PERFORMANCE ASSESSMENT OF SUPERVISED MACHINE LEARNING TECHNIQUES FOR DOCUMENT CATEGORIZATION

Dr. B. Lavanya[1]*, V. Nirmala[2]

Abstract
In many applications, data mining techniques are used as a regular practice to analyze the vast amount of available data and extract relevant knowledge and information to support the main decision-making processes. The content-based document classification system assigns a document to one of the specified classes by using the content and some weighting criteria. The classification of documents using Random Forest, K-Nearest Neighbor, Multinominal Naive Bayes, Multinominal Logistic Regression, and Support Vector Machine is examined in this study. Here, the metadata parameters were chosen from the seven subjects collected from the IEEE dataset domain, such as title, author keywords, and IEEE terms. It is used for classifying data into different classes by considering some constraints. In order to give the best outcome, we compare these five algorithms. The Logistic regression classification technique performs better which might also aid a Course Recommender System.

Keywords: Random forest, K- Nearest Neighbour, Multinominal Naïve Bayes, Multinominal Logistic regression, and Support Vector Machine

[1]*Associate Professor, Department of Computer Science, University of Madras, Chennai, India.
Email: lavanmu@gmail.com
[2]Research Scholar, Department of Computer Science, University of Madras, Chennai, India.
Email: nirmala7488@ymail.com

***Correspondence Author:-** Dr. B. Lavanya
*Associate Professor, Department of Computer Science, University of Madras, Chennai, India.
Email: lavanmu@gmail.com

DOI: - 10.48047/ecb/2023.12.si5a.0151

I. Introduction

Google Scholar provides over 3.2 million relevant research papers in response to a user's query. The majority of these publications are not even the query's area of concern. It will take a long time to read through all of these papers. Ten papers every day must be read for almost 158 years. This is due to incorrectly indexed or categorized documents in these repositories for their respective classes. This paper experiments with 700 IEEE research papers on subject classification. Each article published in that publication receives a subject category that corresponds to that journal. On account of its flaws, this journal-level subject classification of publications has frequently been criticized. We think that placing these publications in- their appropriate areas will improve the performance of these systems.

In As in [1] proposes a comprehensive evaluation of metadata, and combinations to obtain the objective of research paper classification. It offers a comprehensive analysis of the metadata of research publications, first individually and then collectively, using various combinations to classify papers into various categories recommended by ACM. It includes several information elements for studies from the field of computer science. We have taken the title, abstract, general terms, and keywords out of this data.

The Association for Computer Machinery (ACM) created the Computing Classification System (CCS) to categorize computing topics (ACM). Many ACM publications employ CSS to categorize topics.

The ACM Computing Classification System has three tiers (ACM CCS). The suggested model places articles in the most prestigious ACM categories. Instead of using the most recent method for calculating semantic similarity for feature transformation, we employ words' embeddings. As in [2] Researchers have suggested a variety of strategies for classifying research publications in the literature. Citations, meta-data, content-based, and hybrid approaches are the different subcategories of these methods. As in [3] paper outlines our method for developing the hierarchical text classifier for the CINDI Digital Library's pilot project. The categorization process used in the created classification system is top-to-down and coarse-to-fine. We experiment using an identity corpus of the Computer Science papers stored in the Association for Computer Machinery ACM DL to evaluate the performance of our system. For textual document classification.

As in [4] some authors have used hybrid methods. As in [5] author locates the essential important document clusters based on the contextual keywords, and a hybrid document clustering similarity index is optimized in this work. The clustered documents on the huge corpus are finally classified using a hybrid document classification model. As in [6] research proposes two different classification techniques Support Vector Machine (SVM) and Relevance Vector Machine (RVM), In this SVM tries to divide the different classes into a large gap (hyperplane) as is physically possible. In order to do this, it represents the data instances as points in space. On other hand, this separation space is defined by RVM using a probabilistic measure. classification datasets show that while RVM requires more training time than SVM, its categorization is a much better result. As in [7] The researcher's process of classifying data involves grouping information into categories or groups so that information from the same group is more similar and information from different groups is extremely dissimilar. Each instance is given a class via the classification algorithm so that the classification error is minimized. It is used to extract models from the input dataset that precisely describe significant data classes. As in [8] authors have provided a method for classifying the subjects of scientific articles based on an examination of their interrelationships. Citations, common authors, and reference-based metrics have all been used in the study. To do this, a relationship graph has been created in which research papers are represented by nodes, and the connections between these nodes establish the relationships between the publications. The findings of that study showed that dense, tightly packed graphs offer good results.

II. Related work

If the document repository is small, manual grouping is feasible. The exponential rise of data increases the size of the document repository. This makes it challenging to arrange the documents correctly. The importance of document clustering in organizing these documents is very crucial. As in [9] the author presents better optimization results, these algorithms are typically implemented utilizing a domain-independent methodology. To categorize the document sets from a big corpus, many evolutionary techniques are utilized, including genetic algorithms, Rough-set, SVM, etc. On enormous datasets, genetic algorithms are used to discover complex patterns and classification rules. As in [10] the classification procedure consists of two primary stages. The construction of the classification model takes place in the first phase, training. The second is the classification

process itself, in which an unknown data object is assigned to one of a set of class labels using the trained model. Classification is also called supervised learning, as the instances are given with known labels, in contrast to unsupervised learning in which labels are not known. Each instance in the dataset used by the supervised or unsupervised learning method is represented by a set of features or attributes which may be categorical or continuous. Building a classification model using a training set of database instances and associated class labels are used. When the values of the predictor characteristics are known, the resulting model is then used to forecast the class label of the testing cases. Supervised classification is one of the tasks most frequently carried out by intelligent techniques. A large number of techniques have been developed as in [11] the author groups research articles based on factors like citation linkages and type. Based on their findings, the authors created the PRESRI classification tool for research articles. The journal makes use of features based on author names or title words. As in [12] Three metrics have been used by the authors to assess the suggested strategy: Three internal characteristics of the conditional probability of symbols average over matching fragments in suffix trees representing texts and phrases are CPAMF, a famous characteristic of the likelihood of term generation, BM25, typical vector area representations of texts coded with tf-idf weighting, and the cosine relevance score between them. Additionally, they have thought about using an abstract collection of research publications from the ACM digital library for their investigations. According to their experimental findings, the CPAMF performs admirably better than both the cosine measure and BM25. As in [13] Text representation is one of the most essential problems in text mining and information retrieval. Text representation aims to turn unstructured text input into documents that can be quantified statistically. The most up-to-date techniques currently in use make use of conventional statistical measurements including Term Frequency (TF), Bag of Words (BOW), Term Frequency, and Inverse Document Frequency (TFIDF).

III. METHODOLOGY

A. Text pre-processing

The text mining process can be fast and efficient by text preprocessing. First, the text should be converted into the lower case then second the pre-processing methods are applied like removing numbers, punctuations, stop words, and finally stemming.

B. Noise Removal

Noise Removal is the first preprocessing method we used. Noise is an inherent issue that has an impact on the data preparation and collection operations in data mining. programs that may make mistakes. The two basic causes of noise, One is the implicit mistakes that measurement equipment, such as various types of sensors, introduce. The second is random errors that are introduced by batch methods or specialists when the data are obtained, such as during a process of document digitization[14]. The first is called Class Noise and it includes examples that are both contradictory and incorrectly categorized. The second type, Attributes Noise, includes values that are incorrect, missing, or unimportant. In our instance, the missing values in the dataset represent the noise. In the literature, there are various methods for handling missing values, including deleting records[15].

C. Stop words Removal

Nearly all written documents contain commonly used keywords (stop words) that never allow us to identify the text. Unhelpful stop words are eliminated from the text corpus during the stop word removal phase.

D. Punctuation Removal

The text corpus is cleaned up after text mining because punctuation and numerals in unstructured text documents are meaningless.

E. Stemming

Stemming is the process to obtain the words to their root form. In this work, we have used the Porter Stemmer for the stemming process.

G. Vectorization

By using the vectorization technique, we can speed up the execution of our code. When we are implementing an algorithm from the mark, it is a really attractive and significant technique to optimize algorithms. Each and every single word is vectorized by a count vectorizer and the labels are encoded it. Each input is tokenized, preprocessed, and represented as a sparse matrix in this case. The text is changed to lowercase using a Count vectorizer, which also employs word-level tokenization. The most frequent words or features will be chosen using the Count Vectorizer. For the max features, absolute values are required.

IV. Machine learning techniques

C. Random forest:

A random tree is one that is selected at random from a set of potential trees. Because each tree in the collection has an equal chance of being sampled, it

is known as a random tree. It means that a random tree is a tree chosen at random from a set of potential trees, with "m" random properties at each node. These trees are quite effective because they produce models that are more precise [16]. In recent years, the field of machine learning has used random tree models extensively.

B. k- Nearest Neighbor (KNN):

Among all machine learning algorithms, one of the K-Nearest Neighbor Algorithms is the most straightforward [19]. The instance-based learning method K-Nearest Neighbor uses. Because they keep all of the training examples and wait to create a classifier until a new, unlabeled sample needs to be identified, instance-based classifiers are also known as lazy learners [20]. While eager-learning algorithms (such as decision trees, neural networks, and Bayes networks) require more computation time during the classification process, lazy-learning techniques (such as Bayes networks) require less computation time during the training phase[21] [22].

C. Multinominal Naive Bayes

A simplistic probabilistic classifier based on using Baye's Theorem with firm independence assumptions is the multinominal naive Bayes classifier. The posterior probability that a document belongs to one of several classes is calculated by this procedure, and the document is then assigned to the class with the highest posterior probability[23]. The Naive Bayes classifier is a group of many methods, all of which are based on the idea that each feature being classified is independent of every other feature. Then the existence or absence of one feature has no bearing on the other feature's existence or absence.

D. Multinominal Logistic regression

When there are more than two categories and the dependent variable is nominal (equivalently categorical, indicating it fits into any one of a set of categories that cannot be ordered meaningfully), multinomial logistic regression is utilized.

E. Support vector machine

An effective and widely used supervised classification technique for text categorization is the support vector machine. There are many features to take into account when learning text classifiers. SVMs have the capacity to manage these expansive feature spaces because they employ overfitting protection, which is independent of the number of features[17]. Using a training set whose class label is known, the SVM algorithm creates a model and creates a hyperplane that divides the training set according to the class label. The model is then applied with test sets to predict the class label based on the hyperplane[18].

VI. EVALUATION METRICS

The excellent method for evaluating classification tasks is to calculate the percentage of classified corrected documents. It is the precision, recall, and harmonic mean frequently used in text mining, and information retrieval to evaluate the effectiveness of classification.

- **Precision**

A classification model's ability to isolate only the pertinent data points. Precision is calculated by dividing the total number of true positives by the total number of true positives + false positives.

$$Precision = \frac{\text{True Positive}}{\text{True Positive+False Positive}} \quad (1)$$

- **Recall**

The ability of a modal to locate all pertinent instances in a data source. Recall is calculated mathematically as the of the true positives and false negatives divided by the number of true positives.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive+True Negative}} \quad (2)$$

- **F1 Score**

The F1 score is a machine learning evaluation metric that assesses the precision of a model. It combines a model's recall and precision scores. The accuracy statistic determines how frequently a model is correctly predicted throughout the full dataset.

$$\text{F1score} = 2 * \frac{\text{Precision*Recall}}{\text{Precision+Recall}} \quad (3)$$

IV. RESULT AND DISCUSSION

The various classification models, including Support Vector Machine, Random Forest, k Nearest Neighbor, Multinominal Naive Bayes and Logistic Regression shows the precision, Recall, and F1-score in the below tabular column.

A. Dataset

The collection includes a variety of subject journals that were retrieved from IEEE, each with the required metadata in Python Jupyter Notebook implementation, including the title, author keywords, and IEEE terms. There were two phases to this model: training and testing. 60% of the samples in the first set are used for training, and 40% are used for testing. Table 1 describes the number of documents obtained from different subjects. A label was given to each subject to indicate which document belonged to that subject.

TABLE1: DATASET COLLECTION

Class	Subjects	No. of documents
0	Biochemistry	100
1	Chemistry	100
2	Computer science	100
3	Energy	100
4	Environmental science	100
5	Mathematics	100
6	Physics	100

Table 2 illustrates the classification of performance under the three metrics of Accuracy, Recall, and F1-score for each class included. It shows the performance of the random forest classifier, which shows an overall level above 87% accuracy of classified corrected documents in the subject class with this precision metric. It shows the second-best performance among the methods studied,

TABLE 2: RANDOM FOREST METRICS

Sno	RANDOM FOREST		
CLASS	***PRECISION***	***RECALL***	***F1-SCORE***
0	0.91	1.00	0.95
1	0.94	0.85	0.89
2	0.87	0.65	0.74
3	0.95	1.00	0.98
4	0.95	0.90	0.92
5	0.61	0.85	0.71
6	1.00	0.85	0.92
Accuracy			0.87
Macro avg	0.89	0.87	0.87
Weighted avg	0.89	0.87	0.87

Table 3 illustrates the classification of performance under the three metrics of Accuracy, Recall, and F1-score for each class included. It shows the performance of the k- Nearest Neighbor classifier, which shows an overall level above 61% accuracy of classified corrected documents in the subject class with this precision metric. It deoicts the smallest performance under this metric when compared to other models.

TABLE 3:K NEAREST NEIGHBOR METRICS

Sno	KNN		
CLASS	***PRECISION***	***RECALL***	***F1-SCORE***
0	0.88	0.35	0.50
1	1.00	0.55	0.71
2	1.00	0.10	0.18
3	1.00	0.90	0.95
4	1.00	0.40	0.57
5	0.38	1.00	0.55
6	0.47	0.95	0.63
Accuracy			0.61
Macro avg	0.82	0.61	0.58
Weighted avg	0.82	0.61	0.58

Table 4 illustrates the classification of performance under the three metrics of Accuracy, Recall, and F1-score for each class included. It shows the performance of the Multinominal Naïve Bayes classifier, which shows an overall level above 86% accuracy of classified corrected documents in the subject class with this precision metric. It is the third-level performance under this metric when compared to other models.

TABLE 4: MULTINOMINAL NAIVE BAYES METRICS

Sno	**Naïve Bayes**		
CLASS	***PRECISION***	***RECALL***	***F1-SCORE***
0	0.90	0.95	0.93
1	0.89	0.80	0.84
2	0.94	0.75	0.83
3	0.77	1.00	0.87
4	0.94	0.80	0.86
5	0.77	0.85	0.81
6	0.85	0.85	0.85
Accuracy			0.86
Macro avg	0.87	0.86	0.86
Weighted avg	0.87	0.86	0.86

Table 5 illustrates the classification of performance under the three metrics of Accuracy, Recall, and F1-score for each class included. It shows the performance of the multinomial logistic regression classifier, which shows an overall level above 91% accuracy of classified corrected documents in the subject class with this precision metric. It shows the superior quality performance under this metric when compared to other models.

TABLE 5: LOGISTIC REGRESSION METRICS

Sno	**LOGISTIC REGRESSION**		
CLASS	***PRECISION***	***RECALL***	***F1-SCORE***
0	0.91	1.00	0.95
1	0.74	0.85	0.79
2	0.94	0.75	0.83
3	1.00	1.00	1.00
4	0.95	1.00	0.98
5	0.94	0.85	0.89
6	0.90	0.90	0.90
Accuracy			**0.91**
Macro avg	0.91	0.91	0.91
Weighted avg	0.91	0.91	0.91

Table 6 illustrates the classification of performance under the three metrics of Accuracy, Recall, and F1-score for each class included. It shows the performance of the support vector machine classifier, which shows an overall level above 85% accuracy of classified corrected documents in the subject class with this precision metric. It is the fourth-level performance under this metric when compared to other models.

TABLE 6: SUPPORT VECTOR MACHINE METRICS

Sno	**Support Vector Machine**		
CLASS	***PRECISION***	***RECALL***	***F1-SCORE***
0	0.95	0.95	0.95
1	0.58	0.75	0.65
2	0.71	0.75	0.73
3	1.00	0.90	0.95
4	0.95	0.95	0.95
5	0.95	0.90	0.92
6	0.94	0.75	0.83
Accuracy			0.85
Macro avg	0.87	0.85	0.86
Weighted avg	0.87	0.85	0.86

B. Macro averaging/weighted avg

The harmonic mean of micro precision and micro recall is the micro averaging F-measure. To do this, we independently calculate the True Positive (TP), False Positive (FP), and False Negative (FN) of each unique document. It is an effective tool for assessing how well a categorization algorithm performs for specific document instances. An average in which each of the quantities to be averaged is given a weight is referred to as a

weighted average. .We can estimate the average relative relevance of each quantity due to this weighting. Considering that every value in the data set is given the same weight, a weighted average can be thought of as being more accurate than any simple average.

Table 7 shows, comparison of all the classification models with respect to their metrics like Accuracy, Macro avg, and Weighted avg performance. It illustrates that the performance of Multinominal Logistic regression out performs with above 91% accuracy than the other three models RF, NB, and SVM performed with a one-point difference. Here the KNN classification proves to under perform in this multi classification application.

TABLE 7: COMPARING ACCURACY CLASSIFICATION MODELS

Sno	Classification Method	Accuracy	Macro avg	Weighted avg
1	Logistic regression	**0.91**	**0.91**	**0.91**
2	Random forest	0.87	0.87	0.87
3	Naïve bayes	0.86	0.86	0.86
4	Support vector machine	0.85	0.86	0.86
5	KNN	0.61	0.58	0.58

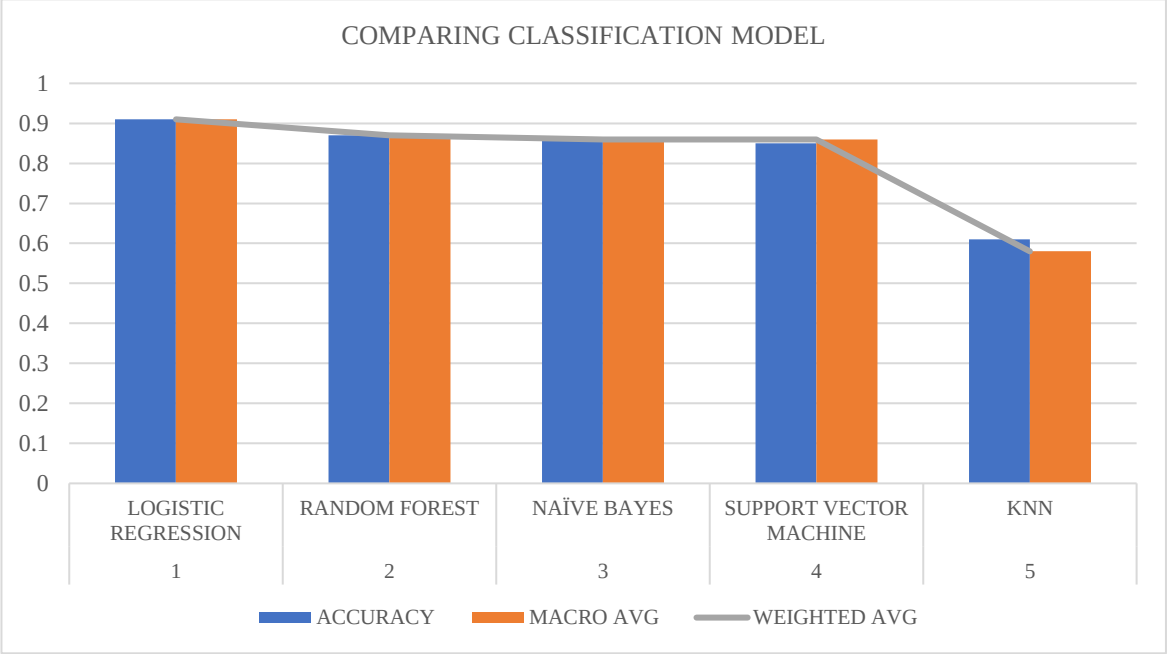

Fig 1. Comparison of models

Fig 1 compares the classification models on three different levels: accuracy, macro average, and weighted average. KNN has the lowest accuracy and logistic regression has the highest accuracy, were as the other three levels are only one point apart.

V. Conclusion

Using the above-mentioned database, various classification models have been tested for multi classification application. Different methods are employed in classification, including SVM, Naive Bayes, Logistic Regression, Random Forest, and K-Nearest Neighbors. From the study it is proven that the logistic regression algorithm performs ahead to other methods with above 91%. The categorization using the metadata parameter characteristics is done. A model can be designed to outperform logistic regression as future scope of this work.

References

1. Mustafa, Ghulam, et al. "A Comprehensive Evaluation of Metadata-Based Features to Classify Research Paper's Topics." IEEE Access 9 (2021): 133500-133509.
2. B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A Bayesian classification approach using class-specific features for text categorization," IEEE Trans. Knowl. Data Eng., vol. 28, no. 6, pp. 1602–1606, Jun. 2016.
3. T. Wang and B. C. Desai, "Document classification with ACM subject hierarchy," in Proc. Can. Conf. Electr. Comput. Eng., 2007,

pp. 792–795.[**** Wang, Tao, and Bipin C. Desai. "Document classification with ACM subject hierarchy." 2007 Canadian Conference on Electrical and Computer Engineering. IEEE, 2007.***]

4. M. N. Asim, M. U. G. Khan, M. I. Malik, A. Dengel, and S. Ahmed, ‘‘A robust hybrid approach for textual document classification,’’ in Proc. Int. Conf. Document Anal. Recognit. (ICDAR), Sep. 2019, pp. 1390–1396.
5. S. A. Devi and S. Siva, ‘‘A hybrid document features extraction with clustering based classification framework on large document sets,’’ Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 7, pp. 1–11, 2020.
6. Rafi, Muhammad, and Mohammad Shahid Shaikh. "A comparison of SVM and RVM for Document Classification." arXiv preprint arXiv:1301.2785 (2013).
7. Sagar S. Nikam, “A Comparative Study of Classification Techniques in Data Mining Algorithms”, Oriental Journal of Computer Science & Technology, Vol. 8, April 2015.
8. M. Taheriyan, ‘‘Subject classification of research papers based on interrelationships analysis,’’ in Proc. 2011 Workshop Knowl. Discovery, Modeling Simulation, 2011, pp. 39–44.
9. Z. Gero and J. Ho, “PMC Vec: Distributed phrase representation for biomedical text processing,” Journal of Biomedical Informatics: X, vol. 3, p. 100047, Sep. 2019, doi: 10.1016/j.yjbinx.2019.100047.
10. H. Bhavsar and A. Ganatra, “A Comparative Study of Training Algorithms for Supervised Machine Learning”, International Journal of Soft Computing and Engineering (IJSCE), Vol. 2, Issue. 4, September 2012
11. H. Nanba, N. Kando, and M. Okumura, ‘‘Classification of research papers using citation links and citation types: Towards automatic review article generation,’’ Adv. Classification Res. Online, vol. 11, no. 1, pp. 117–134, 2011.
12. E. Chernyak, ‘‘An approach to the problem of annotation of research publications,’’ in Proc. 8th ACM Int. Conf. Web Search Data Mining, Feb. 2015, pp. 429–434.
13. Yan, J., Hu, J. (2009). Text Semantic Representation. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_422
14. X. Zhu and X. Wu, ‘‘Class noise vs. attribute noise: A quantitative study,’’ Artif. Intell. Rev., vol. 22, no. 3, pp. 177–210, Nov. 2004.
15. J.-O. Kim and J. Curry, ‘‘The treatment of missing data in multivariate analysis,’’ Sociol. Methods Res., vol. 6, no. 2, pp. 215–240, Nov. 1977.
16. Aggarwal, Charu C., Mining Text Data: A Survey of Text Classification Algorithms, Springer US, 2012.
17. Pawar, Pratiksha Y., and S. H. Gawande. "A comparative study on different types of approaches to text categorization." Internatio nal Journal of Machine Learning and Computing 2.4 (2012): 423.
18. Shruti A, B.I Khodanpur.,” Comparative study of Advanced classification methods” International journal on recent and innovation trends in computing and communication.
19. Ajay, KP Soman Shyam Diwakar V. "Insight into Data Mining Theory, and Practice." Prentice Hall of India. India 11.11.2 (2006): 7-1.
20. Sun, Shiliang, and Daoming Zong. "Lcbm: a multi-view probabilistic model for multi-label classification." IEEE transactions on pattern analysis and machine intelligence 43.8 (2020): 2682-2696.
21. Rafi, Muhammad, and Mohammad Shahid Shaikh. "A comparison of SVM and RVM for Document Classification." arXiv preprint ar Xiv:1301.2785 (2013).
22. Han, Jiawei, Jian Pei, and Hanghang Tong. Data mining: concepts and techniques. Morgan kaufmann, 2022.
23. Kotsiantis, Sotiris B., Ioannis Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." Emerging artificial intelligence applications in computer engineering 160.1 (2007): 3-24.
24. Qayyum, Faiza, and Muhammad Tanvir Afzal. "Identification of important citations by exploiting research articles’ metadata and cue-terms from content." Scientometrics 11 (2019): 21-43.
25. Beel, Joeran, et al. "Research paper recommender system evaluation: a quantitative literature survey." Proceedings of the international workshop on reproducibility and replication in recommender system evaluation. 2013.