IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-6, June 2019 E-ISSN: 2347-2693

Comparative Study of Machine Learning Algorithms for Document Classification

Rahul Jain[1*], Archana Thakur[2]

[1,2]School of Computer Science and IT, Devi Ahilya University, Indore, India

Corresponding Authors: rahuljaincse51@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i6.11891191 | Available online at: www.ijcseonline.org

Accepted: 21/Jun/2019, Published: 30/Jun/2019

***Abstract*-**Text classification is a task of distribution of collection of predefined classes to free-text. Text classifiers are not able to organize, structure, and reason just about something. In this work we have used random forest and naïve Bayes algorithms to perform document classification task. We have trained the machine learning models to inference the respective class of the documents. By working on very big data sets of movie reviews the chosen machine learning models predict whether the reviews are positive or negative and then we analyse and compare the results of each model's individual confusion matrix like precision, recall, f1-score & support. An important observation is that for the same input data random forest provides more relevant results as compared to naïve bayes algorithm. But as the training data grows naïve bayes also performs equally good as random forest.

***Keywords*-**Text Classification, Naïve Bayes, Random Forest, Machine Learning

I. INTRODUCTION

Machine learning algorithms deal with designing programs that learn from past data and is also a field of artificial intelligence [5, 6, 7, 8, 9, 10, 11]. Machine learning methods are more famous as compared to statistical methods as they do not consider basic data assumption. Text classification is a machine learning problem. Basically text classification is a task of distributing a collection of predefined classes to free-text. Text classifiers won't be able to organize, structure, and reason just about something. For instance, new articles will be organized by topics, support tickets will be organized by urgency, chat conversations will be organized by language, and complete mentions will be organized by sentiments. The goal of text classification is to mechanically classify the text documents into one or additional outlined classes. It is a technique to allocate strings or documents into distinct tier or class by assigning tags according to the content.

Report order is a standout amongst the most conspicuous use of AI. It is utilized to naturally relegate predefine classes (marks) to free content records. By grouping content, we are intending to dole out at least one class or arrange to a report, making it simpler to oversee and sort. This is particularly valuable for distributers, news destinations, websites or any individual who manages a great deal of substance.

Report grouping is a noteworthy learning issue that is at the center of numerous data the board and recovery undertakings. Record arrangement plays out a fundamental job in different applications that manages sorting out, grouping, looking and compactly speaking to a lot of data. Report order is longstanding issue in data recovery which has been all around considered.

Programmed report order can be comprehensively arranged into three classes. These are supervised record arrangement, unsupervised archive characterization, and semi-managed report order. In supervised archive order, some system outside to the characterization model gives data identified with the right record grouping. Consequently, if there should arise an occurrence of supervised report grouping, it turns out to be anything but difficult to test the precision of archive order model. In unsupervised record characterization, no data is given by any outside component at all. If is an occurrence of semi-directed record arrangement parts of the archives are named by an outer instrument.

A portion of the strategies that are utilized for archive arrangement incorporate Expedition boost, Naïve Bayes classifier, Support Vector Machine, Decision Trees, Neural Network, and so forth.

In this work we have used random forest and naïve bayes algorithms to perform the document classification. Let's discuss more about the random forest and naïve bayes before we compare our practical observations on the same set of data.

1) ***Random forest:*** Random forest is extension of decision tree model. Random forest will implement multiple decision

tree models and merge them together to produce better and effective model than a single decision tree. The advantage of using random forest is that it can be used for both regression and classification problems. The parameters that we have used in our document classification model are simple to understand and use, this makes random forest easy to use for such classification problems. As we know that the over fitting is a general problem while using machine learning algorithms where the models are very sensitive to the training data. Random forest can avoid this problem by having enough trees the forest.

2) ***Naïve Bayes:*** Unlike random forest the naïve bayes is more suitable only for the classification problem, it is not recommended to use it for the regression problems. This is an algorithm which is much recommended to perform the text classification. It requires comparatively very less training data than random forest for generating good model. This algorithm is not suited to represent the complex problem, and that is why naïve bayes seldom creates the model with over fitting. This algorithm is having limited options for parameter tuning.

II. RELATED WORK

Sentiment analysis for social websites is an important research area now days [1, 2, 3, 4, 15, 16]. In [10] authors have performed archive characterization on the seven class Yahoo newsgroup informational index. The informational collection contained reports partitioned into following classes: International, Politics, Sports, Business, Entertainment, Health, and Technology. They utilized naïve bayes, decision trees, nearest neighbor classifier and the subspace strategy for characterization. They likewise performed arrangement of utilizing the mix of these calculations. They embraced the usually utilized sack of words record portrayal conspires for highlight portrayal. They disregarded the structure of record and course of action of words in their component portrayal. The component vectors contained all the unmistakable words in the preparation set after evacuation of all the stop words. The stop words are the words that don't help in record order, for example, 'the,' 'and,' 'a few,' 'it,' and so on. They likewise evacuated a portion of the low-recurrence words that happen very sometimes in the preparation set of records. In a general situation, there will be a huge number of highlights (given an enormous volume of archives in your dataset) since there are around 50,000 regularly utilized words in the English language. Given an archive D, its component vector is produced. For making the component vector for each archive, they utilized 2 approaches. The first is the twofold methodology where for each word in vocabulary the estimation of 1 is given if the word exists in the record D or 0 on the off chance that it doesn't. In the second methodology, the recurrence of each word is utilized to shape an element vector. In this paper, the authors have utilized a binary portrayal for naïve bayes and decision trees strategy. Though, they utilized frequency portrayal in nearest neighbor classifier and the subspace technique classifier to compute the heaviness of each term.

III. METHODOLOGY

In this work we have used random forest and naïve bayes algorithms to perform the document classification task. Both are well known machine learning algorithms [5, 6]. We have used these machine learning algorithms in Python. The steps to build a text classification model in Python are:

1) Import libraries
2) Import the dataset
3) Content preprocessing
4) Modifying text to numbers
5) Buildup and test sets
6) Tune-up text classification archetypal and predicting sentiments
7) Estimate the model
8) Saving and readying the model [17]

IV. EXPERIMENTS AND RESULTS

Here let's define the performance parameters we are using for comparing random forest and naïve bayes. In the formulations below T_P represents True Positive, F_P represents False Positive and F_N represents False Negatives.

1) Precision - It gives the ratio of correctly predicted positive observations to the total number predicted observations. It provides the percentage of total relevant results.

$$Precision = \frac{T_P}{T_P + F_P} \quad (1)$$

2) Recall - It is also known as sensitivity of the model. It tells us about the percentage of total relevant results correctly classified by any machine learning algorithm.

$$Recall = \frac{T_P}{T_P + F_N} \quad (2)$$

3) F-Score/F-Measure - It is the weighted average of precision and recall.

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Below are the confusion matrices of both the algorithms.

```
for Random Forest
[[180  28]
 [ 30 162]]
              precision    recall  f1-score   support

           0       0.86      0.87      0.86       208
           1       0.85      0.84      0.85       192

   micro avg       0.85      0.85      0.85       400
   macro avg       0.85      0.85      0.85       400
weighted avg       0.85      0.85      0.85       400

0.855
```

Figure 1: Confusion matrix of random forest algorithm

for GuassianKnaivebayes

	precision	recall	f1-score	support
0	0.76	0.73	0.75	208
1	0.72	0.75	0.73	192
micro avg	0.74	0.74	0.74	400
macro avg	0.74	0.74	0.74	400
weighted avg	0.74	0.74	0.74	400
				0.74

Figure 2: Confusion matrix of naïve bayes algorithm

IV. CONCLUSION

It is clear from the confusion matrices above that random forest provides more relevant results than naïve bayes. It also shows the sensitivity to the outliers. Since the model is trained with less training data there might be a smaller number of ensembles of decision trees causing this model to be more sensitive and hence more accurate for small amount of data. On the other hand, naïve bayes seems to be less accurate because it is less sensitive to the outliers, and is robust to avoid over fitting with even the less amount of training data. Although the overall precision and recall is less as compared to random forest, but as the training data grows the naïve bayes provides good results using [12, 13, 14, 16, 18, 19]. It is more recommended to be used for the document classification**.**

REFERENCES

[1] Agarwal, B. Xie, I. Vovsha, O. Rambow, and R.Passonneau, "Sentiment Analysis of Twitter Data," Annual International Conference New York: Columbia University, 2012.

[2] M.Rambocas, and J. Gama, "Marketing Research: The Role of Sentiment Analysis". The 5th SNA-KDD Workshop'11. University of Porto, 2013.

[3] Andrew Mc Callumzy, and Kamal Nigamy. "A Comparison of Event Models for Naive Bayes Text Classification". Learning for Text Categorization: Papers from the 1998 AAAI Workshop, pp. 41-48.

[4] Zu G., Ohyama W., Wakabayashi T., Kimura F., "Accuracy improvement of automatic text classification based on feature transformation": Proc: the 2003 ACM Symposium on Document Engineering, November 20-22, 2003, pp.118-120

[5] Chaudhary, A., Kolhe, S., Kamal, R., 2016. A hybrid ensemble for classification in multiclass datasets: An application to oilseed disease dataset. Computers and Electronics in Agriculture 124, pp.65–72.

[6] Chaudhary, A., Kolhe, S., Kamal, R., 2016. An improved random forest classifier for multi-class classification. Information Processing in Agriculture 3, pp. 215-222.

[7] Chaudhary, A., Kolhe, S., Kamal, R., 2012. Machine learning techniques for mobile intelligent systems: A study. In IEEE Ninth International Conference on Wireless and Optical Communications Networks (WOCN), pp. 1-55.

[8] Chaudhary, A., Kolhe, S., Kamal, R., 2013. Machine Learning Classification Techniques: A Comparative Study. International Journal on Advanced Computer Theory and Engineering 2(4), pp. 21-25.

[9] Chaudhary, A., Kolhe, S., Kamal, R., 2013. Machine Learning Techniques for Mobile Devices: A Review. International Journal of Engineering Research and Applications 3(6), pp. 913-917.

[10] Chaudhary, A., Kolhe, S., Kamal, R., 2013. Performance Examination of Feature Selection methods with Machine learning classifiers on mobile devices. International Journal of Engineering Research and Applications 3(6), pp.587-594.

[11] Thakur, A., Thakur, R., 2018. Machine Learning Algorithms for Intelligent Mobile Systems. International Journal of Computer Sciences and Engineering 6(6), pp. 1257-1261.

[12] http://www.cs.cornell.edu/people/pabo/movie-review-data/poldata.README.2.0.txt

[13] https://www.anaconda.com/distribution/#download-section

[14] https://stackabuse.com/using-regex-for-text-manipulation-in-python/

[15] A. Pak, and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," Special Issue of International Journal of Computer Application, France: Universitede Paris-Sud, 2010.

[16] Forman, G., 2003. "An Experimental Study of Feature Selection Metrics for Text Categorization". Journal of Machine Learning Research, 3 2003, pp. 1289-1305

[17] https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a

[18] Y.H.LI and A.K Jain "Classification of text document", the computer Journal, vol.41, pp. 8,1998

[19] https://monkeylearn.com/text-classification/

[20] https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/\

Authors Profile

Dr. Archana Thakur received M.Tech and Ph.D. from School of Computer Science & IT, Devi Ahilya University, Indore. She is working as an Assistant Professor at School of Computer Science & IT, Devi Ahilya University, Indore. She has guided many research scholars. She is involved in coordinating graduate-level and postgraduate-level training program in computer science for the university. She has published many research papers in various national and international journals & participated in many conferences. Her research areas include Artificial Intelligence, Machine learning, Data Mining and Soft Computing.

Rahul Jain pursuing M.Tech from School of Computer Science & IT, Devi Ahilya University, Indore. He is Gold Medallist in Bachelor of Engineering in Computer Science Engineering from Jabalpur Engineering College, Jabalpur. He has qualified GATE exam and selected in couple of multi national companies include TCS and Wipro at the time of his Bachelor's degree.

