Research Paper Classification using Supervised Machine Learning Techniques

Shovan Chowdhury
Department of Mechanical Engineering
Measurement and Control Engineering Research Center
Idaho State University
Pocatello, USA
chowshov@isu.edu

Marco P. Schoen
Department of Mechanical Engineering
Measurement and Control Engineering Research Center
Idaho State University
Pocatello, USA
schomarc@isu.edu

***Abstract*— In this work, different Machine Learning (ML) techniques are used and evaluated based on their performance of classifying peer reviewed published content. The ultimate objective is to extract meaningful information from published abstracts. In pursuing this objective, the ML techniques are utilized to classify different publications into three fields: Science, Business, and Social Science. The ML techniques applied in this work are Support Vector Machines, Naïve Bayes, K-Nearest Neighbor, and Decision Tree. In addition to the description of the utilized ML algorithms, the methodology and algorithms for text recognition using the aforementioned ML techniques are provided. The comparative study based on four different performance measures suggests that – with the exception of Decision Tree algorithm – the proposed ML techniques with the detailed pre-processing algorithms work well for classifying publications into categories based on the text provided in the abstract.**

Keywords—Machine Learning, SVM, Naïve Bayes, K-NN, Decision Tree, Text Classification, TF-IDF.

I. Introduction

Dissemination through peer review of new findings and proposed theories is a fundamental element in advancing a field of study. When a researcher searches for a publication from the web, it is difficult to find the appropriate paper which they are looking for because a keywords search might lead to publications in a different field of study. Researchers spend a good amount of time finding appropriate publications that support their work. A categorization of research papers based on different research fields can solve this problem. Such a categorization may also be useful for managing a vast number of publications that are submitted to a conference venue or a particular journal. Reviewers may find the use of an automated categorization tool useful if the papers are classified by the respected research fields. It is obvious that manual classification of such documents is time consuming and often leads to unnecessary errors. Thereby, it is beneficial to develop a research paper classification systems that can classify papers automatically and adaptively.

In this present research, the aim is to automatically classify various research papers into three different fields: science, business, and social science using text classification and supervised machine learning techniques. Nowadays, text classification schemes have become very popular. Currently, there is a great deal of research being conducted with regard to text classification. The outcomes of this research are very useful in numerous fields, such as sentimental analysis, product evaluation etc. For example, Vignesh Rao [1] worked on classifying news article based on location. Bo Pang and Lillian Lee [2] worked on sentiment classification where they were able to find out whether a movie review is positive or negative. Application of text classification is also found in poem classification [3] and opinion mining [4].

In this present work, a balanced dataset is created which consists of abstracts from all three research fields: Science, Business and Social Science. For this purpose, abstracts were collected since abstracts mostly contain important information, keywords and specific features of the corresponding paper. These research abstracts are then classified using four different classification techniques. Text pre-processing is performed by using tokenization, stemming, and concluded by removing stop words. For vector representation of text, two different methods are used which are Term Frequency Inverse Documents Frequency method (TF-IDF) and Bag of word method. For classification, Support Vector Machines (SVM), Naïve Bayes, K-Nearest Neighbor (KNN), and Decision Tree classification algorithm are used. Finally, the results obtained by the four different classifiers are compared including the use of two vectorization methods. A very similar type of research is done by Sang-Woon Kim and Joon-Min Gil [5] where they used unsupervised machine learning (clustering) for classification. In contrast to their work, in the research presented in this paper, supervised linear and non-linear classification techniques are utilized.

978-1-7281-4291-3/20/$31.00 ©2020 IEEE

II. Background

A. Support Vector Machine

Support Vector Machine is a popular supervised machine learning technique which is used for classification as well as regression. In this method, multidimensional data is separated by a hyperplane. SVM uses kernel functions which systematically find support vector classifier in higher dimension. It actually doesn't transform the data into higher dimensions but it calculates the relationship between the data if they are in a higher dimension. The technique used for calculating relationships among the data is called kernel trick. There are different types of kernels in SVM from which polynomial, radial, and linear kernels are widely used. In the present work, linear kernel are utilized. There is another tuning method of data called regularization parameters (termed as C parameter) [6] which allows for deciding how much one wants to penalize misclassified points. For larger values of the C parameter, SVM divides all the data with small hard margins which doesn't allow any misclassification of training data. However, using the higher value of C parameter most often produces an overfit of the model which is not desirable.

B. Naïve Bayes

Naïve Bayes is a probabilistic classifier which works based on the Bayes theorem. It determines the probability of each feature occurring in each class and returns the most likely class [7].

The Bayes rule is defined as:

$$P(A/B) = \frac{P(A/B)\times P(A)}{P(B)} \qquad (1)$$

Where, A and B represent class and features, respectively.

P(A/B) stands for the probability of belonging to class *A* with all given features of B. *P(B)* is the probability of all features which is basically used for normalization.

Saleh Alsaleem [8] shows how Naïve Bayes algorithm works for text classification. As this algorithm works on simple probability theory, it works better for high dimensional data as well. The main task of using the Naïve Bayes algorithm is to find the probability of each features.

C. K-nearest Neighbor

K-nearest neighbor is an important classifier among the supervised learning algorithms. In this method, a new data is classified by its nearest neighbor's maximum vote. The distance between nearest neighbor is measured by a distance function which uses Euclidean distance, Manhattan distance, and the Minkowski distance method. The *K* value refers to the number of neighbors. If the *K* value is very low, then one will obtain less stable results. On the other hand, increasing the *K* value will allow to increase the error, however one will obtain stable results. Therefore, in the present work, the *K* value is chosen by trial and error so that no overfitting occurs. Fig. 2 shows a graphical representation of the KNN algorithm.

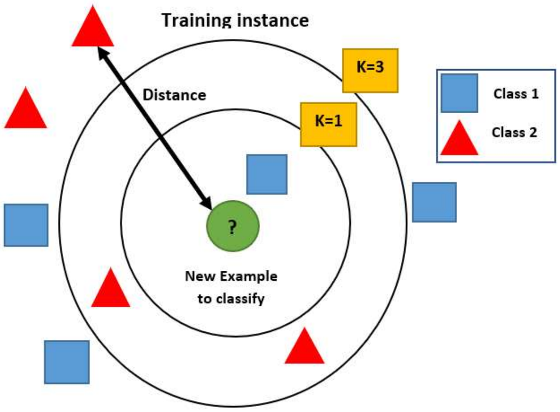

Fig 1: K-nearest algorithm

D. Decision Tree

The Decision tree method is one of the most intuitive machine learning methods among non-parametric supervised machine learning algorithms that can be used for both classification and regression. It is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. The learning algorithm behind decision tree is an inductive approach to learn knowledge on classification by splitting the source datasets into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called *recursive partitioning*. The recursion is completed when the subset at a node has the same value of the target variable, or when splitting no longer adds value to the predictions. Growing a tree involves deciding on which features to choose and what conditions to use for splitting, along with knowing when to stop. There are four types of decision tree algorithms, namely Iterative Dichotomiser (ID3), Classification and Regression Trees (CART), Chi-Square, and Reduction in Variance. The most popular one, the ID3 decision tree algorithm, uses Information Gain to decide the splitting points. In order to measure how much information, one gains from some attribute, one can use entropy to calculate the homogeneity of a sample. Entropy is a measure of the uncertainty of a random variable. Thus, acquisition of information corresponds to a reduction in entropy. The maximum gain of any of the attributes would let it to be chosen by the decision tree learning algorithm. Thus, it creates a comprehensive analysis of the consequences along each branch and identifies decision nodes that need further analysis. Though, decision trees are able to generate understandable rules, they are prone to errors in classification problems with many class and relatively small number of training examples.

III. Methodology

The target of the present research is to classify research abstracts into appropriate classes. The entire process of work is shown in Fig 2. First, one collects research paper abstracts from Science, Business and Social Science field and uses these as input data. In this work, 107 research abstract are collected to build the dataset where science and social science class consists of 36 abstracts and business class consists of 35 abstracts. These abstracts are collected from online sources such as Google Scholar, Research Gate, etc. Two-thirds of the data is used for training and the rest of the data is used for testing. Pre-processing of textual data is done by using natural language processing. After pre-processing, one uses four different machine learning algorithms to classify the data. Finally, the accuracy of the different algorithms is compared using precision, recall, and F1 score.

A. *Pre-processing of data*

Natural language processing is applied on textual data. For doing this, one can use the Natural Language Toolkit (NLTK). Fig 3 shows the steps involved for pre-processing of the data.

a) Tokenization: Tokenization is a process of breaking up a paragraph into words or other meaningful elements which are called tokens. It performs a crucial task in any natural language processing. It basically creates a list of arrays of words. For tokenizing the data, the NLTK library from Python is used in this work. For text processing, convertion of input text into tokens is necessary to keep allowable information in successive phase in machine learning.

b) Clean Text: In the list of words, there are also punctuation marks which are not necessary for classification and hence are removed from the tokenized list of words. Also, all the alphabet is converted into lower case because if an identical word stays in the list with both upper and lower case, it will be treated as two different words.

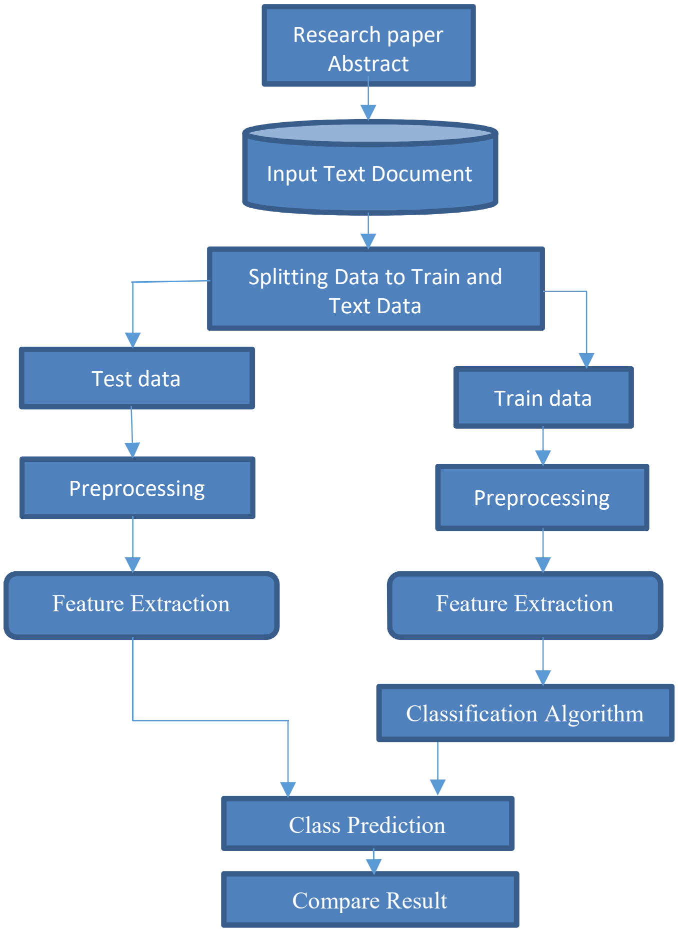

Fig 2: Flow chart of the proposed system

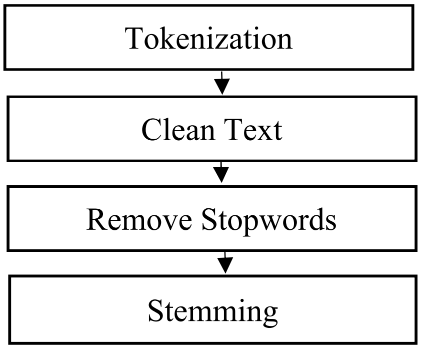

Fig 3: Pre-processing steps

a) Remove Stopword: Stopwords are the words which do not have any significance in classification. For example, if one considers a sentence "I am going there for sure," the words 'am' and 'for' has less importance. Hence, these stopwords are removed from the data.

b) Stemming: Stemming is the process of reducing infected or derived words to their word of base root form. For example: 'go' is the base root of 'go', 'went', 'gone', 'going' etc. This is a very important part of natural language processing. One can reduce such word lists by applying stemming.

B. *Feature Extraction*

For using the data as input in a machine learning algorithm, one needs to vectorize the text data as one has to give a numeric input. In the present research, two types of vectorizer are used. One is bag of words (count vectorizer) and another is the TF-IDF vectorizer. In case of count vectorizer, it counts all of the words as input. All the words have the same importance. No semantic information is preserved in the count vectorizer method. This is a less efficient vectorization method as not all the words are needed. Some words can be present in all science, business, and social science dataset. Hence, it is adventegeous to remove all the uncommon words. That is the reason why another method - the TF-IDF method – has a better performance. TF-IDF means term frequency inverse document frequency. In this method, some semantic information is preserved as uncommon words are given more importance than common words. For Example: let's take a sentence fom a movie review dataset. 'The movie is excellent.' Here 'excellent' will have more importance than 'The' or 'movie.' This 'movie' word can be present in a maximum number of the reviews in the documents. Hence, this word document frequency will be high. When calculating TF-IDF, the corresponding value will be low. Only the word which has higher TF-IDF value will be taken as input. Sang-Woon Kim and Joon-Min Gil [5] showed how to calculate TF-IDF in their research.

C. *Classification of Data*

In this stage, one classifies the data using the four machine learning algorithms. First, the SVM method is used to classify the data. Linear SVM is used with a C parameter equal to 1.0. Then the Naïve Bayes classifier method is employed to predict the class. In this case, multinomial Naïve Bayes are used.

TABLE I. PARAMETER USED FOR DIFFERENT ALGORITHM

Algorithm	***SVM***	***Naïve Bayes***	***KNN***	***Decision Tree***
Parameter Used	Kernel=linear, C=1	Multinomial NB	K=15	Max depth=15

Following this, the *K*-nearest neighbor algorithm is used where a value for *K* of 15 is assumed. Finally, the decision tree algorithm is used with a maximum depth of 15. A summary for all parameters used for different algorithm is shown in TABLE I.

IV. Results

Results are evaluated by precision, recall, F1 score, and accuracy, [8] [1].

- Precision: The precision means the ratio of positively identified outcomes that are correct, i.e.

$$Precision = \frac{True\ positive}{True\ positive + Fals\ \ positive}$$

- Recall: Recall tells what proportion of the data that actually is positive were predicted positive. In other words, the proportion of True Positive in the set of all actual positive data.

$$Recall = \frac{True\ Positive}{True\ positive + False\ Negative}$$

- F1 Score: It combines precision and recall and calculates it as the harmonic mean of precision and recall.

$$F1\ Score = \frac{2 \times precision\ \times recall}{precision + recall}$$

- Accuracy: The accuracy means the proportion of the total number of result which is correct.

$$Accuracy = \frac{True\ positive + True\ Negative}{Total\ Predictions}$$

- Confusion Matrix: It is also known as error matrix [9] which evaluates the performance of a classification algorithm.

A. Result of SVM

In the results presented in TABLE I, the precision, recall and F1 score is presented for three of the individual class separately and this is done by both TF-IDF and Bag of words vectorization method. For the SVM model, better results are obtained when using the TF-IDF vectorization method rather than the Bag of words method. From TABLE II, the overall weighted average accuracy, precision, recall, and F1 score for the testing data is listed. This table indicates that the F1 score is 89% and the accuracy is 88% for TF-IDF method, which outperforms the results from the Bag of words method. Fig 4 shows the confusion matrix for SVM result with TF-IDF.

B. Result of Naïve Bayes

Table III lists the results for the TF-IDF and Bag of words vectorization methods using the precision, recall and F1 score for three of the classes separately.

TABLE II. RESULT FOR THREE INDIVIDUAL CLASS IN SVM

Features		**SVM**		
		Precision	***Recall***	***F1-Score***
TF-IDF	Scientific	0.923	0.800	0.857
	Business	0.889	0.889	0.889
	Social Science	0.857	1.000	0.923
Bag of words	Scientific	0.923	0.800	0.857
	Business	0.857	0.667	0.750
	Social Science	0.750	1.000	0.857

TABLE III. OVERALL RESULTS FOR SVM METHOD

	SVM (Support Vector Machine)			
	Precision	***Recall***	***F1-score***	***Accuracy***
TF-IDF	0.890	0.896	0.890	0.888
Bag of words	0.843	0.822	0.821	0.8333

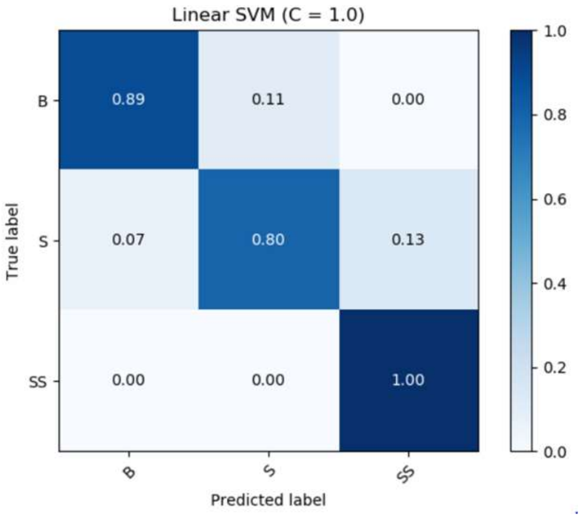

Fig 4: Confusion matrix for SVM

For both TF-IDF and Bag of words methods, one is obtaining good satisfactory results including the F1 score. From TABLE IV, one finds that the F1 score is 83% and accuracy is 83% which is almost the same for both vectorization methods. The confusion matrix acquired after implementing Naïve Bayes method with feature extractor TF-IDF is shown in Fig. 5.

TABLE IV. RESULT FOR THREE INDIVIDUAL CLASS IN NAÏVE BAYES

Features		**Naïve Bayes**		
		Precision	***Recall***	***F1-Score***
TF-IDF	Scientific	0.923	0.800	0.857
	Business	0.857	0.667	0.750
	Social Science	0.750	1.000	0.857
Bag of words	Scientific	0.923	0.800	0.857
	Business	0.727	0.889	0.800
	Social Science	0.833	0.833	0.833

TABLE V. OVERALL RESULTS FOR NAÏVE BAYES METHOD

Features	Naïve Bayes			
	Precision	*Recall*	*F1-score*	*Accuracy*
TF-IDF	0.843	0.822	0.821	0.833
Bag of words	0.828	0.841	0.830	0.833

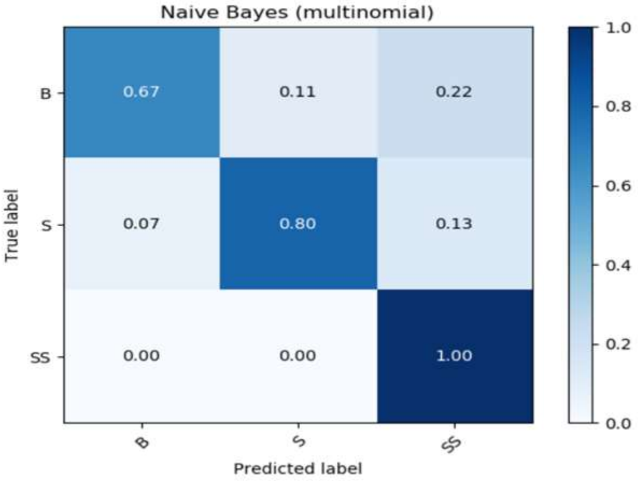

Fig 5: Confusion matrix for Naïve Bayes

C. *Result of KNN*

From TABLE V, one can notice that a better result is obtained when the TF-IDF vectorization method is used rather than the Bag of words method. The Bag of words method is not working well for KNN applications. From TABLE VI, it is observed that overall accuracy, precision, recall and F1 score is better for the TF-IDF vectorization method, which indicates that the Bag of words method does not perform well for this application and corresponding data. Fig 6 represents the confusion matrix for KNN using the TF-IDF method.

TABLE VI. RESULT FOR THREE INDIVIDUAL CLASS IN KNN

Features		KNN algorithm		
		Precision	*Recall*	*F1-Score*
TF-IDF	Scientific	0.923	0.800	0.857
	Business	0.778	0.778	0.778
	Social Science	0.857	1.000	0.923
Bag of words	Scientific	0.452	0.933	0.609
	Business	0.667	0.222	0.333
	Social Science	0.500	0.083	0.143

TABLE VII. OVERALL RESULTS FOR KNN METHOD

Features	K-Nearest Neighbor			
	Precision	*Recall*	*F1-score*	*Accuracy*
TF-IDF	0.853	0.859	0.853	0.861
Bag of words	0.539	0.413	0.362	0.472

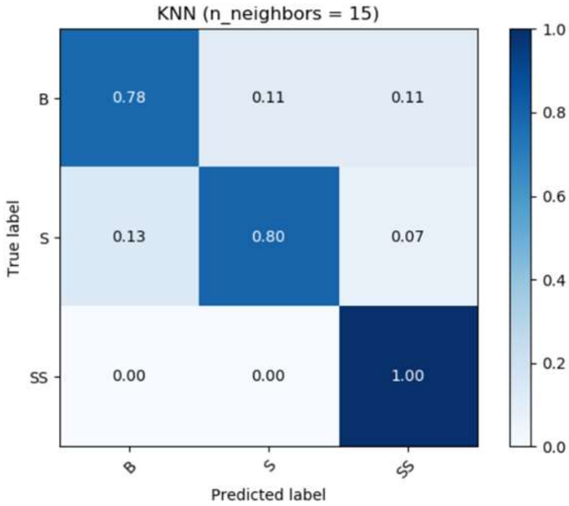

Fig 6: Confusion matrix for KNN

TABLE VIII. RESULT FOR THREE INDIVIDUAL CLASS IN DECISION TREE

Features		Decision Tree		
		Precision	*Recall*	*F1-Score*
TF-IDF	Scientific	0.727	0.533	0.615
	Business	0.833	0.556	0.667
	Social Science	0.526	0.833	0.645
Bag of words	Scientific	0.818	0.600	0.692
	Business	0.833	0.556	0.667
	Social Science	0.474	0.750	0.581

TABLE IX. OVERALL RESULTS FOR DECISION TREE METHOD

Features	Decision Tree			
	Precision	*Recall*	*F1-score*	*Accuracy*
TF-IDF	0.696	0.641	0.642	0.638
Bag of words	0.708	0.635	0.647	0.638

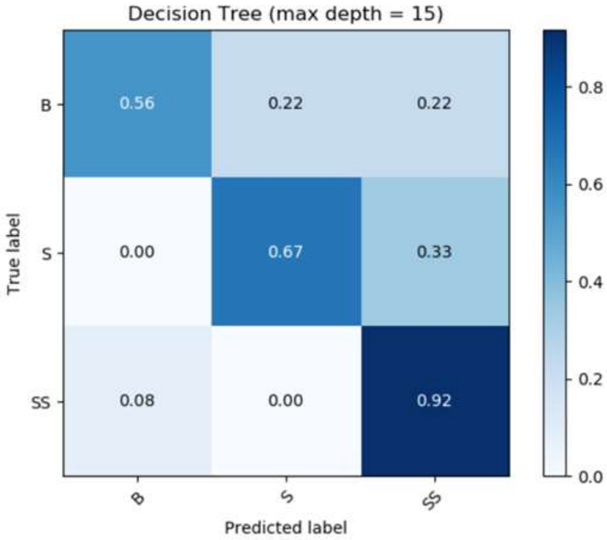

Fig 7: Confusion matrix for Decision Tree

D. *Result of Deicsion Tree*

TABLE VII indicates that one obtains almost the same result for both TF-IDF and Bag of words vectorization methods using Decision Tree algorithm. However, neither of the methods produces results that are satisfactory. The confusion matrix for the Decision Tree method is shown in Fig 7 for TF-IDF.

E. *Comparison of results*

Comparison of accuracy, precision, recall and F1 score is given in Figs 8-11. By observing these figures, it is apparent that the SVM method is performing well and delivers better results while the Decision Tree method performs the worst for this application and data set.

V. CONCLUSION

In this paper, a classification scheme using different machine learning methods is being used to classify research paper abstracts. The methods are based on three different types of domain using four different techniques. The classification scheme is programmed in PyCharm as a Python platform for classification. The machine learning algorithms employed are SVM Naïve Bayes, k-Nearest Neighbor, and Decision Tree. It is found that the SVM method outperforms the other methods, in particular the Decision Tree method, while KNN and Naïve Bayes method do comparatively well. The assessment is done using accuracy, precision, recall and F1-score. In the future, one can collect more abstract by using an automatic tool. The algorithm might perform better when more data is utilized.

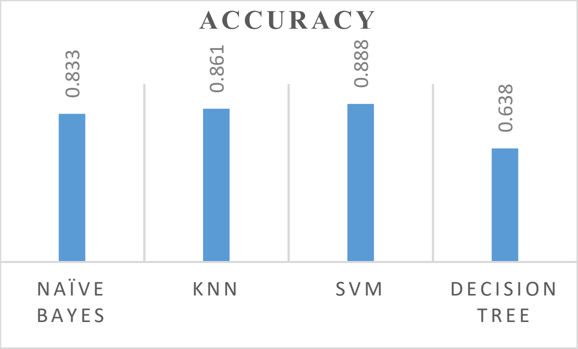

Fig 8: Accuracy Comparison

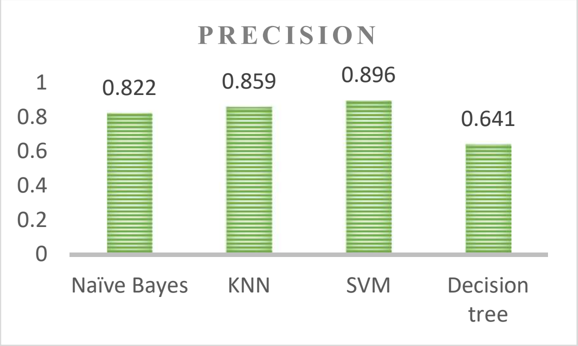

Fig 9: Precision Comparison

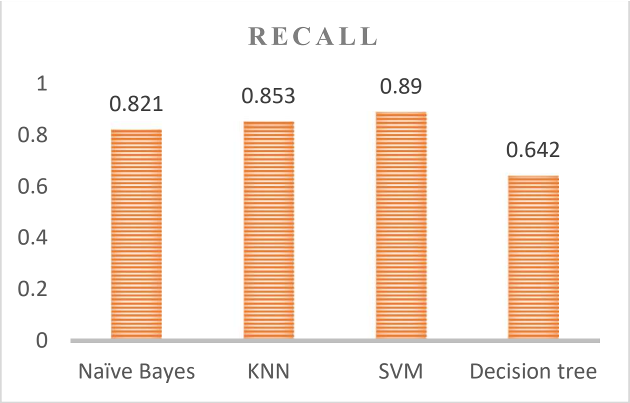

Fig 10: Recall Comparison

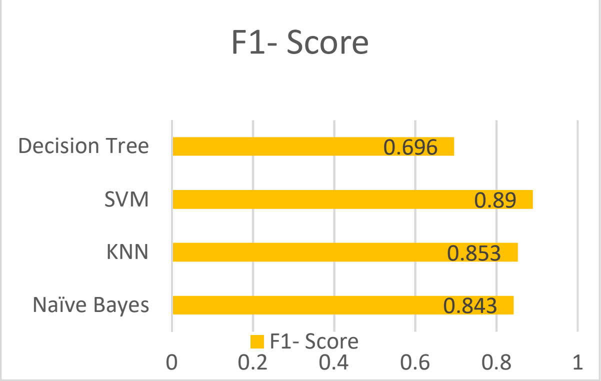

Fig 11: F1 Score Comparison

REFERENCES

[1] V. Rao and J. Sachdev, "A machine learning approach to classify news articles based on location," 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, 2017, pp. 863-867.

[2] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," Language Processing (EMNLP), Philadelphia, July 2002, pp. 79-86.

[3] V. Kumar and S. Minz, "Poem classification using machine learning approach," Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012 pp 675-682.

[4] J. Khairnar and M. Kinikar, "Machine learning algorithm for opinion mining and sentiment classification," International Journal of Scientific and Research Publications, Volume 3, Issue 6, June 2013.

[5] S. Kim and J. Gil, "Research paper classifcation systems based on TF-IDF and LDA schemes" Hum. Cent. Comput. Inf. Sci. (2019) 9: 30. https://doi.org/10.1186/s13673-019-0192-7.

[6] S. Patel, "Chapter 2 : SVM (Support Vector Machine) - Theory", 2017. [Online] Available: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72. [Accessed: 03- May- 2017].

[7] D. Sony, "Introduction to Naive Bayes Classification", 2018. [Online]. Available: https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54. [Accessed: 16- May- 2018].

[8] S. Alsaleem, "Automated arabic text categorization using SVM and NB," International Arab Journal of e-Technology, Vol. 2, No. 2, June 2011.

[9] S. Stehman, "Selecting and interpreting measures of thematic classification accuracy," Remote Sensing of Environment (1997).