ABOUT TIME - BUILDING CREDIT SCORECARDS WITH SURVIVAL ANALYSIS

Zhuo Jia Dai

Supervisor: Dr Feng Chen

School of Mathematics and Statistics,
The University of New South Wales.

June 2010

Submitted in partial fulfillment of the requirements of the degree of
Master of Statistics

Acknowledgments

Thanks must go to my supervisor for his guidance and support.

Zhuo Jia Dai, May 2010.

Preface

Credit Risk Scorecards have been one of the most successful applications of statistics in banking. It is used throughout the world by institutions that lend credit to consumers and has been vital in allowing the "phenomenal" level of growth in consumer credit over the past four decades (see preface to [1]).

Despite its enormous utility the subject has received much less attention when compared to the pricing of exotic financial derivatives. As a result, the literature on consumer Credit Risk Scoring can arguably be described as 'thin on the ground'. This thesis aims to fill some of the gaps in the literature particularly in the field of building Credit Risk Scorecards with Survival Analysis.

The end users of scorecards are the risk managers who often do not have a mathematics background. It is important that a "lay-person" without advanced mathematical skills can understand and apply a scorecard to make credit decisions. The current literature on survival analysis scorecards often do not address this aspect of scorecard building. This thesis develops the techniques involved in building survival analysis scorecards that has the same "look and feel" as the scorecards currently in use in the industry. This is an important issue to address if survival analysis scorecards are to be widely adopted.

Migrating from building traditional scorecards to survival analysis scorecards holds a number of promising opportunities not readily available currently. For example, performing profit scoring is possible with survival analysis scorecards [11] but not with traditional scorecards; and stress testing using macroeconomic variables can be performed using survival analysis models [3] but not with traditional scorecard models. This thesis provides the theoretical and methodological foundations to building those features into the scorecards in the future.

This thesis is organised as follows: **Chapter 1.** gives an overview of credit risk scorecards, the data and the model being employed in this project; **Chapter 2.** details an original coarse classification algorithm, Automatic Binary Binning Algorithm (ABBA), designed for building credit risk scorecards; **Chapter 3.** discusses in detail the modelling methodology including variable selection and model checking/validation; the chapter also discusses in detail how to turn the model into a scorecard.

Contents

Chapter 1

Introduction

1.1 What is a Credit Risk Scorecard?

1.1.1 Business Presentation of Scorecards

A scorecard is a simple recipe for assigning a numeric score to every account. The assumption is that the higher the account scores the less risky it is. A scorecard is often presented in a way that is easy for a "lay person" to understand; for example see the mocked-up scorecard in Table 1.1. To use the scorecard to determine an account's score, simply determine the range in which the account's variables lie and add the corresponding score to the base score. Once this is performed for all three variables then we have the final score of the account. For example, suppose account a's variable 1 equals to 100, variable 2 equals to 350, and variable 3 equals to 5. Then its score is 622 + -8 + 11 + 37 = 662.

As can be seen a scorecard is meant to be extremely simple to explain and apply. The scores from the scorecard may be used in a number of different ways. For example the business may set a cut-off score and if an account's scores is below it then the associated customer will not be offered further loan products. For other practical uses of scorecards see [1] or [5].

1.1.2 The Traditional Model - Binary Logistic Regression

Traditionally credit risk scorecards have been built using binary logistic regression (BLR). Let $\mathcal{A}$ be a set of accounts, and let $\mathbf{x}_a$ be a vector of raw and untransformed attributes associated with the account $a \in \mathcal{A}$ at the present time. Let the default indicator, D, be a function in the form of

$$D : \mathcal{A} \times \mathbb{N} \to \{0, 1\}$$

where

$$D(a, t) := \begin{cases} 1 & ; \text{if the account is in default } t \text{ months in the future} \\ 0 & ; \text{otherwise} \end{cases}$$

Remark 1.1.1. An industry standard for the definition of "in default" is 3 payments behind or 90+ days delinquent. □

Variable	Attribute	Score
Base Score		622
Var 1	Low-<0	-32
Var 1	0-<70	-13
Var 1	70-<275	-8
Var 1	275-<700	-1
Var 1	700-<1200	6
Var 1	1200-<2100	10
Var 1	2100-<6500	16
Var 1	6500<-High	21
Var 1	Other	0
Var 2	Low-<150	27
Var 2	150-<250	25
Var 2	250-<300	20
Var 2	300-<450	11
Var 2	450<-High	3
Var 2	Other	0
Var 3	1-<2	17
Var 3	2-<3	21
Var 3	3<-High	37
Var 3	Other	0

Table 1.1: A mocked-up scorecard

Definition 1.1.2. An account is defined as 'bad' if

$$B(a) := \max_{i=1,2,..,12} D(a,i)$$

is equal to 1; otherwise it is good. That is an accounts is bad if it goes into default any time during the next twelve months.

The idea is to build a model that predicts $\mathbb{E}(B(a|\mathbf{x}_a))$, the expected probability that account a is going to go bad given the list of attributes associated with it at the present time. This is often referred to as the PD (probability of default) model in the industry.

Since $B(a)$ is a binary outcome variable, binary logistic regression seems to be the ideal modelling technique. The model being considered in the BLR setting is

$$\log \frac{p_a}{1-p_a} = \mathbf{T}(\mathbf{x})'\boldsymbol{\beta}$$

where $1-p_a$ is account $a \in \mathcal{A}$'s probability of default , i.e. $p_a := P(B(a)=0|\mathbf{x}_a)$.

However it can be seen that the definition of B depends on twelve months of data; indeed it is a function of $D(a,1), D(a,2), ..., D(a,12)$. The choice of twelve can be seen as arbitrary. The user could easily have defined $B(a)$ using any number of

months of data. If time is such a vital ingredient in the definition of B, is it not better to model the time to default instead? The same question has been asked in a number of papers including [7].

1.1.3 About Time - The Cox Proportional Hazard model

If we aim to be able to take into account time to default (not just if) then Survival Analysis techniques would be ideal. There are a number of popular survival analysis models to choose from but Cox Proportional Hazard Models, pioneered in [8], is invariably chosen to do the modelling in the current literature.

In this project we will use the Cox Proportional Hazards model exclusively. The Cox Proportional Hazard model is chosen as it has been successfully applied in the credit risk setting before, see [2] and [3] for example.

Let T_a be the number of months till the next default for an account $a \in \mathcal{A}$. For now assume that T_a is a continuous variable.

Definition 1.1.3. The *hazard* of T_a, $h(t|\mathbf{x}_a)$, is defined as

$$h(t|\mathbf{x}_a) := \lim_{\delta \to 0} \frac{P(t <= T_a < t + \delta \mid T_a >= t)}{\delta}$$

which can be shown to be (see ([4],p12))

$$h(t|\mathbf{x}_a) = \frac{f(t|\mathbf{x}_a)}{S(t|\mathbf{x}_a)}$$

where $f(t)$ is the probability distribution function of T_a and $S(t) := 1 - P(T_a \leq t)$. We can easily see that

$$h(t|\mathbf{x}_a) = \frac{-S'(t|\mathbf{x}_a)}{S(t|\mathbf{x}_a)}$$

which, by $\frac{d \log g(t)}{dt} = \frac{g'(t)}{g(t)}$ for any differentiable $g(t)$, can be written as

$$-h(t|\mathbf{x}_a) = \frac{d \log S(t|\mathbf{x}_a)}{dt}$$

integrate both sides and exponentiate to obtain

$$\exp(\int -h(t|\mathbf{x}_a)dt) = S(t|\mathbf{x}_a)$$

Definition 1.1.4. The proportional hazard assumption proposed in the *Cox Proportional Hazard model* expects that

$$h(t|x_a) = h_0(t) \exp(\mathbf{T}(\mathbf{x}_a)'\boldsymbol{\beta})$$

where $h_0(t)$ is some unknown baseline hazard function, $\mathbf{T}$ is some function of $\mathbf{x}_a$ chosen by the user, and $\boldsymbol{\beta}$ is a vector of parameters.

Remark 1.1.5. The name proportional hazard comes from the fact that the ratio of any two accounts' hazards is constant over time. Indeed let account a have attribute $\mathbf{x}_a$ and account b have attribute $\mathbf{x}_b$ then their hazard ratio is

$$\frac{h(t|\mathbf{x}_a)}{h(t|\mathbf{x}_b)} = \frac{h_0(t)\exp(\mathbf{T}(\mathbf{x}_a)'\boldsymbol{\beta})}{h_0(t)\exp(\mathbf{T}(\mathbf{x}_b)'\boldsymbol{\beta})} = \frac{\exp(\mathbf{T}(\mathbf{x}_a)'\boldsymbol{\beta})}{\exp(\mathbf{T}(\mathbf{x}_b)'\boldsymbol{\beta})}$$

which is clearly constant over time. □

Remark 1.1.6. In the above derivation the function $\mathbf{T}$ is left unexplained. It is not simply a transformation of $\mathbf{x}$. I fact in this project $\mathbf{T}$ turns each component of $\mathbf{x}$ into a number of factors. This is called Coarse Classification of variables. The next chapter deals with coarse classification in detail by introducing an original automated algorithm that has been applied in this project with considerable success! □

Normally it would be necessary to make assumptions regarding $h_0(t)$ (or equivalently $f(t)$) before we can start modelling. However by a partial likelihood argument, which we will not detail in this thesis, Cox managed to derive a partial likelihood function $L(\boldsymbol{\beta})$ that does not depend on $h_0(t)$. Hence we can maximise $L(\boldsymbol{\beta})$ to find the maximum likelihood parameter estimates for $\boldsymbol{\beta}$ without knowledge of the baseline hazard. This is one reason why the Cox Proportional Hazard model is so popular. The details regarding the derivation of the partial likelihood can be found in [4] or [8].

From the above Definition we have

$$S(t|\mathbf{x}_a) = \exp(-\int_0^t h(t|\mathbf{x}_a)dt) = \exp\left(-\int_0^t h_0(t)dt\exp(\mathbf{T}(\mathbf{x}_a)'\boldsymbol{\beta})\right)$$

setting

$$S_0(t) := \exp\left(-\int_0^t h_0(t)dt\right)$$

allows us to rewrite the above as

$$S(t|\mathbf{x}_a) = S_0(t)^{\exp(\mathbf{T}(\mathbf{x}_a)'\boldsymbol{\beta})}$$

Suppose we applied some maximum likelihood procedure on the partial likelihood $L(\boldsymbol{\beta})$ and obtained some estimates, $\hat{\boldsymbol{\beta}}$, of $\boldsymbol{\beta}$, then the only other estimate needed to build a model of $S(t|x_a)$ is an estimate of $S_0(t)$.

Censoring

One complicating factor in survival analysis models is that we do not always observe the time to default, T_a, for every account $a \in A$. That is because some accounts can close before they go into default or our observation period ends and the account still hasn't defaulted. This problem is often referred to as the censoring problem. Instead of observing T_a we observe the censored time, T_a^*.

Definition 1.1.7. The *censored time*, T_a^*, is defined as

$$T_a^* := \begin{cases} T_a & ; \text{if the account defaults at } T_a \\ \text{the time to closure or end of observation period} & ; \text{otherwise} \end{cases}$$

This particular type of censoring where $T_a \geq T_a^*$ is called right censoring. We will only deal with right censoring in this thesis.

Estimating the survival function

The function $S_0(t)$ is the baseline survival function. In this project it can be thought of as the survival function of an "average" account. The Kaplan-Meier estimator [17] is a well-accepted estimator for $S_0(t)$, and it takes into account the censoring by design.

Definition 1.1.8. Let $R_{\mathbf{x}}(t) \subseteq \mathcal{A}$ be the set of accounts satisfying $\mathbf{T}(\mathbf{x}_a) = \mathbf{T}(\mathbf{x})$ and $T_a^* \leq t$. This set is called the at risk set at time t. Let b_t be the number of accounts in $R_{\mathbf{x}}(t) \subseteq \mathcal{A}$ that have gone bad at time t. The Kaplan-Meier estimator, $\hat{S}_0(t)$, of $S_0(t)$ is defined as

$$\hat{S}_0(t) = \begin{cases} 1 & ; \text{if } t = 0 \\ \left(1 - \frac{b_t}{|R_{\mathbf{x}}(t)|}\right) \hat{S}_0(t-1) & ; \text{if } t \geq 1 \end{cases}$$

Remark 1.1.9. In this project $\mathbf{T}(\mathbf{x}_a)$ is a vector of factors with varying number of levels. Therefore we can "absorb" one level from each factor into the estimate of $S_0(t)$ so that their associated parameters need not be estimated. Therefore $S_0(t)$ is actually the survival function of the account with those "absorbed" levels. We refer to such accounts as the "average" accounts. It is sensible to choose this "average" account to have levels that appear in high frequency. This way any estimate of $S_0(t)$ is likely to be less volatile and have a smaller confidence interval. □

Definition 1.1.10. Another estimator is the Nelson-Aalen estimator [18] which is defined as

$$\tilde{S}_0(t) := \prod_{j=1}^{t} \exp\left(\frac{-b_j}{|R_{\mathbf{x}}(j)|}\right)$$

Remark 1.1.11. When applied to the data in this project both estimators produced fairly similar results. Also it will become apparent that the choice of estimator does not actually affect the risk rank-ordering performance of a scorecard. Therefore the choice of estimator is unimportant in the scorecard build context. □

Utilising the above we have an estimate of $S(t|\mathbf{x}_a)$ in the form of

$$\hat{S}(t|\mathbf{x}_a) = \hat{S}_0(t)^{\exp(\mathbf{T}(\mathbf{x}_a)'\hat{\boldsymbol{\beta}})}$$

where $\hat{S}_0(t)$ comes from the Kaplan-Meier estimator (or the Nelson-Aalen estimator) and $\hat{\boldsymbol{\beta}}$ comes from maximising the partial likelihood.

Recall that

$$S(t|\mathbf{x}_a) := 1 - P(T_a \leq t)$$

which means that we have built up some idea of the distribution of T_a, the time to next default, through obtaining the estimator $\hat{S}(t|\mathbf{x}_a)$. Notice that if

$$\hat{S}(t^*|\mathbf{x}) >= \hat{S}(t^*|\mathbf{y})$$

for some t^* then

$$\hat{S}(t|\mathbf{x}) >= \hat{S}(t|\mathbf{y})$$

for all t since

$$\frac{\hat{S}(t^*|\mathbf{x})}{\hat{S}(t^*|\mathbf{y})} = \frac{\exp(\mathbf{T}(\mathbf{x})'\boldsymbol{\beta})}{\exp(\mathbf{T}(\mathbf{y})'\boldsymbol{\beta})} = \frac{\hat{S}(t|\mathbf{x})}{\hat{S}(t|\mathbf{y})}$$

by definition of $\hat{S}(t|\mathbf{x})$. Therefore we can choose an arbitrary t and use the values of $\hat{S}(t|\mathbf{x}_a)$ for all $a \in A$ to rank-order risk in the accounts. We can suppose that the lower the value of $\hat{S}(t|\mathbf{x}_a)$ the riskier the account. To align our survival analysis model so that it can be compared with BLR we often set $t = 12$.

Remark 1.1.12. In fact we will simply use the score, $\mathbf{T}(\mathbf{x})'\boldsymbol{\beta}$, to rank-order risk, hence why the choice of estimator of S_0 is unimportant. However it is convenient to have a model for $S(t|\mathbf{x})$ as this allows us to perform comparison with the BLR model (which models the probability of default which is essentially equivalent, in concept, to $1 - S(t|\mathbf{x})$). □

1.2 The Data

The data used in this project is two years worths of month-end snapshots from a consumer credit leading product portfolio from an Australian bank; and there were approximately 1,000,000 observations. The data is broken into two datasets: the development sample - used to develop the models, and the holdout sample - used

Month	ID	In Default	Closed	Var 1	Var 2	...	Var n
1	1	No	No	100	-50	...	0.1
2	1	No	No	100	1000	...	0.11
3	1	No	No	200	500	...	0.1
4	1	No	No	100	-50	...	0.1
5	1	No	No	100	1000	...	0.11
6	1	No	No	200	500	...	0.1
7	1	No	No	200	385	...	0.1
8	1	No	No	211	500	...	0.1
9	1	No	No	311	-100	...	0.1
10	1	No	No	429	-100	...	0.1
11	1	No	No	214	560	...	0.1
12	1	No	No	199	-597	...	0.1
13	1	No	No	0	112	...	0.1
14	1	No	No	50	113	...	0.1
15	1	No	No	100	114	...	0.1
16	1	No	No	100	115	...	0.1
17	1	Yes	No	9	-300	...	0.1
1	2	No	No	100	-50	...	0.1
2	2	No	No	100	1000	...	0.11
3	2	No	No	200	500	...	0.1
4	2	No	No	100	-50	...	0.1
5	2	No	Yes	100	1000	...	0.11
...	...	...	...	...	...	...	

Table 1.2: Example Raw Data

to validate the model. The development sample has approximately 800,000 observations.

Each observation corresponds to an account's attributes at one point in time. Each account has an ID variable, a Bad Outcome variable and a Time To Event variable. The Bad Outcome variable serves as the censoring variable as well, where a Bad Outcome of value 0 means the account is right-censored.

For each observation we compute the number of months it takes the account to go into default, or if censored the number of months to censoring. This information is stored in the Time To Event variable.

An example of the raw data can be found in Table 1.2. The variable Month indicates the month from which the data was obtained, ID uniquely identifies the accounts, In Default is equivalent to the default indicator $D(a,t)$ where if $D(a,t) = 1$ then In Default will be a Yes otherwise a NO at month t, and Var 1 to Var n are the attributes associated with the account. The prepared data can be seen in Table 1.3.

Remark 1.2.1. The data was prepared in SAS. □

Month	ID	In Default	Closed	Time To Event	Var 1	Var 2	...	Var n
1	1	No	No	16	100	-50	...	0.1
2	1	No	No	15	100	1000	...	0.11
3	1	No	No	14	200	500	...	0.1
4	1	No	No	13	100	-50	...	0.1
5	1	No	No	12	100	1000	...	0.11
6	1	No	No	11	200	500	...	0.1
7	1	No	No	10	200	385	...	0.1
8	1	No	No	9	211	500	...	0.1
9	1	No	No	8	311	-100	...	0.1
10	1	No	No	7	429	-100	...	0.1
11	1	No	No	6	214	560	...	0.1
12	1	No	No	5	199	-597	...	0.1
13	1	No	No	4	0	112	...	0.1
14	1	No	No	3	50	113	...	0.1
15	1	No	No	2	100	114	...	0.1
16	1	No	No	1	100	115	...	0.1
1	2	No	No	4	100	-50	...	0.1
2	2	No	No	3	100	1000	...	0.11
3	2	No	No	2	200	500	...	0.1
4	2	No	No	1	100	-50	...	0.1
...	...	...	...	...	...	...		

Table 1.3: Example Prepared Data

Chapter 2

Coarse Classification of Variables

One of the problems with fitting a regression model in general is finding appropriate transforms of the explanatory variables. This problem is especially pronounced in the Cox Proportional Hazard model. Consider a continuous variable x; we can try to fit x as a linear explanatory term in the model as below

$$\frac{h(t|x)}{h_0(t)} = \exp(\beta x)$$

However the linearity assumption may not be appropriate for x. So we may use a function g to transform x and fit the following instead

$$\frac{h(t|x)}{h_0(t)} = \exp(\beta g(x))$$

For example g could be the log function. But the fact that $h_0(t)$ and $h(t|x)$ are unknown makes choosing an appropriate transform more difficult than in least-square linear regression. This is complicated further when we have a large number of variables to choose from, as is the case in this project.

What transform should we apply to the variables? In ([4],p88) it was suggested that we band the each variable into 4 or 5 bands with approximately an equal number of observations in each and create a factor with levels corresponding to the banding. Of course here the "4 or 5" bands can be replaced with a general n. The original application was to investigate if linear trends exist in the covariates. However in Credit Risk Scorecard modelling it is common to use those factors and their parameter estimates in the final model instead of incorporating the original continuous variable. The reasons for doing this are well justified in ([1],p132) with the argument being that using factors allows us to make predictions; but fitting a transformed x is better at helping us to explain the underlying drivers of risk. A scorecard is aimed at making predictions and so using the factors approach is appropriate.

Another reason for turning variables into bands (or bins) is that many variables have legitimate nulls values which need to be treated as a separate bin if we were to estimate the parameters associated with them. For example consider the variable "number of credit cards associated with the customer", for customers that do not have credit cards this variable will be null. Therefore the variable can not be

fitted in the model unless it was turned into a factor with null being one of the levels.

Turning a numeric variable into n bins/groups is called coarse classification or binning as we will prefer in this thesis. There are a number of standard ways to bin the numeric variables. The most suggested method seems to be to break the variable into bins where each bin contains 5%, 10% or 20% etc of accounts. This is seen in [2], [4], and [10]. The benefit of this method is that it is simple to implement. Another strategy is introduced in [10] where a simulated annealing [6] method is applied. This method is complex and is not easily achievable with current software; it is likely that the modeller has to write fairly advanced code to achieve this. This thesis proposes a simple algorithm that is more flexible than the equal size binning method, and more realistically implementable in a short time frame than the simulated annealling method. This algorithm is called Automatic Binary Binning Algorithm (ABBA) and it has an extension that incorporates some hill-climbing optimisation ideas.

2.1 Automatic Binary Binning Algorithm (ABBA)

The Automatic Binary Binning Algorithm is an original and novel approach to coarse classification. Since it is an automated algorithm it is sometimes referred to as an adjacent pooling algorithm, see [5]. The ABBA algorithm is fast, easy to implement, and can be used to incorporate business considerations into the binning. The algorithm was coded in R [14] (contact the author for source code). ABBA performs the binnings using a user designed heuristic and from the results it can be seen that the algorithm works extremely well. Indeed, one of the scorecards presented in this thesis was built completely from the automated binnings that ABBA created. The resultant scorecard is competitive against an industry built scorecard.

The central idea behind ABBA is to bin the numeric variables into bins with some clear pattern to the ratio of bads/goods. For example, we may wish to see a decreasing trend in the ratio of bads/goods. This is sensible in the case where the variable in question is Income. Indeed one would expect that the higher the income of the individual the less likely his/her account will go bad hence the decreasing trend.

Definition 2.1.1. Let v be an arbitrary numeric attribute, for example credit card balance, and let

$$U := \{\text{null}, u_1, u_2, \cdots, u_n\}$$

be the complete set of values of v in the data ordered to satisfy $u_i < u_j$ if $i < j$. Also define the two-tuple

$$C_{u_i} := (b_{u_i}, g_{u_i})$$

where b_{u_i} and g_{u_i} is the number of bad accounts and good accounts, respectively, in our data satisfying $v = u_i \in U$. For example, if $u_i = 2000$ then b_{u_i} is the number

of accounts that went bad that has a credit card balance of 2000. And for $V \subseteq U$ define

$$C_V := \left(\sum_{u \in V} b_{u_i} \ , \ \sum_{u \in V} g_{u_i} \right)$$

Finally define an ordered-set

$$C := \left[\ C_{u_1}, C_{u_2}, \ldots, C_{u_{|U|}} \ \right]$$

, that is let C be the ordered-set of all the bins. For convenience we may use the following representations of C

$$C := \left[\ D_1, D_2, \ldots, D_{|C|} \ \right]$$

and

$$C := \left[\ (b_1, g_1), (b_2, g_2), \ldots, (b_{|C|}, g_{|C|}) \ \right]$$

In the above definition each $C_u \in C$ is a bin and C is the ordered-set of all bins. When two bins are "combined" we simply add the number of goods and bads together and form a new bin with the new goods and bads. If the bins C_{u_i} and $C_{u_{i+1}}$ are combined to form a new bin $C_{\{u_i, u_{i+1}\}}$ we write

$$C_{\{u_i, u_{i+1}\}} \leftarrow C_{u_i} + C_{u_{i+1}}$$

In general if bin B is combined with bin D to form bin E we write

$$E \leftarrow B + D$$

Remark 2.1.2. The null value is always treated as a separate bin. In our modelling dataset a lot of numeric variables have legitimate null values that we want to obtain parameter estimates for. This is done in order to get a complete view of the relative risks of all the accounts, even those with null values. □

Definition 2.1.3. Consider an arbitrary subset $V \subseteq U$ and suppose $V := \{\text{null}, v_1, v_2, v_3, ..., v_{|V|-1}\}$ such that $v_i < v_{i+1}$ for all i. A *binning function* T using V as the cut points is defined as

$$T : U \mapsto \{0, 1\}^{|V|+1}$$

with

$$T(u) = \Big(I_{\{\text{null}\}}(u), \ I_{(-\infty, v_1]}(u), \ I_{(v_1, v_2]}(u), ..., \ I_{(v_{|V|-2}, v_{|V|-1}]}(u), \ I_{(v_{|V|-1}, \infty)}(u) \Big)$$

where $I_S(s)$ is the indicator function defined as

$$I_S(s) := \begin{cases} 1 & ; \text{if } s \in S \\ 0 & ; \text{otherwise} \end{cases}$$

In words, T turns a numeric variable into a vector of dummy variables based on some chosen cut points.

Remark 2.1.4. Suppose a null value is a legitimate value of v, then using T as a "transform" allows us to fit v into the model. A continuous transform, such as log, does not have this property as the transform of the null value is undefined. □

Definition 2.1.5. A *focus* is a function f

$$f : \{C\} \mapsto \mathcal{P}(\{1, 2, 3, ..., |C| - 1\})$$

where $\mathcal{P}(X)$ is the powerset of X, i.e. the set of all subsets of X.

Remark 2.1.6. The focus tells us which bins to concentrate our attention on. In the ABBA algorithm only those bins that are in "focus" will be considered for combination. □

Definition 2.1.7. An *information loss* function i is a function of the form

$$i : C \times C \mapsto \mathbb{R}$$

An information loss function is used to measure the "information" lost in combining two bins together.

Definition 2.1.8. Using the representation $C := \{D_1, D_2, \ldots, D_{|C|}\}$. The *ABBA Algorithm* is as follows

1. If $|C| = 1$ then terminate
2. Compute $N := f(C)$
3. If $|N| \geq 1$ then continue, else terminate
4. Find $D_j \in \{D_n \mid n \in N\}$ such that $i(D_j, D_{j+1})$ is minimised
5. Compute a new C by setting $D_j \leftarrow D_j + D_{j+1}$, then removing D_{j+1}, and then renaming every D_{j+k} to D_{j+k-1} for all applicable $k \geq 1$
6. Repeat from Step 1 with the new C

In essence ABBA uses the focus function, f, to identify adjacent pairs of bins where some expected pattern is not being met. It then decides which of those pairs of bins to combine using the information loss function i. Once all the expected patterns are met then the focus function f will return an empty set and the algorithm terminates. ABBA is extremely simple but its flexibility comes from the focus function f and the information loss function i. Choosing a good focus function will allow us to create flexible coarse classifications that other algorithms can not!

2.1.1 Some choices of focus

Upward Trend Focus

Suppose you wish to see a monotonically increasing trend in the ratio of bads/goods in the resultant bins, i.e. at the end of the algorithm the binning should satisfy

$$\frac{b_j}{g_j} < \frac{b_{j+1}}{g_{j+1}}$$

for all applicable j and $(b_j, g_j) \in C$ as defined further above. We can define an upward trend focus, f_T, to be

$$f_T(C) := \left\{ \; j \mid \frac{b_j}{g_j} \geq \frac{b_{j+1}}{g_{j+1}} \text{ where } (b_j, g_j) = D_j \in C \right\}$$

In words let f_T return all the positions where the monotonically increasing trend is not satisfied. It can bee seen that Step 5 of the algorithm would only consider the bins where the monotonically increasing trend is broken. It is then an easy consequence that when the algorithm terminates at Step 3 all the remaining bins satisfy the monotonically increasing criterion for bad/good odds.

Remark 2.1.9. Clearly a downward trend focus, f_F , can be defined similarly. □

Pearson's χ^2 Focus

Suppose you wish to see that every pair of adjacent bins should be "statistically different". One way to achieve this is to require that the Pearson's χ^2 test statistics, T, be greater than some chosen T^*. We define the *Pearson's χ^2 Focus*, f_P, as

$$f_P(C|T^*) = \{ \; j \mid \frac{(\hat{b}_j - b_j)^2}{\hat{b}_j} + \frac{(\hat{g}_j - g_j)^2}{\hat{g}_j} + \frac{(\hat{b}_{j+1} - b_{j+1})^2}{\hat{b}_{j+1}} + \frac{(\hat{g}_{j+1} - g_{j+1})^2}{\hat{g}_{j+1}} <= T^* \}$$

where $(b_j, g_j) \in C$ as defined further above and

$$\hat{b}_j := \frac{(b_j + g_j)(b_j + b_{j+1})}{b_j + g_j + b_{j+1} + g_{j+1}}$$

and

$$\hat{g}_j := \frac{(b_j + g_j)(g_j + g_{j+1})}{b_j + g_j + b_{j+1} + g_{j+1}}$$

and

$$\hat{b}_{j+1} := \frac{(b_{j+1} + g_{j+1})(b_j + b_{j+1})}{b_j + g_j + b_{j+1} + g_{j+1}}$$

and

$$\hat{g}_{j+1} := \frac{(b_{j+1} + g_{j+1})(g_j + g_{j+1})}{b_j + g_j + b_{j+1} + g_{j+1}}$$

are simply the expected number of bads and goods in the bin under the assumption of homogeneity in the two bins. This way, by choosing a value of T^* we can ensure that our bins are arbitrarily statistically significantly distinct.

Remark 2.1.10. The author usually chooses T^* such that $P(\chi^2 > T^*) = 1 - \epsilon_m$ where ϵ_m is the negative machine epsilon, the smallest number such that $1 - \epsilon_m$ is still being stored in the computer memory as different from 1. This ensures that the bins are as distinct as the computer can manage. The R code to compute such a T^* is $Tstar = qchisq(1 - .Machine\$double.neg.eps, 1)$ which yields $T^* \approx 68.76325$. □

Upward Pointing Turning Point Focus

Suppose you wish to see an upward pointing turning point in the bad/good odds i.e. at the end of the algorithm there should exist a j such that

$$\frac{b_i}{g_i} < \frac{b_{i+1}}{g_{i+1}}$$

and

$$\frac{b_k}{g_k} > \frac{b_{k+1}}{g_{k+1}}$$

for all i, k satisfying $i < j < k$ and $(b_j, g_j) \in C$. We can define a upward pointing turning point focus, f_{TU}, as

$$f_{TU}(C) = \begin{cases} \emptyset & ; \text{if the above condition is satisfied} \\ \{1, 2, 3, ..., |C| - 1\ \} & ; \text{otherwise} \end{cases}$$

This way it can be seen that at Step 5 of the algorithm it will consider all bins for combination unless a turning point is found. It is then an easy consequence that when the algorithm terminates at Step 3 all the remaining bins would satisfy the turning point condition. It must be noted that it is entirely possible for a turning point to exist and for this algorithm not to find it. However the author has successfully applied this algorithm to find turnings points in variables; for example see Figure 2.3.

Remark 2.1.11. The Downward Pointing Turning Point Focus, f_{TD}, can be defined by simply reversing the signs in the inequality conditions. □

Turning Point Focus
Suppose we wish to see a turning point but do not want to specify a direction then we can define a general turning point focus as

$$f_{TP}(C) := (f_{TU} \cap f_{TD})(C)$$

Remark 2.1.12. The above turning point focus has an elegant solution in R. Let $b := (b_1, b_2, \ldots, b_{|C|})$ be the vector of bads and $g := (g_1, g_2, \ldots, g_{|C|})$ be the vector of goods, then a turning point (upward or downwards) exists if and only if (in R code)

$$\text{sum(abs(diff(sign(diff}(b/g)\text{))) } == 2) == 1$$

is true. □

Minimum Population Focus
Suppose you wish to see that each bin contains a certain number of bad or a certain number of accounts then you can define the minimum population focus, f_{MP} to be

$$f_{MP}(C|b, p) := \{\ j \mid b_j < b \text{ and } b_j + g_j < p\ \}$$

for some chosen $b, p \in \mathbb{N}$.

Remark 2.1.13. This focus is particular useful in determining an appropriate binning focus for the variables. For example the author found it useful as an initial investigative step to bin each variable using the Minimum Population Focus using $b = \sum_{C_{u_i}} b_{u_i}/20$, $p = \sum_{C_{u_i}} (b_{u_i} + g_{u_i})/20$, i.e. require that each bin must have at least 5% of bads or 5% of the total population. The resulting good/bad ratio plot usually indicates whether a trend or turning point focus is appropriate. This focus is also particular useful as the business may require that the number of accounts in each bin to be quite stable over time, and setting a minimum population requirement is a good proxy to achieving that aim. □

Remark 2.1.14.
As can be imagined the list of focus here is by no means exhaustive. Also in practice the above focus functions are often combined to produce more useful results. For example you may require that each bins has some minimum population requirements and have a linear trend. In that case one can define a new focus $f_N := f_T \cup f_{MP}$. □

2.1.2 Some choices of information loss

ABBA is a greedy algorithm and some heuristic is used to decide which bins to combine. So what is a good heuristic for this purpose? For example it makes sense to combine bins that have a similar good bad profile and it makes sense to combine smaller bins together first as that will affect less accounts. It can be seen that the Pearson's χ^2 statistics is well suited to this. The idea is to compute Pearson's statistics for all the possible combinations and then combine the bins with the lowest test statistic. This ensures that the least statistically significantly distinct bins

gets combined first which is an intuitive heuristic to use.

Definition 2.1.15. The *Pearson's Information Loss*, i_P, is defined as

$$i_P(C_u, C_w) := \frac{(\hat{b}_u - b_u)^2}{\hat{b}_u} + \frac{(\hat{g}_u - g_u)^2}{\hat{g}_u} + \frac{(\hat{b}_w - b_w)^2}{\hat{b}_w} + \frac{(\hat{g}_w - g_w)^2}{\hat{g}_w}$$

where $C_u = (b_u, g_u)$ and $C_w = (b_w, g_w)$ and $\hat{b}_u := \frac{(b_u+g_u)(b_u+b_w)}{b_u+g_u+b_w+g_w}$, $\hat{g}_u := \frac{(b_u+g_u)(g_u+g_w)}{b_u+g_u+b_w+g_w}$, $\hat{b}_w := \frac{(b_w+g_w)(b_u+b_w)}{b_u+g_u+b_w+g_w}$ and $\hat{g}_w := \frac{(b_w+g_w)(g_u+g_w)}{b_u+g_u+b_w+g_w}$ as before.

As can be seen the definition is essentially an application of the Pearson's χ^2 statistics calculation. This statistic gives a measure of how "similar" the two bins are. The smaller the value the more similar in risk profile are the two bins. Hence bins with the smallest test statistics will be combined first.

Definition 2.1.16. The *Binary Information Loss*, i_B, is defined as

$$i_B(C_u, C_w) := (b_u + g_u)\left(\frac{b_u}{b_u + g_u} - \frac{b}{b+g}\right)^2 + (b_w + g_w)\left(\frac{b_w}{b_w + g_w} - \frac{b}{b+g}\right)^2$$

where $b = b_u + b_w$ and $g = g_u + g_w$.

The logic behind the binary information loss is as follows: suppose we were to combine the two bins then we will "misjudge" the probability of bad in the two bins by $\frac{b_u}{b_u+g_u} - \frac{b}{b+g}$ and $\frac{b_w}{b_w+g_w} - \frac{b}{b+g}$. If we square those differences and weight them by the number of accounts in each bin then we have a measure of how much "information" is loss by combining the two bins. If the number of accounts in both bins are small then even if the combined bad rate is hugely different a small information loss value will result; also if one bin is small and the other bin has a large number of accounts then the small bin wouldn't affect the overall bad rate as much so again a small information loss will result. This shows that Binary Information Loss naturally favours combining small bins with large ones, which is a desirable property; it also tends to favour combining bins with similar bad/good ratios which also makes sense.

Remark 2.1.17. It was noted that the Pearson's χ^2 information loss and the Binary Information loss produced similar binnings. However the Pearson's Information Loss was chosen to perform the binning in the model as it has a statistical grounding. □

Remark 2.1.18. ABBA can be extended to categorical variables. The only difference between a categorical variable and a numeric variable is the potentially non-ordinal structure of the variable. Therefore the simplest way is to modify Step 4 of the simple algorithm to compare all possible combinations of bins instead of adjacent ones as in the numerical variables case. However this algorithm is not implemented. □

Table 2.1: The good bad profile

Bin	Bads	Goods	Total	Ratio	χ^2 stat	Broken	$< T^*$
1	243928	17946804	18190732	0.0136	204832.78		
2	363264	8537493	8900757	0.0425	49127.14		
3	109380	1181924	1291304	0.0925	2106.08		
4	55615	467417	523032	0.1190	1691.82	**	
5	17279	210749	228028	0.0820	0.00		*
6	12913	157441	170354	0.0820	100.66		
7	12064	128844	140908	0.0936	48.58	**	
8	8291	98221	106512	0.0844	183.73	**	
9	4676	71565	76241	0.0653	1.13	**	
10	3285	51550	54835	0.0637	21.50		
11	2411	33273	35684	0.0725	84.17		
12	1836	18858	20694	0.0974	100.23	**	
13	1079	16476	17555	0.0655	15.54	**	
14	4190	73499	77689	0.0570			

Remark 2.1.19. Other coarse classification methods exists. One example is the maximum likelihood monotone classifier as introduced in [1]. However these algorithms can not take into account business considerations such as minimum number of bads in each bin. Also the algorithm can only be used to discover monotonic trends where as ABBA can find other patterns such as turning points. □

2.2 An Example of ABBA

Consider a "Number of late payments in the last 14 months" variable with the following good bad profile, see Table 2.1.

Remark 2.2.1. No such variable exist in our data. This is a mocked-up example. □

It is clear from Figure 2.1 that the bad/good odds increased substantially for values from 1 to 4. From 5 onwards the pattern is unclear. However it can be argued that the more consecutive delay payments you have against you the more likely you are to become bad. So we can try to bin this data using the upward trend focus.

We apply the ABBA algorithm with the upward trend focus combined with a Pearson's χ^2 focus (using $T^* = 68.76325$) and Pearson's χ^2 information loss. It can be seen from Table 2.1 that the upward bad/good ratio trend is broken at 6 places and the minimum χ^2 test statistics is between bin 5 and bin 6. Therefore we should combine bin 5-6, which yields Table 2.2.

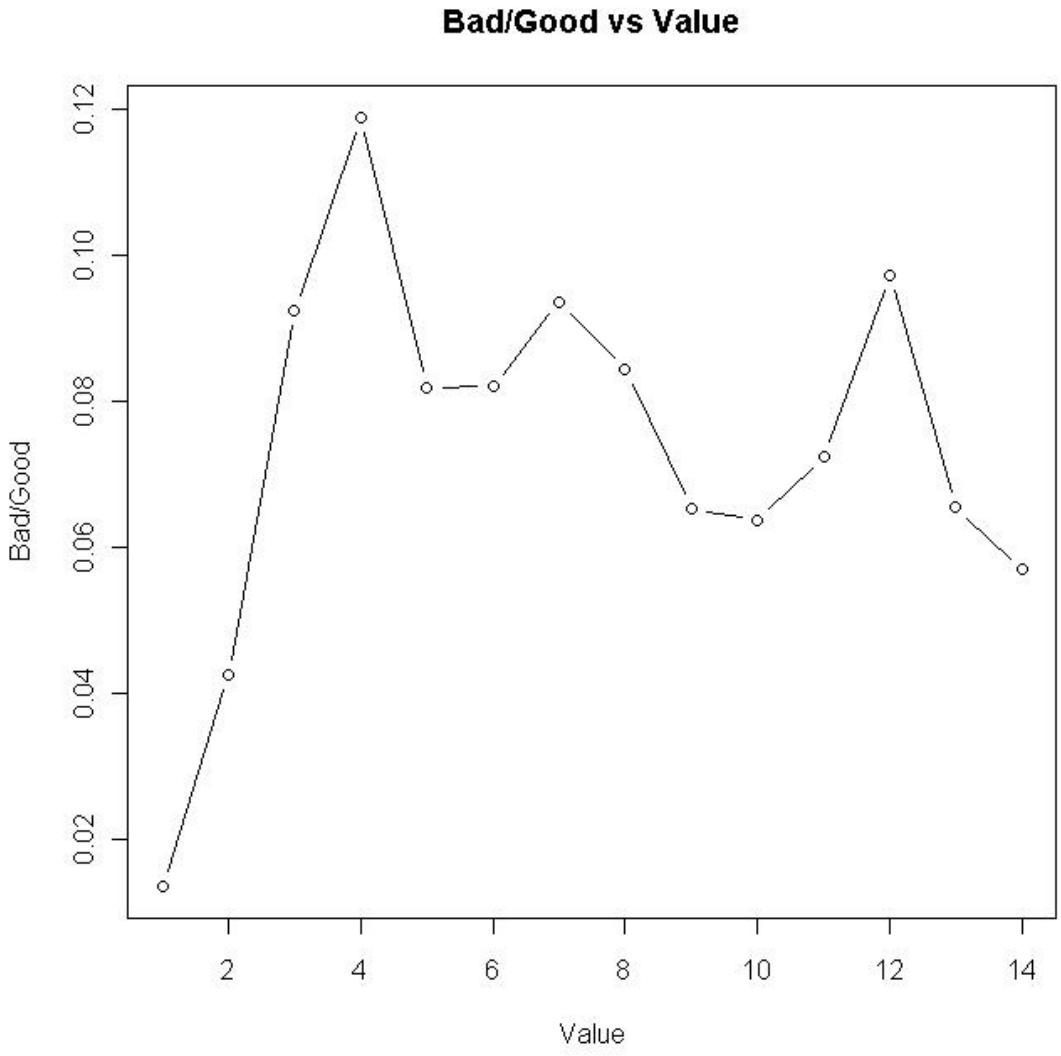


Figure 2.1: Bad/Good Ratios before binning

final.jpg

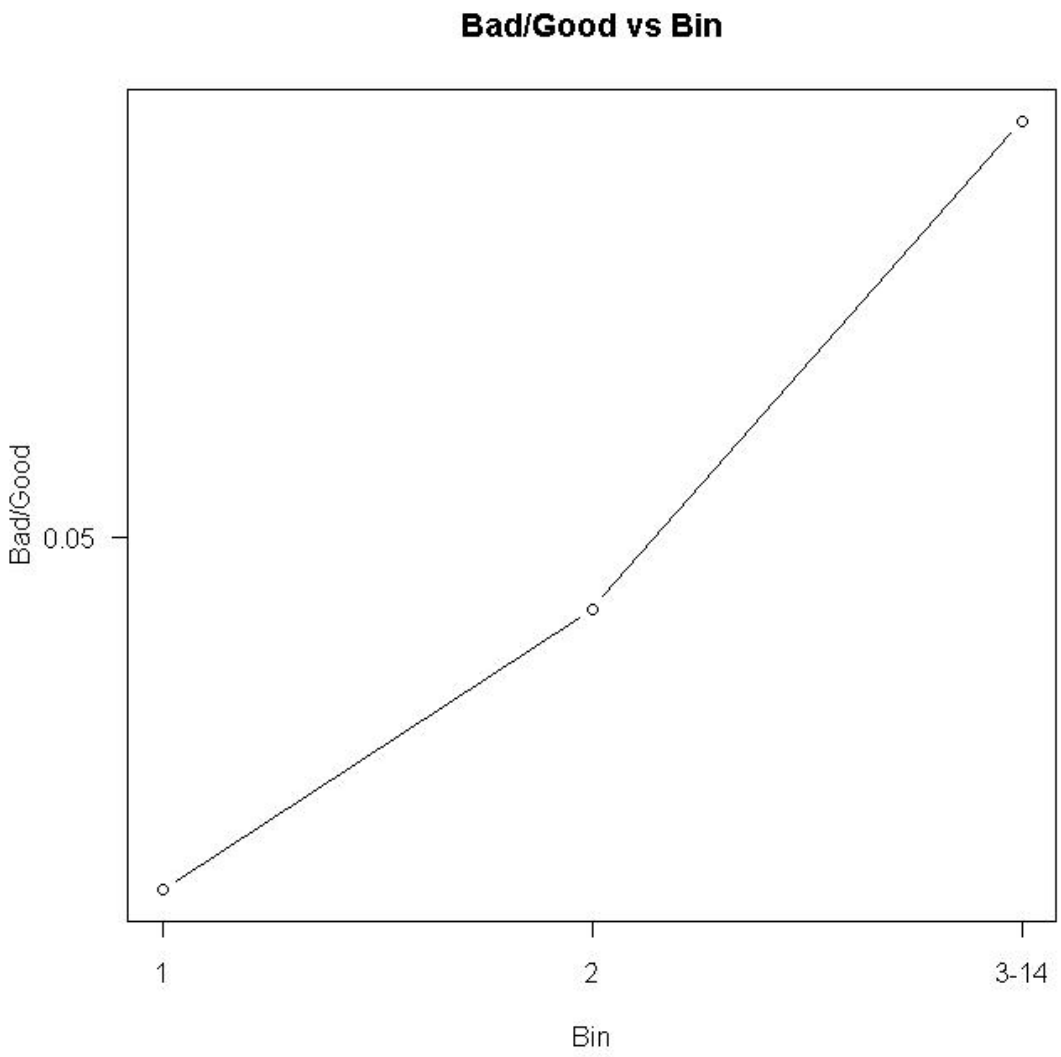


Figure 2.2: Bad/Good Ratios follows linear trend after binning

Table 2.2: After combining just one bin

Bin	Bads	Goods	Total	Ratio	χ^2 stat	Broken	min χ^2
1	243928	17946804	18190732	0.0136	204832.78		
2	363264	8537493	8900757	0.0425	49127.14		
3	109380	1181924	1291304	0.0925	2106.08		
4	55615	467417	523032	0.1190	2498.28	**	
5 & 6	30192	368189	398381	0.0820	139.27		
7	12064	128844	140908	0.0936	48.58	**	
8	8291	98221	106512	0.0844	183.73	**	
9	4676	71565	76241	0.0653	1.13	**	*
10	3285	51550	54835	0.0637	21.50		
11	2411	33273	35684	0.0725	84.17		
12	1836	18858	20694	0.0974	100.23	**	
13	1079	16476	17555	0.0655	15.54	**	
14	4190	73499	77689	0.0570			

Table 2.3: Upward Trend Binning

Bin	Bads	Goods	Total	Ratio	χ^2 stat	Broken	min χ^2
1	243928	17946804.46	18190732	0.0136	204832.78		
2	363264	8537492.763	8900757	0.0425	84086.29		
$3-14$	233019	2509815.183	2742834	0.0928			

Based on Table 2.2 we should combine the bins 9 with 10. By continuing in this way we will obtain Table 2.3 which has the Bad/Good ratio plot as in Figure 2.2.

It can be argued that a trend focus is not appropriate in this case. Perhaps the people who are "chronically" late for their payments are just "lazy"; their risk profile may not necessarily follow an increasing trend as the value of the variable goes up. If we apply a turning point focus on the frequencies instead, then the resultant bin's bad/good ratio plot will be as in Figure 2.3.

2.3 ABBA hill-climbing optimisation extension

The ABBA algorithm, as described in the previous section, is a simple greedy algorithm with no attempt at optimisation. It is known that there are well understood and easy to implement generalised optimisation algorithms designed for heuristics based algorithms such as ABBA. The simplest of these is hill-climbing.

Remark 2.3.1. Generalisation of the hill-climbing algorithms such as simulated-annealing, introduced in [6], and the Great Deluge, introduced in [12], are also applicable here. □

In this context hill-climbing is simply the practice of randomly making adjustments one at a time to the resultant bins to see if your binning could be "improved"; if the adjustment does improve the binning then it keeps the changes and the process

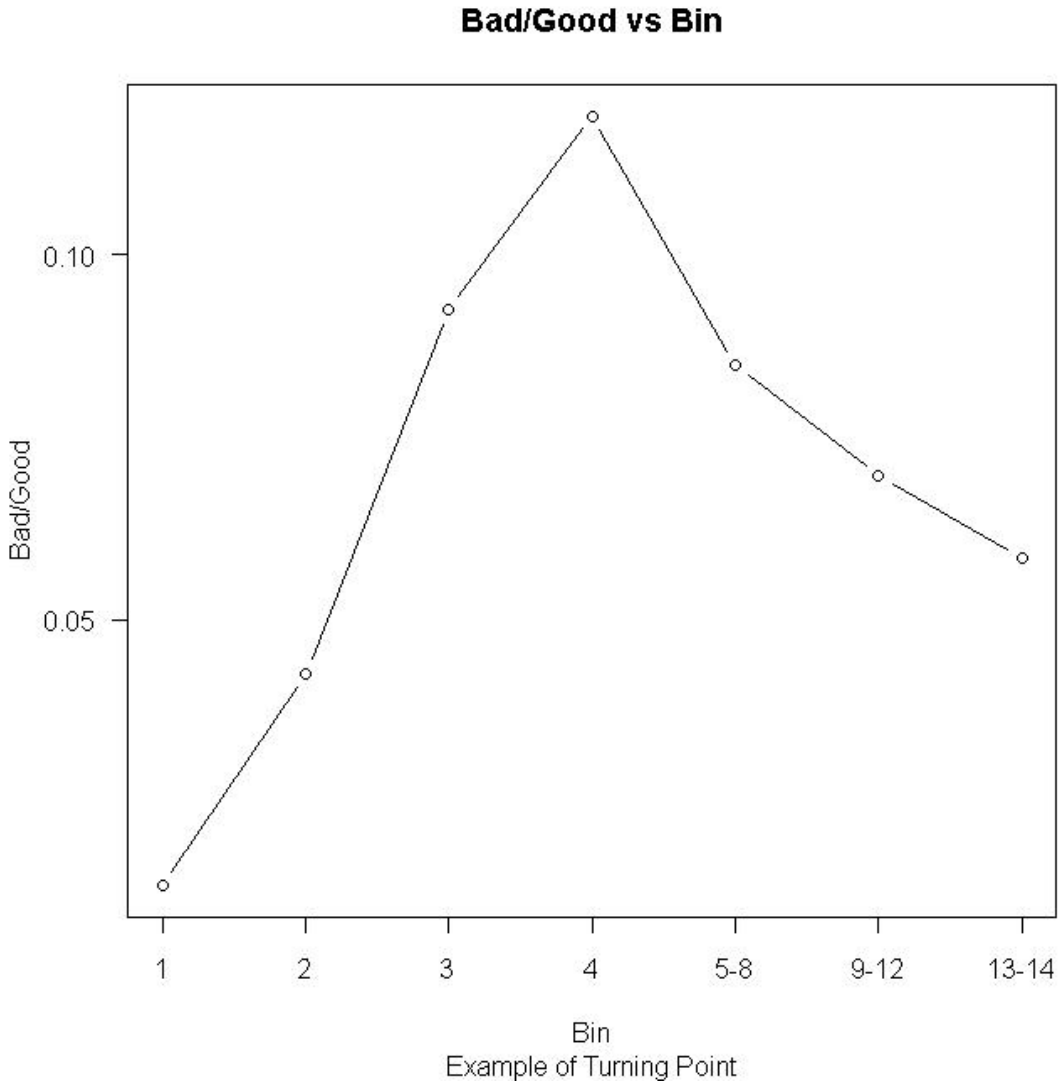


Figure 2.3: Turning point focus

is repeated until no further improvement can be made. At that point the resultant binning is at a "local maximum". The steps taken to reach that point can be thought of as climbing a hill, hence the name.

Suppose we have an ordered-set of bins, $C = [D_1, D_2, \ldots, D_{|C|}]$. The simplest way of making a small adjustments to the binnings is to adjust the boundaries of the bins. That is suppose for some j we have $D_j = [C_{u_{i_{j,1}}}, C_{u_{i_{j,2}}}, ..., C_{u_{i_{j,n}}}]$ and $D_{j+1} = [C_{u_{i_{j+1,1}}}, C_{u_{i_{j+1,2}}}, ..., C_{u_{i_{j,m}}}]$ for some n and m. Let $q < m$, then putting q values from D_{j+1} into D_j to form two new bins is the equivalent of adjusting the boundaries. The new two bins will be

$$D_j^{(1)} = \{C_{u_{i_{j,1}}}, C_{u_{i_{j,2}}}, ..., C_{u_{i_{j,n}}}, C_{u_{i_{j+1,1}}}, C_{u_{i_{j+1,2}}}, ..., C_{u_{i_{j,q}}}\}$$

and

$$D_{j+1}^{(1)} = \{C_{u_{i_{j+1,q+1}}}, C_{u_{i_{j+1,q+2}}}, ..., C_{u_{i_{j,m}}}\}$$

We write the above operation in the following notation

$$[D_j^{(1)}, D_j^{(1)}] = [D_j \leftarrow^q D_{j+1}]$$

and if we are putting q values from the end of D_j into D_{j+1} then we write

$$[D_j^{(1)}, D_{j+1}^{(1)}] = [D_j \rightarrow^q D_{j+1}]$$

2.3.1 The Optimisation

The optimisation we will describe below is applied after the simple ABBA algorithm has terminated. In essence it tries to randomly select bins and adjust the boundaries. Let C^* be the result of applying the simple ABBA on C. Let D^* be some adjustment of C^*, the optimisation algorithms only considers D^* as valid if $f(D^*) = f(C^*) = \emptyset$. This way the optimisation will not undo any of the work done in the simple ABBA algorithm.

Let $C^* := [D_1, D_2, \ldots, D_{|C^*|}]$ be as defined above and let f be the focus function used to derive C^*. We try to optimise C^* by

1. Set $J =$ some random permutation of $[1, 2, ..., |C^*| - 1]$
2. For each $j \in J$
   - create C^{*q} by replacing $[D_j, D_{j+1}]$ with $[D_j^{(1)}, D_{j+1}^{(1)}] = [D_j \leftarrow^q D_{j+1}]$ and compute $l_q = i(D_j^{(1)}, D_{j+1}^{(1)})$ for all possible q
   - find the q such that l_q is minimised subject to $f(C^{*q}) = \emptyset$

- create C^{**s} by replacing $[D_j, D_{j+1}]$ with $[D_j^{(1)}, D_{j+1}^{(1)}] = [D_j \to^s D_{j+1}]$ and compute $r_s = i(D_j^{(1)}, D_{j+1}^{(1)})$ for all possible s
- find the s such that r_s is minimised subject to $f(C^{**s}) = \emptyset$
- if $l_q > i(D_j, D_j)$ or $r_s > i(D_j, D_j)$ then continue else go to the next j
- if $l_q \geq r_s$ then set $C^* = C^{*q}$
- else set $C^* = C^{**s}$

3. if C^* was updated in the last step then repeat from Step 1, else terminate and return C^*

It can be seen that when the algorithm terminates at 3 it is not possible to find a simple adjustment of the boundaries that improves the binning or else the algorithm would not have terminated.

2.4 The Great Deluge Extension of ABBA

The ABBA algorithm can be extended by using a more sophisticated optimisation method. We shall use the Great Deluge [12] as an example of the possible extensions. Suppose there is a measure m of how well the overall binning is, e.g. we may use any of the measures discussed in Section 3.3.

Design a subjective penalties function p so that any binning with undesirable properties gets penalised. For example suppose there are n pairs of bins where the trend is broken in a trend focus binning; we can assign $p(C^*) = 10n$ as a penalty function.

Again let C^* be the result of applying the simple ABBA on C. Choose some small m^* such that $m(C^*) - p(C^*) > m^*$ initially. Also choose a tightening parameter α ¿ 1. The proposed Great Deluge algorithm is as follows:

1. Let A be the set of all possible random adjustments that can be made to C^*; e.g. all possible boundary adjustments
2. For each $a \in A$
   - Make the adjustment a to C^*;
   - Check $m(C^*) - p(C^*) > m^*$ is satisfied
     - if YES
       * Keep the changes
       * Set $m^* \leftarrow \alpha m^*$
       * Go to Step 1
     - if NO then discard the changes and go to next a
3. return C^*

We can think of successive iterations of C^* as a random walk by a blind person on hilly terrain, while m^* represents rising water levels from a great deluge. The blind person would go anywhere that doesn't get his feet wet as the water level rises.

The hill-climbing optimisation is guaranteed to find a local maximum in a neighbourhood of binnings. However the local maximum binning may not be the global maximum. The Great Deluge addresses that issue by introducing the subjective

parameters m^* and α. By choosing appropriate m^* and α the Great Deluge optimisation will often find more optimal solutions than just a simple hill-climb [12].

Remark 2.4.1. This particular extension was not implemented in this project. □

Chapter 3

Modelling and Scorecard Build

3.1 Data

The data was taken from one of the consumer retail portfolios (e.g. credit cards, personal loans, mortgages etc) from an Australian bank. There are a total of 24 months of data. Each observation contains the Month, ID, Bad Outcome, Time to Event, and other attributes associated with the account. The data is organised as in Table 3.1.

The ID variable is a unique identifier of each individual account and the Month variable indicates the month from which the data was extracted. Month can take values from 1 to 24 but the Time to Event variable can take values from 1 to 35. This is due to the fact that there was actually 36 months of raw data available for use. However the data was originally prepared for a Binary Logistic Regression model so the target in that scenario requires 12 months of data to create. Hence the latest 12 months of data cannot be used to build the model as the outcome variable cannot be created. To ensure that the results presented here is comparable to that of the BLR the author chose to restrict the survival analysis modelling dataset to 24 months of data.

3.2 Binning

Let N be the set of numeric variables in the dataset. Each attribute $n \in N$ was binned three times each time with a different focus function. All bins were created using the Pearson's χ^2 Information Loss. The hill-climb optimisation was applied to all binnings. The three focus functions used were

Table 3.1: Example Data

Month	ID	Bad Outcome	Time to Event	Var 1	Var 2	...	Var n
1	1	0	10	100	-50	...	0.1
2	1	0	9	100	1000	...	0.11
3	1	0	8	200	500	...	0.1
...	...	...	...	...	...	...	...
1	2	0	6	100	-50	...	0.1
2	2	0	5	100	1000	...	0.11
3	2	0	4	200	500	...	0.1
4	2	0	3	200	500	...	0.1
...	...	...	...	...	...	...	...

1. The upward trend focus with the Pearson's χ^2 focus

$$f_T \cup f_P(\cdot|T^*)$$

2. The downward trend focus with the Pearson's χ^2 focus

$$f_F \cup f_P(\cdot|T^*)$$

3. The turning point focus with the Pearson's χ^2 focus

$$f_{TP} \cup f_P(\cdot|T^*)$$

As suggested before, $T^* = P(\chi_1^2 < 1 - \epsilon) \approx 68.76325$ where ϵ is the negative machine epsilon on a Windows XP 32 bit machine.

Remark 3.2.1. There are some categorical attributes available in the dataset, however the categorical attributes were not considered in this project. □

Remark 3.2.2. The dataset described in this chapter will not be made available. However a subset of the data with a heavily skewed distribution will be made accessible through the web for research purposes only. Contact the author for details. □

3.3 Binning Assessment and Selection

We aim to have a subjective way of selecting the right binning for each variable/attribute. The methodology being employed in this model build is to compute a number of well-established measures of the three binnings for each variable, and select the binning with the largest number of top ranking assessment statistics. The popular binning measures we will use are information value, Somer's D concordance statistics and χ^2 statistics.

Consider an arbitrary numerical attribute v. Let g be the total number of goods, and b be the total number of bads, and $C = [(b_1, g_1), \ldots, (b_{|C|}, g_{|C|})]$ be a binning of v. We have the following definitions

Definition 3.3.1. The *information statistics*, F, is defined as

$$F := \sum_{(b_j,g_j)\in C} \left(\frac{g_j}{g} - \frac{b_j}{b}\right) \log\left(\frac{g_j b}{b_j g}\right)$$

Definition 3.3.2. The *Somer's D concordance statistics*, D, for a binning with upward bad/good ratio trend is defined as

$$D := \sum_{(b_i,g_i)\in C} \left(\frac{(\sum_{j<i} b_j)g_i - (\sum_{j<i} g_j)b_i}{bg}\right)$$

Table 3.2: The Expected trend for each variable

Variable	Trend
Var 1	Up
Var 2	Up
Var 3	Down
Var 4	Up
Var 5	Up
Var 6	Down
Var 7	Down
Var 8	NA

Remark 3.3.3. The bins needs to be sorted in a order such that the bad/good ratio is trending upwards before applying Somer's D concordance statistics. □

Definition 3.3.4. The χ^2 *statistics*, s^2, is defined as

$$s^2 := \sum_{(b_j,g_j)\in C} \left(\left(\frac{g_j - \hat{g}_j}{\hat{g}_j} \right)^2 + \left(\frac{b_j - \hat{b}_j}{\hat{b}_j} \right)^2 \right)$$

where $\hat{b}_j = \frac{(g_j+b_j)b}{g+b}$ and $\hat{g}_j = \frac{(g_j+b_j)g}{g+b}$.

Remark 3.3.5. A larger information statistics is better; a larger Somer's D concordance statistics is better; and a larger χ^2 statistics is better, see ([1],p132-136). □

As well as having the above 3 measures of binning, one additional ad hoc measure was added. The measure is the AIC from the binary logistic model using b_j as the number of events and $b_j + g_j$ as the number trials with the sole explanatory variable v as a factor with the levels corresponding to the binning.

Together we have 4 measures for each of the 3 binning types. The number of top ranking measures each binning type has is counted. The binning type with the most number of top measures is chosen as the binning for that variable. There were about 235 numeric variables in consideration, and all the selected binning type had either 3 or 4 measures in which it was the top ranking binning. This suggests that the proposed binning assessment methodology is sound.

Remark 3.3.6. The binning methods that were selected were all trend binnings. The expected trend is shown in Table 3.2. □

3.4 Variable Selection

The variable selection process begins once the binning method was selected for the variables. The variable selection methodology being employed is rather conventional. Each of the binned numeric variable is fitted, one at a time, as a factor using the selected binning into a Cox Proportional Hazard Model. Their Bayesian Information Criterion (BIC), introduced in [13], are recorded. The BIC for a model is

$$BIC := -2 \log L + k \log n$$

where L is the likelihood of the model, k is the number of parameters being estimated, and n is the number of observations in the model.

Remark 3.4.1. It was decided that BIC should be used to rank the variables instead of $AIC := -2 \log L + 2k$ as AIC only penalises the number of parameters in the model, but BIC also corrects for sample size. With such a large sample size in our modelling dataset it seems more sensible to take that into account as well. □

It must be noted that it is difficult to employ conventional log-likelihood tests to perform the variable selection. For example, the top two variables by BIC (which by information not available to the reader) are quite closely related. When both are fitted as factors the resultant $-2 \log$ likelihood is 166556.9 and fitting only the top variable yields a $-2 \log$ likelihood of 166614.9. The difference in degrees of freedom is 3 and the $-2 \log$ likelihood difference is 58.3 which yields a p-value of $P(\chi^2_3 > 58.3) = 1.356248 \times 10^{-12}$. This p-value would have lead many to conclude that the second variable added significantly to the model. This effect is most likely due to the fact that we have a large amount of data available.

In view of the above difficulties, the variables were selected using a combination of BIC, business knowledge and common sense. Firstly we order the variable by their BIC from low to high. A lot of the variables that came near the top of the list were quite similar and it makes sense to only include one variable from each type. For example one of the variable is "Number of missed payments in the last 12 months" and it must be highly correlated with "Number of missed payments in the last 6 months"; in this case the variables with the smaller BIC is selected. We work down our list and include variables that are not closely related to the included variables using our business knowledge. Seven numeric variables were selected this way. One categorical variable well-known in the business to be predictive is also included in the model. In total 8 variables were selected to build the survival model.

3.5 Fitting the model

The model being fitted is the Cox Proportional Hazard Model with Month as the strata variable and Bad Outcome is used as the censoring variable where Bad Outcome = 0 implies that the account is right censored. Each of the selected variables was fitted as a factor with the selected binnings. And the levels with the highest frequencies were chosen as the reference for each factor.

3.5.1 Ties Handling

One of the assumptions of the proportional hazard model is that T_a, the survival time, is continuous. Hence it is not possible to have tied survival times. In our data the censored time variable can only take integer values from 1 to 35. Clearly the number of ties in our data is enormous. To address this issue the partial-likelihood is modified by taking into account the ties. The most popular method for ties handling are Breslow [16], Efron [15], exact, and discrete.

Here Breslow is known to be the fastest, Efron is slower but produces results closer to exact, and discrete replaces the continuous hazard with a discrete hazard function and the underlying model is

$$\frac{h(t|\mathbf{x}_a)}{1-h(t|\mathbf{x}_a)} = \frac{h_0(t)}{1-h_0(t)}\exp(\mathbf{T}(\mathbf{x})'\boldsymbol{\beta})$$

where $h_0(t)$ is the discrete baseline hazard and $h(t|\mathbf{x}_a)$ is the discrete hazard.

The choice of ties handling method is only a technical issue and it does not affect the scorecard build significantly as the parameter estimates from all 4 methods are similar. See Table 3.10 for details.

Remark 3.5.1. Our model aims to predict the probability of default in the future given the current month's attributes, so it is sensible to use the Month variable as the strata variable. If we model without using Month as the strata variable then the AIC and BIC respectively are 1568568.6 and 1231412.0. This compares with AIC and BIC of 1162000.4 and 1162556.4 respectively when using Month as the strata variable. This result is highly favourable to the strata approach. □

3.5.2 The Stratified Model

Let $\mathfrak{M}$ be the set of possible values of the Month variable. The model being fitted can be thought of as

$$\frac{h_m(t|\mathbf{x})}{h_{0,m}(t)} = \exp(\mathbf{T}(\mathbf{x})'\boldsymbol{\beta})$$

over all possible stratum $m \in \mathfrak{M}$ where h_m is the hazard function of the stratum m, $h_{0,m}$ is the hazard of the "average" account in stratum m, $\mathbf{x}$ is the data, $\mathbf{T}$ is the binning of $\mathbf{x}$, and $\boldsymbol{\beta}$ is the vector of parameters.

The parameter estimates, $\hat{\boldsymbol{\beta}}$, are found by maximising

$$L(\boldsymbol{\beta}) = \sum_{m\in\mathfrak{M}} L_m(\boldsymbol{\beta})$$

where $L_m(\boldsymbol{\beta})$ is the partial likelihood of stratum m.

Remark 3.5.2. The software used was SAS 9.1.3 and proc tphreg was used to fit the model. The actual code resembles

Table 3.3: Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	1161870.4	1498928.8
AIC	1162000.4	1499058.8
SBC	1162556.4	1162590.2

Table 3.4: Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	69541.5137	65	<.0001
Score	111803.615	65	<.0001
Wald	82140.9906	65	<.0001

Table 3.5: Type 3 Tests

Effect	DF	Wald Chi-Square	Pr > ChiSq
Var 1	3	6738.3246	<.0001
Var 2	10	1615.1616	<.0001
Var 3	18	4386.3066	<.0001
Var 4	10	761.1805	<.0001
Var 5	4	5948.7105	<.0001
Var 6	10	6843.7363	<.0001
Var 7	9	2356.3012	<.0001
Var 8	1	2401.7285	<.0001

```
/* SAS Version 9.1.3 code */
/* For SAS Version 9.2 or higher substitute tphreg with phreg */
proc tphreg data = xxx;
   class var1 var2 var3 var4 var5 var6 var7 var8;
   weight weight;
   strata month;
   model time_to_event*bad_outcome(0) = var1 var2 var3 var4
           var5 var6 var7 var8;
 run;
```

□

3.5.3 Modelling Output

The first model-fit's outputs from SAS are shown in Table 3.3 to 3.5. The outputs indicate that the variables included are all highly significant.

Remark 3.5.3. In Table 3.3 SBC is BIC. □

The (all important) parameter estimates are shown in Table 3.6 to 3.7. It can be seen that some of the parameter estimates are not significant. However, including non-significant parameter estimates in our model does not necessarily reduce the

Table 3.6: Analysis of Maximum Likelihood Estimates

Variable	Bin	Estimate	StdErr	ChiSq	ProbChiSq
Var 1	001.Low-1	-0.12795	0.04047	9.9948	0.0016
Var 1	002.1<-2	0.85411	0.01132	5691.1152	<.0001
Var 1	003.2<-14	0.93143	0.01473	3999.9703	<.0001
Var 2	002.0<-11	0.34012	0.02504	184.552	<.0001
Var 2	003.11<-19	0.45698	0.0286	255.2541	<.0001
Var 2	004.19<-27	0.60543	0.02816	462.2151	<.0001
Var 2	005.27<-39	0.65774	0.02938	501.1093	<.0001
Var 2	006.39<-56	0.70959	0.02916	592.272	<.0001
Var 2	007.56<-79	0.78345	0.0299	686.6614	<.0001
Var 2	008.79<-131	0.83353	0.02777	901.1737	<.0001
Var 2	009.131<-184	0.99099	0.03224	944.646	<.0001
Var 2	010.184<-252	1.14494	0.03861	879.4453	<.0001
Var 2	011.252<-332	1.38119	0.05805	566.0596	<.0001
Var 3	001.Low–174	0.6244	0.02281	749.3416	<.0001
Var 3	002.-174<–89	0.26671	0.03475	58.9069	<.0001
Var 3	003.-89<–1	0.01266	0.02745	0.2126	0.6447
Var 3	004.-1<-8	0.01551	0.03887	0.1592	0.6899
Var 3	005.8<-112	-0.00936	0.02362	0.1572	0.6918
Var 3	006.112<-299	0.01654	0.02348	0.4966	0.481
Var 3	007.299<-733	-0.0667	0.02092	10.1659	0.0014
Var 3	008.733<-858	-0.12832	0.03081	17.3449	<.0001
Var 3	009.858<-1185	-0.18829	0.02418	60.624	<.0001
Var 3	010.1185<-1613	-0.22619	0.02399	88.9133	<.0001
Var 3	011.1613<-2047	-0.22377	0.02564	76.1681	<.0001
Var 3	012.2047<-2545	-0.29583	0.02702	119.8801	<.0001
Var 3	013.2545<-3913	-0.36163	0.02355	235.7831	<.0001
Var 3	014.3913<-6693	-0.46249	0.02415	366.7359	<.0001
Var 3	015.6693<-11385	-0.56108	0.02719	425.7618	<.0001
Var 3	016.11385<-26572	-0.66266	0.02789	564.7209	<.0001
Var 3	017.26572<-306749	-0.79341	0.03216	608.6595	<.0001
Var 3	018.306749<-1858163	-1.71271	0.23412	53.5177	<.0001
Var 4	002.14<-41	-0.11094	0.0665	2.7826	0.0953
Var 4	003.41<-81	-0.12751	0.04381	8.4692	0.0036
Var 4	004.81<-282	0.0032	0.02391	0.0179	0.8936
Var 4	005.282<-514	0.15294	0.02433	39.5067	<.0001
Var 4	006.514<-689	0.22676	0.02836	63.9308	<.0001
Var 4	007.689<-746	0.38481	0.04653	68.3947	<.0001
Var 4	008.746<-965	0.51533	0.0338	232.4257	<.0001
Var 4	009.965<-1480	0.53058	0.03489	231.2347	<.0001
Var 4	010.1480<-2252	0.52211	0.0523	99.6704	<.0001
Var 4	011.2252<-7332	1.8083	0.1194	229.3762	<.0001
Var 5	000.NULL	0.28614	0.01198	570.6578	<.0001
Var 5	002.1<-2	0.46079	0.01312	1233.4272	<.0001
Var 5	003.2<-3	0.88384	0.01748	2555.7268	<.0001
Var 5	004.3<-9	1.1073	0.01635	4587.3176	<.0001

Table 3.7: Analysis of Maximum Likelihood Estimates cont.

Variable	Bin	Estimate	StdErr	ChiSq	ProbChiSq
Var 6	000.NULL	0.65799	0.0757	75.5553	<.0001
Var 6	001.Low-6	1.10157	0.0161	4682.967	<.0001
Var 6	002.6<-23	0.85471	0.0572	223.2424	<.0001
Var 6	003.23<-30	0.58939	0.06795	75.2462	<.0001
Var 6	004.30<-39	0.44783	0.04179	114.824	<.0001
Var 6	005.39<-45	0.33285	0.04511	54.4386	<.0001
Var 6	006.45<-50	0.19693	0.04999	15.5221	<.0001
Var 6	008.143<-4147	0.02829	0.01144	6.1115	0.0134
Var 6	009.4147<-10020	-0.88086	0.24755	12.6618	0.0004
Var 6	010.10020<-1482487620	-1.81174	0.57759	9.8389	0.0017
Var 7	000.NULL	1.0057	0.03188	995.1438	<.0001
Var 7	001.Low-1	0.26021	0.01894	188.7222	<.0001
Var 7	002.1<-2	0.1559	0.01153	182.7665	<.0001
Var 7	004.3<-4	-0.25382	0.01406	325.9736	<.0001
Var 7	005.4<-5	-0.3845	0.0201	365.8605	<.0001
Var 7	006.5<-6	-0.43704	0.02992	213.3754	<.0001
Var 7	007.6<-8	-0.64587	0.03933	269.7325	<.0001
Var 7	008.8<-13	-0.78959	0.07892	100.0871	<.0001
Var 7	009.13<-14	-1.57051	0.50019	9.8584	0.0017
Var 8	S	-0.61485	0.01255	2401.7257	<.0001

predictive power of a variable. The problem where the parameter estimates do not follow the trend specified in the binning is more serious, as the business may reject the model based on that.

It can be seen that the trend is broken at a number of places. For example in Var 3 the levels 003.-89<–1 and 004.-1<-8 have parameter estimates that break the expected trend, see Figure 3.1. One would expect the parameter estimate of 004.-1<-8 to be smaller than that of 003.-89<–1 in order for the trend to be satisfied.

To address the above issue for all variables, a process of fine tuning now occurs. The process mimics ABBA in many ways and it has been coded into an automated process. The steps in the process are as follows:

1. Perform a contrast test on each adjacent parameter estimate pair where the expected trend is broken; that is let $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ be two adjacent parameter estimates and we test the hypothesis $H_0 : \boldsymbol{\beta}_1 - \boldsymbol{\beta}_2 = 0$ vs $H_1 : \boldsymbol{\beta}_1 - \boldsymbol{\beta}_2 \neq 0$. This can be done using the contrast statement in SAS.

2. Combine the bins where the p-value is the largest

3. Refit the model with the new bins

4. Repeat from 1 until all expected trends are satisfied

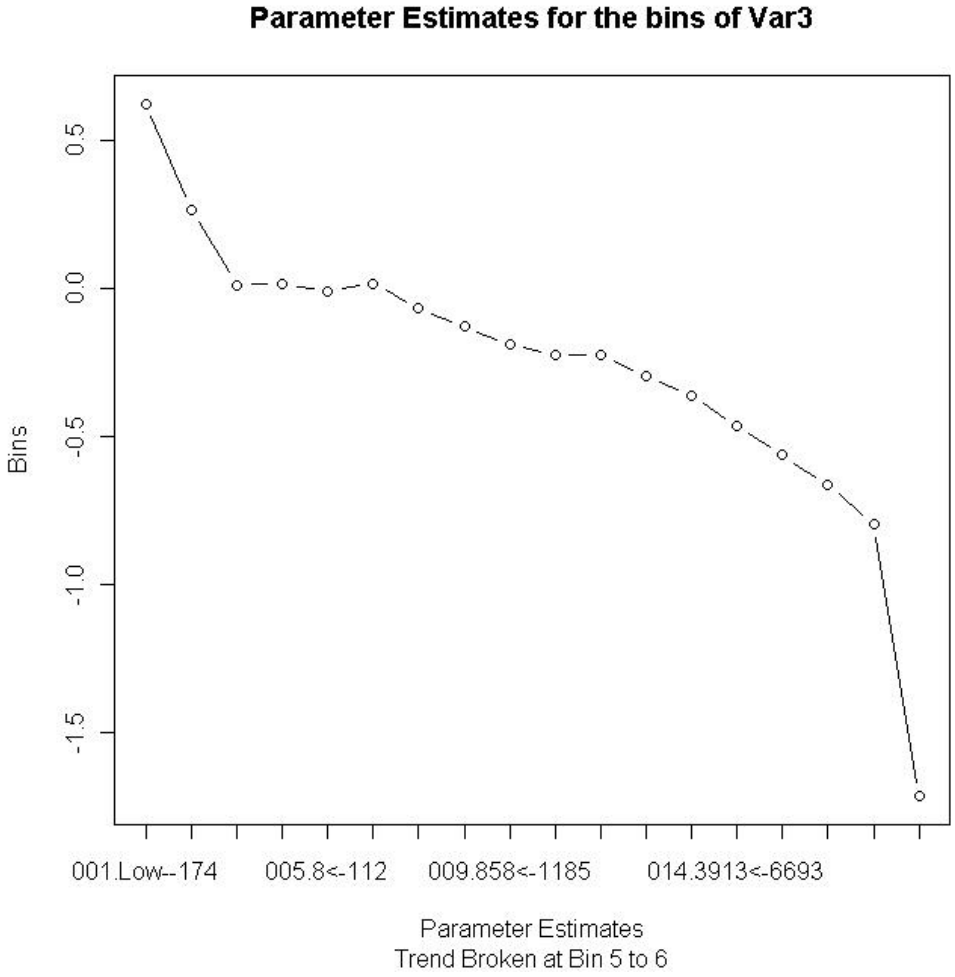


Figure 3.1: Plot of Parameter Estimates of Var 3. Downward Trend broke at Bin 5 to 6

The above process was performed on the model above and the resultant parameters from the final model is presented in Table 3.8 to 3.9. As can be seen the parameter estimates all follow the expected trend. Once this model is turned into a scorecard, it is ready to be taken to the business for approval.

Remark 3.5.4. Care has been taken here so that sensitive information regarding the actual risk profile of the portfolio is not revealed. Hence the censored frequency summaries are not presented. □

3.5.4 Deriving the baseline survival curve

The function $S_0(t) := \exp(-\int h_0(t)dt)$ is referred to as the baseline survival curve. We will produce an estimate of the baseline survival curve, $\hat{S}_0(t)$, using the Kaplan-Meier Estimator on the population with the highest frequencies levels. Those levels were used as the reference levels in the above model so they have an implicit "parameter estimate" of 0. The highest frequency levels were chosen to ensure that the

Table 3.8: Fine Tuned Model - Analysis of Maximum Likelihood Estimates

Variable	Bin	Estimate	StdErr	ChiSq	ProbChiSq
Var 1	001.Low-1	-0.12885	0.04047	10.136	0.0015
Var 1	002.1<-2	0.85557	0.01132	5716.9067	<.0001
Var 1	003.2<-14	0.93287	0.01466	4049.9153	<.0001
Var 2	002.0<-11	0.34087	0.02503	185.5063	<.0001
Var 2	003.11<-19	0.45366	0.0286	251.6008	<.0001
Var 2	004.19<-27	0.59778	0.02815	450.9108	<.0001
Var 2	005.27<-39	0.66095	0.02937	506.5445	<.0001
Var 2	006.39<-56	0.71567	0.02913	603.4079	<.0001
Var 2	007.56<-79	0.79674	0.02985	712.3351	<.0001
Var 2	008.79<-131	0.85436	0.02767	953.069	<.0001
Var 2	009.131<-184	1.01572	0.03216	997.7421	<.0001
Var 2	010.184<-252	1.17076	0.03855	922.4206	<.0001
Var 2	011.252<-332	1.40627	0.05802	587.5009	<.0001
Var 3	001.Low–174	0.6248	0.0228	750.9376	<.0001
Var 3	002.-174<–89	0.26405	0.03473	57.7906	<.0001
Var 3	003.-89<–1	0.01102	0.0248	0.1976	0.6567
Var 3	005.8<-112	0.0003973	0.02044	0.0004	0.9845
Var 3	007.299<-733	-0.06911	0.02091	10.9257	0.0009
Var 3	008.733<-858	-0.12988	0.0308	17.7796	<.0001
Var 3	009.858<-1185	-0.19004	0.02418	61.7869	<.0001
Var 3	010.1185<-1613	-0.22633	0.02398	89.071	<.0001
Var 3	011.1613<-2047	-0.25765	0.02198	137.3423	<.0001
Var 3	013.2545<-3913	-0.36106	0.02355	235.0606	<.0001
Var 3	014.3913<-6693	-0.4623	0.02415	366.3991	<.0001
Var 3	015.6693<-11385	-0.55889	0.02719	422.4537	<.0001
Var 3	016.11385<-26572	-0.6614	0.02789	562.5031	<.0001
Var 3	017.26572<-306749	-0.79235	0.03216	607.0252	<.0001
Var 3	018.306749<-1858163	-1.71376	0.23412	53.583	<.0001
Var 4	002.14<-41	0.05248	0.022	5.6906	0.0171
Var 4	006.514<-689	0.21893	0.02831	59.7987	<.0001
Var 4	007.689<-746	0.37299	0.04646	64.4388	<.0001
Var 4	008.746<-965	0.4999	0.03373	219.622	<.0001
Var 4	009.965<-1480	0.51264	0.03264	246.6905	<.0001
Var 4	011.2252<-7332	1.78828	0.11927	224.8193	<.0001
Var 5	000.NULL	0.28258	0.01196	557.8704	<.0001
Var 5	002.1<-2	0.46202	0.01312	1240.3264	<.0001
Var 5	003.2<-3	0.88347	0.01748	2554.36	<.0001
Var 5	004.3<-9	1.10605	0.01635	4576.6275	<.0001

Table 3.9: Fine Tuned Model - Analysis of Maximum Likelihood Estimates cont.

Variable	Bin	Estimate	StdErr	ChiSq	ProbChiSq
Var 6	000.NULL	0.63139	0.07514	70.6075	<.0001
Var 6	001.Low-6	1.07792	0.01332	6550.4538	<.0001
Var 6	002.6<-23	0.82127	0.05653	211.0575	<.0001
Var 6	003.23<-30	0.55459	0.06739	67.7245	<.0001
Var 6	004.30<-39	0.4115	0.04096	100.9135	<.0001
Var 6	005.39<-45	0.30063	0.04434	45.9619	<.0001
Var 6	006.45<-50	0.15992	0.04929	10.5286	0.0012
Var 6	009.4147<-10020	-0.89605	0.24736	13.1218	0.0003
Var 6	010.10020<-1482487620	-1.83407	0.57751	10.086	0.0015
Var 7	000.NULL	1.00865	0.03187	1001.4854	<.0001
Var 7	001.Low-1	0.26145	0.01893	190.8178	<.0001
Var 7	002.1<-2	0.156	0.01153	183.1085	<.0001
Var 7	004.3<-4	-0.2541	0.01406	326.7309	<.0001
Var 7	005.4<-5	-0.38484	0.0201	366.5044	<.0001
Var 7	006.5<-6	-0.43865	0.02992	214.9677	<.0001
Var 7	007.6<-8	-0.64586	0.03932	269.7459	<.0001
Var 7	008.8<-13	-0.79167	0.07892	100.6204	<.0001
Var 7	009.13<-14	-1.56229	0.50016	9.7567	0.0018
Var 8	S	-0.60987	0.0125	2379.0516	<.0001

survival curve has a narrow confidence interval. Plotting $\hat{S}_0(t)$ as a step function yields Figure 3.2.

3.5.5 Ties Handling Comparison

The method chosen to handle the ties is Breslow [16] as it is the fastest method. More accurate approximations such as Efron [15] exist. However it must be noted that discrete ties handling seems to be the most ideal as our censored time variable is discrete. Unfortunately the weight variable used in the model build is not of integer type, but the discrete and exact methods require the weights to be integers. To investigate the effect of using Breslow we fitted 4 models without weight using different ties handling methods. The results are shown in Table 3.10 to 3.11. This indicates that all four ties handling methods produce estimates that are similar. The author makes the recommendation that Breslow be used when investigating the variable selection and model fit; once the final model has been decided, the data should be refitted using either exact or discrete to produce the most accurate estimates. The running time for these models were Breslow - 1 min 8 seconds, Efron - 1 min 15 seconds, Exact - 2 min and 5 seconds, Discrete - 8 hours.

3.6 Turning the model into Scorecards

Credit Risk Scorecards being built in the industry at the moment all follow a similar convention. Firstly the credit risk score is to be an integer. A linear increase in the score represents an exponential increase in the good/bad odd. Typically a good

Table 3.10: Parameters estimates using all 4 ties handling methods

Variable	Bin	Breslow	Efron	Exact	Discrete
Var 1	001.Low-1	-0.1774	-0.18037	-0.18054	-0.19424
Var 1	002.1<-2	0.90803	0.91006	0.91002	0.91066
Var 1	003.2<-14	0.9796	0.98376	0.98381	0.99073
Var 2	002.0<-11	0.33034	0.33206	0.33205	0.33317
Var 2	003.11<-19	0.46748	0.46939	0.46937	0.47288
Var 2	004.19<-27	0.55258	0.55508	0.55506	0.55913
Var 2	005.27<-39	0.63686	0.63957	0.63953	0.64397
Var 2	006.39<-56	0.69643	0.70014	0.70011	0.70514
Var 2	007.56<-79	0.82506	0.83105	0.83109	0.84046
Var 2	008.79<-131	0.85871	0.86443	0.86447	0.87811
Var 2	009.131<-184	1.01341	1.0202	1.02025	1.03878
Var 2	010.184<-252	1.17349	1.18274	1.183	1.21636
Var 2	011.252<-332	1.32528	1.3396	1.34013	1.37453
Var 3	001.Low–174	0.65205	0.66449	0.66498	0.69877
Var 3	002.-174<–89	0.28172	0.28608	0.28625	0.30888
Var 3	003.-89<–1	0.00776	0.01035	0.01049	0.02117
Var 3	005.8<-112	-0.01008	-0.00776	-0.00761	0.00292
Var 3	007.299<-733	-0.08236	-0.08047	-0.08032	-0.07038
Var 3	008.733<-858	-0.16086	-0.15937	-0.15921	-0.14965
Var 3	009.858<-1185	-0.21787	-0.21667	-0.21652	-0.20782
Var 3	010.1185<-1613	-0.24512	-0.24419	-0.24404	-0.23446
Var 3	011.1613<-2047	-0.28265	-0.28141	-0.28126	-0.27218
Var 3	013.2545<-3913	-0.3864	-0.38521	-0.38505	-0.37628
Var 3	014.3913<-6693	-0.49982	-0.49902	-0.49887	-0.49014
Var 3	015.6693<-11385	-0.56871	-0.56743	-0.56727	-0.55851
Var 3	016.11385<-26572	-0.68961	-0.68841	-0.68823	-0.67873
Var 3	017.26572<-306749	-0.85114	-0.84997	-0.8498	-0.84068
Var 3	018.306749<-1858163	-1.54879	-1.54958	-1.54942	-1.54246
Var 4	002.14<-41	0.09871	0.09698	0.09696	0.09605
Var 4	006.514<-689	0.29609	0.29556	0.29555	0.29905
Var 4	007.689<-746	0.41443	0.41639	0.41646	0.42061
Var 4	008.746<-965	0.54504	0.54803	0.54823	0.57072
Var 4	009.965<-1480	0.54078	0.54832	0.54873	0.562
Var 4	011.2252<-7332	1.64375	1.66571	1.66675	1.76169
Var 5	000.NULL	0.34202	0.34264	0.34259	0.34028
Var 5	002.1<-2	0.4779	0.47882	0.47882	0.48084
Var 5	003.2<-3	0.94021	0.94519	0.94529	0.9552
Var 5	004.3<-9	1.15741	1.16659	1.16681	1.18447

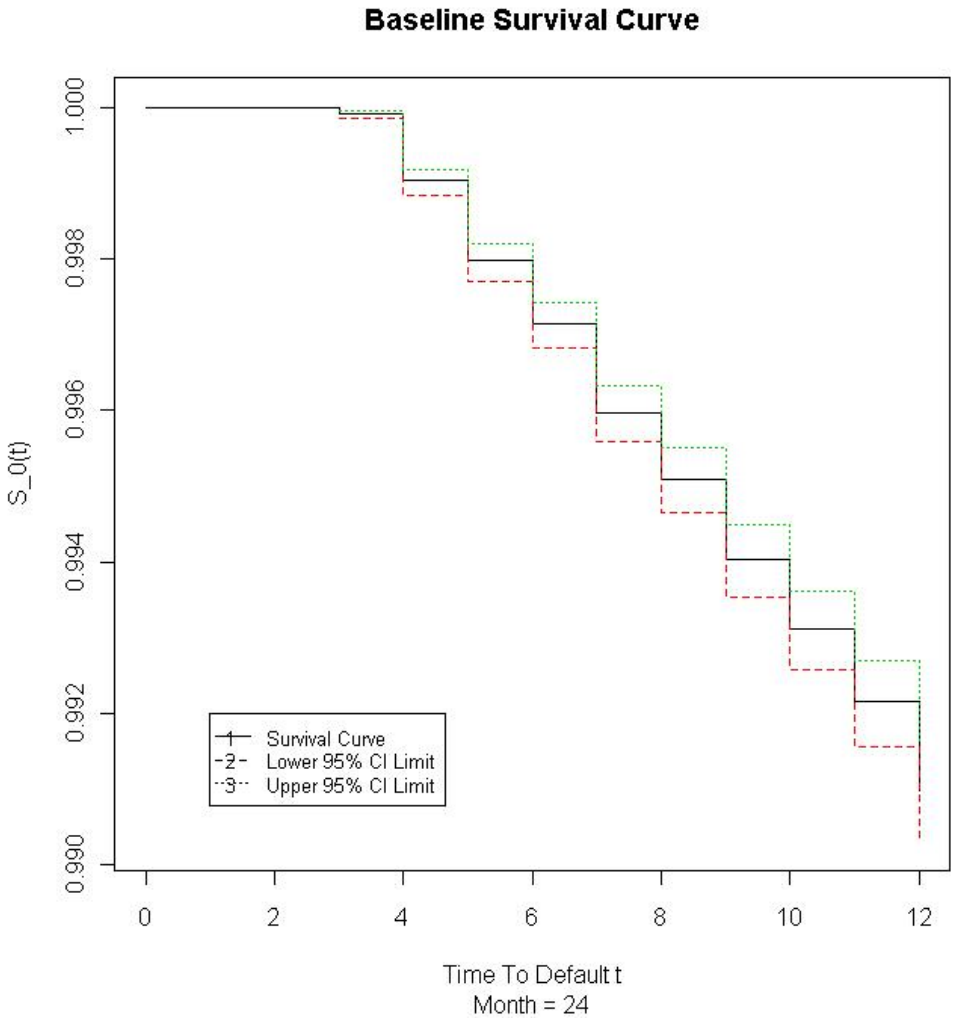


Figure 3.2: Baseline Survival Curve

bad ratio of $g:1$ is set to some score s. Now let w be some fixed integer, then an increase in score of w represents doubling of the good bad odd. So the score $s+w$ has a good bad odd of $2g:1$. This convention comes from scorecards built using Binary Logistic Regression (BLR). In BLR the model being fitted is

$$\log(\frac{p}{1-p}) = \mathbf{x}'\boldsymbol{\beta}$$

where p is the probability of good, $\mathbf{x}$ is the vector of data, and $\boldsymbol{\beta}$ the parameter estimates. Exponentiating both sides yields

$$\frac{p}{1-p} = \exp(\mathbf{x}'\boldsymbol{\beta})$$

The left hand side is now the good/bad odd. The right hand side is the exponential of the raw score. The score, s, is derived from the raw score by scaling it linearly with two parameters a and b, i.e. $s := a\mathbf{x}'\boldsymbol{\beta} + b$, such that the convention is met.

Table 3.11: Parameters estimates using all 4 ties handling methods cont.

Var 6	000.NULL	-0.09094	-0.10028	-0.10052	-0.10332
Var 6	001.Low-6	1.09134	1.10326	1.10357	1.12708
Var 6	002.6<-23	0.89963	0.90646	0.90647	0.92212
Var 6	003.23<-30	0.6313	0.63446	0.63445	0.64142
Var 6	004.30<-39	0.47746	0.48045	0.48048	0.48526
Var 6	005.39<-45	0.34721	0.34934	0.34937	0.35212
Var 6	006.45<-50	0.21906	0.22092	0.22095	0.22289
Var 6	009.4147<-10020	-1.14447	-1.15026	-1.1504	-1.1646
Var 6	010.10020<-1482487620	-1.5816	-1.59772	-1.59811	-1.61783
Var 7	000.NULL	0.88679	0.89658	0.89705	0.91797
Var 7	001.Low-1	0.24714	0.24943	0.24956	0.25891
Var 7	002.1<-2	0.15494	0.15642	0.15647	0.15909
Var 7	004.3<-4	-0.25201	-0.25337	-0.2534	-0.25589
Var 7	005.4<-5	-0.38551	-0.38734	-0.38738	-0.39061
Var 7	006.5<-6	-0.43851	-0.44001	-0.44004	-0.44458
Var 7	007.6<-8	-0.73365	-0.73604	-0.73608	-0.7423
Var 7	008.8<-13	-0.84883	-0.85162	-0.85167	-0.85802
Var 7	009.13<-14	-1.20085	-1.19583	-1.19588	-1.21573
Var 8	S	-0.59912	-0.6012	-0.60123	-0.60707

Hence we get the property of linear increase in the score representing exponentially increasing good/bad odds.

In Survival Analysis the model is

$$S(t) = S_0(t)^{\exp(\mathbf{T}(\mathbf{x})'\boldsymbol{\beta})}$$

taking $\log(-\log)$ we obtain

$$\log(-\log S(t)) = \log(-\log S_0(t)) + \mathbf{T}(\mathbf{x})'\boldsymbol{\beta}$$

We did this to ensure that the score is the sum of linear components. The scorecard format dictates that the scores be derived from a addition/subtraction process so that it is easy to use and implement.

Here the survival function $S(t)$ is an equivalent concept to p in the BLR setting. It can be seen that the score derived in the survival setting can not be interpreted as the same as in the BLR setting. This is because $\log(-\log p) \neq \log \frac{p}{1-p}$. This may cause problems in introducing survival analysis scorecards to the business. If a score from the survival analysis model needs a different interpretation from the traditional scorecards then the business may be reluctant to adopt it. However we will show that the BLR scores and the survival analysis scores have "approximately" the same interpretation by the analytical argument below.

Consider $-\log S(t) = -\log p$. We write $S(t) = p$ for simplicity and we have

$$-\log p = \log p^{-1} = \log\left(1 - (1 - p^{-1})\right) = \log\left(1 - \frac{p-1}{p}\right)$$

Now by Taylor's expansion we get

$$\log\left(1 - \frac{p-1}{p}\right) = -\sum_{i=1}^{\infty} \frac{\left(\frac{p-1}{p}\right)^i}{i}$$

In credit risk p is often very close to 1 (or the banks will be losing big money!) and so $\frac{p-1}{p}$ is small. Hence it makes sense to approximate the left hand side using only the first term in the Taylor expansion yielding

$$-\log p \approx -\frac{p-1}{p} = \frac{1-p}{p}$$

Therefore if we take the log of both sides we will get

$$\log(-\log(p)) \approx \log(\frac{1-p}{p}) = -\log(\frac{p}{1-p})$$

for p close to 1. As can be seen the right hand side is just the negative of the unscaled BLR score.

Remark 3.6.1. This has significant business implications. This suggests that the survival analysis score has the "same" interpretation as the BLR score, so there is virtually no impact to the business if we switch to building survival analysis scorecards instead. Since the scores have the "same" interpretation it will make the business easier to accept such a change. □

This indicates that for p close to 1, which is true in the credit risk setting, the survival analysis score has an approximate linear relationship with the BLR score. So the score as derived in the survival analysis Cox Proportional Hazard model is approximately the same as the one from BLR. We shall aim to make the score as comparable as possible. To achieve this we will scale the score so that the convention described at the beginning of this section is "approximately true".

3.6.1 Deriving the scaling factors

In our model, as before, we aim to have a good bad ratio of $g : 1$ when the score is s and an increase in the score of w should represent a doubling of the good bad odd. We can approximate this relationship in the survival analysis setting by solving a set of simultaneous equations. If we have a good bad ratio of $g : 1$ then $p = \frac{g}{g+1}$ since there are g good accounts out of $g + 1$. In this way we have a set of simultaneous equations

$$s = a\log(-\log\frac{g}{g+1}) + b$$

$$s + w = a \log(-\log \frac{2g}{2g+1}) + b$$

Solve for a and b then we have our scaling parameters. Define $l_x := \log(-\log \frac{x}{x+1})$ then we have

$$a = \frac{-w}{l_g - l_{2g}}$$

and

$$b = \frac{-sl_{2g} + (s+w)l_g}{l_g - l_{2g}}$$

Example 3.6.2. Suppose we wish to have a score of $s = 600$ representing a good/bad ratio of $30:1$ and have $w = 20$ then we have to solve

$$600 = a \log(-\log \frac{30}{31}) + b$$

$$600 + 20 = a \log(-\log \frac{60}{61}) + b$$

Applying the formula above we get $a = \frac{-20}{l_{30} - l_{60}} \approx -29.1978$ and $b = \frac{-600 l_{60} + 620 l_{30}}{l_{30} - l_{60}} \approx 500.2126$ □

The reader may have noticed that if we introduce another equation in the form of

$$600 + 40 = a \log(-\log \frac{120}{121}) + b$$

then we will have an inconsistent system of equations. This highlights the fact that although the survival analysis score is approximately linear with the BLR score they are actually not linearly dependent.

3.6.2 Scaling factors from a linear model

In view of the above we can also derive the scaling factors from a simple linear regression model. The proposed methodology is as follows. Compute the estimated probability of survival 12 months from now, $\hat{S}(12|\mathbf{x})$, for all accounts in the sample and compute their respective BLR scores.

Using the convention in Example 2 their BLR scores, s_{BLR} can be computed by first finding parameters c and d such that

$$600 = c \log \left(\frac{30}{31} \Big/ \frac{1}{31} \right) + d = c \log 30 + d$$

$$620 = c \log \left(\frac{60}{61} \Big/ \frac{1}{61} \right) + d = c \log 60 + d$$

This yields $c = \frac{20}{\log 2} \approx 28.8539$ $d = \frac{-600 \log 60 + 620 \log 30}{-\log 2} \approx 501.8622$. The score for the ith account, $s_{i,BLR}$, is then

$$s_{i,BLR} = c \log \frac{\hat{S}(12|\mathbf{x}_i)}{1 - \hat{S}(12|\mathbf{x}_i)} + d$$

where $\mathbf{x}_i$ is the attributes of the ith account.

Store the BLR scores in a vector y and store the raw survival scores $\log(-\log S_0(t)) + \mathbf{T}(\mathbf{x})'\boldsymbol{\beta}$ for every account in the a vector z. Fit a linear model

$$y = az + b$$

to find the estimates $\hat{a}$ and $\hat{b}$ of a and b respectively and use them as the scaling factors.

In our modelling dataset the scalings factors based on the above methodology are $\hat{a} = -29.2238419$ and $\hat{b} = 499.9061574$ which are not significantly different from the ones derived using just a pair of simultaneous equations. The linear model being fitted has a R^2 of 0.999732.

3.6.3 The Scorecard

The model we have constructed is

$$\hat{S}(t|\mathbf{x}) = \hat{S}_0(t)^{\exp(\mathbf{T}(\mathbf{x})'\hat{\boldsymbol{\beta}})}$$

take $\log(-\log)$ of both sides and we obtain

$$\log(-\log(\hat{S}(t|\mathbf{x}))) = \log(-\log \hat{S}_0(t)) + \mathbf{T}(\mathbf{x})'\hat{\boldsymbol{\beta}}$$

The right hand side is now linear and it can be interpreted as the score. However, to conform to industry standards we scale it using the factors a and b which yields the score s as

$$s := a \log(-\log S_0(t)) + a\mathbf{T}(\mathbf{x})'\hat{\boldsymbol{\beta}} + b$$

rearrange a little we get

$$s := (a \log(-\log S_0(t)) + b) + (a\mathbf{T}(\mathbf{x})'\hat{\boldsymbol{\beta}})$$

Now we have turned the score into two components as below

$$s := \text{Base Score} + \text{Scoring}$$

Remark 3.6.3. The Base Score does not depend on the account's attributes while the Scoring component does. Hence the split. □

Recall that we have fitted a strata variable so each strata has its own baseline survival function $S_0(t)$. Let $\hat{S}_{0,m}(t)$ be the baseline survival function for the stratum

m. We typically choose m to be the latest month. We make this choice as it is reasonable to assume that the more recent the data the more likely it is to be reflective of the current time's risk profile. Let

$$m_{max} = \max_{m \in \mathfrak{M}} m$$

So the 'Base Score" becomes

$$\text{Base Score } := a \log(-\log(S_{0,m_{max}}(t))) + b$$

and

$$\text{Scoring } := a\mathbf{T}(\mathbf{x})'\hat{\boldsymbol{\beta}}$$

is the "Scoring" component.

Using the factors as derived in Example 2. We can construct a scorecard by computing the base score

$$\text{Base Score} := a \log\left(-\log \hat{S}_{0,m_{max}}(12)\right) + b$$

again we choose $t = 12$ so the score is more comparable to the BLR score. The scoring component can be created by multiplying each of the parameters estimates by $a = -29.1978$. The resultant scorecard for our model is shown in Table 3.12 to 3.13. See Section 1.1.1 for how to apply the scorecard.

Remark 3.6.4. This scorecard which was built from a largely automated process is extremely competitive with a scorecard built in the industry by a more manually intensive process. See Table 3.15 for a comparison of the GINIs. The industry built scorecard has a GINI of 0.66 while our model has a GINI of 0.68 using $t = 12$. If a reliable and automated method of variable selection can be found then we have a completely automated process for building scorecards. □

3.7 Model Checking

There are a number of standard ways of checking Proportional Hazard models, however it must be noted that the ultimate measure of a scorecard is how well it rank-orders risk, in other words how good is it at separating bad customers from good. Therefore we shall not make assessment of model fit our primary focus, but rather we shall only use model checking techniques to highlight some deficiencies in our model.

We will use the Cox-Snell residuals to assess our model. The Cox-Snell residuals for the ith account is defined as

$$r_{\mathbf{x_i}} := -\log(\hat{S}(T_i^*|\hat{\mathbf{x}}_\mathbf{i})$$

Table 3.12: Final Scorecard

Variable	**Bin**	**Estimate**	**Score**	**Rounded Score**
Base Score			637.54748854	638
Var 1	001.Low<-1	-0.12885	3.762	4
Var 1	002.1<-2	0.85557	-24.9807	-25
Var 1	003.2<-14	0.93287	-27.2378	-27
Var 2	001.0<-11	0.34087	-9.9526	-10
Var 2	002.11<-19	0.45366	-13.2459	-13
Var 2	003.19<-27	0.59778	-17.4537	-17
Var 2	004.27<-39	0.66095	-19.2984	-19
Var 2	005.39<-56	0.71567	-20.8961	-21
Var 2	006.56<-79	0.79674	-23.263	-23
Var 2	007.79<-131	0.85436	-24.9455	-25
Var 2	008.131<-184	1.01572	-29.6568	-30
Var 2	009.184<-252	1.17076	-34.1837	-34
Var 2	010.252<-332	1.40627	-41.0601	-41
Var 3	001.Low<–174	0.6248	-18.2428	-18
Var 3	002.-174<–89	0.26405	-7.7096	-8
Var 3	003.-89<-8	0.01102	-0.3218	0
Var 3	004.8<-299	0.0003973	-0.0116	0
Var 3	005.299<-733	-0.06911	2.0178	2
Var 3	006.733<-858	-0.12988	3.7922	4
Var 3	007.858<-1185	-0.19004	5.5486	6
Var 3	008.1185<-1613	-0.22633	6.6084	7
Var 3	009.1613<-2545	-0.25765	7.5228	8
Var 3	010.2545<-3913	-0.36106	10.5421	11
Var 3	011.3913<-6693	-0.4623	13.498	13
Var 3	012.6693<-11385	-0.55889	16.3183	16
Var 3	013.11385<-26572	-0.6614	19.3116	19
Var 3	014.26572<-306749	-0.79235	23.1349	23
Var 3	015.306749<-High	-1.71376	50.038	50
Var 4	002.14<-41	0.05248	-1.5322	-2
Var 4	003.41<-689	0.21893	-6.3924	-6
Var 4	004.689<-746	0.37299	-10.8904	-11
Var 4	005.746<-965	0.4999	-14.5959	-15
Var 4	006.965<-2252	0.51264	-14.9679	-15
Var 4	007.2252<-High	1.78828	-52.2139	-52
Var 5	000.NULL	0.28258	-8.2509	-8
Var 5	001.1<-2	0.46202	-13.4901	-13
Var 5	002.2<-3	0.88347	-25.7952	-26
Var 5	003.3<-High	1.10605	-32.2944	-32

Table 3.13: Final Scorecard cont.

Variable	Bin	Estimate	Score	Rounded Score
Var 6	000.NULL	0.63139	-18.4351	-18
Var 6	001.Low<-6	1.07792	-31.473	-31
Var 6	002.6<-23	0.82127	-23.9794	-24
Var 6	003.23<-30	0.55459	-16.1927	-16
Var 6	004.30<-39	0.4115	-12.015	-12
Var 6	005.39<-45	0.30063	-8.7778	-9
Var 6	006.45<-50	0.15992	-4.6693	-5
Var 6	008.4147<-10020	-0.89605	26.1625	26
Var 6	009.10020<-High	-1.83407	53.5508	54
Var 7	000.NULL	1.00865	-29.4503	-29
Var 7	001.Low<-1	0.26145	-7.6338	-8
Var 7	002.1<-2	0.156	-4.5547	-5
Var 7	003.3<-4	-0.2541	7.4191	7
Var 7	004.4<-5	-0.38484	11.2366	11
Var 7	005.5<-6	-0.43865	12.8076	13
Var 7	006.6<-8	-0.64586	18.8577	19
Var 7	007.8<-13	-0.79167	23.115	23
Var 7	008.13<-High	-1.56229	45.6155	46
Var 8	S	-0.60987	17.8068	18

where $\mathbf{x_i}$ is the vector of attributes of the ith account and T_i^* is the censored time for that account.

Let T be a survival time random variable, and let $S(t) := P(T >= t)$ as usual then $Y = -\log(S(T))$ follows the exponential distribution with unit mean, see ([4],p112). Therefore if the fitted model is correct then the Cox-Snell residuals will follow the unit exponential distribution too. Therefore if we plot the residuals against T_i^* then we can examine if it follows an exponential distribution. However this is definitely not ideal as T_i^* can only takes values between 1 to 35 which will invariably result in a very "cramped" looking graph considering the large sample size.

Remark 3.7.1. Of course T_i^* is the censored time not the survival time. A modified version of Cox-Snell residuals ([4],p113) has been proposed to address this issue. However we will not consider the modified Cox-Snell residuals in this thesis. Rather we will place greater focus on measuring the risk rank-ordering properties of our model. □

To address the above issue suppose we think of the residuals $r_{\mathbf{x}_i}$ as coming from a unit exponential distribution, then it has a survival function

$$S^{\#}(t) = e^{-t}$$

snell vs cull hazard.JPG

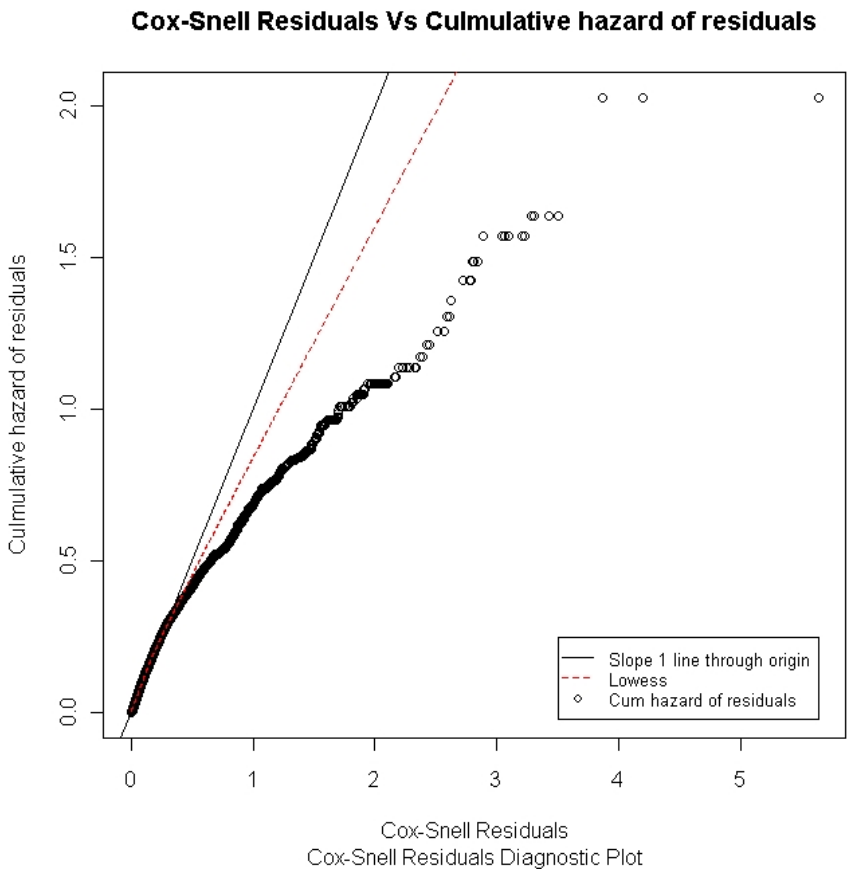


Figure 3.3: Cox Snell Residuals Plot

which implies that

$$-\log(\hat{S}^{\#}(r_{\mathbf{x_i}})) = t$$

Now treating $r_{\mathbf{x}_i}$ as censored survival times if it was from a censored observation then we can estimate $S^{\#}$ with the Kaplan-Meier estimator $\hat{S}^{\#}$. Now the plot of $r_{\mathbf{x_i}}$ against $-\log(\hat{S}^{\#}(r_{\mathbf{x_i}}))$ should be a straight line should the model be correct. Additionally the plot will be more "spread out" due to the larger pool of possible values of $r_{\mathbf{x_i}}$. Figure 3.3 is this plot.

It can be seen that our model seems to perform reasonably well when $r_{\mathbf{x_i}}$ is relatively small and it gets progressively "worse" as $r_{\mathbf{x_i}}$ gets larger. However the lowess curve (using default settings in R) seems to suggest that the majority of our observations are situated in the lower end, as it does not curve towards the residual too much but instead stays closer to the line through origin of slope 1. Finally we stress again that if our model is rank-ordering risk well then we have achieved our purpose. The deviation from the ideal model as indicated here does not necessarily make the model's risk rank-ordering power worse, it simply means that our model does not

Table 3.14: Mock Good Bad Profile by Ordered Score

Score	Goods	Bads	Cum Prop Goods	Cum Prop Bads	Tot Good	Tot Bad
450	0	1	0.00	0.02	22.91	65.04
470	0	1	0.00	0.03	22.91	65.04
476	0	1	0.00	0.05	22.91	65.04
477	0	2.75	0.00	0.09	22.91	65.04
479	0	2	0.00	0.12	22.91	65.04
481	0	3	0.00	0.17	22.91	65.04
483	0	2	0.00	0.20	22.91	65.04
484	0	1	0.00	0.21	22.91	65.04
485	0	2	0.00	0.24	22.91	65.04
486	2.44	2	0.11	0.27	22.91	65.04
487	0	3	0.11	0.32	22.91	65.04
488	2.6	2.42	0.22	0.36	22.91	65.04
489	0	4	0.22	0.42	22.91	65.04
490	4.92	3	0.43	0.46	22.91	65.04
491	2.49	0	0.54	0.46	22.91	65.04
492	0	11	0.54	0.63	22.91	65.04
493	5.04	9.42	0.76	0.78	22.91	65.04
494	2.69	9.45	0.88	0.92	22.91	65.04
495	2.73	5	1.00	1.00	22.91	65.04
...	...	...	...	...	...	...

produce the most accurate PD estimates for all accounts.

3.8 Model Validation

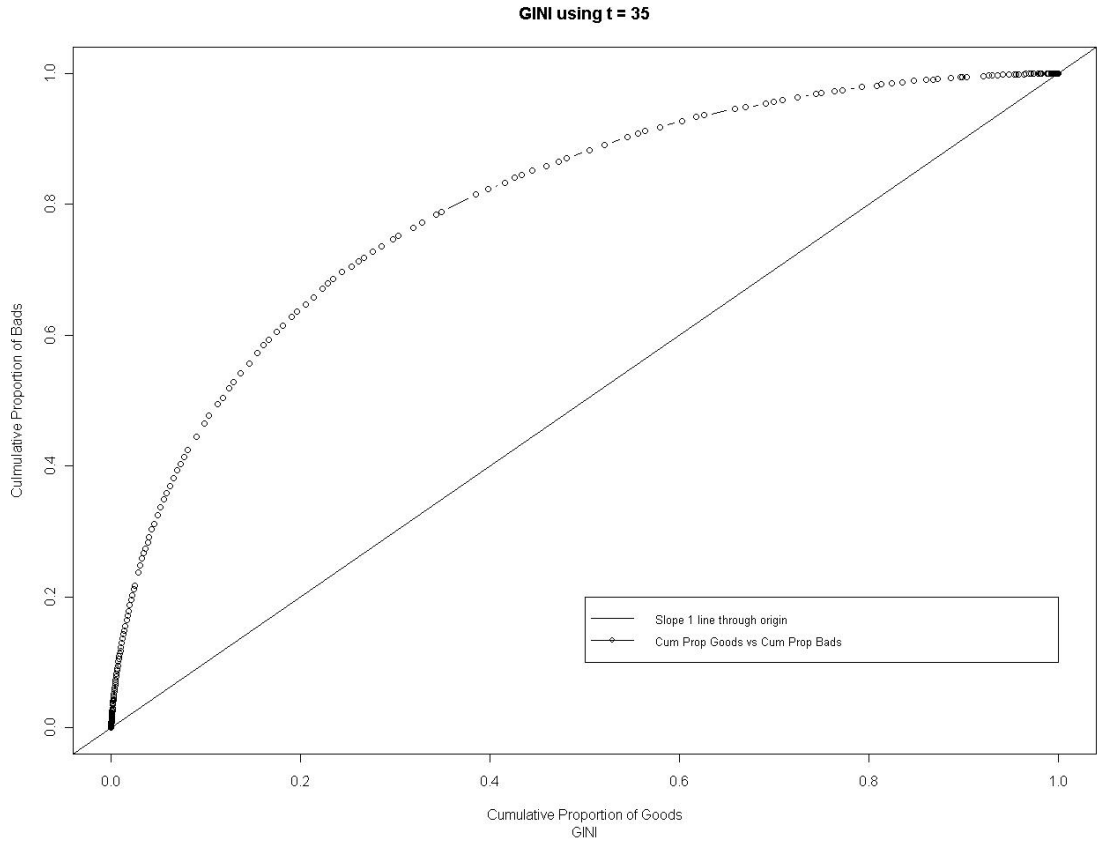
The key measuring yardstick of a risk scorecard is how well it rank-orders risk. The customers with higher scores are suppose to be of lower risk. The most popular measures of the predictive power of a scorecard in the risk rank-ordering sense include the GINI and the KS statistics.

Suppose we scored every account in our dataset. Let s_i be the score of the ith account, and let

$$D_t(i) := \begin{cases} 1 & ; \text{if the } i\text{th account goes bad at month } t \text{ in the future} \\ 0 & ; \text{otherwise} \end{cases}$$

We can produce a frequency of the good bad profile at time t in the future by score, for example see Table 3.14. A GINI Plot can be obtained by plotting the cumulative proportion of goods vs the cumulative proportion of bads, for example see Figure 3.4.

Figure 3.4: GINI Plot Using t =35
t 35.jpg



Remark 3.8.1. The number of goods and bads in Table 3.14 are not whole numbers as weights that are not whole numbers were applied when doing the frequencies. □

The GINI plot of our model using $t = 35$ is Figure 3.4. The GINI coefficient is defined as 2 times the area between the Cumulative Goods vs Cumulative Bad plot and the slope 1 line through origin. Clearly the higher the GINI the better our model. Suppose we scored the accounts randomly, then we would expect our GINI plot to be close to that of the line of slope one, and this will yield a GINI close to 0.

Remark 3.8.2. The best possible GINI is 1. □

We will approximate the area using trapeziums. That is let

$$S := \{s_{(j)} \mid j = 1, 2, \ldots, |S|\}$$

be the ordered set of all possible scores where $s_{(j)} < s_{(j+1)}$ for all j, and let $b_{(j)}$ and $g_{(j)}$be the cumulative proportion of bads and goods, respectively, at score $s_{(j)}$.

Definition 3.8.3. The GINI coefficient, G, is defined as

$$G := \sum_{1}^{|S|-1} \left(b_{(j)} + b_{(j+1)} - g_{(j)} - g_{(j+1)}\right)\left(g_{(j+1)} - g_{(j)}\right)$$

This yields a GINI of ≈ 0.601 from Figure 3.4.

Remark 3.8.4. The ROC curve is often used as well to assess the risk rank-ordering. However it is similar to GINI in definition and it is more common in the industry to look at GINI instead. □

Remark 3.8.5. The GINI requires that the outcome be binary. In survival analysis the outcome is binary with a time element. Therefore we needed to choose a time point t when computing the GINI. Also this is done so that our model can be compared to BLR by GINI. □

We scored our development and holdout sample and produced the Table 3.15 for the GINI using $t = 1, 2, \ldots, 35$. It can be seen that the GINI peaks at $t = 3$ in both samples and drops off towards a plateau. This suggests that the scorecards are very good at distinguishing accounts that are likely to go into default in the first 3 months, and its predictive power degrades over time. Also the GINI is higher in the development sample from $t = 4$ onwards. This suggests some slight overfitting of the model.

Remark 3.8.6. The number of defaults are low for $t = 1, 2, 3$. Hence the reason why the difference in GINI is greatest for those t values. In particular for $t = 1$ there was only one default in the holdout sample, hence the peculiar GINI. □

Remark 3.8.7. The model built using BLR by the business on the exact same data has a GINI of 0.67 and 0.65 on the development and holdout sample respectively. This indicates that the survival analysis scorecards perform just as well as the BLR scorecards. □

Definition 3.8.8. The KS statistics, another popular measure is defined as

$$KS := \max_{j} \left(b_{(j)} - g_{(j)}\right)$$

That is, it is the maximum difference in the cumulative proportion of bads and goods over all possible scores.

The larger the KS the better the rank-ordering. If the scores were randomly assigned then one would expect that the KS statistics will be close to zero. The KS statistics for our model using $t = 12$ is 0.33 which is a significant improvement over

Table 3.15: The GINI on the development and holdout sample using various t

t	GINI Dev	GINI Holdout
1	0.67017	-0.1649
2	0.81096	0.81933
3	0.8595	0.8962
4	0.81247	0.8175
5	0.77693	0.7746
6	0.75627	0.7425
7	0.73893	0.71959
8	0.72414	0.69979
9	0.71225	0.6853
10	0.70145	0.67038
11	0.69025	0.6624
12	0.68101	0.65878
13	0.66774	0.6501
14	0.6549	0.64445
15	0.64677	0.63712
16	0.6403	0.63202
17	0.63498	0.62626
18	0.63123	0.62311
19	0.62672	0.61927
20	0.62334	0.61659
21	0.61991	0.61178
22	0.6171	0.61002
23	0.61476	0.60794
24	0.61282	0.60535
25	0.6113	0.60417
26	0.60925	0.60227
27	0.60801	0.60083
28	0.60666	0.5988
29	0.60578	0.59811
30	0.60493	0.59786
31	0.60407	0.5976
32	0.60361	0.5972
33	0.60318	0.59748
34	0.6028	0.59748
35	0.60265	0.59703

random assignment of scores.

3.9 Comparison with Binary Logistic Regression

A BLR scorecard model was built by the business. We want to be able to perform a comparison of the relative performance of Binary Logistic Regression (BLR) versus Cox Proportional Hazards (CPH) models. To achieve this we fitted a CPH model using the variables and binnings as per in the BLR model. The GINI from that model is listed in Table 3.16. It can be seen that the two methods performed similarly. So there is no evidence that switching to building scorecards using survival analysis techniques will in any way jeopardise the predictive power of the scorecards. On the other hand profit scoring [11] and stress testing with macro economic variables [3] is possible with survival analysis models but not with the traditional BLR models.

3.10 Application Scorecard Reject Inference

So far in this thesis we have discussed how to construct behavioural scorecards. The scorecards we have built are based on customers that have already taken up the loan. There is another type of scorecards that deals with new loan applications. These are called the application scorecards.

In building an application scorecard the population that had their loan applications accepted are not representative of the "through the door" population. This will introduce a bias in our model which was termed "reject bias" in [1]. Hence some reject inference procedure is often applied to try and reduce this bias.

Let R be the set of loan applications that were rejected and let $\mathbf{x}_r$ be the set of attributes associated with the customer $r \in R$ at the time of application. One way to perform reject inference is to assign a good bad outcome based on $\mathbf{x}_r$ to each loan application $r \in R$; and this good bad outcome can be interpreted as their good bad outcome had they been offered the loan. This process is called imputation. Once imputation has been performed we fit the model using all the accounts data including the imputed data. This way the model will be more reflective of the "through the door" population.

As mentioned in ([1],p141) reject inference is an area of "some controversy". This is understandable as there is no reliable way of assigning a good or bad outcome to the rejected loan applications. In the survival analysis setting the matter is complicated further by the fact that a bad outcome needs to be accompanied by a time to default variable.

Let $\mathcal{A}$ be the set of applications including the rejected population R. In [9] it was suggested a parametric survival analysis model be fitted to the population $\mathcal{A}/R$ and call this Model A. From this model a formula of the $(1-\alpha)\%$ confidence interval for the median survival time can be obtained. This median survival time is a function of $\mathbf{x}_r$. Now apply the parameters obtained in Model A to the population R

Table 3.16: The GINI on the development and holdout sample using various t

t	GINI Dev	GINI Holdout	GINI BLR Dev	GINI BLR Holdout
1	0.5247	0.05212		
2	0.73257	0.81934		
3	0.8212	0.85012		
4	0.77899	0.78847		
5	0.74848	0.74548		
6	0.72795	0.71788		
7	0.7113	0.70009		
8	0.69847	0.68486		
9	0.6892	0.67241		
10	0.67947	0.66113		
11	0.67014	0.654		
12	0.66171	0.64871	0.67	0.65
13	0.64883	0.64007		
14	0.63729	0.63614		
15	0.63083	0.63151		
16	0.62417	0.62555		
17	0.61913	0.61976		
18	0.61558	0.6174		
19	0.61124	0.61386		
20	0.60828	0.61152		
21	0.60463	0.60683		
22	0.60166	0.60578		
23	0.59935	0.60242		
24	0.5972	0.59947		
25	0.59538	0.59808		
26	0.59345	0.59612		
27	0.59222	0.59472		
28	0.59131	0.59312		
29	0.5905	0.59232		
30	0.58988	0.59143		
31	0.5889	0.59125		
32	0.58851	0.59112		
33	0.58838	0.59138		
34	0.58814	0.59159		
35	0.588	0.59139		

and let t_r be the lower $(1-\alpha)\%$ confidence interval limit of the predicted median survival time and let t be the median survival time estimate from the whole $\mathcal{A}/R$ population. The paper suggests that we assign a good to the application if $t_r > t$ and a bad otherwise. However this approach does not address how to impute the time to event variable.

In this section we propose a new reject inference scheme using the Cox Proportion Hazard model. The steps involved are as follows:

1. Fit a Cox Proportional Hazard model to the $\mathcal{A}/R$ population using the techniques discussed in the previous sections and call this Model A

2. Choose an α and obtain the formula of the $(1-\alpha)\%$ confidence interval for $\hat{S}(t|\mathbf{x}_r)$ and let $\hat{S}_L(t|\mathbf{x}_r)$ be the lower limit of the confidence interval.

3. Apply Model A to the population R and obtain the $\hat{S}_L(t|\mathbf{x}_r)$ estimates for each account $r \in R$.

4. Let t_{max} be the largest t for which we have an estimate for $\hat{S}_L(t|\mathbf{x}_r)$ and let $w_t := \hat{S}_L(t|\mathbf{x}_r) - \hat{S}_L(t-1|\mathbf{x}_r)$. For each account $r \in R$ we create a set of imputed data points by defining

$$\text{Impute}(r) := \{(x_r, 1, t, w_t) \; | t = 1, 2, \ldots, t_{max}, \; \} \; \cup \; \{(x_r, 0, t, \hat{S}(t_{max}|\mathbf{x}_r)\}$$

5. Refit the model by incorporating all the imputed data points and their survival times from

$$I = \{\cup_{r \in R} \text{Impute}(r) \; \}$$

   Here any $(x_r, o, t, w_t) \in I$ is a weighted data point with attributes $\mathbf{x}_r$, good bad outcome o, time to event t and a weight of w_t.

Essentially we have imputed the time to event and bad outcome using the predicted probability of default as derived from the lower $1-\alpha$ confidence interval limit as weights. In this reject inference scheme the choice of α is subjective. Further research is needed to establish good ways of choosing a value for this parameter.

References

[1] Thomas, Edelman, Crook; Credit Scoring and Its Applications *Society for Industrial and Applied Mathematics* (1983), pp. 431–475.

[2] Stepanova, Thomas; Survival Analysis Methods for Personal Loan Data *Operations Research* Vol. 50, No. 2, March-April 2002, pp. 277-289

[3] Bellotti, Crook; Credit scoring with macroeconomic variables using survival analysis *Journal of the Operational Research Society* 60, 1699-1707 (December 2009) — doi:10.1057/jors.2008.130

[4] Collett; Modelling Survival Data in Medical Research, Chapman and Hall/CRC, (2003)

[5] Anderson; Credit Scoring Toolkit - Theory and Practice for Retail Credit Risk Management and Decision Automation, Oxford University Press (2007)

[6] Kirkpatrick, Gelatt, Vecchi; Optimization by Simulated Annealing . Science. New Series 220 (4598) (1983)

[7] Banasik, Crook, Thomas; Not if but When will Borrowers Default The Journal of the Operational Research Society, Vol. 50, No. 12 (Dec., 1999), pp. 1185-1190

[8] Cox; Regression Models and Life Tables. Journal of the Royal Statistical Society Series B 34 (2): 187?20. JSTOR 2985181 . MR0341758 .(1972).

[9] So Young Sohn, H.W. Shin; Reject inference in credit operations based on survival analysis. *Expert Systems with Applications* 31 (1): 26-29 (2006) 0957-4174

[10] Hand, Adams; Defining attributes for scorecard construction in credit scoring. *Journal of Applied Statistics*, Vol. 27, No 5, (2000) 527-540

[11] Andreeva, Galina and Ansell, Jake and Crook, Jonathan; Modelling profitability using survival combination scores *European Journal of Operational Research* Vol. 183, No 3, (2007) p. 1537-1549

[12] Dueck, Gunter; New Optimization Heuristics The Great Deluge Algorithm and the Record-to-Record Travel *Journal of Computational Physics*, Volume

104, Issue 1, (1993) p. 86-92.

[13] Schwarz; Estimating the dimension of a model Annals of Statistics 6 (2): (1978) 461

[14] R Development Core Team; R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing (2010), ISBN 3-900051-07-0, http://www.R-project.org

[15] Efron; The Efficiency of Cox's Likelihood Function for Censored Data *Journal of the American Statistical Association*, Vol. 72, No. 359 (1977), pp. 557-565 URL: http://www.jstor.org/stable/2286217

[16] Breslow; Covariance Analysis of Censored Survival Data *Biometrics*, Vol. 30, No. 1 (1974), 89-99 http://www.jstor.org/stable/2529620

[17] Kaplan, Meier; Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53, 457-481

[18] Aalen; Nonparametric inference for a family of counting processes, *Statistical Methods in Medical Research*, 3, 227-243