Marketing: Predicting Customer Behaviors

JIAYU LI	WEN JIA	ZIHAN ZHANG	YI WEI
s1717961	s1723911	s1719186	s1716682

Abstract

In this paper, we transform the challenges of the 2009 KDD Cup into a four classification issue. The main stages of out project includes data cleaning (the treatment on missing value, repeat values, heterogeneous variables, dimension reduction and unbalanced dataset) and tuning a deep neural network model to make predictions. The result illustrates that

1 Introduction

Customer Relationship Management (CRM) is a marketing strategy in managing relationship between companies and customers (including potential customers). It plays a vital role in the modern business environment and can affect a long-term development of companies to a great extent. However, many companies have difficulty in dealing with the rapid increase in customer data due to unexpected technique developments. KDD Cup 2009 was organized to identify various data mining skills that can be used to handle a large dataset from the French Telecom company. The predictive goals can be divided into three aspects: 1) predicting the propensity of customers to switch providers (churn); 2) whether they are willing to buy new products or services (appetency); 3) purchase upgrades or add-ons proposed to them so as to make the sale more profitable (up-selling). In this paper, we decided to combine these labels into four classes, which will be discussed in section 3.

The competition was held in 2009, and related attempts on this task were mostly about ensemble decision trees. Now we focus on this data set again, and try to use deep learning methods to deal with the classification problem. Deep learning technique developed rapidly in recent years, and the development of GPUs helps to significantly boost the training rate of deep neural networks (DNN). Compared with decision trees, deep neural networks are usually better at dealing with large data sets. The aim of the project is using DNN to compare the impact of different data processing approaches on AUC results. To accomplish this, this task faces three major challenges. Firstly, this dataset for the masked customer records provided by French Telecom Company is rather large, including 50,000 samples and 15,000 features (in the large version) or 230 features (in the small version). Such large datasets and unknown conversions from the original data make the prediction tasks particularly challenging. Secondly, both datasets are Heterogeneous and unbalanced, which also suffer from the missing value problem in different level. Thirdly, we want to adopt one model accomplishing three prediction tasks, rather than combining various models like what the participants of the competition did.

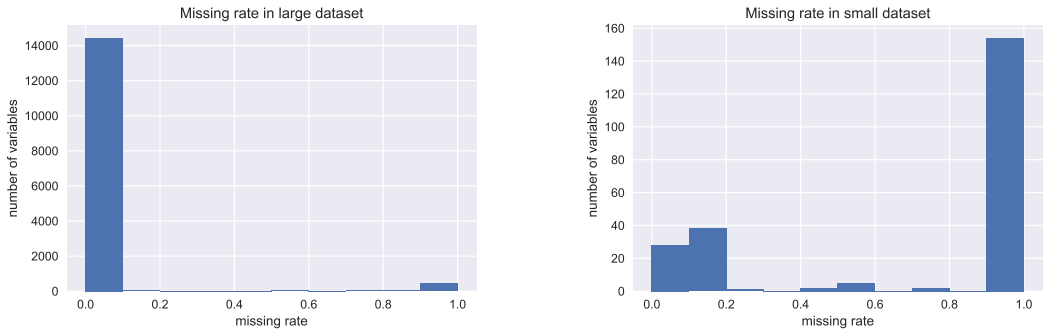
In order to deal with the first the challenges mentioned, we carried out various data processing methods in section 4 targeting different data types and problems. The detailed analysis of original dataset is shown in section 3. The methods used to reduce dimension are explained in section 5. In section 6, we introduce the experiments comparing the effect of data precessing methods and evaluation criteria, and analysis of the results. Finally, We tune hyperparameters in our best model on validation set and test the model performance on the test set.

2 Related work

The task of the KDD Cup 2009 is a classification problem related to the industry, but presenting technical challenges mentioned above. Thus, the core work of this task is to preprocess the data, filter features and then implement the proper classifier algorithm to make predictions. The competition had a fast component predicting for the test data due within 5 days after the full data being released, and a slow component where predictions had to be submitted within 5 weeks, which means the result of fast and slow tracks were evaluated separately. IBM[6] won the challenge in both of the fast (0.8493 score) and slow tracks (0.8521 score). They dealt with the missing data by replacing each one with the mean of the features, and obtained reduced feature set through PCA and filter methods based on Pearson correlation and mutual information. They also generated classifier libraries including random forests and boosted trees and built the ensemble selection by selecting a subset of classifiers from the library which have the best performance. Xie et al.[11] used substitution to handle missing data and mean-variance filter to obtain 1720 variables, then implemented TreeNet . They finally won the second place in the fast track (with 0.8448 score). Lo et al.(2009)[5] generated three main classifiers and post-processed the results of these classifiers to obtain the final result(0.8461 in the slow track). The first classifier is a regularized maximum entropy model, where the heterogeneous and missing data need to be preprocessed. They handled the missing data by filling with 0 and generate a new 0/1 indicator for all data points to show if it has at least one missing value, and they also transform each categorical feature to several binary ones which induced a large number (111, 670) of additional features. The second is Adaboost with s heterogeneous base learner, and the last classifier used selective Naive Bayes.

3 Data preparation

The KDD Cup 2009 has two datasets, one large and one small, each with 50,000 data points. we will first develop our model on the small dataset, then conduct further experiments on the large one. There are 230 features for each data point in the small dataset, and 15,000 features for each data point in the large dataset. The first 190 features in the small data set are numerical, and the last 40 features are categorical while the first 14,740 features in the large dataset are numerical features, and the last 260 features are categorical features. In order to protect the privacy of customers, the datasets do not give the specific meaning of each feature, and each category in the categorical feature is also replaced with meaningless strings.



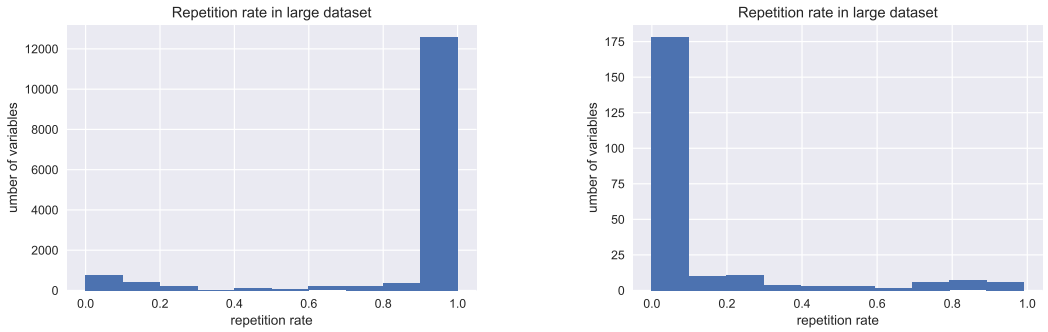
(a) Large dataset

(b) Small dataset

Figure 1: Missing rate distribution in large dataset & small dataset

In addition, the datasets are imbalanced. Most examples of the three binary classification tasks (appetency, upselling, and churn) were negative, and the positive samples only accounted for 1.78%, 7.364%, and 7.344%, respectively. In figure 1, we can see that 3.35% of data in the big dataset is missing. However, the small dataset has a large number of missing values, 69.77% of the data is missing. Specially, the data missing rate of 161 features is greater than 50%, and the missing rate of 153 features is even higher than 90%. In figure 8, the repetition rate of more than 90% variables in

the large dataset is more than 90%. In contrast, the small dataset does not suffer this issue. Therefore, we have different processing tasks on these two datasets.



(a) Large dataset (b) Small dataset

Figure 2: Repetition rate distribution in large dataset & small dataset

In the official website, there is only the test dataset but not the corresponding labels. We have to generate the new test set from the training set by splitting the original training set into three parts: training set, validation set and test set according to 8:1:1, so that the size of each data set is 40,000, 5000, 5000 respectively.

In terms of the labels, we combine the three classification tasks together. Through analyzing the labels of the three tasks, we find that the results of the three tasks are exclusive. That is to say, the three original labels can be combined as a simpler new label. As shown in table 1, the new labels have 4 classes including all situations of the distribution of the old labels, and are encoded by one-hot. Hence, we could make a prediction on all the three tasks simultaneously, taking the relationship between these three indexes into account. As a result, the three tasks are combined as one classification task where there are 4 classes.

given label / transformed label	churn	appetency	upselling
(1,0,0,0)	1	-1	-1
(0,1,0,0)	-1	1	-1
(0,0,1,0)	-1	-1	1
(0,0,0,1)	-1	-1	-1

Table 1: Transformation from the given labels to the new labels, combining three classification tasks to one.

4 Exploratory data analysis

If there is a large amount redundant and irrelevant information present or noisy and unreliable data, then discovering useful knowledge during the training phase will become more difficult [4]. That is why data preprocessing plays a significant role before training models. According to Erhard Rahm and Hong Hai Do [7], data cleaning is used to deal with detecting and removing errors and inconsistencies from the data aiming to enhance the quality of data. Thus, the first step we carried out is to apply data cleaning on our data set as follows:

Missing Values: There are a significant number of variables are poorly populated, which means regarding some variables many values are missing. Firstly, we compute the missing rate of each variable. We suppose that the variables with high missing rate contain quite little useful information, retaining them may adversely affect the learning of the model. Therefore, we discard these variables with the missing rate more than 0.5 directly, and obtain 14,438 numerical variables and 36 categorical variables in the large dataset, 41 numerical variables and 18 categorical variables in the small dataset. There are several well known techniques to approach the missing value problem, in this work, we

utilize a simple substitution under the assumption Missing at Random (MAR) : substituting the missing values with the mean values and mode values in numerical variables and categorical variables respectively.

Heterogeneous dataset: As we mentioned before, the datasets are heterogeneous. They contain both numerical and categorical variables, and the number of categories varies dramatically on different variables (range from two to more than ten thousand). This poses difficulties on most models since they are unfamiliar with handling categorical data. Additionally, it is obvious that the variable is not valuable for our prediction if all the observations of the variable have the same value. We discard the continuous variables which only have single value, and also remove the categorical variables with one categories or more than 100 categories. Then we adopt One-hot method to transform each categorical feature to several binary ones. Each binary feature corresponds to a possible value of the categorical feature. After this step, the number of variables related to categorical ones increased from 36 to 443 in large dataset, and the number varies from 28 to 406 in the small dataset. We also conduct standardization on numerical variables in both datasets. Finally, we have 13,414 variables in the large data set and 447 variables in the small dataset.

So far, we obtain the clean datasets. We use the small dataset to perform our prediction experiment to ensure the feasibility, and then utilize the large dataset to adjust the hyperparameters in our neural network, which is the baseline system of our project.

5 Methodology

It is not uncommon that discovering useful knowledge during the training phase will become more difficult, if there is a large amount redundant and irrelevant information present or noisy and unreliable data[4]. That is why data preprocessing plays a large part before training models and its product will be the final training set. In this section, the main dimensionality reduction techniques and our experiment model structure and setting will be explained as follow.

5.1 Preprocessing techniques

In practice, a range of data preprocessing methods have been applied widely such as data cleaning, transformation, feature extraction and selection. Hopefully, a set of data preprocessing algorithms are capable of performing the best regardless of datasets but this is impossible. In this part, we focus on describing different dimensional reduction approaches applied in our experiments targeting the large data set. There are a broad range of dimensional reduction techniques that can be divided into two classes, linear and nonlinear. We chose the most popular linear one, PCA and a nonlinear approaches to accomplish this transform job (from a high dimension to a low dimension).

Principal component analysis (PCA): In general, PCA aims to find principal components (PCs) that explain the maximum amount of variance possible through n linearly transformed components[3]. For example, the first principal component is calculated such that it accounts for the greatest possible variance in the data set. The later PC is computed in the same way, displaying a decreasing among of variance except that it is uncorrelated with the previous ones. It is given by the linear combination of all variables and their relative weights (PC = $weight_1 * variable_1 + weight_2 * variable_2 + ... + weight_n * variable_n$). Since the value of PC can become larger with the increasing of weights (Var[PC] = $Var[weights * variables] = weights^T * \sum * weights$], the sum of squares of these weights are subject to equal 1. In this way, all of the original information has been accounted for, as the sum of the variances of all of the PCs will be the same as the variances of all of the variables. In other words, by means of reducing the number of PCs, PCA finds a low-dimensional embedding of the data points that best preserves their variance as measured in the high-dimensional input space.

However, many data sets contain essential nonlinear structures that are invisible to PCA. For example, classical linear method fails to detect the true degrees of freedom of the face data set[10]. That is why we introduce a non-linear method in our model as a comparison.

Neural Network Autoencoder: Many popular no-linearity dimensionality reduction methods including Isomap and locally linear embedding (LLE) have been applied on our data, but they take quite long time to complete the task. Hence, we choose another nonlinear dimension reduction method, neural network autoencoder, takes advantage of training a multilayer neural network with a small central layer to reconstruct high-dimensional inputs vectors. Hinton and Salakhutdinov [2]proposed a

pre-training method in initializing the weights that permits deep autoencoder to study low-dimensional codes. Also, in their experiments, autoencoder outperformed the above three methods in training MNIST data set. Simply, autoencoder can be regarded as a neural network whose output is same or similar to that of its input[8]. Specially, a shallow autoencoder has only one hidden layer (total 3 layers) while a deep autoencoder has more than three layers. Hinton also described that Restricted Boltzmann Machines (a two-layer network) stated by smolensky[9] can be used to find good set of weights to initialize the neural networks.

5.2 Neural Network Architecture

The main model we used in this task is deep neural networks with six hidden layers. Every layer of DNN is a linear combination of the input features, followed by a non-linear activation (Relu or Softmax) layer, shown in figure 3. The dataset that will experience different processing processes is used to be the input of the network. Besides, we organize the parameter settings into table 2 and apply the same model setting in later experiments.

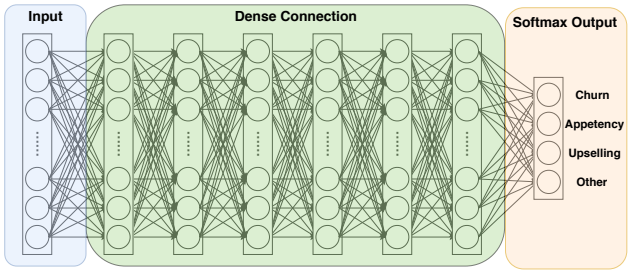


Figure 3: Structure of neural networks.

Hyperparameter Setting	
Activation Function	ReLU
Output Layer	Softmax
Number of Hidden Units	256
Learning Rate	0.0001
Learning Rule	Adam
L2 Regularization	0.01
Batch size	1024
Epoch Number	150

Table 2: Hyperparameter setting.

6 Results and Analysis

In section 3, we preliminarily preprocessed the dataset, making sure that the neural networks could be run, which is used as the baseline experiments. The networks firstly were fed by the small dataset, and ran smoothly, so we then focus on the big dataset. In this section, we conducted three contrast experiment cases with further data processing methods that formed reliable comparison with the baseline. These experiments are evaluated by AUC according to the competition's rule. This section focuses on representing experiment results and related analysis.

The results of the baseline with respect to churn, appetency, upselling and the whole are shown in figure 4, respectively. Apparently, the line graphs all show slight overfitting since the validation line saw a drop regarding the AUC value. In particular, upselling AUC overfits most seriously as its validation AUC declined from 0.62 to 0.55 and fluctuates constantly. With regard to these three aspects, churn validation AUC achieved the highest around 0.78 within the first few epochs then remained at 0.73. As a whole, the best validation AUC is almost 0.75 despite overfitting appeared. In addition, we can see that training AUC in all four graphs closed to 1 during training process, which implies the networks have adequate capacity of modeling the data. In order to form clearer comparison, only validation AUC is represented in the later experiments.

The following three cases are built on the basis of the processed large dataset in the baseline. At the same time, parameter setting in the DNN model and other processing aspect remain the same with the baseline.

Case 1: Deleting the features with more than 50% repeated values.

Case 2: Adjusting the balance among the four classes with three labels (see data preparation) on the grounds of dataset in case 1 (labeled as unbalanced in the result graph). To be specific, we give class weights to different classes into the loss function, where class weights are calculated based on the sample number of each class.

According to figure 5, the left line graph illustrates that the outcome of dataset processed in case 1 outperformed that of the baseline. Although both of them seem to have overfit obviously, the

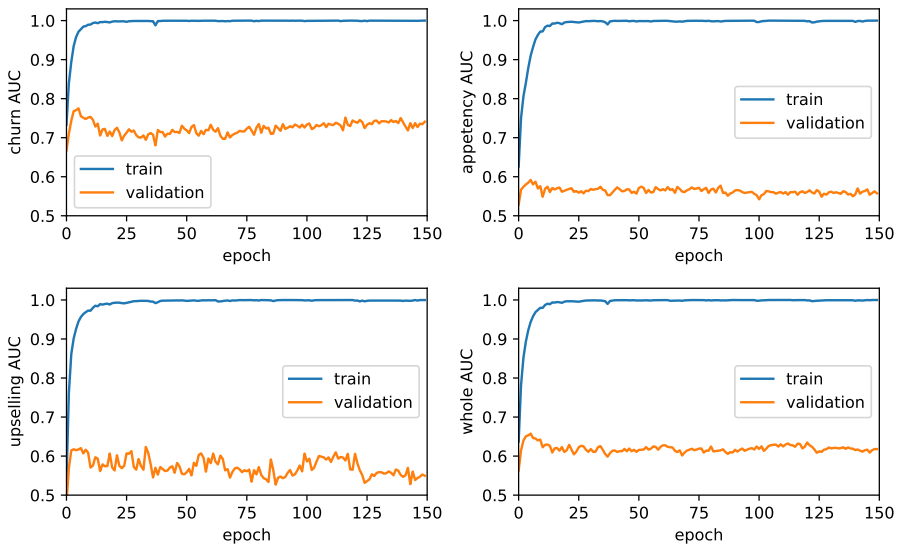


Figure 4: AUC of three tasks and whole AUC of the baseline experiments, on training and validation sets.

preprocessing technique in the case 1 presents around 0.05 validation AUC performance improvement. However, the large dataset experienced the process of cases 2 fails to show the apparent enhancement in our DNN model's validation AUC. This can be convinced from fawcett's paper[1], AUC is not sensitive to balanced and unbalanced dataset. can be see from the right graph, the red line is the same in the left figure. These two lines are very close and appear the similar trend including overfitting, despite the best AUC value of balanced dataset was higher than that of the case 1 dataset.

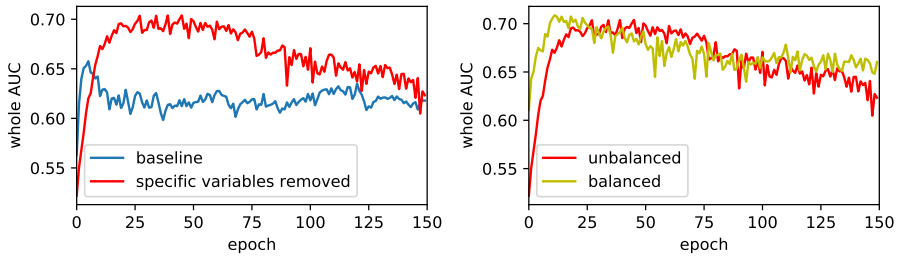


Figure 5: The whole AUC of before and after removing the variables of high repetition rate (left), and before and after the balance operation (right).

Case 3: Adding dimension reduction methods on the dataset processed in case 2 and comparing the linear PCA and non-linear NN methods.

As can be see from the following line graph that includes seven lines in total, three of them means the validation performance using PCA dimension reduction, three of them indicate the validation results using NN method and the light green line coming from the processed dataset in case 2. Interesting, the best AUC value of the dataset without dimensional reduction is close to that of datasets precessed by PCA and NN (the blue line) with only approximate 0.1 gap. Although model performance appears significant overfit using PCA in contrast to the marginal overfitting behavior using NN, the best AUC is obtained when dataset dimension is reduced to 500 by PCA, at 0.76. That is why we chose to adopt PCA to reduce the dataset to 500 dimensions. Then we conducted a lot of trails like adjusting parameter choices in PCA in order to further improve validation performance.

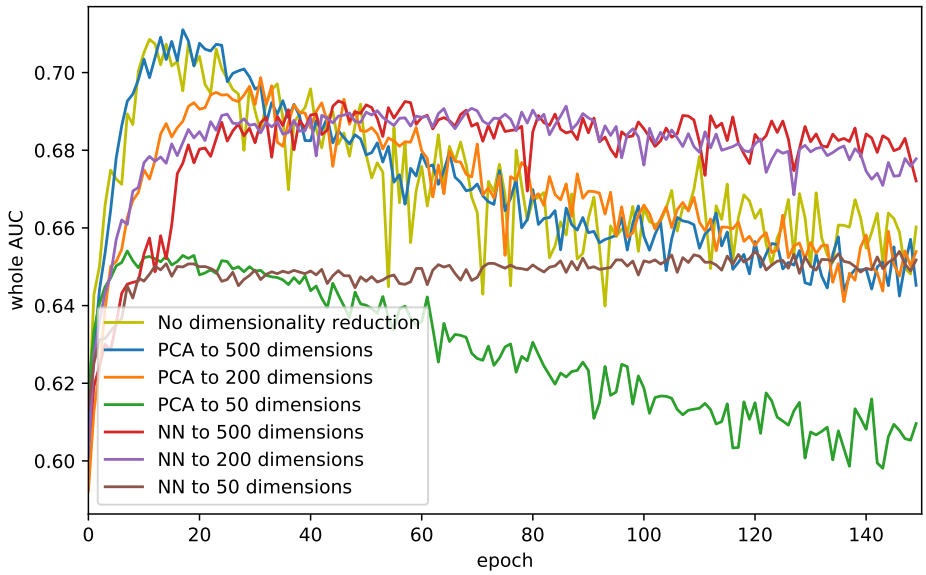


Figure 6: The whole AUC of the several dimensionality reduction methods.

Based on the above experiments, the best model performance is obtained through comparing validation accuracies in case 3 using PCA to reduce data dimensions to 500. After this, the held-out test set is used to test the best model's generation to unseen dataset.

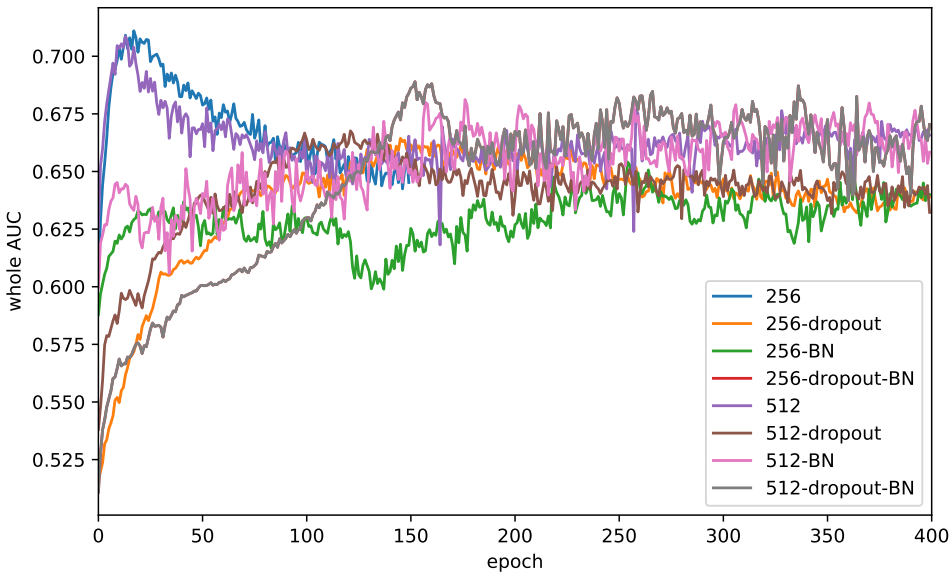
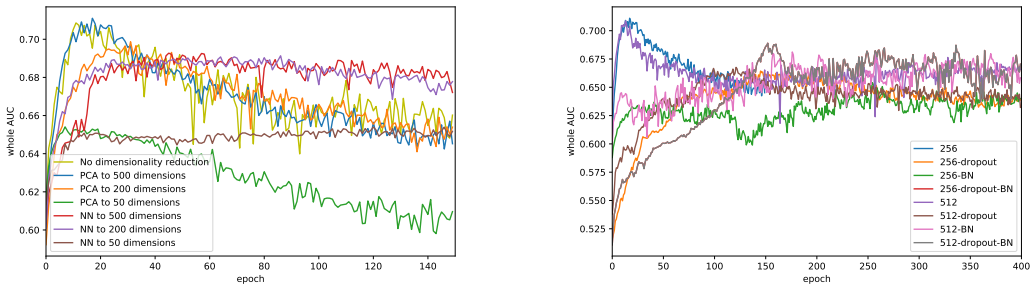


Figure 7: The whole AUC of the several dimensionality reduction methods.



(a) Large dataset

(b) Small dataset

Figure 8: Repetition rate distribution in large dataset & small dataset

7 Conclusion

In conclusion, a broad range of data processing techniques have been tried and related comparison experiments have been tested in our paper, targeting this competition task. Dataset experienced repeated value deletion, balance label classes and dimensionality reduction does perform higher return on validation AUC based on our experiments.

References

[1] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[2] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[3] Steven M Holland. Principal components analysis (pca). *Department of Geology, University of Georgia, Athens, GA*, pages 30602–2501, 2008.

[4] SB Kotsiantis, D Kanellopoulos, and PE Pintelas. Data preprocessing for supervised leaning. *International Journal of Computer Science*, 1(2):111–117, 2006.

[5] Hung-Yi Lo, Kai-Wei Chang, Shang-Tse Chen, Tsung-Hsien Chiang, Chun-Sung Ferng, Cho-Jui Hsieh, Yi-Kuang Ko, Tsung-Ting Kuo, Hung-Che Lai, Ken-Yi Lin, Chia-Hsuan Wang, Hsiang-Fu Yu, Chih-Jen Lin, Hsuan-Tien Lin, and Shou de Lin. An ensemble of three classifiers for kdd cup 2009: Expanded linear model, heterogeneous boosting, and selective naive bayes. In Gideon Dror, Mar Boullé, Isabelle Guyon, Vincent Lemaire, and David Vogel, editors, *Proceedings of KDD-Cup 2009 Competition*, volume 7 of *Proceedings of Machine Learning Research*, pages 57–64, New York, New York, USA, 28 Jun 2009. PMLR.

[6] Alexandru Niculescu-Mizil, Claudia Perlich, Grzegorz Swirszcz, Vikas Sindhwani, Yan Liu, Prem Melville, Dong Wang, Jing Xiao, Jianying Hu, Moninder Singh, Wei Xiong Shang, and Yan Feng Zhu. Winning the kdd cup orange challenge with ensemble selection. In Gideon Dror, Mar Boullé, Isabelle Guyon, Vincent Lemaire, and David Vogel, editors, *Proceedings of KDD-Cup 2009 Competition*, volume 7 of *Proceedings of Machine Learning Research*, pages 23–34, New York, New York, USA, 28 Jun 2009. PMLR.

[7] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[8] Ayushman Singh Sisodiya. Reducing dimensionality of data using neural networks.

[9] Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986.

[10] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[11] Jianjun Xie, Viktoria Rojkova, Siddharth Pal, and Stephen Coggeshall. A combination of boosting and bagging for kdd cup 2009 - fast scoring on a large database. In Gideon Dror, Mar Boullé, Isabelle Guyon, Vincent Lemaire, and David Vogel, editors, *Proceedings of KDD-Cup 2009 Competition*, volume 7 of *Proceedings of Machine Learning Research*, pages 35–43, New York, New York, USA, 28 Jun 2009. PMLR.