A Novel Term Weighting Scheme for Imbalanced Text Classification

Tanapon Tantisripreecha[1] and Nuanwan Soonthornphisaj[2*]
E-mail: tanapon.tan@mahidol.ac.th, nuanwan.s@ku.th
[1]Department of Mathematics, Faculty of Science, Mahidol University, Bangkok, Thailand
[2]Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand

Keywords: text classification, imbalance problem, term weighting schemes, TFIDF, SVM, logistic regression

Received: April 29, 2021

High dimensional feature is the main problem of text domain. If imbalance class is also found in the context, the classifier's performance is worsen. Moreover, solving imbalance problem by oversampling method in this circumstance is very difficult to get performance improvement. In this paper, a new term weighting scheme is proposed by combining Term frequency with an average of inverse document frequency factor. We denoted our scheme by TFmeanIDF. Our proposed method has high potential for imbalance text domain with high dimension. No feature selection or oversampling method is required. Extensive comparison results on 7 datasets validate the advantages of TFmeanIDF in terms of F_1 score obtained from widely used base classifiers, such as Logistic regression and Support Vector Machines. We found that F_1 score of minority class is higher than that of baseline term weighting schemes. Using TFmeanIDF as a term weighting shows promising result for logistics regression and support vector machines.

Povzetek: Avtorji so razvili novo shemo uteževanja termov pri neuravnoteženi klasifikaciji besedil.

1 Introduction

Learning from extreme imbalanced data is a challenging problem in real-world applications. We know that machine learning algorithms struggle with accuracy because of the unequal distribution of classes. This causes the performance of existing classifiers to get biased towards majority class. In this paper, we focus our attention on finding a new term weighting scheme that show promising result on imbalance text classification.

In text classification problem, we found that most well-known classifiers give high accuracy with balanced dataset. However, when apply to an imbalanced dataset, accuracy cannot be used to evaluate the model since the model often predicts as a majority class and rarely predict the minority class. Although the classifier predicts minority class incorrectly, the accuracy is still high because there are few instances of minority class.

Customer feedback on e-commerce platform such as Amazon is an example of imbalanced data set. The negative feedback is important for product owners, hence we consider these feedbacks as the positive class. Although the rating scales are provided for users, most customers do not rate their products. Therefore, the customer feedback in terms of textual data is our main focus. We found that the textual feedback of Amazon products is highly imbalanced. Most reviews are rated as positive feedback and is considered as a majority class whereas the negative feedback is considered as a minority class. The imbalance ratios of Amazon product reviews are quite high (vary from 5:1 to 15:1).

The rest of this paper is structured as follows: Section 2 reviews various studies on imbalance text classification problem; Section 3 presents our propose term weighting scheme; Section 4 presents the experimental results for the classification using various term weighting schemes and Section 5 provides the discussion and conclusions, as well as the directions for future work.

2 Related works

Given a document, a term weighting scheme is normally applied to represent the numerical text vector. Term frequency (TF) is a traditional term weighting technique for text vectorization that considers the frequency of the term found in the document [1]. Moreover, using TF may give a large weight to common terms that would weak the text discriminating ability and leads to adverse impact on the classification performance [2]. To alleviate this problem, a collection frequency factor IDF (inverse document frequency) is introduced into the TF scheme called TFIDF [1]. Considering IDF value, the more frequent documents containing the term, the lower the IDF score. This means that the word is less important, when they occur in several documents. The less a term occurs in different documents, the more weight it is given. We found that the assumption of TF is that multiple

* Corresponding author

occurrences of a term in a text are more important than single occurrences. Secondly, IDF assumes that rare terms are more significant than frequent terms. Thirdly, the normalization assumption which reduces the bias of the document length. We found that TFIDF is considered as a baseline term weighting method in text classification domain [3]

In order to reduce the effect of term with high local weighting factor TF, meanwhile the importance of term with low DF can be further enhanced. Tang, et al [4] replaced the IDF factor with a novel function (or new global weighting factor) that possesses the following two properties: (i) when the DF of the term changes, the value of the function has a larger attenuation rate, (ii) it should be a bounded function. Since the attenuation rate of exponential function is faster than the logarithmic function so inverse exponential frequency is chosen to construct the new function. Their term weighting scheme is called TFIEF. Another unsupervised term weighting method is called TFIHF that incorporate the nonlinear transformation functions. The hyperbolic function contributes as a global weighting factor namely inverse hyperbolic frequency [5].

To eliminate the influence of document length, Chen, et al. [2] proposed a new term frequency factor called term density function to normalize the TF. Their experiments showed promising result on 4 real world datasets which are Amazon (vocabulary size = 10,000 words), Movie Review (vocabulary size = 7,103 words), WebKB (vocabulary size = 8,791 words), and 20 Newsgroup (vocabulary size = 10,000 words). We found that the vocabulary size of these datasets is quite small (less than 20,746 words) and no imbalance problem found in these domains.

For imbalance text classification problem, the study of supervised term weighting schemes is also found in the literature. Lan et al [6] introduced the weighting scheme called TFRF. Their assumption is that if a high-frequency term is more concentrated in the positive class than in the negative class, it will make more contributions in selecting the positive samples from the negative samples. This idea of favouring positive terms can also be observed in [7], which tried to select such "positive" words using the square root of chi-square, as opposed to "negative" words indicative of negative documents.

We found that several term weighting schemes work well on balanced dataset, however, when applied to an imbalanced dataset, they gave high precision but low recall. Various methods to handle imbalance problem can be classified in one of the following categories: under sampling, oversampling, synthetic data generation and cost sensitive learning. Under sampling method randomly reduces the number of majority class instances to make the dataset balanced. This method is best to use when the data set is huge and reducing the number of training samples helps to improve run time and storage troubles.

Oversampling method randomly replicates the minority class instances to balance the data. An advantage of oversampling is that there is no information loss. However, the disadvantage of using this method is that it leads to overfitting problem.

Synthetic data generation method adds the minority class instances by generating artificial data. The well-known method namely SMOTE algorithm synthesis new instances based on feature space by using K nearest neighbours.

The concept of cost sensitive learning is to evaluate the cost associated with misclassifying instances [8]. This method does not create balanced data distribution. It deploys the cost matrices which represent the misclassification cost, cost of False Negative and cost of False Positive under the constraint that cost of False Negative is higher than that of False Positive. The goal of this method is to choose a classifier with lowest total cost.

3 Proposed term weight

To tackle the imbalanced text classification problem, we introduce a new supervised term weighting scheme namely *TFmeanIDF* that use class information of training documents. *TF* stands for term frequency whereas the *meanIDF* is calculated from the average of inverse document frequency of instances in majority and minority class. For each term, *i*, in Customer Feedback, *j*, *meanIDF* can be calculated using equation (1).

$$meanIDF_i = \frac{\left(\log\left(\frac{n_{major}}{df_{major,i}+1}\right)+1\right)+\left(\log\left(\frac{n_{minor}}{df_{minor,i}+1}\right)+1\right)}{2} \quad (1)$$

where

n_{major} = total number of customer feedbacks in majority class.

n_{minor} = total number of customer feedbacks in minority class.

$df_{major,i}$ = total number of customer feedbacks in majority class that contain term *i*.

$df_{minor,i}$ = total number of customer feedbacks in minority class that contain term *i*.

Finally, the feature of customer feedbacks, *j*, is obtained from the term frequency, tf_{ij} multiplied by the global weight, meanIDF, using the following equation.

$$TFmeanIDF = tf_{ij} * \frac{\left(\log\left(\frac{n_{major}}{df_{major,i}+1}\right)+1\right)+\left(\log\left(\frac{n_{minor}}{df_{minor,i}+1}\right)+1\right)}{2} \quad (2)$$

Where tf_{ij} = total number of occurrences of term *i* in customer feedbacks, *j*

Figure 1 illustrates the changing value of the traditional IDF and meanIDF when the number of documents in minority and majority class containing term *i* is varied. The X-axis represents the ratio of document frequency in minority to that of majority class. We found that the meanIDF reaches its minimum value when the document frequency ratio is equal to 1. That means the number of documents in minority class with the occurrence of term i is equal to the number of documents of majority class in which term *i* occur. It means that term, *i*, is less important.

From Figure 1, we found that the value of IDF is stable that means it has no feature selection power in text imbalance problem. The value of meanIDF is dynamic

when the document frequency ratio is changed. Therefore, incorporate the meanIDF as term weighting can increase the weight of important term in minority class and majority class as well. When the document frequency of term, i, likely occurs in both minority and majority class, that term becomes less important.

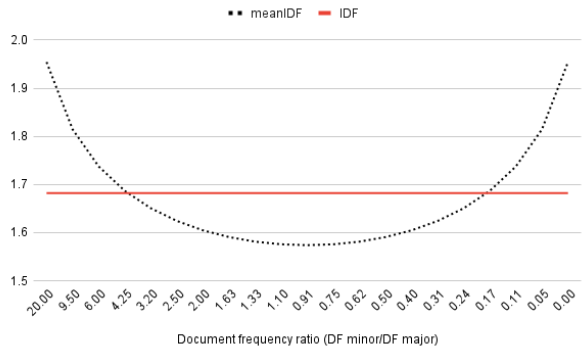

Figure 1: The behaviour of meanIDF and IDF under different document frequency ratio.

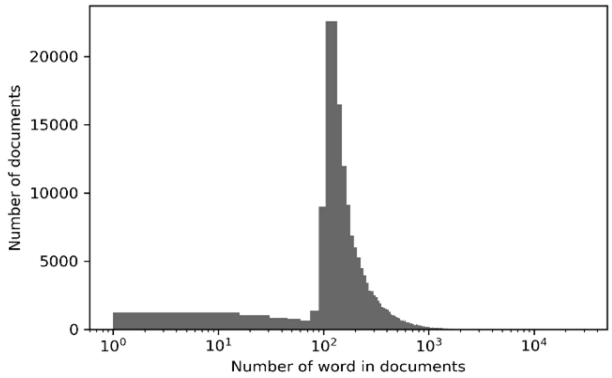

Figure 2: Frequency distribution of word occurrences found in customer feedback.

Product Category	Number of instances		Vocabulary Size	Imbalance Ratio (IR)
	Majority Class (Positive feedback)	Minority Class (Negative feedback)		
Instant Video	523,653	60,280	135,996	8.69
Musical Instruments	442,627	57,549	203,638	7.69
Digital Music	786,273	49,733	268,783	15.81
Baby	571,064	97,210	133,321	5.87
Patio Lawn and Garden	816,370	177,120	178,671	4.61
Automotive	1,187,496	186,272	214,752	6.38
Apps for Android	2,209,990	428,183	305,499	5.16

Table 1: Datasets used in our experiments.

4 Experiments and results

4.1 Dataset

Amazon customer feedback dataset obtained from seven product categories [9] are collected to validate the performance of our proposed method. The raw data consists of the product review text and the rating scales that represent the satisfaction levels. We consider the scales value 1-2 as the negative feedback whereas the scale 3-5 means positive feedback. Table 1 shows class distribution and vocabulary size of seven product categories which are Instant Video, Musical Instruments, Digital Music, Baby, Patio Lawn and Garden, Automotive and Apps for Android.

Note that the imbalance ratio (IR) is defined as follows:

$$\text{IR} = \frac{\#\text{instances in majority class}}{\#\text{instances in minority class}} \tag{3}$$

We found that all product categories are imbalanced. The highest imbalance ratio is found in Digital Music product with IR = 15.81. The lowest imbalance ratio is 4.61 found in Patio Lawn and Garden.

Amazon product feedback consists of different text lengths as shown in Figure 2. We visualize the histogram of text length in log scale (x-axis) It reveals that the minimum text length of product review is 1 (10^0) word and the maximum length is 30,000 ($3x10^4$) words. The average of text length is 360 ($3.6x10^2$) words and the median is 200 ($2x10^2$) words.

We do text preprocessing by removing HTML tags, converting text to lowercase, replacing punctuation with spaces, removing numbers, removing non English text and removing duplicate characters. Then tokenization is performed to separate the sentence into word tokens. Finally input vectors are prepared by text vectorization with different term weighting methods which are TFmeanIDF, TFIDF, TFRF, TFIEF and TFIHF using equation 2, 4, 5, 6 and 7, respectively.

$$TFIDF = tf_{ij} * \log\left(\frac{N}{df_i + 1}\right) + 1 \tag{4}$$

$$TFRF = tf_{ij} * \log_2\left(2 + \frac{A_{ij}}{\max(1, C_{ij})}\right) \tag{5}$$

$$TFIEF = tf_{ij} * e^{-\frac{df_i}{N}} \tag{6}$$

$$TFIHF = tf_{ij} * \frac{\frac{N}{df_i}}{1 + (\frac{N}{df_i})} \tag{7}$$

Where tf_{ij} = total number of occurrences of term i in j

df_i = total number of customer feedbacks containing term i.

A_i = total number of customer feedbacks containing term *i* in the positive class.

C_i = total number of customer feedbacks containing term *i* in the negative class.

4.2 Experimental results

We compare the performance of our term weighting method with several baseline approaches. In particular, supervised term weighting methods that considered the class label as a domain knowledge, and unsupervised term weighting methods. Three classifiers are experimented based on 5-fold cross validation. The selected classifiers are Logistic Regression, Support Vector Machine and multinomial Naïve Bayes. These classifiers are chosen since they are state of the art in sentiment analysis research [5, 10, 11, 12, 13, 14]. Therefore it is more challenging to see the potential of term weighting schemes in imbalanced text classification domain to see clearly whether *TFmeanIDF* can increase the performance or not.

Note that given the training data, the Logistic Regression algorithm calculates the event's probabilities and applies a logistic function to create the model [15]. Support Vector Machine (SVM) creates a decision boundary or hyperplane to predict the classes in high dimensional space [16]. Multinomial Naïve Bayes (MultinomialNB) creates the model using joint probabilistic distribution [17].

Product Category	TF mean IDF	TF Mean IDF+ over Sampling	TF IDF	TF IDF+ over Sampling	TF RF	TF IEF	TF IHF
Logistic Regression							
Instant Video	**0.69 (0.94)**	0.65 (0.90)	0.64 (0.93)	0.63 (0.89)	0.03 (0.89)	0.64 (0.93)	0.64 (0.93)
Musical Instruments	**0.69 (0.93)**	0.66 (0.89)	0.64 (0.93)	0.64 (0.89)	0.01 (0.88)	0.64 (0.92)	0.63 (0.92)
Digital Music	**0.61 (0.96)**	0.54 (0.91)	0.55 (0.95)	0.52 (0.90)	0.01 (0.94)	0.56 (0.95)	0.55 (0.95)
Baby	**0.72 (0.92)**	0.69 (0.88)	0.68 (0.91)	0.67 (0.87)	0.01 (0.85)	0.64 (0.92)	0.66 (0.91)
Patio Lawn and Garden	**0.74 (0.91)**	0.73 (0.88)	0.7 (0.90)	0.7 (0.87)	0.02 (0.82)	0.68 (0.9)	0.68 (0.9)
Automotive	**0.7 (0.92)**	0.67 (0.88)	0.65 (0.91)	0.65 (0.87)	0.01 (0.86)	0.64 (0.91)	0.64 (0.91)
Apps for Android	**0.74 (0.91)**	0.72 (0.88)	0.7 (0.91)	0.69 (0.87)	0.2 (0.85)	0.69 (0.91)	0.69 (0.91)
SVM							
Instant Video	**0.69 (0.941)**	0.65 (0.90)	0.65 (0.97)	0.62 (0.89)	0.07 (0.9)	0.62 (0.92)	0.62 (0.92)
Musical Instruments	**0.69 (0.93)**	0.65 (0.90)	0.65 (0.92)	0.63 (0.89)	0.02 (0.88)	0.62 (0.91)	0.62 (0.91)
Digital Music	**0.62 (0.96)**	0.54 (0.92)	0.57 (0.95)	0.51 (0.91)	0.01 (0.94)	0.54 (0.95)	0.54 (0.95)
Baby	**0.72 (0.92)**	0.69 (0.88)	0.68 (0.91)	0.67 (0.87)	0.01 (0.85)	0.62 (0.91)	0.65 (0.90)
Patio Lawn and Garden	**0.74 (0.91)**	0.72 (0.88)	0.7 (0.90)	0.69 (0.86)	0.05 (0.82)	0.67 (0.894)	0.67 (0.89)
Automotive	**0.7 (0.92)**	0.67 (0.88)	0.65 (0.91)	0.64 (0.83)	0.02 (0.86)	0.62 (0.91)	0.62 (0.91)
Apps for Android	**0.74 (0.91)**	0.71 (0.88)	0.69 (0.91)	0.68 (0.87)	0.39 (0.86)	0.69 (0.90)	0.69 (0.97)
Multinomial NB							
Instant Video	0.07 (0.9)	0.57 (0.86)	0.09 (0.90)	0.56 (0.86)	0.5 (0.92)	**0.62 (0.92)**	**0.62 (0.92)**
Musical Instruments	0.01 (0.88)	0.55 (0.84)	0.02 (0.88)	0.55 (0.84)	0.5 (0.91)	**0.61 (0.90)**	**0.61 (0.91)**
Digital Music	0.01 (0.94)	0.42 (0.86)	0.01 (0.94)	0.41 (0.86)	0.3 (0.94)	**0.5 (0.94)**	**0.5 (0.94)**
Baby	0.15 (0.86)	0.6 (0.83)	0.19 (0.86)	0.59 (0.83)	0.5 (0.88)	0.61 (0.90)	**0.62 (0.88)**
Patio Lawn and Garden	0.19 (0.83)	0.66 (0.84)	0.24 (0.84)	0.64 (0.83)	0.55 (0.87)	**0.67 (0.88)**	**0.67 (0.88)**
Automotive	0.07 (0.86)	0.61 (0.85)	0.12 (0.87)	0.6 (0.84)	0.43 (0.89)	**0.63 (0.90)**	**0.63 (0.90)**
Apps for Android	0.35 (0.86)	0.65 (0.84)	0.38 (0.87)	0.63 (0.84)	0.63 (0.88)	**0.67 (0.89)**	**0.67 (0.89)**

Table 2: F_1-score of minority class and (Micro-average) of various term weighting methods.

The performance of each model is evaluated using F_1-score as shown in equation 8.

$$F_1 = \frac{2*precision*recall}{precision+recall} \tag{8}$$

Where

$$precision = \frac{TP}{TP+FP} \tag{9}$$

$$recall = \frac{TP}{TP+FN} \tag{10}$$

Note that:

TP represents the number of customer feedbacks that are correctly classified as minority class.

FP represents the number of customer feedbacks that are incorrectly classified as minority class.

FN represents the number of customer feedbacks belonging to minority class but are incorrectly classified in the majority class.

Three experimental set ups are explored as follow:

A) Binary classification.

B) Multi-class classification.

C) Long text binary classification.

A) Binary classification

For binary classification, the class label is positive feedback and negative feedback.

The experimental results depicted in Table 2 confirm that the *TFmeanIDF* outperforms all selected supervised and unsupervised term weighting methods. For Logistic Regression and SVM, we found that *TFmeanIDF* outperforms other term weighting methods in all product categories. *TFmeanIDF* obtains the highest F_1 score at 0.74 on minority class of App for Android products.

Consider the highest imbalance ratio dataset which is Digital Music Product category (imbalance ratio =15.81), we found that using *TFmeanIDF* also gets the highest F_1 score at 0.61 and 0.69 on minority class for Logistic Regression and SVM respectively.

To validate whether the oversampling method can increase F_1 score of the baseline method (TFIDF), we found that random oversampling method has no contribution in this problem domain. Since the feature space (vocabulary size) is very high (vary from 133,321 to 305,499; see Table 1), no performance improvement obtained from the datasets.

The average of F1 score of minority classes in 7 product categories is illustrated in Figure 3. We found that Logistic Regression and SVM learned from TFmeanIDF gets the highest performance with the average F1 score at 0.70. Moreover, no performance loss on majority class since the micro-average of F1 score is 0.93 which is the highest among other term weighting schemes (See Figure 4 for details).

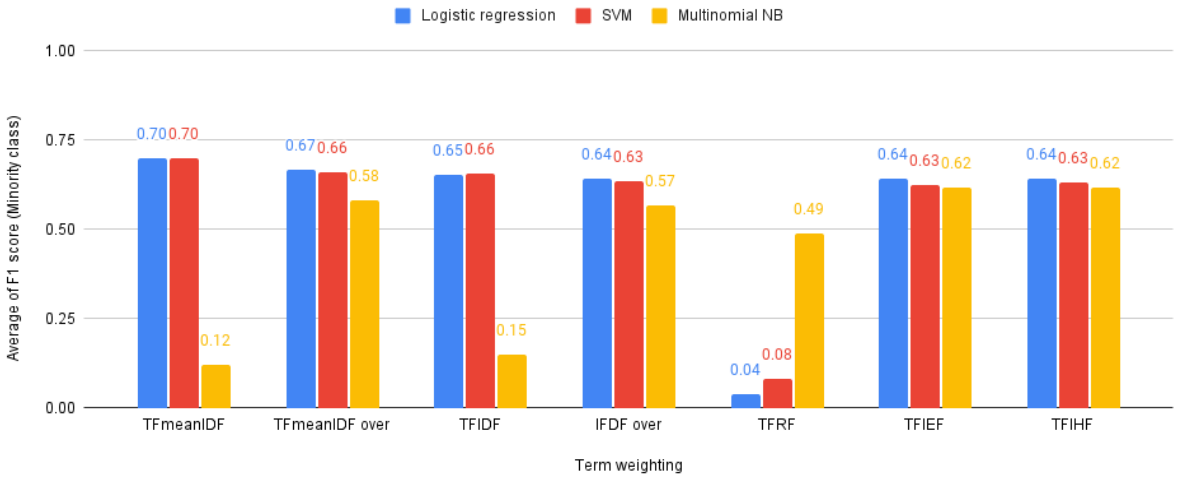

Figure 3: The average of F_1 score on Minority class from 7 products categories (Experiment A: binary classification).

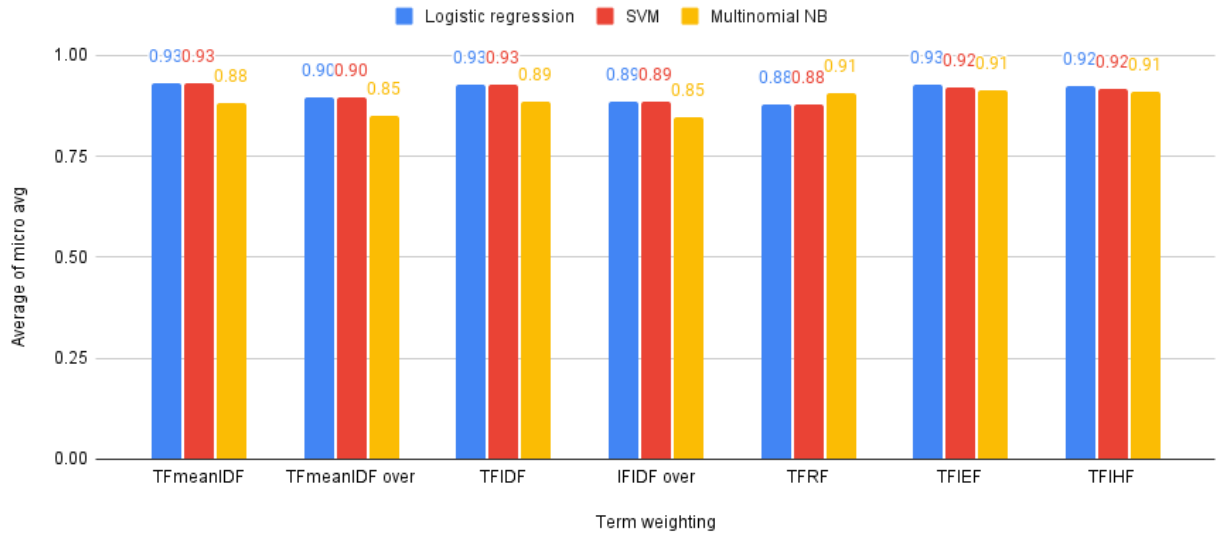

Figure 4: The average performance of 7 product categories measured in term of Micro average (F_1 score) (Experiment A: binary classification).

B) Multi-class classification

To see the potential of *TFmeanIDF* on multi-class classification, we consider the rating scales obtained from customers as the class labels. The range of satisfaction level starts from 1 to 5. (see Table 3).

The performance of multi-class classification with imbalance problem is normally measured in terms of the weighted average recall and weighted average F_1 as shown in equation (11).

$$Weight\_average\_recall = \frac{\sum_{i}^{n}(recall_i \times |c_i|)}{\sum_{i}^{n}|c_i|} \quad (11)$$

$$Weight\_average\_F1 = \frac{\sum_{i}^{n}(F1_i \times |c_i|)}{\sum_{i}^{n}|c_i|} \quad (12)$$

Where c_i is class i

n is number of classes

The result of multi-class classification shown in Table 4 confirms that the proposed term weighting method significantly outperforms all selected baselines supervised and unsupervised term weighting methods.

For Logistic Regression and SVM, we found that TFmeanIDF outperforms other term weighting methods in all product categories. TFmeanIDF obtains the highest

Product Category	Number of instances in C_i				
	1	2	3	4	5
Instant Video	36,785	23,495	79,349	137,840	306,465
Musical Instruments	34,931	22,618	38,537	93,306	310,784
Digital Music	29,988	19,745	41,344	122,479	622,450
Baby	58,913	38,297	12,557	94,749	463,758
Patio Lawn and Garden	119,633	57,487	80,891	174,356	561,123
Automotive	122,160	64,112	103,857	230,293	853,346
Apps for Android	294,284	133,899	253,549	561,831	1,394,611

Table 4: Data distribution for multi-class classification. problem.

weighted recall and weighted F1-score score at 0.78 and 0.73 respectively on the Digital Music.

In multi-class classification, the average of weight recall and weight F1-score illustrated in Figure 5 and 6 respectively. We found that Logistic Regression and SVM learned from TFmeanIDF gets the highest performance with the average weighted recall score at 0.71 and 0.70 respectively. Moreover, no performance loss on majority class since the weight-average of F_1score is 0.66 and 0.65 respectively which is the highest as well (See Figure 5).

C) Long text classification

To further explore the potential of *TFmeanIDF*, we set up another experiment for long text classification problem. This problem is the most difficult since we select only the long text as a test set by including customer feedbacks that contain more than 5000 words [18]. Note that imbalance

Product Category	TF mean IDF	TF Mean IDF+ over Sampling	TF IDF	TF IDF+ over Sampling	TF RF	TF IEF	TF IHF
Logistic Regression							
Instant Video	**0.73 (0.68)**	0.63 (0.66)	0.72 (0.67)	0.61 (0.64)	0.71 (0.66)	0.71 (0.66)	0.66 (0.52)
Musical Instruments	**0.70 (0.65)**	0.62 (0.64)	0.69 (0.63)	0.59 (0.62)	0.68 (0.63)	0.68 (0.63)	0.62 (0.48)
Digital Music	**0.78 (0.73)**	0.65 (0.68)	0.77 (0.72)	0.63 (0.67)	0.77 (0.72)	0.77 (0.72)	0.74 (0.64)
Baby	**0.70 (0.66)**	0.65 (0.66)	0.68 (0.64)	0.62 (0.64)	0.68 (0.63)	0.68 (0.63)	0.58 (0.43)
Patio Lawn and Garden	**0.68 (0.64)**	0.62 (0.63)	0.67 (0.61)	0.58 (0.61)	0.66 (0.61)	0.66 (0.61)	0.56 (0.41)
Automotive	**0.71 (0.66)**	0.66 (0.66)	0.70 (0.64)	0.68 (0.64)	0.69 (0.64)	0.69 (0.64)	0.62 (0.48)
Apps for Android	**0.64 (0.59)**	0.63 (0.58)	0.63 (0.58)	0.63 (0.58)	0.54 (0.39)	0.54 (0.39)	0.65 (0.61)
SVM							
Instant Video	**0.72 (0.68)**	0.62 (0.64)	0.71 (0.66)	0.6 (0.62)	0.69 (0.65)	0.69 (0.65)	0.65 (0.52)
Musical Instruments	**0.7 (0.65)**	0.6 (0.62)	0.68 (0.63)	0.57 (0.6)	0.66 (0.62)	0.66 (0.62)	0.62 (0.48)
Digital Music	**0.78 (0.73)**	0.61 (0.66)	0.77 (0.72)	0.59 (0.64)	0.75 (0.71)	0.75 (0.71)	0.74 (0.64)
Baby	**0.69 (0.66)**	0.63 (0.65)	0.68 (0.63)	0.6 (0.62)	0.66 (0.62)	0.66 (0.62)	0.58 (0.43)
Patio Lawn and Garden	**0.68 (0.63)**	0.6 (0.62)	0.67 (0.61)	0.56 (0.59)	0.65 (0.6)	0.65 (0.6)	0.56 (0.41)
Automotive	**0.71 (0.66)**	0.66 (0.66)	0.7 (0.64)	0.59 (0.6)	0.68 (0.63)	0.68 (0.63)	0.62 (0.48)
Apps for Android	**0.63 (0.56)**	0.63 (0.58)	0.62 (0.54)	0.62 (0.54)	0.53 (0.37)	0.53 (0.37)	0.64 (0.58)
Multinomial NB							
Instant Video	0.66 (0.53)	0.59 (0.63)	0.66 (0.53)	0.58 (0.61)	0.69 (0.65)	**0.69 (0.65)**	0.68 (0.58)
Musical Instruments	0.62 (0.48)	0.53 (0.57)	0.62 (0.48)	0.53 (0.56)	0.66 (0.61)	**0.66 (0.61)**	0.64 (0.52)
Digital Music	0.74 (0.64)	0.58 (0.63)	0.74 (0.64)	0.57 (0.62)	0.75 (0.71)	**0.75 (0.71)**	0.75 (0.66)
Baby	0.59 (0.46)	0.57 (0.6)	0.6 (0.46)	0.56 (0.58)	0.65 (0.62)	**0.65 (0.62)**	0.61 (0.5)
Patio Lawn and Garden	0.58 (0.45)	0.55 (0.57)	0.59 (0.46)	0.53 (0.56)	0.64 (0.6)	**0.64 (0.6)**	0.61 (0.5)
Automotive	0.63 (0.49)	0.63 (0.5)	0.63 (0.5)	0.62 (0.58)	0.67 (0.64)	**0.67 (0.64)**	0.64 (0.53)
Apps for Android	0.58 (0.46)	0.62 (0.58)	0.62 (0.58)	0.62 (0.58)	0.59 (0.5)	**0.59 (0.5)**	0.57 (0.45)

Table 3: Performance on Multi-class problem.

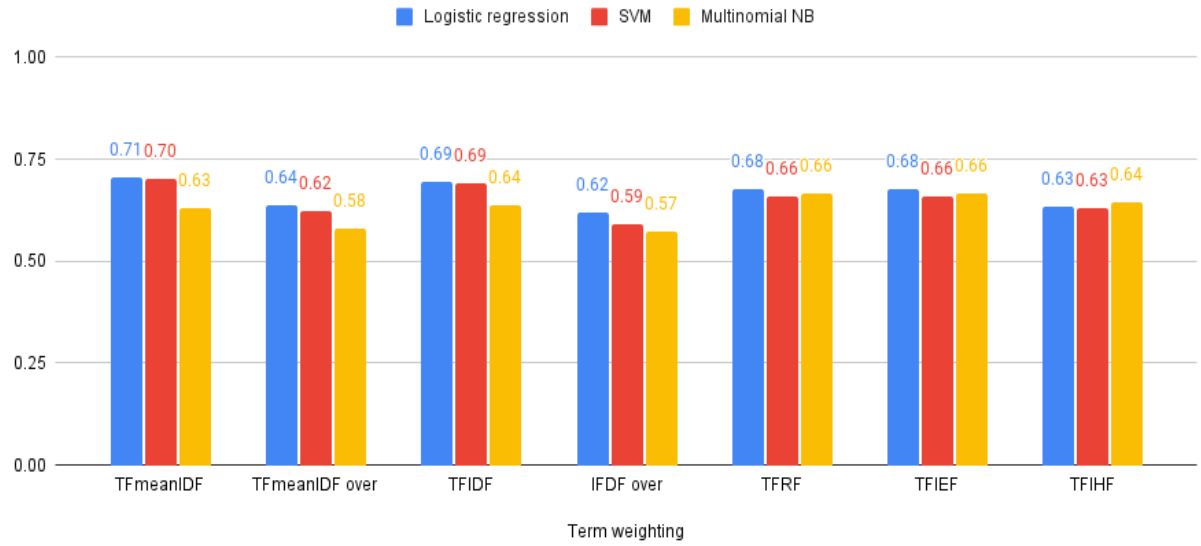

Figure 5: The average of weighted drecall obtained form 7 product categories (Experiment B: multi-class classification).

Product Category	Number of instances		Vocabulary Size	Imbalance Ratio (IR)
	Majority Class (Positive feedback)	Minority Class (Negative feedback)		
Instant Video	366,785	42,180	135,996	8.70
Musical Instruments	310,316	40,235	203,638	7.71
Digital Music	550,924	34,827	268,783	15.82
Baby	547,782	93,231	133,321	5.88
Patio Lawn and Garden	571,593	124,106	178,671	4.61
Automotive	831,179	130,595	214,752	6.36
Apps for Android	1,546,589	300,185	305,499	5.15

Table 6: The data distribution of long text feedbacks.

problem is presented in this experiment as well. The data distribution is shown in Table5. The performances of minority class for long text classification shown in Table 6 and Figure 8 reveal that SVM and Logistic Regression learning from *TFmeanIDF* are as good as those learning from TFIDF with the average F_1 score at 0.91. However, the average of F_1 score of minority class (Figure 7) obtain from 7 product categories is slightly dropped. SVM learning from *TFmeanIDF* gets the micro-average of F_1score at 0.60 where as SVM learning from TFIDF get the average of F_1 score at 0.62 on minority classes.

We found that TFRF cannot classify minority class instances in long text at all. This is the limitation of TFRF since it is supervised approach that needs the label information. As shown in equation [7], TFRF requires the value of number of customer feedbacks containing $term_i$ and the number of customer feedbacks containing $term_i$ (A_i, C_i). In this scenario, TFIDF is applied to create the test vectors [6]. Moreover this dataset is imbalance, the model learning from TFRF get the worst performance.

Product Category	TF mean IDF	TF Mean IDF+ over Sampling	TF IDF	TF IDF+ over Sampling	TF RF	TF IEF	TF IHF
Logistic Regression							
Instant Video	0.64 (0.89)	0.65 (0.82)	**0.7 (0.91)**	0.65 (0.83)	0 (0.81)	0.67 (0.89)	0.66 (0.88)
Musical Instruments	0.49 (0.96)	0.37 (0.88)	**0.53 (0.96)**	0.43 (0.91)	0 (0.94)	0.53 (0.94)	0.52 (0.93)
Digital Music	0.48 (0.97)	0.46 (0.93)	**0.53 (0.97)**	0.45 (0.94)	0 (0.97)	0.54 (0.97)	0.57 (0.97)
Baby	0.42 (0.88)	**0.51 (0.8)**	0.42 (0.88)	0.48 (0.79)	0 (0.87)	0.57 (0.87)	0.57 (0.87)
Patio Lawn and Garden	0.47 (0.87)	0.61 (0.85)	0.48 (0.87)	**0.63 (0.86)**	0 (0.85)	0.62 (0.89)	0.63 (0.89)
Automotive	**0.71 (0.93)**	0.57 (0.83)	0.69 (0.93)	0.51 (0.79)	0 (0.87)	0.75 (0.93)	0.73 (0.92)
Apps for Android	**0.77 (0.87)**	0.67 (0.72)	0.76 (0.87)	0.64 (0.7)	0.2 (0.7)	0.71 (0.83)	0.7 (0.83)
SVM							
Instant Video	0.65 (0.89)	0.65 (0.86)	0.68 (0.9)	**0.7 (0.89)**	0 (0.81)	0.56 (0.87)	0.56 (0.87)
Musical Instruments	**0.59 (0.96)**	0.42 (0.91)	0.56 (0.96)	0.43 (0.92)	0.14 (0.94)	0.41 (0.91)	0.37 (0.91)
Digital Music	0.5 (0.97)	0.45 (0.95)	0.57 (0.97)	0.49 (0.95)	0 (0.97)	**0.58 (0.97)**	0.56 (0.97)
Baby	0.47 (0.88)	0.49 (0.82)	0.52 (0.88)	0.52 (0.84)	0 (0.87)	0.55 (0.86)	**0.57 (0.86)**
Patio Lawn and Garden	0.45 (0.87)	**0.61 (0.87)**	0.51 (0.88)	0.59 (0.86)	0 (0.85)	0.59 (0.88)	0.58 (0.88)
Automotive	**0.81 (0.96)**	0.65 (0.88)	0.79 (0.95)	0.6 (0.86)	0 (0.87)	0.7 (0.91)	0.71 (0.91)
Apps for Android	**0.73 (0.85)**	0.65 (0.7)	0.71 (0.85)	0.59 (0.68)	0.6 (0.77)	0.67 (0.79)	0.68 (0.79)
Multinomial NB							
Instant Video	0 (0.8)	0.59 (0.76)	0 (0.8)	**0.82 (0.57)**	0.48 (0.64)	0.55 (0.75)	0.55 (0.75)
Musical Instruments	0 (0.94)	0.24 (0.86)	0.07 (0.94)	0.28 (0.88)	0.31 (0.89)	**0.34 (0.91)**	0.34 (0.91)
Digital Music	0 (0.97)	0.33 (0.89)	0 (0.97)	0.3 (0.88)	0.55 (0.96)	**0.55 (0.97)**	0.52 (0.96)
Baby	0.23 (0.87)	**0.43 (0.69)**	0.31 (0.87)	0.38 (0.68)	0.35 (0.69)	0.36 (0.7)	0.36 (0.7)
Patio Lawn and Garden	0.18 (0.86)	**0.5 (0.78)**	0.3 (0.87)	0.47 (0.78)	0.48 (0.8)	0.45 (0.79)	0.46 (0.79)
Automotive	0.29 (0.89)	0.39 (0.64)	0.43 (0.91)	0.37 (0.61)	**0.56 (0.82)**	0.5 (0.77)	0.49 (0.77)
Apps for Android	0.54 (0.77)	0.6 (0.62)	0.59 (0.79)	**0.61 (0.64)**	0.52 (0.45)	**0.61 (0.64)**	0.6 (0.64)

Table 5: Long text classification performance measured in terms of F_1-score of minority class and (micro-average of F_1 score).

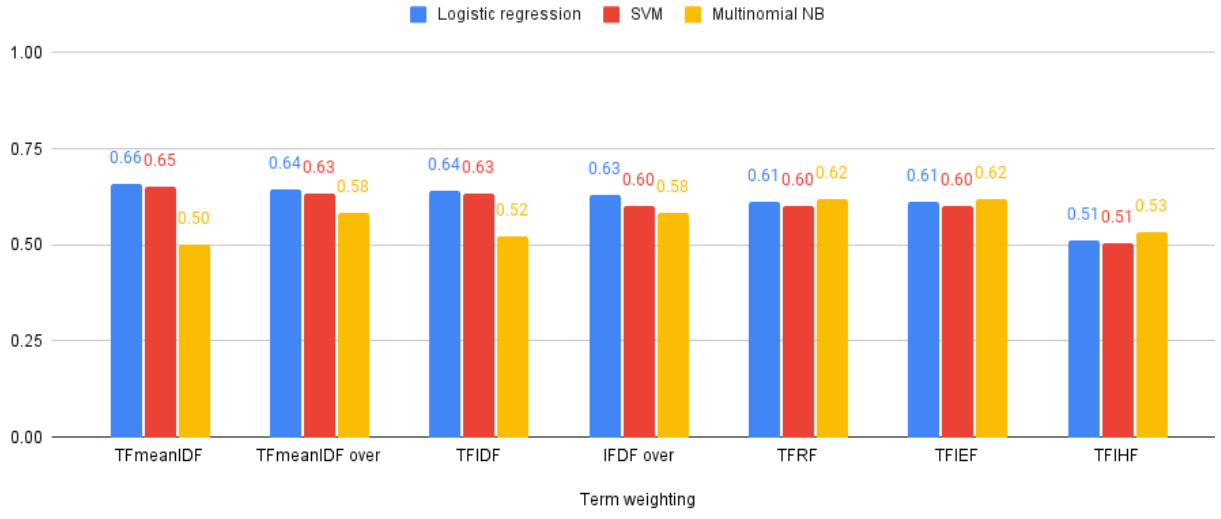

Figure 6: Weighted average of F1 score on multi-class classification problem (Experiment B: multi-class classification).

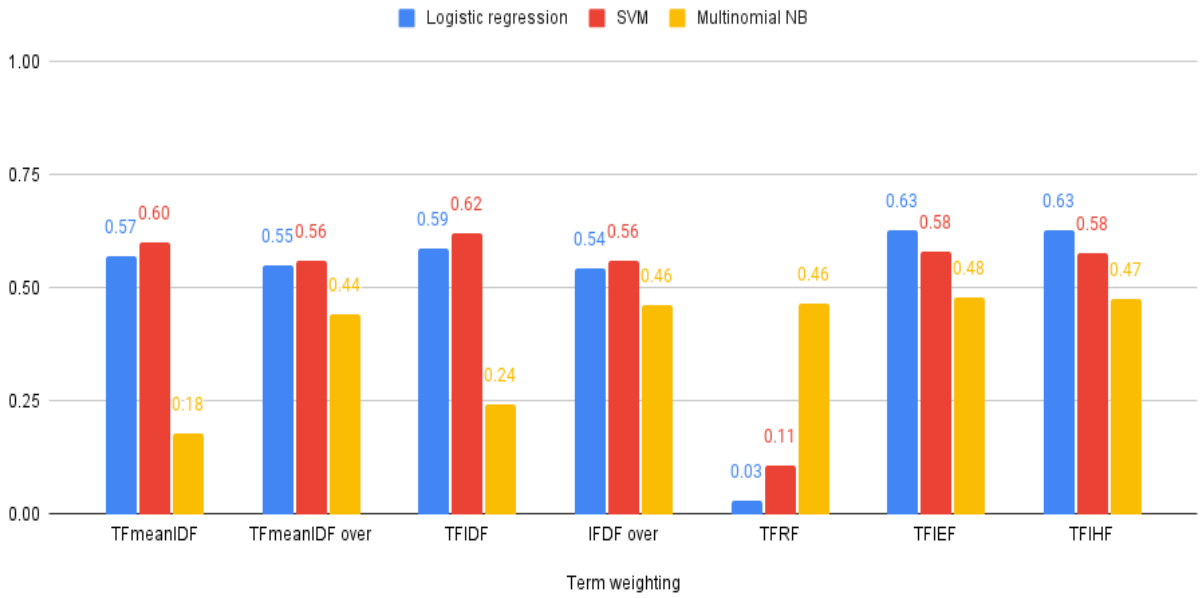

Figure 7: F1 score of minority class obtained from 7 product categories (Experiment C).

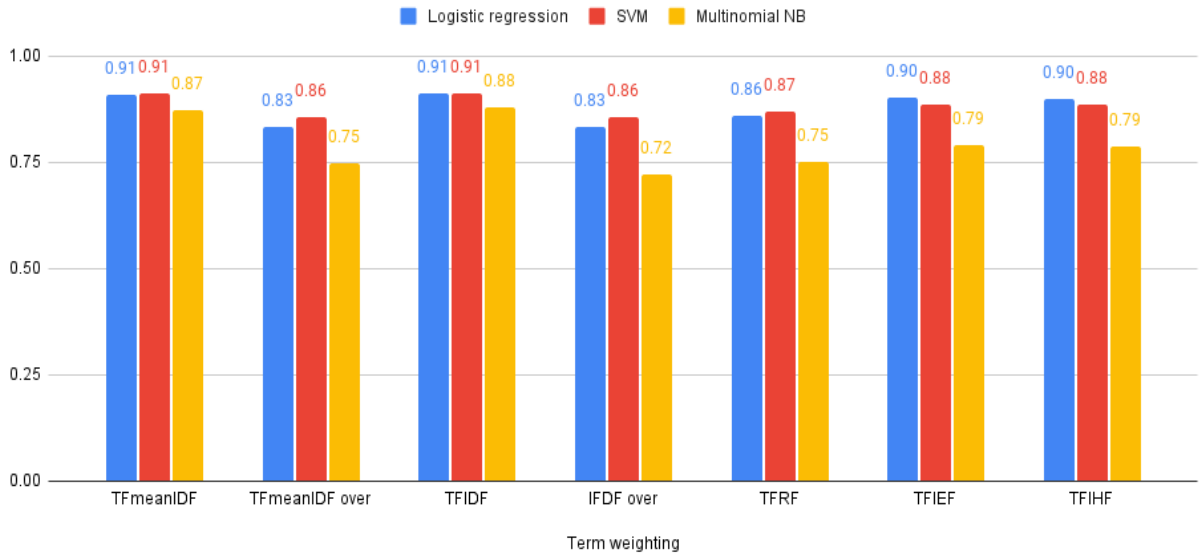

Figure 8: Comparision of Micro average of F1 s core obtained from various term weighting methods (Experiment C Long text classification problem).

We do statistical analysis to compare the performance of classifiers using different term weighting approaches on 3 classification problems. Table 7 represents the p-value obtained from Friedman test [19]. As shown in Table 7, all p-value are less than 0.05. Therefore, the classification performance of all classifiers are statistically significant.

Classification problem	Logistic Regression	SVM	NB	ALL
Experiment A	$2.98x10^{-06}$	$2.60x10^{-06}$	$3.58x10^{-21}$	$3.58x10^{-21}$
Experiment B	$5.10x10^{-06}$	$1.14x10^{-05}$	$3.20x10^{-05}$	$1.90x10^{-18}$
Experiment C	4.87x10-05	0.35x10-4	1.04x10-3	6.15x10-13

Table 7: The p-value from Friedman test.

5 Conclusion

This paper proposed a new term weighting scheme namely *TFmeanIDF* that can enhance the performance of SVM and Logistic Regression on imbalance text classification problem. The value of *TFmeanIDF* reflects the document frequency in minority class and majority class so that the word features that frequently found in minority class is more important than those found in majority class.

TFmeanIDF can be expected to increase the text classification performance for text documents with long length, since the proposed term weight method is computed by TF multiple by meanIDF. The term TF represents the frequency of words in a document. In addition, meanIDF illustrates the average of IDF in majority and minority class. When *TFmeanIDF* are computed, the length of the document is not biased to the output values, since meanIDF is normalized with the log scale as the same as the IDF term. Therefore, *TFmeanIDF* is able to represent the importance of each word in a document. The value of *TFmeanIDF* value is increase if the term frequently occur in the document. We found that *TFmeanIDF* can represent the proper term weighting when the term appears only in the majority or minority classes. However, the term weight will be small if the word occurs in both minority and majority class which is different from traditional TFIDF term weighting method that the weight is equal if the ratio is same.

To evaluate *TFmeanIDF*, three well-known algorithms which are Logistic Regression, SVM and multinomialNB are experimented on seven Amazon product datasets to compare the performance. The experimental result shows that *TFmeanIDF* works well on imbalance problem with high dimension. Our proposed term weighting scheme gets promising result for Logistic Regression and SVM compared to other term weighting schemes. Moreover, *TFmeanIDF* has low complexity. Future studies may consider to incorporate other domain knowledges, such as ontology, into the term weighting scheme to handle the imbalanced text problem.

Acknowledgement

This work was supported by Mahidol University, Research Fund under Grant No. A4/2563, Faculty of Science, Mahidol University, and Centre of Excellence in Mathematics, CHE, Thailand.

References

[1] Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. Vol. 24 (5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

[2] Chen, L., Jiang, L. & Li, C. (2021). Using modified term frequency to improve term weighting for text classification, Engineering Applications of Artificial Intelligence. https://doi.org/10.1016/j.engappai.2021.104215

[3] Samant, S., Murthy, N., Malapati, A. (2021). Categorization of event clusters from twitter using term weighting schemes. Informatica, (45) 405–414. https://doi.org/10.31449/inf.v45i3.3063

[4] Tang, Z., Li, W. & Li, Y. (2020). An improved term weighing scheme for text classification. Concurrency Computation Practice and Expert, Vol. 32, 1-19. https://doi.org/10.1002/cpe.5604

[5] Tang, Z., Li, W., Li, Y., Zhao, W. & Li, S. (2020). Several alternative term weighting methods for text representation and classification. Knowledge-Based Systems, Vol. 207, 1-14. https://doi.org/10.1016/j.knosys.2020.106399

[6] Lan, M., Tan, C.L., Su, J. & Lu.Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31(4), 721-735. https://doi.org/10.1109/TPAMI.2008.110

[7] Ng, H.T., Goh, W.B. & Low, K.L. (1997). Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization, Proc. SIGIR '97, 67-73. https://doi.org/10.1145/278459.258537

[8] López, V., Fernández, A., Moreno-Torres, J.G.& Herrera, F. (2012). Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Expert System with Applications, Vol. 39, 6585–6608. https://doi.org/10.1016/j.eswa.2011.12.043

[9] Ruining H. & Julian M. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. Proc of Int. World Wide Web Conference Committee (IW3C2), Canada. http://dx.doi.org/10.1145/2872427.2883037

[10] Ying L., Han T.L. & Aixin S. (2009). Imbalanced text classification: A term weighting approach. Expert System with Applications, 690-701. https://doi.org/10.1016/j.eswa.2007.10.042

[11] Abhilasha S.R., Amit A. & Preeti D. (2018). Comparative Study of Machine Learning Approaches for Amazon Reviews. International Conference on Computational Intelligence and Data Science (ICCIDS 2018), 1552-1561. https://doi.org/ 10.1016/j.procs.2018.05.119

[12] Kaur S. & Rajni Mohana, R. (2018). Prediction of Sentiment from Macaronic Reviews. Informatica. (42), 127–136.

[13] Tiwari, P., Pandey, H.M., Khamparia, A. & Kumar, S. (2019). Twitter-based opinion mining for flight service utilizing machine learning. Informatica 43 (2019) 381–386 https://doi.org/10.31449/inf.v43i3.2615

[14] Yayla, R. (2021). Determining of the User Attitudes on Mobile Security Programs with Machine Learning Methods. Informatica 393–403 https://doi.org/10.31449/inf.v45i3.3506

[15] Mark S., Nicolas L.R., & Francis B. (2017). Finite Sums with the Stochastic Average Gradient. Mathematical Programming, Springer Verlag, 83-112.

[16] John C. P. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized

Likelihood Methods. Advance in Large Margin Classifiers, 61-74.

[17] Manning, C.D., Raghavan, P., & Schuetze, H. (2008) Introduction to Information Retrieval. Cambridge University Press, 234-265.

[18] Wan, L., Papageorgiou, G., Seddon, M. & Bernardoni, M. (2019). Long-length Legal Document Classification. https://doi.org/10.13140/RG.2.2.36657.12646

[19] Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc, 32, 675–701.