Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 5, No. 2, May 2020, Pp. 139-146

Classification of lexile level reading load using the k-means clustering and random forest method

Harits Ar Rosyid*[1], Utomo Pujianto[2], Moch Rajendra Yudhistira[3]
Universitas Negeri Malang, Indonesia[1,2,3]

Article Info
Keywords:
Text Classification, Lexile Levels, Clustering, K-Means, Random Forest

Article history:
Received 27 July 2019
Revised 05 August 2019
Accepted 17 January 2020
Published 06 May 2020

Cite:
Rosyid, H., Pujianto, U., & Yudhistira, M. (2020). Classification of Lexile Level Reading Load Using the K-Means Clustering and Random Forest Method. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 5(2).
doi:https://doi.org/10.22219/kinetik.v5i2.897

*Corresponding author.
Harits Ar Rosyid
E-mail address:
harits.ar.ft@um.ac.id

Abstract
There are various ways to improve the quality of someone's education, one of them is reading. By reading, insight and knowledge of various kinds of things can increase. But, the ability and someone's understanding of reading is different. This can be a problem for readers if the reading material exceeds his comprehension ability. Therefore, it is necessary to determine the load of reading material using Lexile Levels. Lexile Levels are a value that gives a size the complexity of reading material and someone's reading ability. Thus, the reading material will be classified based a value on the Lexile Levels. Lexile Levels will cluster the reading material into 2 clusters which is easy, and difficult. The clustering process will use the k-means method. After the clustering process, reading material will be classified using the reading load Random Forest method. The k-means method was chosen because of the method has a simple computing process and fast also. Random Forest algorithm is a method that can build decision tree and it's able to build several decision trees then choose the best tree. The results of this experiment indicate that the experiment scenario uses 2 cluster and SMOTE and GIFS preprocessing are carried out shows good results with an accuracy of 76.03%, precision of 81.85% and recall of 76.05%.

1. Introduction

Computing and internet network technology always growing from time to time. One of the impacts is the increased ease of sharing information through the Internet. By this condition, information becomes accessible quickly, easily, and cheaply. In addition, the device development getting smaller and mobile makes information easier obtained. As a result, reading material or article can be obtained easily. This facilitates the process of improving insight, knowledge and certainly can increase interest in a problem.

But this can be a problem towards the reader. One of them is a problem in the difficulty level of a word to read and understand, and the difficulty arrangement or syntax of a text [1]. If someone read a reading material that exceeds his reading ability, it will take a lot of time even though he got help from the instructor [2]. To solve this problem, we need to determination of reading load from a reading material, one of them uses Lexile Levels Lexile Levels.

Lexile levels Lexile levels are a value that gives a complexity value of reading material as well reading ability of each person [3]. But not all materials already have Lexile levels. Therefore, for simplify and accelerate in determining the reading load from a reading material, a system is needed for categorize and classify reading load from a reading material levels.

To be able to categorize the reading load from reading material, reading material will be grouped and classified based on the Lexile Levels. By rapid technological development, technology can be used for solving this problem, namely machine learning and information retrieval.

Machine learning is a branch of artificial intelligence who composes and makes connections between data and information by systematically utilizing algorithms [4]. The methods that exist in machine learning, can used to classify documents [5]. By using machine learning, the data obtained can be processed into a solution to solving the problem at hand. However, in machine learning there are various kinds choice of algorithms and methods that can be utilized. Of course, each algorithm has advantages and disadvantages deficiencies of each [6]. One of good algorithm in the text classification part is Random Forest [5]. This algorithm can build decision trees [7]. Decision tree is a tree with children (branch) as a determinant the possibility of each decision in each leaf [8]. Algorithm this not only builds a decision tree, however several decision trees using multiple sample obtained from the dataset. Then, the algorithm will be comparing to the classification results between new decision trees that built to form new classification values.

Thus, this research will classify and categorize the level of reading load from reading material using the K-Means Clustering and Random Forest methods who can classify well because it consists of a combination of several decision trees. While the K-Means Clustering, method is a simple and fast clustering method [9].

2. Research Method

This research was conducted in 5 stages, (1) Data Collection, (2) Selection and Addition, (3) Preprocessing data, (4) Classification using Random Forest, and (5) Evaluation.

2.1 Dataset Collection

The data used is in the form of articles obtained from site https://www.tweentribune.com. This site contains articles are also daily news that has educational themes. Every article in Tweentribune had Lexile Levels and its expected for children and adolescents with various categories, arts, entertainment, culture, national news, and so on. From this site, we will be scrapping with an attribute in the form of article titles, article content and Lexile level of article. Following Table 1, it's is a list of internal attributes dataset and Table 2 are samples from the dataset.

Table 1. List of Attributes in Dataset

Name of Attribute	Explanation of Attribute	Type Data
Title	Article title	text
Content	Article content	text
Lexile Levels	Lexile Level value from article	number

Table 2. Dataset Sample

Title	Content	Lexile Levels
7 things you might not know about San Franciscos cable cars	Cable cars are a symbol for San Francisco but they are also a big part of the city's history ...	1200
Which is worse: a light-colored suit or mom jeans?	Tongues are wagging over President Barack Obamas audacity to wear taupe. The sight of Obama ...	1020

2.2 Selection and Addition

Before classification, data needs to be given class attributes as a load label read from the article. Due to absence data attribute that can be used as an inner class attribute dataset, it is necessary to add new class attributes. The process of adding classes is done by clustering attributes Browse levels into 2 classes, which are difficult clusters and clusters easy as well as clusters with 3 classes namely difficult, easy and normal cluster. The clustering phase also aims to group articles based on load level reading, where the higher the Lexile Levels value, the higher the reading load and vice versa.

The clustering process is done using the K-Means Clustering method. K-Means Clustering Method is an iterative based blocking algorithm group data into the specified cluster K. Results the cluster of the K-Means method is influenced by the initial value centroid given. In each cluster process, if the value is the initial given can give cluster results different.

Here steps to cluster using K-Means method [10]:

1. Determine many cluster points made, represented by the parameter k.
2. Points k will be distributed randomly, and made as the center of a cluster or centroid.
3. All instances are grouped by centroid closest to a certain calculation, generally use the Euclidean distance. Following is the Euclidean distance Equation 1.

$$(x_i, c_j) = \sqrt{\sum_{i=1}^{N} (x_i, c_j)^2} \quad (1)$$

4. Then the centroid of all instances in each cluster is recalculated to become the new cluster center.
5. Repeat process 3-4 until there is no change in centroids.

The results from clustering Lexile Levels attribute will be added as a class attributes in the dataset. Then the Lexile attribute levels are deleted because they have been replaced with class and the Lexile Level attribute is not used in the next process.

2.3 Preprocessing Data

Before the dataset is ready to be used, every classification process needed to do preprocessing steps first. Preprocessing stages the data carried out are (1) Transformation and Weighting, (2) Oversampling (3) Feature Selection.

2.3.1 Transformation and Weighting

Transformation and weighting are the process that can carried out to convert text to index with tokens along with the number of occurrences in each document. Transformation processes begins with Case Folding, then Tokenizing, Stopword removing and Stemming.

After the transformation, the next step is weighting each word by changing the frequency number of occurrences. This process is done by using Term Frequency – Inverse Document Frequency (TF-IDF) methods.

Term frequency (TF) – Inverse document Frequency (IDF) having the abbreviation TF-IDF, it's a method that used in the document to find the weight of a term [11]. The TF-IDF method calculates word weights with use two ways, namely counting frequency appearance of each word in a document (TF) and count (IDF) of documents containing words that is. The Equation 2 used in the TF IDF calculation is [11].

$$\mathrm{TFIDF} = \log\left(1 + f_{ij}\right) * (\log(\frac{td}{td_{wi}})) \tag{2}$$

Where:
1. f_{ij} is the frequency of occurrence of the word i on document j
2. td is the total of all documents in the dataset
3. td_{wi} is the number of documents that contains the word i

2.3.2 Oversampling

Oversampling is a distribution balancing process class by duplicating minority data [12]. This process carried out in a certain way until the distribution of members the class in the dataset is balanced. This process is carried out by using the Synthetic Minority Over Sampling Technique or called SMOTE.

SMOTE is an oversampling method that uses the word synthesis approach or data replication from minority data. SMOTE works by looking for k-nearest neighbors for each data in the minority class, then synthesis data will be created as much as the desire percentage between minority data and k-nearest neighbors specified. This is the steps to measuring nearest neighbor in this dataset is:
1. Determine the value of the k-nearest neighbors.
2. Calculate differences of each attribute between data minority with its k-nearest neighbors.
3. Then this difference multiplied by a random value between 1 and 0.
4. The results are added to the value of a minority data, the results of all these calculations are new synthesis data which will be added to the dataset.

An illustration of how SMOTE works can be seen in Figure 1.

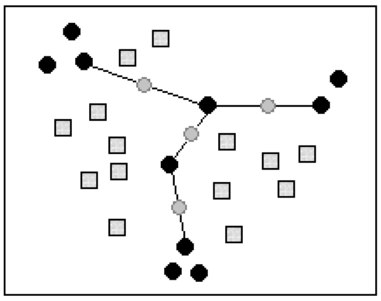

Figure 1. An illustration of SMOTE

In Figure 1, square is a representation class of the majority, then a black circle is represented for minority classes in and synthesis data created by SMOTE and represented by a gray circle.

2.3.3 Feature Selection

Feature Selection is a data processing technique with the aim to sort out the attributes that most influence the determination of data classification. One of method in feature selection is Gini-index Feature Selection.

Gini-index is the method that was first introduced by Breiman in 1984 [13]. This method is commonly used in many decision tree algorithms such as CART, SLIQ and SPRINT. This method can also be applied to the selection of attributes in text classification with scope centroid-based classification [13]. Gini-index Feature Selection method works by calculating the weight of the attribute against the class attribute or label by calculating Gini index from the distribution class. This is the following Equation 3 to measuring Gini index.

$$Gini\ (S) = 1 - \sum_{i=1}^{m} P_i^{\,2} \tag{3}$$

Where:
1. p_i is a probability value of variable "i" to class
2. m is a total number of variables

2.4 Random Forest

Random forest was first introduced by Breiman in 2001. Random Forest Algorithm is an ensemble learning algorithm that builds several classifiers, namely the decision tree building algorithm, then combines the classification results of each classifiers to become a result of the classification of new data points [14].

In the ensemble development process in Random Forest algorithm, there is two different ways, namely: bagging and boosting. For example, manipulating data training to present the decision tree diversity formed. Some other approaches exist that manipulate the input or output target features [7]. Bagging is a method for generating several classifiers and using a collection of classifiers to produce new aggregates [15].

The advantages of Random Forest are [14]:
1. Can handle data training with a large number of instances;
2. Produces better classification than ordinary decision tree building algorithms;
3. Produce a lower error;
4. Has a method for handling lost data effectively; and
5. Efficient because it only requires n sample data as training data and test data.

There are several stages in forming a classification model with Random Forest [16]. These stages are:
1. Bootstrap stage:
   Form data training by taking n samples from data training.
2. Feature Selection stage:
   Create a new dataset with random sample attributes.
3. Decision Tree Formation stage:
   Form a decision tree from data training that has been made before.
4. Pruning stage
   Cut the decision tree to avoid overfitting.
5. Repeat steps '1 to 4' as 'k' times

2.5 Evaluation

After the classification process, the results of the Random Forest algorithm can be tested using the Confusion Matrix method can be seen in Table 3. This method, using a matrix as a medium to provide classification results [17].

Table 3. Confusion Matrix

Correct Classification	Result of Classification	
	+	**-**
+	TP	FN
-	FP	TN

Where:
1. TP (true positive) is a data value "Positive" on actual data and classification data then that it's right (True).
2. TN (true negative) is a data value "Negative" on actual data and classification data then that it's right (True).
3. FP (false positives) and FN (false negative) is a data value "Positive" or "Negative" on the classification data but has the opposite value on the actual data so it's worth the wrong value (False).

From the Confusion Matrix table can be used in evaluating algorithms with three tests, that is:

2.5.1 Accuracy

Accuracy is a value that shows how much percentage the accuracy of the algorithm classification results. Here are the equations used to calculate accuracy. The accuracy can be calculated by Equation 4.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{4}$$

2.5.2 True Positive Rate

True Positive Rate is the value that used to calculate the proportion of classification or True Positive on all data that is positive on the actual class label. Recall can be calculated with Equation 5.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

2.5.3 Precision

Precision is the value that is used as the value of testing the classification algorithm in determining the classification of "True" that matches the actual data. Equation 6 to calculate precision.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

3. Results and Discussion

After doing various stages of data processing, the data is classified using the Random Forest method and followed by testing stage. Testing stage is done by using Cross-Validation method with 10 folds.

3.1 Results

In this process, we compared the results of the Random Forest classification in the data with 2 class of cluster such as (1) without SMOTE and Gini Index Feature Selection, (2) only SMOTE, (3) only Gini Index Feature Selection, and (4) combination of SMOTE and Gini Index Feature Selection. The comparison of class divisions and number of attributes can be seen in Table 4; for accuracy, precision and recall can be seen from Table 5.

Table 4. Comparison of Attribute and Class Distribution of Each Scenario

Scenario	Number of classes 0	Number of classes 1	Number of Attribute
(1)	192 (60%)	127 (40%)	10983
(2)	192 (50%)	192 (50%)	10983
(3)	192 (60%)	127 (40%)	144
(4)	192 (50%)	192 (50%)	353

In Table 4, it can be seen that by doing SMOTE on the dataset and it will flatten the data from the ratio of class 6: 4 to 5: 5 and when using Gini Index Feature Selection can decrease the previous attribute number 10983 to 144 in the dataset without SMOTE and 353 in the dataset with SMOTE.

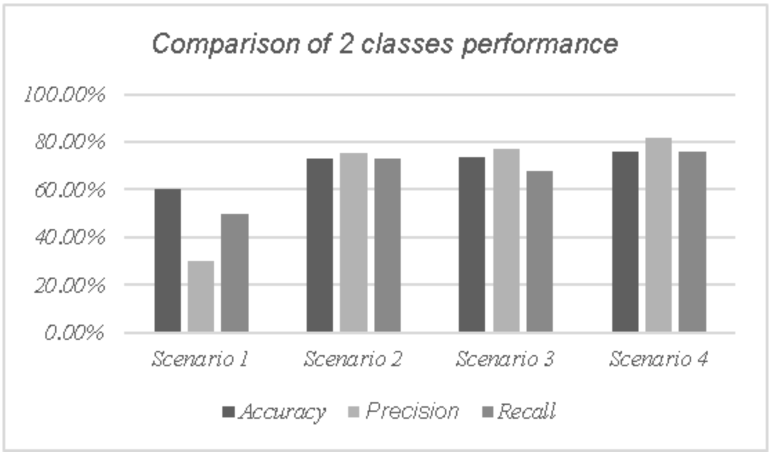

Figure 2. Visualization of Table 5

Table 5. Comparison of 2 Classes Performance

Scenario	Accuracy	Precision	Recall
(1)	60.19%	30.10%	50.00%
(2)	73.14%	75.09%	73.18%
(3)	73.38%	77.35%	67.87%
(4)	76.03%	81.85%	76.05%

In Figure 2 and Table 5, it shows that the random forest classification of data with 2 classes that has been done by SMOTE and Gini Index Feature Selection gives the best results of accuracy, precision and recall of 76.03%.

Then the comparison of the Random Forest classification results data with 3 class clusters such as (5) without the SMOTE and Feature Selection Gini, (6) only SMOTE, (7) only the Gini Index Feature Selection, (8) combination of the SMOTE and Gini Index Feature Selection. By comparing the class of divisions and number of attributes, it can be seen in Table 6, for accuracy, precision and recall it can be seen from Table 7.

Table 6. Comparison of Attribute and Class Distribution of Each Scenario

Preprocessing	Number of classes 0	Number of classes 1	Number of classes 2	Number of Attribute
(5)	147 (46%)	99 (31%)	73 (23%)	10983
(6)	147 (33.3%)	147 (33.3%)	147 (33.3%)	10983
(7)	147 (46%)	99 (31%)	73 (23%)	129
(8)	147 (33.3%)	147 (33.3%)	147 (33.3%)	429

In Figure 3 and Table 6, it can be seen that by doing SMOTE on the dataset it will flatten the data from class 33:33:33 to 46:31:23 and by using the Gini Index Feature Selection it will decrease the number of attributes that were previously 10983 to 129 in the dataset without 429 in the dataset with SMOTE.

Table 7. Comparison of 3 Classes Performance

Preprocessing	Accuracy	Precision	Recall
(5)	46.09%	15.36%	33.33%
(6)	69.60%	73.41%	69.62%
(7)	54.57%	63.27%	46.17%
(8)	65.31%	73.28%	65.30%

In Table 7, shows the result of random forest classification on data with 3 classes that have been done SMOTE without Gini the Feature Selection Index gives the best results of accuracy, precision and recall of 69.60%.

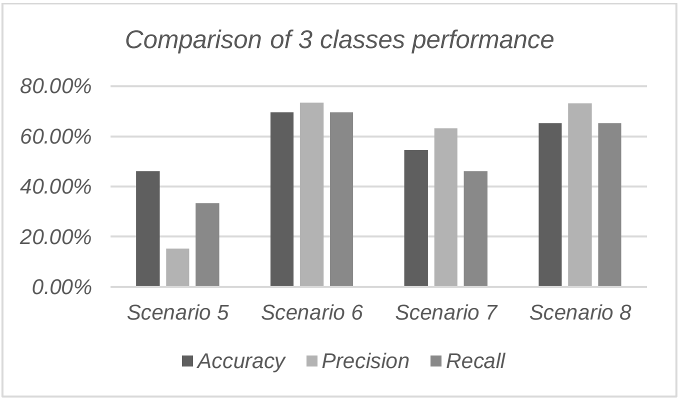

Figure 3. Visualization of Table 6

3.2 Discussion

Based on the comparison table of the classification results in the Table 4, the highest results of accuracy, precision and recall in the dataset with 2 classes are in the data that has been done by SMOTE and carried out by the Gini Index Feature Selection, which is 76.03% accuracy, 81.85% precision, and 76.05% recall. This value has a difference of 15.84% on accuracy, 51.75% at precision and 26.05% on recall from the first experiment dataset without preprocessing.

As a comparison from Table 4, there is Table 6 which contains the classification results of the dataset with 3 classes. The comparison in Table 6 gives the best results of accuracy, precision and recall in the dataset with SMOTE and without the Gini Feature Selection Index, the result is 69.60% of accuracy, 73.41% of precision, and 69.62% of recall. This value has a difference of 23.51% on accuracy, 58.05% on precision and 36.29% on recall from the initial dataset without treatment when preprocessing.

From the results above, it can be concluded that Random Forest with the dataset in this experiment that has been carried out by SMOTE and selected attributes with the Gini Index Feature Selection gives better results than the dataset without preprocessing stage. Meanwhile for the classification results in the two types of datasets with different number classes, the dataset with 2 classes gave the highest results with an accuracy of 76.03% after compared to the dataset of 3 classes are73.41%.

4. Conclusion

Based on the research that has been done, it can be concluded that the Random Forest algorithm can be used to classify the reading load of an article. This is evidenced by the good accuracy value of 76.03% in the dataset with 2 classes and 69.60% in the dataset with 3 classes. In addition, the use of preprocessing techniques also affects the level of performance of the classification algorithm. Especially by using of SMOTE in balancing datasets to unbalanced class distribution. This is indicated by increased in classification accuracy of dataset with 2 classes, the results are 60.19% in scenario 1 becomes 73.18% in scenario 2; and when using classification dataset with 3 classes, the results is 46.09% in scenario 5 becomes 69.60% in scenario 6. The same thing happened when we use attribute selection techniques with the Gini Index Feature Selection method. There is an increase in accuracy when we try to classification of datasets with 2 classes, the result is 60.19% in scenario 1 becomes 67.87% in scenario 3. The best combination of the preprocessing method contains a combination of SMOTE and the Gini Index Feature Selection. The results of testing show that using SMOTE and Gini Feature Selection preprocessing techniques can increase the performance level of the random forest algorithm in the case of the article reading load classification in this study.

References

[1] J. Oakhill, “Children’s difficulties in reading comprehension,” *Educational Psychology Review*, Vol. 5, No. 3, Pp. 223–237, 1993. https://doi.org/10.1007/BF01323045

[2] K. Glasswell and M. P. Ford, “Teaching flexibly with leveled texts: More power for your reading block,” *The Reading Teacher*, Vol. 64, No. 1, Pp. 57–60, 2010. https://doi.org/10.1598/RT.64.1.7

[3] C. Lennon and H. Burdick, “The lexile framework as an approach for reading measurement and success,” *electronic publication on www. lexile. com*, 2004.

[4] M. Awad and R. Khanna, *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. Apress, 2015. https://dx.doi.org/10.1007/978-1-4302-5990-9

[5] S. Yaram, “Machine learning algorithms for document clustering and fraud detection,” in *2016 International Conference on Data Science and Engineering (ICDSE)*, Pp. 1–6, 2016. https://doi.org/10.1109/ICDSE.2016.7823950

[6] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?,” *Journal of Machine Learning Research*, Vol. 15, Pp. 3133–3181, 2014.

[7] L. Breiman, “Random Forests,” *Machine Learning*, Vol. 45, No. 1, Pp. 5–32, Oct. 2001. https://doi.org/10.1023/A:1010933404324

[8] J. R. Quinlan, “Induction of decision trees,” *Mach Learn*, Vol. 1, No. 1, Pp. 81–106, Mar. 1986. https://doi.org/10.1007/BF00116251

[9] B. Wang, “a new clustering algorithm compared with the simple K-Means,” in *2009 International Conference on Management and Service Science*, Pp. 1–5, 2009. https://doi.org/10.1109/ICMSS.2009.5302386

[10] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016. https://doi.org/10.1016/C2009-0-19715-5

[11] S. Robertson, “Understanding inverse document frequency: on theoretical arguments for IDF,” *Journal of Documentation*, Vol. 60, No. 5, Pp. 503–520, Oct. 2004. https://doi.org/10.1108/00220410410560582

[12] Z. K. A. Baizal, M. A. Bijaksana, and A. S. Sastrawan, “Analisis pengaruh metode over sampling dalam churn prediction untuk perusahaan telekomunikasi,” *Jurnal Fakultas Hukum UII*, 2009.

[13] W. Zhu, J. Feng, and Y. Lin, “Using Gini-Index for Feature Selection in Text Categorization,” presented at the 2014 International Conference on Information, Business and Education Technology (ICIBET 2014), 2014. https://dx.doi.org/10.2991/icibet-14.2014.22

[14] A. Van Assche, C. Vens, H. Blockeel, and S. Džeroski, “First order random forests: Learning relational classifiers with complex aggregates,” *Mach Learn*, Vol. 64, No. 1, Pp. 149–182, Sep. 2006.

[15] L. Breiman, “Bagging Predictors,” *Machine Learning*, Vol. 24, No. 2, Pp. 123–140, Aug. 1996. https://doi.org/10.1023/A:1018054314350

[16] U. Pujianto, “Random forest and novel under-sampling strategy for data imbalance in software defect prediction,” *International Journal of Engineering and Technology(UAE)*, Vol. 7, Pp. 39–42, Jan. 2018. http://dx.doi.org/10.14419/ijet.v7i4.15.21368

[17] E. Olivetti, S. Greiner, and P. Avesani, “Statistical independence for the evaluation of classifier-based diagnosis,” *Brain Inf.*, Vol. 2, No. 1, Pp. 13–19, Mar. 2015. https://doi.org/10.1007/s40708-014-0007-6