Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 7, July 2021: 8236 - 8248

Research Article

A Comparative Analysis of Classification Algorithms in Authorship Attribution

Bommideni Revathi[a], Srinivasu Badugu[b]

[a]Assistant Professor Stanley College of Engineering and Technology for Women
[b]Professor Stanley College of Engineering and Technology for Women

***Corresponding author:** [a]revathibommideni@stanley.edu.in, [b]srinivasucse@gmail.com

Abstract

Authorship attribution, the role of identifying the author of a text, has been limited to works of historical importance, but today it is still of great significance. The primary objective of this paper is to lay down the rules for characteristic extraction strategies. Feature extraction and implementation techniques with various classifiers in simple ways so that a move to the attribution of authorship can also operate. With the help of count vectors and term-frequency inverse document frequency(TF-IDF), we presented this paper using three supervised machine learning algorithms such as support vector machine, multinomial naive bayes, and logistic regression. We used the Sklearn library for implementation. The dataset of 3 authors consists of 19579 instances. We split 70 percent of the training dataset, which is 13705 instances, and 30 percent of the test dataset, which is 5874 instances randomly picked and split from the initial dataset. In the Naive Bayes classifier, we have the highest accuracy of 82.09 percent using 24823 vector (vocabulary) size

Keywords: Machine Learning, Authorship attribution, SVM, Naive Bayes, Logistic Regression, NLP, Vectorization

1. Introduction

Authorship attribution, the role of identifying the author of a text, has been limited to works of historical importance, but today it is still of great significance. The classification of conventional authors attempts to use the whole corpus of the published work of the author as training data. Such works tend to be lengthy, including thousands of sample sentences that are typically identical in ideological content and structure as part of a structured work. Along with online forums and blog pages, the widespread use of social media networks such as Facebook and Twitter means that there is an immense amount of information available on how people write. Furthermore, The differences between social media posts and conventional types of writing, such as books, newspaper articles, and academic papers, make the issue of author recognition for the general public difficult. One important difference is that the amount of sentences available from social media posts is far lower for a average individual than what can be obtained from the life production of a skilled author. In addition , social media posts tend to be more casual, succinct (in some cases subject to a word limit), and convey a variety of diverse ideas in contrast to traditional published works, rather than supporting one coherent train of thought. The attribution of authorship from a wider contextual viewpoint is often part of Forensic Linguistics Research.

In this research, the primary focus is on using different steps to identify the author of a given text. Step 1 separates the label and text from dataset, step 2 pre-processes text using the library of the natural language tool kit (NLTK), step 3 constructs vectors using count-vector and TF-IDF vectors, step 4 classifies text, and finally identifies the author of the text.

2. Related Works

Houvard et. al [1] approached authorship identification via n-gram[2,3] feature selection. Zhang et . al. [4] proposed a semantic association model for dependency relation between words and unsupervised approach to identify authorship of unstructured texts. Asir et al.[5], through multiple kernel learning,approached Authorship Attribution as semi-supervised anomaly detection. Luke et. al. [6] proposed models of authorship attribution that use natural language processing techniques[7] to classify the author of Twitter messages to derive lexical, syntactic , and semantic features that are used as inputs to multi-class classifiers of Naive Bayes[8,9,10], SVM[11], and neural network.

3. Methodology

Based on features and classification algorithms, we propose a method that will help classify the author of the text document. We perform some preprocessing tasks such as corpus cleaning, stop word elimination, suffix stripping using stemming for extracting features. Utilizing count and TF-IDF vectorization, we generate document vectors-based vocabulary size after preprocessing. For the identification of authorship attribution, we used three classification algorithms such as logistic regression, multi-nominal naive bayes, support vector machine(SVM).

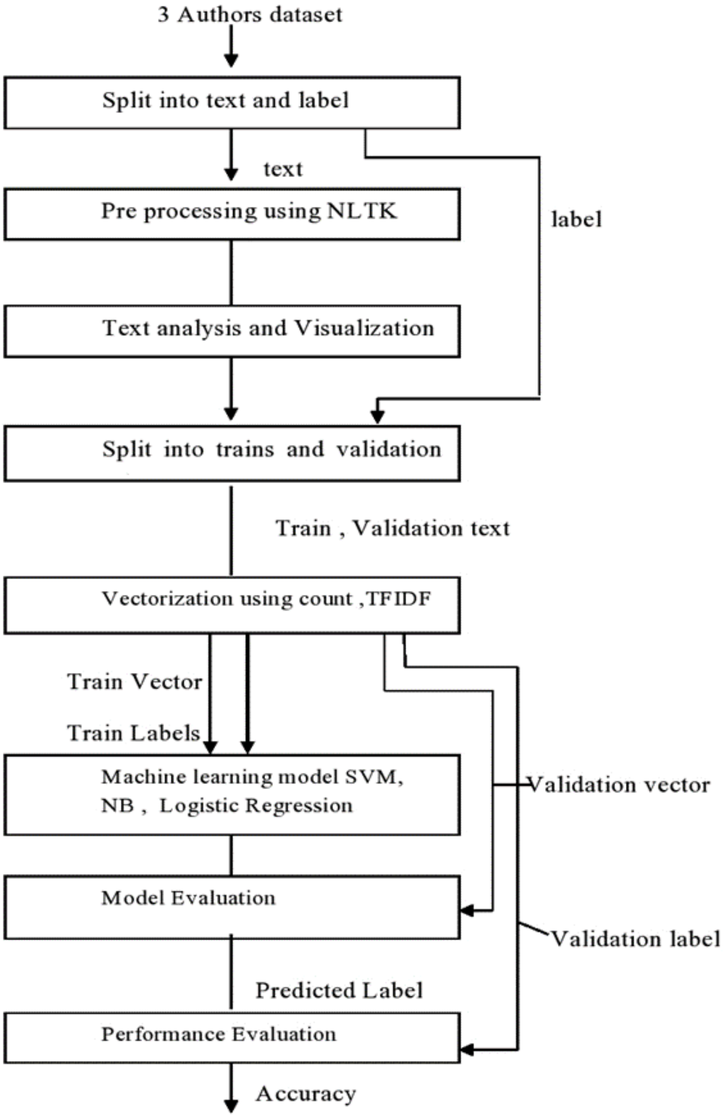
3 Authors dataset
Split into text and label
text
label
Pre processing using NLTK
Text analysis and Visualization
Split into trains and validation
Train , Validation text
Vectorization using count ,TFIDF
Train Vector
Train Labels
Machine learning model SVM, NB , Logistic Regression
Validation vector
Model Evaluation
Validation label
Predicted Label
Performance Evaluation
Accuracy

Figure 1. System Flow Diagram

3.1. Corpus

The dataset that we have supplied consists of 3 authors, 19579 instances where each instance is a text document of 10 to 100 words. The works of Edgar Allan Poe (EAP), HP Lovecraft (HPL) and Mary Shelley (MWS) in sentences from this dataset. EAP consists of 7900 cases, 5635 consists of HPL and 6044 is composed of MWS.

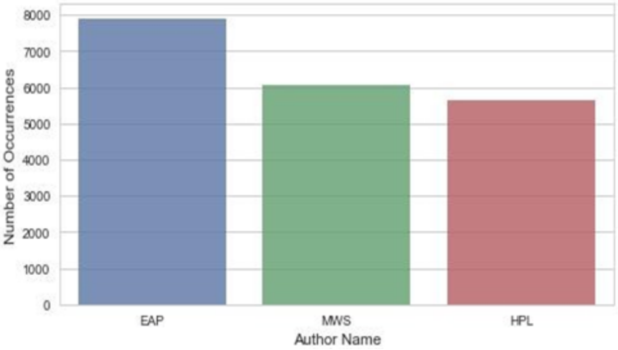

Figure 2. Distribution of 3-Authors in Data Set

The distribution of authors in the text data is shown in Figure 2. There is a slight difference between HPL and MWS whereas EAP is dominating the other two authors.

3.2 Text extraction from corpus

Using the pandas kit, we separated text and corresponding author type. All text stored in one data frame and all class values stored in another data frame are easy to process in the next steps.

3.3 Pre-Processing

A very significant phase in the attribution of authorship is text pre-processing. Text documents are not in the required form for learning in their original form. They must be translated into an input format that is acceptable. It can be transformed to a vector space since the representation of the attribute value is used by most learning algorithms. In determining the quality of the next stages, that is, the extraction and classification stage of the feature, this step is crucial.

Tokenization: The first step in text analytics is tokenization. The process of breaking down a text paragraph into smaller bits, such as words or phrases, is called tokenization.

Procedure for tokenization:

```
Input: D  Dataset/corpus
output: T  set of tokens

For $d_i$  in D do
  tk ← split $d_i$ using space is a delimiter
  T ← tk
end-for
return T
```

Stop Words: A stop word is a widely used word (such as "the", "a", "an", "in") designed to be overlooked by a search engine, both when indexing search entries and when retrieving them as a result of a search query. We would not want these terms to take up space in our database, or to take up precious processing time.

Procedure for stop word removal:

```
Input: T  list of tokens
       S set of stop words
```

```
output: NT  set of tokens without stop words

For $t_i$ in T do
  for tk in $t_i$ do
     if tk not in S then
         nt ← tk
    endif
 endfor
     NT ← nt
end-for
return NT
```

Table 1 : Stylometric Feature Distributions

Features	Authors		
	EAP	HPL	MWS
Number Of Punctuation's	4.10	3.21	3.83
Number Of Title Case	2.10	2.33	2.12
Upper Case Words	0.55	0.50	0.75
Average Words Length	4.64	4.63	4.60
Number Of Stop words	12.62	12.94	13.74

Stemming: Stemming is the process of removing affixes (prefixes and suffixes) from features i.e., the process derived for reducing inflected words to their stem.

Step1: Gets rid of plurals and –ed or –ing suffixes.

Step2: Turns terminal y to i when there is another vowel in the stem.

Step3: Maps double suffixes to single ones: -ization-ational, etc.

Step4: Deals with suffixes, -full, -ness etc.

Step5: takes off –ant, -ence etc

Step6: Removes a final –e

Vectorization: Vectorization is the process of converting text into vectors. The process of converting Natural Language Processing text of any order into numbers is called Vectorization in Machine Learning. The result is a matrix knows as word count or Vector matrix. The conversion is important as Machine Learning algorithms take only numerical as the input.

In this research we used two vectorization approaches. Count vectorization and term-frequency inverse document frequency(TF-IDF) vectorization with different vocabulary size. Line 3000, 5000, etc..

Data set Splitting: The dataset of 3 authors consists of 19579 instances in order to perform logistic regression, Naive Bayes, and SVM algorithms to our data Splitting 70:30 technique. In this dataset, we split 70 percent of the training dataset, which is 13705 instances, and 30 percent of the test dataset, which is 5874 instances randomly picked and split from the initial dataset.

4. Classification Model

We explain in detail how we conducted our experiments in this section, how we constructed the optimal classifier and how we evaluated them. We used the Sklearn library [12,13] to create a model for machine learning. Using a classifier function Object such as LogisticRegression for logistic regression, naive bayes. MultinomialNB for naive bayes and svm. SVC for support vector machines, we establish a classifier. Using the fit method to train the algorithm and evaluate the algorithm using the method of predict. We used a linear kernel on the SVM.

5. Result Analysis

The models were trained on 13705 cases, and checked on the other 5874, to evaluate the performance of the classification model using different vectorization. We used various vocabulary sizes, such as 5000, 10000 and 15363. Table 2 shows statics of actual dataset, training set and test dataset.

Table 2 statics of actual dataset, training set and test dataset.

3_author Dataset	EAP	HPL	MWS	Total
Actual dataset	7900(40%)	5635(29%)	6044(31%)	19579(100%)
Training set	5529 (40%)	4210(29%)	3966(31%)	13705(100%)
Test set	2371(40%)	1834(29%)	1669(31%)	5874

We trained the three classification models using 5000 vocabulary size. After removed punctuation marks and stop words we selected vocabulary of 5000 out of 24823. blow tables show the confusion matrix and performance of three classification algorithms. We assigned label values to class values like EAP is 0 , HPL is 1 and MWS is 2.

For Logistic Regression, Naive Bayes and Support Vector Machines, we obtained accuracy 79.24, 79.80 and 78.49 by using counter vectorization with a vector size of 5000.

Table 3 Confusion matrix for Three Classifier Using Count Vectorization without stopwords

	Logistic Regression			Naive Bayes			SVM		
	EAP	HPL	MWS	EAP	HPL	MWS	EAP	HPL	MWS
EAP	1996	183	199	2015	147	216	1971	191	216
HPL	285	1239	118	311	1219	112	295	1238	109
MWS	326	108	1420	306	94	1454	346	106	1402

Table 4 performance of Three Classifier Using Count Vectorization without stop words

	Logistic Regression			Naive Bayes			SVM		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
EAP	77	84	80	77	85	80	75	83	79
HPL	81	75	78	83	74	79	81	75	78
MWS	82	77	79	82	78	80	81	76	78

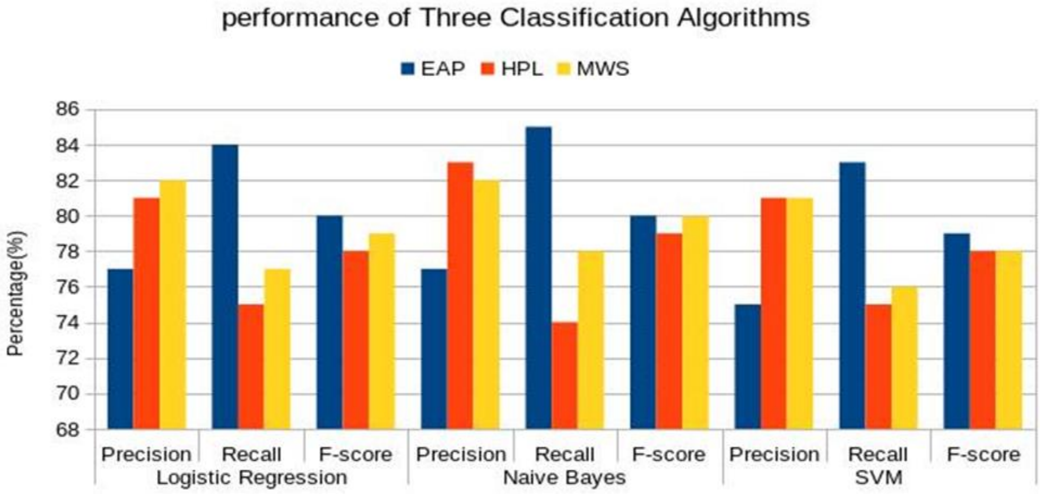

Figure 3. performance of Three Classifier Using Count Vectorization without stop words

We trained and tested the three classification models with help of counter vectorization using 24823 vector size. After removed punctuation marks and stop words we selected vocabulary of 24823 out of 24823. blow tables show the confusion matrix and performance of three classification algorithms. For Logistic Regression, Naive Bayes and Support Vector Machines, we obtained accuracy 81.12, 82.09 and 81.20 by using counter vectorization with a vector size of 24823

Table 5 Confusion matrix for Three Classifier Using Count Vectorization without stopwords

	Logistic Regression			Naive Bayes			SVM		
	EAP	HPL	MWS	EAP	HPL	MWS	EAP	HPL	MWS
EAP	2088	128	184	2153	61	186	2056	152	192
HPL	319	1263	93	368	1193	114	297	1286	92
MWS	295	90	1414	281	42	1476	285	86	1428

Table 6 performance of Three Classifier Using Count Vectorization without stop words

	Logistic Regression			Naive Bayes			SVM		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
EAP	77	87	82	77	90	83	78	86	82
HPL	85	75	80	92	71	80	84	77	80
MWS	84	79	81	83	82	83	83	79	81

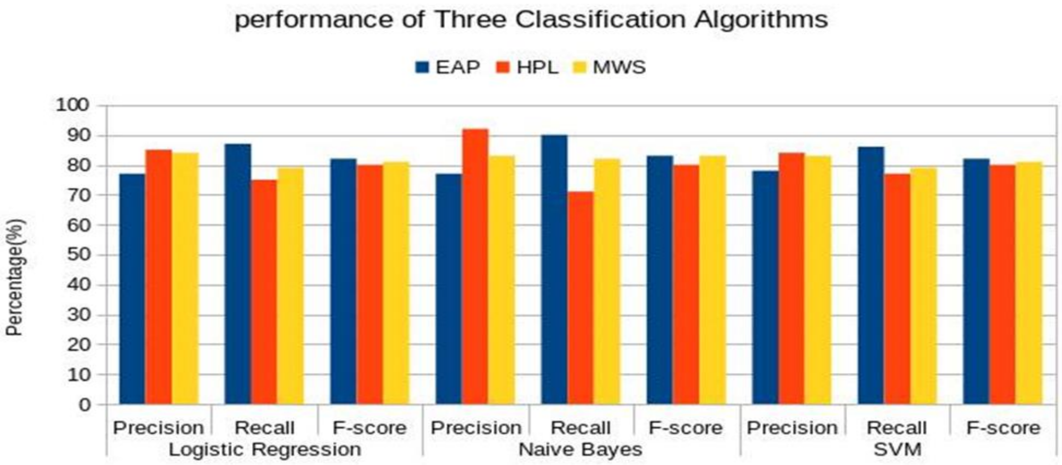

Figure 4. performance of Three Classifier Using Count Vectorization without stop words

We trained the three classification models using 5000 vocabulary size. After removed punctuation marks, stop words and stemming. we selected vocabulary of 5000 out of 15363. blow tables show the confusion matrix and performance of three classification algorithms. For Logistic Regression, Naive Bayes and Support Vector Machines, we obtained accuracy 78.20, 80.06 and 74.49 by using counter vectorization with a vector size of 5000.

Table 7 Confusion matrix for Three Classifier Using Count Vectorization with stemming

	Logistic Regression			Naive Bayes			SVM		
	EAP	HPL	MWS	EAP	HPL	MWS	EAP	HPL	MWS
EAP	1976	182	216	1881	202	291	1928	197	249
HPL	317	1274	121	236	1364	112	405	1177	130
MWS	333	111	1344	220	110	1458	401	116	1271

Table 8 performance of Three Classifier Using Count Vectorization with stemming

	Logistic Regression			Naive Bayes			SVM		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
EAP	75	83	79	80	79	80	71	81	75
HPL	81	74	78	81	80	81	79	69	74
MWS	80	75	77	78	82	80	77	71	74

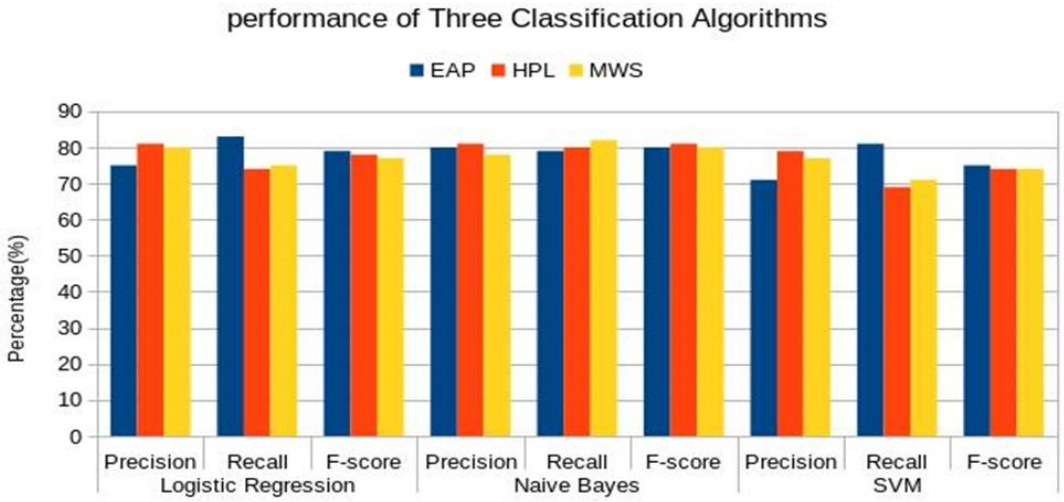

Figure 5. performance of Three Classifier Using Count Vectorization with Stemming

We trained the three classification models using 15363 vocabulary size. After removed punctuation marks, stop words and stemming. we selected vocabulary of 15363 out of 15363. blow tables show the confusion matrix and performance of three classification algorithms. For Logistic Regression, Naive Bayes and Support Vector Machines, we obtained accuracy 79.74, 82.75 and 76.60 by using counter vectorization with a vector size of 15363.

Table 9 Confusion matrix for Three Classifier Using Count Vectorization with stemming

	Logistic Regression			Naive Bayes			SVM		
	EAP	HPL	MWS	EAP	HPL	MWS	EAP	HPL	MWS
EAP	2000	128	205	1917	144	272	1915	154	264
HPL	299	1295	125	190	1402	127	334	1233	152
MWS	330	103	1389	196	84	1542	364	106	1352

Table 10 performance of Three Classifier Using Count Vectorization with stemming

	Logistic Regression			Naive Bayes			SVM		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
EAP	76	86	81	83	82	83	73	82	77
HPL	85	75	80	86	82	84	83	72	77
MWS	81	76	78	79	85	82	76	74	75

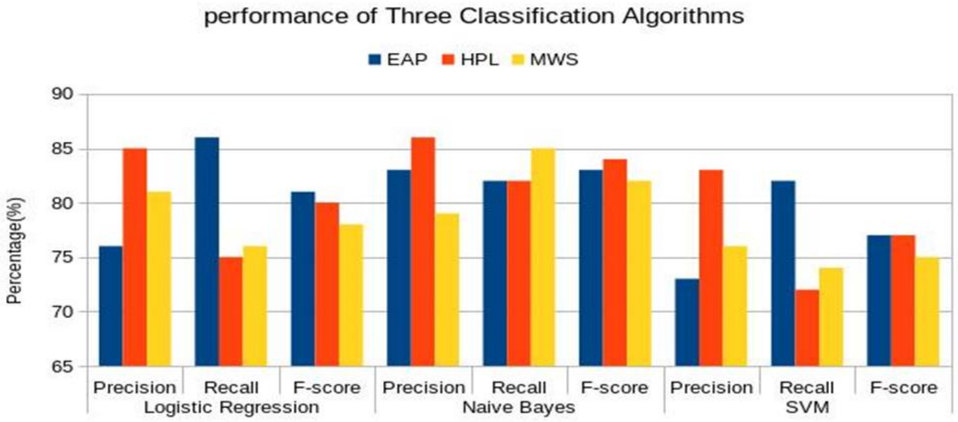

Figure 6. performance of Three Classifier Using Count Vectorization with stemming

We trained and tested the three classification models with help of TF_IDF vectorization using 5000 vector size. After removed punctuation marks and stop words. We selected vocabulary of 5000 out of 24823. blow tables show the confusion matrix and performance of three classification algorithms. For Logistic Regression, Naive Bayes and Support Vector Machines, we obtained accuracy 79.21, 79.97 and 78.26 by using counter vectorization with a vector size of 5000.

Table 11 Confusion matrix for Classifier Using TF-IDF Vectorization without stop words

	Logistic Regression			Naive Bayes			SVM		
	EAP	HPL	MWS	EAP	HPL	MWS	EAP	HPL	MWS
EAP	1986	172	194	2009	147	196	1968	187	197
HPL	283	1252	110	305	1232	108	294	1240	111
MWS	340	122	1415	333	87	1457	357	131	1389

Table 12 performance of Three Classifier Using TF-IDF Vectorization without stop words

	Logistic Regression			Naive Bayes			SVM		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
EAP	76	84	80	76	85	80	75	84	79
HPL	81	76	78	84	75	79	80	75	77
MWS	82	75	79	83	78	80	82	74	78

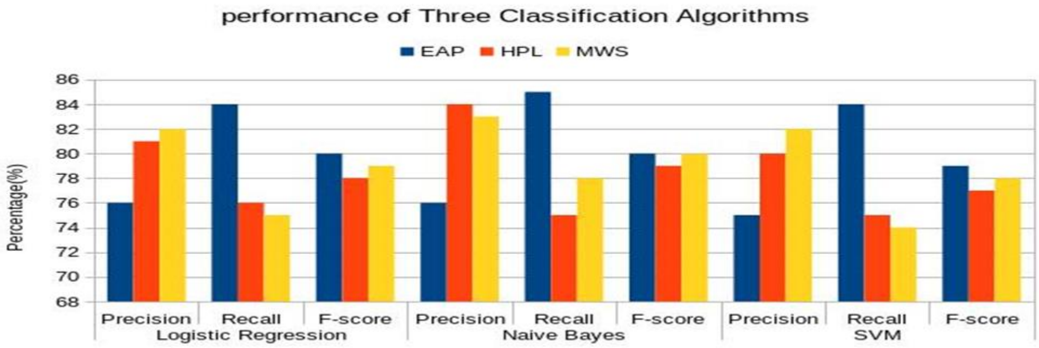

Figure 7. performance of Three Classifier Using TFIDF Vectorization with stop words

We trained and tested the three classification models with help of TF_IDF vectorization using 24823 vector size. After removed punctuation marks and stop words. We selected vocabulary of 24823 out of 24823. blow tables show the confusion matrix and performance of three classification algorithms. For Logistic Regression, Naive Bayes and Support Vector Machines, we obtained accuracy 79.50, 79.58 and 79.65 by using counter vectorization with a vector size of 24823.

Table 13 Confusion matrix for Classifier Using TF-IDF Vectorization without stop words

	Logistic Regression			Naive Bayes			SVM		
	EAP	HPL	MWS	EAP	HPL	MWS	EAP	HPL	MWS
EAP	1997	143	169	2065	71	173	1959	166	184
HPL	358	1285	89	454	1162	116	336	1299	97
MWS	330	115	1388	343	42	1448	311	101	1421

Table 14 performance of Three Classifier Using TF-IDF Vectorization without stop words

	Logistic Regression			Naive Bayes			SVM		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
EAP	74	86	80	72	89	80	75	85	80
HPL	83	74	78	91	67	77	83	75	79
MWS	84	76	80	83	79	81	83	78	80

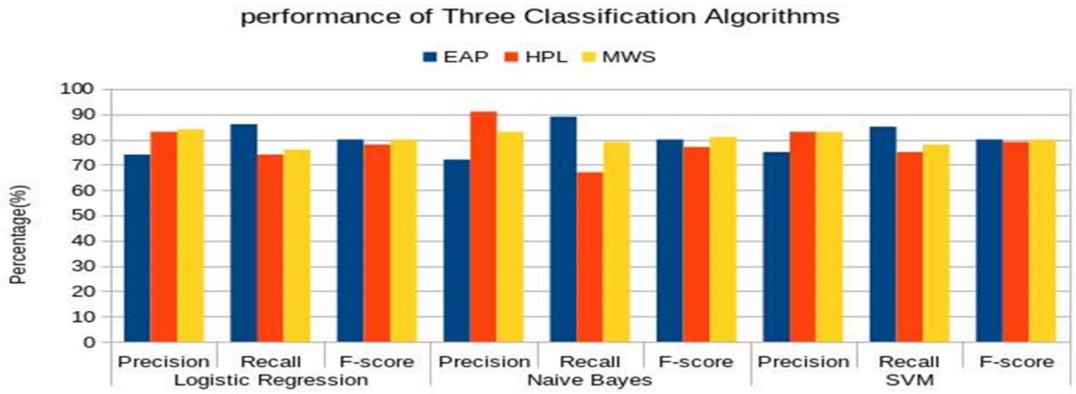

Figure 8. performance of Three Classifier Using TFIDF Vectorization with stop words

We trained and tested the three classification models with help of TF_IDF vectorization using 5000 vector size. After removed punctuation marks, stop words and stem. We selected vocabulary of 5000 out of 15363. blow tables show the confusion matrix and performance of three classification algorithms. For Logistic Regression, Naive Bayes and Support Vector Machines, we obtained accuracy 80.23, 80.88 and 78.99 by using counter vectorization with a vector size of 5000.

Table 15 Confusion matrix for Three Classifier Using TF-IDF Vectorization with stemming

	Logistic Regression			Naive Bayes			SVM		
	EAP	HPL	MWS	EAP	HPL	MWS	EAP	HPL	MWS
EAP	1996	170	178	2017	126	201	1966	195	183
HPL	281	1324	101	320	1274	112	305	1288	113

MWS	313	113	1398	279	85	1460	325	113	1386

Table 16 performance of Three Classifier Using TF-IDF Vectorization with stemming

	Logistic Regression			Naive Bayes			SVM		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
EAP	77	85	81	77	86	81	76	84	80
HPL	82	78	80	86	75	80	81	75	78
MWS	83	77	80	82	80	81	82	76	79

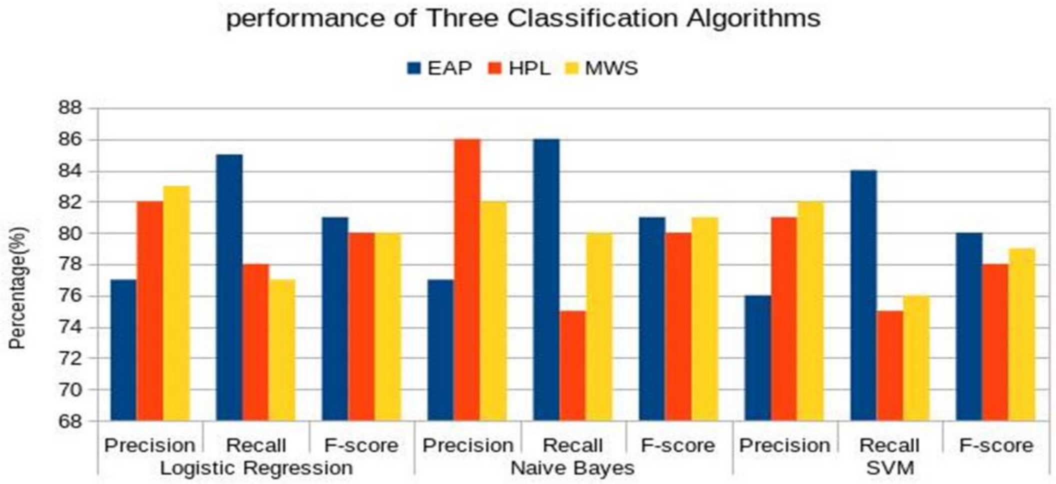

Figure 9. performance of Three Classifier Using TFIDF Vectorization with stemming

We trained and tested the three classification models with help of TF_IDF vectorization using 15363 vector size. After removed punctuation marks, stop words and stem. We selected vocabulary of 15363 out of 15363. blow tables show the confusion matrix and performance of three classification algorithms. For Logistic Regression, Naive Bayes and Support Vector Machines, we obtained accuracy 80.13, 81.06 and 80.09 by using counter vectorization with a vector size of 15363.

Table 17 Confusion matrix for Three Classifier Using TF-IDF Vectorization with stemming

	Logistic Regression			Naive Bayes			SVM		
	EAP	HPL	MWS	EAP	HPL	MWS	EAP	HPL	MWS
EAP	2060	164	191	2125	82	208	2049	162	204
HPL	289	1272	118	362	1181	136	290	1273	116
MWS	315	90	1375	278	46	1456	295	102	1383

Table 18 performance of Three Classifier Using TF-IDF Vectorization with stemming

	Logistic Regression			Naive Bayes			SVM		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
EAP	77	85	81	77	88	82	78	85	81
HPL	83	76	79	90	70	79	83	76	79
MWS	82	77	79	81	82	81	81	78	79

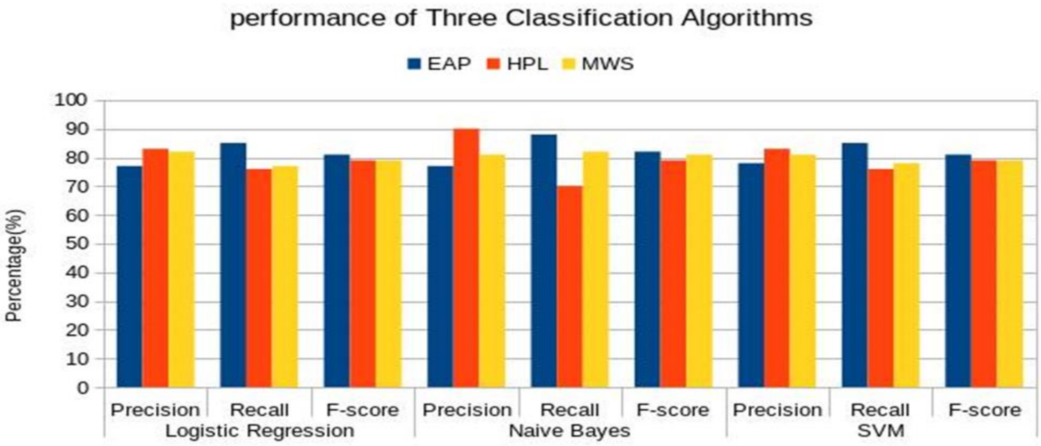

Figure 10. performance of Three Classifier Using TFIDF Vectorization with stemming

Table 19 Accuracy of three Algorithms with different features and Vector size

Accuracy	Vector Size	Logistic Regression	Naive Bayes	SVM
Accuracy using Count Vector with Stop words	5000	79.24	79.80	78.49
Accuracy using Count Vector with Stop words	24823	81..12	82.09	81.20
Accuracy using Count Vector with Stop words+Stem	5000	78.20	80.06	74.49
Accuracy using Count Vector with Stop words+Stem	15363	79.74	82.75	76.60
Accuracy using TF-IDF Vector with Stop words	5000	79.21	79.97	78.26
Accuracy using TF-IDF Vector with Stop words	24823	79.50	79.58	79.65
Accuracy using TF-IDF Vector with Stop words+Stem	5000	80.32	80.88	78.99
Accuracy using TF-IDF Vector with Stop words+Stem	15363	80.13	81.06	80.09

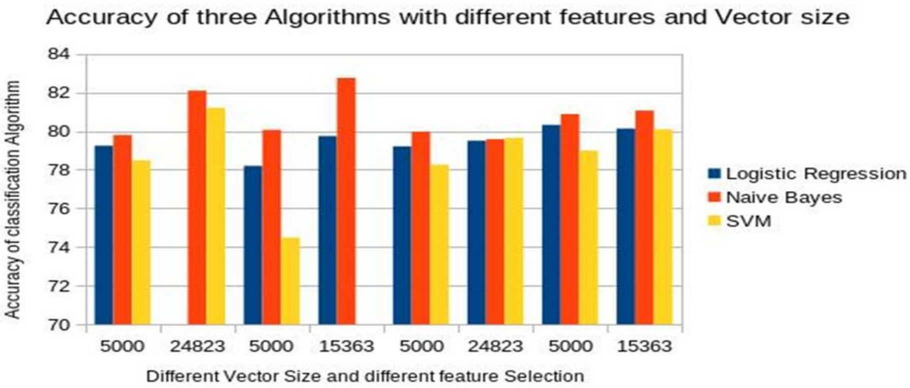

Figure 11. Accuracy of three classification algorithms

5. Conclusion and Future Approaches

We have designed and tested three machine learning algorithms in this paper. Using 70% of training and 30% of test data, Logistic Regression (LR), Naive Bayes (NB) and Support Vector Machine(SVM) were used. With the help of vocabulary size for fixing the vector size, we used count vector and TF-IDF vector for building vectors. We delete punctuation characters and stop words from the corpus for vocabulary collection. After that, apply stemming for suffix elimination. For Logistic Regression, Naive Bayes and Support Vector Machines, we obtained accuracy 81.12, 82.09 and 81.20 by using counter vectorization with a vector size of 24823. By using TF-IDF vectorization with the vector size is 15363, we have accuracy of 80.13, 81.06 and 80.09 for Logistic Regression, Naive Bayes and Support Vector Machines. We have the best performance with maximum vocabulary length in counter vectorization.

The research may also be spent in different ways in future. The first is an increase in the number of class labels in the training set and also an increase in the number of documents in the training set per class, expanded by experimenting with the word2vec mode for semantic vectors

References

[1] Houvardas, John, and Efstathios Stamatatos. "N-gram feature selection for authorship identification." In International conference on artificial intelligence: Methodology, systems, and applications, pp. 77-86. Springer, Berlin, Heidelberg, 2006.

[2] Ferreira da Silva, J., Dias, G., Guilloré, S., Pereira Lopes, J.G.: Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. In: Barahona, P., Alferes, J.J. (eds.) EPIA 1999. LNCS (LNAI), vol. 1695, pp. 113–132. Springer, Heidelberg (1999)

[3] Silva, J., Lopes, G.: A local Maxima Method and a Fair Dispersion Normalization for Extracting Multiword Units. In: Proc. of the 6th Meeting on the Mathematics of Language, pp. 369–381 (1999)

[4] Zhang, Chunxia, Xindong Wu, Zhendong Niu, and Wei Ding. "Authorship identification from unstructured texts." Knowledge-Based Systems 66 (2014): 99-111.

[5] A. Jamal Nasir, Nico Gornitz, and Ulf Brefeld. “An off-the-shelf approach to authorship attribution”. In: (2014).

[6] Chen, Luke, Eric Gonzalez, and Coline Nantermoz. "Authorship Attribution with Limited Text on Twitter." (2017): 1-6.

[7] Loper, Edward, and Steven Bird. "NLTK: the natural language toolkit." arXiv preprint cs/0205028 (2002).

[8] Howedi, Fatma, and Masnizah Mohd. "Text classification for authorship attribution using Naive Bayes classifier with limited training data." Computer Engineering and Intelligent Systems 5, no. 4 (2014): 48-56.

[9] Gungor, Abdulmecit. "Benchmarking authorship attribution techniques using over a thousand books by fifty Victorian era novelists." PhD diss., 2018.

[10] Badirli, Sarkhan, Mary Borgo Ton, Abdulmecit Gungor, and Murat Dundar. "Open Set Authorship Attribution toward Demystifying Victorian Periodicals." arXiv preprint arXiv:1912.08259 (2019).

[11] Fatima, Shugufta, and B. Srinivasu. "Text Document categorization using support vector machine." International Research Journal of Engineering and Technology (IRJET) 4, no. 2 (2017): 141-147.

[12] Raschka, Sebastian, and Vahid Mirjalili. Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd, 2019.

[13] Garreta, Raul, and Guillermo Moncecchi. Learning scikit-learn: machine learning in python. Packt Publishing Ltd, 2013.