Imbalanced Dataset Exploration by Various Resampling Approaches and Genetic Algorithm Feature Selection Study Case: KKBox Churn Prediction

Ludia Eka Feri
201683475
Email: ludia@pusan.ac.kr

Hani Ramadhan
201793254
Email: hani042@pusan.ac.kr

Yoga Yustiawan
201783629
Email: yoga017@pusan.ac.kr

***Abstract*—In customer relationship management, customer retention treatment is preferable than customer acquisition as it consumes less cost. One of customer retention cases is churning user prediction, which is considered minority towards non-churners. The goal of this work is to investigate the characteristic of churn prediction of KKBox imbalanced dataset by combining user behavior data and static demography data, then applying weighted classifier, resampling methods, and feature selection approaches as improvement efforts. The timestamped user logs and transaction data were transformed using longitudinal behavior transformation to capture the user behavior in a specific time window interval. Combination of datasets by differentiating time window intervals and the static dataset were also examined. The results of the experiments show that combination of music playing logs, transaction logs, and static member data yields log loss of 0.242 in validation log loss function using regular random forest classifier. Another scenario with the aim to compare the performance of imbalanced classifier which is weighted random forest, however, resulted worse than regular random forest. Resampling methods worked better to improve the imbalanced dataset classification by 0.224 log loss value. Feature selection as dimension reduction method to simply remove the noisy features did not enhance the original classification of full features set of the dataset. Thus, it is important to investigate the characteristic of other classifiers for imbalanced dataset as well as the extensions of feature extraction approaches in the future. In WSDM 2018 Kaggle Competition, this approach yields 0.188 log loss value, which ranked 423 out of 575 teams.**

***Index Terms*—churn prediction, imbalanced dataset, resampling, feature selection, random forest**

1. Introduction

Customer Relationship Management (CRM) strategies hold important role in a company to stay competitive in the market. In the term of CRM, customer acquisition and customer retention are important tasks. Between those tasks, customer retention is preferable to customer acquisition since it has lower cost and is more profitable to do (as cited in [1]). Churn itself is defined as the possibility of a customer leaves out the company's service for other company, just discontinue the contract, or the unpredictable one: leave the subscription for non-subscription reasons (moving to different regions, financial problem) [2]. Thus, to predict this set of customers is important in CRM strategies.

KKBox is a online music player service based in Taiwan. Its service is based on user subscription, thus churning subscribers case is relevant. KKBox restrict its criteria of churner to ones who did not apply for subscription 30 days after their subscription expire date [3]. As for example (while specified in the case description also), a person did not apply a subscription on Feb 1st 2017 (his/her expire date) until Mar 3rd 2017 (30 days after expire date). Then, that person will be counted as 'churner' in March 2017.

Several works around churn prediction has been developed, especially in data preparation as preprocessing before actually getting into training to take care imbalanced class (churner and non-churner) dataset [4]. Several data preparation algorithms [5] include sampling and value transformation techniques. SMOTE sampling approach lead to a good performance for comprehensible models [6], yet engineers in eBay [7] implemented approach of distributed churner and non-churner class towards feature space distribution resampling to achieve good distribution approximation. Another sampling approach as implemented in improved balanced random forests [8] also studied in banking churner dataset. Several type of classifiers also implemented to predict churning such as decision tree [7] and random forest [8], Support Vector Machines (SVM) [1], [6], and Neural Network (NN) [9].

Feature selection to produce better churn prediction had been studied in several areas, especially telecommunication. GA on feature selection [10] and PCA, F-score, Fishers ratio and mRMR based feature selection [11] proved that selecting important features were beneficial to increase the performance of classification towards telecom dataset.

Originally, the task of the competition as described in Kaggle is to predict is 'Will customer X be a churner next month?' given historical user behavior and static dataset. However, this work proposes the exploration of KKBox Churn Prediction dataset characteristic using several resampling methods and Random Forest as classifier. In parallel, using GA, feature selection effects towards the classification performance will be studied. Also, data preparation technique for user behavior related dataset and missing value treatment will be discussed in this work.

TABLE 1. SHORT DESCRIPTION OF PROVIDED DATASET TABLES

Table Name	Description	Type	Rows
Train	List of churner and non-churner members of March 2017	-	970,960
Test	List of to be predicted churner and non-churner members of April 2017	-	970,471
Members v3	List of registered members, some data were outliers (missing gender, invalid age)	Static Demographic	6,769,473
Transaction v2	Transaction logs of the members	User behavior	1,431,009
User_logs v2	Playing song activity logs of the members on each day	User behavior	18,396,362
User_logs (old)	Users playing activity since 2015 (additional)	User behavior	393,227,724

TABLE 2. CHURNING CLASS DISTRIBUTION IN TRAINING SET

is_churn	Num. of Observation	Percentage (%)
0 (Not churner)	883,630	91.01
1 (Churner)	87,330	8.99

2. Material and Method

2.1. Dataset Description

KKBox provided several tables in csv format for dataset in churn prediction case. Short summary of the dataset tables is depicted in Table 1. Considering the number of rows of each table, the complexity of the memory should be considered when performing the churn prediction. As example, the most space consuming table in this dataset is *user_logs (old)* table which size reaches 28.4 GB. Each table's short summary is presented in the Churn Prediction data explanation in Kaggle website [3]. The scarcity of churner in the training dataset is described in Table 2. Thus, the dataset has imbalanced class distribution which very likely for churning users case studies. The treatment of imbalanced dataset will be explained later subsection.

2.2. Data Preprocessing Technique

2.2.1. Missing value and outliers treatment. Most dataset found in real world tends to be large, noisy, messy, and unstructured, while the KKBox churn prediction dataset also falls into those categories. Whereas already described in 1, Table members has some outliers. There are users (nearly 1%) whose ID exist in training set and test set but their demographic nor behavior data cannot be found in any tables: members, transaction, user logs, and old user logs. This leads to extensive exploration of missing data treatment.
There are approaches to handle the missing value of a feature:

1) Fill out the missing/null value to zero
2) Fill out the missing/null value as common value in the feature
3) Remove observations that has missing values
4) Remove features that has most missing values

The first, second, and third options are widely used while the fourth option is risky approach since those missing value may lead to the 'rarity' of the event. However, first option would not be applicable to feature that has categorical value or ordinal value, of which value 0 would be unavailable (there is no 0 in gender of 'male' and 'female'). Thus, it would be preferable to use the second option, but it may lead to bias other independent observations which value is close to common value. The common value can be generated by simple aggregation such as average for numerical value, or mode for categorical value. Third option tends to give cleaner observation even though it will increase the bias. In this work, the first, second, and third options will be used alternatively to handle the missing value as well as missing users' information.

2.2.2. Additional feature extraction. The provided churn prediction dataset are raw log files which only preprocessed lightly as input to other program input. Hence, there are few to-be-extracted hidden features from the log files. As example from a single transaction date feature, we may see which day of week, week, or month of the time the user did the transaction. Even though there were endless additional feature extraction possibilities, this study only extracted features as described in Table 3.
As seen in Table 3, hot encoded features are included. Hot encoded features are l boolean features which processed from a single categorical value such as gender, city, or registration method, where l is distinct values of the feature. They are made because of Random Forest Classifier in Python cannot handle categorical values directly. This hot encoded feature application lead to highly sparse matrix of features.

2.2.3. Longitudinal behavior Transformation. Predicting future churning subscribers could be performed by using historical data of user behavior and static user data. This historical data is called longitudinal behavioral data. Chen [1] proposed ensemble SVM method to predict churning users by time-series transformation longitudinal behavioral data. However, because of KKBox dataset timestamp is very finely grained to a single day per user in user logs and transaction table, we propose another approach of aggregation and time windowed transformation of longitudinal behavioral data.
Suppose behavioral data of daily music playing logs X has $N \times M$ dimension where N represents the number of rows, where each row is represented as $x_{1,2,...,n}$ feature vectors and M is number of features, of which represented by $p_{1,2,...,m}$. Each single element is represented by $x_{i,j}$, where i represents i-th row and j-th feature. Each row x_i

TABLE 3. EXTRACTED ADDITIONAL FEATURES

Features	Table	Description
Hot-encoded gender	members	2 boolean features
Hot-encoded registered_via	members	17 boolean features
Hot-encoded city	members	21 boolean features
Registered days	members	number of days since member's registered day until March 31st 2017
week	transaction	week of the month when member did a transaction, it may lead to most common week a member did/cancel a transaction weekly
day of week	transaction	day of week when member did a transaction
month	transaction	month of year when member did a transaction used as grouping criteria
contract length	transaction	the subscription length between its transaction date and membership expiration date

has identification of song playing date d_i, thus an interval of time window D_t longer than a single day, such as 3-days, weekly, or monthly, can be applied as grouping aggregator. This aggregator will group all the features of specified time window to be new features of the interval. A new transformed longitudinal behavioral data X' will have $N \times KM$ dimension, where K is number of aggregated groups.
Determining K is specified as previous K time window D_t before a date. For example, a user behavior dataset is transformed to contain 3 previous months ($D_{monthly}$,$K = 3$) of user behavior data. This dataset has features p_1 ... p_m. These features will be grouped by the time window interval $D_{monthly}$ The new transformed dataset will contain $N \times 3M$ features p' where each M features represents a month ($k = 1$) before, two months ($k = 2$) before, and three months ($k = 3$) before aggregation groups. Illustration of this example is depicted in Figure 1.

2.3. Resampling for Imbalanced Dataset

Imbalanced dataset is a condition where the distribution of one class highly dominates dataset as majority class or it can be also described as very rare event. The imbalanced condition, applies also in this dataset as seen in Table 2, can be commonly found in churn prediction or fraud detection cases, as they exist as rare events in the population. Some resampling approaches can be used to approximately balanced dataset, which involves removing majority class observation (under-sampling) or repeating minority class observation (over-sampling) or combination of both. Those three approaches, which already included in *imblearn* Python Toolbox [12], will be covered and compared to original dataset in this work to the performance. The methods performed for the resampling as described

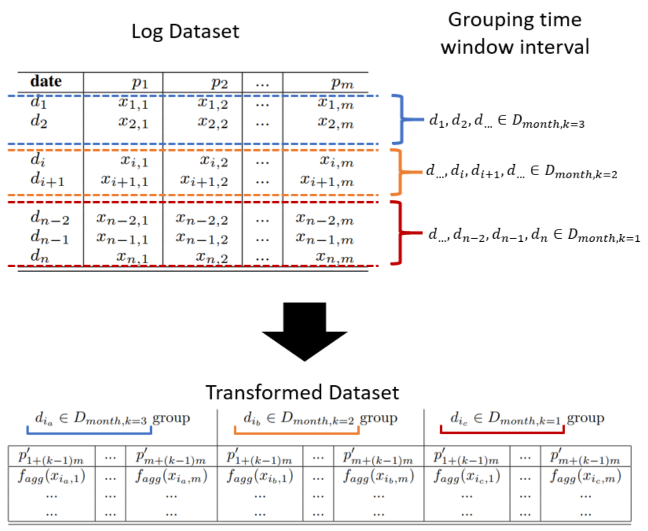

Log Dataset
Grouping time window interval
date
p_1
p_2
p_m
$d_1, d_2, d_{...} \in D_{month,k=3}$
$d_{...}, d_i, d_{i+1}, d_{...} \in D_{month,k=2}$
$d_{...}, d_{n-2}, d_{n-1}, d_n \in D_{month,k=1}$
Transformed Dataset
$d_{i_a} \in D_{month,k=3}$ group
$d_{i_b} \in D_{month,k=2}$ group
$d_{i_c} \in D_{month,k=1}$ group

Figure 1. User behavior log data transformation example

below, please note that several other resampling methods were not investigated due to memory limitation even though nicely provided in the *imblearn* library.

- Under sampling: Random Under Sampling
- Over sampling: Random over sampling, SMOTE
- Combined sampling: SMOTE+ENN, SMOTE+Tomek

Note:

- SMOTE: Synthetic Minority Over-sampling Technique
- ENN: Edited Nearest Neighbor (under-sampling)
- Tomek: Tomek links (under-sampling)

2.4. Random Forest Classification

Random forest (as cited in [8]) is known as classifier that is robust against over-fitting. Conceptually, random forest is an ensemble of decision tree of which a single decision tree is build based on randomly picked subset of predictors. However, the random forest classifier did not work very well on imbalanced dataset. Thus several preparation must be done including resampling or modify the original random forest to become weighted random forest.
Weighted random forest [13] uses cost-sensitive learning to penalize the mislabeled minority class. Thus, this modified random forest can improve the prediction performance of the minority class which will be later discussed, even though [8] implied that it has some disadvantages to noisy mislabeled class.

2.5. GA Feature Selection

As it is related to time-stamp based dataset which is user behavior plus several treatments to handle missing values, the KKBox churn prediction dataset could be very

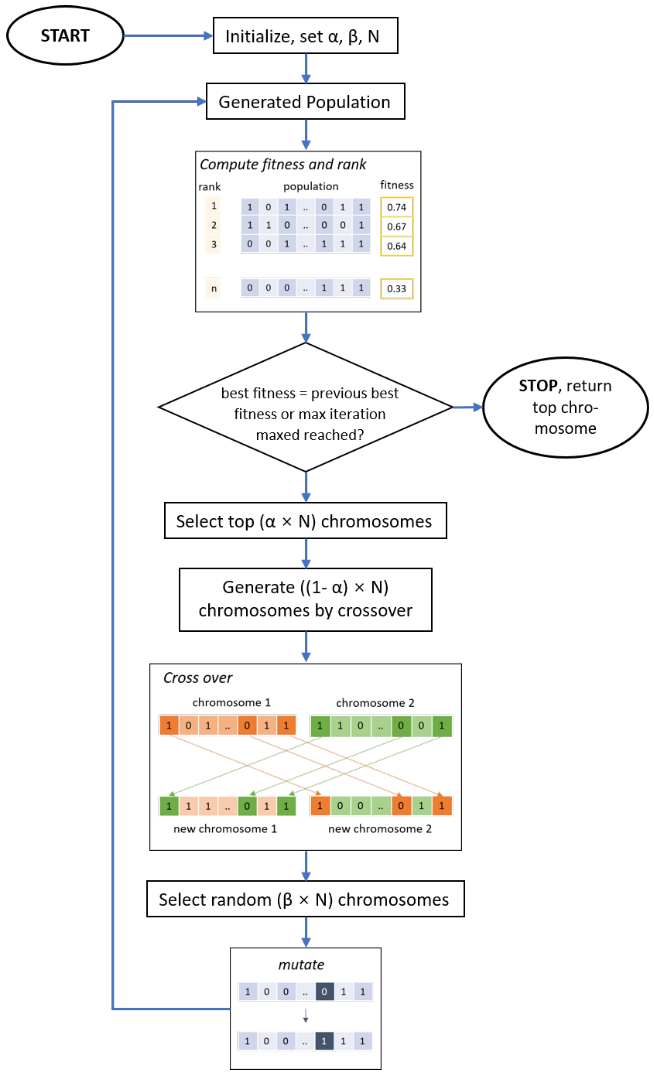
high dimensional. In addition, its size was already large. Feature selection might reduce computation cost whereas improve prediction performance while tends to remove noisy features.
Abbasimehr [10] used Genetic Algorithm (GA) to improve both of performance and comprehensibility of telecom churn prediction. It used a special cost function as performance and comprehensibility measurement which generated from geometric mean and number of generated rule. However, this work focuses on improving the performance which represented by log loss function. This performance measurement is also used as cost function in GA.
In this work, GA is used to restrict the combination of the possible features in Random Forest classifier even though Random Forest itself already chose its building features randomly. GA also adds possibility of avoiding local minimum using its mutation procedure. The GA process is depicted in Figure 2.
In GA, selected features were represented as single chromosome, which took form of boolean set which length is M (number of features). The value TRUE or 1 indicates that the feature is selected and considered into model while FALSE or 0 otherwise. Number of population N, selection probability α, and mutation probability β will be specified by user. The fitness function is computed as average log loss evaluation metrics by 5-fold cross validation using random forest classifier. The best features will be used as input for building model using full training set, which then will be used to predict the test set. However, the test set is only available in Kaggle. This paper will focus on validation performance based on K-fold cross validation.

3. Experiment Setup

The experiment flow is performed as described in Figure 3. The performance of each test scenario is assessed by log loss function of 5-fold cross validation, which is same as feature selection fitness function. The formula of log loss function between two class: 0 (not churning) and 1 (churning) is described in equation 1. Log loss measures the quality of the classifier according to its penalization of misclassification. If an observation was given higher probability on wrong class, it will yield higher log loss. Lower log loss value means the classifier has better performance in labelling each observation. As explained before, this setup covers mainly on validation performance to estimate the test performance. A few submission details on test performance will be provided in later section.

$$logloss = \sum_{i=1}^{N} (y_i)(p_i) + (1 - y_i)(1 - p_i) \qquad (1)$$

Experiments will be performed according to several considerations in terms of sampling method, dataset importance, and feature selections which summarized in Table 4. Scenario #1 explores about selecting proper dataset and time windows parameter for churn prediction. Scenario #2 considers effects of cost-sensitive learning by comparing normal


START
Initialize, set α, β, N
Generated Population
Compute fitness and rank
rank
population
fitness
best fitness = previous best fitness or max iteration maxed reached?
STOP, return top chromosome
Select top (α × N) chromosomes
Generate ((1- α) × N) chromosomes by crossover
Cross over
chromosome 1
chromosome 2
new chromosome 1
new chromosome 2
Select random (β × N) chromosomes
mutate
Figure 2. GA generic process

random forest with weighted random forest. After that, scenario #3 took resampling methods role into performance results. Options for sampling are under-sampling, over-sampling, SMOTE, SMOTE-ENN, and SMOTE-Tomek, plus the original ones as comparison. Furthermore, scenario #4 investigated the importance of feature selection using GA, compared with non-feature selected dataset and PCA (Principal Component Analysis) dimensional reduction. Note that on other than scenario#1, which explores the significance of feature extractions, only members table were used for predicting churn because of memory limitation.
When dealing with timestamp based tables, for instance transaction logs or user activity logs, approach as illustrated in Figure 4 to separate training and testing time window is used. To predict the churn probability in month T using timestamp based data of month T-1 (and older if needed), the known churn/not churn class of month T-1 will be used as training class while timestamp based data of month T-2

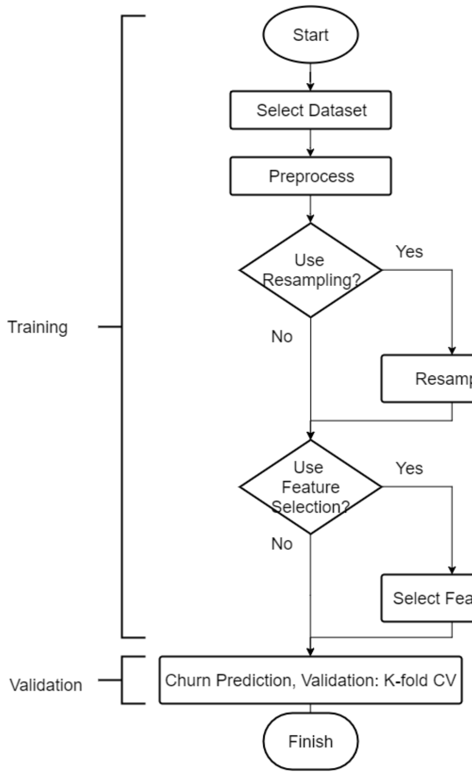


Figure 3. Experiment design of assessing performance by resampling and feature selection

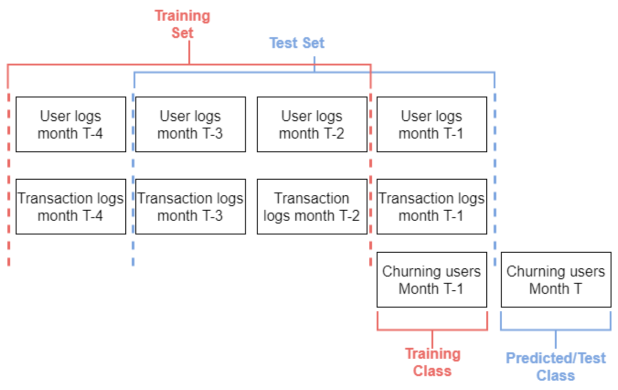


Figure 4. Training and testing design of churn prediction using timestamp/log based data

(and older if needed) will be used as training feature sets. In addition, the static data will be used in both approaches.

4. Result and Discussion

4.1. Dataset Selection and Feature Extraction

This scenario goal is to choose which dataset and feature set that shoud be considered in KKBox Churn prediction and which set that should be chosen for scenario #4 feature selection. Parameter setting of the random forest classifier, consisted of number of estimators and maximum depth. Both of two parameters were set as avoidance to overfitting, as they were compared to their performance in 5-fold cross validation log loss. The number of estimator is varied between 5 and 30 by interval of 5. And in each experiment of number of estimator, the maximum depth of the random forest is controlled by 2^1, 2^2, to 2^5 exponential interval. Then, each dataset selection performance will be assessed by acquiring value of 5-fold cross validation log loss in its best parameter setting. The minimum log loss of each datasets' churn prediction performance is described in Table 5, where the best is formatted in bold along with the parameters setting and number of features extracted.

From the result, all the log loss performance higher 0.25 except the best combination dataset: Members+User Logs-1 month LB+Transaction Logs-1 month LB (MU1T1) which has log loss of 0.2422. Even though it has the most number of features, random forest can separate class distribution very well among the others although the KKBox dataset is included as highly imbalanced dataset. Thus this also implied that combination between members' static demographic data and members' behavior in playing music and doing transaction altogether tend to represents churn prediction behavior.

The 2nd and 3rd position was occupied by combination of Members+User Logs-1 (MU1) month LB and only Members (M) dataset, which has log loss of 0.2503 and 0.2725 respectively. Regarding this top three position the features were in increasing number, which are 43, 51, and 60 for M, MU1, and MU1T1 respectively. However, this did not imply more features lead to better performance as dataset MT2, which has more features (61 features), has log loss of 0.2746.

4.2. Classifier Performance

The result of prediction using Random Forest Classifier and Weighted Random Forest Classifier is presented in Figure 5 and Figure 6. Figure 5 describes about the comparison of the behavior between random forest and weighted random forest if the one of forest parameter which is number of decision tree random used as ensemble is increased. In the other hand, Figure 6 show the same behavior comparison when the parameter maximum depth is increased. In both charts, it can be concluded that Weighted Random Forest has worse performance in validation as compared with normal or regular Random Forest. This observation implied that cost sensitive learning has some limitation in predicting KKBox Churn Prediction dataset. As discussed before, weighted random forest was not very robust to handle mislabeled minority class. Hence, it could be inferred that KKBox dataset contained noisy churning users' data. This is comprehensible as churning user's behavior is likely hard to detect thus being one of *state-of-the-art*.

Regarding both random forests' behavior, the setting is similar to previous scenario. In Figure 5, it can be seen that the validation log loss is almost stable and training log loss is monotonically slightly decreasing for regular random forest.

TABLE 4. EXPERIMENT SCENARIOS SUMMARY

Scenario	Classifier	Dataset
#1 Preprocessing: Dataset Selection and Feature Extraction	Random Forest	• Members (M) • User logs-1 month LB (U1) • User logs-2 month LB (U2) • Transaction logs-1 month LB (T1) • Transaction logs-2 month LB (T2) • Members+User logs 1-month LB (MU1) • Members+User logs 2-month LB (MU2) • Members+Transaction logs 1-month LB (MT1) • Members+Transaction logs 2-month LB (MT2) • Members+User logs 1-month LB+Transaction logs 1-month LB (MU1T1)
#2 Classifier Performance	• Random Forest • Weighted Random Forest	Members
#3 Resampling Methods	Random Forest	Members: • Original • Random Oversampling • Random Undersampling • SMOTE • SMOTE+Tomek • SMOTE+ENN
#4 Feature Selection	• Random Forest • GA+Random Forest • PCA+Random Forest	• Members • Members+ User logs 1-month LB+Transaction logs 1-month LB

TABLE 5. DATASET SELECTION RESULT

Dataset	Features	Log Loss	Parameter	
			Num. Estimators	Max Depth
M	43	0.2725	30	16
U1	8	0.3014	30	8
U2	16	0.2987	25	8
T1	9	0.2887	15	8
T2	18	0.2737	30	8
MU1	51	0.2503	25	16
MU2	59	0.2741	25	32
MT1	52	0.2890	30	8
MT2	61	0.2746	30	16
MU1T1	60	**0.2422**	30	16

Weighted random forest, although has higher error, has similar behavior, except in validation which has larger decrease instead of stable. While figure 6, while the maximum tree depth value is modified, showed delicate overfitting phenomenon in regular random forest performance: while the training performance gets better, the validation tends to perform more inferior. Yet, in weighted random forest, the overfitting phenomenon was still bleak. Therefore, tuning the maximum depth parameter in random forest gave significance in increasing random forest's classifying capability.

Effects of choosing longitudinal behavior transformation parameter, which is the time window interval, can be seen in Table 5. Dataset pairs U1-U2, T1-T2, MU1-MU2, MT1-MT2, signifies the increase of time window interval towards user logs and transaction logs by 1 month and 2 months. Log loss of classification using U1-U2, T1-T2, and MT1-MT2 pairs proved that increment of time window interval could improve the performance of classification, except for MU1-MU2 pairs. Thus, it might be interesting to be discussed furthermore.

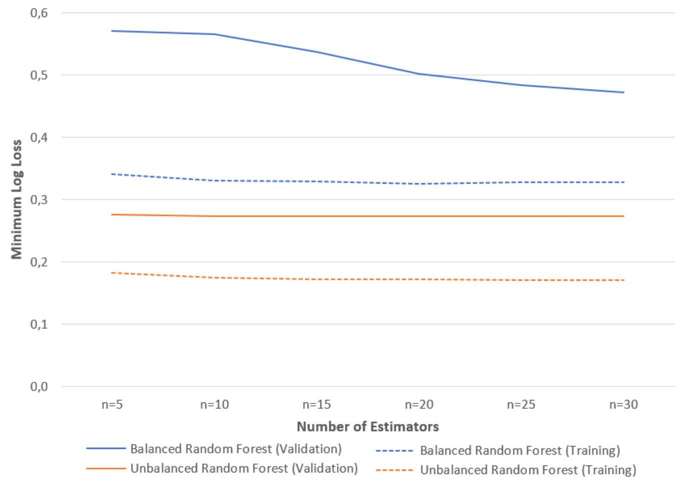


Figure 5. Regular and Weighted Random Forest performance comparison by varying number of estimator parameter (Train and Validation)

4.3. Resampling Methods

The best performance of each resampling method is depicted in Figure 7. Those results were produced by varying the maximum observations in the leaf node from $N \times 10^{-1}$, $N \times 10^{-2}$, $N \times 10^{-3}$, and $N \times 10^{-4}$ 5-fold cross validation log loss, where N is the number of observations. This restricts the splitting the tree to be deeper if the number of observations in a node when building a decision tree is reached. Each resampling has different value of N since resampling changes the number of observation by adding and removing observation from the dataset. From the chart, it can be seen that SMOTE+ENN resampling method give the best performance of 0.225 log loss. Meanwhile, the original dataset (without resampling) follows the best performance

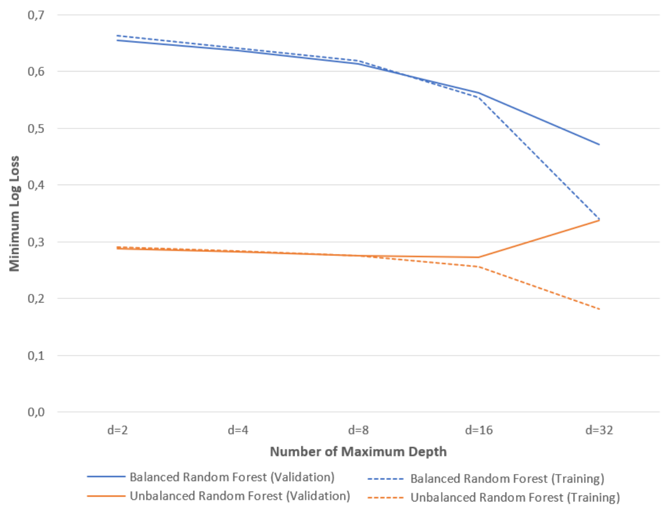


Figure 6. Regular and Weighted Random Forest performance comparison by varying number of maximum tree depth (Train and Validation)

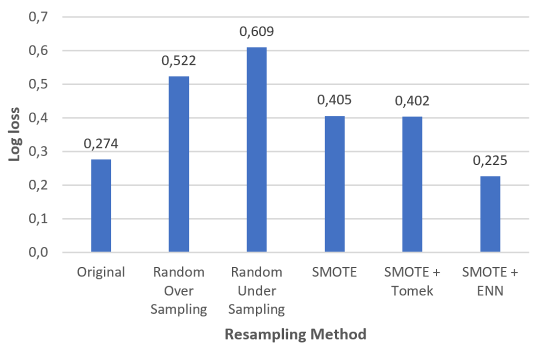


Figure 7. Scenario #3 each resampling's best log loss

with 0.274. This means that not all resampling method give better result as expected, except for SMOTE ENN.

It is possible that having only *Members* table resampled gave biased minority observations and noisy data reproduction, which perturbs the real distribution in original dataset. Random under sampling and random over sampling removes and adds, respectively, the same data naively. Combination of Tomek under sampling to SMOTE delivers similar performance to SMOTE over sampling, which is not very significant if compared to combination of ENN under sampling to SMOTE over sampling. Tomek links removes the most similar group of data (neighbors) which is from different class while Edited Nearest Neighbor removes a single data point that does not agree enough which its nearest neighbors. This removal of 'disagreed' class data points by ENN combined with SMOTE oversampling shows that correct treatments regarding minority class and balancing the class distribution gave significant improvement to highly imbalanced dataset.

TABLE 6. FEATURE SELECTION AND DIMENSIONALITY REDUCTION RESULT

Dataset	Approach	Features	Log Loss
Members (M)	-	43	-0,2725
	GA	37	0,2727
	PCA	20	0,2729
	PCA	10	0,2840
	PCA	5	**0,2722**
	PCA	4	0,2726
	PCA	3	0,2723
	PCA	2	0,2729
Members + User logs 1-month LB + Transaction logs 1-month LB (MU1T1)	-	60	-0,2422
	GA	37	0,2433
	PCA	30	**0,2397**
	PCA	15	0,2423
	PCA	5	0,2994
	PCA	4	0,2992
	PCA	3	0,2988
	PCA	2	0,2974

4.4. Feature Selection

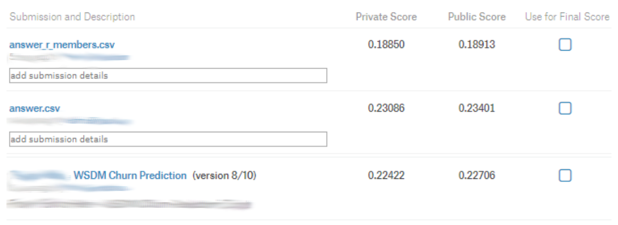
The result of the application of GA and PCA is described in Table 6. It shows the comparison of non-feature selection approach, GA, and various number of principal components parameter of PCA represented in *Features* column, while non-feature selection's represented number of its features and GA's number of selected features in final iteration. Prediction performance is measured by 5-fold cross validation log loss function. The classifier used in this scenario was random forest classifier with parameter of maximum depth of 16 and number of estimators 30. GA used parameter of 20 populations, maximum iteration of 5, 0.3 selection probability, and 0.01 mutation rate. These parameters was chosen in sake of memory availability and the computational power of the system.

For both datasets, GA performed slightly worse than the non-feature selection approach. This implied that importance of full features is cannot be captured by feature selection. Meanwhile, for dimensionality reduction, PCA performed similarly to the non-feature selection and GA approach, or even better in particular parameters. Even though PCA outperformed GA feature selection and non-feature selection approaches, there is no exact rule of thumb to pick the best number of principal components. The difference is noticeable in both *M* and *MU1T1* dataset. In the first group (dataset *M*), PCA of 10 number of components performed worst among all feature reduction approaches, while numbers above or below 10 gives better performance. PCA did not take account the importance of class distribution to define the principal components. In the second group (dataset *MU1T1*), all PCA with number of components 5 or below had higher log loss, which implied it removes the importance of several features.

According to this results, it is preferable to observe other classifier since random forest also took subset of features as building process. The feature selection to restrict random forest to choose features while building tree did not work very well in noisy dataset. Meanwhile, PCA performed better but unstable regarding its parameters when imposed

TABLE 7. CHURN PREDICTION KAGGLE SUBMISSION

Setup			Log loss Performance	
Dataset	Approach	Platform	40% test	60% test
Members	Random Forest	R	0.189	0.189
Members	Random Forest - SMOTE+ENN resampling	Python	0.231	0.234
Members	Random Forest - No resampling	Python	0.224	0.227



Submission and Description	Private Score	Public Score	Use for Final Score
answer_r_members.csv [redacted] add submission details	0.18850	0.18913	☐
answer.csv [redacted] add submission details	0.23086	0.23401	☐
[redacted] WSDM Churn Prediction (version 8/10) [redacted]	0.22422	0.22706	☐

Figure 8. Screenshot of Kaggle Submission of WSDM 2018 KKBox Churn Prediction

to different dataset.

4.5. Additional Discussion: Random Forest Parameters in Python

In several testing scenarios, particularly between Scenario #1 and scenario #3 on *Members* only dataset, showed slightly different performance. The scenario #1 prediction performance was 0,2725 while scenario #3's was 0,274. This is caused by different parameter adjustment in random forest fitting procedure and the random assignment of training and validation set of 5-fold cross validation. In Scenario #1, there are two adjusted parameters, which were number of estimators and maximum depth, while in scenario #3 the adjusted parameter was the maximum observations number in leaf node which causes not to split the node furthermore deeper.

5. Performance on Kaggle Test Set

WSDM 2018 Churn Prediction Competition allows only 5 submissions each 24 hours. Thus, the opportunity to submit the prediction is limited. Nevertheless, as far as this work extended to many levels, this work only display the best ones which can be seen in Table 7. As a benchmark, a code in R with regular Random Forest is also submitted and got the best prediction (row 1), with log loss lower than 0.2, while all the python codes using Random Forest gave log loss higher than 0.2 (row 2 and 3). R's Random Forest fitting has advantage to handle categorical data as input feature, different from the one that provided by Python in *sklearn* library. The snapshot of the submissions and final ranking also provided in Figure 8. This work's best submission ranked in position 423 of 575 submission, while the first rank achieved log loss of 0.079.

6. Conclusion

Churn prediction is one of the most challenging task in customer relationship management scope. In case of KKBox's churn prediction, this task was also incorporated with imbalanced dataset including its demographic static data and music playing plus transaction timestamped data. User behavior features using longitudinal behavior transformation was performed to gather features that represents user activity, however this leads to high dimensional data. Several options of datasets was created by adjusting the data transformation parameters combined with the static demographic data. Those datasets were explored by applying several resampling methods, weighted random forest and random forest classifier, and dimensionality reduction using Genetic Algorithm feature selection and Principal Component Analysis.
This work showed that the longitudinal behavior did well as feature of prediction using Random Forest Classifier. To tackle the highly imbalanced dataset, certain resampling method, namely combination of SMOTE and Edited Nearest Neighbor, significantly improved the performance of the prediction. Weighted Random Forest, as improvement of random forest approach to deal with imbalanced dataset, has worse performance of prediction compared to the regular random forest. Dimension reduction, as an effort to minimize noise and remove irrelevant features, only slightly improve the performance of the classifier itself. Another issue that emerged during this work was the capability of handling categorical data yet the classifier characteristic itself. In future, another improved classifier for imbalanced dataset should be considered such as improved balanced random forest or ones that more sensitive to minority class. Furthermore, preprocessing approach to extract relevant features should be examined as churning behavior is hard to capture.

References

[1] Z.-Y. Chen, Z.-P. Fan, and M. Sun, "A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data," *European Journal of operational research*, vol. 223, no. 2, pp. 461–472, 2012.

[2] L. Katelaris and M. Themistocleous, "Predicting customer churn: Customer behavior forecasting for subscription-based organizations," in *European, Mediterranean, and Middle Eastern Conference on Information Systems*. Springer, 2017, pp. 128–135.

[3] "Wsdm - kkbox's churn prediction challenge kaggle." [Online]. Available: https://www.kaggle.com/c/kkbox-churn-prediction-challenge/data

[4] J. Burez and D. Van den Poel, "Handling class imbalance in customer churn prediction," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4626–4636, 2009.

[5] K. Coussement, S. Lessmann, and G. Verstraeten, "A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry," *Decision Support Systems*, vol. 95, pp. 27–36, 2017.

[6] M. A. H. Farquad, V. Ravi, and S. B. Raju, "Churn prediction using comprehensible support vector machine: An analytical crm application," *Applied Soft Computing*, vol. 19, pp. 31–40, 2014.

[7] R. Vadakattu, B. Panda, S. Narayan, and H. Godhia, "Enterprise subscription churn prediction," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1317–1321.

[8] Y. Xie, X. Li, E. Ngai, and W. Ying, "Customer churn prediction using improved balanced random forests," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5445–5449, 2009.

[9] C.-F. Tsai and Y.-H. Lu, "Customer churn prediction by hybrid neural networks," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12 547–12 553, 2009.

[10] H. Abbasimehr and S. Alizadeh, "A novel genetic algorithm based method for building accurate and comprehensible churn prediction models," *International Journal of Research in Industrial Engineering*, vol. 2, no. 4, p. 1, 2013.

[11] A. Idris, A. Khan, and Y. S. Lee, "Intelligent churn prediction in telecom: employing mrmr feature selection and rotboost based ensemble classification," *Applied intelligence*, vol. 39, no. 3, pp. 659–672, 2013.

[12] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

[13] C. Chen and L. Breiman, "Using random forest to learn imbalanced data," 01 2004.