International Journal of Advanced Research in Arts, Science, Engineering & Management (IJARASEM)

| ISSN: 2395-7852 | www.ijarasem.com | Bimonthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 5, September 2020 |

Text -Classification of 20Newsgroups Data Set using Linear SVC Model

Deep Katira

IT Student, Department of Information Technology , B.K. Birla College of Arts , Science and Commerce (Autonomous), Kalyan, Maharastra, India

ABSTRACT: Machine-Learning is something that have fantasized world on a large scale in dawn of technological age with its Natural Language Processing algorithms. One of its popular research fields is text mining which concerns predictive pattern discovery from enormous documents. In recent decades, to improvise the performance of text classification and text clustering efforts have been invested to enrich text representation. As a result, to search, organize and store the enormous text data, text document classification approaches gained huge importance. In this paper, supervised learning approach towards 20Newsgroups data set with 18,846 newsgroup documents to study text-classification accuracies of most used Naïve Bayes and SVM classifiers .It involves stages like data exploration, feature extraction, training the classifier, parameter scaling and evaluation of trained model. Experimental analysis tell us that classification accuracy of SVM classifier was better than Naïve Bayes Classifier and it increased further more by feature extraction and parameter scaling.

KEYWORDS: Naïve Bayes classifier, Linear SVC classifier, feature extraction, text classification.

I.INTRODUCTION

Millions or even billions of users are being served due to increasingly centralization of computer and web services and evaluation of applications. Therefore, daily increase in online data leads to need that the data generated should be organized and regularized. Organising text by topic is often done using text classification. This is mostly used in reviews, emails, articles, etc. We generally train a classifier to tag what texts are about, provided we specify associations between tag and text so that machine learning models can learn on their own. Generating automatic text classifier can label natural language text with relevant pre-defined categories and also organize text which can speed-up processing time. A major problem of text categorization is most of features (i.e. terms) are not irrelevant for classification and redundant. In recent time many text classifier came into existence for text classification and assured many promising directions in field of text mining.

SDGC, SVM, Naïve Bayes, Perceptron, Decision Tree, Logistic Regression, k-Nearest Neighbor are some of often used classifiers. From this in proposed paper, implementation and comparative study of Multinomial Naïve Bayes classifier which is "probabilistic classifier" based on applying Bayes' theorem and Linear SVC which aims to fit the data provide and returns a best fit hyper plane that categorizes data was done on 20Newsgroups data set with 18,846 newsgroup documents and 20 categories .Our goal was to study which classifier out of two mentioned above classified each document in 20Newsgroups data set more efficiently and in more accurate manner. Each document in dataset was text written in English with lot of punctuations. Use of popular approach TF-IDF is used for feature extraction.

The further sections of paper are summarized as follows: section II consists of related work, section III consists of methodology of study, section IV of experimental results of study and section V concludes our study with future scope.

II. RELATED WORK

In[1] Kim et.al. along with weight-enhancing method proposed Poisson naïve Bayes classification model which cost little extra time and space but was useful in building probabilistic classifier and believed that it can be implemented in practical systems applications as in spam-filtering and news-alert system due to its efficiency, simplicity and iterative learning. In[2]Wen et.al. studied the impact of multi-word for text representation on text classification where they extracted multi-words from documents based on syntactical structure and second strategy included two things based on semantic level to categorize multi-words based on topics of general concepts and on subtopics of same concepts

| ISSN: 2395-7852 | www.ijarasem.com | Bimonthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 5, September 2020 |

and classification series was done with help of SVM linear and non-linear kernels on Reuters dataset. In [13] Kyrre et.al. demonstrated usage of high assurance data guards for controlling the information flow between security domains by introducing the concept of applying machine learning techniques to construct automated, data-driven content checkers. They significantly yield in performance by presenting concept of controlled environments. In[14] Armand et.al. proposed a FastText model which simple baseline for text classification. They used n-grams features and then compared the trained classifier to several classifiers to determine speed of classification and then evaluated model to scale space on a large tag prediction dataset. In[4] Amani et.al. proposed a depth study on various machine learning techniques for data classification like SVM, ASVM, KNN and NB to make learn the models to manage energy consumption and generation because renewable energy resources are unpredictable on daily basis .The paper aims to integrate all models so efficiency of powergrid to generate from renewable sources can be increased by doing data classification.

III.METHODOLOGY

1) Datasets included:

- 20Newsgroups :

In proposed study 20Newsgroups data was used which was imported from sci-kit learn using fetch_20newsgroups (). The dataset consists of 18,846 newsgroup documents and 20 categories whichwere originally collected by Ken Lang for his Newsweeder: Learning to filter netnews paper. The 20Newsgroups collection is famous dataset for text applications like text classification and clustering in Machine Learning.

2) Proposed Model:

As shown in Fig.1 requirement of annotated dataset 20newsgroup collection was imported to train and test classifier.Then a general train function was defined because when making first model it may result in less accuracy so best practice is “trial and error method”. Then feature extraction process was done using Scikit-learn build-in function *TfidfVectorizer* which calculate frequency of words in a document and reduce the weight of most common words because you have extract features in order train model because algorithms can understand only numerical feature vectors.The collection was split into train and test size in ratio 8:2. Then training set data was trained using Multinomial Naïve Bayes and Linear SVC classifiers. Pipeline was created for both classifiers.Then parameter scaling was done to achieve accuracy results which are discussed in experimental results.

- Parameter scaling

For Multinomial Naïve Bayes classifier first model was trained using hyper-parameter alpha =0.1 which is default one .Further we used stop words and kept alpha value default one. Then we changed alpha parameter values and calculated accuracy of best parameter. Then *min_dif*parameter was set to 5 which ignored the words that appeared less than five times in all document and accuracy was noted. Next, we tried to stem the data using parameter *tokenizer* within *TfidfVectorizer* and also added punctuation to list of stop words. Same process was followed for Linear SVC where hyper-parameter was C=0.1(default) and in later process parameter was changed to achieve greater accuracy. Stemming and tokenizing process, stopwords was all done in same manner as was done for naïve Bayes classifier.

- Model Evaluation:

Test dataset was loaded in both trained classifier models with best hyper-parameter tuningand confusion matrix was plotted for predicting result as referred in Fig.1.

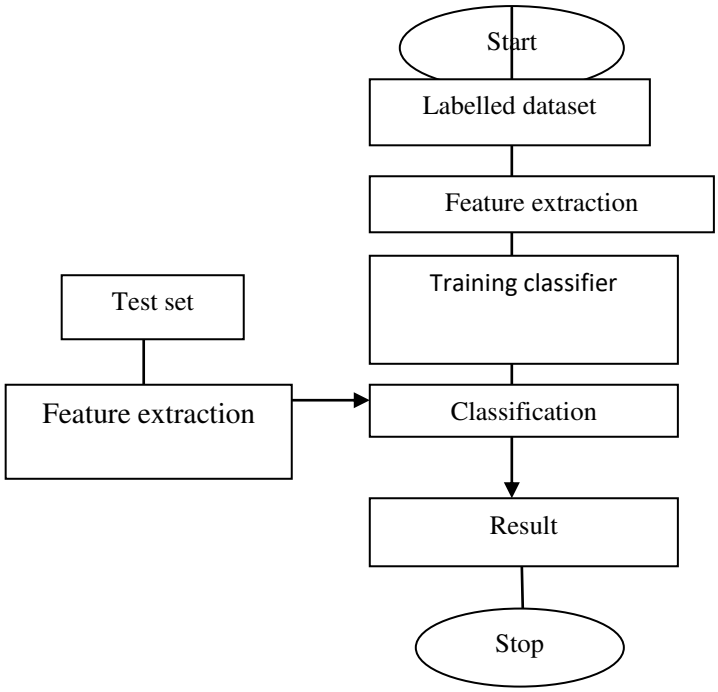

Fig.1. Flow chart of the proposed model

IV.EXPERIMENTAL RESULTS

18,846 newsgroup documents were split into train –test split of 11,314 train and 7532 test documents.Using parameter scaling we trained classifiers with best tuned hyper-parameters and test the test dataset using the final selected hyper-parameters .The accuracies of parameter scaling are shown in Table.1. The accuracy of naïve Bayes classifier increased by parameter scaling but was less than Linear SVC classifier. The final hyper-parameter decided for naïve Bayes classifier was alpha=0.005 and for Linear SVC classifier the it was C=10 and based on these, confusion matrix was plot for both the classifiers as shown in Fig.2 and Fig.3 . The accuracy was summarized with multinomial Naïve Bayes classifier was 0.917 and that of Linear SVC was 0.93. The precision and F1-score for Linear SVC were 0.94and 0.93 respectively and that of multinomial Naïve Bayes classifier were 0.92 respectively. ROC and AUC curve, PR and AUC curve for both the models is shown in Fig.4, Fig.5, Fig.6 and Fig.7 respectively.

Parameter Scaling	**Linear SVC classifier**		**Naïve Bayes classifier**	
	Hyper-parameter	**Accuracy**	**Hyper-parameter**	**Accuracy**
1)Using stopwords	C=1.0	0.93	Alpha=0.1	0.88
2)setting min_dif =5	C=10	0.93	Alpha=0.005	0.91
3)By word tokenizing and stemming of data	C=10	0.93	Alpha=0.0005	0.91

Table 1.Parameter scaling accuracies.

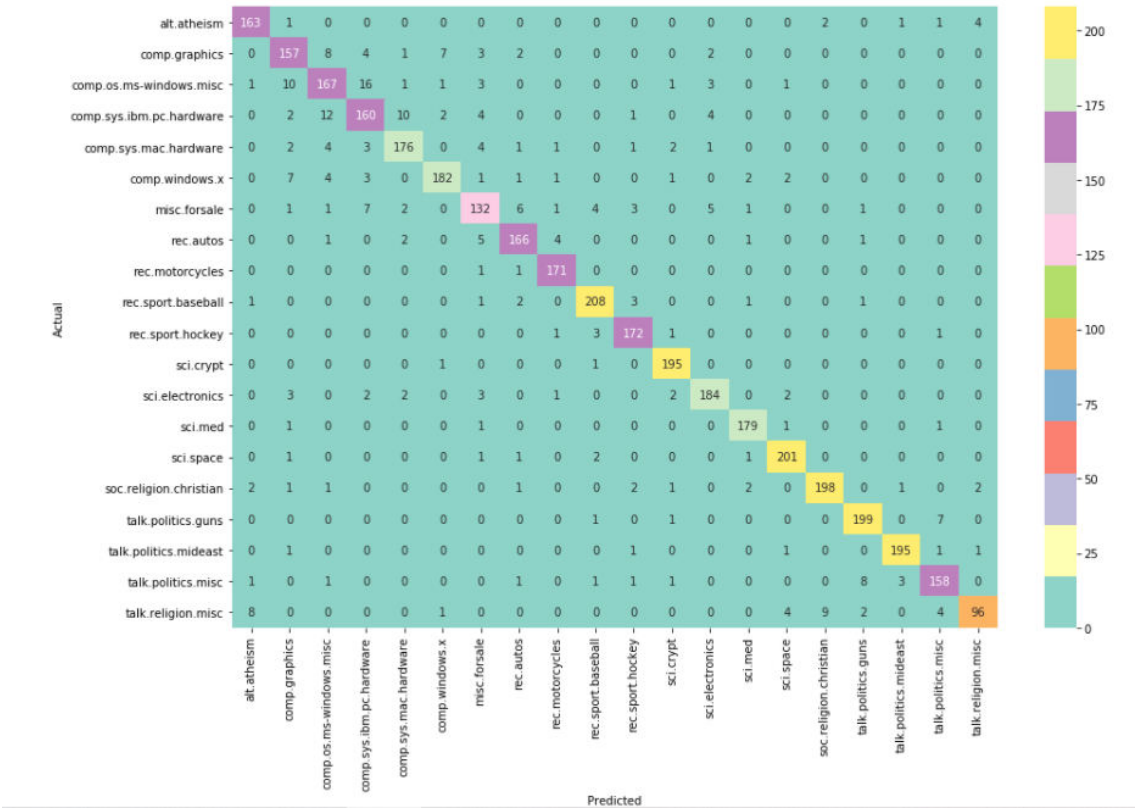

Fig.2 Confusion matrix for Naïve Bayes classifier.

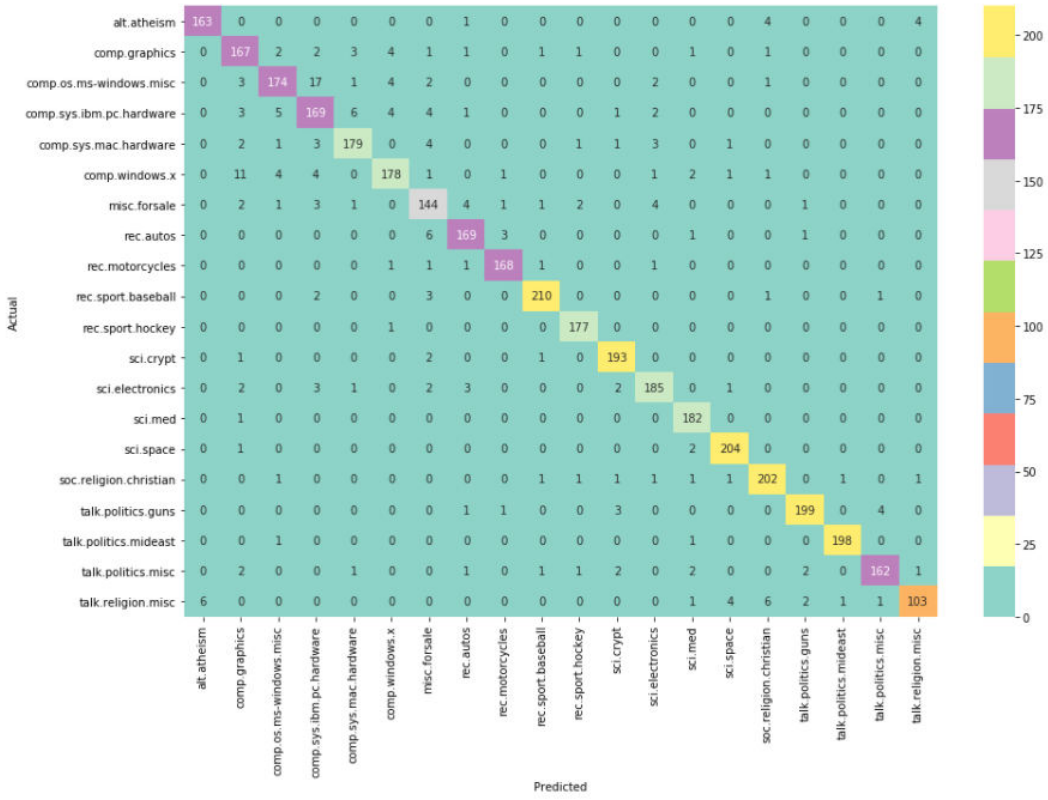

Fig.3 Confusion matrix for Linear SVC classifier.

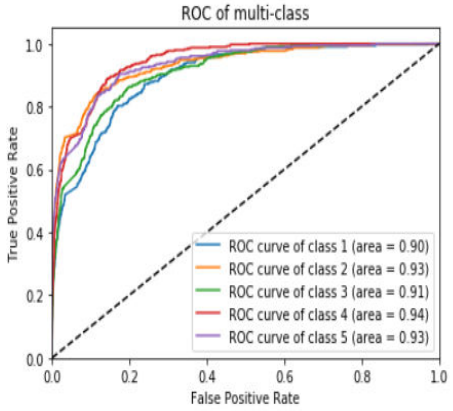

Fig.4 ROC and AUC curve for Naïve Bayes classifier

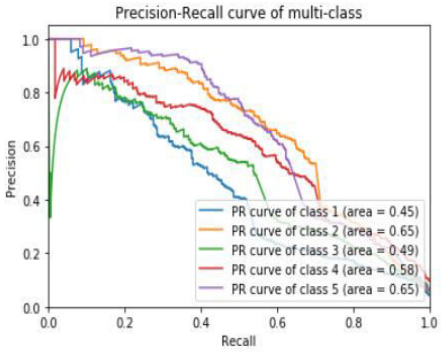

Fig.5 PR and AUC curve for Naïve Bayes classifier

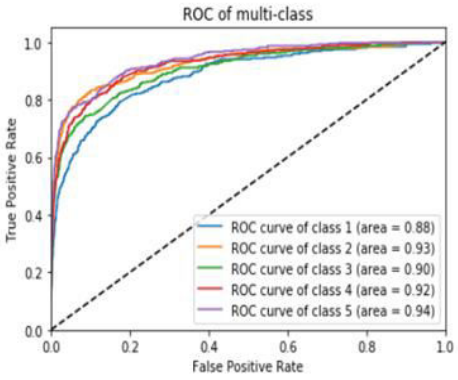

Fig.4 ROC and AUC curve for Linear SVC classifier

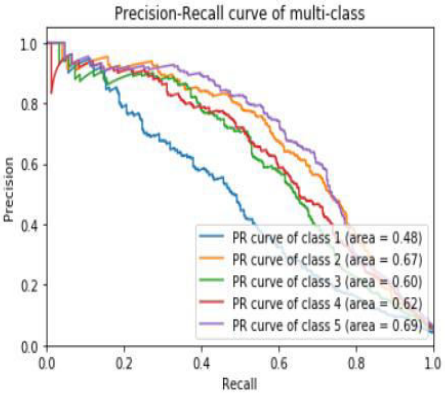

Fig.5 PR and AUC curve for Linear SVC classifier

V.CONCLUSION

In our study, we compared the accuracy of two well-known and frequently use classifiers .Feature vectors were extracted through *TfidfVectorizer* .We experimented with both trained classifiers by parameter scaling and model evaluation.It can be concluded that Linear SVC model showed a higher accuracy and precision in classifying the text into defined categories. Naïve Bayes also predicted with good accuracies and can achieve further more by more text-cleaning and parameter scaling.In future ,Linear SVC model can be trained with more efficient hyper-parameter tuning and text pre-processing.

VI. ACKNOWLEDGEMENT

I would like to thank Prof.Swapna Augustine Nikale,Department of Information Technology,B.K Birla College Kalyan for guiding throughout the research work.

VII. GLOSSARY

SVC- Support Vector Classifier

TF-IDF – Term Frequency –Inverse Document Frequency

| ISSN: 2395-7852 | www.ijarasem.com | Bimonthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 5, September 2020 |

REFERENCES

[1] Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, & Sung Hyon Myaeng. (2006). Some Effective Techniques for Naive Bayes Text Classification. *IEEE Transactions on Knowledge and Data Engineering*, *18*(11), 1457–1466. https://doi.org/10.1109/tkde.2006.180

[2] Zhang, W., Yoshida, T., & Tang, X. (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems*, *21*(8), 879–886. https://doi.org/10.1016/j.knosys.2008.03.044

[3] Wan, C. H., Lee, L. H., Rajkumar, R., & Isa, D. (2012). A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. *Expert Systems with Applications*, *39*(15), 11880–11888. https://doi.org/10.1016/j.eswa.2012.02.068

[4] Yahyaoui's, A., Yahyaoui, I., & Yumuşak, N. (2018). Machine Learning Techniques for Data Classification. *Advances in Renewable Energies and Power Technologies*, 441–450. https://doi.org/10.1016/b978-0-12-813185-5.00009-7

[5] Kurtanovic, Z., & Maalej, W. (2017). Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 490–496. https://doi.org/10.1109/re.2017.82

[6] Casamayor, A., Godoy, D., & Campo, M. (2010). Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. *Information and Software Technology*, *52*(4), 436–445. https://doi.org/10.1016/j.infsof.2009.10.010

[7] Khalid, F., Hanif, M. A., Rehman, S., Qadir, J., & Shafique, M. (2019). FAdeML: Understanding the Impact of Pre-Processing Noise Filtering on Adversarial Machine Learning. *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 902–907. https://doi.org/10.23919/date.2019.8715141

[8] Lu, Y., Huang, X., Ma, Y., & Ma, M. (2018). A Weighted Context Graph Model for Fast Data Leak Detection. *2018 IEEE International Conference on Communications (ICC)*, 1–6. https://doi.org/10.1109/icc.2018.8422280

[9] Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., & Li, B. (2018). Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. *2018 IEEE Symposium on Security and Privacy (SP)*, 19–35. https://doi.org/10.1109/sp.2018.00057

[10] Alneyadi, S., Sithirasenan, E., & Muthukkumarasamy, V. (2015). Detecting Data Semantic: A Data Leakage Prevention Approach. *2015 IEEE Trustcom/BigDataSE/ISPA*, 910–917. https://doi.org/10.1109/trustcom.2015.464

[11] Huang, X., Lu, Y., Li, D., & Ma, M. (2018). A Novel Mechanism for Fast Detection of Transformed Data Leakage. *IEEE Access*, *6*, 35926–35936. https://doi.org/10.1109/access.2018.2851228

[12] Alneyadi, S., Sithirasenan, E., & Muthukkumarasamy, V. (2016). A survey on data leakage prevention systems. *Journal of Network and Computer Applications*, *62*, 137–152. https://doi.org/10.1016/j.jnca.2016.01.008

[13] Kongsgard, K. W., Nordbotten, N. A., Mancini, F., Haakseth, R., & Engelstad, P. E. (2017). Data Leakage Prevention for Secure Cross-Domain Information Exchange. *IEEE Communications Magazine*, *55*(10), 37–43. https://doi.org/10.1109/mcom.2017.1700235

[14]Bag of Tricks for Efficient Text Classification. (2017). *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, *2*, 427–431. https://www.aclweb.org/anthology/E17-2068

[15] Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications*, *36*(3), 5432–5435. https://doi.org/10.1016/j.eswa.2008.06.054

[16] Sun, A., Lim, E.-P., & Liu, Y. (2009). On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems*, *48*(1), 191–201. https://doi.org/10.1016/j.dss.2009.07.011

[17] Dilrukshi, I., De Zoysa, K., & Caldera, A. (2013). Twitter news classification using SVM. *2013 8th International Conference on Computer Science & Education*, 287–291. https://doi.org/10.1109/iccse.2013.6553926

[18] Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and Word2vec for text classification with semantic features. *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, 136–140. https://doi.org/10.1109/icci-cc.2015.7259377

[19] Yeom, S., Giacomelli, I., Fredrikson, M., & Jha, S. (2018). Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 268–282. https://doi.org/10.1109/csf.2018.00027

[20] Jiang, L., Cai, Z., Zhang, H., & Wang, D. (2013). Naive Bayes text classifiers: a locally weighted learning approach. *Journal of Experimental & Theoretical Artificial Intelligence*, *25*(2), 273–286. https://doi.org/10.1080/0952813x.2012.721010