CSE574 Milestone 2 Report

## Team Member

Name ： PinLyu Huangwei Ding Matthew Sah

Email ID: [pinlyu@buffalo.edu](mailto:pinlyu@buffalo.edu) [hding3@buffalo.edu](mailto:hding3@buffalo.edu) [msah@buffalo.edu](mailto:msah@buffalo.edu)

UB Person ID: 50321560 50155734 50320376

# Topic

# Improvements on the Prediction of Diabetes and its Relevance to Dietary Habits

## Abstract

In our attempt to make improvements to our model, the biggest challenge we've came across was insufficient positive diabetes patient data. Our data was highly unbalanced in favor of the healthy and our model was greatly affected by this imbalance. Our objective during this session was to resolve the issue with imbalance data and how we improved our model based on this circumstance.

## 

## Issues

There were some significant issues we've noticed whilst improving our implementation of our diabetes prediction model.

Firstly, although the results of the models we've used showed obvious low error rates, those are not good signs of modeling since we noticed that our data was inevitably imbalanced. The people who classify as Type 2 diabetes are the minority in the crowd. In this way, we felt that the original naive way of calculating error rate was imprecise or even to say incorrect.

To follow, in our original method, the training set is picked by randomly choosing 80% of the whole dataset, which could lead to some situation where there is no Positive case(i.e., people with Type 2 diabetes) for testing. In this way, the result of our model may be even more inaccurate. Imbalanced data very commonly results in biased predictions and misleading accuracies. This is mainly because the model has insufficient information about the minor class to make an accurate prediction.

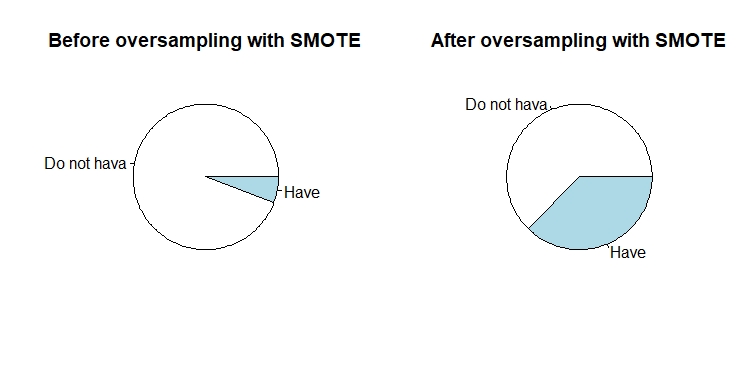
Therefore we focused on the unbalanced data as a main objective in milestone 2.

## Improvements

To resolve the issues we've come across, the issue of imbalanced classes in our dataset. We've come to a solution of a few methods.

The first thing we did to improve our model was to solve the accuracy paradox. The reasoning behind this was due to an scenario when the model can make a single prediction for the whole data(eg. this patient does not have diabetes) and still receive extremely high accuracy due to very low percentage of the patients having diabetes, resulting in the accuracy only reflecting the underlying class distribution. To solve this we updated the way we calculated our accuracy, from the original target accuracy to F1 score as a metric. Using the f-score, we can get an average of the precision score(measurement of classifier's exactness) and the recall score(measurement of the completeness of the classifier) for both classes, which in overall provides better insight on the performance of our model.

The second improvement we made was the resampling of the data. Initially we just split our data with a 8:2 train-test ratio. An example of the issue we came across was when our testing data consisted of 0 diabetes patients therefore granting perfect accuracy. Now we've implemented cross validation to help get more balanced results with different combinations of the dataset, and possibly get better results for future unseen data.

Another step we took in resampling the data was oversampling the diabetes patients from the training data set. Originally the patients only took up around 1/10, by creating more samples in the diabetes category of the training set with SMOTE and testing out the F1 score with the testing set, we achieved significantly greater accuracy for both the negative and positive patients. 

*Figure 1. Modifying our dataset using SMOTE*

Typically the issue that over sampling would cause is overfitting to the original data, and distorting the original shape of the data. We used methods such as ROSE and SMOTE (Figure 1) as informative oversampling to oversample the diabetes patients in the training dataset. We also randomly undersampled the patients without diabetes to bring the data closer to a 1:1 ratio. Other figures of the methods implemented and compared will be shown later in this report.

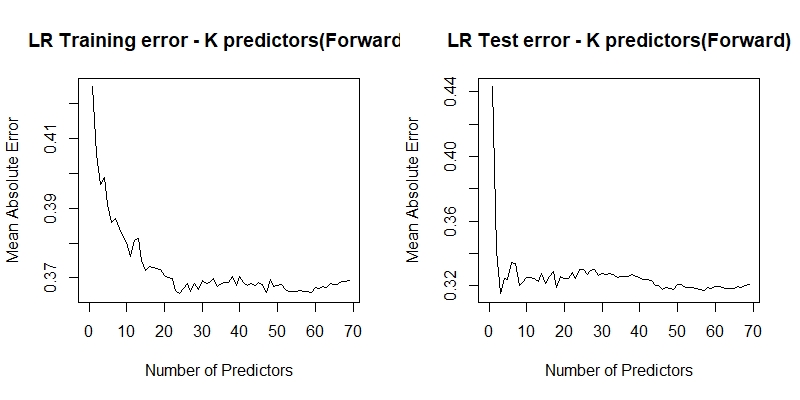
In brief, SMOTE generates artificial samples based on the feature space(rather than the data space) of the minority class's k-nearest-neighbors. Whereas ROSE uses smoothed bootstrap to draw artificial samples from the feature space neighborhood of the minority class. This is called informative resampling, since these two methods don't produce duplicated data randomly but rather produces what it believes is also considered a minority case data.

We deduced that resampling was the best way to resolve imbalanced dataset instead of tuning the data and would provide a better result than modifying the model used. The results will be shown later in this report.

The third improvement we've tried out was using non-linear classifiers; in this case we're using Classification and Regression Trees, also known as CART, along with gradient boosted decision trees, methods such as XGBoost. The reasoning behind this step was due to the characteristics of decision trees, decision trees tend to perform pretty well on imbalanced data since the splitting rules that look at class variables used in the creation of trees can force both classes to be addressed. XGBoost was also selected for being known for improving performance as well as execution speed despite speed not being a target of our project in this case.

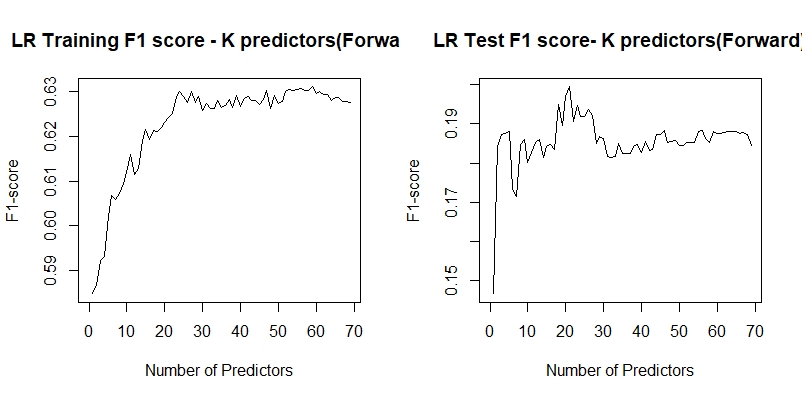
## 

## Results



### Figure 2. Logistic Regression Training/Testing Error

We observe the training error using the original accuracy method to visualize how well this metric works for Logistic Regression. We can see that after using a certain number of accuracies, the RMSE decreases sharply to around 30 percent, which in theory should yield a pretty sufficient model. However with our understanding of our imbalanced data, we know that this metric is incorrect and we then proceed to verify with F1 score as our new metric to verify our model's accuracy.

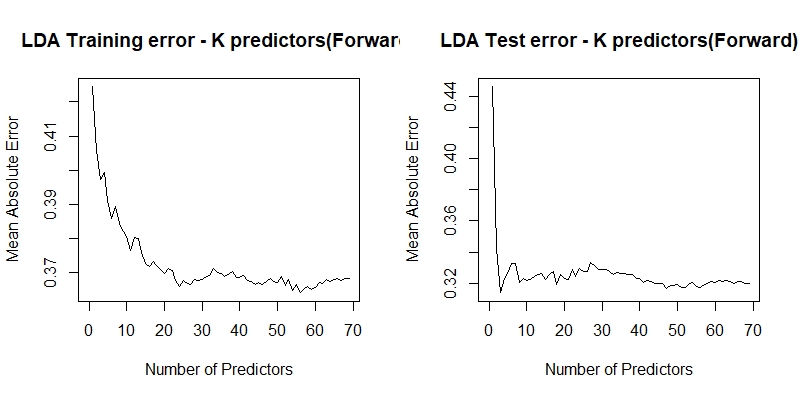


### Figure 3. Logistic Regression Training/Testing F1 Score

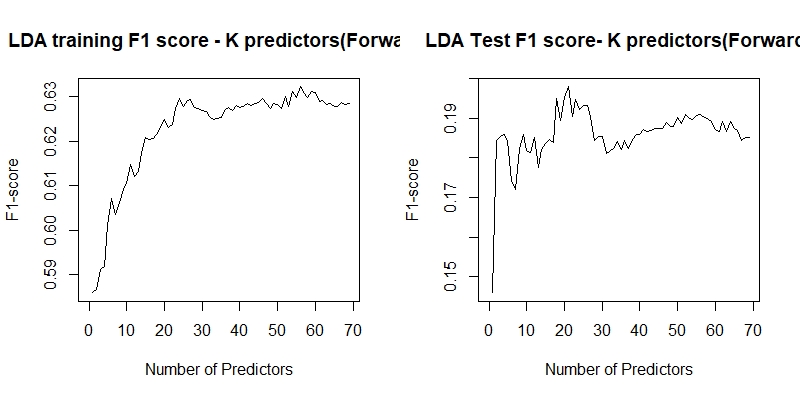
### Figure 4. Logistic Regression Training/Testing Area Under Curve

Using F1 score we can more easily see the problem with our original method. On the test set we see that despite the model gaining accuracy, it fails to reach anything greater than a 20% accuracy. Which when used in practice is a terrible idea. However Logistic Regression yields the best performance throughout our whole project.

We've chosen AUC as another metric to verify our model's performance since it performs better when faced with an imbalanced class set and also allows us to validate our model regardless of the threshold. The AUC for our model hovers slightly over the 0.5 line, which indicates that it has little ability to verify between positive and negative classes.

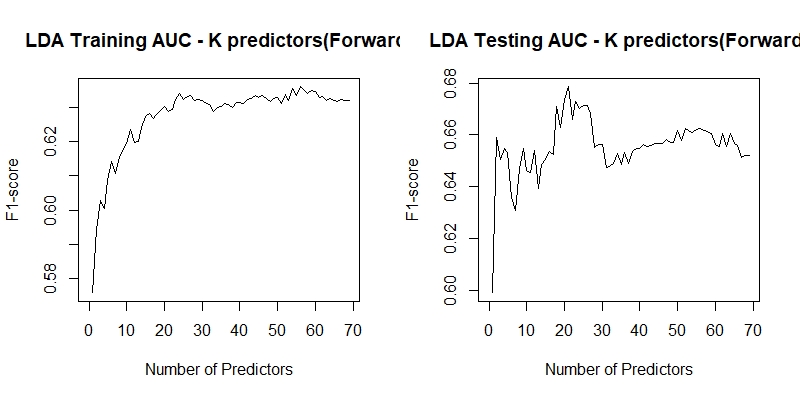


### Figure 5. Linear Discriminant Analysis Training/Testing Errors

After implementing Logistic Regression, we try to use LDA ( Linear Discriminant Analysis) to validate the performance of our model. We see that the RMSE lies around 0.3 percent, which again looks pretty nice but not great for a model. Again with the knowledge of an imbalanced dataset, we try using other methods to resolve this issue. 

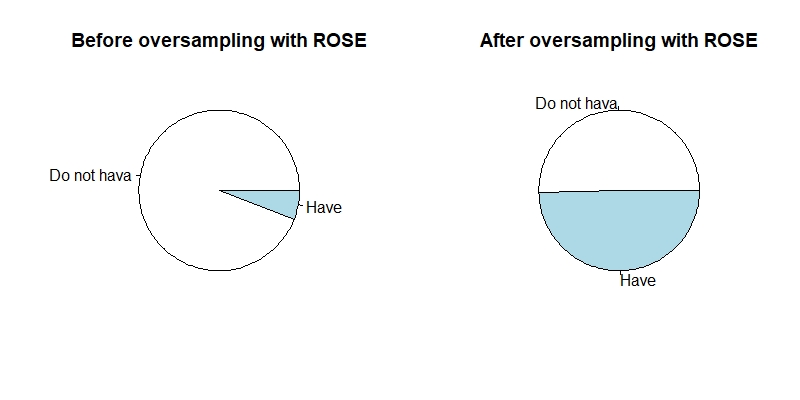
### Figure 6. Linear Discriminant Analysis Training/Testing F1 Score

The performance whilst training looks pretty good at first, lying at around 60 percent. However when we test the model on unseen dataset(our test set), the accuracy drops significantly to 17% which is on the verge of unusable.



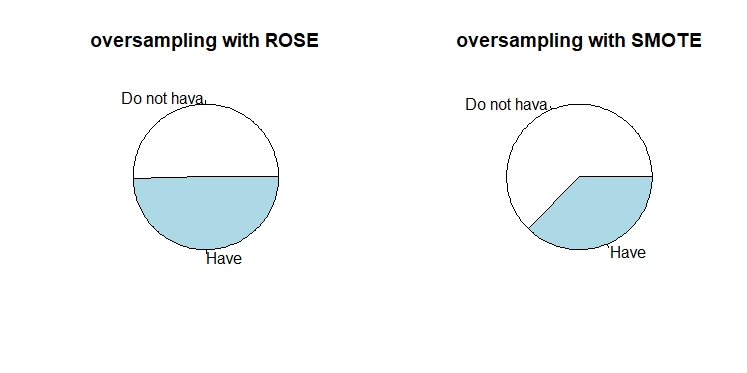
### Figure 7. Linear Discriminant Analysis Training/Testing F1 Area Under Curve

Using AUC for our LDA model apparantly yields the same result as the one we have in Logistic Regression, we at this point have come to a conclusion that our model wasn't the main issue and the imbalanced dataset was extremely hard to deal with. We next proceed to use CART methods in addition to SMOTE and ROSE to oversample/generate synthetic data for our model to see how well tree based methods work in this case.



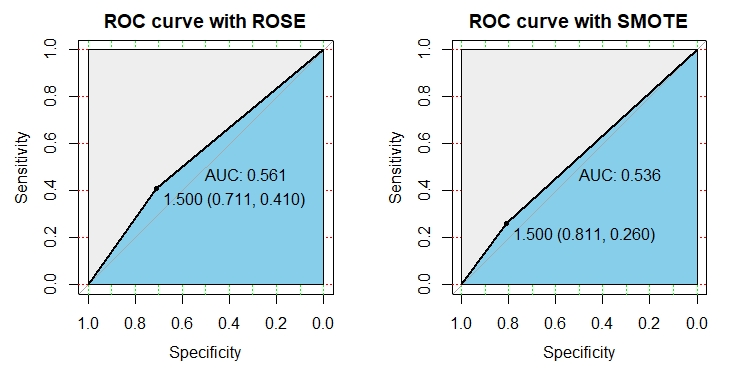
### Figure 8. Data oversampling with ROSE(see Figure 1. for SMOTE)

Using the ROSE method of generating synthetic data similar to the minority class we approach a nearly 1 to 1 ratio in the training data set, but later tested on the test set.



### Figure 9. Data Oversampling comparison with ROSE and SMOTE

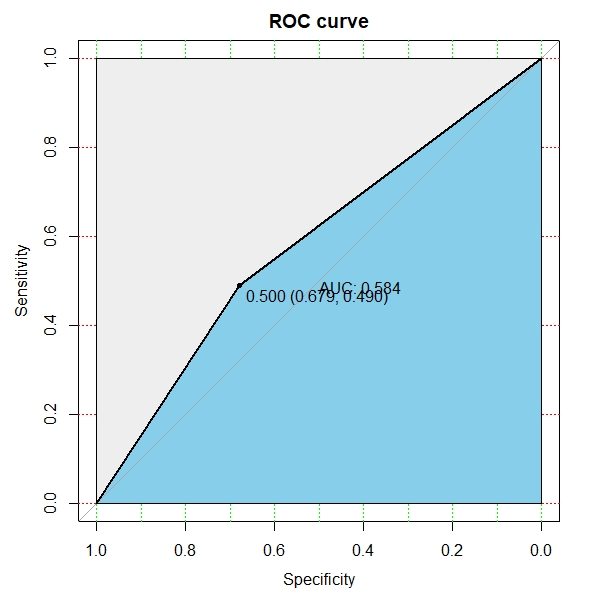
A quick comparison of ROSE and SMOTE's effect on the training data. We've tested with different variations of datasets and we settled with an almost 1:1 ratio for ROSE and a 3:2 ratio for the SMOTE method.



### Figure 10. Comparison of ROC curve using SMOTE and ROSE

Using ROSE and SMOTE we see that the AUC did not have any significant difference, or we can even say the performance has decreased and did not perform as well as Logistic Regression or LDA without SMOTE or ROSE.

We then proceed to use XGBoost as another method to test whether we can save this imbalanced dataset.



### Figure 11. ROC Curve of XGBoost

Checking if XGboost made any actual results, we see that although the results weren't as tragic as the CART methods, the AUC score did not improve much, only up till 60% which means the model can still barely differentiate the positive classes to the negative ones. With an F1 score of 0.67 for the positive case and 0.49 for the negative cases.

## Conclusion

|  |  |  |
| --- | --- | --- |
| Method | F1-score | AUC |
| LDA(p = 21, with ROSE) | 0.1979 | 0.679 |
| LR(p = 21, with ROSE) | 0.1992 | 0.682 |
| LR(p = 42, without ROSE) | 0.0196 | 0.504 |
| CART(with ROSE) | 0.1355 | 0.561 |
| CART(with SMOTE) | 0.1212 | 0.536 |
| XGBoost | 0.1473 | 0.584 |

## Table 1: Final Results

The table shown above summarizes the result from our 5 primary methods. Linear Discriminant Analysis(LDA) and Logistic Regression(LR) are the main methods that worked for us in Milestone1 without oversampling.

The F1-scores of LDA and LR improved by 0.18 with oversampling compared to the original results without oversampling. The additional methods (CART, XGBoost) that we focused on in this milestone have a slight decrease in F1-score despite tree-based methods traditionally having better results with imbalanced classes. The AUCs of CART and XGBoost are around 0.5, which means they do not have discrimination capacity to distinguish two classes even after applying oversampling methods mentioned above in this report. The AUCs of LDA and LR are close to 0.7.Although it is significantly higher, this is by no standards a good enough model to describe this set of imbalanced data.

To summarize, during the start of Milestone 2 we thought the problem was the imbalanced dataset. We've made several attempts to fix the results and produce a better model. However, there is no significant improvement after applying various oversampling methods, CART and XGBoost.