DATA REPRESENTATIVENESS PROBLEM IN CREDIT SCORING

Josef Ditrich

Abstract

When building models, it is common to split the whole dataset into a development and a validation sample. In some cases, using random sampling instead of stratified sampling can lead to loss of representativeness of final samples. In such cases, a model built on these data gives different or unexpected results when its performance is measured on the validation sample. In the business area, a lack of representativeness can cause interpretative problems and can have a huge financial impact when a biased model is involved in the credit granting process. The aim of this paper is to examine and understand why representativeness should be checked before the start of modelling. The paper deals with methods of identification of selection bias in time. It recommends using three tests as a common part of the data preparation process.

Key words: credit scoring, credit risk models, selection bias, random sampling, stratified sampling, data splitting
JEL classification: C18, C80, C83

1. Introduction

In statistical modelling, it is common for the database to be split into two samples – the development sample (DEV) and the validation sample (VAL). The development sample is used to develop the model (learning and estimating parameters of the model), while the validation sample is used to evaluate the model and for final model selection. In a later phase of model development, a third type of sample – the testing sample(s) – can be used for assessing the predictive performance of the model [Borovicka et al., 2012]. If the same dataset would be used for the development, validation and calibration, the estimation of the predictive ability of the model would be overly optimistic.

In an ideal situation, two (or more) independent datasets are collected. However, in a situation where only one dataset is available and there is no opportunity to collect new data, it is necessary to split the data file. According to Snee [1977], data splitting is the most effective method of model validation when it is impossible to collect new data to examine the model. It is very important to create both (DEV and VAL) samples in such a way that represents the total population as they can cause a lot of problems due to bias. This requirement is absolutely natural, since the model reflects the specifics of the

dataset used for its development. In order to make sure that the sample is representative, it is important to consider carefully how the sample was collected. If a sample is chosen for the sake of convenience alone, it becomes difficult to interpret the final model with confidence [Geoff, Everitt, 2001].

Bias refers to the tendency for selected samples to contrast with the corresponding population in some methodical manner. Bias can arise if the sample was chosen wrongly [Peck et al., 2012]. When sampling, the most common types of bias that may occur are selection bias, response or measurement bias, and nonresponsive bias.

In most applications, simple random sampling is used. Nevertheless, there are several sophisticated statistical sampling methods more suitable for various types of datasets. The purpose of this paper is to show what would happen if both the development and validation datasets were created poorly in such a way that they were not representative of the population. To demonstrate the consequences of the impacts on the performance of the scorecards, two different and most common data splitting methods were employed.

The rest of this paper is organized as follows. The next section presents a brief overview of various sampling methods. Section 3 explains the methodology used in performed tests. Section 4 describes the data used for impact illustration. Section 5 contains a case study and discusses analysis results. The final section presents conclusions.

2. Data splitting

In many fields, representative large independent samples can be used for training and validating (and testing) of models and can be obtained simply by partitioning one large dataset (holdout method). However, in other fields, only datasets limited in size are available as measurements are expensive or work-intensive. To solve the dilemma of partitioning a small pool of data into independent data subsets, re-sampling procedures can be used (repeated holdout method). It is believed that the more data, the better model performance. However, some recently published studies show that this is not necessarily true [Meng, Xie, 2013; Faraway, 2014].

Stone [1974] may be considered a pioneer of data splitting. Since then, many data splitting methods have been designed. Their quality and complexity differ, and there is no single method which is, in general, viewed as superior. Their choice mostly depends on the purposes of the analysis. Sampling methods can be divided according to their principles, goals, and overall complexity [Reitermanová, 2010]. Data splitting algorithms and also their comparison can be found in many studies [e.g., Molinaro et al., 2005; Snee, 1977]. Data splitting is easy to implement and thus presents an attractive alternative to complex methods of adjusting for the effect of model selection on inference [Faraway, 1998].

Simple random sampling is the most common holdout method. It is efficient and easily feasible. Samples are selected randomly with uniform distribution, i.e., each observation has equal probability of selection. This quite simple method leads to low bias of model performance. However, in cases where the dataset is not uniformly distributed or the number of selected cases is much lower compared to the original database, simple random sampling can lead to subsets that do not cover the data properly (e.g., one or more classes might be missing) and therefore the estimate of the model error will have a high variance [Lohr, 1999].

Stratified sampling is probability sampling and stands on the idea to explore the internal structure and distribution of a dataset and to divide it into (relatively) homogeneous

non-overlapping groups called strata (or clusters). The observations are then selected from each stratum proportionally to the appropriate probability. It ensures that each class is represented with the same frequency into subsets. The important question is how to select observations from each cluster. There are two most common principles: to select one sample from each cluster [Bowden et al., 2002] or samples from each cluster are selected with a uniform probability [May et al., 2010]. The second approach is often referred to as stratified random sampling.

Systematic sampling can be used in the case of (naturally) ordered datasets. The most common form of systematic sampling is the equal-probability method. From the ordered dataset (e.g., a time series), a starting observation is randomly chosen and then each i^{th} observation is selected [Elsayir, 2014]. The sampling interval (skip) i is calculated as the ratio of sample size to population size. Systematic sampling is a very efficient method and it is easy to implement. However, in many cases it is very difficult to find appropriate ordering. For disordered data, the results of systematic sampling are comparable to those of simple random sampling and it therefore suffers the same problems. Also, its sensitivity to periodicities in the dataset is one of the disadvantages of the method.

Cross-validation ranks among the most popular re-sampling methods. For k-fold cross-validation, the original dataset is partitioned into k equal-sized parts (folds). The first fold is used for evaluation purposes; the rest $(k-1)$ of the folds are used for model learning. In the next step, the second fold is used for evaluation and the rest are used for learning. This procedure is repeated k-times and the results are averaged (Picard and Cook, 1984). There are no clear rules on how many folds should be used for the cross-validation. In practice, the set $k=10$ is often sufficient.

A special variant of cross-validation is called leave-one-out cross-validation (full cross-validation, jack-knife). It assumes $k=n$, where n is the size of the original dataset. This setting gives nearly unbiased estimates (lower root mean square errors of predictions) of the model performance but usually with large variability. This deficiency is known as asymptotic inconsistency [Shao, 1993].

The main principle of the bootstrap method, first introduced in 1996 [Tibshirani, Efron, 1996], is to get B bootstrap samples by uniform sampling with replacement from the original dataset with the size n. On each bootstrap, sample parameters of a model are estimated while the estimation of prediction performance is carried out on the original dataset. Bootstrapping is not affected by asymptotic inconsistency and might be the best way of estimating error for very small datasets whereby the complete procedure can be repeated arbitrarily. For more information, see for instance Kohavi [1995] or Andrews [2000].

3. Methodology

In this section, we present three quick analyses that can be used for checking representativeness between two created subsamples when building a scoring model. Further in the case study, the proposed tests are illustrated on credit scoring model development but they can be used in other areas as well.

It is possible to check for different variables of the database used for the computation of the final score for whether the repartition of the modalities is significantly different between the development and validation samples. This is called demand stability.

Risk stability examines whether the event rate between corresponding classes for a variable is appropriate between both samples.

A gap table can also be constructed. Rows represent categories in the case of explanatory variables or score deciles in the case of scorecard output examination. For each class of the analysed variable, columns in the table contain the following information:

- points of each category (for explanatory variables only),
- average total score,
- numbers of observations and column percentage,
- numbers of observed and predicted events and their differences,
- observed and predicted event rates and their difference,
- p-value of one-tailed test (H_0: $event\_rate_{predicted} \leq event\_rate_{observed}$), and
- 95% two-sided confidence interval for $event\_rate_{predicted}$.

4. Data description

For the illustration of impacts of data splitting in the case study described in the next section, a database from the credit risk area was used. It contains 20 behavioural characteristics (explanatory variables) of clients' credit behaviour in a bank; all of them are categorical. The definition of the two-valued explained variable was used as follows: a client is marked as bad if he has reached 3 or more instalments past due. Otherwise a client is marked as good. The goal is to build a behavioural scoring model. For this purpose, binary logistic regression was employed with stepwise selection of explanatory variables. The output of the scoring model is probability of default, but for better orientation, the values were transformed to a credit score where the higher score means the better client.

The approach to the modelling process is to use the holdout method. The database will be split such that 70% of the data will belong in the development dataset and 30% in the validation dataset. The data splitting will be carried out in two main ways:

1. Using a stratified random sampling that maintains the proportion of good/bad.
2. Using a simple random sampling that does not maintain the good/bad proportion.

For these purposes, the SAS 9.1.3 procedure PROC SURVEYSELECT (with explained variable in strata option) can be used to split the database.

Taking the example constructed from a real database, let's have a look at the bad rate distribution (Table 1). The database chosen is large enough so that it gives a good platform for examining the impact of stratified random sampling and simple random sampling on the predictability of the chosen samples, both the validation and development databases.

Table 1 | Overview of analysed database

Number of observations	16,646
Number of bads	1,229
Bad rate	7.38%

Source: calculated by the author

5. Case study

5.1 Data splitting

The SAS procedure with a strata option is used to designate variables defining a dataset or strata or nested sets in a case control study. The results obtained with stratified random sampling are shown in Table 2.

Table 2 | Split of database using stratified random sampling

Sample	Number of observations	Number of bads	Bad rate
DEV	11,653	860	7.38 %
VAL	4,993	369	7.39 %
Total	16,646	1,229	7.38 %

Source: calculated by the author

In this type of data splitting, the sample size is split into two; however, the good and bad observations are incorporated into the modelling process at the same ratio as is in the original database. We, therefore, always have 369 bad contracts that can be used to validate the model.

In simple random sampling, it is assumed that the sample selected is absolutely random and that no biases occur in the data. However, the failure to identify a serious bias in the sample can result in inaccurate test statistics and standard errors. Looking at the database previously selected, we find that if we do not use the stratified random sampling, different values for validation arise. To be relevant, in this case the database was tested (split) 1,000 times and the confidence interval was calculated. The results are as follows (Table 3):

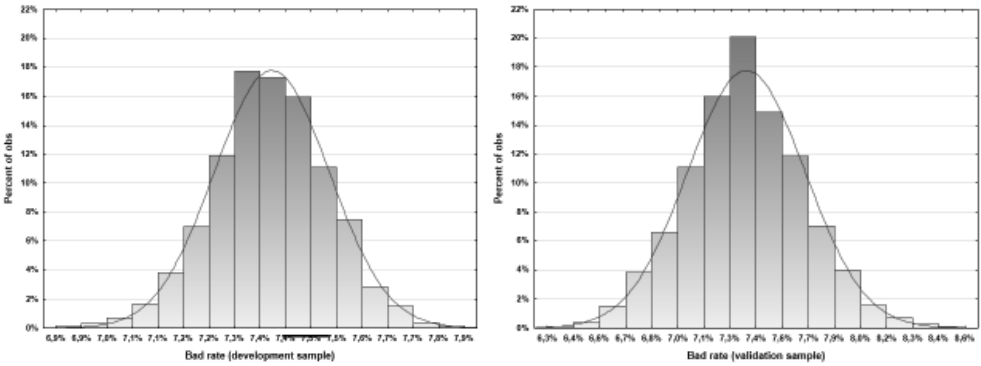
Table 3 | Split of database using simple random sampling

Sample	Number of observations	Number of bads	Bad rate
DEV	11,653	833 to 887	7.15 % to 7.58 %
VAL	4,993	342 to 396	6.85 % to 7.93 %
Total	16,646	1,229	7.38 %

Source: calculated by the author

Looking at Table 3, we find that in extreme cases 54 bads could be lost to validate the model. This proportion (14.6 %) is very high as the percentage of the total number of bads (7.38 %) is quite low and the quantity of the validation database is quite small (30 % of the total sample size). As such, the choice to keep or remove the 54 bads from the validation database has significant consequences. Using a strata option in the SAS procedure is thus important in order to avoid this problem in the case of a low number of bads. By considering the bad proportion of the database, the dataset so selected is more representative of the population tested.

Figure 1 | Observed bad rate on DEV (left) and VAL (right) with simple random sampling



95% conf. interval for bad rate is (7.15%; 7.58%)

95% conf. interval for bad rate is (6.85%; 7.93%)

Source: calculated by the author

From the graphs in Figure 1, we find that the development database can have between 833 (7.15 %) and 887 (7.58 %) bads. On the other hand, the validation databases can have between 342 (6.85%) and 396 (7.93 %) observations marked as bad. In both these cases, there is an interval of 54 contracts between the lower and upper confidence levels. The difference witnessed is not that relevant with respect to the development database as a deviation of + or – 6.3 % of bads among the 860 bads is expected. However, the impact is very important when considering the validation database as a deviation of + or – 14.6 % of bads among the 369 bads is expected.

5.2 Consequences of bias on explained variable

In order to verify the details gained from the above sections, we will carry out two other impact analyses to see the impacts identified in the gap analysis. In the first analysis, we split the whole sample such that both the development and validation samples are still in a 70:30 ratio and have purposely highly different proportions of bads in both samples. On analysis, the repartition is as follows:

Table 4 | Purposeful splitting of database into two non-representative samples

Sample	Number of observations	Number of bads	Bad rate
DEV	11,653	902	7.74 %
VAL	4,993	327	6.55 %
Total	16,646	1,229	7.38 %

Source: calculated by the author

5.2.1 Demand stability

When we look at the stability of demand, we find that it experiences no impact whatsoever between both the DEV and the VAL. For example, the explanatory variable "Client duration", which represents the length of the relationship between the client and the institution, is as follows (Table 5):

Table 5 | Demand stability of explanatory variable

Client duration (profile in %)	DEV	VAL	TOTAL
0 – 12 months	6.39 %	5.71 %	6.19 %
13 – 30 months	17.69 %	17.14 %	17.50 %
31 – 96 months	45.60 %	46.83 %	46.00 %
97 + months	30.32 %	30.32 %	30.30 %
Total	100.00 %	100.00 %	100.00 %

Source: calculated by the author

The values obtained in this analysis are similar to the initial values (column Total) above, showing that the bias does not affect the demand stability of the explanatory variable.

5.2.2 Risk stability

However, when we look at the risk stability (Table 6), we find out that the risk between the DEV and the VAL is not similar. It gives us a first warning. This is expected as the risk should be lower for the validation sample compared to the development sample based on information displayed in Table 4.

Table 6 | Risk stability of explanatory variable

Client duration (bad rate in %)	DEV	VAL	TOTAL
0 – 12 months	25.91 %	19.30 %	24.08 %
13 – 30 months	15.19 %	13.20 %	14.60 %
31 – 96 months	5.83 %	5.09 %	5.61 %
97 + months	2.43 %	2.65 %	2.50 %
Total bad rate	7.74 %	6.55 %	7.38 %

Source: calculated by the author

In the initial sample, the total bad rate was 7.38 %. For the development sample, it was 7.74%, while that for the validation sample was 6.55 %. Anyway, the values received from this test have the same trend as the original values, and this indicates that the risk between the two samples is stable.

5.2.3 Gap analysis

The gap analysis (Tables 7–10) reveals that the gap present is inconsequential and does not have a large impact on the model. In this test analysis, a bias induced on the explained variables poses no problem. However, a problem could arise when a low number of bads exists such that in the first test carried previously. Thus, keeping the same good/bad proportion in the development and validation samples proves advantageous.

Table 7 | Gap analysis of score on DEV

SCORE (development sample)	Category points	Avg. total score	Observations		Number of bads			Bad rate			p-value	95 % conf. interval	
			N	%	observed	predicted	gap	observed	predicted	gap		min	max
1st decile (0 – 116)	–	83.6	1,256	10.8 %	303.0	309.7	+ 6.7	24.1%	24.7 %	+ 0.5 %	0.3378	23.8 %	25.5 %
2nd decile (116 – 149)	–	143.9	1,124	9.6 %	223.0	221.4	- 1.6	19.8%	19.7 %	- 0.1 %	0.4360	19.5 %	19.9 %
3rd decile (149 – 194)	–	175.2	1,137	9.8 %	84.0	85.8	+ 1.8	7.4%	7.5 %	+ 0.2 %	0.0551	7.3 %	7.8 %
4th decile (194 – 226)	–	213.9	1,251	10.7 %	92.0	83.9	- 8.1	7.4%	6.7 %	- 0.7 %	0.0809	6.4 %	7.0 %
5th decile (226 – 251)	–	236.8	1,073	9.2 %	51.0	62.9	+ 11.9	4.8%	5.9 %	+ 1.1 %	0.2936	5.6 %	6.1 %
6th decile (251 – 265)	–	261.6	1,244	10.7 %	48.0	42.4	- 5.7	3.9%	3.4 %	- 0.5 %	0.4121	3.3 %	3.5 %
7th decile (265 – 287)	–	276.3	1,419	12.2 %	21.0	14.9	- 6.1	1.5%	1.1 %	- 0.4 %	0.0719	1.0 %	1.1 %
8th decile (287 – 297)	–	294.4	983	8.4 %	47.0	43.2	- 3.8	4.8%	4.4 %	- 0.4 %	0.3442	4.3 %	4.5 %
9th decile (297 – 322)	–	310.9	1,162	10.0 %	17.0	18.6	+ 1.6	1.5%	1.6 %	+ 0.1 %	0.1083	1.5 %	1.7 %
10th decile (322+)	–	337.0	1,004	8.6 %	16.0	19.3	+ 3.3	1.6%	1.9 %	+ 0.3 %	0.0553	1.9 %	2.0 %
Total	–	231.2	11,653	100.0 %	902.0	902.0	0.0	7.7%	7.7 %	0	T	7.6 %	7.9 %

Source: calculated by the author

Table 8 | Gap analysis of score on VAL

SCORE (validation sample)	Category points	Avg. total score	Observations		Number of bads			Bad rate			p-value	95 % conf. interval	
			N	%	observed	predicted	gap	observed	predicted	gap		min	max
1st decile (0 – 116)	–	85.1	514	10.3 %	98.0	105.6	+ 7.6	19.1 %	20.5 %	+ 1.5 %	0.2097	19.4 %	21.7 %
2nd decile (116 – 153)	–	145.0	541	10.8 %	80.0	81.4	+ 1.4	14.8 %	15.0 %	+ 0.3 %	0.4351	14.6 %	15.5 %
3rd decile (153 – 197)	–	180.2	489	9.8 %	45.0	36.8	- 8.2	9.2 %	7.5 %	- 1.7 %	0.1020	7.2 %	7.9 %
4th decile (197 – 226)	–	215.8	455	9.1 %	25.0	24.7	- 0.3	5.5 %	5.4 %	- 0.1 %	0.4745	5.0 %	5.8 %
5th decile (226 – 255)	–	238.5	543	10.9 %	23.0	27.2	+ 4.2	4.2 %	5.0 %	+ 0.8 %	0.1897	4.7 %	5.3 %
6th decile (255 – 269)	–	265.1	744	14.9 %	16.0	15.4	- 0.6	2.2 %	2.1 %	- 0.1 %	0.4422	1.9 %	2.2 %
7th decile (269 – 287)	–	283.3	338	6.8 %	11.0	5.2	- 5.8	3.3 %	1.5 %	- 1.7	0.0581	1.4 %	1.7 %
8th decile (287 – 297)	–	294.2	463	9.3 %	14.0	17.2	+ 3.2	3.0 %	3.7 %	+ 0.7 %	0.1949	3.6 %	3.8 %
9th decile (297 – 319)	–	308.1	422	8.5 %	3.0	3.7	+ 0.7	0.7 %	0.9 %	+ 0.2 %	0.3382	0.9 %	0.9 %
10th decile (319+)	–	334.7	484	9.7 %	12.0	9.9	- 2.1	2.5 %	2.0 %	- 0.4 %	0.2678	2.0 %	2.1 %
Total	–	232.2	4,993	100.0 %	327.0	327.0	0.0	6.5 %	6.5 %	0	T	6.3 %	6.8 %

Source: calculated by the author

Table 9 | Gap analysis of explanatory variable on DEV

CLIENT DURATION (development sample)	Category points	Avg. Total score	Observations		Number of bads			Bad rate			p-value	95 % conf. interval	
			N	%	observed	predicted	gap	observed	predicted	gap		min	max
0–12 months	0.0	100.5	745	6.4 %	193.0	193.0	+ 0.0	25.9 %	25.9 %	+ 0.0 %	0.4985	24.9 %	27.0 %
13–30 months	52.0	167.7	2,061	17.7 %	313.0	313.0	+ 0.0	15.2 %	15.2 %	+ 0.0 %	0.4967	14.7 %	15.7 %
31–96 months	149.0	272.7	5,314	45.6 %	310.0	310.0	+ 0.0	5.8 %	5.8 %	+ 0.0 %	0.4856	5.7 %	6.0 %
97 + months	110.0	233.4	3,533	30.3 %	86.0	86.0	+ 0.0	2.4 %	2.4 %	+ 0.0 %	0.4921	2.3 %	2.5 %
Total	110.5	231.2	11,653	100.0 %	902.0	902.0	0.0	7.7 %	7.7 %	0	T	7.6 %	7.9 %

Source: calculated by the author

Table 10 | Gap analysis of explanatory variable on VAL

CLIENT DURATION (validation sample)	Category points	Avg. total score	Observations		Number of bads			Bad rate			p-value	95 % conf. interval	
			N	%	observed	predicted	gap	observed	predicted	gap		min	max
1: 0–12 months	0.0	103.6	285	5.7 %	55.0	61.7	+ 6.7	19.3 %	21.7 %	+ 2.4 %	0.1686	20.2 %	23.2 %
2: 13–30 months	52.0	168.2	856	17.1 %	113.0	112.7	- 0.3	13.2 %	13.2 %	- 0.0 %	0.4884	12.5 %	13.8 %
3: 31–96 months	149.0	271.5	2,338	46.8 %	119.0	119.5	+ 0.5	5.1 %	5.1 %	+ 0.0 %	0.4807	4.9 %	5.3 %
4: 97 + months	110.0	231.7	1,514	30.3 %	40.0	33.0	- 7.0	2.6 %	2.2 %	- 0.5 %	0.1349	2.1 %	2.3 %
Total	112.0	232.2	4,993	100.0 %	327.0	327.0	0.0	6.5 %	6.5 %	0	T	6.3 %	6.8 %

Source: calculated by the author

5.3 Consequences of bias on explanatory variables

In this test analysis, we have a database with a bias purposely placed on the explanatory variable "Client duration" used for the calculation of the final score. Note that the repartition is the same as in the previous subsection (Table 4).

5.3.1 Demand stability

In this case, the stability of the demand has not been affected (Table 11). This analysis reveals that there is no change in the demand stability when compared to initial values.

Table 11 | Demand stability of explanatory variable

Client duration (profile in %)	DEV	VAL	TOTAL
0 – 12 months	6.32 %	5.87 %	6.19 %
13 – 30 months	17.54 %	17.48 %	17.52 %
31 – 96 months	46.07 %	45.74 %	45.97 %
97 + months	30.07 %	30.90 %	30.32 %
Total	100.00%	100.00 %	100.00 %

Source: calculated by the author

5.3.2 Risk stability

The risk between the DEV and the VAL is very dissimilar, which again raises a first alert on the problem of representativeness. From the risk stability (Table 12), we find that a bias on the explanatory variable used for the calculation of the final score causes the whole sample to lose confidence in the representation of the population.

Table 12 | Risk stability of explanatory variable

Client duration (bad rate in %)	DEV	VAL	TOTAL
0 – 12 months	23.88 %	24.57 %	24.08 %
13 – 30 months	15.61 %	12.26 %	14.60 %
31 – 96 months	5.87 %	4.99 %	5.61 %
97 + months	2.63 %	2.20 %	2.50 %
Total bad rate	7.74 %	6.55 %	7.38 %

Source: calculated by the author

5.3.3 Gap analysis

Significant gaps between the observed and the predicted values are observed in the gap analysis. The highlighted values in Table 14 indicate a poor fit of the model, implying that a higher rate of misclassification should be expected. Once again, it alerts the statistician to a problem of either instability or correlation.

Table 13 | Gap analysis of score on DEV

SCORE (development sample)	Category points	Avg. total score	Observations		Number of bads			Bad rate			p-value	95 % conf. interval	
			N	%	observed	predicted	gap	observed	predicted	gap		min	max
1st decile (0 – 116)	–	83.9	1,269	10.9 %	290.0	299.0	+ 9.0	22.9 %	23.6 %	+ 0.7 %	0.2839	22.8 %	24.3 %
2nd decile (116 – 149)	–	144.0	1,130	9.7 %	218.0	216.3	- 1.8	19.3 %	19.1 %	- 0.2 %	0.3712	19.0 %	19.3 %
3rd decile (149 – 194)	–	174.8	1,156	9.9 %	92.0	89.5	- 2.5	8.0 %	7.7 %	- 0.2 %	0.4406	7.5 %	8.0 %
4th decile (194 – 226)	–	213.8	1,220	10.5 %	92.0	86.5	- 5.5	7.5 %	7.1 %	- 0.4 %	0.0908	6.8 %	7.4 %
5th decile (226 – 251)	–	236.7	1,093	9.4 %	58.0	64.6	+ 6.6	5.3 %	5.9 %	+ 0.6 %	0.2531	5.7 %	6.1 %
6th decile (251 – 265)	–	261.8	1,195	10.3 %	45.0	41.9	- 3.1	3.8 %	3.5 %	- 0.3 %	0.2429	3.4 %	3.6 %
7th decile (265 – 287)	–	276.4	1,434	12.3 %	26.0	16.9	- 9.1	1.8 %	1.2 %	- 0.6 %	0.0198	1.1 %	1.3 %
8th decile (287 – 297)	–	294.3	994	8.5 %	41.0	45.0	+ 4.0	4.1 %	4.5 %	+ 0.4 %	0.2390	4.4 %	4.6 %
9th decile (297 – 322)	–	310.8	1,179	10.1 %	21.0	21.7	+ 0.7	1.8 %	1.8 %	+ 0.1 %	0.3445	1.8 %	1.9 %
10th decile (322+)	–	336.8	983	8.4 %	19.0	20.7	+ 1.7	1.9 %	2.1 %	+ 0.2 %	0.1690	2.1 %	2.2 %
Total	–	230.9	11,653	100.0 %	902.0	902.0	0.0	7.7 %	7.7 %	0	T	7.6 %	7.9 %

Source: calculated by the author

Table 14 | Gap analysis of score on VAL

SCORE (validation sample)	Category points	Avg. total score	Observations		Number of bads			Bad rate			p-value	95 % conf. interval	
			N	%	observed	predicted	gap	observed	predicted	gap		min	max
1st decile (0 – 116)	–	84.2	501	10.0 %	111.0	102.0	- 9.1	22.2 %	20.3 %	- 1.8 %	0.1460	19.3 %	21.5 %
2nd decile (116 – 153)	–	144.8	524	10.5 %	83.0	79.7	- 3.3	15.8 %	15.2 %	- 0.6 %	0.2930	14.8 %	15.6 %
3rd decile (153 – 197)	–	180.5	474	9.5 %	39.0	35.8	- 3.2	8.2 %	7.5 %	- 0.7 %	0.4691	7.2 %	7.9 %
4th decile (197 – 226)	–	217.2	554	11.1 %	26.0	30.7	+ 4.7	4.7 %	5.5 %	+ 0.8 %	0.3894	5.2 %	5.9 %
5th decile (226 – 255)	–	240.4	475	9.5 %	16.0	22.8	+ 6.8	3.4 %	4.8 %	+ 1.4 %	**0.0043**	4.5 %	5.1 %
6th decile (255 – 269)	–	262.2	488	9.8 %	18.0	16.0	- 2.0	3.7 %	3.3 %	- 0.4 %	0.4022	3.1 %	3.5 %
7th decile (269 – 287)	–	276.4	615	12.3 %	6.0	6.4	+ 0.4	1.0 %	1.0 %	+ 0.1 %	0.3019	1.0 %	1.1 %
8th decile (287 – 297)	–	294.3	452	9.1 %	20.0	18.0	- 2.0	4.4 %	4.0 %	- 0.4 %	**0.0301**	3.9 %	4.1 %
9th decile (297 - 319)	–	310.6	494	9.9 %	2.0	7.8	+ 5.8	0.4 %	1.6 %	+ 1.2 %	**0.0311**	1.5 %	1.7 %
10th decile (319+)	–	338.0	416	8.3 %	6.0	7.9	+ 1.9	1.4 %	1.9 %	+ 0.4 %	**0.0172**	1.8 %	2.0 %
Total	–	233.0	4,993	100.0 %	327.0	327.0	0.0	6.5 %	6.5 %	0	T	6.3 %	6.8 %

Source: calculated by the author

Table 15 | Gap analysis of explanatory variable on DEV

CLIENT DURATION (development sample)	Category points	Avg. Total Score	Observations		Number of bads			Bad rate			p-value	95 % conf. interval	
			N	%	observed	predicted	gap	observed	predicted	gap		min	max
0-12 months	0.0	102.9	737	6.3 %	176.0	176.0	+ 0.0	23.9 %	23.9 %	+ 0.0 %	0.4933	22.9 %	24.9 %
13-30 months	52.0	166.1	2,044	17.5 %	319.0	319.0	+ 0.0	15.6 %	15.6 %	+ 0.0 %	0.4925	15.1 %	16.1 %
31-96 months	149.0	272.2	5,368	46.1 %	315.0	315.0	+ 0.0	5.9 %	5.9 %	+ 0.0 %	0.4951	5.7 %	6.0 %
97 + months	110.0	232.2	3,504	30.1 %	92.0	92.0	+ 0.0	2.6 %	2.6 %	+ 0.0 %	0.4977	2.5 %	2.7 %
Total	110.8	230.9	11,653	100.0 %	902.0	902.0	0.0	7.7 %	7.7 %	0	T	7.6 %	7.9 %

Source: calculated by the author

Table 16 | Gap analysis of explanatory variable on VAL

CLIENT DURATION (validation sample)	Category points	Avg. Total Score	Observations		Number of bads			Bad rate			p-value	95 % conf. interval	
			N	%	observed	predicted	gap	observed	predicted	gap		min	max
0-12 months	0.0	97.5	293	5.9 %	72.0	63.3	- 8.7	24.6 %	21.6 %	- 3.0 %	0.0291	20.2 %	23.0 %
13-30 months	52.0	171.9	873	17.5 %	107.0	111.9	+ 4.9	12.3 %	12.8 %	+ 0.6 %	0.3119	12.2 %	13.4 %
31-96 months	149.0	272.7	2,284	45.7 %	114.0	117.1	+ 3.1	5.0 %	5.1 %	+ 0.1 %	0.4577	4.9 %	5.3 %
97 + months	110.0	234.5	1,543	30.9 %	34.0	34.6	+ 0.6	2.2 %	2.2 %	+ 0.0 %	0.3844	2.1 %	2.4 %
Total	111.2	233.0	4,993	100.0 %	327.0	327.0	0.0	6.5 %	6.5 %	0	T	6.3 %	6.8 %

Source: calculated by the author

6. Conclusion

In statistical modelling, it is usually common for researchers to split the original database into the development and validation samples. The representativeness of these samples should be checked to ensure that they do not introduce bias into the study.

Two most common data splitting methods were used for the purposes of splitting the database. Stratified random sampling ensures that the researcher has guaranteed that the samples created from the database will be similar, while simple random sampling offers no guarantee that the subgroups will be represented proportionately or equally.

First, we compared both the methods and noted that stratification random sampling reduces the loss of important datasets and ensures that the confidence level of the samples is high. Without using the strata function, we find that large variability may occur within the response variable (±14.5% in our case) and that the confidence level is low especially

when dealing with a small sample size. This means that there is a very significant risk of the response values (good/bad) being lost in the final sample.

Further, the paper examines the effects on the quality of the credit score when a bias occurs. For this analysis, the database was purposely split poorly using simple random sampling. The impacts are different in dependence on whether the problem was identified on the explained variable or is present on the variable used for the credit score calculation.

To reveal the consequences, three analyses were made. It was shown that if the problem was detected only on the explained variable while all the explanatory variables used in the model are all right, there will be no problem in the model output as long as the risk stability remains unaffected and there are no serious gaps in the gap analysis. However, if there is a bias found in any of the model inputs, weak model performance is highly probable. Serious problems can be indicated by unstable risk between the DEV and the VAL and the presence of huge gaps in the gap analysis.

Stratified random sampling offers various advantages over simple random sampling. First of all, it increases the efficiency of predictors of the overall population parameters through selection of strata that are homogenous over each dataset. It is also advantageous as it focuses on subpopulations of special interest, such as the bad applicants in our case. Finally, stratified random sampling is also convenient and can be used on a smaller sample size as well, which in turn saves time and money.

References

ANDREWS, D., 2000. Inconsistency of the Bootstrap when a Parameter is on the Boundary of the Parameter Space. *Econometrica*. Issue 2, pp 399–405. ISSN 0012-9682.

BOROVICKA, T; JIRINA, M. jr.; KORDIK, P; JIRINA, M., 2012. Selecting Representative Data Sets. In: KARAHOCA, A., ed. *Advances in Data Mining, Knowledge, Discovery and Applications*. InTech. ISBN 978-953-51-0748.

BOWDEN, G.; MAIER, H.; DANDY, G., 2002. Optimal Division of Data for Neural Network Models in Water Resources Applications. *Water Resources Research*. Issue 2, pp 1–11.

ELSAYIR, H. A., 2014. Comparison of Precision of Systematic Sampling with Some Other Probability Samplings. *American Journal of Theoretical and Applied Statistics*. Issue 4, pp 111–117.

FARAWAY, J. J., 1998. Data Splitting Strategies for Reducing the Effect of Model Selection on Inference. [online]. [accessed January 20, 2015]. Available at: http://www.maths.bath.ac.uk/~jjf23/papers/interface98.pdf.

FARAWAY, J. J., 2014. Does Data Splitting Improve Prediction? [online]. [accessed January 20, 2015]. Available at: http://arxiv.org/abs/1301.2983v2.

GEOFF, D.; EVERITT, B. S. 2001. *Handbook of Statistical Analyses using SAS*. 2nd edition. Chapman&Hall/CRC Press. ISBN 9781584882459.

KOHAVI, R., 1995. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence*. Vol. 14, pp 1137–1145.

LOHR, S. L., 1999. *Sampling: Design and Analysis*. 2nd edition. Cengage Learning. ISBN 9780495105275.

MAY, R. J.; MAIER, H. R.; DANDY, G. C., 2010. Data Splitting for Artificial Neural Networks using SOM-based Stratified Sampling. *Neural Networks*. Issue 2, pp 283–294.

MENG, X.; XIE, X., 2014. I Got More Data, My Model is More Refined, but My Estimator Is Getting Worse! Am I Just Dumb? *Econometric Reviews*. Issue 1–4, pp. 218–250.

MOLINARO, A.; SIMON, R.; PFEIFFER, R., 2005. Prediction Error Estimation: a Comparison of Resampling Methods. *Bioinformatics*. Issue 15, pp 3301–3307.

PECK, R.; OLSEN, CH.; DEVORE, J. L., 2012. *Introduction to Statistics and Data Analysis*. 4th edition. Cengage Learning. ISBN 9780840054906.

PICARD, R. R.; COOK, R. D., 1984. Cross-validation of Regression Models. *Journal of the American Statistical Association*. Issue 387, pp 575–583.

REITERMANOVÁ, Z., 2010. Data Splitting. Proceedings of the WDS'10 Nineteenth Annual Conference of Doctoral Students, Prague, Czech Republic, Part I - Mathematics and Computer Sciences, pp 31–36.

SHAO, J., 1993. Linear Model Selection by Cross-validation. *Journal of the American Statistical Association*. Issue 422, pp 486–494.

SNEE, R., 1997. Validation of Regression Models: Methods and Examples. *Technometrics*. Issue 4, pp 415–428.

STONE, M., 1974. Cross-validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society*, series B. Issue 2, pp 111–147.

TIBSHIRANI, R. J.; EFRON, B., 1996. An Introduction to the Bootstrap. *Journal of Economic Literature*. Issue 3, pp 1340–1342.