심리학 석사 학위논문

A Comparison of Oversampling Effects on Imbalanced Topic Classification of Korean Texts

한국어 주제 분류에서 오버 샘플링 효과 비교

2017 년 8 월

서울대학교 대학원

협동과정 인지과학전공

서 이 레

Abstract

A Comparison of Oversampling Effects on Imbalanced Topic Classification of Korean Texts

Yirey Suh

Program in Cognitive Science

The Graduate School

Seoul National University

Imbalanced data is a widely-acknowledged problem in supervised learning classification tasks. Oversampling is one way to overcome the problem and there are many methods of oversampling that have been discovered. While researches on the effect of oversampling on other languages have been widely conducted, studies comparing oversampling methods on Korean texts are scarce. This study compares the effect of oversampling methods on the task of classifying Korean internet news articles. This study finds that support vector machines (SVM) and logistic regression reacted with

stability and performed best when paired with borderline-SMOTE2 in imbalanced conditions.

Keywords: Imbalanced data, Korean text analysis, oversampling, SMOTE, supervised learning, topic classification,

Student Number: 2013-22801

Table of Contents

List of Tables

List of Figures

List of Equations

Introduction

Machine Learning and Korean Text Classification

Text classification is the task of assigning documents to categories through machine learning. The class of a document can be predicted through supervised learning by training a classification model based on pre-labeled training data. Quantification of text data can be achieved by capturing frequencies of words as features (Aggarwal & Zhai, 2013). If used well, text classification can be implemented on various tasks such as categorizing news articles in news aggregator sites and filtering spam mails.

Many machine learning algorithms can be used for text classification tasks. According to a survey on text classification algorithms, decision trees, pattern (rule)-based classifiers, support vector machines (SVM), neural networks, Bayesian (generative) classifiers are some notable classifiers that have been used for text classification (Aggarwal & Zhai, 2013). Sebastiani (2002) provides a more exhaustive list of classifiers including threshold determining policies, decision rule classifiers (DNF), regression methods, on-line methods, Rocchio methods, example-based classifiers such as the K-nearest neighbor method, and ensemble methods.

In the case of text classification studies on Korean texts, the support vector machine (SVM), naïve Bayes, and K-nearest

neighbor were most frequently implemented. For example, Oh, Zhang & Kim (1999) attempted to classify news articles into multiple categories using the naïve Bayes and SVM classifier. Kim (2009) compared SVM, naïve Bayes and the expectation-maximization (EM) algorithm to classify 500 computer-related articles, and Lee, Choi & Lee (2002) compared naïve Bayes, and the K-nearest neighbor classifier on yahoo web pages and categories. Kang et al (2016) compared the K-nearest neighbor, naïve Bayes and linear SVM classifier on documents on wearable IoT (Internet of things) patent articles. Ren & Kang (2015) also classified news categories with the SVM classifier. As it can be seen, the SVM, naïve Bayes, K-nearest neighbors were the most frequently used classifiers, and much attention is given to machine learning based text classification projects.

A Brief Introduction of the Main Classifiers

Most of the widely-used text classification algorithms implemented in Korean text classification tasks are the SVM, naïve Bayes, K-nearest neighbors. The general idea of classification algorithms will be briefly introduced in this section.

Logistic regression is a type of regression analysis that is conducted when the dependent variable is binary. As the logistic model provides predictions in the form of 0 or 1, the logistic regression is widely implemented in classification tasks (Friedman, Hastie, & Tibshirani, 2009). Assuming the input is a linear function with one independent variable x, the logistic function can be written as below.

$$F(x) = \frac{1}{1 + e^{-(\beta_o + \beta_1 x)}} \quad (1)$$

The support vector machine (SVM) implements a hyperplane to classify data in high dimensional feature space. To find the optimal hyperplane that would maximize distance between the margins, the SVM uses kernel tricks such as the radial basis function (RBF), the linear kernel or the sigmoid kernel. The optimal marginal data points that anchor the hyperplane is known as support vectors. $f(x)$ predicts the class for a new sample u with the weights w, where w are the weights for the hyperplane that provide the maximal margin trained on training data x and class y when label i is given.

$$f(x) = w \cdot u + b = \left(\sum_{i=1}^{l} a_i y_i x_i \right) + b \quad (2)$$

$$w = \sum_{i=1}^{l} a_i y_i x_i \quad (3)$$

The naïve Bayes is a probabilistic classifier that assumes independence between features. The algorithm learns the probability of each feature and predicts the class of a sample based on the sum of feature probabilities. The Laplace smoothing method and Lidstone smoothing method are generally implemented to ensure no prior probability of a feature is 0 (Friedman, Hastie, & Tibshirani, 2009). The estimate of class y having feature i can be written as below. α indicates the smoothing method.

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad (4)$$

The K-nearest neighbors (KNN) is a probabilistic classifier that predicts classes based on neighboring samples. By limiting neighborhoods to a set number of neighbors such as three, five or eleven, a sample is predicted as the majority class within the neighborhood (Friedman, Hastie, & Tibshirani, 2009). As it can be seen in Figure 1, the black circle can be predicted as a triangle when the neighborhood is set to a population of three samples, or a square when the neighborhood is set to a population of five.

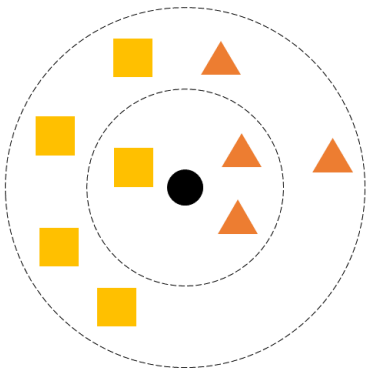


Figure 1. An illustration of the KNN algorithm. The dataset consists of triangle and square classes. The class of the black circle is unknown.

The Problem of Imbalanced Data

While machine learning holds much promise for automated text classification tasks, there are prerequisites that must be satisfied to deliver the best results. Data balance is one of the issues that interfere with the performance of machine learning algorithms. Data are imbalanced when the number of training data for each class is strikingly different. In the case of the yahoo webpage hierarchy text classification task carried out in 1999, there were 79,011 entertainment related pages and 1,995 reference related pages, resulting in a 1:200 imbalanced ratio (Grobelnik & Mladenic, 1999). Imbalanced data has been studied closely by the Data Mining society as many classifiers rely on the distribution of classes to predict the class of an unseen sample (Chawla, 2005). Weiss & Provost (2003)

shows that a balanced class distribution performs better than a natural distribution, showing that classifiers are indeed influenced by the class distribution of data. After experimenting with 26 diverse data sets and using the C4.5, the performance of the classifier was compared to when the balanced class distribution was implemented to classify imbalanced data or the natural distribution which encompasses the imbalanced state of data. In result, it was discovered that even if the data are imbalanced, a classifier that is trained on a balanced distribution performs better.

Imbalanced data can occur in natural settings or even in multi-label classification settings. In multi-label classification settings, many classification methods rely on repeatedly applying binary classification tasks for each label combination. In this process, certain combinations occur more than other combinations, leading to an imbalanced training set between the major combinations and rare combinations (Tsoumakas & Katakis, 2007). Thus, it can be said that text classification cannot be immune from the problem of imbalanced data and the need to address the imbalanced data problem is pertinent to future studies.

Approaches to Solve the Problem of Imbalanced Data

As balanced data is crucial to the performance of classifiers, the data science society has attempted to approach the problem of

imbalanced data in numerous ways. In 2000, at the Association for the Advancement of Artificial Intelligence (AAAI) workshop, class imbalance has received much attention from researchers that by the time of the workshop three years later, profuse methods such as resampling, cost-sensitive learning and expectation-maximization (EM) algorithm have been proposed and continued to be researched in the following years (Chawla et al, 2004).

According to a survey on imbalanced data, the many attempts to address the problem of imbalanced data can be broken down into the data level approach, algorithm level approach, and hybrid level approach (Krawczyk, 2016). The easiest way to solve the problem of imbalanced data is to supply more minority class samples at the data level. However, this may be realistically difficult due to the scarcity of obtaining minority class data such as data from rare diseases. Therefore, research on creating minority samples has been studied. The term oversampling is used when minority class samples are newly created to balance class ratio. Undersampling is used when majority class samples are discarded from the original data, approaching the imbalanced data in the opposite manner. It is noted that the terms oversampling and undersampling are also used in the fields of signal processing, along with similar terms such as upsampling and downsampling, all referring to techniques in signal amplification processes. While such methods share the same

terminology, the methods do not necessarily refer to the same processes used in data analysis.

The random resampling method has been used before sophisticated methods have been discovered, which randomly selects minority class samples and replicates them or randomly selects majority class samples and discards them from the data. Random oversampling would often cause overfitting, the phenomenon of the classifier being unable to generalize the decision borders and show drastic differences when a new data distribution is provided. Random undersampling holds the possibility of removing important samples in the distribution. Thus, a more sophisticated method of resampling method was called for, which led to the discovery of synthetic minority oversampling technique (SMOTE) (Chawla, 2005). SMOTE methods create new minority class data samples in feature space to balance training data. This results in a more generalized data distribution as well as a balanced training data. A detailed explanation on SMOTE methods will be provided in the next chapter.

Other than combatting the effects of imbalanced data at the data level, modification can be made at the algorithm level as well. In this case, the classification algorithms are either modified to incorporate the cost of misclassifying minority class samples or modified to integrate one class learning algorithms. Kim, Lee & Choi (2011) is an example of modifying the algorithm by combining offsets

and sampling weights in classifying customer behavior after telemarketing campaigns.

Lastly, other heuristics can be integrated to create hybrid methods. For example, an ensemble of classifiers can be used at the algorithm level, and different sampling methods and cost-sensitive learning methods can be hybridized at the data level.

While many methods are available to combat data imbalance, Kraczyk suggests approaching the issue at the data level to be more suitable to the general user as modifying algorithms requires expert insight to the algorithms. As machine learning based classification tasks are gaining much interest in the field of Korean text classification, it can be expected that studies on resampling methods applied on Korean text will also be on the rise. Therefore, this study would like to review resampling methods on Korean text classification tasks.

Literature Review

Imbalanced Data in Korean Studies

Many Korean studies are emerging that compare different resampling methods over diverse data types. For example, Lee and Kim (2016) analyzes radar data with naïve Bayes and SMOTE. Heo and Kim (2006) compare resampling methods on health insurance data with the decision tree. The resampling methods that were considered were random undersampling, random oversampling, SMOTE, and selective undersampling. Hong and Park (2016) compares three sampling methods with the J48 decision tree on software fault prediction. The compared sampling methods are random oversampling, SpreadSubsampling which randomly undersamples data incorporating proportionality, and SMOTE. Lee & Jun (2014) compares oversampling methods on Abalone9vs19, Vehicle0, Vowel0, and Pima from the KEEL dataset repository. The compared resampling methods were random undersampling, random oversampling, SMOTE, borderline-SMOTE, SPIDER which replicates majority instances, and the neighborhood cleaning rule which removes noisy minority samples. Implemented classifiers were the KNN classifier and Naïve Bayes classifier. Kim et al (2010) compares the effects of random undersampling and SMOTE with the multi-layer perceptron, naïve Bayes, random forest and SVM

classifier on randomly generated data. Many studies contribute by comparing the performance of diverse methods and suggesting the best resampling method. However, the studies mentioned above all deal with randomly generated or non-text data. No study compares resampling on Korean text studies, which leaves the question open on which resampling method performs best on text imbalanced text data.

Unfortunately, as much as machine learning is receiving much attention in Korean text classification tasks, studies directed toward imbalanced data are few. Most studies either artificially balance data samples per class, or disregard the imbalanced nature as a whole. For example, in the case of Oh, Zhang & Kim (1999), Kim (2009), and Kang et al (2016), the total number of samples are specified while the number of training samples per class are not specified. In the case of Kim & Seo (2013), while the number of training data per class is specified, the imbalanced ratio which reaches 2:1 is not addressed throughout the study. Thus, it can be said that imbalanced data is actively researched on non-text data such as randomly generated arrays of numbers or biological data, but not on text data.

Characteristics of Text Data

The structure of language gives rise to unique data qualities, calling for separate studies. Data sparsity and assumption of

dependence may be some factors that contribute to unique qualities of text data. Difference arising from such qualities may not be reflected well when algorithms implemented on non-text data are directly applied to text data.

First, text data has the tendency to be sparse. Text data contain large features once each word represents a feature in the training process. It is known that an adult native English speaker has a range of 20,000 to 35,000 words (Johnson, 2013) and a single news article would contain over 200 vocabularies. When data are collected from over hundreds of articles, the number of features in the whole dataset grows to be extremely large. Not only is calculation of large features a difficult task, data spread out among the features over limited data often leads to a distribution that is spread out and does not hold significant information that could be utilized in the classification process. Such problems are unique to text classification and research in the field are active, resulting in K-means algorithms or data reduction methods (Balbi, 2012).

Second, independence of data features should not be the underlying assumption of text data. Many languages have the characteristics of collocations, meaning that some words are habitually juxtaposed with other specific word. In English, phrases such as "make the bed", "feel free", "save time", and "wash up" are all words that are commonly used together. It is difficult to

assume independence between phrases that are collocations, calling for methods that can reflect such qualities in the algorithms.

Recognizing the special characteristics of text data, an exhaustive comparison on diverse SMOTE methods and classifiers has been conducted in the English language in the thesis of Liu (2004), comparing over ten SMOTE methods paired with three classifiers: SVM, naïve Bayes, and K−nearest neighbors. Yet, no such studies have been conducted on the Korean language.

Characteristics of the Korean Language

Different language characteristics should be considered in text analysis as they can give rise to different data structures. The Korean language is known to be focused around verbs while the English language is focused around nouns (Lee, 2002). This difference gives rise to different sentence structures and implementing different parts of speech, even the sentences convey the same meaning.

1. (a) The children watched in wide−eyed amazement.

   (b) 그 아이들은 놀라서 눈을 동그랗게 뜨고 바라 보았다.

In 1(a), it can be seen that "wide−eyed" is an adjective in English, but the same expression "눈을 동그랗게 뜨고" in 1(b) includes the

word “눈(eye)” which is a noun. Also “amazement” is a noun in English but the verb “놀라서” is used in the Korean expression. Similarly, the number of nouns can differ between both languages due to the different form of compound nouns or the level of sophistication Parts of Speech (POS) taggers provide. Therefore, it cannot be said that what works for English will work for Korean as well.

Compound nouns in the English language are nouns created by adjoining two or more nouns. At times, compound nouns may have the form of one word such as “basketball” and “snowman”. In other instances, compound nouns may have space between them, appearing as two words such as “swimming pool” and “rock star”. While spacing is occasionally used in the English language to refer to compound nouns, Korean compound nouns are used as a single word. While many of the words can be two independent nouns put together as illustrated in 2(a), often times it could be Sino-Korean compound nouns originating from hanja as illustrated in 2(b) (Lu, Nakazawa & Kurohashi, 2015).

2. (a) 사과나무 (apple tree)
= 사과 (apple) + 나무 (tree)

(b) 중학교 (middle school)
= 중 (中, middle) + 학 (學, learn) + 교 (校, educational institution)

The performance of Part of Speech (POS) taggers is instrumental to gaining exacting insight on the structure of the text data, and different qualities of the data will be overlooked when such subtle differences are overlooked. When English based text analysis algorithms on Korean text should be applied including the subtle structural differences, either the English POS tagger should be able to identify compound nouns even when they are separated by a space, or the Korean POS taggers should be able to break down single words based on meaning. However, POS taggers are still undergoing development and are not able to handle such detailed nuances. Therefore, compound nouns appearing like two words are considered separate words in the English POS taggers while in Korean POS taggers, compound nouns appear as a single word. Table 1 is an illustration of such differences. The python NLTK library (Bird et al, 2009) was implemented to analyze the English words while the python KONLPY twitter library's POS tagger (Park & Cho, 2014) was implemented for the Korean words as an example of the current state of POS taggers.

Subtle structural differences between language are acknowledged in the field of automatic Korean–English translation and paraphrasing techniques as well. Thus, researchers extract data by sentences rather than words or phrases to capture contextual information when developing algorithms to translate Korean and English texts, rather than looking at the words alone (Lee & Yun,

2006). Such incorporation of language characteristics should also be considered in text analysis tasks to secure optimum performance.

Table 1. Example of vocabulary differences when analyzing text data. While English POS tagger recognizes the Korean word '기관총(machine gun)' as two nouns, the Korean POS tagger recognizes the word as one noun.

Word	POS	Word Count
Machine gun	('machine', 'NN'), ('gun', 'NN')	2
기관총	('기관총', 'Noun')	1
Fish tank	('fish', 'NN'), ('tank', 'NN')	2
수조	('수조', 'Noun')	1

Research Question

Seeing how text classification is receiving much interest, solving the data imbalanced problem at the data level seems like the most viable option for those who cannot modify the algorithm directly. While it can be said that text data and each language data have unique qualities that have the potential to affect learning algorithms, there is a lack of comparative studies especially in the Korean imbalanced text classification studies. Therefore, this study would like to compare major resampling methods in imbalanced situations.

This study would like to especially focus on oversampling methods as undersampling methods hold the danger of eradicating important data features. In text data, although some data may seem erratic, at times the isolated data may hold important characteristics representative of the class. To minimize any possible data loss, this study would like to forego undersampling and compare oversampling methods alone.

Thus, this study would like to compare the performance of different SMOTE methods paired with four different classifiers on Korean news articles to determine the best performing combination on Korean text classification tasks.

Introduction to SMOTE Methods

SMOTE

Many methods have been proposed to conduct a sophisticated version of oversampling. Chawla (2002) introduced the Synthetic Minority Oversampling TEchnique (SMOTE), a novel method to generate new minority class samples. This method generates more minority class samples so that they are positioned randomly between a minority sample and its nearest neighbor. This method helps prevent over-fitting as it populates empty data space instead of simply replicating existing samples, while trying to stay true to the minority example space representation.

SMOTE will take a minority class sample x_i from the feature space, and randomly choose a neighbor x_{zi} from K-nearest neighbors. A synthetic sample s_i is generated by following the equation below where $\lambda \in [0, 1]$.

$$s_i = x_i + (x_{zi} - x_i) \times \lambda \quad (5)$$

This procedure of minority sample synthesis is carried out on all randomly chosen minority samples until the minority and majority classes reach the desired ratio of balance (Chawla, 2002).

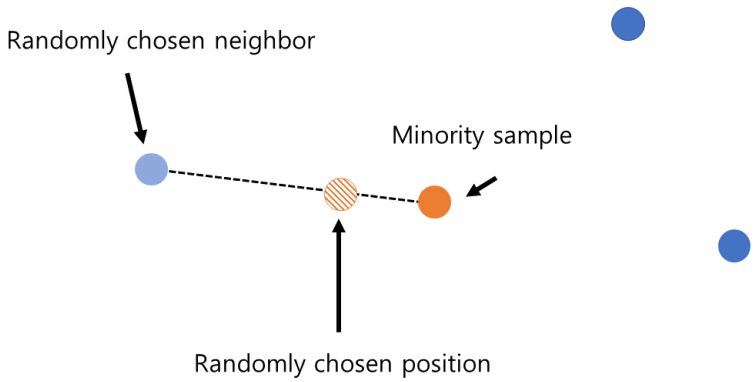


Figure 2. Illustration of the synthesis of a new sample by SMOTE in feature space. The orange circle indicates x_i in feature vector while the blue circles indicate majority samples. The light blue circle indicates x_{zi}, and the orange line-pattern circle indicates s_i.

Borderline-SMOTE

Since SMOTE was proposed, many studies have utilized SMOTE methods to improve its performance. While regular SMOTE generates synthetic examples from all minority samples indiscriminately, borderline-SMOTE, proposed by Han, Wang and Mao (2005), generates synthetic minority samples only for samples that are the minority amongst its neighbors. The algorithm first finds all endangered minority samples P. Minority samples are endangered if the number of majority K-nearest neighbors $pnum$ is larger than the number of minority sample neighbors $dnum$.

$$\mathrm{P} = \left\{ p'_1, p'_2, \ldots, p'_{dnum} \right\}, \qquad 0 \le dnum \le pnum \tag{6}$$

If the minority sample is endangered, the algorithm creates more minority samples with the equation below. The endangered minority sample is represented by x_i and p'_i represents an endangered minority sample within K−nearest neighbors while $p' i \in \mathrm{P}$. A minority synthesis s_i is made where $\lambda \in [0,1]$. This has the effect of strengthening minority sample neighborhood at the forefront of decision boundaries, earning its name, borderline−SMOTE.

$$s_i = x_i + \left(p'_i - x_i \right) \times \lambda \tag{7}$$

Borderline−SMOTE2 is also introduced in the same paper which proposes to oversample the majority samples as well as carrying out borderline−SMOTE. The majority samples are synthesized according to the equation below. While s_j represents a newly generated majority sample, k'_i represents a randomly chosen majority sample within K−nearest neighbors of x_i. λ in this case a random number where $\lambda \in [0, 0.5]$.

$$s_j = x_i + \left(k'_i - x_i \right) \times \lambda \tag{8}$$

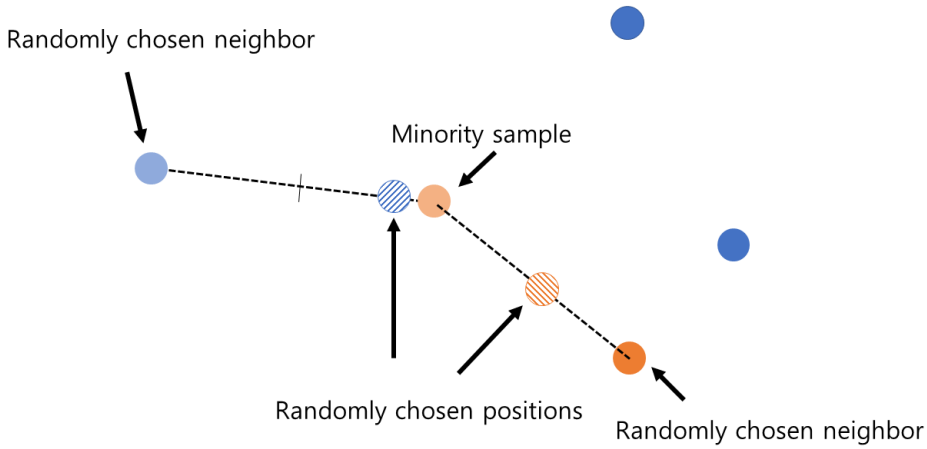


Figure 3. Illustration of the synthesis of borderline-SMOTE2. The blue circles indicate majority samples, outnumbering the minority samples which are orange. The light color orange circle indicates x_i. The light colored blue circle indicates k'_i and the dark orange circle is p'_i. The line-pattern orange circle represents s_i and the line-pattern blue circle represents s_j. Borderline-SMOTE would not generate a majority sample.

This operation of oversampling majority and minority samples is carried out until desired balance between classes is achieved. Oversampling majority samples as well as oversampling minority samples results in a decision borderline consisted of balanced data. In this paper, it is shown that in certain cases, borderline-SMOTE2 responds better than borderline-SMOTE.

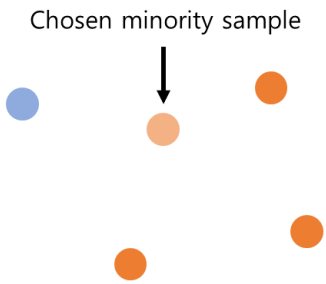


Figure 4. Illustration of when the borderline−SMOTE algorithm will not be activated. If the minority samples are the majority, the algorithm will not synthesize data.

SVM−SMOTE

While SMOTE methods mentioned above identified the borderline through the nearest neighbor method, Nguyen, Cooper & Kamei (2009) tried strengthening the borderline by identifying the borderline as support vectors from the support vector machine (SVM). SVM aims to find a hyperplane that separates the classes with maximum margin. In the process, support vectors play an integral role of anchoring the separating plane. SVM−SMOTE attempts to take advantage of the support vectors' characteristics and perform SMOTE on them. This can be done so by maximizing the equation below while $0 \leq a_i \leq C, \forall_i$ and $\sum_i a_i y_i = 0$ where a_i are Lagrangian multipliers of training samples, C is the parameter chosen to penalize training instances that are dislocated and y_i is the class label where $y_i \in \{+1, -1\}$. $\mathrm{K}(x_i, x_j)$ is a kernel function for

computing dot products of two samples x_i and x_j in feature space. x_i is a sample in the training set $\{(X_i, y_i)\}, i = 1, \ldots, N$. The support vectors will be $a_i > 0$.

$$\sum_i a_i - \frac{1}{2}\sum_{i,j} a_i\, a_j y_i y_j K(x_i, x_j) \tag{9}$$

After identifying support vectors that belong to the minority class, the algorithm will identify its nearest neighbor and carry out equation (10) or (11), depending on the situation. If the minority sample is outnumbered by majority samples, a new minority sample s_i is placed so that it is extrapolated by equation (10) where sv_i is a minority support vector, x_{im} is the m closest neighbor, and $\lambda \in [0, 1]$. If the minority sample is not outnumbered, a new minority sample will be interpolated by equation (11). This process is carried out on all minority samples until desired balance is reached.

$$s_i = sv_i + (sv_i - x_{im}) \times \lambda \tag{10}$$

$$s_i = sv_i + (x_{im} - sv_i) \times \lambda \tag{11}$$

ADASYN

Bai, Garcia, He & Li (2008) proposed ADaptive SYNthetic Sampling Approach for Imbalanced Learning (ADASYN) which utilizes the density distribution of a minority example as criteria for how much synthetic data should be generated for each minority example. This is done so by finding the differing density distribution

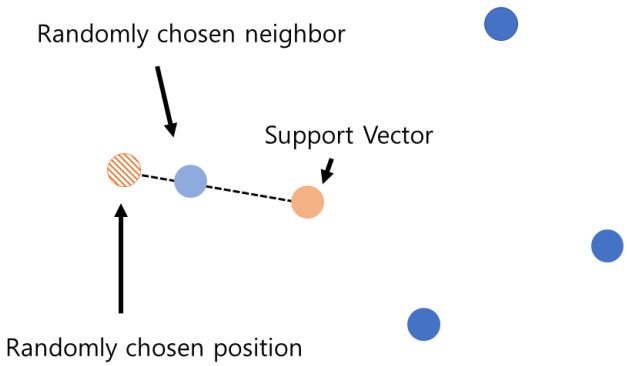


Figure 5. Illustration of the SVM−SMOTE algorithm' s extrapolation. The light orange colored circle represents sv_i which is a minority sample. The light blue circle indicates the first nearest neighbor x_{i1}. As the minority sample is outnumbered by majority samples, the algorithm will extrapolate by following equation (10).

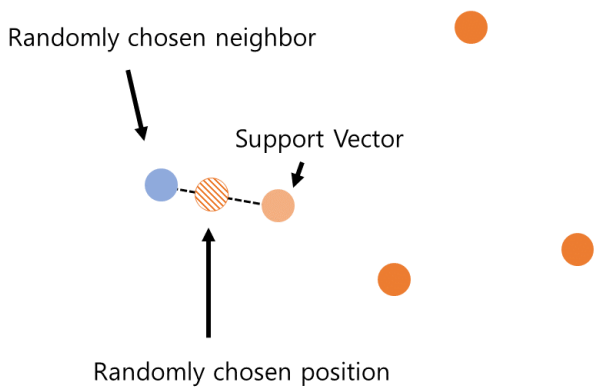


Figure 6. Illustration of the SVM−SMOTE algorithm' s interpolation. The light orange colored circle represents sv_i which is in a neighborhood where the majority belong to the minority class. The light blue circle indicates the first nearest neighbor x_{i1}. As the

minority sample is outnumbered by majority samples, the algorithm will interpolate by following equation (11).

for each minority sample and supplementing minority samples as much as each sample requires to become balanced with the majority class. This approach helps focus attention on minority examples depending on how difficult they are to learn. If ADASYN were to balance to a 1:1 ratio, the formula to find the density distribution $\hat{r}_i$ for minority sample x_i would be as the following where Δ_i represents the number of examples in K−nearest neighbors of minority sample x_i.

$$r_i = \frac{\Delta_i}{K}, i = 1, \ldots, m_s \tag{12}$$

r_i can be normalized in the following way:

$$\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i \tag{13}$$

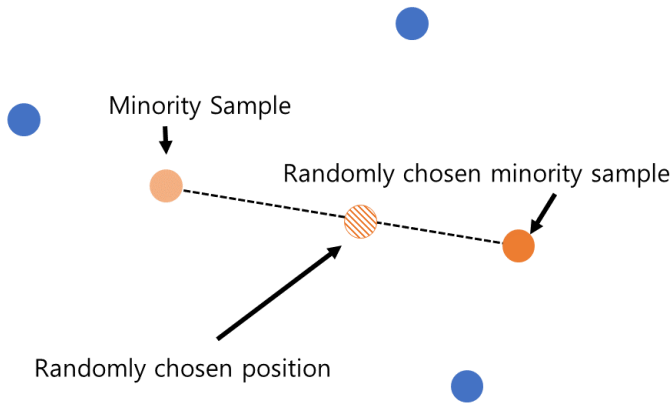


Figure 7. Illustration of the ADASYN algorithm. The light orange circle represents x_i while the blue circles represent majority samples. The dark orange circle represents x_{zi}. The line-pattern circle is the newly generated s_i.

Then, the number of data examples that need to be generated for each minority sample g_i can be found in the following formula where m_l represents the number of majority samples and m_s represents the number of minority samples, and β is the level of balance between the two classes. If the classes are balanced equally, β would be 1. If the classes are used originally as they were, β would be 0.

$$\mathrm{G} = (m_l - m_s) \times \beta \tag{14}$$

$$g_i = \hat{r}_i \times G \tag{15}$$

Finally, the synthetic sample s_i can be made g_i times for x_i in the following formula where x_i represents the minority sample of interest, x_{zi} represents a randomly selected minority sample within K−nearest neighbors of x_i, and $\lambda \in [0,1]$.

$$s_i = x_i + (x_{zi} - x_i) \times \lambda \tag{16}$$

A Framework for Comparing the Effectiveness of SMOTE Methods

Relevant Factors in Classification Tasks

Japkowicz (2000) shows that the imbalanced ratio and the complexity are influential factors in classification tasks. In a study that used a multi-layer perceptron as a binary classifier, Japkowicz conducted an experiment varying size, imbalanced ratios and complexity of the data to find characteristic factors of the data that influence classification performance. Japkowicz generated data from random numbers from 0 to 1 and assigned classes by assigning either 0 or 1 to the random numbers. Then, complexity was added by dividing sectors within the random numbers and assigning classes alternatively. For example, if data were generated at complexity 0, class 0 data will be generated from 0 to 0.5 while class 1 data will be generated from 0.5 to 1. The amount of numbers generated from each class was determined by the level of imbalance. If data were generated at complexity 1, class 0 data will be generated from 0 to 0.25 as well as from 0.5 to 0.75, while the rest will be classified as 1. This study shows that no matter how small or large data are, projected error levels are linked to imbalanced ratios and difficulty of the concept.

Performance Measures

Traditionally, precision, recall and accuracy are basic tools of measuring the performance of a classifier (Ling & Li, 1998; Fawcett & Provost, 2001). Precision, recall and accuracy utilize information about the actual and predicted classes that result from classification tasks. All possible cases of the results can be represented in the following confusion matrix.

		Predicted	
		Positive	Negative
Actual	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

A positive case is when the example belongs to a class, while the negative case represents that the example does not belong to the certain class. A true positive is when the classifier predicts the case as belonging to the relevant class, and it indeed belonged to the relevant class. A false positive is when the classifier predicted that the case belongs to the class, but actually didn't. True negatives and false negatives hold the same, only indicating that the example did not belong to the certain class.

Precision is how well the classifier can correctly predict a positive case when it is indeed positive, indicating confidence. Recall is how well the classifier can predict the predicted positive out of the real positive cases, indicating the coverage of the performance.

Accuracy is how well the classifier can predict a positive case positive, and negative case negative. Error rate is when the classifier predicted a positive case negative and the negative case positive. The performance measures can be computed in the following ways:

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

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

$$\text{Accuracy} = \frac{(True\ Positive + True\ Negative)}{(True\ Positive + False\ Positive + True\ Negative + False\ Negative)}$$

$$\text{Error Rate} = 1 - \text{Accuracy}$$

When researchers apply machine learning on imbalanced classification tasks, accuracy is not relied upon. This is because high accuracy can be achieved by predicting that all of the cases as the majority case, and can get away with yet a high score. Therefore, the machine learning community has embraced the Receiver Operating Characteristic (ROC) curve and F-value as sound measurements (Chawla, Japkowicz & Kotcz, 2004; Ailab & Powers, 2011). The ROC curve shows the tradeoff between true positives and false positives. If the false positive cases are neglected and the focus is skewed toward the true positives only, it will most likely represent precision. On the other hand, if the true positive cases are neglected and the focus is skewed toward the false positives, the scores will most likely represent recall. The area under the curve (AUC) represents the performance of the classifier. The F-value is a measure that

encompasses the tradeoff between precision and recall and reflects how "good" a classifier is in a single value. Thus, the F-value is widely used in imbalanced data classification tasks and will be used as the main performance measurement in this study.

$$\mathrm{F-value} = \frac{2 * recall * precision}{recall + precision}$$

Implementation

All classifiers were implemented utilizing the scikit-learn python library (Blondel et al, 2011) and the SMOTE methods were implemented through the Imblearn python library (Lemaître, Nogueria & Aridas, 2017). Part of speech tagging for processing natural language was implemented with KoNLPY's Twitter library (Park & Cho, 2014). The parameters for classifiers can be found in the Appendix. All imbalanced data were resampled until class ratio reached 1:1.

Text Preparations

1,000 documents were gathered by randomly selecting news articles that were longer than 800 characters from subcategories assigned by Naver's news page. Naver's news page is divided into large categories such as 'Politics', 'Economy' and 'Society', and large categories are further divided into smaller subcategories

such as ‘The National Assembly and Political Parties’, ‘North Korea’, ‘Education’, ‘Labor’, ‘Health’ and so on. A detailed list of categories and subcategories can be found in the Appendix.

The classification task was conducted on two different classification tasks. One was classifying articles between the classes of 'finance' and 'stock'. Another was classifying articles between the classes of 'education' and 'environment'.

The ‘finance’ related articles had an average count of characters (white space included) of 1,478, and ‘stock’ related articles had 1,438 characters. ‘Finance’ related articles had an average of 317 words and ‘stock’ related articles had an average of 303 words. The two articles had an overall cosine similarity of 0.746, showing a relatively high level of similarity. A sample of the articles from each subcategory can be found in the Appendix.

‘Education’ related articles had an average count of 1,644 characters (white space included), and 'environment' related articles had 1,444 characters. Articles related to ‘education’ had an average of 364 words and ‘environment’ related articles had an average of 316 words. These articles had a cosine similarity of 0.472, showing a lower level of similarity compared to the similarity of 'finance' and 'stock' related articles. A sample of the articles from

each subcategory can be found in the Appendix.

Table 2. Cosine similarities of news articles by category

Categories	Finance	Stock	Education	Environment
Finance	1	0.746	0.335	0.427
Stock	0.746	1	0.281	0.382
Education	0.335	0.281	1	0.472
Environment	0.427	0.382	0.472	1

2,000 nouns were extracted for the term-document matrix features. A term-document matrix contains information on how often a feature word has appeared per document. The matrix was transformed through TF-IDF weighting to help extract more meaning from the term-document matrix. First introduced in Jones (2004), the TF-IDF is found by normalizing the term frequency by frequency over document length, and then computing the inverse document frequency by putting the total number of documents over the number of documents that have the specified term in natural logarithm. TF-IDF weighting is widely used in text classification tasks as it assigns higher weight to terms that occur often in a small number of documents while assigning low weight to terms that do not occur often in a document or if it occurs in most of the documents. This weighting method thus provides emphasis on features distinct to the document.

$$Term\ Frequency(t) = \frac{Frequency\ of\ t\ in\ the\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

$$Inverse\ Document\ Frequency(t) = log_e(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ that\ have\ t})$$

Method

To simulate imbalanced data, 'stock' and 'environment' articles were chosen to represent the minority class and the number of minority class articles used for training were set at different numbers: 50, 100, 200, 300, 400, 500, 600, 700 and 850 to represent imbalanced data. The number of documents from 'finance' and 'education' was fixed at 900 to represent the majority class. The experiment sought to examine the differing performance between imbalanced data and SMOTE-resampled balanced data. The performance of five oversampling methods, random oversampling, regular SMOTE, borderline-SMOTE, SVM SMOTE and ADASYN, were compared to discover the best performing SMOTE method.

100 documents out of the 1,000 documents for each class were set aside for a test set to compare the classifier's performance. Each classification task was given 10 trials when possible, to ensure that each minority training set was not biased. The F1 scores were averaged in this study.

For each experiment, three data scenarios were given (balanced, data, imbalanced data, and resampled data), and were

examined over four classifiers (naïve Bayes, KNN, SVM, logistic regression), nine imbalanced scenarios (50, 100, 200, 300, 400, 500, 600, 700, 850), and six resampling methods (random oversampling, SMOTE, borderline−SMOTE, borderline−SMOTE2, SVM−SMOTE, ADASYN).

	Classifiers	Imbalanced scenarios	Resampling Methods	Possible cases
Balanced Data	4	None	None	4
Imbalanced Data	4	9	None	36
Resampled Data	4	9	6	216

Experiments

Study 1: Articles with High Cosine Similarities

In study 1, 'finance' and 'stock' related articles were trained and classified upon 100 test data. As mentioned earlier in table 2, the 'Finance' articles and 'Stock' related articles had a high cosine similarity of 0.746.

Table 3. Performance of classifiers when data was balanced. Logistic regression and SVM show the highest level of performance.

Classifiers	F1 Scores
KNN	0.761421
Logistic regression	0.819048
Naïve Bayes	0.743961
SVM	0.819048

When training data was balanced, meaning that 900 'finance' related articles and 900 'stock' related articles were used as training data and tested on 100 articles from each category, all classifiers showed the performance of achieving F1 scores above 0.7. As it can be seen in table 3, the best performing classifier was logistic regression and SVM, reaching an F1 score of 0.81. This aligns with the findings of Yang (1999), discovering that SVM was the best classifier for text.

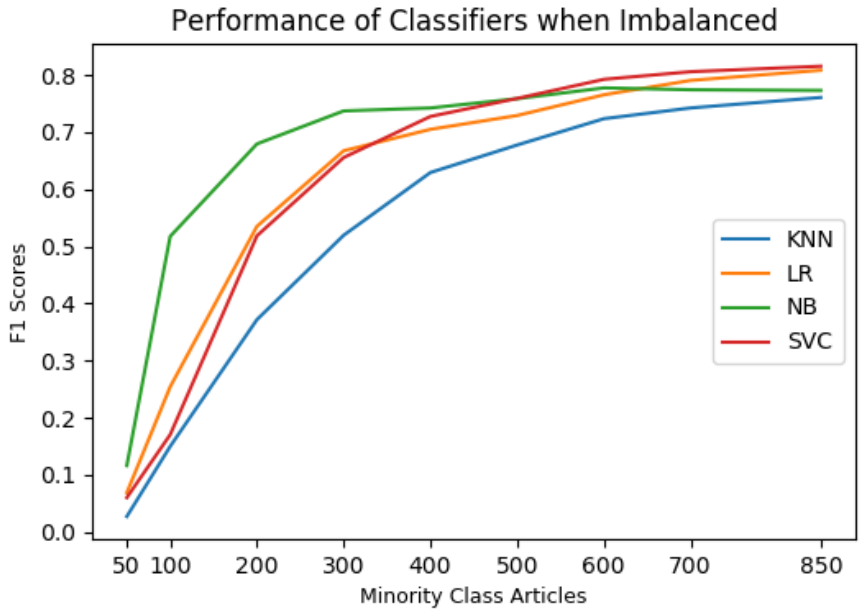


Figure 8. Performance of classifier by differing imbalanced levels. Naïve Bayes performs best when data is imbalanced while KNN performs worst.

When training data was imbalanced, meaning that 900 'finance' related articles were trained with a smaller number of 'stock' related articles, F1 scores differed drastically. Table 4 and figure 8 represent the performance of classifiers by imbalanced levels. When minority class training data were scarce, naïve Bayes performed relatively better than the other classifiers. Naïve Bayes reached an F1 score of 0.5 when there were only 100 minority training data while others required 200 minority training data to reach the same F1 score. However, performance of the naïve Bayes

classifier did not increase after minority training data reached 300 while other classifiers gradually showed higher performance as more minority class training data were supplied.

Table 4. Performance of classifiers when training data were imbalanced. Best performing classifiers are shaded in grey.

n	KNN	Logistic regression	Naïve Bayes	SVM
50	0.027332	0.068824	0.116473	0.06032
100	0.150016	0.254315	0.517599	0.170509
200	0.371882	0.535163	0.679433	0.518628
300	0.519582	0.667471	0.737197	0.655603
400	0.629376	0.704972	0.742342	0.727486
500	0.677553	0.729175	0.758595	0.759077
600	0.723573	0.765454	0.777529	0.792475
700	0.74239	0.790594	0.774266	0.80582
850	0.760713	0.808398	0.773047	0.815412

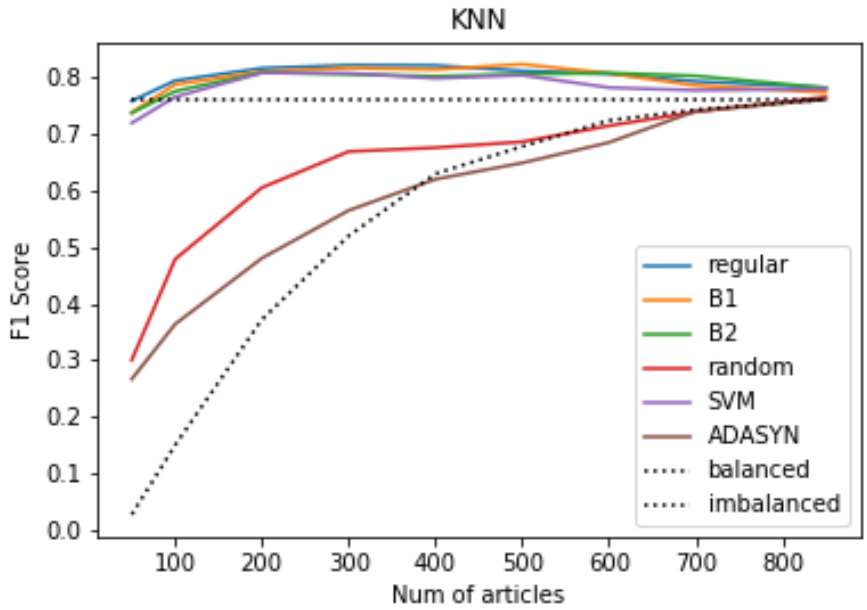
Overall, the KNN classifier performed the worst, requiring 600 minority class cases to achieve an average F1 score above 0.7 while the rest of the classifiers required at least 400 minority class cases to reach an average F1 score above 0.7.

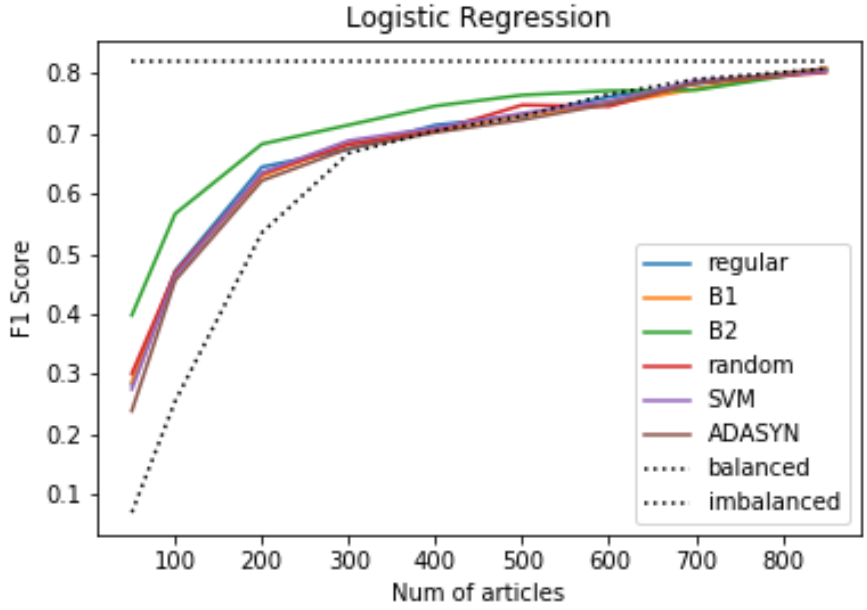
Table 5. F1 scores of classifiers when training data w343 resampled to balance. The p-value of F1 scores were obtained through t-test between the best performing SMOTE method and imbalanced data. P-values below 0.005 are accented in bold. Best performing combinations are shaded in grey.

n		KNN	Logistic regression	Naïve Bayes	SVM
50	F1 score	0.758040	0.397366	0.531628	0.419299
	p-value	**0.0000**	**0.0000**	**0.0000**	**0.0000**
100	F1 score	0.793889	0.566581	0.684861	0.573388
	p-value	**0.0000**	**0.0000**	**0.0000**	**0.0000**

200	F1 score	0.815841	0.682789	0.752904	0.711924
	p-value	0.0000	0.0000	0.0000	0.0000
300	F1 score	0.821162	0.714483	0.782136	0.746383
	p-value	0.0000	0.0170	0.0000	0.0000
400	F1 score	0.820775	0.746172	0.781124	0.769218
	p-value	0.0000	0.0007	0.0008	0.0000
500	F1 score	0.822862	0.764239	0.793044	0.772302
	p-value	0.0000	0.0253	0.0008	0.4011
600	F1 score	0.807350	0.771418	0.794735	0.788579
	p-value	0.0000	0.7202	0.0232	0.6317
700	F1 score	0.802277	0.787784	0.788455	0.804207
	p-value	0.0000	0.8162	0.0228	0.8564
850	F1 score	0.781436	0.808889	0.773237	0.816554
	p-value	0.0021	0.9393	0.9814	0.9045

As it can be seen in table 5, all classifiers benefited from the effects of SMOTE resampled data. Logistic regression and SVM had valid p-values compared to imbalanced F1 scores until original minority class supplied data reached 400. Naïve Bayes benefited until original minority class data were 500, and KNN benefited drastically overall. Afterwards, resampling methods showed no difference from using imbalanced data. Also, KNN reached the F1 score of 0.75 from the start while others started off at 0.39 to 0.53. This is a significant improvement from imbalanced data which required a minimum of 400 minority class samples to reach such F1 scores.


KNN
F1 Score
Num of articles
regular
B1
B2
random
SVM
ADASYN
balanced
imbalanced


Logistic Regression
F1 Score
Num of articles
regular
B1
B2
random
SVM
ADASYN
balanced
imbalanced

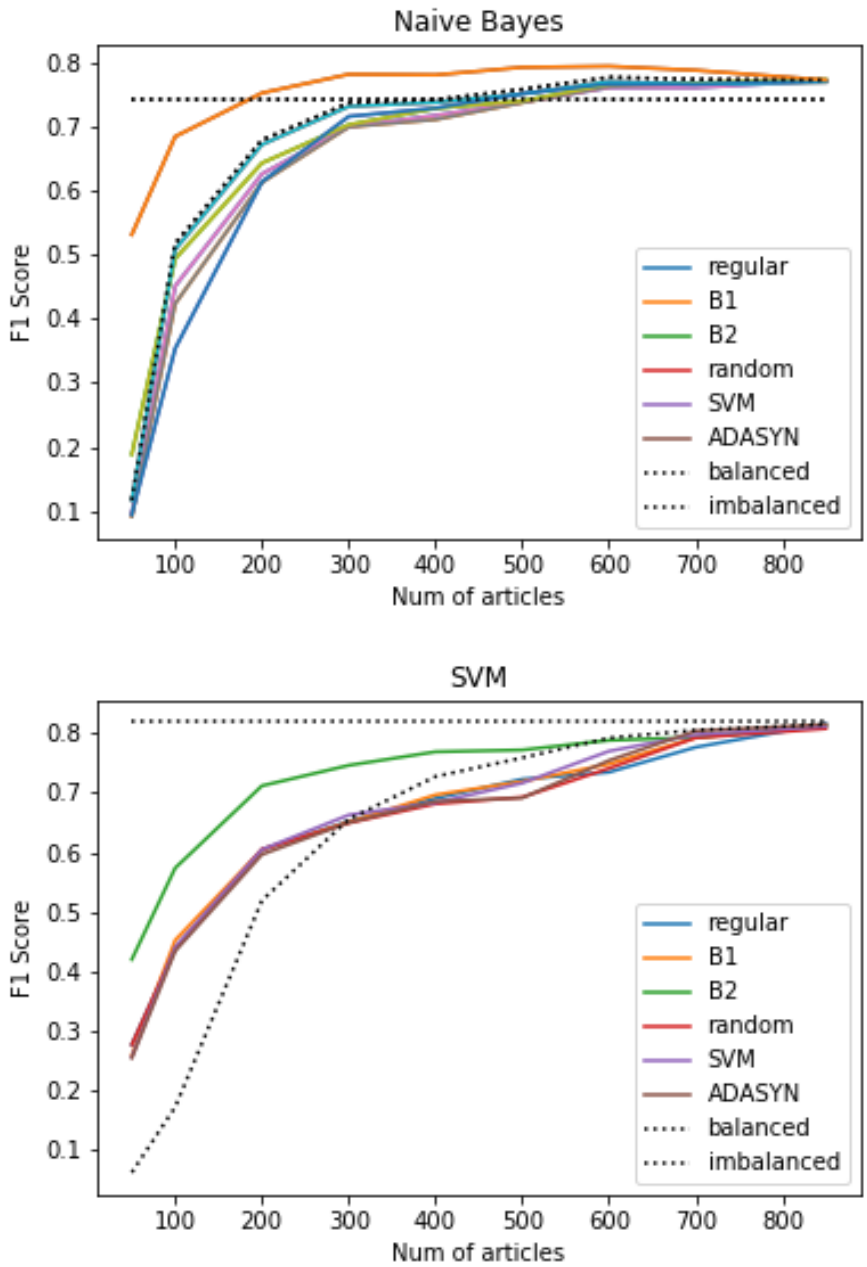


Figure 9. Performance of classifier by the number of minority class training data and the use of resampling methods. 'Balanced' indicates the F1 score when majority and minority class examples were balanced at 1:1. 'Imbalanced' indicates the F1 score before the data were resampled.

As it can be seen in figure 9, table 6 and table 7, each classifier reacted slightly differently depending on the applied SMOTE method. SVM was better when it was paired with borderline-SMOTE2 until minority data reached 500. KNN performed well with SMOTE, borderline-SMOTE, borderline-SMOTE2, and SVM-SMOTE, showing statistical difference from ADASYN and regular oversampling. Logistic regression reacted better with borderline-SMOTE2 until minority data reached 300. There was no statistical significance afterwards, compared to using other SMOTE methods. Naïve Bayes performed best when it was paired with ADASYN until minority data reached 600. Afterwards, there was no statistical significance compared to using other SMOTE methods. Overall, SMOTE methods proved to improve classifier performance when imbalanced levels were extreme.

Table 6. F1 scores for SMOTE by classifiers. Highest scores are shaded in grey. Resampling methods distinguishable among other resampling methods accented in bold. Actual p-values can be found in Table 7.

SVM

n	Regular	B1	B2	Random	SVM	ADASYN	Imbalanced
50	0.275852	0.257674	**0.419299**	0.275852	0.255327	0.253247	0.06032
100	0.436702	0.451858	**0.573388**	0.436702	0.440427	0.433537	0.170509
200	0.60394	0.60436	**0.711924**	0.604629	0.604471	0.59616	0.518628
300	0.651934	0.650586	**0.746383**	0.649243	0.662688	0.64979	0.655603
400	0.690695	0.696627	**0.769218**	0.681483	0.683482	0.685723	0.727486
500	0.72303	0.71910	**0.772302**	0.69269	0.71647	0.691447	0.759077

	8	7		1			
600	0.734813	0.748544	0.788579	0.741325	0.77025	0.754542	0.792475
700	0.776933	0.793139	0.793428	0.792786	0.798033	0.804207	0.80582
850	0.816554	0.813902	0.812599	0.808214	0.811987	0.815412	0.815412

KNN

n	Regular	B1	B2	random	SVM	ADASYN	Imbalanced
50	0.75804	0.737523	0.736909	0.300386	0.71897	0.266596	0.027332
100	0.793889	0.787046	0.775045	0.478582	0.765191	0.36375	0.150016
200	0.815841	0.810032	0.808366	0.604861	0.807538	0.480229	0.371882
300	0.821162	0.816147	0.805089	0.668629	0.807084	0.564583	0.519582
400	0.820775	0.813574	0.801629	0.675399	0.79787	0.619634	0.629376
500	0.810935	0.822862	0.805285	0.685813	0.803502	0.648486	0.677553
600	0.805898	0.806894	0.80735	0.714117	0.781575	0.684628	0.723573
700	0.792586	0.785275	0.802277	0.73867	0.776847	0.739948	0.74239
850	0.781436	0.773138	0.781033	0.764869	0.779283	0.760713	0.760713

Logistic Regression

n	Regular	B1	B2	random	SVM	ADASYN	Imbalanced
50	0.282191	0.285655	0.397366	0.300294	0.274239	0.238111	0.068824
100	0.47131	0.466467	0.566581	0.468523	0.463217	0.455813	0.254315
200	0.64449	0.629443	0.682789	0.633203	0.635278	0.621861	0.535163
300	0.674162	0.683421	0.714483	0.682344	0.688399	0.675278	0.667471
400	0.714372	0.706104	0.746172	0.704236	0.710039	0.702157	0.704972
500	0.729436	0.731072	0.764239	0.747738	0.733813	0.72297	0.729175
600	0.761486	0.748844	0.771418	0.745521	0.754508	0.749656	0.765454
700	0.783039	0.774672	0.772667	0.785107	0.787784	0.783953	0.790594
850	0.805475	0.808889	0.806684	0.801987	0.80384	0.808398	0.808398

Naïve Bayes

n	Regular	B1	B2	random	SVM	ADASYN	Imbalanced
50	0.095738	0.091014	0.188216	0.116473	0.094245	**0.531628**	0.116473
100	0.451083	0.423599	0.493827	0.508362	0.353635	**0.684861**	0.517599
200	0.626316	0.61318	0.64342	0.672184	0.613993	**0.752904**	0.679433
300	0.700239	0.699851	0.703191	0.731876	0.716225	**0.782136**	0.737197
400	0.717863	0.711345	0.728901	0.73816	0.729199	**0.781124**	0.742342
500	0.738608	0.73776	0.740213	0.751492	0.751763	**0.793044**	0.758595
600	0.761204	0.762993	0.766561	0.770682	0.767082	**0.794735**	0.777529
700	0.761147	0.768388	0.76841	0.766144	0.766765	0.788455	0.774266
850	0.772037	0.772562	0.773237	0.769914	0.771794	0.773047	0.773047

Table 7. P−values obtained by ANOVA among the resampling methods. P−values below 0.005 are accented in bold

	KNN	Logistic regression	Naïve Bayes	SVM
50	**0.0000**	**0.0000**	**0.0000**	**0.0000**
100	**0.0000**	**0.0000**	**0.0000**	**0.0000**
200	**0.0000**	**0.0000**	**0.0000**	**0.0000**
300	**0.0000**	0.1063	**0.0000**	**0.0000**
400	**0.0000**	0.0455	**0.0000**	**0.0000**
500	**0.0000**	0.0999	**0.0000**	**0.0001**
600	**0.0000**	0.4583	**0.0004**	0.0089
700	**0.0000**	0.6705	**0.0001**	0.1797
850	0.0399	0.9678	0.9997	0.9838

Study 2: Articles with Low Cosine Similarity

In study 2, 'Education' and 'Environment' related articles were examined in the same manner. As mentioned earlier in table 2, the articles had a comparatively lower cosine similarity of 0.472.

Table 8. Performance of classifiers when data were balanced. Logistic regression and SVM show the highest level of performance.

Classifier	F1 score
KNN	0.928571
Logistic regression	0.96
Naïve Bayes	0.94359
SVM	0.964824

When training data were balanced, meaning that 900 'education' related articles and 900 'environment' related articles were used as training data and tested on 100 articles each, all classifiers showed the performance of achieving F1 scores above 0.9. As it can be seen in Table 8, the best performing classifiers were logistic regression and SVM, reaching the F1 score of 0.96, which yielded the same results as in study 1.

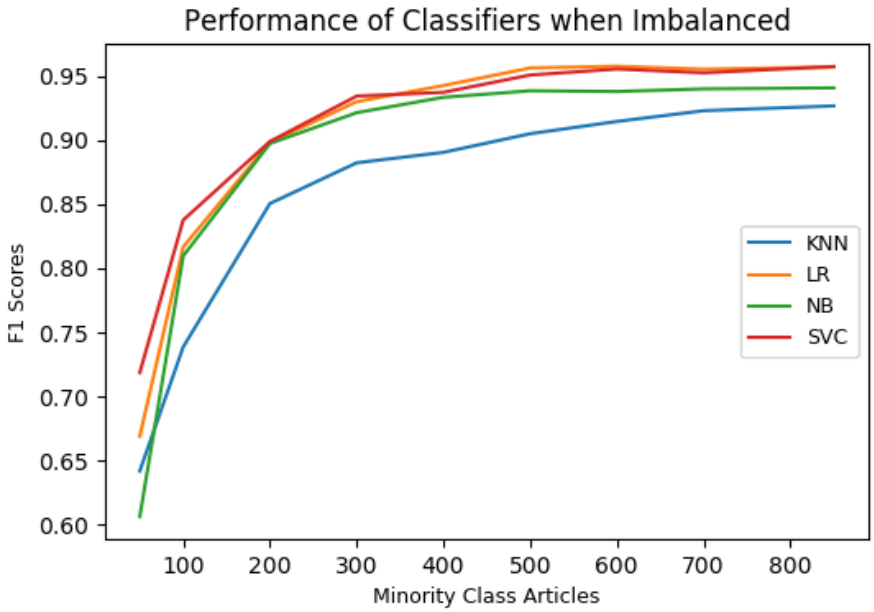


Figure 10. Performance of classifiers by differing imbalanced levels. SVM performs best when data were imbalanced.

When training data were imbalanced, meaning that 900 'education' related articles were trained with a smaller number of 'environment' related articles to represent imbalanced data, F1 scores differed drastically to when data were balanced. Table 9 and figure 10 represent the performance of classifiers by imbalanced levels. When minority class training data were scarce, SVM performed relatively better than the other classifiers. SVM reached the F1 score of 0.7 even when there were only 50 minority training data while others soon caught up when 100 minority training data were supplied. Logistic regression shows a slightly higher performance when there were more minority data.

Table 9. Performance of classifiers when training data were imbalanced. SVM performs best when data were imbalanced until the minority class example count reaches 400. After 400 minority training samples, logistic regression performed slightly better than SVM. The best classifiers per minority size are shaded in grey.

n	KNN	Logistic	Naïve Bayes	SVM
50	0.642090	0.669201	0.606620	0.718779
100	0.738660	0.816583	0.809593	0.837534
200	0.850470	0.898144	0.897230	0.898816
300	0.882168	0.929800	0.921212	0.934158
400	0.890316	0.942379	0.933202	0.937086
500	0.904852	0.956079	0.938284	0.950575
600	0.914390	0.957427	0.937735	0.955268
700	0.922780	0.955096	0.939796	0.952282
850	0.926509	0.956826	0.940558	0.957197

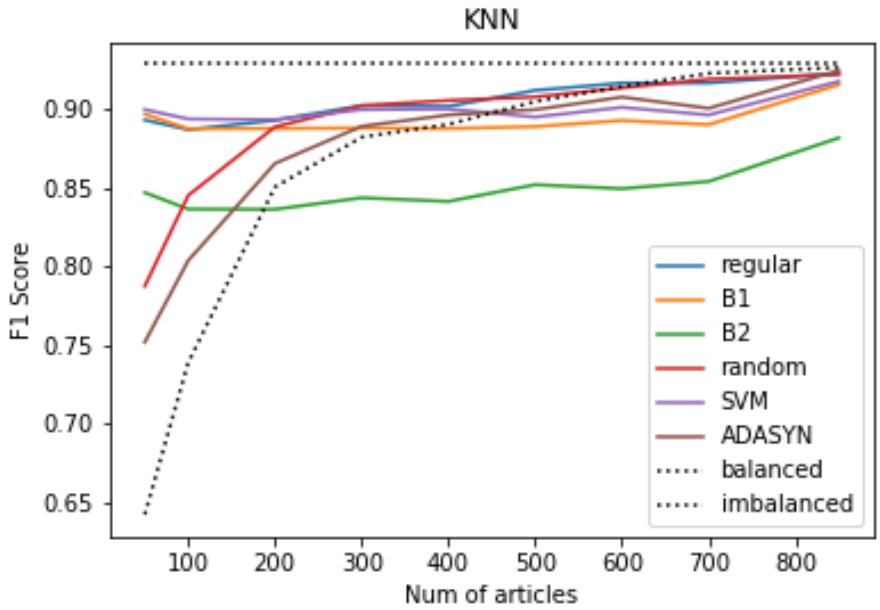
Overall, the KNN classifier performed slightly less than the other classifiers, nevertheless scoring an average F1 score of 0.64 when there were only 50 cases of minority examples while the majority training data had 900. This suggests that in similar Korean text classification tasks that have a cosine similarity around 0.47, if the data imbalanced ratio is above 1:9, the F1 score may reach above 0.7.

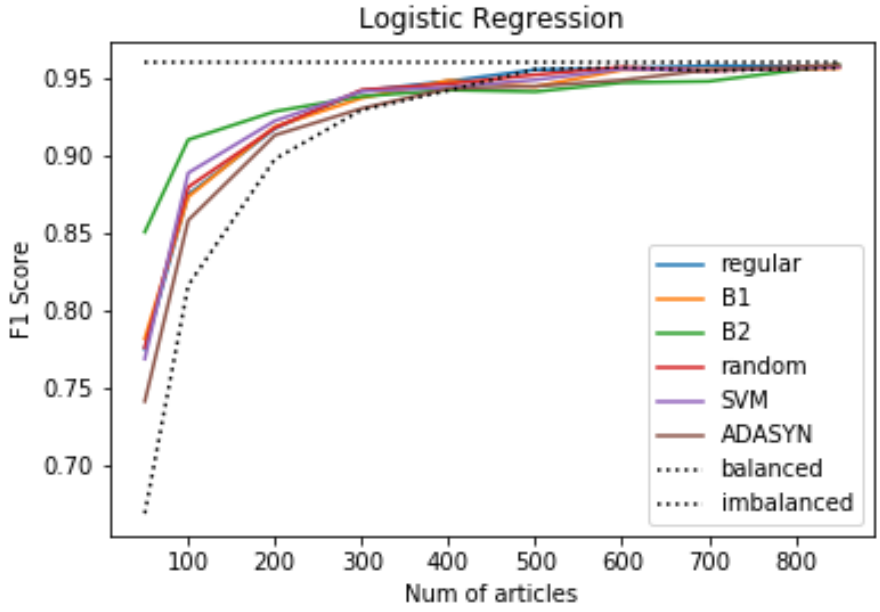
Table 10. F1 scores of classifiers when training data were resampled to balance. The p-value of F1 scores obtained through a t-test between the best performing SMOTE method and imbalanced data is provided. P-values below 0.005 are accented in bold. Best performing combinations are shaded in grey.

n		KNN	Logistic regression	Naïve Bayes	SVM
50	F1 score	0.899759	0.851004	0.852072	0.843208
	p-value	**0.0000**	**0.0000**	**0.0000**	**0.0000**
100	F1 score	0.893709	0.910590	0.899300	0.906937

	p-value	0.0000	0.0000	0.0000	0.0000
200	F1 score	0.893297	0.928866	0.928375	0.939964
	p-value	0.0000	0.0021	0.0002	0.0000
300	F1 score	0.902252	0.942741	0.923578	0.943568
	p-value	0.0000	0.0581	0.6491	0.0626
400	F1 score	0.905537	0.948848	0.935984	0.949446
	p-value	0.0081	0.1419	0.5120	0.0335
500	F1 score	0.911949	0.956079	0.938284	0.950575
	p-value	0.0979	0.9442	0.9672	0.6093
600	F1 score	0.916588	0.957438	0.938274	0.955268
	p-value	0.5951	0.9966	0.8407	0.1246
700	F1 score	0.922780	0.958364	0.941437	0.952282
	p-value	0.4700	0.2946	0.6021	0.7134
850	F1 score	0.926509	0.958985	0.941103	0.957197
	p-value	0.7401	0.2108	0.8251	0.8585

As seen in table 10, all classifiers benefited from the effects of SMOTE-resampled data in some imbalanced situations such as imbalanced ratios at 1:20, 1:9, and 2:9. Logistic regression, naïve Bayes and SVM had valid p-values compared to imbalanced F1 scores until original minority class supplied data were 200. KNN benefited until the imbalanced ratio reached 400. Also, all classifiers reached an F1 score above 0.8 while there were only 50 minority class cases. This shows that even in cases where F1 scores are not suffering drastically, resampling data with SMOTE will still be effective in providing a better performance.


KNN
F1 Score
0.90
0.85
0.80
0.75
0.70
0.65
100
200
300
400
500
600
700
800
Num of articles
regular
B1
B2
random
SVM
ADASYN
balanced
imbalanced


Logistic Regression
F1 Score
0.95
0.90
0.85
0.80
0.75
0.70
100
200
300
400
500
600
700
800
Num of articles
regular
B1
B2
random
SVM
ADASYN
balanced
imbalanced

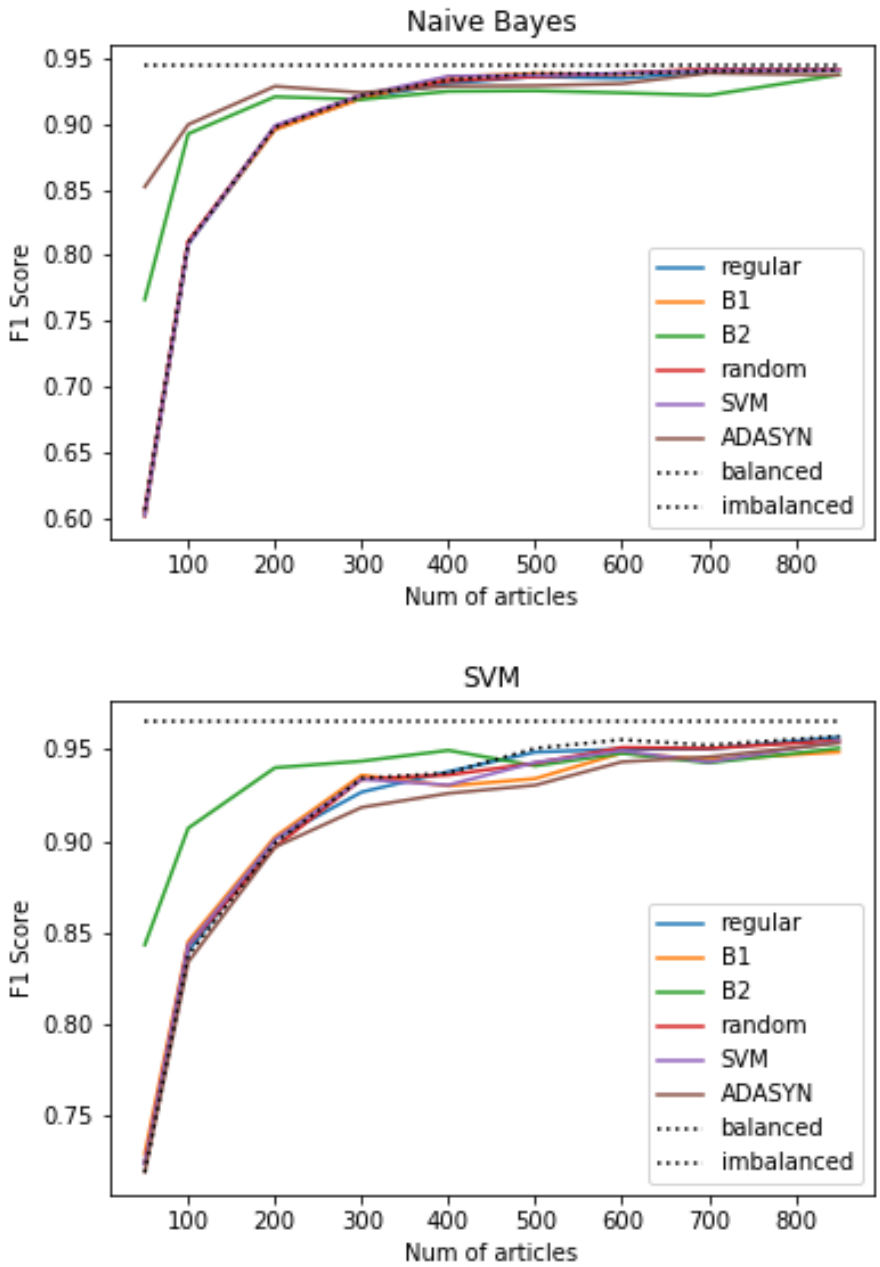


Figure 11. Performance of classifiers by the number of minority class training data and the use of resampling methods. 'Balanced' indicates the F1 score when all 900 majority and 900 minority class examples were used for training data. 'Imbalanced' indicates the F1 score before the data were resampled.

Each classifier reacted slightly differently depending on the resampling method that was applied as seen in figure 11, table 11 and table 12. SVM reacted better with borderline−SMOTE2. Statistical significance among the resampling methods were revealed until minority data reached 400. KNN was better paired with SVM−SMOTE, borderline−SMOTE and SMOTE. Compared to borderline−SMOTE2 and random oversampling, the other sampling methods showed statistical significance throughout the minority samples. Logistic regression reacted slightly better with borderline−SMOTE2 until minority data reached 200. Afterwards, there was no statistical significance among the resampling methods. Naïve Bayes performed best when it was paired with ADASYN. However, random Oversampling also showed statistical significance in the early stages. Overall, SMOTE methods proved to improve classifier performance in study 2 as well.

Table 11. F1 scores for resampled data shown by classifiers. Highest scores are shaded in grey. Resampling methods distinguishable among other resampling methods with a p−value below 0.005 are accented in bold. Actual p−values can be found in table 12.

SVM

n	regular	B1	B2	random	SVM	ADASYN	imbalanced
50	0.724037	0.728904	**0.843208**	0.724037	0.723239	0.719133	0.718779
100	0.839634	0.844982	**0.906937**	0.842755	0.843104	0.833745	0.837534
200	0.900873	0.902129	**0.939964**	0.896607	0.900677	0.896606	0.898816
300	0.926684	0.935887	**0.943568**	0.93365	0.933685	0.918304	0.934158
400	0.937787	0.93	**0.949446**	0.93601	0.93051	0.92596	0.937086

					2	9	
500	0.948509	0.933997	0.940977	0.942413	0.942844	0.930417	0.950575
600	0.950118	0.948038	0.947862	0.951013	0.949207	0.943182	0.955268
700	0.950138	0.944262	0.942587	0.950527	0.94305	0.945845	0.952282
850	0.956638	0.948571	0.950553	0.954781	0.954352	0.953434	0.957197

KNN

n	regular	B1	B2	random	SVM	ADASYN	imbalanced
50	0.893075	0.896797	0.8469	0.787427	0.899759	0.751735	0.64209
100	0.88699	0.8875	0.836396	0.845045	0.893709	0.803635	0.73866
200	0.892846	0.887705	0.836248	0.888728	0.893297	0.865409	0.85047
300	0.902252	0.887989	0.843593	0.902247	0.899701	0.889092	0.882168
400	0.901774	0.887668	0.841264	0.905537	0.900065	0.896308	0.890316
500	0.911949	0.888899	0.851925	0.907761	0.894962	0.899668	0.904852
600	0.916588	0.892793	0.849411	0.913521	0.90112	0.907673	0.91439
700	0.916666	0.89013	0.853961	0.919071	0.896323	0.900528	0.92278
850	0.922735	0.915571	0.881697	0.922077	0.917637	0.924557	0.926509

Logistic regression

n	regular	B1	B2	random	SVM	ADASYN	imbalanced
50	0.77542	0.782125	0.851004	0.77634	0.769129	0.741638	0.669201
100	0.87515	0.873675	0.91059	0.879999	0.88916	0.858535	0.816583
200	0.917857	0.919061	0.928866	0.91827	0.922812	0.913569	0.898144
300	0.94215	0.937516	0.938613	0.942741	0.941838	0.930791	0.9298
400	0.948337	0.948848	0.942725	0.946661	0.944458	0.943643	0.942379
500	0.955814	0.944894	0.941619	0.952569	0.949233	0.945134	0.956079
600	0.956547	0.955658	0.947487	0.957438	0.956232	0.949114	0.957427
700	0.958364	0.955702	0.948312	0.955215	0.955313	0.955666	0.955096
850	0.95788	0.95682	0.958985	0.95731	0.95773	0.95893	0.956826

	2			2	5	3	

Naïve Bayes

N	regular	B1	B2	random	SVM	ADASYN	imbalanced
50	0.603246	0.601768	0.76651	0.606121	0.602125	**0.852072**	0.60662
100	0.808725	0.809916	0.892257	0.810559	0.808385	**0.8993**	0.809593
200	0.895939	0.895399	0.920438	0.897559	0.898495	**0.928375**	0.89723
300	0.919562	0.919137	0.918349	0.921644	0.921778	0.923578	0.921212
400	0.930484	0.935222	0.924542	0.93249	0.935984	0.928465	0.933202
500	0.935734	0.938138	0.924995	0.935591	0.936815	0.928818	0.938284
600	0.934292	0.937642	0.923376	**0.938274**	0.938181	0.930369	0.937735
700	0.939246	0.940961	0.921524	**0.941437**	0.940417	0.938354	0.939796
850	0.941103	0.941103	0.937005	0.940622	0.940558	0.937013	0.940558

Table 12. P-values obtained by ANOVA among the resampling methods. P-values below 0.005 are accented in bold

	KNN	Logistic regression	Naïve Bayes	SVM
50	**0.0000**	**0.0000**	**0.0000**	**0.0000**
100	**0.0000**	**0.0000**	**0.0000**	**0.0000**
200	**0.0000**	**0.0031**	**0.0001**	**0.0000**
300	**0.0000**	0.1120	0.9151	**0.0005**
400	**0.0000**	0.6433	0.0858	**0.0010**
500	**0.0000**	0.0048	0.0005	**0.0008**
600	**0.0000**	0.0255	**0.0001**	0.1338
700	**0.0000**	0.1906	**0.0000**	0.2437
850	**0.0000**	0.9053	0.3545	0.1208

Discussion and Conclusion

Discussion

When training data from both studies were balanced at 1:1 with 900 articles for each class, study 1 showed an F1 score around 0.8 whiles Study 2 showed an F1 score around 0.9. The high scores indicate that the classifiers perform relatively well when data is balanced. The best classifiers for both studies were logistic regression and SVM.

When data were imbalanced at a ratio of 1:20, study 1 showed an F1 score below 0.2, even scoring as low as 0.02 in the case of KNN. However, all classifiers scored an F1 score above 0.6 in study 2. It is important to note that study 1 articles had a cosine similarity of 0.7 while study 2 articles had a cosine similarity of 0.4. This aligns with the findings of Jackowicz (2000) showing that complexity is a significant factor in classification tasks. Therefore, it can be said that especially when articles have high similarities, imbalanced data is even more harmful to the performance of classifiers.

Naïve Bayes was the best classifier for study 1 and SVM was the best classifier for study 2. It is difficult to conclude that there is a singular classifier that performs best in imbalanced data situations through this study alone. However, as a common pattern, when the level of imbalanced-ness lessened, F1 scores started to climb in both

studies, indicating that imbalanced levels were significant factors that affected classifier performance.

Table 13. A summary of the best performing resampling combinations by minority size number. Grey shades represent the best performing classifier in the minority size tier, bold represents statistical significance from using resampled data instead of implementing imbalanced data. Italics represents statistical significance from other resampling methods. 'SVM' represents SVM–SMOTE while 'RO' represents random oversampling. 'None' indicates that imbalanced data scored higher than resampled data.

	Study	KNN	Logistic regression	Naïve Bayes	SVM	F1 score
50	1	*SMOTE*	*B2*	*ADASYN*	*B2*	0.758
	2	*SVM*	*B2*	*ADASYN*	*B2*	0.8998
100	1	*SMOTE*	*B2*	*ADASYN*	*B2*	0.7939
	2	*SVM*	*B2*	*ADASYN*	*B2*	0.9106
200	1	*SMOTE*	*B2*	*ADASYN*	*B2*	0.8158
	2	*SVM*	*B2*	*ADASYN*	*B2*	0.94
300	1	*SMOTE*	B2	*ADASYN*	*B2*	0.8212
	2	*SMOTE*	RO	ADASYN	*B2*	0.9436
400	1	*SMOTE*	B2	*ADASYN*	*B2*	0.8208
	2	*RO*	B1	SVM	*B2*	0.9494
500	1	*B1*	B2	*ADASYN*	*B2*	0.8229
	2	*SMOTE*	None	B1	*SMOTE*	0.9561
600	1	*B2*	B2	*ADASYN*	None	0.8073
	2	*SMOTE*	RO	*RO*	RO	0.9574
700	1	*B2*	None	ADASYN	None	0.8058
	2	*None*	SMOTE	*RO*	RO	0.9584
850	1	SMOTE	B1	B2	SMOTE	0.8166
	2	*None*	B2	B1	SMOTE	0.959

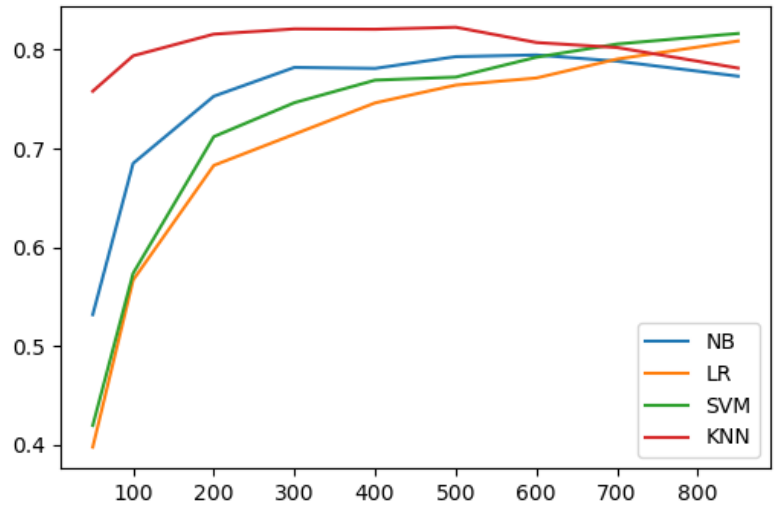


Figure 12. Performance of classifier after imbalanced data were resampled study 1. Respective methods for each data point can be found in table 13.

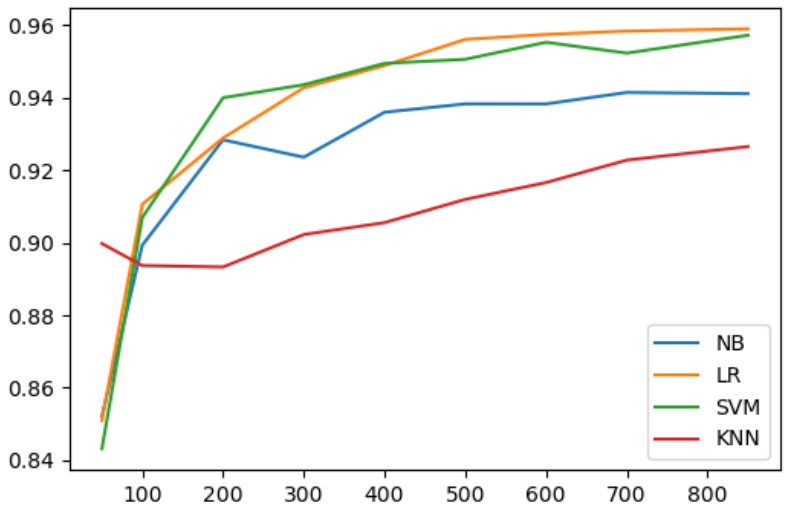


Figure 13. Performance of classifier after imbalanced data were resampled study 2. Respective methods for each data point can be found in table 13.

As it can be seen in table 13, after imbalanced data were resampled in study 1, F1 scores for data with a 1:20 ratio rose to 0.3 to 0.4 (K−nearest neighbors exceptionally 0.75). Also, the F1 scores reached 0.7 when ratios reached 1:5. In the case of study 2, F1 scores for resampled data with 1:20 ratio scored 0.8, and reached 0.9 when ratio reached 1:5. The best combinations were different per study. KNN classifiers with SMOTE performed best in study 1. Study 2 however had no uniform classifier and resampling method that performed the best. Therefore, it is difficult to say that there is a single best−performing classifier found through this study.

However, this study is able to show that the logistic regression and SVM classifiers reacted with most stability when paired with borderline−SMOTE2 throughout both studies. Naïve Bayes also reacted with stability when paired with ADASYN.

Also, this study revealed that the effect of SMOTE was most dramatic with the KNN classifiers for both studies. KNN classifiers did not show outstanding performance when data were severely imbalanced. However, after resampling methods were implemented, the classifier showed high F1 scores, regardless of levels of imbalance in training data. It is speculated that the SMOTE method's basis for generating new examples holds the same basis for KNN's decision algorithm, resulting in drastic improvement in its performance.

Conclusion

In conclusion, this study has shown that oversampling methods significantly improve the performance of classifiers in imbalanced situations. Logistic regression and SVM classifiers were stable classifiers regardless of balance levels, and responded best with the borderline-SMOTE2 method with most stability.

Also, complexity such as cosine similarities has shown to affect classifier performance in imbalanced situations. Depending on complexity levels, the starting points for classifier performances on imbalanced data were drastic. Also, this study has shown that that implementing SMOTE generally improves classifier performances regardless of complexity.

Text classification has high prospects in the real world, it is likely that the researcher comes across imbalanced data. Luckily, approaching data imbalance at the data level is comparatively approachable, and prove to be effective, even in Korean texts. As much as it has the possibility of wide implementation, there is still a lack of studies guiding users to deal with imbalanced data found in Korean text classification tasks. Thus, this paper hoped to provide a comparison of oversampling methods for real-world Korean text-classification researchers who are confronted with imbalanced data.

References

강지호, 김종찬, 이준혁, 박상성, 장동식. (2016). 특허 문서 분류 알고리즘 비교 연구. 『한국지능시스템학회 학술발표 논문집』, 26(1), 9-10.

강필성, 조성준. (2006). 데이터 불균형 해결을 위한 Under-Sampling 기반 앙상블 SVMs. 『대한산업공학회 춘계공동학술대회 논문집』, 2006.04, 291-298.

김경민, 장하영, 장병탁. (2014). 불균형 데이터 처리를 위한 과표본화 기반 앙상블 학습 기법. 『정보과학회 컴퓨팅의 실제 논문지』, 20(10), 549-554.

김병주. (2009). 문서분류에서의 SVM 및 나이브베이지안, EM 알고리즘의 특성 비교. 『대한전자공학회 2009년 하계종합학술대회』, 2009.7, 683-684.

김영인, 김선종, 김병철, 신범주. (2010). 클래스간 구간 중첩 데이터를 이용한 희소 클래스 분류 문제의 기법 비교. 『한국정보기술학회논문지』, 8(2), 137-144.

김윤석, 서영훈. (2013). 기계 학습을 이용한 한글 텍스트 감정 분류. 『한국엔터테인먼트산업학회 학술대회 논문집』, 2013.11, 206-210.

김은나, 이성건, 최종후. (2011). 목표 범주가 희귀한 자료의 과대표본추출에 대한 연구. 한국통계학회 『응용통계연구』, 24(3), 477-

484.

김은진, 허욱, 김병철, 엄일규, 김영인. (2011). 불균형 클래스 문제를 위한 보다 현실적인 데이터의 생성. 『한국정보기술학회논문지』, 9(11), 143-150.

김형도. (2013). 불균형 신용평가 데이터의 분류 향상을 위한 균형 교차검증. 『한국정보기술학회논문지』, 11(4), 169-175.

노지성, 최영식. (2005). 웹 문서 분류에 대한 연구 조사. 『한국인터넷정보학회 2005 정기총회 및 추계학술발표대회』, 6(2), 2005. 11, 413-417.

박은정, 조성준. (2014). KoNLPY: 쉽고 간결한 한국어 정보처리 파이썬 패키지. 『한글 및 한국어 정보처리 학술대회 논문집』, 26.

오장민, 장병탁, 김영택. (1999). SVM 학습을 이용한 다중 클래스 뉴스 그룹 문서 분류. 『한국정보과학회 1999년도 가을 학술발표논문집』, 26(2), 1999. 10, 60-62.

이공주, 윤보현. (2006). 정렬된 성경 코퍼스로부터 바꿔쓰기표현(paraphrase)의 자동 추출. *인지과학*, 17(4), 323-336.

이상순 최정민, 장근, 이병수. (2002). 문서분류 기법을 이용한 웹 문서 분류의 실험적 비교. 『한국정보과학회 2002년도 가을 학술발표논문집』, 29(2), 2002. 1, 154-156.

이상준, 전용주, 최두철, 김상우. (2014). 불균형 데이터를 대상으로 한 변형된 SVM. 『정보 및 제어 논문집』, 10, 51-52.

이영옥. (2000). 한국어와 영어간 구조의 차이에 따른 번역의 문제. *번역학연구*, 1(2), 47-76.

이영옥. (2002). 한국어와 영어간 언어구조의 차이에 따른 번역의 문제. 『번역학연구』, 3(1), 59-81.

이재식, 권종구. (2013). 불균형 데이터 집합의 분류를 위한 하이브리드 SVM 모델. 『지능정보연구』, 19(2), 125-140.

이채진, 박정술, 김준석, 백준걸. (2015). 다중공선성과 불균형분포를 가지는 공정데이터의 분류 성능 향상에 관한 연구. 『대한산업공학회지』, 41(1), 25-33.

이한수, 김성신. (2016). 클래스 불균형 데이터를 이용한 나이브 베이즈 분류기 기반의 이상전파에코 식별방법. 『한국정보통신학회논문지』, 20(6), 1063-1068.

임미영, 강신재. (2015). 한중 자동 문서분류를 위한 최적 자질어 비교. 『한국지능시스템학회 논문지』, 25(4), 386-391.

허준, 김종우. (2006). 의사결정나무 분석에서 불균형 자료의 분석 연구. 『대한산업공학회 춘계공동학술대회 논문집』, 2006. 05, 1244-1253.

홍의석, 박미경. (2016). 클래스 불균형 데이터를 사용한 심각도 기반 소프트웨어 품질 예측. 『한국컴퓨터정보학회논문지』, 21(4), 73-80.

Aggarwal, C. C., & Zhai, C. (2013). A survey of text classification

algorithms. In *Mining Text Data,* 163-222. Springer US.

Ailab, & Powers, D.M. (2011). Evaluation: from Precision, Recall and F-measure to Roc, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 2(1), (2011), 37-63.

Aridas, C.K., Lemaitre, G., & Nogueira, F. (2016). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Computing Research Repository*, abs/1609.06570.

Bai, Y., Garcia, E.A., He, H., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence),* 1322-1328.

Balbi, S. (2012). Beyond the curse of multidimensionality: High dimensional clustering in text mining. Italian journal of Applied Statistics. 22 (1). 53-63.

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python.*

Blondel, M., Brucher, M., Cournapeau, D., Dubourg, V., Duchesnay, E., Gramfort, A., Grisel, O., Michel, V., Pedregosa, F., Prettenhofer, P., Passos, A., Perrot, M., Thirion, B.,

Varoquaux, G., VanderPlas, J., & Weiss, R. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Chawla, N. V. (2002). SMOTE: Synthetic Minority Over-sampling Technique. In *Journal of Artificial Intelligence Research*, 16, 321-357

Chawla, N. V. (2005). Data Mining for Imbalanced Datasets: An Overview. In O. Maimon & L. Rokach (ed.), *The Data Mining and Knowledge Discovery Handbook*, 853-867. Springer.

Chawla, N.V. (2003). C4.5 and Imbalanced Data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the ICML' 03 Workshop on Class Imbalances*.

Chawla, N.V., Japkowicz, N., & Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. In *Special Interest Group on Knowledge Discovery and Data Mining Explorations*, 6, 1-6.

Fawcett, T., & Provost, F.J. (2001). Robust Classification for Imprecise Environments. *Machine Learning*, 42, 203-231.

Friedman, J.H., Hastie, T.J., & Tibshirani, R. (2009). *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition*. Springer series in statistics.

Grobelnik, M., & Mladenic, D. (1999). Feature Selection for Unbalanced Class Distribution and Naive Bayes. In *International Conference on Machine Learning.*

Han, H., Wang, W.Y., Mao, B.H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In Huang, D.S., Zhang, X.-P., Huang, G.-B. (ed.), *ICIC* 2005, Part I, 878-887.

Japkowicz, N. (2000). Learning from Imbalanced Data Sets: A Comparison of Various Strategies. *Association for the Advancement of Artificial Intelligence*.

Johnson, S. (2013, May 29). Lexical facts. *The Economist*. Retrieved from: https://www.economist.com/blogs/johnson/2013/05/vocabulary-size

Jones, K. (1972) " A statistical interpretation of term specificity and its application in retrieval ", *Journal of Documentation*, 28(1), 11-21.

Krawczyk, Bartosz. (2016). Learning from Imbalanced Data: Open Challenges and Future Directions. *Progress in Artificial Intelligence*, 5(4), 221-232.

Kubat, M. & Matwin, S. (1997). Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In D. H. Fisher (ed.),

International Conference on Machine Learning, 179-186.

Lee, D.G., Rim, H.C., Yook, D.S. (2007). Automatic Word Spacing Using Probabilistic Models Based on Character n-grams. *IEEE Intelligent Systems*. 22(1). 28035.

Ling, C., & Li, C. (1998). Data Mining for Direct Marketing Problems and Solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (KDD-98) New York, NY. AAAI Press.

Liu, A. (2004). The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets. Unpublished master's thesis, The University of Texas at Austin, Austin, Texas.

Liu, X., & Yang, Y. (1999). A Re-Examination of Text Categorization Methods. *Special Interest Group on Information Retrieval.*

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250, 113-141.

Lu, Y., Nakazawa, T., Kurohashi, S. (2015). Korean-to-Chinese Word Translation using Chinese Character Knowledge. *Proceedings of MT Summit.* 15(1). 256-269.

Nguyen, H., Cooper, E., Kamei, K. (2009). Borderline Over-sampling

for Imbalanced Data Classification. In *Fifth International Workshop on Computational Intelligence & Applications*. IEEE SMC Hiroshima Chapter, Hiroshima University, Japan, November 10, 11 & 12, 2009.

Provost, F.J., & Weiss, G.M. (2003). Learning When Training Data are Costly: The Effect of Class Distribution on Tree Induction. J. Artif. *Intell. Res. (JAIR)*, 19, 315-354.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computer Survey*, 34, 1-47.

Tsoumakas, G. & Katakis, I. (2007). Multi Label Classification: An Overview. *International Journal of Data Warehouse and Mining*, 3, 1-13.

Wonji Lee, Chi-Hyuck Jun. (2014). Comparison of data pre-processing techniques for relaxing class imbalance problem. 『대한산업공학회 추계학술대회 논문집』, 2014.11, 2373-2383.

Appendix

Categorization of Naver News[1]

Category	Subcategory
경제(Economy)	***금융(Finance)**
	***증권(Stock)**
	산업/재계(Industry)
	중기/벤처(Venture business)
	부동산(Real Estate)
	글로벌경제(Global Economy)
	생활경제(Living Economy)
	경제 일반(Economy General)
	속보(Breaking News)
사회(Society)	사건사고(Incidents)
	***교육(Education)**
	노동(Labor)
	언론(Media)
	***환경(Environment)**
	인권/복지(Welfare)
	식품/의료(Health)
	지역(Local)
	인물(Persons)
	사회 일반(Society General)
	속보(Breaking News)
…	…

[1] http://news.naver.com/

* Starred categories were used in this study.

Example Articles

(Education)

한승주 총장서리 "기여입학제는 진지한 논의 필요"서울대 이장무 총장에 이어 고려대 한승주 총장서리와 연세대 정창영 총장이 정부의 '3불정책(본고사, 기여입학제, 고교등급제 실시 금지)' 중 기여입학제를 제외한 고교등급제와 본고사를 허용해야 한다는 입장을 밝혔다. 미 조지타운대에서 열리는 국제동북아포럼에 참석하기 위해 워싱턴에 머물고 있는 한 총장서리는 29일 본보와 전화 인터뷰를 갖고 "현실적으로 고교간 학력차가 존재하고 논술시험 등 본고사와 같은 평가 방식이 시행되는 현실에서 (교육인적자원부가) 무조건 불허하는 것은 문제가 있다"고 지적했다. 그는 "입학생의 출신고교 성적을 산출한 통계를 보면 우수한 졸업생을 많이 배출하는 고교들이 나타나기 마련"이라며 "서열화, 등급화 하겠다는 것이 아니라 이미 나타난 현실을 고려해 합리적인 방법을 찾아야 한다는 뜻"이라고 말했다. 그는 이어 "대학이 자율적으로 학생들을 평가하는 잣대로 활용하는 현행 논술시험이 일종의 본고사가 아니냐"며 "사실상 시행되고 있는 제도를 흑백논리로 보는 것은 현실성이 없다"며 본고사 찬성 입장을 밝혔다. 하지만 기여입학제에 대해선 "교육적인 측면에서 찬반이 있을 수 있고 위화감 조성 등 부정적인 부분도 무시할 수 없다"며 "3불이라는 교육정책에 도식적으로 집어 넣어 판단하기 보다는 사회적 문제로 진지한 논의를 통해 검토하는 것이 옳다"고 덧붙였다. 정 총장도 "고교 간 차이가 있는 것이 사실이며 입학시험은 대학자율에 맡겨야 한다"고 고교등급제와 본고사에 사실상 찬성의사를 밝혔다. 그는 "다만 기여입학제는 국민이 허용하지 않는 상황이니 아직 시기상조"라고 말했다. 이현정 기자

(Environment)

예비군 훈련장의 훼손된 나무들"꽃가루 날린다" 30여그루 나무껍데기 벗겨
(담양=연합뉴스) 이세원 기자 = 전남 담양의 한 예비군 대대에서 영내 나무 수십 그루를 고사(枯死)시킬 목적으로 훼손해 논란이 일고 있다. 30일 육군 모 사단과 예비군 훈련 참석자 등에 따르면 전남 담양의 모 예비군 대대가 최근 영내 은사시나무 약 30그루를 고사시키기 위해 껍데기를 벗겼다. 지름이 20-40㎝가량인 이들 나무는 밑둥에서 높이 1-2m가량이 껍데기가 벗겨진 채 방치돼 있다. 부대 측에서는 "은사시나무에서 꽃가루가 날려 식사에 방해가 된다는 예비군 훈련 참석자들의 민원제기에 따라 식사 장소 주변에 있는 은사시나무 7그루를 고사시키기 위해 껍데기를 벗겼다"고 해명했지만 현장 방문자를 통해 확인한 결과 30그루 가까운 나무가 훼손됐다. 이 밖에도 일부 참석자는 부대 측에서 훈련장 확보를 위해 소나무를 대량으로 벤 흔적이 있다고 문제제기를 하고 있다. 이에 대해 부대 측은 "관목 등을 솎아내는 목적의 벌채는 허가나 신고 없이 소유주의 동의에 따라 할 수 있다\'는 관할 군의 답변에 따라 40-50 그루를 벌채했을 뿐"이라고 해명했다. 산림자원의 조성 및 관리에 관한 법률 시행규칙에서는 \'솎아베기 대상 임지로서 평균 가슴높이의 지름이 20㎝ 이하인 입목을 솎아내기 위하여 벌채 또는 굴취하는 경우\'를 신고에 따른 벌채 대상으로 규정하고 있어 관련

법규를 위반해 산림 훼손을 했다는 의혹도 낳고 있다. 한편 담양군 관계자는 "해당 부대에서 화재 방지 목적으로 사격장 주변의 나무를 베고 싶다\'고 문의해 병해충 방지 목적이나 솎아내기는 산 주인이 할 수 있다\'고 답했으며 은사시나무 관련 언급이나 훈련장 확보를 위한 소나무 벌채 등에 대한 언급은 없었다"고 말했다.

(Finance)

윤용로 행장 강조 “최고의 전략은 현장의 소리를 반영한 생동감 넘치는 전략입니다. 이는 곧 모든 전략은 현장에 뿌리를 두고 수립되어야 함을 말합니다.” 지난 해 7회의 타운미팅을 통해 136건 건의사항을 접수하고, 이 중 100건을 제도에 반영했다는 윤용로 기업은행장은 올해도 어김없이 현장을 직접 발로 뛰겠다는 포부를 밝혔다. 현장영업을 통해 기업의 경영전반을 이해하고 있어야만 올바른 의사결정과 신속한 중기지원이 가능하다고 믿기 때문이다. 윤 행장은 국책은행으로써 기업은행은 올해 중기금융지원에 총력을 다해 경주할 것임을 강조했다. 2009년의 경영슬로건을 ‘이제 중소기업이다’라고 정한 것도 이같은 기업은행의 다짐을 공고히 하기 위함이다. 윤 행장은 “기업은행은 그간 쌓아온 중기금융 노하우, 다양한 중기관련 정보, 신용평가시스템 등 수많은 강점을 보유하고 있다고 자부한다”며 “경기가 어려우면 가장 먼저, 가장 큰 어려움을 겪는 것이 바로 중소기업이며, 이를 덜어주는 것이 우리의 책무”라고 말했다. 그는 지금과 같은 위기에 건실한 중소기업들이 버틸 수 있도록 특히 신규업체 발굴에 힘쓸 예정이다. “은행들의 중기대출을 보면 건당 대출금액이 점점 많아지고 있는데, 이는 은행이 안전성을 위해 한 곳에 많은 금액을 지원하기 때문입니다. 앞으로는 지원 금액을 단순히 확대하는 것에 그치지 않고 되도록 많은 기업에 지원이 가도록 할 것입니다.” 윤 행장은 ‘2009년 상반기 전국 영업점장 회의’에서도 ‘위기는 고통스럽지만 반드시 끝난다’는 긍정적인 마인드로 올해를 중소기업이 재도약하는 해로 만들겠다고 밝혔다. 정지연 기자/

(Stock)

복제약 美 판매 기대로한달 만에 32%이상 상승승인 발표하자 7% 떨어져바이오시밀러(복제약)의 판매승인 기대감에 지난해 하반기부터 무섭게 상승했던 셀트리온이 정작 판매승인 발표에는 차익실현 매물에 급락했다. 6일 셀트리온은 코스닥시장에서 전날 대비 7.31%(8,700원) 떨어진 11만300원에 거래를 마감했다. 전문가들은 바이오시밀러 제품인 램시마가 미국 식품의약국(FDA) 판매승인을 획득했다는 소식으로 차익실현 매물이 급증한 것을 급락의 원인으로 봤다. 셀트리온은 지난해 말 8만4,500원에서 불과 한 달 만에 32% 이상 상승하는 등 FDA 판매승인에 대한 기대감으로 연초에 무서운 상승세를 보였다. 이 때문에 증권가에서는 판매승인 등에 대한 기대감이 이미 주가에 반영돼 오히려 FDA 승인 후 주가 상승은 크지 않을 것으로 전망하고 있다. 실제로 이날 셀트리온의 거래량은 414만3,409주로 전날(73만5,885건)보다 562.25%나 많았다.이날 셀트리온은 “CT-P13(램시마)은 미국 내 최초의 항체 바이오시밀러 허가 제품으로 글로벌 시장 최고 규제기관으로부터 기술력을 인정받았다”며 “이번 FDA 최종 허가 획득에 따라 미국 내 독점판매권자인 화이자사와의 협의를 통해 미국 내 판매시기를 최종 결정할 예

정”이라고 공시했다. 이승호 NH투자증권 연구원은 “미국 공중보건 서비스법이 규정하고 있는 복제약 시판 고지 의무에 따르면 복제약 개발자는 바이오신약 개발자에게 시판 180일 전에 시판 사실을 고지해야 한다”며 “이에 따라 오는 10월 미국에서 시판될 가능성이 있다”고 분석했다. 한편 일각에서는 공매도에 대한 트라우마도 급락에 영향을 미친 것으로 보고 있다. 셀트리온은 올해부터 지난 5일까지 공매도량이 794만1,229주에 달해 전체 코스닥 종목 중 가장 많은 공매도량을 기록하는 등 공매도로 인한 주가하락 논란에 휩싸였었기 때문이다. /김연하기자

Classifier Parameters

Logistic regression

Default parameters were set for the logistic regression classifier throughout the experiment.

Parameter	Value
Cs	10
Class_weight	None
CV	None
Dual	False
Fit_intercept	True
Intercept_scaling	1.0
Max_iter	100
Multi_class	OVR
N_jobs	−1
Penalty	L2
Random_state	None
Refit	True
Scoring	None
Solver	LBFGS
Tol	0.0001
Verbose	0

Naïve Bayes

Default parameters were set for the multinomial naïve Bayes classifier throughout the experiment.

Parameter	Value
Alpha	1.0
Fit_prior	True
Class_prior	None

K−nearest Neighbor (KNN)

The KNN classifier allows size adjustments on the neighborhood. The number of neighbors was adjusted throughout the experiment among a sample population of 3, 5, 7, and 9. Below are the parameters selected through the cross−validation process.

	Minority Size	SMOTE method	Average	Mode
		Study 1		
Balanced	900	–	7	7
Imbalanced	50	–	7.4	7
	100	–	6.4	7
	200	–	6.2	7
	300	–	7.2	9
	400	–	7.6	9
	500	–	7.6	9
	600	–	7.2	9
	700	–	8.2	9
	850	–	7.6	7
Resampled	50	ADASYN	4.8	3
		B1	3	3
		B2	3	3
		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	100	ADASYN	3.2	3
		B1	3	3
		B2	3	3
		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	200	ADASYN	3	3
		B1	3	3
		B2	3	3
		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	300	ADASYN	3	3
		B1	3	3
		B2	3	3
		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	400	ADASYN	3	3
		B1	3	3
		B2	3	3

		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	500	ADASYN	3	3
		B1	3	3
		B2	3	3
		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	600	ADASYN	3	3
		B1	3	3
		B2	3	3
		Random Oversampling	4	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	700	ADASYN	7	9
		B1	3	3
		B2	3	3
		Random Oversampling	5.8	3
		SMOTE	3.2	3
		SVM-SMOTE	3.8	3
	850	ADASYN	7.6	7
		B1	5.4	3
		B2	3.2	3
		Random Oversampling	8	9
		SMOTE	6.4	7
		SVM-SMOTE	7	9
		Study 2		
Balanced	900	-	7	7
Imbalanced	50	-	3.8	3
	100	-	4.4	5
	200	-	6	5
	300	-	8.2	9
	400	-	6.8	7
	500	-	8.8	9
	600	-	8.6	9
	700	-	8.4	9
	850	-	7.4	9

Resampled	50	ADASYN	4.4	3
		B1	3	3
		B2	3	3
		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	100	ADASYN	4.2	3
		B1	3	3
		B2	3	3
		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	200	ADASYN	3.8	3
		B1	3	3
		B2	3	3
		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	300	ADASYN	3	3
		B1	3	3
		B2	3	3
		Random Oversampling	3	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	400	ADASYN	4.6	3
		B1	3	3
		B2	3	3
		Random Oversampling	4	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	500	ADASYN	4	3
		B1	3	3
		B2	3	3
		Random Oversampling	4	3
		SMOTE	3	3
		SVM-SMOTE	3	3
	600	ADASYN	3.6	3

		B1	3	3
		B2	3	3
		Random Oversampling	3.8	3
		SMOTE	3.2	3
		SVM-SMOTE	3	3
	700	ADASYN	3	3
		B1	3	3
		B2	3	3
		Random Oversampling	7	7
		SMOTE	3	3
		SVM-SMOTE	3	3
	850	ADASYN	6.4	9
		B1	3	3
		B2	3	3
		Random Oversampling	7	9
		SMOTE	4.6	3
		SVM-SMOTE	46	3

SVM

C and kernel methods were the only parameter for the SVC classifier that was adjusted throughout the experiment. Kernels refer to the kernel tricks used to map and calculate the distance of each data sample in feature space. This study cross-validated the best parameters between the options of the RBF kernel and the Linear kernel. C refers to the penalty parameter, determining how smooth the surface of the hyperplane will be, in respect of error. This study cross-validated the best C value among the choices 1, 10, 100, and 1000. The table represents the selected parameters through the cross-validation process.

	Minority Size	SMOTE method	C		kernel
			Average	Mode	Mode
		Study 1			
Balanced	900	-	1	1	Linear
Imbalanced	50	-	600	1000	RBF
	100	-	400	1	RBF
	200	-	300	1	Linear

	300	–	400	1	Linear
	400	–	100	1	Linear
	500	–	100	1	Linear
	600	–	200	1	Linear
	700	–	400	1	Linear
	850	–	200	1	Linear
Resampled	50	ADASYN	19.0	10.0	Linear
		B1	703.0	1000.0	RBF
		B2	900.1	1000.0	RBF
		Random Oversampling	604.0	1000.0	RBF
		SMOTE	604.0	1000.0	RBF
		SVM-SMOTE	604.0	1000.0	RBF
	100	ADASYN	46.0	10.0	Linear
		B1	226.0	10.0	Linear
		B2	800.2	1000.0	RBF
		Random Oversampling	28.0	10.0	Linear
		SMOTE	28.0	10.0	Linear
		SVM-SMOTE	136.0	10.0	Linear
	200	ADASYN	73.0	100.0	Linear
		B1	55.0	10.0	Linear
		B2	900.1	1000.0	RBF
		Random Oversampling	64.0	100.0	Linear
		SMOTE	73.0	100.0	Linear
		SVM-SMOTE	55.0	10.0	Linear
	300	ADASYN	64.0	100.0	Linear
		B1	73.0	100.0	Linear
		B2	1000.0	1000.0	RBF
		Random Oversampling	64.0	100.0	Linear
		SMOTE	64.0	100.0	Linear
		SVM-SMOTE	163.0	100.0	Linear
	400	ADASYN	64.0	100.0	Linear
		B1	253.0	100.0	Linear
		B2	900.1	1000.0	RBF
		Random Oversampling	145.0	10.0	Linear
		SMOTE	154.0	100.0	Linear
		SVM-SMOTE	154.0	100.0	Linear
	500	ADASYN	55.0	10.0	Linear
		B1	226.0	10.0	Linear

		B2	602.2	1000.0	RBF
		Random Oversampling	64.0	100.0	Linear
		SMOTE	235.0	10.0	Linear
		SVM-SMOTE	235.0	10.0	Linear
	600	ADASYN	405.1	10.0	Linear
		B1	522.1	1000.0	Linear
		B2	900.1	1000.0	RBF
		Random Oversampling	424.0	10.0	Linear
		SMOTE	424.0	10.0	Linear
		SVM-SMOTE	313.3	1.0	Linear
	700	ADASYN	500.5	1.0	Linear
		B1	502.3	1000.0	Linear
		B2	800.2	1000.0	RBF
		Random Oversampling	501.4	1000.0	Linear
		SMOTE	602.2	1000.0	RBF
		SVM-SMOTE	800.2	1000.0	RBF
	850	ADASYN	200.8	1.0	Linear
		B1	400.6	1.0	Linear
		B2	300.7	1.0	Linear
		Random Oversampling	500.5	1.0	Linear
		SMOTE	200.8	1.0	Linear
		SVM-SMOTE	300.7	1.0	Linear
Study 2					
Balanced	900	-	100	100	RBF
Imbalanced	50	-	600	505	Linear
	100	-	400	700	RBF
	200	-	300	401	Linear
	300	-	400	200	Linear
	400	-	100	321	Linear
	500	-	100	40	Linear
	600	-	200	60	RBF
	700	-	400	120	Linear
	850	-	200	142	Linear
Resampled	50	ADASYN	901	1000	RBF
		B1	603	1000	RBF
		B2	420	1	RBF
		Random Oversampling	901	1000	RBF
		SMOTE	901	1000	RBF

		SVM-SMOTE	405	10	Linear
	100	ADASYN	307	10	Linear
		B1	403	1000	Linear
		B2	510	1000	RBF
		Random Oversampling	503	1000	Linear
		SMOTE	603	1000	RBF
		SVM-SMOTE	403	1000	Linear
	200	ADASYN	406	10	Linear
		B1	603	1000	RBF
		B2	360	100	RBF
		Random Oversampling	505	10	Linear
		SMOTE	503	1000	Linear
		SVM-SMOTE	404	10	Linear
	300	ADASYN	406	10	Linear
		B1	208	1	Linear
		B2	270	100	RBF
		Random Oversampling	304	10	Linear
		SMOTE	404	10	Linear
		SVM-SMOTE	601	1000	RBF
	400	ADASYN	307	10	Linear
		B1	602	1000	RBF
		B2	1	1	Linear
		Random Oversampling	501	1000	Linear
		SMOTE	202	1	Linear
		SVM-SMOTE	604	1000	RBF
	500	ADASYN	406	10	Linear
		B1	703	1000	RBF
		B2	301	1	Linear
		Random Oversampling	701	1000	RBF
		SMOTE	200	1	Linear
		SVM-SMOTE	602	1000	RBF
	600	ADASYN	504	1000	Linear
		B1	504	1000	Linear
		B2	500	1	Linear
		Random Oversampling	312	1	Linear
		SMOTE	312	1	Linear
		SVM-SMOTE	403	1000	Linear

	700	ADASYN	503	1000	Linear
		B1	503	1000	Linear
		B2	701	1000	RBF
		Random Oversampling	211	1	Linear
		SMOTE	200	1	Linear
		SVM−SMOTE	204	1	Linear
	850	ADASYN	143	100	Linear
		B1	116	10	Linear
		B2	511	1000	RBF
		Random Oversampling	222	1	Linear
		SMOTE	142	100	Linear
		SVM−SMOTE	222	1	Linear

국문 초록

서이레

협동과정 인지과학전공

서울대학교 대학원

텍스트 주제 분류를 진행 할 때, 자료의 불균형 때문에 생기는 문제를 해결하기 위해 많은 연구가 진행되어 왔다. 그 중 하나로 다양한 오버 샘플링(oversampling) 기법이 연구되고 있으나, 한국어 텍스트 분석에 오버 샘플링 기법이 도입된 연구는 부족하다. 본 연구는 인터넷 신문기사를 지도 학습 방법으로 주제별로 분류하고자 할 때 생길 수 있는 불균형 자료 문제에 다양한 오버 샘플링 방법을 도입해, 각 성능을 비교하여 한국어 기사 주제분류시 가장 큰 효과를 보이는 기법을 실험한다. 실험 결과 support vector machine 과 logistic regression 은 Borderline-SMOTE2, naïve Bayes 는 ADASYN 과의 조합이 가장 높은 결과를 가져다 준다.

주제어: 불균형 자료, 오버 샘플링, 지도 학습, 텍스트 분석, 토픽 분류, SMOTE

학번: 2013-22801