Contents

[Why sampling? 1](#_Toc88994346)

[Sampling Problems 1](#_Toc88994347)

Sampling

Sampling takes a subset of historical data such as past applicants and uses that to build a credit risk model. The first obvious question that comes to mind concerns the need for sampling.

# Why sampling?

It is true that with the availability of high performance computing facilities such as grid and cloud computing one can also try to directly analyze the full data set. However, a key requirement for a good sample is it should be representative for the future entities on which the credit risk model will be run.

## timing

The timing becomes important since data of today is more like data of tomorrow than data of yesterday. Choosing the optimal time window of the sample involves a trade-off between lots of data and hence a more robust credit risk model and recent data which may be more representative.

## average business period

The sample should also be taken from an average business period to get an accurate picture of the target population. As an example it makes no sense to take a sample of sales from Black Friday to analytically understand what drives average sales. Note that most tools provide easy out-of-the-box support for various types of sampling methods.

## sufficient to stabilize the default rate

It is also important to make sure the performance window is long enough to stabilize the default rate. In application scoring one commonly adopts an eighteen-month performance window.

# Sampling Problems

Two popular examples of sampling problems are the following: in application scoring reject inference, and behavioral scoring: a seasonality depending upon the choice of the observation point.

## reject inference

Remember when we discussed sampling previously, we said that when you think about a sample the first thing to think about is the future population on which you're going to apply your credit risk model so let's apply that idea in application scoring. In application scoring you want to score everybody who knocks on the door of your bank and applies for credit, let's say a mortgage. This population can be referred to as the through the door population so in other words you need a sample of the historical through the door population because the assumption you make is that the future through the door population is similar to the historical through the door population. However when looking at this historical through the door population; you will find two segments the ones that got accepted in the past and the ones that got rejected in the past with the old credit scoring method.

For the accepts you can see which ones turned out to be good payers and which ones turned out to be bad payers. For the rejects you don't know since they were never given credit. You can only assume or hope that many of them will have turned out to be bad payers but you will never be absolutely sure. Hence if you would only consider the accepts as your development sample to build an application scorecard then this sample will obviously be biased since it contains proportionally too many goods and is thus not representative for the future through the door population. To restate the reject inference problem the good bad target information is only available for the past accepts and not for the past rejects. This creates a reject bias when we only build models on the past accepts. We need to extend the development sample such that it becomes representative for the future through the door population.

Reject inference is now the process whereby the performance of previously rejected applicants is analyzed to estimate their behavior and add them to the development sample for model estimation. Note that if the past acceptance policy would have been random then both accepts and rejects would have been random samples and there would have been no reject inference problem. However, this is a very unrealistic assumption since many banks already used sophisticated credit scoring models for quite some time now hereby creating a strong reject inference problem when developing new scorecards.

Various methods for reject inference have been suggested.

1. A first method to do reject inference is to classify all rejects as bad payers. Obviously this method will reinforce the credit scoring policy and prejudices of the past. It is also a conservative approach since it is plausible to assume that not all rejects would have turned out to be bad payers. Hence, by classifying all rejects as bads the bad rate in a sample will be too high. A softer version can also be adopted whereby only a subsample of the rejects is classified as bads based upon expert knowledge.
2. Another way to get more information about the rejects is via the credit bureau. Remember that a credit bureau or credit reference agency is an institution that gathers information from various financial institutions about the delinquency behavior of their customers. So what you could do is you could ask the bureau if some of your past rejects got credit elsewhere. You can also ask for their performance at the other financial institutions. Actually you can ask for two pieces of information from the credit bureau. You can give the credit bureau a sample of your past rejects and ask them to classify them as good or bad or, if privacy regulations would not allow you to do so, you could also provide the credit bureau with a sample of your past rejects and ask them about the bad rate in that sample.
3. You can also analyze the rejects with the analytical model you built on the accepts as we discuss later.

## Withdrawal inference

Remember the reject inference problem is essentially a sampling problem because no information about the target good/bad class is available for the previously rejected customers. However when you think a bit more closely about the historical through-the-door population, you will see that you also have withdrawals in there.

These are customers who decided themselves to not take up the offer because they found a better offer elsewhere. In other words these are the shoppers. Also for these withdrawals we may not know the true good/bad class and procedures for withdrawal inference should be adopted. To the best of my knowledge withdrawal inference is typically left unconsidered by the majority of banks when developing application scorecards. One easy way to do withdrawal inference could again be via the credit bureau whereby a sample of past withdrawals is given to the credit bureau so as to obtain information about their good bad status at other financial institutions.

### low number of bads

### undersampling & oversampling

Undersampling and oversampling are also two popular sampling methods to deal with a low number of defaulters or bads. Both are applied on the training or model development set and not on a test set. The idea of undersampling is to remove good customers from the training set. The idea of oversampling is to replicate bad customers in the training set. The aim of both these approaches is to make the target distribution less skew, say from 99% goods and 1% bads to 90 versus 10%, 80 versus 20%, or even 50 versus 50% which represents a balanced sample. Trial-and-error can be used to determine the ideal distribution. The latter can be found when the area on the ROC curve is optimal as we discuss later. Note that this also depends on the analytical technique such as logistic regression, decision trees etc.

Here you can see undersampling being illustrated. Remember both under and oversampling are only conducted on the training set. The original training set has three bads and seven goods. By using undersampling customers 5, 7 and 10 are being thrown out so yet we have a nicely balanced training set with 3 bads and 3 good. Here you can see oversampling being illustrated. We started from the same underlying training set. However now we replicated customers 1, 4 and 9 such that we have a more balanced training set now with 6 bads and 7 goods. Again note that oversampling should only be considered on the training set which is used for model development.

### smote

The synthetic minority oversampling technique or SMOTE is another interesting approach to deal with skewed class distributions. It oversamples the minority class by creating synthetic examples.

1. In step one of SMOTE, for each minority class observation the k nearest neighbors are determined in Euclidean sense for example. Let's say we set k equal to one and look at the nearest neighbor.
2. Step two then generates the synthetic examples as follows. Take the difference between the variables of the current minority sample and those of its nearest neighbor. Multiply this difference with a random number between zero and one and add it to the sample. The synthetic examples are thus added to the data set hereby increasing the frequency of the minority class.

To further level the distribution this procedure can also be easily combined with undersampling the majority class. Throughout our research we have found that this SMOTE method works really good when dealing with skewed class distributions. You can see the creation of synthetic examples illustrated here. We have the attributes amount and ratio of both Tim and Bart. First, SMOTE chooses a random number between 0 & 1 for example 0.6. A synthetic sample is created between Tim and Bart by taking the difference between the variables of Tim and Bart and multiplying these differences with 0.6. The result is a synthetic defaulter with an amount equal to 2782 and a ratio of 0.88.