyper-parameter search estimator. User can set how much percentile top ranking features are required without degrading performance of the system[22].

5.2 Used Classifiers

We used three classification algorithms belonging to different class in this paper for empirically comparing the results. The brief introductions of these algorithms are as follows.

5.2.1 Multinomial Naive Bayes

The multinomial naive Bayes implements Naïve Bayes algorithm (NB) for multinomially distributed data. The NB is based on Bayes' theorem with the assumption of independency between features. It is widely used classifier for classification of documents[22]. This paper employs 0.5 for smoothing parameter.

5.2.2 Support Vector Machines

Multiclass support vector machines have been implemented as 'one-vs-all' approach for multi-class classification[22]. It is very effective in high dimensional spaces. This paper uses a linear kernel function.

5.2.3 Multiclass Logistic Regression

It is a linear model for classification. It is based on cross-entropy loss. It is also known as maximum-entropy classification[22]. This paper employs limited-memory Broyden Fletche Shanno optimization algorithm.

5.3 Evaluation Measures for Classifiers

This section discusses the evaluation measures for three- class classification system. The standard evaluation matrices are calculated on the basis of the entries of the confusion matrix. In this paper, four measurements namely accuracy, precision, recall and F-score are used for the assessment of efficiency of our classifiers at classifying the unknown tweets. Accuracy of system is defined as the ratio of total true predicted tweets to the total num-

ber of tweets in the test set. Precision, recall and F-score are defined with respect to each class namely positive, negative or neutral. Precision is defined as a ratio of correctly predicted tweets to the total number of predicted tweets in a class. Recall is a ratio of correctly predicted tweets to the total number of actual tweets in a class. The F-score can be defined as the harmonic mean of precision and recall. A better classification system has maximum values of all four standard metrics[15].

6. Experimental Results and Discussion

This section presents the results of subtask 1 and subtask 2. The following Figure 4 depicts the tweet distribution into three classes month-wise.

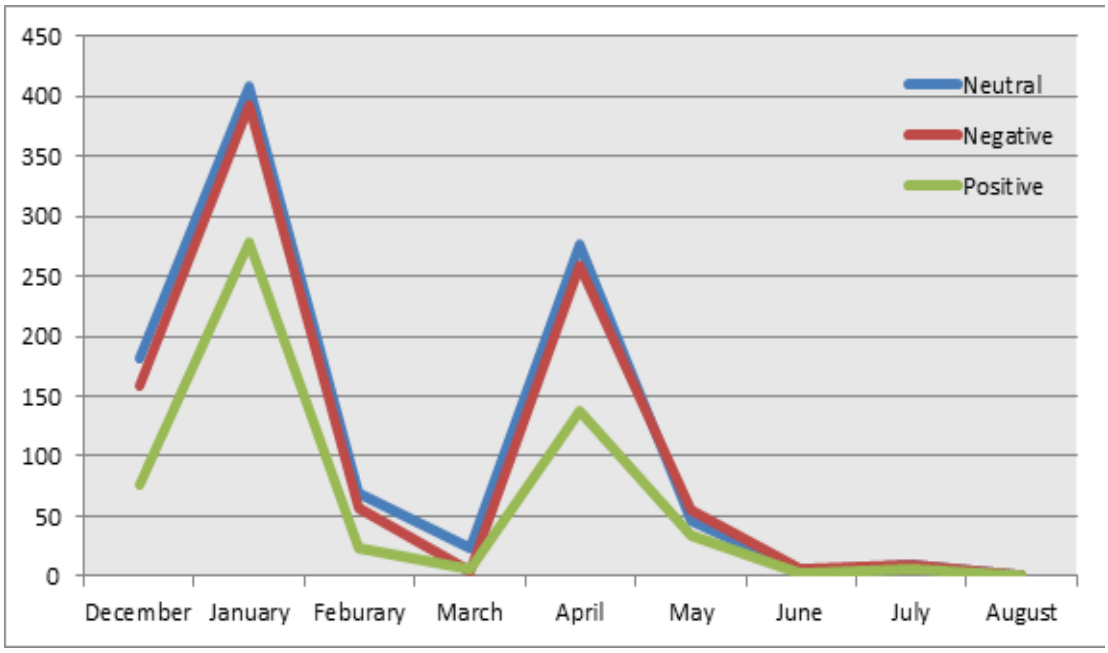

Figure 4. Distribution of tweets into three classes month wise.

Table 2. Training and testing data set

Classes	Training set	Testing set	Total
Negative	844	96	940
Neutral	912	111	1023
Positive	520	46	566

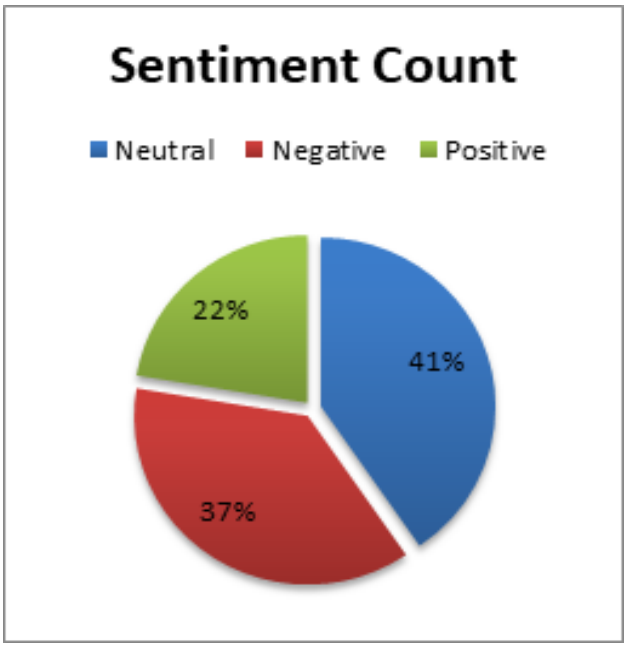

Figure 5. Overall sentiment distribution.

Figure 5 shows the overall sentiment percentage of the total tweets.

The whole data set is randomized first and then split into training and testing set into the ratio of 90 % to 10%. The details of these sets are given in the following Table 2. The numbers of instances in training and testing set are given in column 2 and column 3 in Table 2.

The three selected classification algorithms were trained on selected features using training set. Some trial and error experiments are executed for finding the optimum training parameters. The final performances of these classifiers are measured on testing set.

Figure 6 explains the variation in accuracy by changing the percentile of top features of three different classifiers.

From Figure 6 it is concluded that the accuracy of the system didn't improve by taking more features. We considered only top 10 percentile of features.

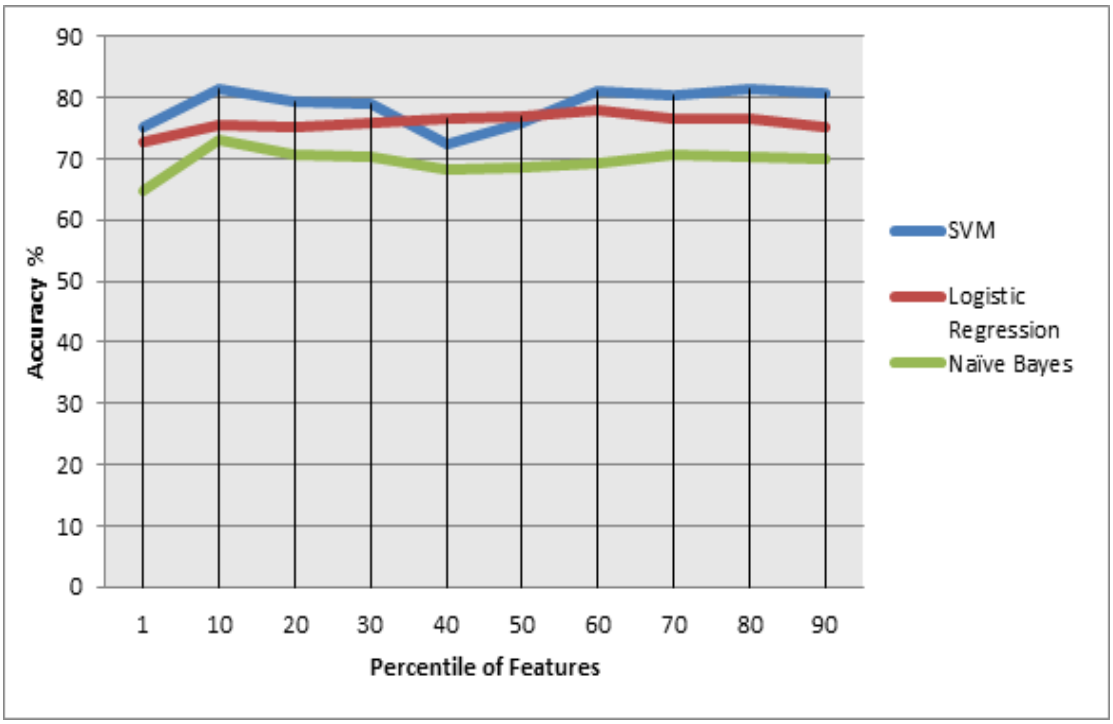

Figure 6. Accuracy variation with percentile of features selected.

Table 3. SVM Multilabel

	Precision	Recall	F-score	Accuracy
Negative	0.86	0.69	0.76	81.42
Neutral	0.77	0.92	0.84	
Positive	0.86	0.80	0.83	
Average	0.82	0.81	0.81	

Table 4. Multi class logistic regression

	Precision	Recall	F-score	Accuracy
Negative	0.83	0.60	0.70	75.49
Neutral	0.71	0.91	0.80	
Positive	0.82	0.72	0.77	
Average	0.77	0.76	0.75	

Table 5. Multi nominal Naïve Bayes

	Precision	Recall	F-score	Accuracy
Negative	0.71	0.70	0.71	73.12
Neutral	0.71	0.85	0.77	
Positive	0.92	0.52	0.67	
Average	0.75	0.73	0.73	

Simulation results are presented in Tables 3–5. Precision, Recall, F-score, and Accuracy are shown in these tables.

Although all three classification algorithms are able to assign labels of unseen tweets into three classes namely positive, negative and neutral/objective. The best

performing SVM algorithm achieves 0.81 and 81.42 % average F-score and accuracy respectively. The least performing Naïve Bayes algorithm achieves 0.73 and 73.12 % average F-score and accuracy. On the basis of simulation results, the performance of Naïve Bayes algorithm is least in comparison of all three algorithms considered in this study. The simulation reveals that the performance of multiclass SVM classifier is significantly better than the other two classifiers.

The built machine learning model using SVM can be used to classify the unseen tweets for the same context.

It is concluded that the proposed model using Support Vector Machine classifier can be considered a statistically well-performing system for the sentiment analysis and opinion mining task which is comparable to the other models available in literature. The research methodology used in this paper can be applied for sentiment analysis and opinion mining for any other context. The advantages and disadvantages of annotations of tweets using *Dandelion API* can be further explored in future.

7. Conclusions

In this paper, we analyzed the sentiments and opinions on twitter data for Odd-Even traffic scheme, Delhi using freely available open resources. This paper employed *Dandelion API* for annotation of the tweets into three classes. We empirically compared the three classifiers namely, Support Vector Machines (SVM), Multinomial Naïve Bayes and Multinomial Logistic regression Classifier. The simulation results revealed that SVM outperforms the other two classifiers.

The proposed model may be applied for decision making of similar events to some extent. The proposed approach can be explored for sentiment analysis and opinion mining for any other context.

8. References

1. Odd-Even formula: Delhi Government's Notification [Internet]. 2015 Dec 28. Available from: http://it.delhigovt.nic.in/writereaddata/egaz20157544.pdf.
2. Chaudhari PR, Verma SR, Singh DK. Experimental Implementation of odd-even scheme for air pollution control in Delhi, India. International Journal of Latest Research in Engineering and Technology. 2016; 2(21): 57–65.
3. Pavani VS, Aryasri AR. Pollution control through odd-even rule: A case study of Delhi. Indian Journal of Science. 2016, 23(80):403–11.
4. Analysis of Odd-Even scheme phase-II [Internet]. Available from: http://www.teriin.org/files/TERI-Analysis-Odd-even.pdf.
5. Goel R, Tiwari G, Mohan D. Evaluation of the effects of the 15-day odd-even scheme in Delhi: A preliminary report. Transportation Research and Injury Prevention Programme Indian Institute of Technology Delhi ; p.1–18.
6. Ministry of Environment Forest & Climate change. Report on ambient air quality data during ODD and EVEN period; 2016 Apr 15th - 30th; 2016.
7. Parikh J, Parikh K. Making odd-even work better. Sunday Business; 2016 Apr 10.
8. Bonzanini M. Mastering social media mining with Python; 2016.
9. Russell MA. Mining the social web: Data mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More. O'Reilly Media, Inc.; 2013 Oct 4.
10. Liu B. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies. 2012 May 22; 5(1):1–67.
11. Ciubotariu CC, Hrişca MV, Gliga M, Darabană D, Trandabăţ D, Iftene A. Minions at SemEval-2016 Task 4: Or how to build a sentiment analyzer using off-the-shelf resources? Proceedings of SemEval; 2016. p. 247–50.
12. Medhat W, Hassan A, Korashy H. Sentiment analysis algorithms and applications: A survey Ain Shams Engineering Journal. 2014 Dec 31; 5(4):1093–113.
13. Imran M, Castillo C, Diaz F, Vieweg S. Processing social media messages in mass emergency: A survey. ACM Computing Surveys (CSUR). 2015 Jul 21; 47(4):67. Crossref.
14. Pang B, Lee L. Opinion mining and sentiment analysis. Foundations and trends in information retrieval. 2008 Jan 1; 2(1–2):1–35. Crossref.
15. Giachanou A, Crestani F. Like it or not: A survey of twitter sentiment analysis methods. ACM Computing Surveys (CSUR). 2016 Jun 30; 49(2):28. Crossref.

16. Ribeiro FN, Araújo M, Gonçalves P, Gonçalves MA, Benevenuto F. SentiBench-a benchmark comparison of state-of-the-practice sentiment analysis methods. EPJ Data Science. 2016 Dec 1; 5(1):1–29. Crossref.
17. Go A, Bhayani R, Huang L. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford 1. 2009; 12:1–6.
18. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011 Jan 2; 12:2825–30.
19. Twitter Documentation [Internet]. 2017 Jun 07. Available from: https://dev.twitter.com/overview/documentation.
20. Marco Bonzanini [Internet]. 2015 Mar 09. Available from: https://marcobonzanini.com/2015/03/09/mining-twitter-data-with-python-part-2.
21. Sentiment Analysis: detect sentiment and emotions in short texts [Internet]. Available from: https://dandelion.eu/semantic-text/sentiment-analysis-demo/?appid=it%3A333903271&exec=true.
22. Hackeling G. Mastering machine learning with scikit-learn. Packt Publishing Ltd; 2014. p. 1–238.