Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Content analytics based on random forest classification technique: An empirical evaluation using online news dataset

Puteri N. E. Nohuddin [1, *], Wan M. U. Noormanshah [1], Zuraini Zainol [2]

[1] *Institute of IR4.0, National University of Malaysia, Bangi, Malaysia*
[2] *Department of Computer Science, Faculty of Science and Defence Technology, National Defence University of Malaysia, Kuala Lumpur, Malaysia*

ARTICLE INFO

Article history:
Received 21 June 2020
Received in revised form
1 October 2020
Accepted 7 October 2020

Keywords:
Classification
Random forest
Document term matrix
Term frequency–inversed document
Frequency

ABSTRACT

In this paper, a study is established for exploiting a document classification technique for categorizing a set of random online documents. The technique is aimed to assign one or more classes or categories to a document, making it easier to manage and sort. This paper describes an experiment on the proposed method for classifying documents effectively using the decision tree technique. The proposed research framework is a Document Analysis based on the Random Forest Algorithm (DARFA). The proposed framework consists of 5 components, which are (i) Document dataset, (ii) Data Preprocessing, (iii) Document Term Matrix, (iv) Random Forest classification, and (v) Visualization. The proposed classification method can analyze the content of document datasets and classifies documents according to the text content. The proposed framework use algorithms that include TF-IDF and Random Forest algorithm. The outcome of this study benefits as an enhancement to document management procedures like managing documents in daily business operations, consolidating inventory systems, organizing files in databases, and categorizing document folders.

1. Introduction

The invention of many advanced computer technologies allows more people to have more tools to generate and share information like never before. Data growth is easily unnoticeable when most of it happens behind the scenes. Thus, the era of digital data explosion has increased a large volume of data. It is also reported that 80%-90% of future growth data in the form of unstructured text databases that may potentially contain interesting patterns and trends (Zainol et al., 2018). According to Google, they managed about 20 petabytes of data per day, and yet it is still steadily accumulative yearly up to 2018. The size of data has increased up to 2.5 quintillion bytes of data (Dean and Ghemawat, 2008). Nevertheless, one of the big challenges in handling big data is that we are going to process these raw data into interesting and useful information and insight. Data can be categorized in many forms such as structured (e.g., databases), semi-structured (e.g., markup language XML, open standard JSON, NoSQL, etc.), and unstructured (e.g., text files, email, social media data, websites, etc.).

In general, data is processed and cleaned to be analyzed, measure, and visualize as information for a specific purpose. Then, significant information derives valuable and nontrivial knowledge. Knowledge discovery in a database (KDD) is a systematic process of mining interesting patterns and knowledge in a massive dataset. KDD consist of seven (7) main steps, which are: data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation, and knowledge representation (Han et al., 2011). One of the core KDD activities is data mining that performs the extraction of interesting knowledge patterns. Data Mining (DM) embraces several different techniques and algorithms that are attempted to fit as an example of DM techniques can be found in Nohuddin et al. (2018). Regression, Link analysis, and Segmentation (Dean and Ghemawat, 2008). Association rules, clustering prediction, and classification are important techniques in DM. These techniques are divided into two (2) forms: Supervised learning and unsupervised learning. Both types cover functions capable of discovering different hidden patterns in large datasets.

* Corresponding Author.
Email Address: puteri.ivi@ukm.edu.my (P. N. E. Nohuddin)
https://doi.org/10.21833/ijaas.2021.02.011
Corresponding author's ORCID profile:
https://orcid.org/0000-0003-0627-5630
2313-626X/© 2020 The Authors. Published by IASE.

Classification is supervised learning typically used for predicting group membership for data instances. Many researchers applied classification technique such as predicting customer behavior (Caigny et al., 2018; Amin et al., 2019), medical diagnosis (Priyadarshini, 2018; Kumar et al., 2018), education data (Rahayu et al., 2018; Hussain et al., 2018), transportation (Kamarudin et al., 2018; Dabiri and Heaslip, 2018) and many more. Over the last few decades, numerous classification algorithm has been proposed, such as Naïve Bayes, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree, etc. However, all these algorithms or techniques share the same goal which is, to predict accurately the target class for each case in the data and to identify in which category (class) a new data will fall under.

Among sub techniques in classification, the decision tree is a powerful and popular method for classification and prediction (Dunham, 2006). A Decision tree is a flow chart like a 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 strengths of decision tree methods are: Decision trees can generate understandable rules. Decision trees perform classification without requiring much computation. Decision trees can handle both continuous and categorical variables. Decision trees provide a clear indication of which fields are most important for prediction or classification.

This paper describes a document classification experiment using the random forest classification technique. The paper is organized as follows: Section ll discusses the background and related work on document classification, document term matrix, TF-IDF algorithm, and decision tree technique. Section lll elaborates the framework of the Text Analysis Document Classification. Followed by Section lV presents the findings on classifying documents using the proposed framework. Finally, Section V concludes with a summary and future research directions.

2. Background and related work

2.1. Document management and classification

Documents have been digitized starting in the 1980s in line with the growth of computer software technology. Soon after that, many organizations produced these files, and records were generated by software applications, and electronic document management systems (DMS) are developed to organize business files and records digitally. Originally, paper files are first converted to electronic format by scanning. This provides a more demand for compact means of document management and storage, universal access for retrieval, and higher levels of data security and privacy.

An enterprise document or content management software provides several functions like storage, controls digital files that are generated directly through applications such as those in the Microsoft Office applications, accounting software, CAD, email, and so on. The ability to manage documents can be beneficial to organizations. Thus, it becomes an effective process for finding relevant information and for filtering documents straight to users.

Document classification is an application of Machine Learning (ML) that uses both Natural Language Processing (NLP) and text mining (Caigny et al., 2018). The growing number of unstructured digital document collections has attracted many researchers to explore more on document classification techniques (Amin et al., 2019). This technique basically assigns a document into a set of predefined categories based on its content. In this experiment, the research focused on the supervised classification technique, which is capable of: (i) label and train document dataset, (ii) find terms variable of a trained dataset, and (iii) perform document analysis and visualize it.

Eventually, the process of classifying large volumes of documents is necessary to make them more manageable and discover valuable insight from the content analysis. But managing a massive number of documents manually are tedious and inefficient.

Efforts on introducing an automatic document classification come in useful. This is a process driven by Natural Language Processing (NLP), by which algorithms automatically assign one or more categories to the text-based documents such as articles, emails, or survey responses. Integrating the classification process with machine learning is quicker and more practical than manual classification.

Documents are a set of information that can be turned into an electric form and stored into data storage or a computer in the form of one file or more. A file is often treated as individual data items, and a set of it counted as a part of the database. Good document classification is significant for an organization from small to large entities that deals with mountains of data as it may involve various processes such as organization, classification, analyses, knowledge sharing, and process storing (Priyadarshini, 2018). Respectively, maintaining and classifying a large amount of information manually from a variety types of paper-based documents or electronic documents will be time–consuming cost matter in terms of labor (Kumar et al., 2018). In this study, we will focus on the supervised classification technique, which of capable of: (i) labeling and training dataset, (ii) find terms variable of a trained dataset, and (iii) perform document analysis and visualize it. Combination of DTM and TF-IDF to do term count term weighting on each term that appears across documents. Random forest technique will be applied to do classification on the dataset (online news).

Terms Knowledge Discovery in Databases (KDD) and Data Mining (DM) have been used interchangeably to describe the process of extracting useful and meaningful information. KDD is defined as the whole process of discovering useful information and knowledge within data, whereas DM is defined as the tasks within the KDD process where tools and mechanisms are used to identify (mine) the knowledge of interest -KDD models or steps.

2.2. Document term matrix (DTM)

DTM is a two–dimensional matrix table constructed rows representing document vectors and columns contain terms of a corpus. The purpose of generating DTM is to hold frequencies of terms in a corpus in terms of frequency counts (Zhou et al., 2016).

Consider a corpus of documents, and a dictionary of terms contains a bag of words that exist in the document. In the rows of the matrix *i* represents the term to be analyzed, while the column *j* represents the document used in the analysis. Each entry of *(i,j)* represents the count of the term *i*, appears in document *j*. The range of the term count is between 0 and *n*. Therefore, DTM records the terms of similarities between one documents to another document in the same corpus. Also, a significant term of a document can be recognized by analyzed sparse matrix or also called sparsity.

In general, a sparse matrix is an efficient way to represent related information contained in DTM, and it is used whenever many words or cases are encountered (Zainol et al., 2017). In most cases, DTM records the frequency count of the terms as 0. For this case, the sparsity result is used to show the absence of terms. For example, there are 3 simple text documents: i) "I own the blue car," ii) "He is the owner of the blue car," iii) "The blue car is owned by him." In DTM, terms' frequency counts from these three documents are recorded into simple numerical representation by column and row, as shown in Table 1. The table allows a user to view each distinct term from the corpus for further processing and analysis.

Table 1: DTM of 3 simple documents

	I	own	the	blue	car	he	is	owner	of	by	Owned	him
Doc 1	1	1	1	1	1	0	0	0	0	0	0	0
Doc 2	0	0	2	1	1	1	1	1	1	0	0	0
Doc 3	0	0	1	1	1	0	1	0	0	1	1	1

2.3. Term frequency–inversed document frequency (TF-IDF)

TF-IDF is a prominent method to calculate a term's weight across documents in a corpus that is usually applied in information retrieval and text mining. The term weight is calculated based on a formula that produced a statistical measure used to estimate the significance of a term in a corpus (Fortuna et al., 2005). TF is referred to the frequency count for each time the words appear in a document, while IDF is applied to quantify how significant a word or term for the whole documents or corpus. Most search engines such as Google, YouTube, and Facebook applied the TF-IDF for ranking words. Each word/term has its respective TF and IDF score, and the score results simply call as TF*IDF weight of each term. TF-IDF functions with a formula of term *t*, document *d*, and weight Wt,d of term *t*, and document *d* in a corpus are given by with equation (Ramos, 2003).

$$Wt,d = \{TFt,d\, log(N/DFt)\} \qquad (1)$$

where TF*t, d* is the total of incidences for *t* in document *d*; DF*t* is the number of documents that have the term *t*; N represents the total amount of documents in a corpus.

The significant value of the term increases proportionally to the number of terms that appear in the documents. It is functioned to check how relevant the term is throughout documents contained in a corpus (Kalra et al., 2019). TF-IDF gives a notable effect of notable effect in information retrieval to rank the term's results (Zhang et al., 2011).

In this study, we integrate TF-IDF as an absolutely factual method to assess the significance of words by weight dependent on its frequency of occurrence in the document and in its related corpus with a decision tree classification technique to determine the theme grouping of document datasets.

2.4. Random forest algorithm for classification document subject domain

Decision Tree (DT) is one of the classification techniques that uses the branches method to illustrate decision-making in each possible outcome (Felton et al., 2019). Structurally, DT comprises three kinds of nodes that frame an established tree, which a tree required to have 'root node,' 'internal node,' and 'leaf.'

DT breaks down a set of data into smaller and smaller subsets while, at the same time, an associated decision tree is incrementally built. The root node is known as the initial attribute or the topmost decision node in a tree, which corresponds to the best predictor for a tree to make a decision making that has zero incoming and outgoing edges. While internal nodes have both incoming and outgoing edges, at least one followed by a leaf node that has no outgoing edges represents a classification or decision. The deeper the tree, the more complex DT can exist in decision rules and fitter the model. Also, Markel and Bayless (2019) described the complexity of a tree be likely to affect the result of accuracy for a tree to do the decision making. According to Fortuna et al. (2005), they deduce that DT is more efficient as a classification method when it involves decision making, instead of able to compute both categorical and numerical data. Advantages of DT are termed as it is easily accessible and interpreted, it involves less calculation, it is capable of illustrating the relationship between the dependent and independent variables, and it is computationally low end.

Sub technique of DT, random forest is determined as a suitable method for this study. It works as a large collection of decor related decision trees due to it creates a lot of decision trees and uses them as classification branches. It is known as a bagging technique.

The rationale for selecting random forest classification is because of the bagging process in which we use different training models as we try to increase the accuracy of the dataset classification. Bagging is a process to average noisy and unbiased models to create another model with low variance in terms of classification. Random forest corrects some overfitting issues to their training sample in DT. During the classification process, the random forest algorithm builds many decision trees of the original training sample set with a random subset of training sample using random values. Each of the decision trees created has a different value of the original training sample, and with all decision tree, it produces a different variation of the main classification. Ideally, by using many sets of decision trees, we can create a ranking of classified documents.

3. Framework of document analysis based on random forest algorithm

The proposed framework consists of 5 components which are (i) Document dataset, (ii) Data Pre-processing, (iii) Document Term Matrix, (iv) Random Forest Classification, and (v) Visualization. Fig. 1 depicts the proposed framework for Document Analysis based on the Random Forest Algorithm (DARFA). The first stage starts with converting a collection of documents into a corpus. Then the corpus goes through a data preprocessing stage. Data pre-processing embraces activities that include reformatting and cleaning the text data. The next stage is terms extraction for building the DTM. The process of ranking and extracting these keywords is based on the TF-IDF method. Then, based on DTM, keywords are trained and classified using the random forest method and classify according to their group. Finally, the results of the classification are visualized.

3.1. The dataset

Datasets for this experiment consist of a selected news document from the BBC news website. The selected online news pages contain a random mixture of 4 different topic categories. In this experiment, we aim to classify them into 4 categories of documents: education, sport, crime, and marriage. In this experiment, 801 pages of news dataset are collected and will be treated as 801 distinct documents. The dataset collection is manually downloaded from the Kaggle web portal.

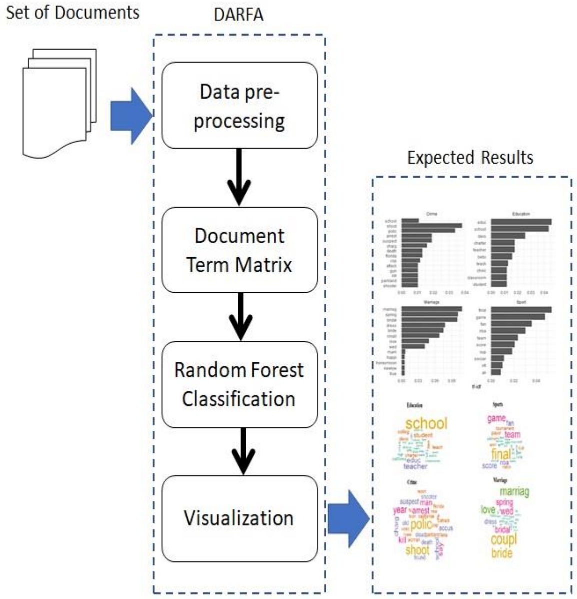


Fig. 1: Framework of document analysis based on random forest algorithm (DARFA)

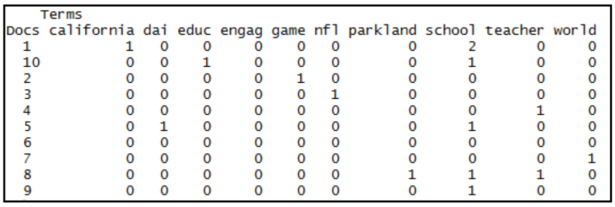
3.2. Data preprocessing

Data Preprocessing is a fundamental stage in data analysis. It is a common technique used to cleaned raw documents to reduce noise, unstructured, and inconsistent data to get a useful and efficient format dataset. In this stage, all documents are combined as a corpus. After that, standardization of text is done by changing terms into a lowercase format and removing symbols and numbers. The next process is the removal of all stop words and white space from the corpus. This is to improve the quality of the datasets in which meaningless terms are pruned from the corpus. The next important process in data cleaning is stemming. Stemming is a process where terms will be trimmed into its root term; for example, term “writing” and “writer” comes from its root word which is “write.” The stemming process is required so that terms with the same root words are converted into its basic terms.

3.3. DTM and text weighting

Term weighting is performed during the text indexing process as it calculates the value of each term related to the documents. In this paper, the method used for term weighting is the TF-IDF method, then construct a DTM. DTM constructs each term that appears through all documents to form a matrix table based on the term weighting. This step is vital for identifying the significant terms across the document and visualizing them so that we able to perform text analysis using the DTM. Furthermore, DTM produces other important information such as non-sparse entries, sparsity percentage, maximal term length, and term weights, which is term frequency that appears in the form of a matrix table.

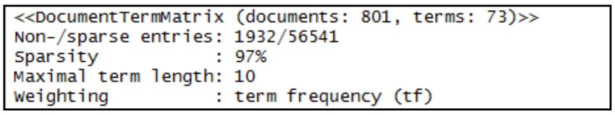
Sparsity percentage or sparse matrix provides information on how much percentage of the terms in the corpus having "0" in the DTM matrix table, which means terms that do no appears in many documents. By removing sparse terms, it is an alternative approach to reduce the dataset's complexity by setting the percentage of sparsity when we want and trim the datasets before calculating the term weights. In this experiment, we set sparse equal to 0.98. It would take effect to remove terms that are missing over 98% of the documents in the corpus, and it would cost the terms at least 2% to appear to be retained in the DTM. Part of rows and columns in DTM is shown in Fig. 2.



```
    Terms
Docs california dai educ engag game nfl parkland school teacher world
  1           1   0    0     0    0   0        0      2       0     0
  10          0   0    1     0    0   0        0      1       0     0
  2           0   0    0     0    1   0        0      0       0     0
  3           0   0    0     0    0   1        0      0       0     0
  4           0   0    0     0    0   0        0      0       1     0
  5           0   1    0     0    0   0        0      1       0     0
  6           0   0    0     0    0   0        0      0       0     0
  7           0   0    0     0    0   0        0      0       0     1
  8           0   0    0     0    0   0        1      1       1     0
  9           0   0    0     0    0   0        0      1       0     0
```

Fig. 2: Inspection result of document term matrix for analyses news datasets

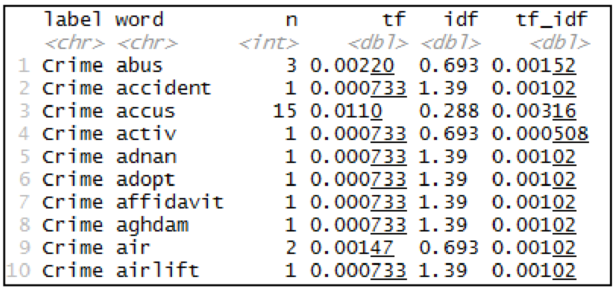
Multiple experiments have been conducted for attaining the precise number of terms that generate less complexity to the model. If the terms are too few, it tends to produce a high error percentage of document classification, and if the number of terms is too large, it takes a longer time for the model to be trained. Thus, it is important to set effective sparsity in DTM for the training of the document classification model. The process can be repetitively done until the fittest sparsity percentage is reached. An example of a DTM summary can be viewed in Fig. 3.



```
<<DocumentTermMatrix (documents: 801, terms: 73)>>
Non-/sparse entries: 1932/56541
Sparsity           : 97%
Maximal term length: 10
Weighting          : term frequency (tf)
```

Fig. 3: Inspection result and summary of the document term matrix

In the process of constructing DTM, firstly, distinctive terms that appear in each document are gathered in the TF-IDF list to create a keyword set for each document. Second, the TF-IDF score is calculated for each term in the document, and all the terms are sorted according to their scores. Then, the process is to build features set for representation, for example, in Fig. 4, in which the top 10 terms are portrayed. The keyword set for the entire document collection is created by uniting each document's preserved distinct phrases. Finally, using the built keyword set, the term-document variable is produced for each document in the DTM. The aim of this process is to measure the importance of a term about a topic in a set of documents.



```
   label word           n       tf   idf    tf_idf
   <chr> <chr>      <int>    <dbl> <dbl>     <dbl>
 1 Crime abus           3 0.00220  0.693 0.00152
 2 Crime accident       1 0.000733 1.39  0.00102
 3 Crime accus         15 0.0110   0.288 0.00316
 4 Crime activ          1 0.000733 0.693 0.000508
 5 Crime adnan          1 0.000733 1.39  0.00102
 6 Crime adopt          1 0.000733 1.39  0.00102
 7 Crime affidavit      1 0.000733 1.39  0.00102
 8 Crime aghdam         1 0.000733 1.39  0.00102
 9 Crime air            2 0.00147  0.693 0.00102
10 Crime airlift        1 0.000733 1.39  0.00102
```

Fig. 4: Example of term weighting using TF, IDF, and TF–IDF

3.4. Classification using random forest

Basically, when a term has relatively high frequent counts over documents, the term becomes more significant as a variable factor to a certain topic. Thus, it is a crucial process to discover the terms variable so that the classification model can use the variable as a reference for future classification of the same dataset. The set of terms variable will be an output for the further stage. Next, a chain of terms is constructed to characterize each class theme for classifying the dataset with the same attribute content. As an exemplar, with a 30% Sports news dataset, the generated chain of terms includes chain terms like "tournament," "final," "game," "fans," "score," "player," "team," "win," "match" and "NBA" as shown in Fig. 5.

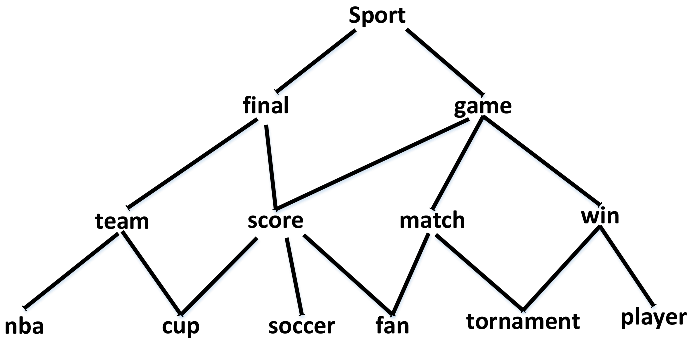


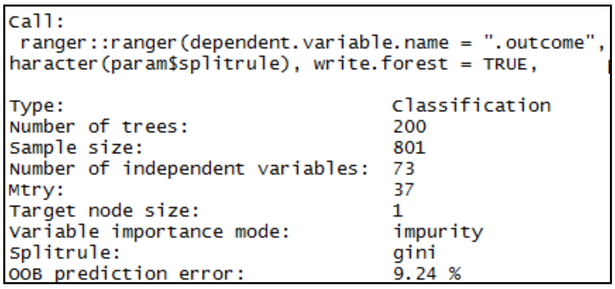
Fig. 5: Chain of terms for sports news dataset

The output of the weighted terms from TF-IDF results is placed into a random forest model. Based on the chain of terms variable, the model will start to create a random subset of training sample with random value accordingly, and each random subset will not have the same value as the other random subset. From each random subset, the model will generate a decision tree, and in this experiment, we set it with a total number of trees=200. The higher the number of decision tree created it increases the accuracy of the news classification. However, this may lead to higher processing time, and some application platform cannot support the process.

In the later stage, the news documents are tested into the model. The model loads the dataset into each decision tree in the random forest model. Each decision tree-labels the document into the respective class theme of the document based on the chain of terms variable. The labeling result of the classification tree may be different from one another as each decision tree comes from a different random subset. However, the final classification of the document will be count based on the most voted group, the decision tree labeled.

4. Experimental results

In this section, the outcomes of the experiment are presented and discussed. As shown in Fig. 6, the classification model produces the out-of-bag (OOB) prediction error rate, which indicates how many the terms are missing during the classification of the dataset. The lower the OOB error rate, the better and more accurate the classification process. In this experiment, the OOB prediction rate of the proposed classification model is 9.24%. The model generates 200 decision trees by using the set of chains produced by 73 term variables, *mtry* are 37 trials, and variable of importance mode is Gini impurity. The dataset consists of 801 online news documents is expected to be classified into four different class themes. All four categories have their own ID and can be easily differentiated. The evaluation has two stages, which are (i) training and (ii) testing. During the training stage, 30% (i.e., 241 documents) of the dataset is used for training the classification model. 30% of the dataset are extracted randomly to build the classifier. Then, the other 540 documents are used as the testing dataset to test the classifier. In Fig. 7, the graph plots present the results of classification and sub-classification based 4 class themes. The top 10 highest terms for each group of the testing dataset are selected for the graph plot. The illustration of document classification is labeled into 4 main classes (i) "Crime," (ii) "Education," (iii) "Marriage," and (iv) "Sport." For example, in the "Sport" group, the most frequent terms are "Final," "game," "fan," and "nba." The horizontal graphs show that the top frequent terms appeared for each news that related to the "Sport" section. By using the chained term as terms variable, the new set of datasets related to the sports section will be automatically classified into the "Sport" major group.



```
Call:
 ranger::ranger(dependent.variable.name = ".outcome",
haracter(param$splitrule), write.forest = TRUE,

Type:                             Classification
Number of trees:                  200
Sample size:                      801
Number of independent variables:  73
Mtry:                             37
Target node size:                 1
Variable importance mode:         impurity
Splitrule:                        gini
OOB prediction error:             9.24 %
```

Fig. 6: Final model classification details

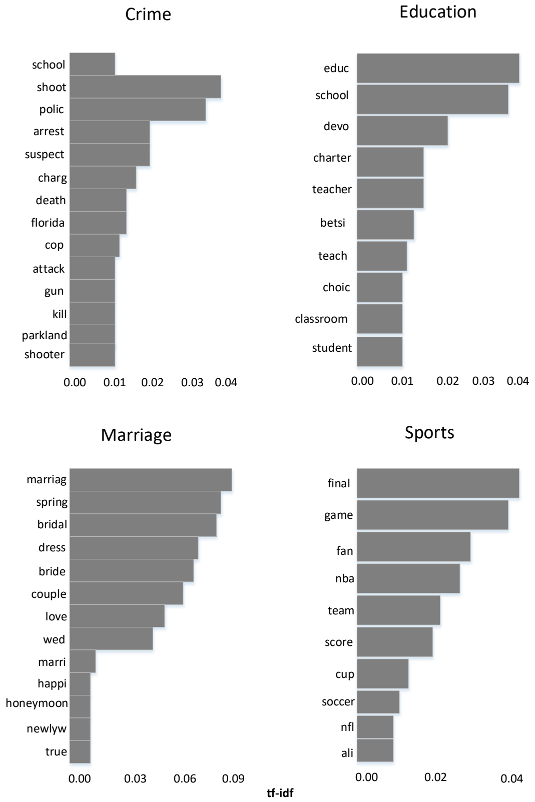


Fig. 7: Graph plot of terms classified into 4 major groups

The proposed classification model is considered as a fundamental method to group a stack of multi-themed documents and not limited to news dataset only. As described in the previous section, every partial (30%) new dataset is trained, then new term variables are stowed in the classification model as the classifier. Inevitably, the rest of that dataset is loaded for generating classification graphs, as shown in Fig. 7. Finally, a word cloud visualization of each classified term based on document content is presented in Fig. 8. The word cloud deliberately illustrates the document content of the news dataset. The bigger size of terms in the word cloud is deemed as frequent terms in the classified document datasets.

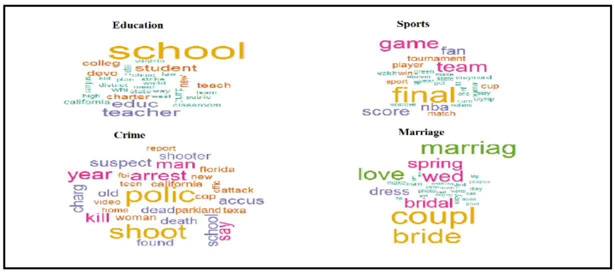


Fig. 8: Word cloud for grouped dataset after classified into 4 major group

5. Conclusion and future work

In conclusion, the objective of this study is to develop a classification model for classifying documents as effectively compared to some related works reviewed in Section 2. Moreover, this proposed technique delivers an effective visualization of classified word clouds and bar graph plots for users to view the outcomes of document content analysis. Moreover, the constructed model can analyze the document content. In future work, this study will extend the proposed framework of Document Analysis based on the Random Forest Algorithm (DARFA) into a prototype application for classifying documents by using the model template with a user interface. The results of this study can be an added value business application as an improvement for daily operation in business, for example, managing inventory system, organizing database, and administering massive document storage.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Amin A, Al-Obeidat F, Shah B, Adnan A, Loo J, and Anwar S (2019). Customer churn prediction in telecommunication industry using data certainty. Journal of Business Research, 94: 290-301. https://doi.org/10.1016/j.jbusres.2018.03.003

Caigny AD, Coussement K, and De Bock KW (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. European Journal of Operational Research, 269(2): 760-772. https://doi.org/10.1016/j.ejor.2018.02.009

Dabiri S and Heaslip K (2018). Inferring transportation modes from GPS trajectories using a convolutional neural network. Transportation Research Part C: Emerging Technologies, 86: 360-371. https://doi.org/10.1016/j.trc.2017.11.021

Dean J and Ghemawat S (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1): 107-113. https://doi.org/10.1145/1327452.1327492

Dunham MH (2006). Data mining: Introductory and advanced topics. Pearson Education India, Bengaluru, India.

Felton BR, O'Neil GL, Robertson MM, Fitch GM, and Goodall JL (2019). Using random forest classification and nationally available geospatial data to screen for wetlands over large geographic regions. Water, 11(6): 1158. https://doi.org/10.3390/w11061158

Fortuna B, Grobelnik M, and Mladenic D (2005). Visualization of text document corpus. Informatica, 29(4): 497–502.

Han J, Pei J, and Kamber M (2011). Data mining: Concepts and techniques. Elsevier, Amsterdam, Netherlands.

Hussain S, Dahan NA, Ba-Alwib FM, and Ribata N (2018). Educational data mining and analysis of students' academic performance using WEKA. Indonesian Journal of Electrical Engineering and Computer Science, 9(2): 447-459. https://doi.org/10.11591/ijeecs.v9.i2.pp447-459

Kalra S, Li L, and Tizhoosh HR (2019). Automatic classification of pathology reports using TF-IDF Features. arXiv:1903.07406. Available online at: https://arxiv.org/abs/1903.07406

Kamarudin ND, Rahayu SB, Zainol Z, Rusli MS, and Ghani KA (2018). Performance comparison of machine learning classifiers on aircraft databases. Defence Science and Technology Technical Bulletin, 11(2): 154-169.

Kumar PM, Lokesh S, Varatharajan R, Babu GC, and Parthasarathy P (2018). Cloud and IoT based disease prediction and diagnosis system for healthcare using Fuzzy neural classifier. Future Generation Computer Systems, 86: 527-534. https://doi.org/10.1016/j.future.2018.04.036

Markel J and Bayless AJ (2019). Performance of random forest machine learning algorithms in binary supernovae classification. arXiv:1907.00088. Available online at: https://arxiv.org/abs/1907.00088

Nohuddin P, Zainol Z, Lee ASH, Nordin I, and Yusoff Z (2018). A case study in knowledge acquisition for logistic cargo distribution data mining framework. International Journal of Advanced and Applied Sciences, 5(1): 8-14. https://doi.org/10.21833/ijaas.2018.01.002

Priyadarshini MG (2018). Decision tree algorithms for diagnosis of cardiac disease treatment. International Journal of Computer Science and Mobile Computing, 7(7): 138-144.

Rahayu SB, Kamarudin ND, and Zainol Z (2018). Case study of UPNM students performance classification algorithms. Journal Engineering and Technology, 7(4.31): 285-289.

Ramos J (2003). Using TF-IDF to determine word relevance in document queries. In the First instructional Conference on Machine Learning, New Jersey, USA, 242: 133-142.

Zainol Z, Jaymes MT, and Nohuddin PN (2018). VisualUrText: A text analytics tool for unstructured textual data. Journal of Physics: Conference Series, 1018: 012011. https://doi.org/10.1088/1742-6596/1018/1/012011

Zainol Z, Nohuddin PN, Mohd TA, and Zakaria O (2017). Text analytics of unstructured textual data: A study on military peacekeeping document using R text mining package. In the International Conference on Computing and Informatics, Kuala Lumpur, Malaysia: 1-7.

Zhang W, Yoshida T, and Tang X (2011). A comparative study of TF* IDF, LSI and multi-words for text classification. Expert Systems with Applications, 38(3): 2758-2765. https://doi.org/10.1016/j.eswa.2010.08.066

Zhou M, Padilla OHM, and Scott JG (2016). Priors for random count matrices derived from a family of negative binomial processes. Journal of the American Statistical Association, 111(515): 1144-1156. https://doi.org/10.1080/01621459.2015.1075407