On the Development of Production Yield Classification Model Under Imbalanced Data and Categorical Variable Constraints

Chung Cheng Huang

Advisor: Jakey Blue, Ph.D.

July 2021

Institute of Industrial Engineering College of Engineering

National Taiwan University

Abstract

The classification system for fault detection and prediction is an important analysis tool in advanced processes. We often use classification or clustering to tell if the state of product is anomaly or not. The classification models based on yield by machine learning usually get good results, but it will be difficult identifying the minority when the training data is imbalanced. However, the cost of imbalanced data is relatively higher is also a common problem in the process of the technology industry. For example, there is generally one-thousandth or even one-in-a-million probability of detective products in the process, and the algorithm usually results in that all the products are good to reach a high accuracy rate. Therefore, the classifiers do not learn the difference between categories, ignoring the extremely high cost of misclassification.

Recent studies show that the imbalanced data problem is mostly tackled by data augmentation or model parameter optimization. Some researches start by analyzing the data characteristics first, such as the imbalance ratio, distribution density, and overlap between categories, and then augment the minority data. Nevertheless, the data augmentation is purely based on numeric variables. In this thesis, we study and develop the novel data augmentation with regards to discrete variables, in particular, binary ones. Hamming distance is employed to calculate the similarity among binary features, and a new oversampling method based on the interaction between the minority and majority is proposed. New minority data are generated after taking the noise of data distribution and the confusion of the majority category into account.

Finally, by combining conventional undersampling methods and controlling the balance ratio of the training data, this thesis conducts a variety of experimental analyses on applying the proposed oversampling algorithm to generating minority data, which are then trained by machine learning models. The results found that with the proposed upsampling method, model performances are consistently better compared with the benefits of model optimization.

Keywords: imbalanced data, categorical variables, oversampling, undersampling, data augmentation, fault detection and classification, machine learning algorithms

Table of Contents

List of Figures

1 Introduction

There are different types of classification problems in real life, especially the problems given imbalanced data occur in many fields, such like financial fraud, airplane crash, fault detection or disease diagnosis, and they cost a lot once they happen. Recently, more and more research has started to focus on classification of imbalanced data because the traditional methods tend to fail when predicting the minority. Traditional machine learning classifiers usually bias towards to the majority and regard the class of all data as majority for getting the high accuracy. In addition to the condition of imbalanced data, the data with binary classes for real application sometimes contains more text or other categorical variables.

Based on the two common characteristics (imbalanced data and binary variables), we can follow the steps below.

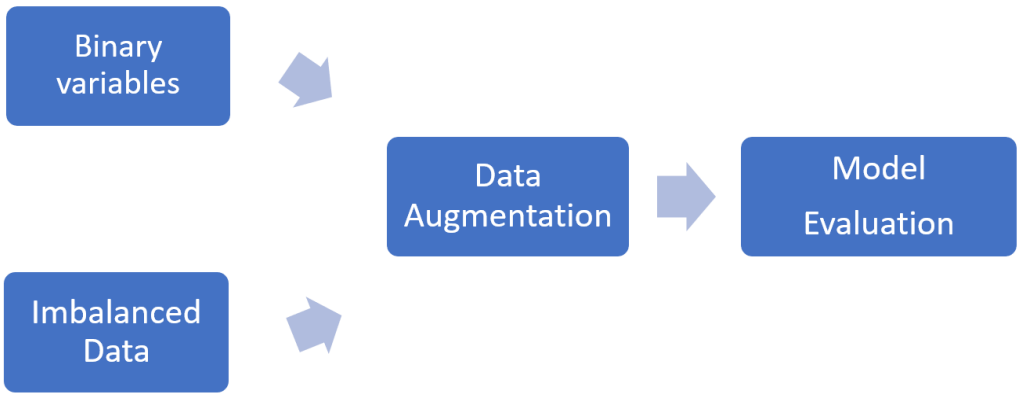


Figure 1. The framework of the research process

When it comes to the methods of improving the evaluation criteria, we usually use the data augmentation to produce new data of minority, remove the data of majority, and we can call them resampling. Moreover, the cost-sensitive approaches which assign different cost for each class or using other loss functions to train the model are commonly used as well. First, the constraints here are the binary variables and imbalanced data, so the space of the new data is limited. Thus, we should make sure that the new data is reasonable and the new data will improve the model-building. We can also balance the classes by removing the majority, but it may remove the key information from time to time. As for setting cost of each class, it seems more intuitive to analyse, but it is difficult to set the clear cost properly when the class is inestimable.

After we find out the appropriate pre-process, then we can start to train the model by a fitting loss function and choose a set of better parameters, making the system automatically finally. When we train the models, it's also a good way to enhance the evaluation by ensembles of classifiers which may contain the concept of cost-sensitive approaches. When it comes to the evaluation, what we use frequently in this case are precision and recall instead of accuracy. In this paper, we will focus on pre-processing methods on the data level.

There are other fundamental characteristics also influence the result, such as overlapping, small disjuncts. Sometimes the overlapping problems are more challenging especially in the situation of limited data space and big data. Besides, the small disjuncts tend to be misclassify as the noise, therefore the noise reduction is also an important step.

Because it is easier to get big data on the advanced process so far, so we will try to balance the big dataset first for some degree, and emphasize the methods of oversampling which try to get new data of minority. The case study of our research is about manufacturing data contains binary class which represents normal or anomaly, and our main objective is to detect the anomaly precisely for reducing the cost and build an effective yield classification system.

Many kinds of techniques we can use usually design for specific problems because we do not know the root characteristics comprehensively, but we should at least know the different situation in same field so that we can adopt a strategy in time when we want to produce new product.

In the following parts, we will review the literature of some basic characteristics, resampling methods and models first. The third chapter will show the purposed method of resampling, the fourth chapter will implement the purposed methods on TFT-LCD dataset. At last, we give some conclusion from the experiments and provide some extension problems for future research.

2 State-of-The-Art

The application of the data augmentation is wide and difficult with imbalanced data, so Haixiang (2017) surveyed some relative literature and grouped up them by different balance ratio which defined as the ratio of the number of instances in the majority class to the number of instances in the minority class, data augmentation or their fields. It is useful to study the nature of imbalanced data and how their characteristics will influence their classification performance. This chapter will come up with some reviews on characteristics commonly observed with imbalanced data, some general ideas of resampling methodologies and evaluation metrics for this classification problem, summing up which combination is suitable.

2.1 Basic characteristics and challenge with imbalanced data

Before we select the methodologies, what we need to do first is to analyse the data with whole picture by kinds of estimators or visualization. There are two concepts here in imbalanced data which are minority class and minority cases individually, and we usually ignore the impact of minority cases (Japkowicz, 2001) . In figure 2.1, It shows two classes with two labels which are "+" and "-", and they mean minority class (A) and majority class (B). We can observe that there are five subsets associated with minority are labelled A1-A5, and the subsets A2-A5 correspond to the minority cases. In this case, the areas with dash line mean the scope we predict them as minority class, therefore we can know if it is lack of data like A2, it is difficult to classify it when being surrounded by majority class. As for A4 and A5 may similar enough which can be more easily to joint each other by resampling. The small disjunct A3 tends to be detect to noise (Weiss, 2004).

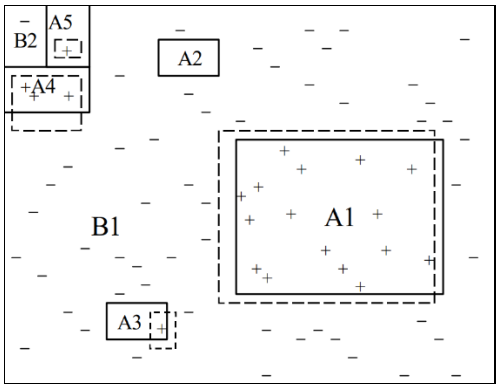


Figure 2.1. misclassification of rare case and noise (Weiss, 2004)

Another key problem is overlapping. Figure2.2 shows that the different level of overlapping. Left side is more imbalanced while right side is more balanced. In terms of the level of overlapping, we can observe left side is lower than right side, so it is more tough to classify the border of two classes. According to overlapping research, the evaluation is stable and better with the relatively low overlapping rate (Parti, 2004). As the result of the impact of overlapping, we must need to check if the data has significant overlapping rate.

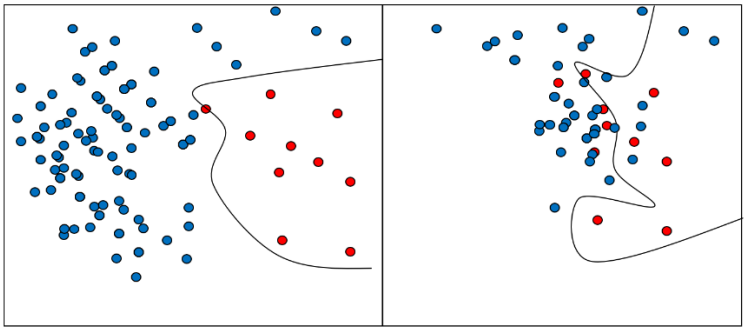


Figure2.2. different levels of overlapping with imbalanced data

In order to tell if the instances are noise or if they have the high degree of hardness so that easy to be misclassified, we can define some attributes to each instance (Napierala, 2016). For example, if we follow a rule to search the 5 neighbours of each instance, then we can define the data has attribute of safe because most of its neighbours are the same class with it. If the number of its minority neighbours are similar to majority neighbours, we can call it borderline while the its neighbours with high rate of different class will be outlier. Most of time, we focus on find out the borderline because the data with this attribute tends to be misclassified. We can see the figure below:

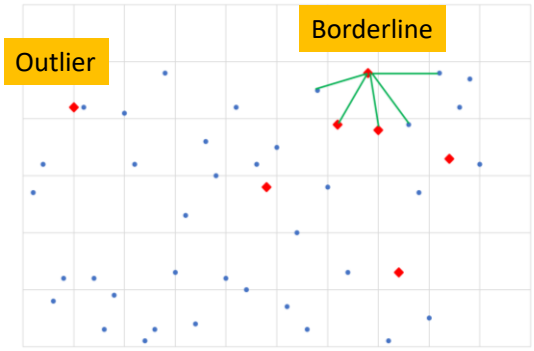


Figure2.3. define the new attribute for minority by 5-NN

In addition to those characteristics, we can also calculate some estimators like range which means the farthest distance of two instances by specific similarity, or we can visualize the data directly by MDS or others.

2.2 Resampling

Resampling includes oversampling and undersampling, one of the simplest oversampling techniques is to replicate minority data randomly while the simplest undersampling is to eliminate the examples from the majority class. However, if we use the methods of random oversampling or undersampling, it may result in overfitting or removing the important examples. Therefore, there are some methods like SMOTE (synthetic minority oversampling techniques) and SMOTE-N which is suitable for data with categorical variables (Chawla, 2002) . For example, SMOTE is to generate new synthetic artificial data based on their feature space and the similarity between every two minority examples. When we look for the nearest neighbours, the similarity is usually calculated with the Euclidean distance. As for the SMOTE-N is based on finding the neighbours by hamming distance which is calculated by how many the different features between two examples.

Noise or otherwise unreliable data from the majority class usually have to be careful, the data may be small parts of majority but with significant impact for minority. Under this circumstances, undersampling is also a prevalent way to cope with the imbalanced data, such like random undersampling, instance hardness threshold (Smith, 2014) and Near Miss (Mani, 2003) . Most of the experiments showed the version two of Near Miss usually got better performance.

Furthermore, we can also combine these methods together because both of them have their advantage and disadvantage. After the data augmentation, we can get a new trainingset whose balanced ratio is higher than the original. However, the balanced ratio of the dataset is also a parameter we can decide it sometimes. Zhou (2013) analysed the result with different balanced ratio and found out that it is not necessary to let the balanced ratio equal to 1, instead it depends on the other characteristics we discuss previously.

2.3 Evaluation criteria

The common machine learning models for classification contain Decision Tree Model, Random Forest (Breiman, 2001) , Naïve Bayesian Model, One Class SVM for anomaly detection or some ensemble classifiers. Figure2.4 shows some combination of pre-process and models (Galar, 2011) .

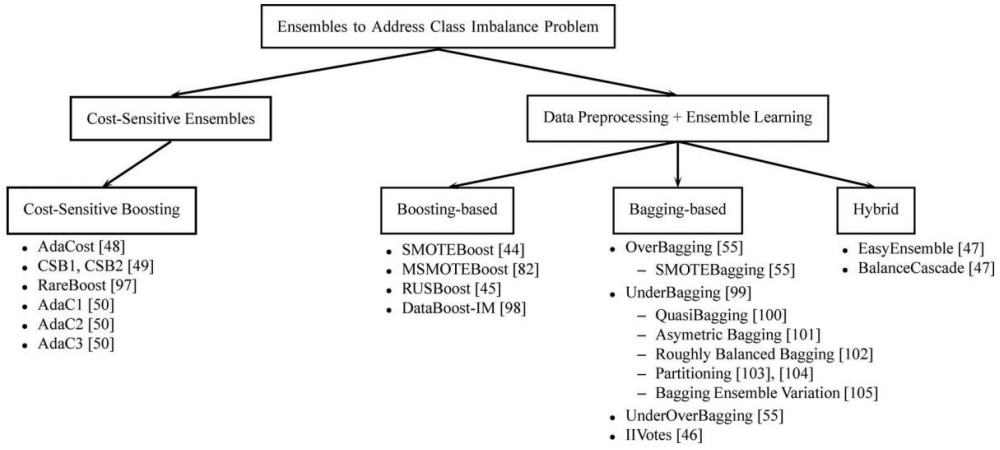


Figure.2.4. A taxonomy of different combination of pre-process and classifiers

In this problem, we can get a confusion matrix first in table2.1 (Stehman, 1997) , and then we can usually get precision, recall and the evaluation from these two value, such as AUC (area under the curve) of PRC (Precision-Recall curve) or F1 score after classifying by models for imbalanced data. In general, if we enhance the recall by predicting more minority, precision usually drop because of the more misclassification.

Table2.1. Confusion matrix

	True minority	True majority
Predict minority	*TP* (True Positive)	*FP* (False Positive)
Predict majority	*FN* (False Negative)	*TN* (True Negative)

Here are some formulas of key evaluation:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}, \tag{1}$$

$$Precision = \frac{TP}{TP+FP}, \tag{2}$$

$$Recall = \frac{TP}{TP+FN}. \tag{3}$$

$$F1\ Score = \frac{2\times Presicion\times Recall}{Precision+Recall}. \tag{4}$$

3 Research Methods

Based on the literature review in chapter 2, this paper will propose a process structure for imbalanced data, which can be divided into three stages, namely data preview, data pre-process and model building which includes classifiers and regression, and show in figure3.1. However, due to the diverse characteristics of data, there is not only a pre-process technique or a model that is suitable for all kinds of situation, so we need to design a customized method.

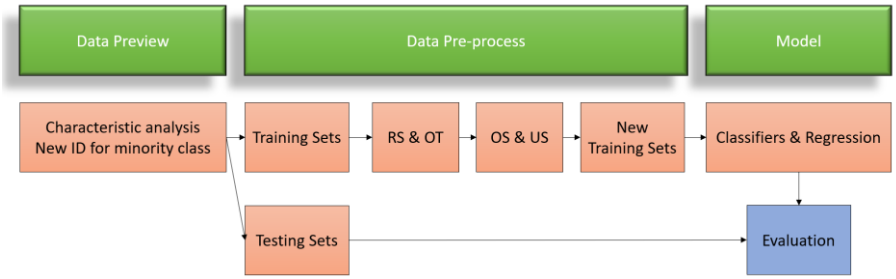


Figure 3.1. Framework for imbalanced data with categorical variables

First of all, this article defines each symbol first and selects the appropriate method at each stage of the process through the characteristics of data. Moreover, we emphasize the oversampling method in pre-process. Here are some abbreviation for the following content and their meaning:

Table 3.1. The abbreviation of key marks

symbols	meaning
N	Total number of dataset
p	Number of features
X	Categorical variables
Y	Binary and dependent variables
d	Hamming distance of any two samples
α	All minority data
I	Total number of minority
c	Kinds of α
β	All the majority data
b_j	Each data in majority class ($j = 1, 2, \dots, N - I$)
θ	All the number of minority class divide by all the majority

n	The number of training sets after data augmentation
$Training\_A$	The original training set
$Training\_B$	The training set after RS and OT
T	All the training sets after data augmentation
T_cont	The training sets by transferring the labels of T into continuous one
$Rank(d)$	All the possible corner points given the d and center
$score_j$	The continuous value for each β by multiplying weight
L	The continuous labels from Y
THR	The threshold for defective probability of AdaBoostRegressor
A_{PR}	The area of under the PRC

3.1 Preview of the basic characteristics

When we observe the characteristics of the imbalanced data, we can also set the new ID for the minority data, and divide it into a training set and a testing set. There is an important thing we need to be careful, the balanced ratio of training set cannot be too low.

In addition, through the 5-NN mentioned in the previous chapter, the minority data can be distinguished whether they have the attribute of safe, borderline or outline. However, if the θ of training set is still extremely low, such like only one-thousandth, it will result in all the α are regarded as outlier while most the β are regarded as safe. In addition to 5-NN which searches only 5 neighbours for local area, we can search all data for global area.

As shown in figure 3.2, the horizontal axis represents the number of neighbours searched from nearest neighbours so far, and the vertical axis is the cumulative number of α. For example, the dash line indicates there are about 16 data of α when we have searched 5000 nearest data, we can also call it Search Curve in this paper. Therefore, $umber\ of\ search = N - 1$, and $\alpha\ search = I - 1$.

After the data preview, we can start to divide the dataset into training set and testing set. We can divide it by the timing or divide it randomly before the following pre-process.

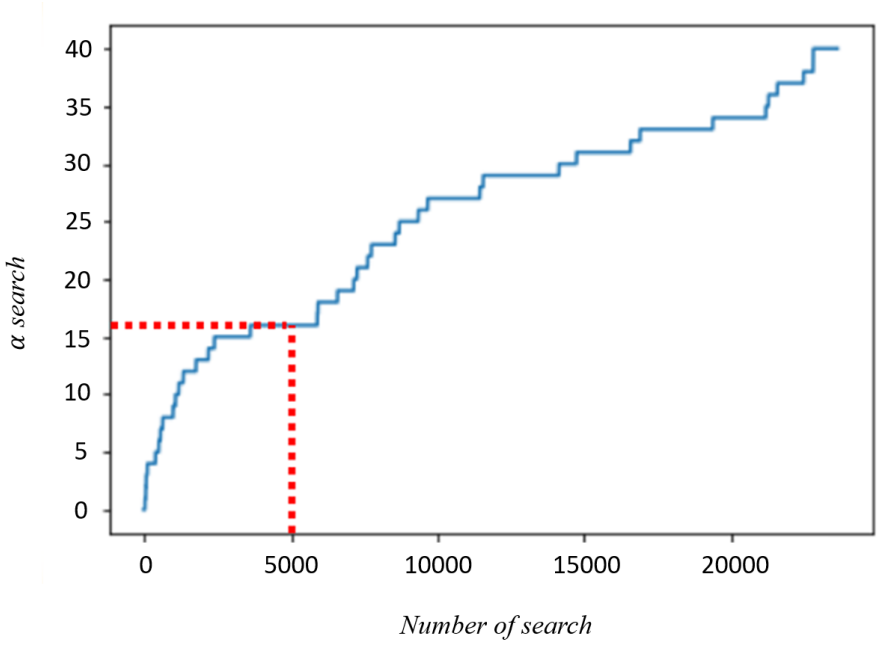


Figure 3.2. Method of search all data for global area

3.2 Data Augmentation

We can find out not only the outliers of α but also outliers of β. Before we start to do data augmentation, we should filter the noise and overcome the problems of overlap. Here are two purposed methods, Remove Single (RS) and Overlap Transformation (OT) individually in the pre-process.

RS is to remove the β which appears only once. When we do OT, we need to set a threshold which represents the ratio of minority on single point first. For instances, if we set the threshold to $\frac{1}{10}$, it means if there are one α and 10 times of β overlap on single point, we will transfer the all 10 β into 10 α. On the contrary, if there are one α with more than 10 β, we will transfer all α into β.

After these two steps, then we introduce the oversampling methods revised from the concept of SMOTE-N. It is different from SMOTE-N because it selects the range of distance first instead of selecting the fixed number of neighbours from each α. Therefore, it does not have certain number of neighbours, and we use the Rank to represent the similarity from each α given d here:

$$Rank(d) = C_d^p, \tag{5}$$

Our oversampling (OS) method also have two ways, Boarder (BDR) and Self-Expand (SE). As shown in figure 3.3, the BDR contains three parameters, namely Near Minor Rank, Major Ratio Max and Major Corner.

For example, when the Near Minor Rank is 3, we will take d which is lower than 3 between any two α. Here we use the hamming distance to evaluate the similarity between samples because this distance is more appropriate for categorical variables. The possible data space between α with $d = 3$ contains 6 corner points where we can generate new α, and we set the Major Ratio Max to 0.5, then checking if the ratio of β among these data is higher than 0.5. We generate the new α on the other corner points and check if the remaining data β occurred many times. If we set the Major Corner to 50, we will replace it with α when the number of β on the point does not exceed 50.

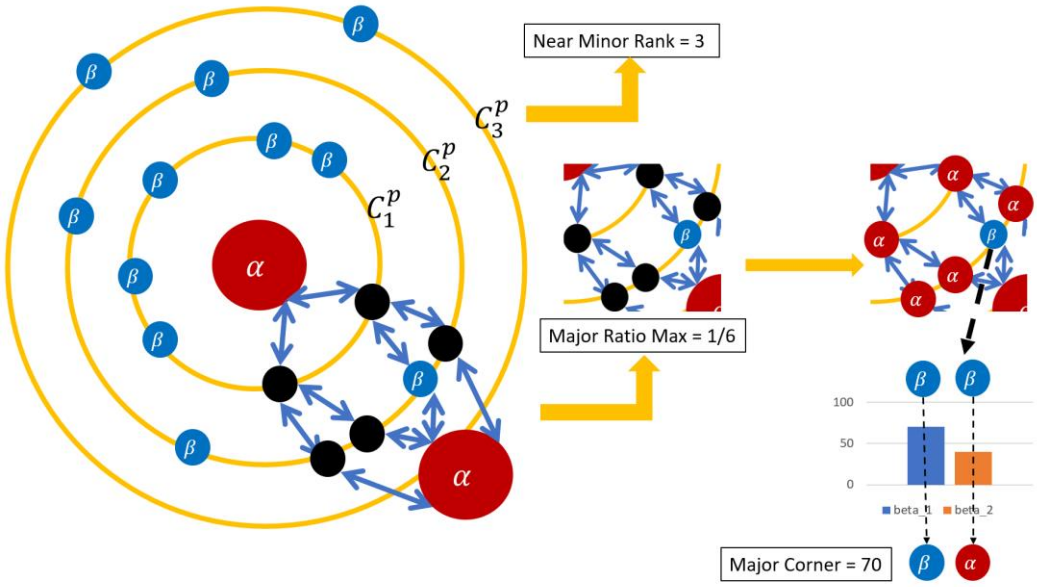


Figure 3.3. Oversampling technique-BDR

As shown in figure 3.4, the SE contains two parameters. First one is Near Major, it means to select the farthest β with d equals to Near Major from each α, generating new α surrounding the center α with $d = 1$. Moreover, we have to avoid approaching the β nearby. We can observe that if we set the Near Major to 2, we cannot generate new α on three corner points in this example.

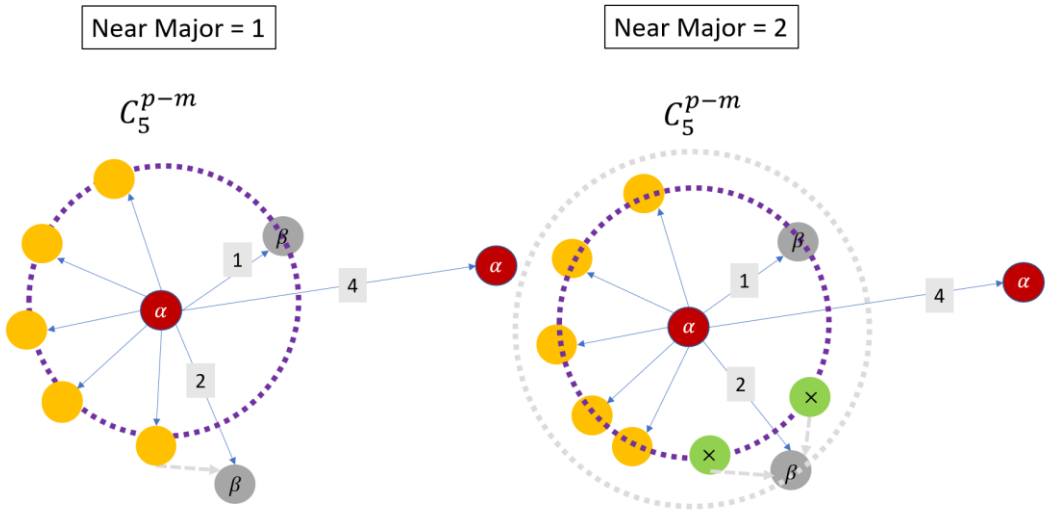


Figure 3.4. Oversampling-SE

Besides, we can combine OS and undersampling (US) in different kinds of combinations. There are four kinds of combinations including (1) doing OS before US, (2) doing US only, (3) doing OS and US independently and (4) doing US before OS. The US techniques here are Random Undersampling (RUS), Instance Hardness Threshold (IHT) and version 2 of Near Miss (NM) which are commonly used.

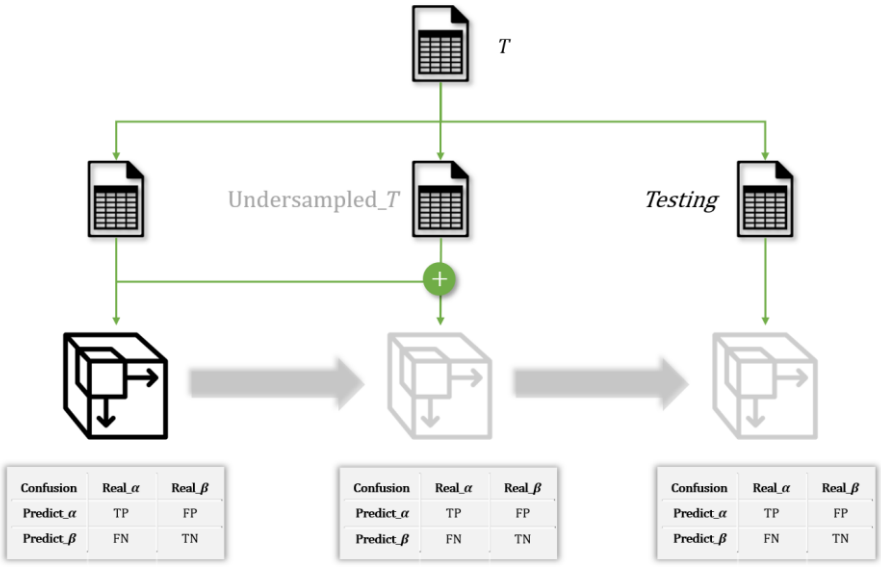
After doing the data augmentation, we can get n times of training sets and call them T:

$$T = \{Training\_1, Training\_2, \dots, Training\_n\}, \tag{6}$$

3.3 Models and Evaluation

There are some prevalent models including AdaBoost, XGBoost, Random Forest, Naïve Bayesian, One Class SVM and AdaBoostRegressor. Based on the confusion matrix, we can get three kinds of confusion matrix by different testing sets.

The three testing sets: (1) same as training set, (2) same as training set but adding the data removed by US, (3) testing set. Then, we can get three kinds of precision and recall, namely $(p1, r1)$, $(p2, r2)$ and $(p3, r3)$ individually, shown in below:



In addition to $p3$ and $r3$ calculated in third matrix, we can also take $Test\ rate$ into account under the condition of high cost when predict data to α.

$$Test\ rate = \frac{TP + FP}{TP + FP + FN + TN}, \tag{7}$$

Because it is usually better to transfer the training set's Y into continuous labels when using regression, so we transfer the majority data β to (0,1). Our concept of this step is to let the new labels of β which is surrounded by many α or gets close to α is higher. Therefore, we take one of β and find its neighbours which have $d \leq 3$. As shown in figure 3.5, the weight of the first circle (Rank (1)) from center β is 3 while the Rank(2) is 2 and Rank(3) is 1, then we calculate the score by the product of weight and the number of α on each Rank(d):

$$score_j = \sum((weight\ of\ Rank(d)) \times (number\ of\ \alpha\ in\ Rank(d)\ for\ b_j)),$$
$$d = 1, 2, 3, j = 1, 2, \dots, N - I. \tag{8}$$

After that, we do the standardize for these labels and multiply a constant 0.95 for avoiding the 1 on labels. At last, we can get another set of training data.

$$L = \frac{score_j - score_{min}}{score_{max} - score_{min}} \times 0.95,\ \ j = 1, 2, \dots, N - I. \tag{9}$$

In addition to the T, we can also get more n training sets with continuous labels below:

$$T\_cont = \{Training\_cont\_1, Training\_cont\_2, \dots, Training\_cont\_n\}. \tag{10}$$

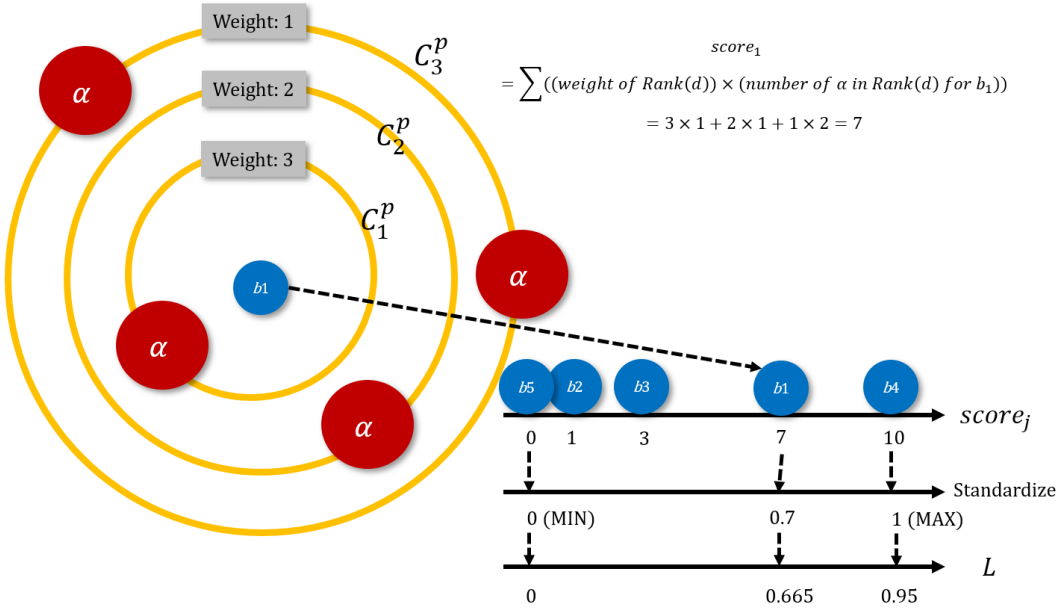


Figure 3.5. Continuous labels from categorical labels

The precision and recall from regression are found by $THR = 0.5$. We can also get other evaluation like A_{PR} which means the AUC of PRC, and the precision and recall may not the best values when $THR = 0.5$, instead it's better to point towards the (1,1), shown in figure below:

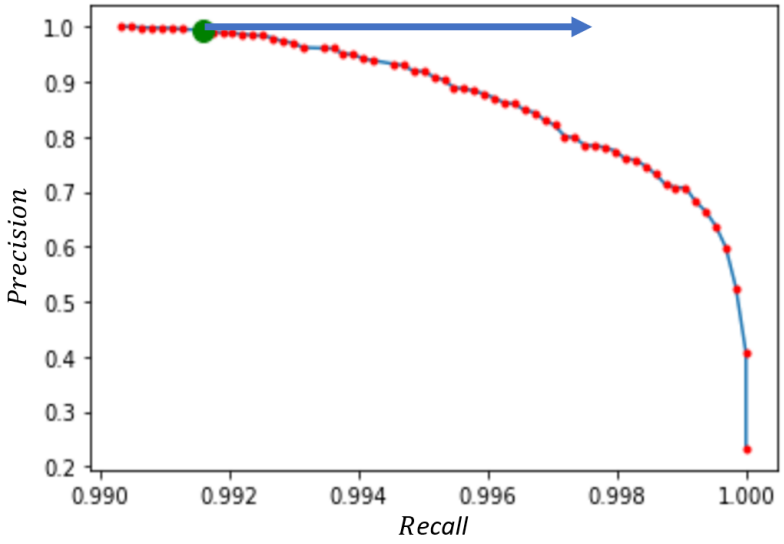


Figure 3.6. The adjustment for threshold of Regression

4 Case Study

The dataset we study is from a company manufacturing the TFT-LCD products. In general, they will do the aging test to the most of the finished goods before they ship them to customers for meet the requirement of customers because they cannot classify the imbalanced data easily. The key is that the cost of aging test including time, space and other resource is very high so that they need an effective classifier to help them select the potentially defectively products. In other words, they want to detect maximum or all the defective product by detecting minimum samples.

The X of dataset are the features from manufacturing execution system which 1 means the occurrence and Y is the quality of product after undergoing the aging test, 1 means defective and 0 means normal product, shown in figure 4.1. We will not take the features of process time into account when we train the models.

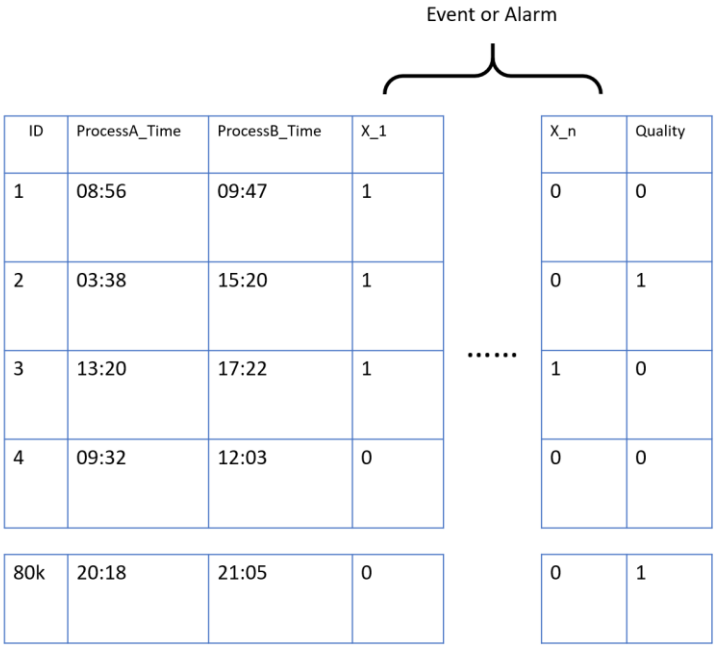


Figure 4.1. The dataset of TFT-LCD process

4.1 Characteristic of dataset

There are some estimators for this data in table 4.1. We can observe that the kinds of defectives are slightly lower than the number of them, it means some of the defectives are same. Moreover, it has extremely low θ and relatively low range which represents the farthest distance between any two samples.

Table 4.1. Basic estimators for TFT-LCD dataset

Total data (N)	132698
months	8
Number of defectives (I)	67
Kinds of defectives (c)	62
Balanced ratio (θ)	0.000492
Number of features (p)	82
Comparison of θ	M7 < M1 < M8 < M6 < M4 < M2 < M5 < M3

In figure 4.2, we check the sum of each features, there are many features with low rate, so this dataset may contain some features which is not important. In general, we can use the techniques of feature selection or feature extraction, but owing to the low dimension of this dataset and the overlapping problems which easily get worse by feature reduction, so we do not use any feature selection or feature extraction.

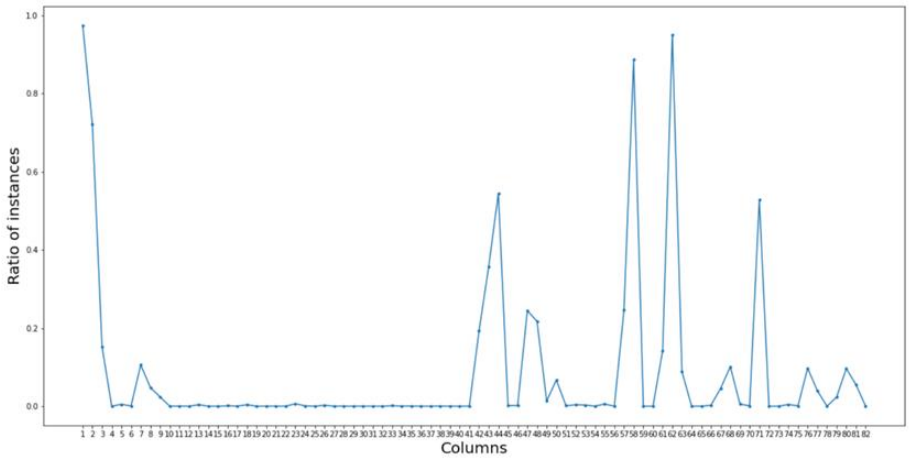


Figure 4.2. The ratio of occurrence of each feature

In order to know the overlapping level of each α and β, we group up each kind of α and count the number of β overlapping with them. It contains 4 kinds of α occurs more than once, and 8 kinds of α overlap with more than 500 β. Besides, based on the occurring times of each α and the month they appeared, most of them appeared again in a single month or consecutive months, thus we have difficult telling if this kind of α will show up afterwards.

Because most of the features in this case mean some kind of events or warning signs, we add up all the features which occur with label 1 for every sample in table 4.2

below. We can know the number of events does not have significant relationship with defectives, and we also find out the data with maximal events only has at most 17 flags.

Table 4.2. Number of flags of each sample and their ID of α

Flags	**Unique Types**	α	β	***α_ID***
3	18	1	795	54
4	76	0	848	
5	280	7	22026	0, 33, 50, 51, 60
6	942	21	31971	2, 3, 8, 19, 20, 21, 29, 30, 32, 35, 37, 43, 47, 49, 52, 53, 56, 59, 61
7	1911	11	34628	17, 25, 26, 36, 39, 41, 42, 44, 45, 48, 55
8	2590	13	22869	7, 9, 10, 11, 14, 15, 16, 18, 22, 23, 28, 40, 57
9	2524	6	12691	5, 12, 34, 38, 46, 58
10	1662	6	6380	1, 4 ,13, 24, 27
11	934	2	2850	6, 31
12	350	0	894	
13	132	0	223	
14	40	0	42	
15	9	0	9	
16	4	0	4	
17	1	0	1	

In order to distinguish different kinds of α, we can search all the other data in order by hamming distance. Therefore, we calculate the AUC of Search Curve defined in previous chapter, the value is between o.37 to 0.64 and most of α are sparse.

4.2 Experimental design and data pre-process

There are two kinds of experiments with chronological splitting or random splitting, the θ of their training sets which called $Training\_A$ are both about 0.0006 in the beginning. First one is to take the last two months (M7 and M8) as testing set and take the other as training set, random splitting is to divide the dataset into five groups

randomly, and take one of them as testing set each time, testing by five-fold (cross validation).

After that, we take the $Training\_A$ to RS, OT and four kinds of data augmentation with different combinations. We called the training set after RS and OT as $Training\_B$ while the training set after different data augmentation are T, and these 12 datasets with $Training\_A$ and $Training\_B$ will be training sets for all the models. In addition, the labels transfer from binary class to continuous labels are T_cont, and these 12 datasets with $Training\_cont\_A$ and $Training\_cont\_B$ will also be training sets for the regression model (AdaBoostRegressor). Therefore, here are 28 training sets for training models. Besides, the θ of T is controlled to 0.3 here first. The framework is shown below:

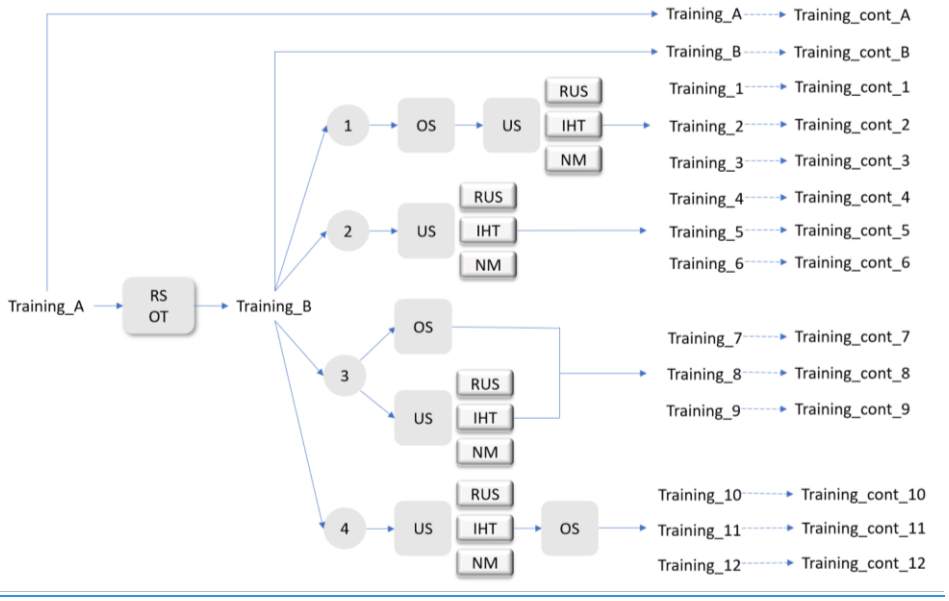


Figure 4.3. the framework of pre-process and data augmentation for training sets

4.3 Experimental performance

We use six kinds of model of machine learning for the training set, regression is training by training set with categorical labels and training set with continuous labels individually. The experiments here are carried out by the Scikit-Learn of Python (Pedregosa, 2011). Moreover, models trained by like $Training\_A$ and $Training\_B$ is for benchmark in these experiments. There are the parameter setting which is decided by common parameters and grid search for better result in training score of all models

in table 4.3. We will focus on the result which $r3 > 0.5$ and $r3 > Test\ rate$ as our candidates.

Table 4.3. The parameter setting for six models

AdaBoost	*Estimators=50, Learning rate=0.8*
AdaBoostRegressor	*Estimators=50, Learning rate=0.8, THR=0.5*
XGBoost	*Estimators=50, Max_derpth=3, Subsample=1*
Random Forest	*Estimators=50, Max_depth="unlimited"*
Naïve Bayes	*Alpha=0.1*
One Class SVM	*Kernel="poly", Nu=0.05*

First of all, we fix the $\theta = 0.3$ of the AdaBoost in table 4.4, we can see the evaluation which $r3 > 0.5$ and $r3 > Test\ rate$ with red mark. There is no significant difference between $Training\_A$ and $Training\_B$, and the $Training\_3$, $Training\_6$, $Training\_9$ and $Training\_12$ which is undersampled by NM are better in terms of $r3$

Table 4.4. key evaluation of AdaBoost with $\theta = 0.3$

AdaBoost ($\theta = 0.3$, chronologically divided)								
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$
A						0.00000	0.00	0.00
B						0.00085	0.06	0.02
1	0.30	1.00	0.99	0.98	0.99	0.00068	0.17	0.08
2	0.30	1.00	1.00	0.16	1.00	0.00016	0.11	0.22
3	0.30	1.00	0.99	0.15	0.99	0.00030	**0.56**	0.60
4	0.30	1.00	1.00	0.97	1.00	0.00000	0.00	0.03
5	0.30	1.00	1.00	0.10	1.00	0.00024	0.17	0.22
6	0.30	1.00	1.00	0.09	1.00	0.00031	**0.67**	0.70
7	0.30	1.00	0.99	0.98	0.99	0.00068	0.17	0.08
8	0.30	1.00	1.00	0.17	1.00	0.00026	0.17	0.21
9	0.30	1.00	0.99	0.14	0.99	0.00026	**0.56**	0.68
10	0.30	1.00	0.99	0.98	0.99	0.00000	0.00	0.03
11	0.30	1.00	1.00	0.21	1.00	0.00000	0.00	0.13
12	0.30	1.00	1.00	0.12	1.00	0.00033	**0.83**	**0.81**

Then we fix the $\theta = 0.5$ of the AdaBoost in table 4.5, we can find out the evaluation enhance in terms of both $r3$ and $Test\ rate$. In addition to AdaBoost, the other models also have better performance with $\theta = 0.5$. Most of $p1$ and $r1$ are

extremely high and $p2$ is still higher with RUS than the other US techniques. We can see the NM dominates the others in table 4.4 and it has significant improvement between different from $\theta = 0.3$ to $\theta = 0.5$.

Table 4.5. key evaluation of AdaBoost with $\theta = 0.5$

AdaBoost ($\theta = 0.5$, chronologically divided)								
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$
A						0.00000	0.00	0.00
B						0.00085	0.06	0.02
1	0.50	1.00	0.99	0.96	0.99	0.00045	0.11	0.08
2	0.50	1.00	1.00	0.12	1.00	0.00033	0.28	0.27
3	0.50	1.00	1.00	0.11	1.00	0.00036	**1.00**	**0.89**
4	0.50	1.00	1.00	0.95	1.00	0.00000	0.00	0.03
5	0.50	1.00	1.00	0.09	1.00	0.00022	0.17	0.25
6	0.50	1.00	1.00	0.07	1.00	0.00035	**0.94**	**0.87**
7	0.50	1.00	0.99	0.96	0.99	0.00045	0.11	0.08
8	0.50	1.00	1.00	0.14	1.00	0.00028	0.22	0.26
9	0.50	1.00	1.00	0.11	1.00	0.00034	**0.89**	**0.85**
10	0.50	1.00	0.99	0.95	0.99	0.00000	0.00	0.05
11	0.50	1.00	1.00	0.17	1.00	0.00010	0.06	0.19
12	0.50	1.00	1.00	0.11	1.00	0.00035	**0.94**	**0.88**

Because of the better performance when we fix $\theta = 0.5$, we compare the AdaBoostRegressor trained by discrete labels (AdaBoostRegressor (D)) and it trained by continuous labels (AdaBoostRegressor (C)) in table 4.6 and table 4.7.

Table 4.6. key evaluation of AdaBoostRegressor (D) with $\theta = 0.5$

AdaBoostRegressor (D) ($\theta = 0.5$, chronologically divided)									
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$	A_{PR}
A						0.00085	0.06	0.00	0.00034
B						0.00085	0.06	0.02	0.00034
1	0.50	0.98	0.99	0.88	0.99	0.00050	0.11	0.07	0.00036
2	0.50	1.00	1.00	0.13	1.00	**0.00068**	**0.61**	**0.29**	0.00026
3	0.50	0.99	1.00	0.12	1.00	0.00034	**0.89**	**0.85**	**0.00024**
4	0.50	1.00	1.00	0.93	1.00	0.00000	0.00	0.02	0.00034
5	0.50	1.00	1.00	0.15	1.00	0.00024	0.11	0.15	0.00043
6	0.50	1.00	1.00	0.07	1.00	0.00035	**0.94**	**0.87**	**0.00026**
7	0.50	0.98	0.99	0.88	0.99	0.00049	0.11	0.07	0.00032
8	0.50	1.00	1.00	0.12	1.00	**0.00056**	**0.50**	**0.29**	0.00030
9	0.50	0.99	1.00	0.11	1.00	0.00031	**0.83**	0.87	0.00028
10	0.50	0.99	0.99	0.89	0.99	0.00000	0.00	0.05	0.00035
11	0.50	1.00	1.00	0.13	1.00	0.00014	0.11	0.25	0.00029
12	0.50	0.98	1.00	0.12	1.00	0.00036	**0.94**	**0.86**	0.00030

Table 4.7. key evaluation of AdaBoostRegressor (D) with $\theta = 0.5$

AdaBoostRegressor (C) ($\theta = 0.5$, chronologically divided)									
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$	A_{PR}
A						0.00085	0.06	0.00	0.00034
B						0.00085	0.06	0.02	0.00034
1	0.50	0.55	1.00	0.18	1.00	0.00021	0.11	0.17	0.00032
2	0.50	0.55	1.00	0.10	1.00	0.00042	0.44	0.34	0.00037
3	0.50	0.36	1.00	0.09	1.00	0.00035	**1.00**	**0.92**	0.00023
4	0.50	0.58	1.00	0.13	1.00	0.00000	0.00	0.11	0.00032
5	0.50	0.61	1.00	0.08	1.00	0.00024	0.22	0.30	0.00034
6	0.50	0.34	1.00	0.05	1.00	0.00032	**1.00**	**0.99**	0.00038
7	0.50	0.55	1.00	0.18	1.00	0.00022	0.11	0.16	0.00030
8	0.50	0.62	1.00	0.11	1.00	0.00044	0.44	0.32	0.00033
9	0.50	0.35	1.00	0.09	1.00	0.00034	**1.00**	**0.95**	0.00022
10	0.50	0.55	1.00	0.17	1.00	0.00022	0.11	0.17	0.00032
11	0.50	0.62	1.00	0.11	1.00	0.00022	0.22	0.32	0.00036
12	0.50	0.35	1.00	0.08	1.00	0.00033	**1.00**	**0.96**	0.00031

In order to check if the continuous labels have any influence to the result, so we compare these two tables. In terms of training score, it has distinguishing $p1$ so that the models trained by T seem to have tendency to overfit in this case. Some of $p3$ by the IHT is better because of the relatively low $Test\ rate$. Besides, We can see the IHT is lower when use the model of AdaBoostRegressor (C), instead the NM is higher in terms of $r3$. However, the extremely high $Test\ rate$ is an impact in this case, so we can also observe the T_cont may enhance the cost in terms of rising $Test\ rate$. As for A_{PR}, it is also extremely low like $p3$, so we just compare the A_{PR} when we have difficult deciding which training set is better. For example, when we compare the $Training\_3$ and $Training\_6$ in table 4.6, it is difficult telling which one is better because we want to get higher $r3$ and lower $Test\ rate$ in the meantime. Therefore, we can compare them by A_{PR}, so we choose the $Training\_6$ which has the relatively better performance based on A_{PR}.

Compare to AdaBoost and AdaBoostRegressor, XGBoost has better performance and One Class SVM usually with high average $r3$ and $Test\ rate$ by 14 kinds of result, shown in table 4.8 and table 4.9 individually.

Table 4.8. key evaluation of XGBoost with $\theta = 0.5$

XGBoost ($\theta = 0.5$, chronologically divided)								
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$
A						0.00	0.00	0.00
B						0.00	0.00	0.02
1	0.50	0.98	0.98	0.86	0.98	0.00079	0.11	0.05
2	0.50	1.00	1.00	0.12	1.00	0.00023	0.17	0.24
3	0.50	1.00	0.97	0.11	0.97	0.00039	**1.00**	**0.83**
4	0.50	1.00	1.00	0.95	1.00	0.00000	0.00	0.02
5	0.50	1.00	1.00	0.09	1.00	0.00027	0.17	0.20
6	0.50	1.00	1.00	0.07	1.00	0.00035	**0.94**	**0.86**
7	0.50	0.99	0.98	0.89	0.98	0.00081	0.11	0.04
8	0.50	1.00	1.00	0.13	1.00	0.00008	0.06	0.21
9	0.50	1.00	0.98	0.11	0.98	0.00039	**1.00**	**0.83**
10	0.50	1.00	0.99	0.96	0.99	0.00000	0.00	0.03
11	0.50	1.00	1.00	0.13	1.00	0.00018	0.11	0.20
12	0.50	1.00	1.00	0.12	1.00	0.00036	**0.89**	**0.78**

Table 4.9. key evaluation of One Class SVM with $\theta = 0.5$

One Class SVM ($\theta = 0.5$, chronologically divided)								
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$
1	0.50	0.32	0.92	0.08	0.92	0.00034	0.89	0.84
2	0.50	0.31	0.86	0.07	0.86	0.00032	0.89	0.89
3	0.50	0.30	0.84	0.09	0.84	0.00027	0.28	0.33
4	0.50	0.32	0.89	0.05	0.89	0.00035	0.89	0.83
5	0.50	0.27	0.71	0.04	0.71	0.00035	0.89	0.81
6	0.50	0.32	0.87	0.07	0.87	0.00024	0.22	0.30
7	0.50	0.32	0.92	0.08	0.92	0.00033	0.89	0.86
8	0.50	0.29	0.81	0.07	0.81	0.00033	0.89	0.88
9	0.50	0.30	0.84	0.09	0.84	0.00027	0.28	0.33
10	0.50	0.32	0.92	0.07	0.92	0.00034	0.89	0.84
11	0.50	0.29	0.80	0.06	0.80	0.00033	0.89	0.88
12	0.50	0.30	0.82	0.08	0.82	0.00026	0.28	0.34

Other models are relatively fail with them, and the models which is trained by chronologically division and $\theta = 0.5$ are shown in appendix A. Moreover, we can simulate a best model which misclassify the data only when α and β are overlapping, and we can observe the idea PRC with minimum $Test\ rate$, shown in figure 4.4. We can know the idea $Test\ rate$ has significant difference with the $Test\ rate$ of the candidates above.

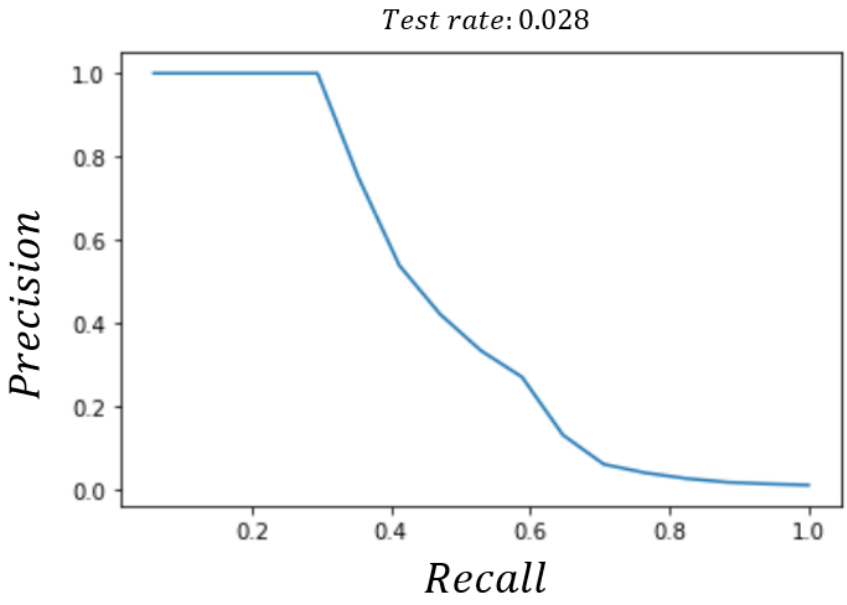


Figure 4.4. Idea PRC when testing set is from M7 and M8

Due to the better performance when we fix $\theta = 0.5$ and chronologically divide the dataset. Therefore, we carried out the experiment which testing set is divided randomly and the IHT of US enhanced a lot, table 4.10 shows the result of AdaBoost calculated by the mean of 5-folds.

Table 4.10. key evaluation of AdaBoost with $\theta = 0.5$ (Randomly)

AdaBoost ($\theta = 0.5$, randomly divided)								
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$
A						0.0000	0.00	0.00
B						0.0000	0.00	0.00
1	0.50	0.81	0.43	0.32	0.39	0.0011	0.14	0.06
2	0.50	0.98	0.93	0.07	0.87	0.0006	**0.81**	**0.69**
3	0.50	0.95	0.87	0.07	0.85	0.0005	**0.76**	**0.69**
4	0.50	0.67	0.50	0.09	0.37	0.0008	0.20	0.13
5	0.50	0.99	0.99	0.04	0.92	0.0005	0.71	0.73
6	0.50	1.00	0.99	0.04	0.96	0.0005	**0.83**	**0.82**
7	0.50	0.81	0.43	0.31	0.38	0.0011	0.14	0.07
8	0.50	0.95	0.88	0.08	0.81	0.0006	**0.69**	**0.62**
9	0.50	0.97	0.92	0.07	0.91	0.0005	**0.79**	**0.77**
10	0.50	0.75	0.43	0.25	0.43	0.0011	0.21	0.09
11	0.50	0.96	0.89	0.07	0.89	0.0005	**0.71**	**0.64**
12	0.50	0.99	0.97	0.07	0.97	0.0005	**0.81**	**0.77**

Because of the better performance of XGBoost of chronological version, we also check if XGBoost is still better among six models. It seems that the AdaBoost has more training sets to choose for the candidates, and also has advantage with better $r3$.

Table 4.11. key evaluation of XGBoost with $\theta = 0.5$ (Randomly)

XGBoost ($\theta = 0.5$, randomly divided)								
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$
A						0.0000	0.00	0.00
B						0.0004	0.03	0.03
1	0.50	0.99	0.98	0.92	0.98	0.0012	0.09	0.04
2	0.50	1.00	1.00	0.09	1.00	0.0010	**0.73**	**0.61**
3	0.50	1.00	0.98	0.07	0.98	0.0002	**0.76**	0.79
4	0.50	1.00	1.00	0.91	1.00	0.0006	0.05	0.04
5	0.50	1.00	1.00	0.06	1.00	0.0004	**0.59**	**0.58**
6	0.50	1.00	1.00	0.04	1.00	0.0006	**0.79**	0.80
7	0.50	0.99	0.98	0.92	0.98	0.0008	0.06	0.04
8	0.50	1.00	1.00	0.10	1.00	0.0004	**0.60**	**0.54**
9	0.50	1.00	0.98	0.07	0.98	0.0002	**0.80**	0.82
10	0.50	1.00	0.99	0.93	0.99	0.0012	0.09	0.04
11	0.50	1.00	1.00	0.12	1.00	0.0006	**0.56**	**0.47**
12	0.50	1.00	1.00	0.07	1.00	0.0004	**0.76**	0.81

Naïve Bayesian here has more candidates than XGBoost as well. Besides, $Training\_A$ has a relatively distinguished compare to the other models, but it fail when we use the $Training\_B$ as the training set. In this result, it shows the decline after undergoing the RS and OT. However, this model do not take the mutual relationship between features into account so that there is more difference among models.

Table 4.12. key evaluation of Naïve Bayesian with $\theta = 0.5$ (Randomly)

Naïve Bayesian ($\theta = 0.5$, randomly divided)								
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$
A						0.0010	**0.51**	**0.27**
B						0.0004	0.39	0.37
1	0.50	0.59	0.61	0.15	0.61	0.0006	0.35	0.23
2	0.50	0.89	0.93	0.09	0.93	0.0008	**0.72**	**0.61**
3	0.50	0.88	0.87	0.07	0.87	0.0008	**0.82**	**0.78**
4	0.50	0.53	0.83	0.08	0.83	0.0004	0.41	0.37
5	0.50	0.94	0.90	0.05	0.90	0.0004	**0.54**	0.62
6	0.50	0.96	0.88	0.03	0.88	0.0004	**0.91**	**0.89**
7	0.50	0.59	0.61	0.16	0.61	0.0006	0.35	0.23
8	0.50	0.87	0.90	0.09	0.90	0.0004	**0.62**	**0.58**
9	0.50	0.93	0.88	0.06	0.88	0.0006	**0.84**	**0.81**
10	0.50	0.57	0.60	0.14	0.60	0.0006	0.33	0.24
11	0.50	0.87	0.90	0.08	0.90	0.0006	**0.62**	**0.59**
12	0.50	0.93	0.89	0.06	0.89	0.0006	**0.84**	**0.82**

Overall, the result of classifiers can be observed by different data augmentation while the AdaBoostRegressor can also get optimal $p3$ and $r3$, figure 4.5 shows two of the PRC and the evaluation when $THR = 0.5$ marked by green dot in this case. It is obvious that the current evaluation are always not optimal, thus it is tough mission to choose a best THR for all the models can get better performance constantly.

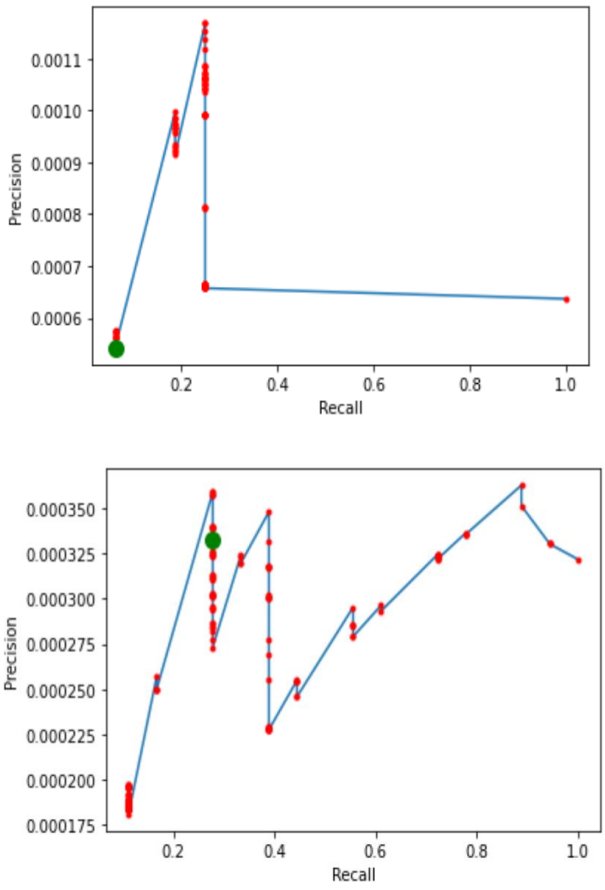


Figure 4.5. The PRC of AdaBoostRegressor and the point with $THR = 0.5$

To sum up, we can observe the $p3$, $r3$ and $Test\ rate$ (Aging_rate) together in figure 4.6, the IHT of US is best especially doing the OS before US ($Training\_2$). In terms of $p3$, the last six training sets are better while the first 3 is better in terms of $r3$. Besides, One Class SVM don't have better performance with T because the $Test\ rate$ are almost the same or higher than their $r3$. However, the result with better $p3$ usually has extremely poor performance in $r3$.

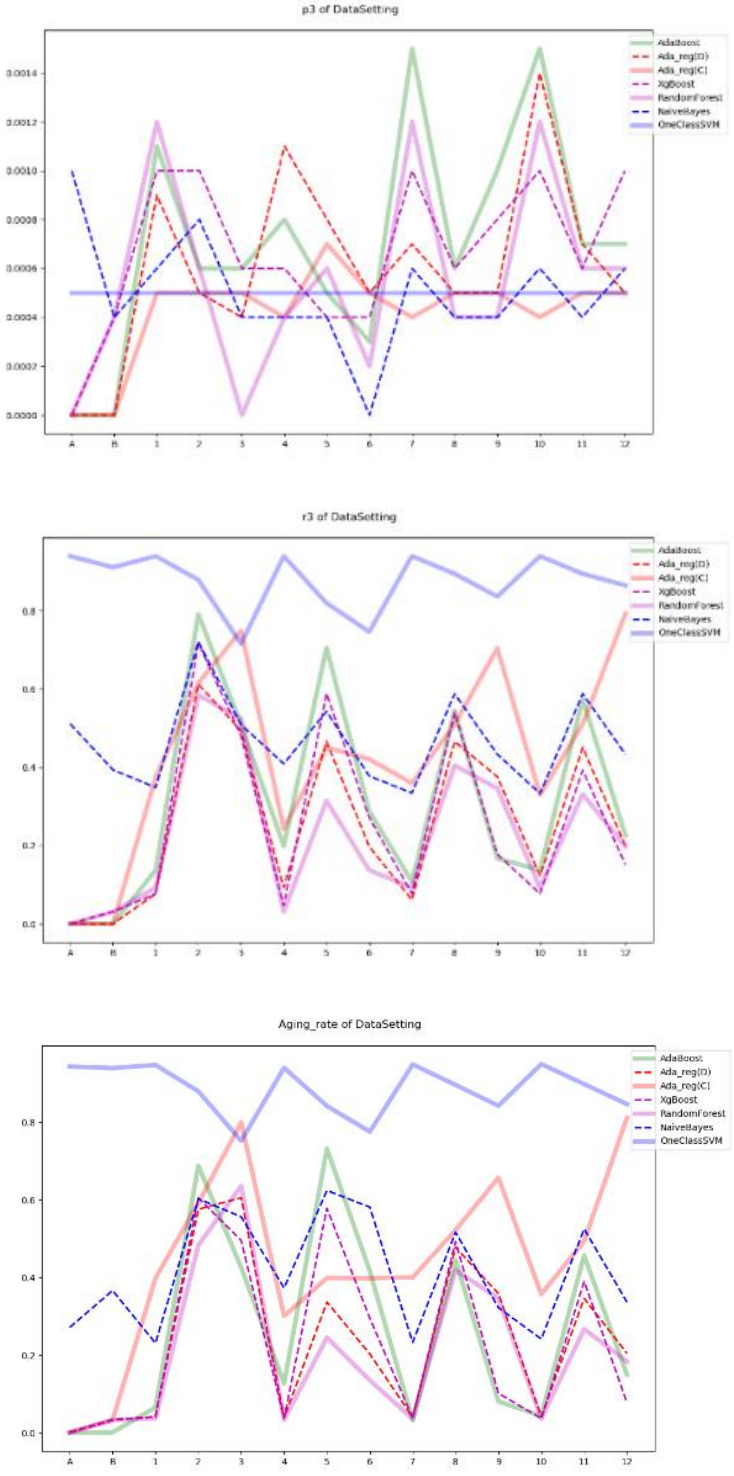


Figure 4.6. The curves of $p3$, $r3$ and $Test\ rate$ with different models

Finally, we only obtain the information of total number with classifying correctly above, but we don't have idea about what kind of α is hard to detect in most of models. Therefore, we list and compare each α for knowing their detective level in this case in

table 4.12. If the Detective Level is equal to 1, it means the sample is easy to recognize by most of the models.

Table 4.12. The hardness of each kind in minority class

Detectable Level	***α_ID***	**Detectable Level**	***α_ID***
1	2	11	23
2	0	12	6
3	1	13	59
4	15	14	14
5	3	15	44
6	41	16	43
7	32	17	36
8	17	18	60
9	16	19	21
10	12	20	26

From this table, we can find out that the data including 0, 1, 2 and 3 which occur more than once is easily detected by models. However, the data such like 21 or 26 is hard to classify correctly, so it may be our next research by merging different models which can classify the data like 21 or 26 to cope with problems from hard samples.

5 Conclusion and future research

This paper has a new interpretation of the fundamental characteristics of extremely imbalanced data under the constraints of categorical variables, designing an innovative oversampling method. However, most of research solved this problem with extremely imbalanced by US more, we find out that the OS can also improve the result to some degree, revising the SMOTE-N for the imbalanced data under the constraints of binary variables.

Although $Training\_A$ and $Training\_B$ are not the main concern of this paper, but we can observe the effectiveness by this benchmark, and find out that RS and OT are relatively insignificant for the classification. There is significant difference between chronological division and randomly one. When we test by the last months, NM dominate the others. However, the mean of the five result from IHT improved a lot when we divided the training set and testing set randomly.

Overall, T without OS usually is dominated by the other with OS, that is the purposed technique can improve the result. In terms of the combinations of OS and US, we will know it is slightly better when we do OS before US. In addition, the impact from data augmentation is more significant than model's variants, but the precision and recall of regression are not unique, instead it changes depends on the threshold of regression.

In the future research, it is useful to adjust the pre-process techniques for mixed data type. Adopting different similarities metrics for binary variables shall be tested as well, because the other similarities may result in significantly different value of evaluation.

However, the two ways of OS generate different number of new samples, SE can control the generating number easily and tend to be higher than BDR especially with extremely imbalanced data. Thus, we can test if the small amount generated by BDR is sensitive or not. As for US, we can combine the two methods, IHT and NM together because these two methods both have their advantage and disadvantage if we divide the training set and testing set with different ways.

References

[1] Greene, D. and Williams, P. C. *Linear Accelerators for Radiation Therapy*, Second Edition. IOP Publishing Ltd., Bristol and Philadelphia, 1997.

[2] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32.

[3] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321-357.

[4] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(4), 463-484.

[5] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications, 73*, 220-239.

[6] Japkowicz, N. (2001, June). Concept-learning in the presence of between-class and within-class imbalances. In Conference of the Canadian society for computational studies of intelligence (pp. 67-77). Springer, Berlin, Heidelberg.

[7] Mani, I., & Zhang, I. (2003, August). kNN approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets* (Vol. 126). United States: ICML.

[8] Napierala, K., & Stefanowski, J. (2016). Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems, 46*(3), 563-597.

[9] Prati, R. C., Batista, G. E., & Monard, M. C. (2004, April). Class imbalances versus class overlapping: an analysis of a learning system behavior. In Mexican international conference on artificial intelligence (pp. 312-321). Springer, Berlin, Heidelberg.

[10] Smith, M. R., Martinez, T., & Giraud-Carrier, C. (2014). An instance level analysis of data complexity. *Machine Learning, 95*(2), 225-256.

[11] Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment, 62*(1), 77-89.

[12] Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter, 6*(1), 7-19.

[13] Zhou, L. (2013). Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods. *Knowledge-Based Systems, 41*, 16-25.

Appendix A

Table A.1. key evaluation of Random Forest with $\theta = 0.5$

Random Forest ($\theta = 0.5$, chronologically divided)								
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$
A						0.00	0.00	0.00
B						0.00	0.00	0.02
1	0.50	1.00	0.99	0.98	0.99	0.00066	0.11	0.05
2	0.50	1.00	1.00	0.16	1.00	0.00032	0.22	0.22
3	0.50	1.00	1.00	0.12	1.00	0.00033	**0.89**	**0.87**
4	0.50	1.00	1.00	0.99	1.00	0.00000	0.00	0.02
5	0.50	1.00	1.00	0.17	1.00	0.00015	0.06	0.12
6	0.50	1.00	1.00	0.07	1.00	0.00034	**0.83**	**0.80**
7	0.50	1.00	0.99	0.98	0.99	0.00083	0.17	0.06
8	0.50	1.00	1.00	0.17	1.00	0.00019	0.11	0.19
9	0.50	1.00	1.00	0.11	1.00	0.00034	**0.94**	**0.90**
10	0.50	1.00	0.99	0.96	0.99	0.00000	0.00	0.04
11	0.50	1.00	1.00	0.21	1.00	0.00012	0.06	0.15
12	0.50	1.00	1.00	0.13	1.00	0.00035	**0.94**	**0.86**

Table A.2. key evaluation of Naïve Bayesian with $\theta = 0.5$

Naïve Bayesian ($\theta = 0.5$, chronologically divided)								
T	θ	$p1$	$r1$	$p2$	$r2$	$p3$	$r3$	$Test\ rate$
A						0.00052	0.39	0.24
B						0.00030	0.11	0.12
1	0.5	0.67	0.68	0.26	0.68	0.00058	0.61	0.34
2	0.5	0.82	0.94	0.11	0.94	0.00046	0.61	0.43
3	0.5	0.80	0.82	0.10	0.82	0.00034	**0.94**	**0.91**
4	0.5	0.53	0.80	0.11	0.80	0.00027	0.11	0.13
5	0.5	0.99	0.94	0.07	0.94	0.00022	0.17	0.24
6	0.5	0.90	0.83	0.05	0.83	0.00035	**1.00**	**0.91**
7	0.5	0.67	0.68	0.26	0.68	0.00058	0.61	0.34
8	0.5	0.86	0.92	0.12	0.92	0.00042	0.50	0.38
9	0.5	0.85	0.75	0.09	0.75	0.00034	**0.94**	**0.90**
10	0.5	0.61	0.67	0.20	0.67	0.00054	0.61	0.37
11	0.5	0.86	0.93	0.11	0.93	0.00037	0.44	0.38
12	0.5	0.84	0.73	0.09	0.73	0.00034	**0.94**	**0.91**