ISSN (e) 2520-7393
ISSN (p) 2521-5027
Received on 21st June, 2019
Revised on 10th July, 2019
www.estirj.com

Performance Analysis of off the shelf and customized Twitter Sentiment Classification Models

Rakesh Kumar[1], Sheeraz Memon[1], Muhammad Bux Alvi[2]

[1]*IICT, Mehran University of Engineering and Technology, Jamshoro*
[2]*The Islamia University of Bahawalpur*

Abstract: Advent of social networks has provided voice to common people. People utilize these medium to raise their issues or express their point of view about a trending topic. Especially Twitter, that has become de facto standard to discuss current issues, problems, events and policies. This scenario demands computer mediated systems that may be able to produce their collective opinion. Research is being carried out to develop sentiment classification models that may use social media data through general purpose programming languages like python or by using off the shelf tools such as Rapid Miner. This work is an attempt to know the comparative suitability of customized built models using python and off the shelf tools made models using Rapid Minder on twitter text data. We found customized built sentiment classification models more efficient, flexible and robust than off the shelf solutions. Customized twitter sentiment classification model, developed using python, achieved 3-4 points more accuracy than off the shelf classification models. Additionaly custom sentiment classification models takes less amount of time in building dictionary, training models and predicting class labels for given tweets.

Keywords: *Machine learning, off the shelf tools, Rapid Miner, sentiment analysis, Twitter Sentiment Classification (TSC)*

1. Introduction

Increased usage of communication tools and technologies has resulted in the storage of huge amount of data. Such data generation resources include but not limited to organizational blogs, product reviews, Facebook posts, comments and Twitter tweets. Among them, Twitter has become de facto social media network for people to express their inclination towards current trending topic or event [1, 2].

Accumulation of such huge amount of data instigates researcher to mine them and try to know collective sentiment of the folk about a policy, event, product or service. Sentiment analysis is the field of knowledge that helps to know about sentiment polarity of an individual about an entity [3, 4]. Researchers around the world are actively developing various models to find out overall sentiment orientation of masses on a particular topic using twitter data [5-8] . Twitter sentiment analysis system can be developed using any one of the following techniques: 1) machine learning 2) ensemble 3) lexicon based 4) hybrid techniques i.e. combination of above three methods.

Since Twitter text data is highly prone to noise and inconsistencies in tweeting styles, therefore, a lot of research work is also centered around data wrangling. Data wrangling is the set of techniques that can clean and standardize the given twitter dataset and prepare it for forwarding it to any of given four modeling techniques. In [9-11], the authors have been actively investigating the impact of text pre-processing techniques on final accuracy of the models. They have experimented with different text cleaning techniques such as case conversion, stop work removal, twitter specific cleaning, spelling correction, word length handing, negation handling etc., separately and in combination to determine best text cleaning pipeline. Other major work in twitter sentiment analyses encompasses feature extraction methodologies [12-15] using n-grams, tf-idf and word count frequencies.

More work on twitter sentiment analyses attempts to find the suitability of applying machine learning algorithms [16, 17]. Machine learning algorithms such as naïve bayes algorithms, support vector machine based classifier; logistic regression and random forest are quite popular for text processing. Ensemble technique is also practiced by the researchers [12, 18-20]. Ensemble methods are very effective methods to enhance the robustness and accuracy of the models. Another area for exploring new methods for twitter sentiment analysis is combination of all the previous techniques i.e. hybrid models [21-23].

All the previous work is expedited using any popular programming language such as python and supported libraries or off the shelf tools such Rapid Miner that have built in text pre-processing modules and machine learning algorithms. A few people have also tried their combination.

This work encompasses implementation and performance comparison of twitter sentiment models built using off the shelf tool such as Rapid Miner and python with allied libraries as shown in figure 1. Python allied libraries include pandas, natural language tool kit (NLTK), sci-kit learn (sklearn) and matplotlib. Pandas is utilized for data analysis, NLTK is used for text pre-processing, Sci-kit

Corresponding author Email address: *oad.rakesh@gmail.com*

learn helps in modeling while matplotlib is used in result visualization. We have used same text pre-processing steps for both the models to keep both models as par and used Naïve Bayes and support vector machine classifiers for modeling purpose.

We found custom built models using python more efficient in comparison to the models developed using off the shelf tool i.e. Rapid Miner. Custom built models are 1) more flexible 2) more efficient and 3) more robust. These results are obtained using (75-25) % dataset train test split division strategy.

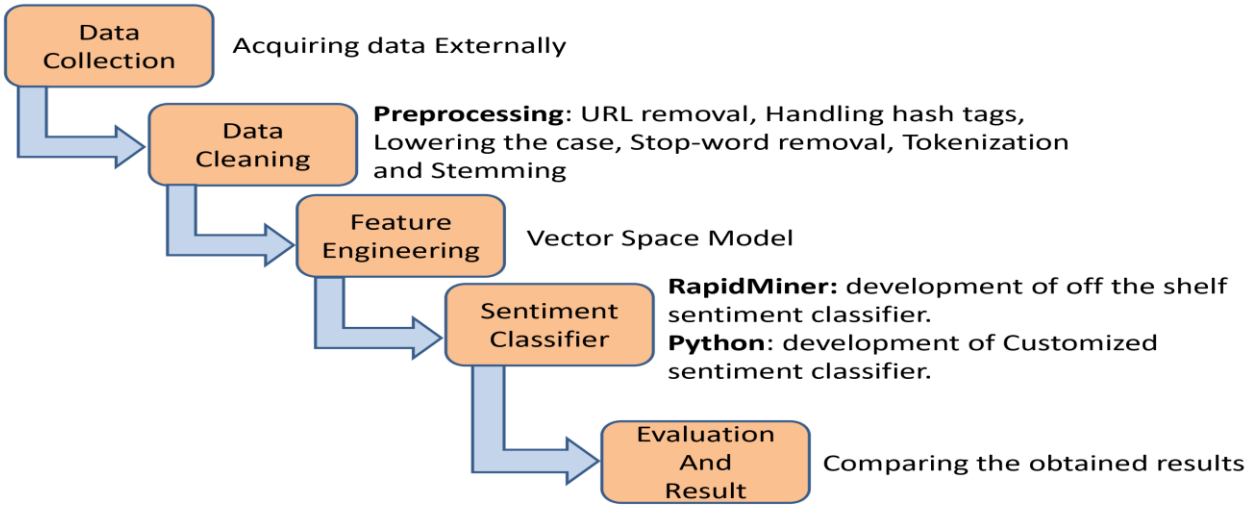

Figure. 1. Workflow of the Models

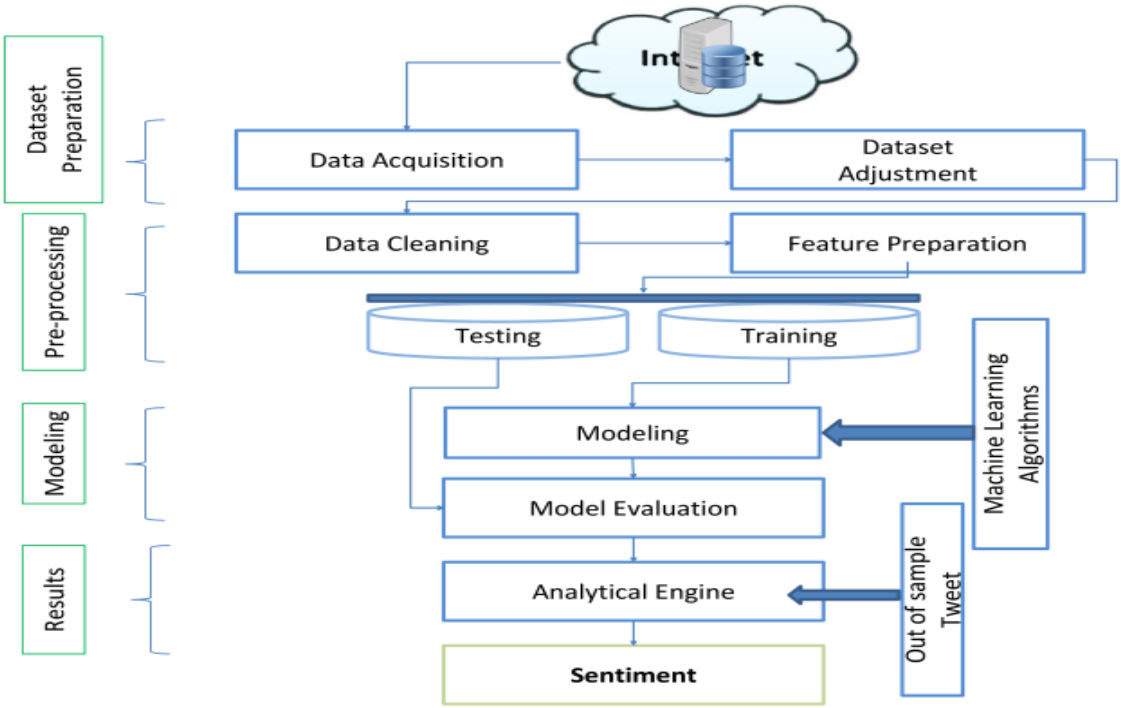

Figure. 2. Methodology for Customized Sentiment Classification Model

This work is organized as: 1) Introduction 2) Experimental Methodology 3) Experimental Results and Discussion 4) Research conclusion 5) Future Direction and at the end references.

2. Experimental Methodology

The figure 2 depicts the details of experimental methodology expedited in this work. Figure.2. Experimental Methodology

2.1 Self Drive care Dataset Acquisition

The twitter dataset has been acquired externally from online repository[1]. The obtained dataset is already labeled.

[1] data.world/crowdflower/

The dataset is about acceptability of common people about self-drive cars. Some people may be in favor of adapting new technology while others may be against its usage on the grounds of security issues or think it to be too early. Another category of people may have neutral views on this topic. Table 1 represents some statistics about these dataset. The obtained datasets have majorly two parts i.e. tweets and meta-data. The dataset is manually annotated by experts.

Table 1: Dataset Statistics

Dataset Theme	Self-Drive Cars
Total Tweets	7156
Positive Tweets	1904
Negative Tweets	795
Neutral Tweets	4248
Missing Value Tweets	209
Duplicate Tweets	10

2.2 Data Wrangling

The obtained dataset is cleaned by handling missing value tweets, removing duplicate and irrelevant tweets as a first step. Afterwards, the dataset is converted into pandas dataframe then data cleaning pipelineis performed.

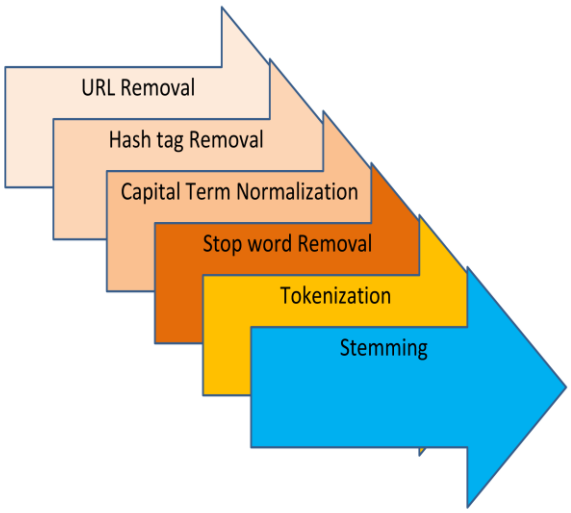

Figure.3. Data Wrangling Pipeline

Data wrangling also referred as data cleaning is most important task for any text analytics application. Since twitter data is more prone to noise and tweeting style of each individual Twitter user, therefore, it needs more attention. The given data cleaning pipeline removes noise from the given dataset and produces cleaner dataset.

The data cleaning pipeline, as given in figure 3, does not only remove noise from the dataset but also brings the tweets in standard format. Initially given tweets are freed from URLs, hash tags and retweets.

The next data cleaning step is case normalization and stemming processes. Case normalization changes the case of terms and converts them to lower case. Stemming changes all inflected form of words and brings them into one form which is called stem. An alternate stem like normalization method includes lemmatization which requires more computational resources to implement. While using lemmatization, the tradeoff between resource utilization and efficiency is not favorable, therefore, lemmatization is not used in this work.

The impact of stemming is shown in figure 4. People may have used infinitive or its forms, nouns or their variations in their tweets. All of them increase the complexity of vector space model (VSM). Stemming not only brings the text in standard form but also mitigates document term matrix (DTM).

Another major data clearing step is to handle language stop word. There are 153 defined language stop word in natural language tool kit (NLTK). These terms rarely have any sentiment impact. Therefore, they all are disregarded in this work. Removal of stop words is also a step to reduce the feature set size of given dataset.

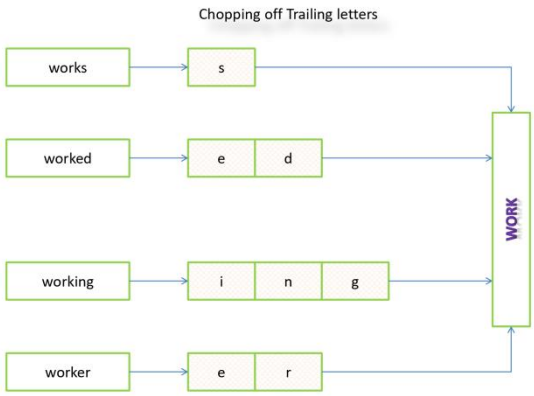

Figure.4. Stemming

2.3 Feature Preparation

Features are the most significant attributes that may be enough to determine the response for a tweet. We have sued vector space model (VSM) to extract features. There are two possible implementations of VSM. 1) Binary term occurrence i.e. either a term is present or absent in the given document 2) frequency based term occurrence, in which the system will count the number of times a term appears in the document. We have used the former one in this work. Furthermore, since we have unstructured data and unstructured data cannot be directly feed into machine learning algorithms; therefore, all features are converted using text Vectorization technique.

Repeated random test train split method is used to divide the feature set to train the model using Multinomial Naïve Bayes and support vector machine learning algorithms and then evaluate the same. We have used (75-25) % division strategy to perform training and testing respectively. That is 75% of feature set is used to train each model and 25% of feature set is used to test the trained

model for evaluation purpose. The following equation represents the goal of developed model.

$$clf : tweet \mid class_{+|-}$$

where clf represents Twitter Classifier, tweet is the to be assigned positive or negative class.

We have used accuracy_ score as our evaluation metric. Accuracy score may be calculated manually or confusion matrix may be used for the same purpose. Accuracy is given by;

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN)$$

where: TP = True positive; FP = False positive; TN = True negative; FN = False negative

3. Experimental Results and Discussion

Table 2, Table 3, Table 4 and Table 5 represent detailed experimental results using Multinomial Naive Bayes and Linear SVC machine learning algorithms. Selection of these two-machine learning algorithms is due to their effectiveness for natural language processing application and coverage of two popular families i.e. Probabilistic and Non-probabilistic machine learning techniques.

Table.2. Represents results (linear SVM Classifier)

Customized Model using LinearSVC

01	Feature Extraction Model	VSM
02	Dataset Split	(25-75)%
03	Dictionary Building Time	140 milliseconds
04	Training Time	90 milliseconds
05	Prediction Time	30 milliseconds
06	Training Tweets	2024
07	Testing Tweets	675
08	Features	3224
09	**Accuracy**	**76.47 %**

Table.3. Represents results (linear SVM Classifier)

Off the shelf Model using LinearSVC

01	Feature Extraction Model	VSM
02	Dataset Split	(25-75)%
03	Dictionary Building Time	4.1s
04	Training Time	5.0s
05	Prediction Time	730 milliseconds
06	Training Tweets	2024
07	Testing Tweets	675
08	Features	5024
09	**Accuracy**	**72.80 %**

Execution time and accuracy are two main criteria for evaluating these two models i.e. customized sentiment classification model and off the shelf sentiment classification model. As shown in Table 2 and Table 3, we have used Linear SVC machine learning algorithms and same text pre-processing steps but dictionary building time if significantly less in case of customized model i.e. 140 milliseconds and 4.1 seconds for off the shelf model. Similarly there is huge difference for training time and predicting time. Furthermore, the customized model achieves nearly 4 point more accuracy with customized model.

With Multinomial Naïve Bayes algorithm as given in Table 4 and Table 5, the customized model builds the dataset dictionary within 120 milliseconds, model is trained within 75 milliseconds and 25 % dataset labels gets predicted in 22 milliseconds. On the contrary, off the shelf model takes 3.2 seconds, 4.25 seconds and 640 milliseconds for dictionary building, training and predicting respectively. When comparing the accuracy, the former model achieves 75.25 % and later obtains 71.75 %. Again 3.5 point deficit.

These are the statistical advantages achieved after experimental analysis. In practice, the customized model using python gives flexibility to the model developer. It is very complicated and tedious task to work with text data. With change in dataset, we have to change the model altogether, specially its text data cleaning modules. In Rapid Miner, there are fixed entitites to work with. However, in python and its supported libraries you have huge ecosystem available along with facility to customize things as per your requirement. Such flexibility also makes the developed model more robust to tackle new datasets along with added advantage of scalability.

Table.4. Represents results (MNB Classifier)

Customized Model using Multinomial NB

01	Feature Extraction Model	VSM
02	Dataset Split	(25-75)%

03	Dictionary Building Time	120 milliseconds
04	Training Time	75 milliseconds
05	Prediction Time	22 milliseconds
06	Training Tweets	2024
07	Testing Tweets	675
08	Features	3224
09	**Accuracy**	**75.25 %**

Table. 5. Represents results (MNB Classifier)

Off the shelf Model using Multinomial NB

01	Feature Extraction Model	VSM
02	Dataset Split	(25-75)%
03	Dictionary Building Time	3.2 seconds
04	Training Time	4.25seconds
05	Prediction Time	640 milliseconds
06	Training Tweets	2024
07	Testing Tweets	675
08	Features	4845
09	**Accuracy**	**71.75 %**

4. Research Conclusion

In this work, we have developed two models 1) custom built model using python and data analyses libraries 2) off the shelf model using Rapid Miner. We found:

1. Custom built classification model more flexible
   - Pre-processing can be performed as per requirement and dataset.
   - Machine learning algorithms can be customized as needed.
2. Custom built classifer is more efficient
   - It obtained 3-4 points more accuracy than off the shelf model.
3. Mitigating vector space
   - Custom based model is better at mitigating feature vector space as also depicted in Table 2 and Table 4.

5. Future Direction

As future works, we propose:

1. Testing of more configuration of text pre-processing stages
2. Application of more machine learning and ensemble methods for the comparison
3. This research can be extended by more dataset
4. This work can be tested using other train test split dataset division or K-fold division strategies.

References

[1] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *LREc*, 2010, pp. 1320-1326.

[2] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, 2011, pp. 30-38.

[3] L. Bing and L. Zhang, "A Survey of Opinion Mining and Sentiment Analysis."

[4] B. Pang and L. Lee, "Opinion mining and sentiment analysis," 2008.

[5] M. A. Razzaq, A. M. Qamar, and B. Hafiz Syed Muhammad, "Prediction and analysis of Pakistan election 2013 based on sentiment analysis," pp. 700-703, 2014.

[6] S. Banik, A. F. Khodadad Khan, and M. Anwer, "Hybrid machine learning technique for forecasting Dhaka stock market timing decisions," *Comput Intell Neurosci*, vol. 2014, p. 318524, 2014.

[7] V. M. Prieto, S. Matos, M. Alvarez, F. Cacheda, and J. L. Oliveira, "Twitter: a good place to detect health conditions," *PloS one*, vol. 9, p. e86191, 2014.

[8] M. B. Alvi, N. A. Mahoto, M. A. Unar, and M. A. Shaikh, "An Effective Framework for Tweet Level Sentiment Classification using Recursive Text Pre-Processing Approach," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 10, 2019.

[9] D. Effrosynidis, S. Symeonidis, and A. Arampatzis, "A Comparison of Pre-processing Techniques for Twitter Sentiment Analysis," vol. 10450, pp. 394-406, 2017.

[10] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, "A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis," *Expert Systems with Applications*, vol. 110, pp. 298-310, 2018.

[11] Z. Jianqiang, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis" *IEEE Access*, 2017.

[12] M. S. Akhtar, D. Gupta, A. Ekbal, and P. Bhattacharyya, "Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis," *Knowledge-Based Systems*, vol. 125, pp. 116-135, 2017.

[13] F. Koto, "A Comparative Study on Twitter Sentiment Analysis: Which Features are Good?," *Conference Paper*, 2015.

[14] J. Liang, X. Zhou, L. Guo, and S. Bai, "Feature Selection for Sentiment Classification Using Matrix Factorization," pp. 63-64, 2015.

[15] Y. Yang, "A comparitive study of feature selection in text categorization."

[16] B. Baharudin, L. H. Lee, and K. Khan, "A Review of Machine Learning Algorithms for Text-Documents Classification," *Journal of Advances in Information Technology*, vol. 1, 2010.

[17] B. Gupta, "Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python," 2017.

[18] Ankit, "An Ensemble Classification System for Twitter Sentiment Analysis," *procedia*, 2018.

[19] M. Camara, "Ensemble classifier for Twitter Sentiment Analysis," 2015.

[20] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, pp. 170-179, 2014.

[21] M. B. Alvi, N. A. Mahoto, M. Alvi, M. A. Unar, and M. A. Shaikh, "Hybrid Classification Model for Twitter Data-A Recursive Preprocessing Approach," in *2018 5th International Multi-Topic ICT Conference (IMTIC)*, 2018, pp. 1-6.

[22] M. Z. Asghar, F. M. Kundi, S. Ahmad, A. Khan, and F. Khan, "T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme," *Expert Systems*, vol. 35, p. e12233, 2018.

[23] F. H. Khan, S. Bashir, and U. Qamar, "TOM: Twitter opinion mining framework using hybrid classification scheme," *Decision Support Systems*, vol. 57, pp. 245-257, 2014.

About Authors

Rakesh Kumar: He did Bachelor of engineering from Quaid e Awam University of Engineering, Science and Technology, Nawabshah in Computer Systems Engineering. He is Master of Engineering student in IICT, Mehran UET, Jamshoro, Pakistan. He is interested in the field of Text mining and machine learning

Dr. Sheeraz Memon: He is Associate Professor in the department of Computer Systems Engineering, Mehran University of Engineering and Technology, Jamshoro, Pakistan. He was awarded Ph.D. in Pattern Recognition and speech processing from school of Electrical & Computer Eng. From RMIT University, VIC, Australia. His work is published in impact factor journals and reputable international conferences. He is also reviewer of International Research Journals.

Muhammad Bux Alvi: He is Assistant Professor in the department of Computer Systems Engineering, The Islamia University of Bahawalpur, Bahawalpur, Pakistan. He did Bachelor of Engineering from Quaid e Awam University of Engineering, Science and Technology, Nawabshah, Master of Engineering from MUET, Jamshoro. He is Ph.D. scholar in MUET, Jamshoro. His work is published in International Journals and Conferences. His research area is machine learning, Data Mining, Text analytics and hybrid models for sentiment analysis.