MACHINE LEARNING FOR SURVIVAL ANALYSIS ON CLINICAL DATA

Fotis Paraschiakos
Department of Informatics,
Faculty of Science,
Utrecht University, Netherlands

May 23, 2016

This dissertation is submitted for the degree of MSc Computing Science

Abstract

The usage of Machine Learning in medicine is a new and a very fast moving technology which is getting more and more attention by information technology companies, doctors, patients and scientists. This technology holds promise for several aspects of medicine, including improving diagnosis of disease, early detection of disease and personalized health care.

Currently, experiments with real-world clinical data are necessary to investigate how models based on different statistical analysis methods perform in clinical practice. Previous research has observed and measured the influence of various predictors on survival after a cardiac arrest event, both in the form of biomarkers present in the results of blood analysis and from other types of patient information. This master's thesis project continues this study, trying to find better models using different advanced modeling methods for the prediction of several factors related to disease outcome using a large and comperhensive dataset of clinical data.

List of Tables

List of Figures

1 Introduction

Machine Learning is a new and promising subfield of Algorithmic Data Analysis that is rapidly progressiong over the last years. Due to the available storage space, processing power and the rapid increase of network connectivity, there have been huge advancements in data collection, sharing and processing technologies. If wae also take into account the recent large increase of the volume of data generated from almost all sources, it is possible to implement increasingly complex machine learning methods, something impossible with past technology. Consequently, the field has started to accomplish impressive results in real-world scenarios. Among other sectors, it has shown great promise in clinical data analytics. However, the field is still very young and in order to evaluate how different models perform in scenarios of clinical practice, experiments with clinical data need to be performed.

The aim of this thesis is to compare the performance of several different analysis methods and machine learning techniques for survival analysis based on clinical data. It will test different approaches of tackling the usual issues that may occur in an analysis task like that.

For this purpose, clinical data from heart disease patients treated in the University Medical Center of Utrecht will be used. The data is taken from UPOD (Utrecht Patient Oriented Database). Over the past years, the UPOD group has published several scientific papers in this field. In these papers, conventional epidemiological methods like logistic regression analysis were used to assess associations between hematological parameters and outcome. In these approaches first order models were studied.

The UPOD database contains, on a patient level, time-stamped data on laboratory test results, ICD (International Classification of Diseases)-coded diagnoses, medication orders and medical procedures for all patients treated at the UMC Utrecht. In addition to a database comprising data from the Laboratory Information System, UPOD contains a worldwide unique database with hematology data on automated blood cell analyses performed with Abbott Cell-Dyn Sapphire automated blood cell analyzers used at UMC Utrecht (around 400 unique patients per day).

In the current project, special attention will be paid to the exploration of this hematology data for the prediction of length of survival in a cohort of patients. Previous research has observed and measured the influence of various predictors, both in the form of biomarkers present in the results of blood analysis and from other types of patient information. The size of the database and especially the depth of information present in the hematology data provides for a unique opportunity to add to the current research by developing new predictive models, identifying new biomarkers and comparing the performance of different analysis techniques.

The Current research interest of the UPOD group concerns the prediction of several health outcomes including sepsis, myocardial infarction, fracture healing after severe trauma, survival at admission at the emergency department and outcome of cardiac surgery, based on hematologic parameters. Another related issue is the mortality of patients that have been diagnosed with outcomes of this type. Together with the heamatological information, the cohorts that the group studies contain information about diseases, outcomes of tests and medication provided. The purpose of the thesis is to use advanced data analytic techniques to explore the hematological data to the full potential, while utilizing any other information that may be of value.

One characteristic of the questions that are answered during research on this kind of data is that they apply to subsets of patients. Therefore it is hard to find a large number of homogenous cases from the same hospital that can be examined. If the sample size is small then there is a risk of increasing variation. The most probable sources of variation are proven to be the non-heamatological and environment variables like age, sex and basic clinical measurements. For variables like this, there has been research based on datasets that are orders of magnitude larger than this, so they could be possibly taken into account in our models by using them to calculate prior probabilities.

Generally, an attempt will be made to combine different parts of the database that have not been used together in the past, using techniques not used before on those data, in order to create more complex models than the ones in the past research. Other than finding relevant biomarkers, another goal is to test the different techniques mentioned above in order to figure out which one has the best performance on the specific dataset and provide a roadmap for future research.

.

1.1 Dataset

The dataset was built from 2 different sources. The first one was taken from a cohort in the Utrecht Coronary Biobank (UCORBIO), with patients enrolled from October 2010 until April 2013.

Patients are followed-up for five years, of which three have passed at the moment of writing. The vital status of patients was collected from the hospital administration system, which is strictly updated by research staff. The cause of death was obtained from medical reports from our or other hospitals or institutions. If a patient died at home, the general practitioner involved in the case was consulted to obtain further information on the cause of death.

The variables taken into account for each patient include personal information (like sex, age, weight), information about previous diseases and pre-existing conditions and the data from the Cell-Dyn automated blood analysis machine.

All patients undergoing coronary angiography at the University Medical Centre of Utrecht are asked to participate in this biobank. Hence, this cohort comprises a wide spectrum of heart disease patients and controls.

The study has been approved by the medical ethics committee of the University Medical Centre Utrecht and all patients provided written informed consent.

From the full dataset, several variable selection steps were taken, both manual and algorithm-based in order to deduce which variables are the relevant ones and discard all the others. After descriptive statistics, the first step taken was an initial manual selection of variables. This will be discussed in more detail in the next chapters.

The full list of working variables together with some basic descriptive statistics can be seen on Appendices 1 and 2.

1.2 Related Work

Vanneschi citeVanneschi2011 et al compared techniques of Survival Forest, Suport Vector Machines and Random Forests to predict survival of breast cancer patients using as predictors a well known group of 70 genes that have predictive power over breast cancer survival. They did not follow the traditional Survival Analysis procedure (using the survival function) but just classified the patients in groups of patients survived or not at a given time point. The research showed that the Genetic Programming method outperformed the other ones, while also being the only one that has a way to implicitly perform feature selection.

SurvivalSVM is an algorithm that implements the SVM algorithm for survival models, developed by Van Belle et al [21] and implemented by the same group as the SurvivalSVM R package. It works by reformulating the survival problem into a rank regression one, optimizing the concordance index between observed event times and estimated ranks of the event occurence. Instead of the naive approach of comparing all data pairs in the response and time domain, it is optimized for selecting only the appropriate pairs to compare, significantly reducing the computation time without loss of performance.

SurvivalSVM has already successfully been used in medical research, including a study by Van Belle et al [21] where SurvivalSVM was used to predict cancer from a high-dimensional microarray gene expression data and showed that it could give better results than traditional proportional hazards models.

Panahiazar et al[17] used Machine Learning techniques on laboratory data, disease information and generally clinical data in clinical care databases in order to calculate a risk score of survival based on patient-specific characteristics. Methods used were random forest, logistic regression, SVM, decision trees and boosting methods. The work managed to improve the ROC score calculated by using the already well known Seattle Heart Failure Model (SHFM).[13] The best pest performing methods were logistic regression and the Ada Boosting method based on the work by Collins[6].

Gijberts at al [8] used data from the same patients that this thesis is using, also using a baseline set of clinical as well as some haematological lab-based predictors to construct survival graphs using the Proportional Hazards model, with the goal to predict death and other major cardiovascular events. Along with the AUC scores, the Continuous net reclassification improvement (cNRI) and the Integrated Discrimination Improvement (IDI) were calculated in order to find the improvement between different models.

Resulting from the analysis in the work above, the features found to be most predictive for survival were :

1. Red cell distribution width.
2. MLR : Ratio of Monocytes to Lymphocytes
3. LMR : Ratio of Lymphocytes to Monocytes
4. Monocyte percentage in Leukocytes

5. Monocyte count

6. Lymphocyte count

7. Lymphocyte percentage in Leukocytes

1.3 Chapter Synopsis

- Chapter 1 is the introduction of the dissertation, showing the dataset and information about the nature of the research.

- Chapter 2 discusses the theoretical background and methodology used in the research.

- Chapter 3 discusses the steps taken to implement the methodology discussed above in order to get the needed results.

- Chapter 4 demonstrates the results and the discussion relevant to them, together with the conclusion and ideas for future work.

- Chapter 5 provides the concluding remarks of the research, together with some ideas for the continuation of the work.

2 Methods

2.1 Survival Analysis

Survival analysis [5] is defined as the branch of statistics that deals with analyzing data where the outcome variable is the amount of time until one or more events happen. It can be defined as a way to analyze time-to-event data.

One of the prerequisites to implement survival analysis is to define the event and the time units used to measure the 'lifetime' or the time until that event occurs. In survival analysis, the dependent variables can be considered as :

1. A binary variable that represents the event happening or not.
2. A numeric variable for the length of time that passed from the beginning of the study until the event happens or the study finishes.

The definition of these variables can contain some ambiguity. For example if the event is organ or mechanical failure there could be multiple definitions for it. The usual survival analysis models assume that this ambiguity has been cleared up. Usually, in addition to the mortality data, there are other observation data, which serve the role of independent variables.

Survival analysis was first used in medical research, and continues to extensively be used. However, widespread use of survival analysis started to take place during and after WW2 where the reliability of military equipment needed to be accurately measured. After the end of the war, it also rapidly spread to the private sector, generally testing the reliability of products. During the 70's, survival analysis started to also be used in the social sciences. In that context, it can investigate phenomena such as employment, inflation, supply and demand for bank loans and life expectancy of products.

Other names for Survival analysis are : reliability theory or reliability analysis in engineering, duration analysis or duration modelling in economics, and event history analysis in sociology.

In the case of survival analysis in medical research, the most popular events used are death, development of an adverse reaction, relapse from remission, and development of a new disease. The duration of the research varies, and could last weeks or even years. The unit used to count the time is usually days. Theoretically, the time until the event can be zero. However, it should always be non-negative.

For simplicity's sake, and since this is the case with the particular problem tackled by the thesis, we will follow the usual terminology of survival analysis literature, referring to the event of interest as 'death' and to the time to the event as survival time.

2.1.1 Censoring

Censoring[20] is a very fundamental issue that occurs in survival analysis when the amount of time units until an event is only partially known. Several types of censoring can occur :

1. Right censoring occurs when a subject leaves the study (also defined in medical literature as lost to follow-up) or when the study is finished before the event occurs for that subject.
2. Left censoring occurs when the event has happened at some point of time before the study begins. Obviously, it does not make sense in the case of the event being death.

In our case, we encounter the possibility of right-censoring, which is encountered in the subjects that are alive when the last information about them was taken. Therefore, there is measurement data until one point, and the death date is some time after that.

The reasons that censoring can occur in the context of a study are:

1. The study ends before the event happens for this particular subject.
2. A subject withdraws from the study.

3. The measurement data for a subject is missing from a certain time point onwards.

In order for the analysis of censored data to work properly, censoring times should be randomly distributed and must be considered non-informative, otherwise the non-randomness should be in some way incorporated in the analysis.[9]

If there are no cases that are considered censored, survival analysis could be emulated by standard linear regression procedures. This can be attained by using time as a target variable. However, using this might not be as informative as survival analysis for the estimation of survival. Possible reasons include:

1. Time-to-event is always defined to be positive so if it is used as an target variable it should have a skewed distribution.
2. The probability of survival beyond a certain point (survival function) and the hazard function used are more informative than just using a method like regression, which does not provide a survival function.

Possible types of right-censoring are :

1. Fixed Type I censoring which occurs when a study has a fixed duration and is decided from the beginning that it should end after time t. In that case, every subject that has not experienced the event after t units of time is considered censored.
2. Random Type I censoring which occurs when a study has a fixed end date but does not begin at the same time for each subject, meaning that censoring times are different for each subject.
3. Type II censoring which signifies a study that ends after a specified number of events.

In our case, the data is taken from the patients on admission to the hospital, and a death date is provided if the patient died before the last day of the study, which is a fixed date. Therefore, the censoring falls into the case of Random Type I censoring.

2.1.2 Survival Function

Let T be a non-negative continuous random variable that represents the time until the event occurs. The probability density function of T is demoted by $f(t)$ and its cumulative distribution function is :

$$F(t) = \Pr(T < t) \tag{1}$$

.

for the specified time t.

The survival function $S(t)$ gives the probability that the object will survive longer than t.

$$S(t) = \Pr(T \geq t) = 1 - F(t) \tag{2}$$

The survival function has the following properties :

1. It is non-increasing, so $S(u) \leq S(t)$ for all $u > t$.
2. When $t = 0$, it is defined that $S(t) = 1$, so every subjects survives at $t = 0$.
3. When $t = \infty$, it is defined that $S(t) = 0$

The survival function can be expressed both with parametric and non-parametric methods. Parametric methods assume that the underlying distribution of the survival times follows certain known probability distributions. Popular ones include the exponential, Weibull, and log normal distributions. For parameter estimation, they usually use a form of maximum likelihood.

However, the most prevalent method of estimating the survival function is using the Kaplan-Meier estimator.

2.1.3 The Hazard Function

An alternative characterization of the distribution of T is provided by the hazard function. The hazard function (also known as the failure rate or force of mortality) is the event rate of mortality for a certain time t, conditional on survival until time t or later (T>t). The hazard function can be considered as the instantaneous risk of the event happening at a certain time t after admittance, or as the instantaneous rate of occurrence of the event. It is defined as :

$$h(t) = \lim_{dt \to 0} \frac{Pr(t \leq T < t + dt | T \geq t)}{dt} \tag{3}$$

The numerator of this expression is the conditional probability that the event will occur in the interval $[t, t + dt)$ given that it has not occurred before, and the denominator is the width of the interval. Dividing one by the other we obtain a rate of event occurrence per unit of time. Taking the limit as the width of the interval goes down to zero, we obtain an instantaneous rate of occurrence.

The probability in the numerator can also be explained as the ratio of the joint probability that T is in the interval $[t, t + dt)$ and $T \geq t$,also written as $f(t)dt$, to the probability of $T \geq t$ which is $S(t)$ by definition. Thus, the hazard function can also be expressed as :

$$h(t) = \frac{f(t)}{S(t)} \tag{4}$$

2.1.4 Kaplan-Meier estimator

The Kaplan-Meier estimator (also called the product limit estimate) is a simple non-parametric statistic that is used to compute the survival function based on time-to-event data. The results of the Kaplan-Meier estimator are expressed in a curve which plots the probability of survival over time when time is measured in small intervals.

For a sample size of N patients, let the observed times-to-event for each patient be :

$$t_1 \geq t_2 \geq t_3 \geq \ldots \geq t_N \tag{5}$$

Let n_i demote the total number of non-censored subjects until time t_i and d_i the total number of events happening (in our case deaths) until time t_i.

The Kaplan-meier estimator is the non-parametric maximum likelihood estimate of S(t) where the maximum is taken over the set of all piece-wise constant survival curves with breakpoints at the event times t_i. The estimated survival function can be expressed as

$$\hat{S}(t) = \prod_{t_i < t} \frac{n_i - d_i}{n_i} \tag{6}$$

The Kaplan-meier curve shows a series of declining horizontal steps, which, in a large enough sample size, should approximate the plot of the true survival function. Between these steps, the value of the survival function is implied to be constant.

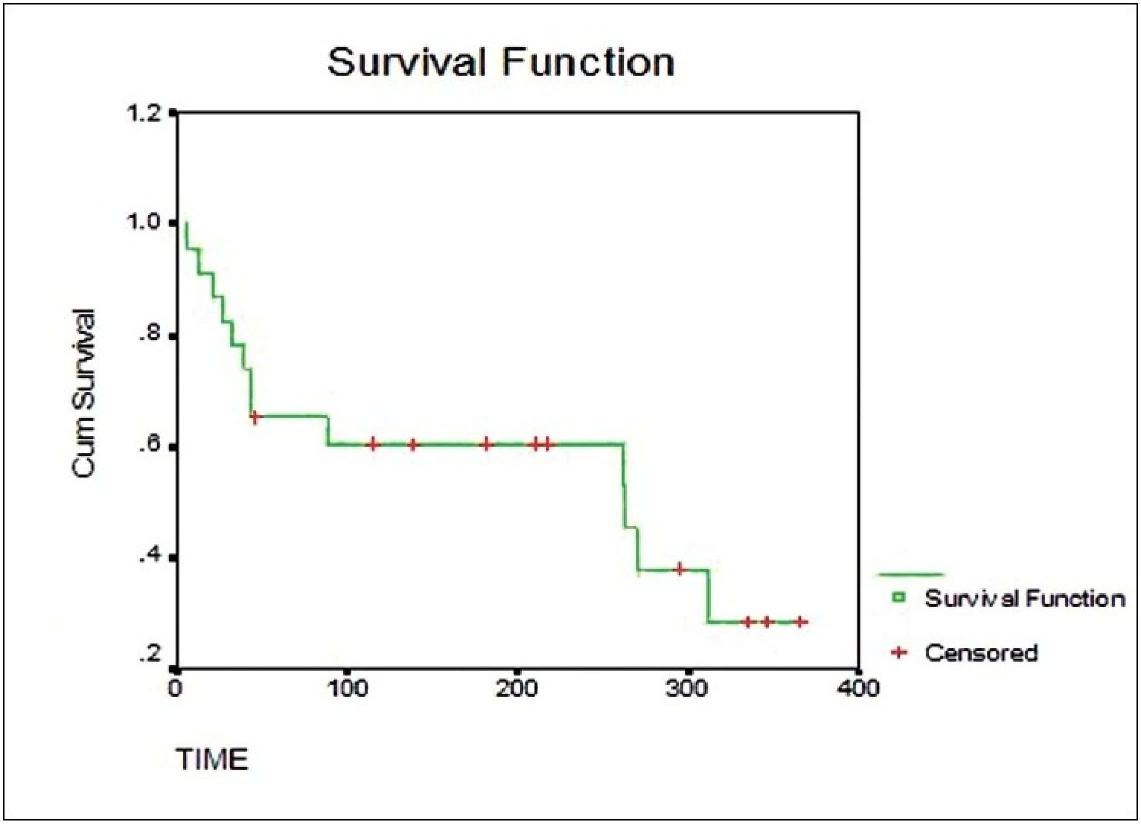


Figure 1: Example of a Kaplan-Meier Curve

In the usual case, a vertical drop in Kaplan-meier curve signifies an event. Right censored cases are also followed until the censoring time, which in most representations of Kaplan-meier curves is shown by a checkmark.

2.1.5 The log-rank test statistic

It is also possible for two groups of subjects to be compared with each other by testing the null hypothesis that there is no difference in the survival distribution between these groups. Simply comparing the proportions of survival at a specific time does not give an overall picture. Alternatively, the most widely used method to for this is the log-rank test, a form of chi-square test that compares estimates of the hazard functions of the groups at each given time.

For each time, it measures the observed and expected number of events in each group and aggregates them to get an overall summary across all the time points where there is an event.

The log-rank test is also called Mantel-Cox test, and is equivalent to a time-stratified Cochran-Mantel-Haenszel test(Mandel, 1963).

For each distinct time j that there is an observed event in either group, N_{1j} and N_{2j} is defined as the number of subjects eligible for an event (haven't had en event and not being censored) and O_{1j} and O_{2j} as the observed number of events, so $Oj = O_{1j} + O_{2j}$ and $Nj = N_{1j} + N_{2j}$.

Under the null hypothesis of the two groups having identical survival and hazard functions, the distribution has expected value of

$$E_{1j} = N_{1j} \frac{O_j}{N_j} \tag{7}$$

and variance of :

$$V_j = \frac{O_j (N_{1j}/N_j)(1 - N_{1j}/N_j)(N_j - O_j)}{N_j - 1} \tag{8}$$

The null hypothesis is tested by comparing $O_1 j$ to $E_1 j$ for each j and is defined as :

$$Z = \frac{\sum_{j=1}^{j} O_{1j} - E_{1j}}{\sqrt{\sum_{j=1}^{j} V_j}} \tag{9}$$

The log-rank test statistic can be understood as the score function of proportional hazards model comparing two groups, and therefore is equivalent to the likelihood ratio test based on that model.

2.1.6 Hazard Ratio and Relative Risk

The hazard ratio is defined as the ratio of the respective survival functions obtained by performing an analysis on two levels of an explanatory variable. In practice, it is a measure that shows how much the effect of a specific variable contributes to the value of a target variable (the time until the event). For example, from a medical perspective, the hazard ratio could describe the odds of a patient healing faster under a particular treatment as opposed to using another treatment or no treatment at all. However, it is not a time-based method and therefore does not convey information about how much each treatment shortens the patient's healing.

In clinical trials, the term hazard ratio is often used interchangeably with the term of relative risk. However, there are some subtle yet important differences between them. The relative risk measures the ratio of the probability of an event occurring in an exposed group by the probability of an event occurring in a non-exposed group. In a clinical trial example, it could compare the risk of developing a disease for two groups of patients taking a different kind of medication.

The main conceptual difference between the hazard ratio and the relative risk is that relative risk is cumulative over the entire study using a defined endpoint and describes the whole time period whereas the hazard ratio describes the instantaneous risk over a subset of the time period.

2.2 The Proportional Hazards Model

2.2.1 Proportional Hazards Assumption

A Proportional hazards model is one particular class of survival models that assumes that the effects of the predictor variables (and thus the hazard function) are constant over time.

For example, a treatment with a certain drug could constitute a 10% reduction in risk of dying to a patient. The proportional hazards assumption would hold if the reduction percentage is the same at any time t. If the proportional hazards assumption holds, then it is possible to determine the model's parameters without having considered the hazard function.

One way to test the proportional hazards assumption is to test for a non-zero slope in a generalized linear regression of the scaled Schoenfeld residuals on functions of time[19] . In some cases, just looking at the results of the numerical test cannot detect a non-proportionality (the relation of the residuals to the time), even though it will be obvious if the residuals are plotted versus a function of time.

2.2.2 Cox Regression

It has been claimed [7] that if the proportional hazards assumption holds, then the effect of the covariates can be estimated without using the hazard function. The method proposed is called Cox regression and the model output is the Cox Proportional Hazards Model. Since h(t) does not have to be specified, it is considered a semi-parametric method. An essential prerequisite of implementing Cox Regression is checking if the proportionality assumption exists. Methods for doing this will be discussed further.

The hazard function for the proportional hazards model has the following form :

$$h(t) = h_0(t) \times exp\{b_1x_1 + b_2x_2 + ... + b_px_p\} \tag{10}$$

$h_0(t)$ is called the *baseline hazard* and is the value of the hazard function when all the coefficients are set to 0. The baseline hazard function is estimated non-parametrically, so the survival times are not expected to follow a particular distribution. In essence, cox regression is a linear regression of the logarithm of the hazard function on the x$_i$ variables, with h(0) being an intercept that varies with time.

The quantities $exp(bi)$ are the model's *hazard ratios.* A hazard ratio greater than 1 for b$_i$ implies that as the value of the parameter increases, the hazard is going to increase in proportion, and thus the probability of survival will decrease.

Based on the above function, the ratio of the actual hazard function and the baseline hazard function can be computed. This is called *relative risk* and can be expressed with summing all the individual hazard functions for each parameter. Dividing the hazard function left and right by $h_0(t)$ results in the hazard ratio:

$$\frac{h(t)}{h(t_0)} = exp \sum_{i=1}^{p} x_i b_i \tag{11}$$

The estimation of the coefficients is made through the partial likelihood. This function is constructed by conditioning on the observed event times and computing a conditional probability that individual i experienced the event, given that someone did. :

$$L_p(b) = \prod_{ti} \frac{e^{x_i b_i}}{\sum_{j \epsilon R_i} e^{x_i b_i}} \tag{12}$$

where R_i is the set of subjects that could experience an event at t_i, and subject i is the subject with the event at t_i. We can think of the partial likelihood as the joint density function for subjects' ranks in terms of event order, if there were no censoring and no tied event times. Consequently if we use the partial likelihood for estimation of parameters we are losing information, because we are suppressing the actual times of events even though they are known, hence the name ”partial likelihood”.

One important property of the above model is that in order to estimate the regression coefficients, the only information needed is the ranks of the failure times. The actual absolute times are not used except in generating the ranks, therefore the time interval between the units should not matter. The shape of the baseline hazard is also irrelevant since it does not appear in the calculation.

There are several approaches for handling ties in the data. Breslow's method is the most commonly used method, mainly because it is easy. It is an approach in which the partial likelihood method is ran in a similar way as shown above. In our case the Effron approximation will be used, a method that is generally agreed to produce better results[1].

2.3 Performance Scoring

The c-index or Concordance Index is one of the most popular scoring measures for scoring the prediction performance of different time-to-event models. It was introduced by Harrel, Lee, and Mark [10]. It can also be defined as a measure of the amount of 'concordance' or agreement between the predicted and the observed survival. Unlike other similar performance measures, Harrell's C-index does not depend on choosing a fixed time for evaluation of the model and specifically takes into account censoring of individuals. When there are no censored data, the C-Index is equivalent to the estimation of the Mann-Whitley parameter $Pr(X > Y)$ [14].

Let T_i define the time to event for case i. The concordance index is computed as follows:

1. Form all possible pairs of observations over all the data.

2. Omit those pairs where the shorter event time is censored. Also,

omit pairs i and j if $T_i = T_j$ and both are censored, therefore T_i and T_j have maximum values. Ties are only allowed if one of the observations stops at an event and the other is a censored observation. Let P denote the total number of permissible pairs.

3. Count 1 for each permissible pair in which the shorter event time had the shorter predicted event time. Count 0.5 if the predicted outcomes are tied. Let C denote the total sum over all permissible pairs.

The concordance index is defined as $CI = \frac{C}{P}$.

2.3.1 Validation

One of the main characteristics of our particular dataset is that it is both relatively small (as a ratio of rows to columns) and sparse, meaning that death occurs in less than 10 percent of the cases. With that in mind, it is not practical to follow a classical approach of dividing the data into a training and a test set, since the data omitted from the training set is too vital for the model. Also, it is almost certain that any model produced this way would be overfitted and would not generalize well. Thus, it is imperative that a more effective type of approach should be used.

In our case, a bootstrapping method with replacement was implemented. In this case, the observed dataset is repeteadly sampled and the samples form a large number of bootstrapped datasets (usually given by the user), each having the same size. The purpose is to treat the original data as the general population and the bootstrap samples as samples from that population. The model is fitted to all the bootstrap samples and an error measure of some kind is estimated. The estimate of the variance of the original data is the variability of the point estimates accross the bootstrapped datasets.

The bootstrapping approach used for validation in the present work is also described in Harrel's 1996 paper[10]. It uses the C-Index as a performance measure, outputting it in the form of the Sommer's D (Dxy value) which can be transformed to the C-Index by calculating $0.5 * (Dxy + 1)$. It consists of the following steps :

1. Fit the model to the original data and estimate C using this fitted model. This estimate of C or Apparent Concordance Index can be denoted as $Capp$.
2. for $b = 1, .., B$:
   (a) Take a bootstrap sample with replacement from the original data, with size equal to the number of cases.
   (b) Fit the model to the bootstrap sample, and estimate C using this fitted model and this bootstrap dataset. Denote the estimate by $Cb, boot$
   (c) Estimate C by applying the fitted model from the bootstrap sample to the original data. Let $Cb, orig$ denote the estimate
3. Calculate the estimate of optimism $O = B^{-1} \sum_{b=1}^{B} C_{b,boot} - C_{b,orig}$
4. Calculate the optimism adjusted measure of predictive ability as $Capp- O$

This bootstrap approach is very intuitive: usually when we apply a model fitted using a bootstrap dataset to the original data, the predictive accuracy will be lower than the apparent accuracy when evaluating the fitted model using the same data that was used to fit it. We calculate the difference in these predictive abilities for each bootstrap sample, and take the average across many (Harrell et al suggest 100-200 times) bootstrap samples. This estimate of optimism is then subtracted off the naive estimate of predictive ability.

2.4 Preprocessing

2.4.1 Imputation

After the initial discarding of the variables that convey information of no clinical use, there may still be some variables with missing values. The process of replacing missing data is called imputation. The type of imputation used here is multiple imputation, [15]. The method works by performing stochastic regression methods on multiple imputed data sets, and the results are analyzed seperately and averaged. It models the noise in the distribution of the data, and therefore is able to model the uncertainty of the process that created it.

The goals of an imputation model are to account for the process that created the data and preserve both the relations in the data and the uncertainty about them.

The method chosen was fully conditional specification (FCS), also known as MICE (Multiple Imputations with Chained Equations). FCS works by specifying the imputation model on a variable to variable basis by a set of conditional densities, one for each incomplete variable. It draws the imputations based on iterating over the conditional densities.

2.4.2 The MICE Algorithm

Let Y_j with $(j = 1, ..., p)$ be one of p incomplete variables, where $Y = (Y_1, ..., Y_p)$. The observed and missing parts of Y_j are denoted by Y_j^{obs} and Y_j^{mis} respectively, so $Y_j^{obs} = (Y_1^{obs}, \ldots, Y_p^{obs})$ and $Y_j^{mis} = (Y_1^{mis}, \ldots, Y_p^{mis})$ stand for the observed and missing data in Y respectively.

The number of imputations is equal to $m \geq 1$. The hth imputed data sets is denoted as $Y^{(h)}$ where h = 1, . . . , m. Let $Y_{-j} = (Y_1, ..., Y_{j-1}, Y_{j+1}, ..., Y_p)$ denote the collection of the $p - 1$ variables in Y except Y_j . Let Q denote the quantity of scientific interest (e.g., a regression coefficient). In practice, Q is often a multivariate vector. More generally, Q encompasses any model of scientific interest.

The analysis starts with an observed, incomplete data set Y^{obs}. The problem encountered when trying to solve this is that it is impossible to estimate Q from Y^{obs} without making unrealistic assumptions about the unobserved data. Multiple imputation methods try to solve this problem by creating several imputed versions of the data, replacing the missing values with plausible ones. These plausible values are drawn from a distribution specifically modeled for each missing entry. The result is a number of datasets that have the same values in the places where there were no missing values but different imputed values for each one. Since all the datasets are now complete, normal analysis techniques can be used to estimate Q on each imputed dataset. Finally, all the estimates $Q^{(1)}, ..., Q^{(m)}$ are pulled into one estimate and the variance is calculated. For quantities Q that are approximately normally distributed, we can calculate the mean over $Q^{(1)}, ..., Q^{(m)}$ and sum the within- and between-imputation variance according to the method proposed by Rubin.

The chained equations method works as follows :

Let the hypothetically complete data Y be a partially observed random sample from the p-variate multivariate distribution $P(Y \mid \theta)$. We assume that the multivariate distribution of Y is completely specified by θ, a vector of unknown parameters. The problem is how to get the distribution of $P(Y \mid \theta)$, either explicitly or implicitly. The algorithm manages that by sampling iteratively from conditional distributions of the form $P(Y_1 \mid Y_{-1}, \theta_1), \ldots, .P(Y_p \mid Y_{-p}, \theta_p)$.

The parameters θ_1, . . . , θ_p are specific to the respective conditional densities and are not necessarily the product of a factorization of the 'true' joint distribution $P(Y|\theta)$. Starting from a simple draw from observed marginal distributions, the tth iteration of chained equations is a Gibbs sampler (Geman, 1984) that successively draws :

$$\theta_1{}^{*(t)} P(\theta_1 \mid Y_1{}^{obs}, Y_2{}^{(t-1)}, \ldots, Y_p{}^{(t-1)})$$

$$Y_1{}^{*(t)} P(Y_1 \mid Y_1{}^{obs}, Y_2{}^{(t-1)}, \ldots, Y_p{}^{(t-1)}, \theta_1{}^{*(t)})$$

$$\ldots$$

$$\theta_p{}^{*(t)} P(\theta_p \mid Y_p{}^{obs}, Y_1{}^{(t)}, \ldots, Y_{p-1}^{(t)})$$

$$Y_p{}^{*(t)} P(Y_p \mid Y_p{}^{obs}, Y_1{}^{(t)}, \ldots, Y_p{}^{(t)}, \theta_p{}^{*(t)})$$

where $\hat{Y}(t)_j = (Y_j^{obs}, Y_j^{*(t)})$ is the jth imputed variable at iteration t. Previous imputations $Y_j^{*(t-1)}$ only enter $Y^{*(t)}$ through its relation with other variables, and not directly. Convergence can therefore be quite fast. The number of iterations can often be a small number, say 10-20. The name chained equations refers to the fact that the MICE algorithm can be easily implemented as a concatenation of univariate procedures to fill out the missing data.

In the process of imputing multivariate data, one can also encounter a number of problems [22]:

1. For a given Y_j , predictors Y_{-j} used in the imputation model may themselves be incomplete.

2. Circular dependence can occur, where Y_1 depends on Y_2 and Y_2depends on Y_1 because in general Y_1 and $Y2$ are correlated, even given other variables.

3. Especially with large p and small n, collinearity and empty cells may occur.

4. Rows or columns can be ordered, e.g., as with longitudinal data.

5. Variables can be of different types (e.g., binary, unordered, ordered, continuous), thereby making the application of theoretically convenient models, such as the multivariate normal, theoretically inappropriate.

6. The relation between Y_j and Y_{-j} could be complex, e.g., nonlinear, or subject to censoring processes.

7. Imputation can create impossible combinations (e.g., pregnant fathers), or destroy deterministic relations in the data (e.g., sum scores).

8. Imputations can be nonsensical (e.g., body temperature of the dead).

2.4.3 Regularisation

In machine learning, regularization is the process that adds additional information to a model in order to prevent overfitting and to solve wrongly posed problems by penalizing models with extreme coefficient values. In practice, it is a model selection technique that imposes a complexity penalty in order to reduce the number of variables in the model.

The two fundamental requirements for regularization are :

1. A way of validating the predictive power of the model, for example cross-validation.

2. A parameter that lets the user choose the strictness of the complexity penalty, and therefore the complexity of the model.

For our case, the regularization process involves penalized maximum likelihood[16]. The most common variants are L1 and L2 regularization, which work by modifying algorithms that minimize a loss function $b(X, Y)$ to minimize $b(X, Y) + \lambda||w||$, where b is the model's weight vector, the $||.||$ operator is either the L1 or L2 norm and λ is the regularization parameter.

Specifically, the LASSO penalty will be used. It uses the constraint that $||w||_{1,}$ the L1-norm of the parameter vector is no greater than a given value. This way, an upper limit is set to the sum of the absolute values of the coefficients.

One of the advantages of the LASSO method in respect to the other methods (for example using the L2 penalty), is that as the regularization penalty increases, more parameters will turn to zero. So, the resulting variable set has less but more relevant nonzero variables than in the other cases, making the LASSO method better for variable selection purposes.

The R package used for implementing the regularization is glmnet. It computes a grid of values for different values of the regularization parameter alpha. The algorithm is fast and can exploit sparsity in the input value matrix. The glmnet algorithms use cyclical coordinate descent (Windham et al, 1987) which successfully optimizes the objective function over each parameter with others fixed, and cycles repeatedly until convergence.

2.4.4 Stepwise Regression

Another method for model selection used was forward and backward stepwise regression. The algorithm starts with a candidate variable set and, using a certain measure for each step, looks for a best model through the possible variable sets by checking the addition and removal of variables. It uses various methods of evaluating the models in order to find the best possible improvement for each step.

Forward stepwise selection implements a greedy hill climbing technique. The search starts from a model with no variables and adds for each step the variable that gives the greatest improvement in score. until no further improvement can be made. Conversely, backward stepwise selection starts with the full model containing all candidate variables and removes the variable whose deletion has the best impact on the model fit, repeating this process until there is no variable that improves the model.

There is also the possibility of bidirectional elimination, testing all backward and forward moves for each step and determining the best fit.

A multitude of measures can be used as the model comparison criterion. It can be a simple test based on F-statistics (adding significant terms or dropping non-significant ones). Some implementations also use other kinds of criteria such as adjusted R-square, Akaike information criterion, Bayesian information criterion, PRESS, or false discovery rate.

Stepwise regression generally should obey the hierarchy principle. This states that if a model contains either X^k or some interaction tern involving X and is shown to be a statistically significant predictor of Y, then the model should also include X and in the case of X^k all X^j where $j < k$, whether these lower-order terms are significant or not. Similarly, if the result model contains parameters that function as interaction terms between 2 or more variables, it also should contain the original variables that form these interaction terms.

2.5 Random Forest

Random Forests [4]are generally defined as an ensemble learning method for classification, regression and other tasks, that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees' habit of overfitting to their training set. Other alternative methods to Random Forest are bagging [3] and boosting [18].

In Random Forests, randomization is introduced in two forms. First, a randomly drawn bootstrap sample of the data is used for growing the tree. Second, the tree learner is grown by splitting nodes on randomly selected predictors.

In the general case, each tree is grown with the following method :

1. Draw a bootstrap sample of the training data
2. Grow a decision tree on the data by recursively repeating the following steps for each terminal node of the tree until a minimum node size *nmin* (provided by the user) is reached :
    (a) Select m variables at random from the variable set. The value of m should be provided in the algorithm call.
    (b) Pick the best variable split point among the chosen variables.
    (c) Split the node into 2 child nodes.

2.5.1 Random Survival Forests

Random Survival Forests [12], implemented by the randomSurvivalForest package in R, is a method modeled after the one proposed by Breiman and can be used for the analysis of right censored data. Its main notable feature is that it is highly adaptive and assumption free. Therefore, it does not rely on the usual restrictive assumptions like proportional hazards that other methods do.

The survival forest algorithm, as implemented by Iswaran et al is described as follows :

1. Draw B bootstrap samples from the data having a size of 2/3 of the original data.
2. For each one of the samples grow a tree. For each node of the tree, randomly select m variables (where m is given by the user) for splitting on. Split the node on the variable which maximizes survival time differences across daughter nodes.

3. Grow the tree to the maximum size satisfying the constraint that the total number of events in a node should be larger than a predefined number.

4. Calculate a total cumulative hazard function for each tree and average them to obtain the ensemble cumulative hazard function

5. Compute an OOB (out of the bag) error rate for the ensemble.

For each split, a predetermined survival criterion is used. Starting from the root node, every node is split into a left and right daughter node in a recursive repetition of the process. The best split for a node is found by searching over all possible combinations of variables and split values of the variables and maximizing the survival difference. This way, dissimilar cases will be pulled apart, iteratively increasing the homogenity of each node in each split.

The value of m, n and the minimum events per node are parameters that should be provided by the user in the function call.

The models' performance is scored by the error rate, calculated as 1 minus the model's concordance index. The error rate must be between 0 and 1. An error of 0.5 signifies that the model scores no better than random guessing, while an error rate of 0 shows perfect accuracy.

3 Analysis

In this chapter, we will discuss the practical implementation of the theory discussed in the previous chapters, including manual data selection, imputation, variable selection and validation.

3.1 Data collection

At the moment of inclusion, blood was drawn from the arterial sheath inserted for coronary angiography. Differential blood counts were performed according to routine clinical practice and all hematological parameters were subsequently stored in the UPOD. From the UPOD, we collected an extraction of the Abbott Cell-Dyn18 data. The data that corresponded to the blood sample drawn for the purpose of this study were used for analysis.

Parameters that were used in this study comprised all 15 routine UPOD leukocyte parameters: leukocyte, neutrophil, monocyte and lymphocyte counts and percentages, neutrophil cell size (mean and coefficient of variation (CV)) and complexity (mean and CV) and lymphocyte cell size (mean and CV) and complexity (mean and CV). These numbers are derived from the Abbott Cell-Dyn18 machine which uses multi angle polarized scatter separation to classify cell properties. By shining an Argon laser on individual cells, cell size is defined as the axial light loss at 0 degrees and cell complexity is defined as the intermediate angle forward scatter at 7 degrees. The CV represents the standard deviation of the measurements, indicating the variation in cell size or complexity within one patient.

Additionally, other types of data were collected separately. These include : demographical data, history of acute coronary syndrome (ACS), history of percutaneous coronary intervention (PCI), history of coronary artery bypass grafting (CABG), cerebrovascular accident/transient ischemic attack (CVA/TIA) or peripheral arterial disease (PAD), medication use, cardiovascular risk factors (diabetes, body mass index (BMI), hypertension, hypercholesterolemia, smoking), the indication for catheterization, the angiographic severity of CAD and details concerning the procedure were collected.

3.2 Initial Selection

From the full UPOD dataset, 76 variables were chosen. They include the full set of the patients' hematological measurements taken from the Abbot Cell-Dyn Sapphire analyzers, as well as a number of binary variables indicating various diseases and standard demographic and physical data including age, sex and weight. The latter is the baseline model used for survival model research in the UMC hematology department. Also, a BMI variable was created from the height and weight variables.

The sample population measures 2450 patients, of which 142 died before the end of the study. Subjects were all over 18 years old, with 1791 of them being men and 659 being women. There are 27 variables with missing values, most of them having around 10 values missing. The only ones with considerably more are the binary variables of Smoking (187) and LV function (548).

3.3 Imputation

The first step of the MICE imputation pipeline is to create a prediction matrix. This is an $m * m$ binary matrix where m is the number of variables. For each variable, it indicates which ones of the others will be used for the calculation of imputing its missing values. For the purpose of imputation, we had to exclude the variables that are expected to have no predictive power over any of the rest of them, namely the times and dates of admittance or death and the variable containing the patient IDs.

The multiple imputation produced 15 imputed datasets each having different values, which were joined together. The fact that our dataset had 15x more cases than the original data was solved by weighting each case by 1/15 in each subsequent analysis step. This was defined as the 'long' version of the data.

Additionally, a 'short' version was created by one of the 15 imputed datasets output, in order to compare the effectiveness creating multiple datasets during multiple imputation.

3.4 Proportional Hazards Assumption

For testing the assumptions of proportional hazards, we used the *cox.zph* method from the *rms* package. The method checks proportionality of all model predictors by looking for its interactions with time. It produces the Pearson-product-moment correlation between the scaled Schoenfeld residuals and the time variable for each covariate, in the form of p-values. It also includes a global test that counts for the interactions of all the predictors tested at once.

The null hypothesis is that the scaled Schoenfield residuals are not correlated with time. Therefore, we can deduce that a p-value of less than 0.05 signifies a violation of the proportionality assumption.

Using plot() on a cox.zph model results in plots for the residuals for each individual variable against time, together with a smooth curve. This is useful in order to capture different kinds of correlation of the residuals with time that can't be made clear with the numerical test, such as quadratic relations. Generally, in order for the assumption to be satisfied, the line has to be as straight and 'flat' as possible, with a slope of zero or close to it.

The scoenfield residuals method for testing the proportional hazards assumption of the long models does not give accurate results, probably because of the lack of support for weighting, therefore the assumption was impossible to determine by numerical tests. For most cases, the plots for the residuals look similar in the short and long models with the same variables, however for the long models there is no definite way to guarantee that the assumption is satisfied.

3.5 Main Analysis

Initially, we built 3 models :

1. The baseline model, containing only the patient information variables without the haematological data.
2. The lab data model, containing only the data resulting from blood analysis, plus the Age and Gender of the subject.
3. The full model, containing all variables.

After performing L2 regularization and stepwise regression in those models and the models resulting from these procedures, cox regression and survival random forest models were fitted for each one and the resulting models were compared to each other. Each model was tested both on the long and the short dataset.

3.5.1 Proportional Hazards model Fitting.

The proportional hazards model fitting was performed by the cph function in Harrel's rms package [9], using the implementation of the partial likelihood method originally present in the coxph function of the survival pagkage . In order to avoid overflow in the argument to the exponential function, the algorithm automatically scales and centers data.

The output from the cph function produces the following statistics:

1. The coefficient for each variable. Their exponentiated versions can be interpreted as multiplicative effects on the hazard.
2. The standard errors for each coefficient

3. A z score, representing the ratio of each regression coefficient to its standard error, a Wald statistic[9] that is asymptotically standard normal under the hypothesis that the corresponding β is 0.

4. The p-values of the total likelihood ratio and logrank score, which are asymptotically equivalent tests of the omnibus null hypothesis that all of the βs are equal to 0.

5. Discrimination indexes like the model's R squared and Sommer's D, which can be used for ranking the models' performance.

The coxph method documentation describes a case where the maximum likelihood estimate of a coefficient can be infinity. That implies a dichotomous variable that divides the dataset to 2 groups, one with all the events happening and another that none happen.

In that case, the following can happen : Either the log likelihood converges, the information matrix becomes effectively singular, an argument to the exp() function becomes too large for the computer hardware, or the maximum number of interactions is exceeded, with the most probable cause being the first. The primary consequence for the user is that, in that case, the Wald statistic is not useful. However, the likelihood ratio and the score tests remain valid.

After the manual variable selection and imputation, the dataset being output as just a full model with all the original variables as predictors would still be potentially inefficient to directly input into machine learning models. The main reason is that the variable set is still too large for traditional machine learning algorithms, both simply because of its size and also because the row-to-column ratio is lower than usual.

There should also be a lot of intercorrelation between the variables due to the nature of the data. Therefore, automatic variable selection techniques can play a very vital role in solving this problem by offering simpler models with less variables that capture the most possible amount of variance.

The literature offers a multitude of variable selection techniques. The ones used in this project are regularization using penalized maximum likelihood via the lasso penalty (from R package glmnet) and model selection via stepwise regression using the BIC penalty (from the stepAIC package).

All the different models that resulted from variable selection algorithms were used for further analysis and comparison with each other.

Before the LASSO regularization step began, proportional hazards models were fitted for each of the 3 initial variable sets. The function used was cph from the rms package. cph is a modification of the coxph, the standard function of fitting cox models in R, offering some extra options that enable more efficient validation of the cox regression results.

The function also supports weighting of the input data. For the cases that the long model is an input, each variable was weighted with weights of value 1/15.

3.5.2 Regularization

The first step taken was automatic variable selection. This was achieved by executing LASSO regression on each model through the cv.glmnet function contained in the glmnet R package.

Glmnet computes the grid of values for different values of the regularization parameter lambda and outputs the result for each respective value. The algorithm is fast and can exploit sparsity in the input value matrix. The glmnet algorithms use the method of cyclical coordinate descent [2], which successively optimizes the objective function over each parameter with others fixed, and cycles repeatedly until convergence. While in our case the LASSO penalty was used, it would also be possible to use the elastic-net penalty. The cv.glmnet function runs the glmnet algorithm together with k-fold cross-validation. In our case the default value of 10 was used.

The output is a set of different models for different lambda values. The lambda value finally used is the one that produced the minimal cross-validation error. The resulting models contain all variables that have non-zero values of coefficients after the algorithm execution.

Results

1. In the baseline model, only the *Indication* variable was removed by glmnet.
2. In the lab data model, *neutrophil percentage*, *c-b-ht* and *c-b-rbcfmn* were removed.
3. In the full model, only *c-b-rbcicv* and *c-b-HDW* were removed. However, in addition to imputing the long dataset, imputation of the short version was also tried, with the results being much different. In that case, only 36 variables remained in the final model. This may be due to the smaller variability of the imputed values in the short dataset.

3.5.3 Stepwise regression

The implementation that was used is the StepAIC function contained in the MASS package. Its input is an initial model –from where the search starts– plus upper and a lower variable formulas used for providing boundaries to the search process. The model is an R object representing a fitted model, in our case a result of the cph function of the R rms package. The upper and lower formulas represent the range of models that the stepwise algorithm will search. Using formulas of variables as an input object also allowed us to specify more complex models to search, in particular models containing interaction terms between variables or squared variables together with their regular form.

The package natively used the AIC as a model selection criterion, however it also allows to manually input the penalty per parameter used in the complexity penalty. Setting $k = 2$ gives the original AIC. In our case the BIC score was preferable and was achieved by using $k = ln(n)$, where n is the number of rows of the dataset.

The algorithm was ran for every starting model and each output of LASSO. The search always started from the empty model, was configured to move both forward and backward, and used as the upper model the following :

1. A model containing the original variables and all interaction terms between them.
2. A model containing the original variables with the squared forms of the variables added.
3. Models containing both.

The StepAIC algorithm supports the hierarchy principle [23]. This states that lower order interactions should always be included together with the higher order ones that consist of them, mainly for reasons of interpretation of the model. In practice, this means that in any case that an interaction or squared term is found in the formula, the algorithm will also add all the lower order terms that it consists of.

3.6 Validation and scoring

The bootstrap approach described in section 2.3.1 is also implemented in the rms package. The cph.validate() method is its implementation for the cox model case. It takes as input a cox regression model created by cph() and calculates several measures, including Sommer's D which is the one measure that will be used in our evaluation of the models. Reparametrized to a probability scale from 0 to 1, it gets transformed into Harrel's C-Index by calculating $1/2 * (D + 1)$. Other measures calculated are :

1. The model's R-Squared
2. Slope shrinkage
3. The discrimination index D [(model Likelihood Ratio chi-square - 1)/L]
4. The unreliability index U = (difference in -2 log likelihood between uncalibrated X beta and X beta with overall slope calibrated to test sample) / L
5. The overall quality index $Q = D - U$.

L equals -2*log likelihood with beta=0. As Harrell states, the corrected slope can be thought of as shrinkage factor that takes into account overfitting.

The measures are calculated in different ways :

1. **Index.orig** : The naive measure, obtained by fitting the model and also evaluating it in the original data.
2. **Training** : The mean across the bootstrap samples of $C_{-b,boot}$
3. **Test** : The mean across the bootstrap samples of $C_{-b,orig}$
4. **Optimism** : The difference between the accuracy of Training and Test
5. **Index.corrected** : The $Index.orig$ value corrected by subtracting the Optimism value

The last value is the most reliable one so we will use the corrected naive measure of Sommer's D in order to rate the model's accuracy. To validate each model, 1000 bootstrap samples were generated.

As an additinal scoring together with with the C-Index, ROC plots were created for the purpose of measuring the model's accuracy on predicting 2 year mortality. The algorithm is based on the method created by Haegerty et al [11] and the implemented in the risksetROC R library created by the same group. To assess the performance, the Area Under Curve score was extracted.

4 Results

4.1 Models

The results of the analysis both from the Cox Regression and the Random Forest techniques are shown in the following table : Cox regression and random forest analysis was ran for the variable sets described below. Analysis was ran both on the short and long dataset. Both 10 and 100 bootstrap sample validation methods were tried, as well as Random Forest analysis using 10 trees. In the cases that it is not specified, the regularization was ran on the long model.

:

1. Baseline Model (A model with all the variables except the laboratory ones)
2. Baseline model after running LASSO regularization on the long dataset for variable selection (Just the indication variable removed)
3. Baseline model after running LASSO and Stepwise Regression
4. Lab Data model (A model having just the data taken from the CellDyn analyzer)
5. Lab Data model after LASSO regularization ran on the long dataset

6. Lab Data model after LASSO regularization ran on the long dataset plus stepAIC finding squared terms
7. Lab Data model after LASSO regularization ran on the long dataset plus stepAIC finding squared terms and interaction terms
8. Full model (All possible variables)
9. Model after LASSO regularization ran on the short dataset
10. Model after LASSO regularization ran on the long dataset
11. Model after LASSO regularization ran on the long dataset plus stepAIC finding interaction terms
12. Model after LASSO regularization ran on the long dataset plus stepAIC finding squared terms
13. Model after LASSO regularization ran on the long dataset plus stepAIC finding squared terms and interaction terms

4.1.1 Proportional Hazards

Looking at the cox regression results, the first thing that becomes obvious is that, when the bootstrap algorithm is ran on datasets with a lot of variables, the failure rate of the bootstrap method is larger. The issue gets worse when the variables are numeric, or there are a lot of interaction terms. For example, the models tested that include a large number of numeric variables (most labdata model and the full one) have a much lower convergence rate in the bootstrap processing, usually not exceeding 50 percent of the cases. The failures or non-convergence in the bootstrap executions are caused by the singularity problem discussed above.

Similarly, in the model resulting from regularization plus stepwise regression finding interaction terms when run in the long dataset, the rate of convergence is 18/1000, which can be attributable to the large number of variables (including pairs of interaction terms) being found by the stepAIC algorithm. Similar problems exist in all the algorithm runs with large variable sets. When StepAIC is asked to find both interaction terms and squared terms, the result is a simple model that contains way less variables.

Comparing the concordance indexes of the cox regression on the short and the long model, we can see that, in the vast majority of the cases the scores get better when the long model is used. The most probable cause of this is that the imputed values replacing the missing ones are significantly more predictive when using 15 aggregated multiply imputed datasets instead of just one. It would be expected that averaging the results would capture more of the true variability of the distribution that produced the data. The downside of using the long version is that, since the case set gets multiplied by 15, the models become more complex. Another downside of the long version of the dataset could be that, since the non-missing values get duplicated 15 times, and since the bootstraping methods divide the set randomly into training and test sets, it is practically certain that there will be cases with almost or exactly the same values in all their variables that exist both in the training and the test set. That would result in lower reliability for the models using the long dataset and also possibly overfitting.

Four of the strongest performing models are shown in Appendix 2. We can see that they mostly agree with each other in calculating the significance of each variable and the selection of the squared and interaction terms. Generally, the coeffients also agree with the ones present in the rest of the models. For example, in all our models Age has a positive coefficient and the squared term gets added any time that it is possible in the stepwise regression runs. Similarly, Gender always has a negative coefficient and some interaction terms reappear in all the models. Thus, we can confidently assume about the significance of certain variables or interactions between them.

From the c-indexes, it is also clear that generally the models containing just the laboratory variables are better than the baseline ones, and the models containing variables from both sets are better than the former.

The AUC scores for 2-year mortality generally seem to generally follow the s the concordance indexes, having higher AUC score in the models with higher concordance indexes. However, the process of bootstrapping was not followed for getting the AUC scores, and so the AUC scores should not be considered as reliable as the reported concordance indexes.

4.1.2 Random Forest

In a similar fashion, Random Forest models were tested in the majority of the models. Because of the computational complexity of the algorithms, the algorithms were unable to provide meaningful results for all the variables sets used for the proportional hazards models analysis. The concordance indexes of the working models created are also shown in the tables in Appendix 2.

In most cases, 10 trees are used for the analysis. Attempts to test the algorithm with 100 and 1000 trees did not produce results after running for more than 48 hours, so they were abandoned. However, the attempts were made on a desktop computer so they could theoretically produce results when ran in faster hardware, such as a high speed processing cluster.

The Random Forest results show that, contrary to the proportional hazards models, the random forest analysis can work well with datasets having a large number of numeric dimensions, even with some of them being intercorrelated. In fact, the best result came from the full model. On the contrary, the models resulting from the automatic variable selection and containing interaction or squared terms produced concordance indexes close to 0.5. This can be attributed to the inherent ability of random forest methods for variable selection. However, in all cases, random forest enforced on the long datasets provides significantly better fit than when enforced on the short ones, with a much larger difference than the one observed in the proportional hazards models. This could also be an indicator of overfitting.

Another observation is that, as in the proportional hazard models, there is a large -better than in the proportional hazards case- improvement in the concordance indexes of the long models comparing to the scores of the single dataset ones.

4.1.3 Specific Models

Model 8 :Full Model The first model analyzed is the full model, containing the original 73 variables used for prediction. ?t is clear that the model is not suitable by itself for any kind of analysis. All available statistics show a worse performance than the models derived from it through variable selection.

As expected, the p-values for most of the variables are above a confidence interval of 0.05, meaning that they are not significant. A notable example is the Age variable which has a small positive coefficient, showing that as age increases, the risk of death increases too. Although in most cases being not significant, most of the binary variables representing the presence of diseases also have a positive coefficient.

Model 2 : Baseline After regularization ?t is obvious that there was an improvement over the full and the baseline model, since the performance scores are significantly better with the variables being much fewer. As in all the other similar cases, the squared version of Age has a positive coefficient and a low p-value, contrary to the normal version that has a negative one and very low singnificance. The gender variable is also significant and with a negative coefficient, something that also corresponds to the established knowledge that women generally live longer.

Model 9 : After regularization starting from the full short model Running the LASSO regularization to full model is very effective, resulting at the most efficient cox regression model, judging by its score after validation. Almost all the p-values of the variables have been reduced and all of the variables indicating diseases (except hyperchoresterolemie) have a positive coefficient as expected.

Model 10 : After regularization starting from the full long model Running the regularization on the long model gives us a model with way more variables and almost similar concordance indexes in the case of the proportional hazards models. The random forest results are noticeably better than the ones in the previous model, and still much better when the model is trained on the long dataset than the short one.

Model 11 : After regularization finding interaction terms starting from the full long model Enforcing stepwise regression on the reduced model looking just for interaction terms results in a very complicated model with lots of variables, most of them being interaction terms. The models have the best concordance indexes than all the ones testing. However, they seem way too complicated for use in real-world applications, and possibly a result of overfitting.

As with almost all the models that contain a lot of variables, the concordance index of the random forest method applied to it is relatively high, although still lower than the one corresponding to the cox model.

Model 12 :Lab Data After regularization ran on the long dataset plus stepAIC finding squared terms This is the best performing model from the ones that started from the labdata variable set. It contains mostly squared terms from the set, plus the Age variable (with coefficients very similar to the ones from the rest of the models) and the gender one, again having a large negative coefficient.

The concordance index is similar to the one from the labdata model containing interaction terms, but the number of variables is smaller and the R-Squared score is much better.

Model 13 : After regularization started from the long dataset, after Stepwise Regression for interaction and squared terms This model looks a lot like model 8 and 9, and is the model that accomplishes the best combination of predictive power and number of variables. It consists mostly of squared and interaction terms with coefficients similar to the ones at the other models, but interestingly without the gender variable, whose variability is probably captured by the squared and interaction terms created.

5 Conclusion

This work showed the application of machine learning and statistical analysis methods on prediction of survival of heart disease based on laboratory and disease information. The main conclusion to be drawn by the results is that the random forest and cox regression methods do not work well when all dimensions of the data are used. However, the variable selection method proposed here, using LASSO combined with stepwise regression, seems to reduce the issue.

Generally, the models that predict better are the ones that directly result from the LASSO analysis, starting either from the long or the short model. Running StepAIC on these models to find interaction terms gives us models with lower concordance indexes but with a much lower number of variables, and thus a lower convergence rate in the bootstrap process. Apart from that, a small number of variables is important also because it helps the interpretability of the model.

The models after LASSO regularization from the short dataset (Model 9) and the one with both squared and interaction terms (Model 13) seem to have the best combination of a small number of variables and high concordance index.

In any case, we see a significant difference between the models where the analysis started from a subset of the vaiable set (baseline or Lab Data) and the models starting from the full dataset. Thus, we can deduce that both types of variables have predictive power over survival, and models using only one type of them can benefit from adding selected variables from the other type.

Furthermore, implementing multiple imputation in order to account for the missing values seems to not work all the times as needed, and caused more problems than it solved, making the proportional hazards assumption not work as needed for all cases and rendering posing a threat to the reliability of bootstrap methods.

5.1 Future Work

The most relevant future work that should be done as a continuation of this one is a continuation of exploring the models presented in this thesis with a dataset of more patients. The results of the bootstrap validation shown in this thesis strongly imply that the models presented can be successfuly generalized and used on new patient datasets. However, in the case of starting with a new and larger dataset, one could also repeat the variable selection steps in order to result in different and improved models.

Furthermore, in order to tackle the issue of the large number of variables and the possible intercorrelation with each other, algorithms like Principal Component Analysis can also be used to combine some variables with each other, and thus result in a more robust and easier to explain model without a large loss of useful information.

Finally, other machine learning techniques can be used. Like the random forest algorithm in our case, there are more machine learning techniques already adapted to the survival analysis methodology shown in this thesis, including Support Vector Machines (SurvivalSVM), boosting (CoxBoost) and bayesian models (bayesSurv).

In addition, as discussed in the 'Related Work' section, there are methods to simulate survival analysis without creating the full Kaplan-Meier curves. For example, the predictive survival and ROC scores can be found for a specific amount of time after admittance. This can be a preset time period (like the 2-year mortality used in this thesis) or even a time that will divide the patient set into equal sized groups of alive and dead ones, something that could not be done in this case since the number of censored patients exceeds the number of dead ones in all time periods.

References

[1] Methods for handling tied events on the cox proportional hazards model. Studia Oeconomica Posnaniensia, 2014.

[2] J. Bezdek, R. Hathaway, R. Howard, C. Wilson, and M. Windham. Local convergence analysis of a grouped variable version of coordinate descent. Journal of Optimization Theory and Applications, 54(3):471–477, 1987.

[3] L. Breiman. Bagging predictors. Machine Learning, 24(2):123–140, 1996.

[4] L. Breiman. Random forests. Machine Learning, 45(1):5–32, 2001.

[5] D. Collett. Modelling survival data in medical research. Chapman & Hall/CRC, Boca Raton, Fla, 2003.

[6] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. Medicine, 2000.

[7] D. R. Cox. Regression models and life-tables. Journal of the Royal Statistical Society. Series B (Methodological), 34(2):187–220, 1972.

[8] C. Gijsberts, H. den Ruijter, D. de Kleijn, A. Huisman, M. Ten Berg, and I. Hoefer. Hematological parameters improve prediction of mortality and secondary adverse events in coronary angiography patients: A longitudinal cohort study. Medicine, 2015.

[9] F. Harrell. Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis. Springer Series in Statistics. Springer International Publishing, 2015.

[10] F. E. Harrell, K. L. Lee, and D. B. Mark. Multivariable prognostic models: Issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Statistics in Medicine, 15(4):361–387, 1996.

[11] P. J. Heagerty and Y. Zheng. Survival model predictive accuracy and curves. Biometrics, 61(1):92–105, 2005.

[12] H. Ishwaran. Variable importance in binary regression trees and forests. Electron. J. Statist., 1:519–537, 2007.

[13] W. C. Levy, D. Mozaffarian, D. T. Linker, S. C. Sutradhar, S. D. Anker, A. B. Cropp, I. Anand, A. Maggioni, P. Burton, M. D. Sullivan, B. Pitt, P. A. Poole-Wilson, D. L. Mann, and M. Packer. The seattle heart failure model: Prediction of survival in heart failure. Circulation, 113(11):1424–1433, 2006.

[14] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Statist., 18(1):50–60, 03 1947.

[15] R. J. Mislevy. Statistical analysis with missing data. Journal of Educational Statistics, 16(2):150–155, 1991.

[16] P. P. Panahiazar, Taslimitehrani. Using ehrs and machine learning for heart failure survival analysis. Regression shrinkage and selection via the lasso: a retrospective: Series B (Statistical Methodology), 2011.

[17] P. P. Panahiazar, Taslimitehrani. Using ehrs and machine learning for heart failure survival analysis. Studies in Health Technology and Informatics, 2015.

[18] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. Ann. Statist., 26(5):1651–1686, 10 1998.

[19] D. Schoenfeld. Partial residuals for the proportional hazards regression model. Biometrika, 69(1):239–241, 1982.

[20] J. Tobin. Estimation of relationships for limited dependent variables. Econometrica, 26(1):24–36, 1958.

[21] V. Van Belle, K. Pelckmans, S. Van Huffel, and J. A. K. Suykens. Improved performance on high-dimensional survival data by application of survival-svm. Bioinformatics, 27(1):87–94, 2011.

[22] S. Van Buuren and K. Groothuis-Oudshoorn. Mice: Multivariate imputation by chained equations in r. Journal of Statistical Software, 45(3):1–67, 2011.

[23] C. Wu and M. Hamada. Experiments: Planning, Analysis, and Parameter Design Optimization. Wiley Series in Probability and Statistics. Wiley, 2000.

Appendices

A Statistics for numeric variables

Table 1: List and Statistics for the Numeric variables

Statistic	N	Mean	St. Dev.	Min	Max	Name
Age	2,445	64.221	11.327	18	93	Age (in years)
c_b_wbc	2,450	7.801	3.122	1.400	83.700	White blood cells
c_b_neu	2,450	5.006	2.369	0.430	28.600	Neutrophil amount
c_b_lym	2,444	1.937	1.709	0.220	76.840	Lymphocyte amount
c_b_mon	2,444	0.632	0.240	0.000	3.340	Monocyte amount
c_b_eos	2,450	0.181	0.165	0.000	3.490	eos
c_b_bas	2,449	0.040	0.030	0.000	0.760	bas
c_b_pneu	2,450	62.941	10.319	4.420	94.720	Neutrophil Percentage
c_b_plym	2,444	25.617	8.924	2.580	91.810	Leukocyte Percentage
c_b_pmon	2,444	8.428	2.621	0.040	33.900	Monocyte percentage
c_b_peos	2,450	2.445	1.927	0.020	22.320	procentueel aantal erytroblasten per 100 WBC
c_b_pbas	2,449	0.535	0.345	0.000	3.250	procentueel aantal basofiele granulocyten
c_b_rbco	2,450	4.460	0.484	2.690	6.200	RBC optical result
c_b_hb	2,450	8.469	0.965	5.170	11.930	HB result
c_b_mcv	2,450	89.833	5.006	56.390	111.210	MCV result
c_b_rdw	2,450	12.427	1.495	10.460	35.470	RDW result
c_b_mch	2,450	1.911	0.131	1.090	2.500	MCH result
c_b_mchc_usa	2,450	34.260	1.143	28.454	38.266	MCHC result (USA)
c_b_ht	2,450	39.831	4.343	25.290	55.980	c_b_ht
c_b_plto	2,443	232.601	74.251	8.090	928.980	PLT optical result
c_b_mpv	2,430	7.945	1.016	5.680	16.130	MPV result
c_b_pct	2,430	0.182	0.050	0.010	0.650	PCT result
c_b_pdw	2,431	16.203	0.711	11.480	21.510	PDW result
c_b_irf	2,444	0.284	0.172	0.030	3.990	Immature reticulocyte fraction Absolute reticulocyte count
c_b_namn	2,450	143.481	10.297	112.961	176.675	Neutrophil position , 0 grades
c_b_nimn	2,450	135.362	4.942	105.214	155.640	Neutrophil position , 7 grades
c_b_npmn	2,450	125.816	10.818	61.910	166.928	Neutrophil position, 90 grades
c_b_ndmn	2,450	28.262	3.527	8.960	43.322	Neutrophil position 90 grades polarized
c_b_nfmn	2,450	70.352	2.255	62.545	92.858	Neutrophil position FL3 canal
c_b_nacv	2,450	2.652	0.462	1.272	5.513	Neutrophil CV 0 grades
c_b_nicv	2,450	3.491	0.467	1.555	7.321	Neutrophil CV 7 grades
c_b_npcv	2,450	7.413	2.297	1.113	17.371	Neutrophil CV 90 grades
c_b_ndcv	2,450	15.004	1.549	5.805	36.163	Neutrophil CV 90 grades polarized
c_b_nfcv	2,450	7.802	1.503	1.111	12.361	Neutrophil CV FL3 canal
c_b_Lamn	2,450	100.976	3.980	82.949	116.841	Lymphocyte position 0 grades
c_b_Limn	2,450	75.705	2.670	62.068	98.715	Lymphocyte position 7 grades
c_b_Lacv	2,450	4.798	1.437	1.161	11.393	Lymphocyte CV 0 grades
c_b_Licv	2,450	4.848	0.974	1.904	8.646	Lymphocyte CV 7 grades
c_b_Pimn	2,450	145.972	5.750	117.611	167.626	Trombocyte position 7 grades
c_b_Ppmn	2,450	124.671	5.048	70.671	140.849	Trombocyte position 90 grades
c_b_Picv	2,450	17.049	1.624	13.691	38.381	Trombocyte CV 7 grades
c_b_Ppcv	2,450	13.843	5.141	10.802	93.498	c_b_Ppcv
c_b_rbcimn	2,450	180.722	5.681	0.000	185.840	c_b_rbcimn
c_b_rbcicv	2,450	1.691	0.219	0.000	2.729	c_b_rbcicv
c_b_rbcfmn	2,450	83.810	4.388	0.000	103.187	c_b_rbcfmn
c_b_rbcfcv	2,450	11.159	2.625	0.000	67.846	c_b_rbcfcv
c_b_MCHCr	2,450	30.315	1.705	0.000	35.529	c_b_MCHCr
c_b_HDW	2,450	7.306	1.235	0.000	12.545	c_b_HDW
c_b_MCHr	2,450	31.239	7.490	0.000	106.370	c_b_MCHr
c_b_MCVr	2,450	98.777	11.354	0.000	126.967	c_b_MCVr
c_b_pHPO	2,450	3.130	6.047	0.000	68.168	c_b_pHPO
c_b_pHPR	2,450	0.312	1.488	0.000	45.107	c_b_pHPR
c_b_pMAC	2,450	1.962	2.055	0.000	30.720	c_b_pMAC
c_b_pMIC	2,450	1.255	3.224	0.190	63.606	c_b_pMIC
c_b_prP	2,450	2.370	3.013	0.000	33.804	c_b_prP
duration	2,450	624.684	348.264	0	1,276	Time Alive
BMI	2,410	0.003	0.0004	0.001	0.005	Body Mass Index

B Concordance Indexes and Area Under Curve

Table 2: Baseline Models Concordance Indexes and AUC

	Cox regression - Validation using 1000 botstrap iterations	**Successful iterations in 1000 bootstrap samples**	**Random Forest using 10 trees**	**Area Under Curve in the Cox Model predicting 2 year mortality**
Baseline Model				
Single Imputation	0.7825	1000/1000	0.5293172	0.8029906
Multiple Imputation	0.8103	1000/1000	0.5288815	0.8030557
Baseline Model - Regularization				
Single Imputation	0.7926	1000/1000	0.5261406	0.7936398
Multiple Imputation	0.8086	1000/1000	0.5269771	0.7936618
Baseline Model - Regularization - Paired and squared terms				
Multiple Imputation	0.82005	1000/1000	0.8244383	0.8041328
Single Imputation	0.79765	1000/1000	0.4992876	0.8040851

Table 3: Labdata Models Concordance Indexes and AUC

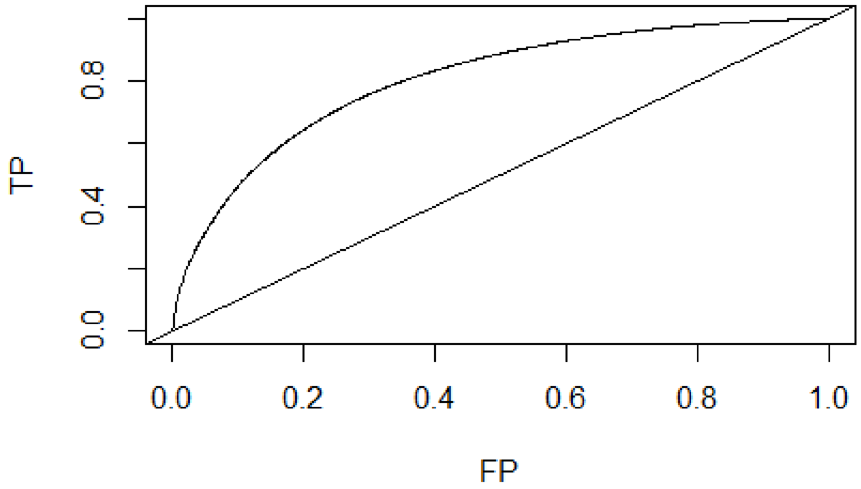
	Cox regression - Validation using 1000 botstrap iterations	**Successful iterations in 1000 bootstrap samples**	**Random Forest using 10 trees**	**Area Under Curve in the Cox Model predicting 2 year mortality**
Labdata Model				
Multiple Imputation	0.83675	223/1000	NA	0.687966
Single Imputation	0.7929	1000/1000	0.7599817	0.8160625
Labdata - Regularization				
Multiple Imputation	0.83015	187/1000	NA	0.8047997
Single Imputation	0,79405	1000/1000	NA	0,7542689
Labdata - Regularization - squared terms				
Multiple Imputation	0.82395	6/1000	0.5340478	0.529123
Single Imputation	0.8022	1000/1000	0.5685178	0.5340478
Labdata - Regularization - Paired and squared terms				
Multiple Imputation	0.82395	6/1000	0.5340478	0.529123
Single Imputation	0.8022	1000/1000	0.5685178	0.5340478

Table 4: Full Models Concordance Indexes

	Cox regression - Validation using 1000 botstrap iterations	Successful iterations in 1000 bootstrap samples	Random Forest using 10 trees	Area Under Curve in the Cox Model predicting 2 year mortality
Full Model				
Multiple Imputation	0.77185	146/1000	0,8614834	0.6771592
Single Imputation	0.5157	94/1000	NA	0.8523014
Full - Regularization from Single Imputation				
Multiple Imputation	0.85635	1000/1000	0,5954256	0.8382514
Single Imputation	0,83755	1000/1000	0,602731	0.8383571
Full - Regularization from Multiple Imputation				
Multiple Imputation	0.8642	274/1000	0,7440487	0.8529711
Single Imputation	0,8166	1000/1000	0,6759528	0.8523412
Full - Regularization from Multiple Imputation - Paired				
Multiple Imputation	0,9061	18/1000	0,7590678	0.6152721
Single Imputation	0,80885	1000/1000	0,5789405	0.8811398
Full - Regularization from Multiple Imputation - Squared				
Multiple Imputation	0,81235	1000/1000	0.7770257	0.7770257
Single Imputation	0,8084	1000/1000	0,5161702	0.7769471
Full - Regularization from Multiple Imputation - Paired and Squared				
Multiple Imputation	0,8146	1000/1000	NA	0.7978701
Single Imputation	0,7978	1000/1000	0.5642293	0.7987653

C ROC Charts for 2-Year Mortality

Figure 2: ROC Chart : Area Under Curve of Area Under Curve of Baseline Model



[!]

Figure 3: ROC Chart : Area Under Curve of Baseline Long

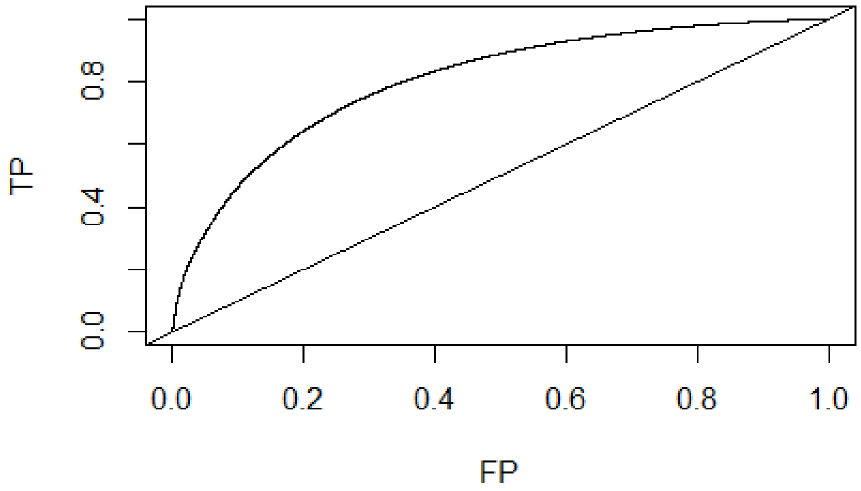


Figure 4: ROC Chart : Area Under Curve of Baseline Reduced

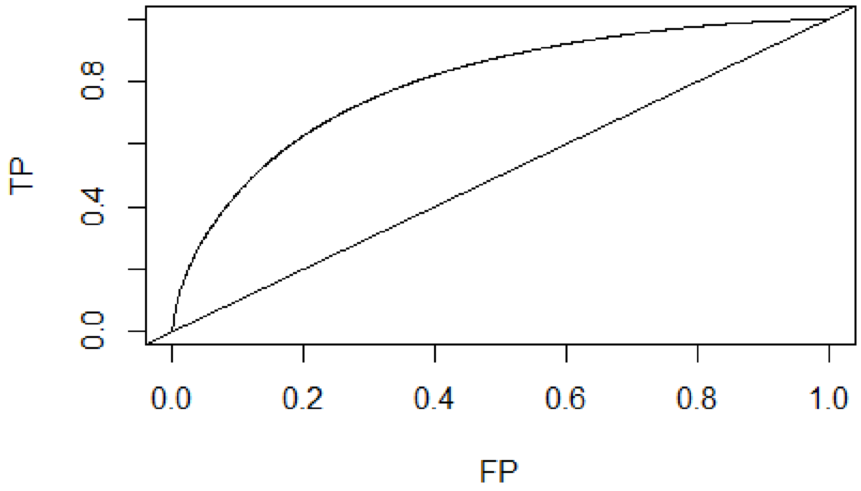
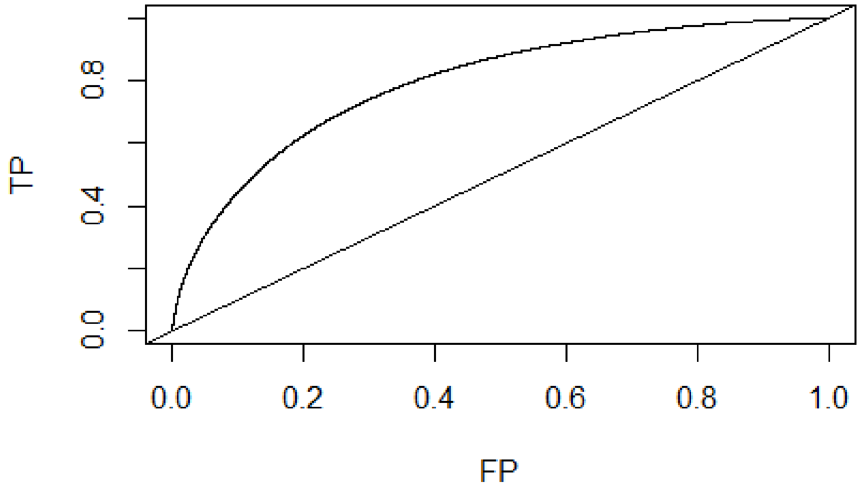


Figure 5: ROC Chart : Area Under Curve of Baseline Reduced Long



D Features of selected Proportional Hazards Models

Figure 6: ROC Chart : Area Under Curve of Baseline Reduced Paired Squared

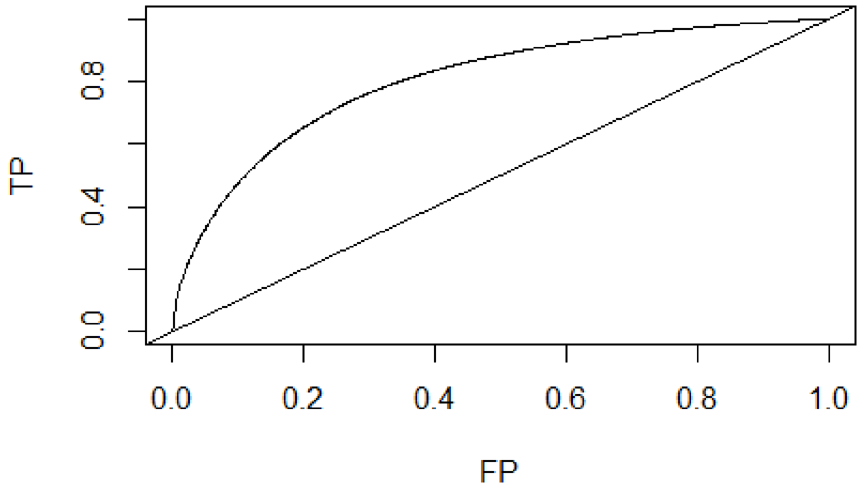


Figure 7: ROC Chart : Area Under Curve of Baseline Reduced Paired Squared Long

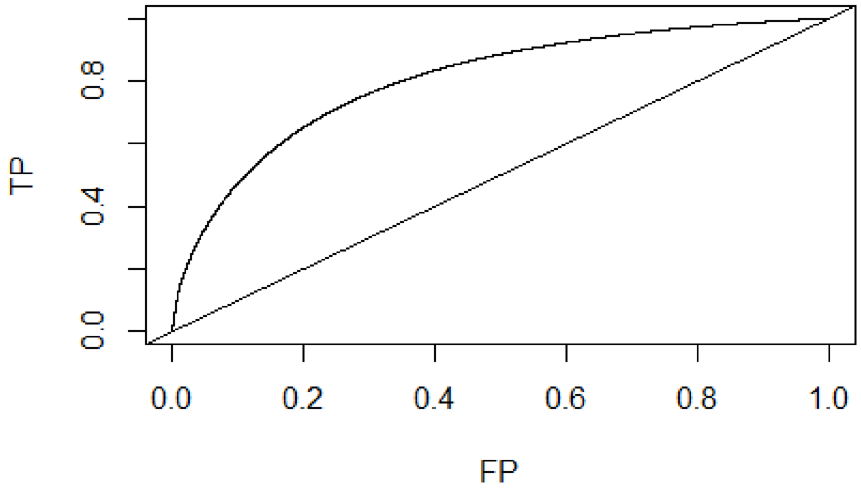


Figure 8: ROC Chart : Area Under Curve of LabData

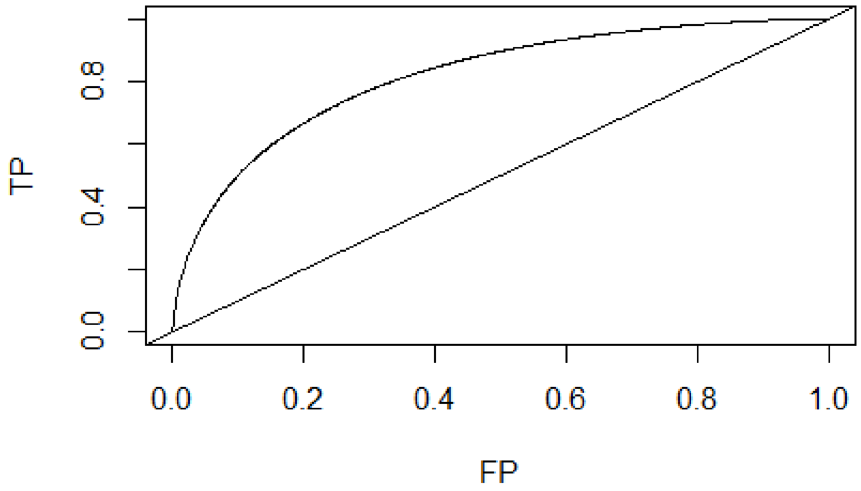


Figure 9: ROC Chart : Area Under Curve of Labdata Long

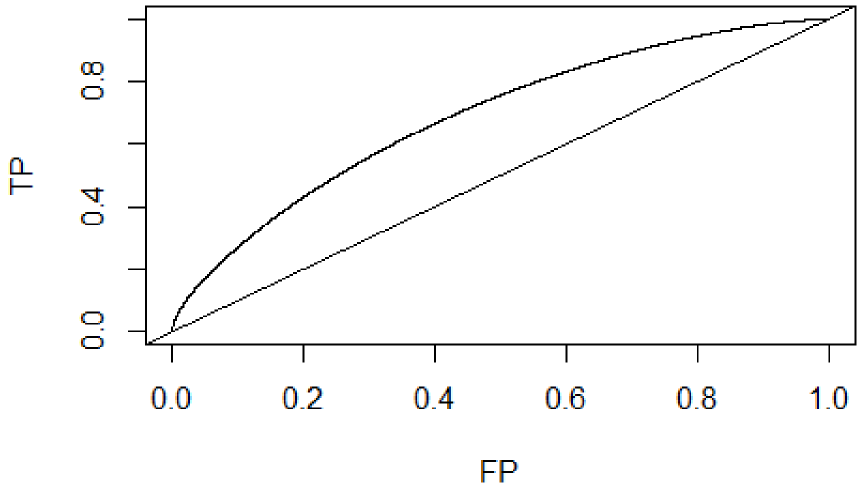


Figure 10: ROC Chart : Area Under Curve of Labdata Reduced

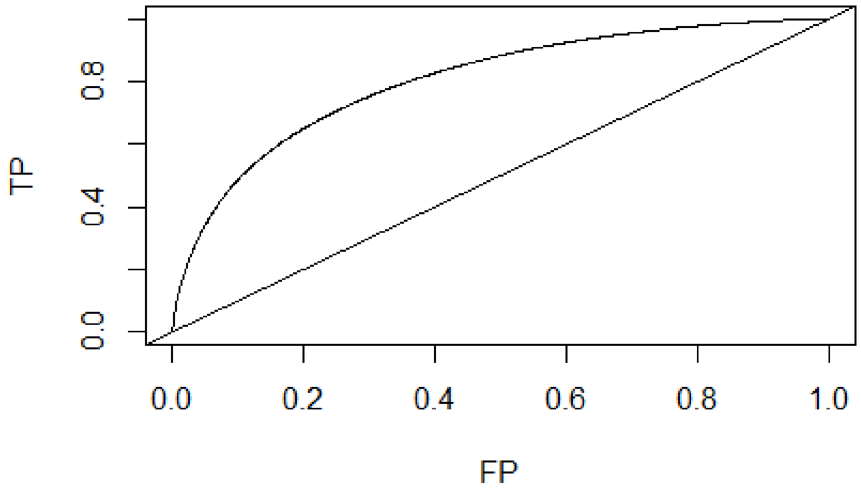


Figure 11: ROC Chart : Area Under Curve of Labdata Reduced Long

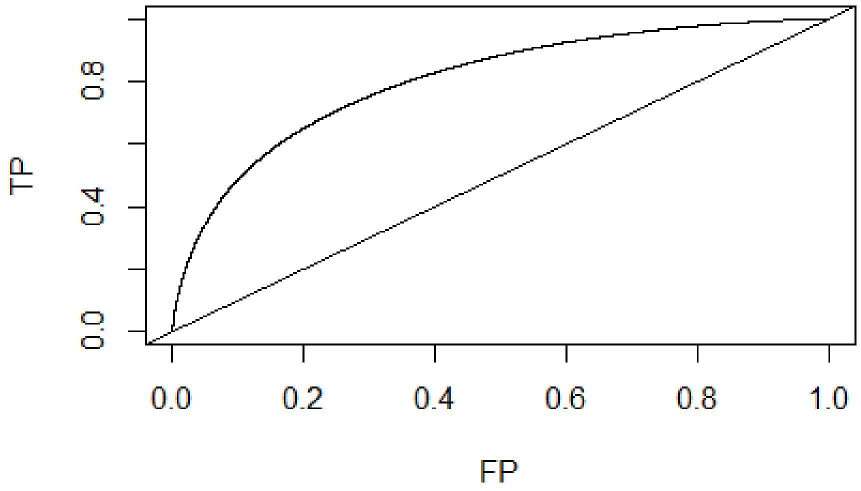


Figure 12: ROC Chart : Area Under Curve of Labdata Paired and Squared

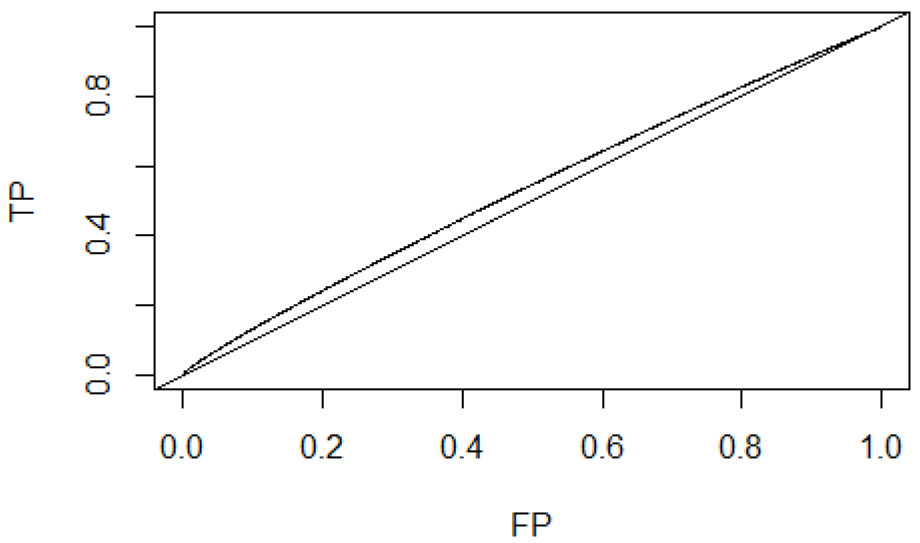


Figure 13: ROC Chart : Area Under Curve of Labdata Paired and Squared Long

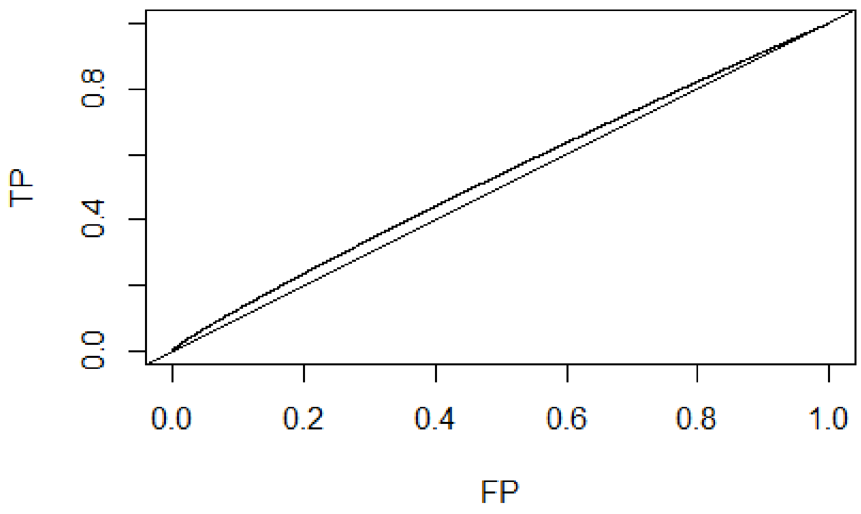


Figure 14: ROC Chart : Area Under Curve of Full model

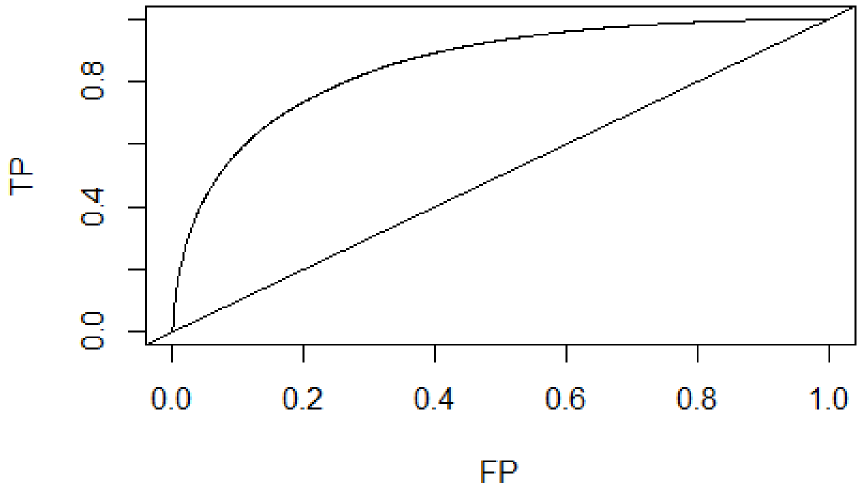


Figure 15: ROC Chart : Area Under Curve of Full model Long

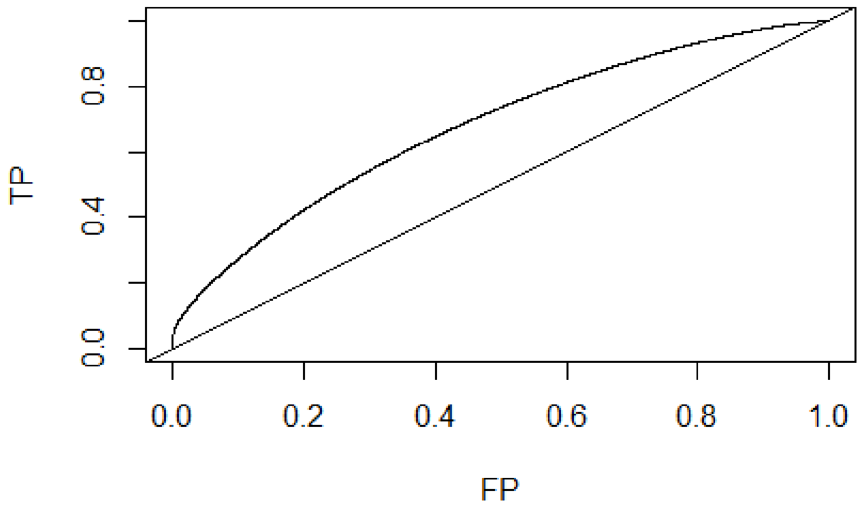


Figure 16: ROC Chart : Area Under Curve of Reduced (from short model)

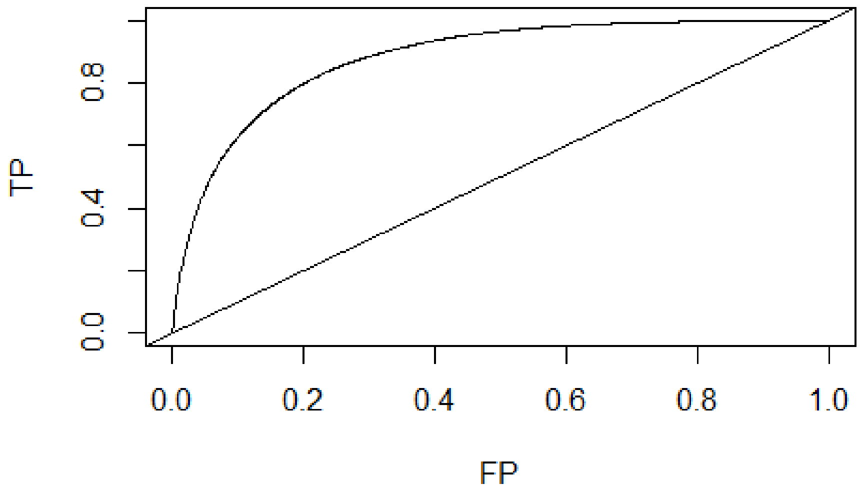


Figure 17: ROC Chart : Area Under Curve of Reduced (from short model) Long

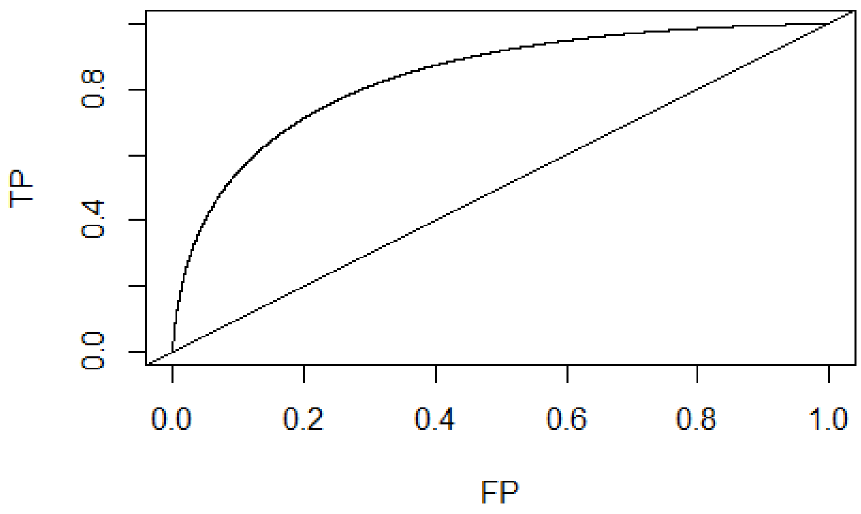


Figure 18: ROC Chart : Area Under Curve of Reduced (from long model)

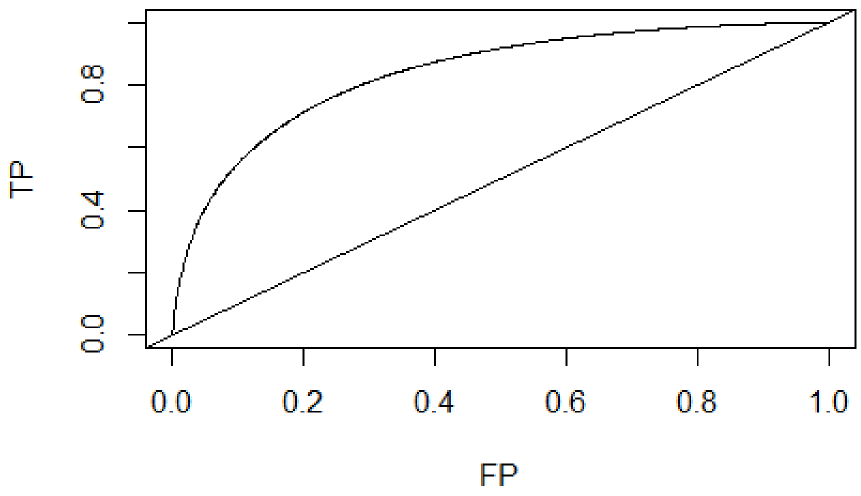


Figure 19: ROC Chart : Area Under Curve of Reduced (from long model) Long

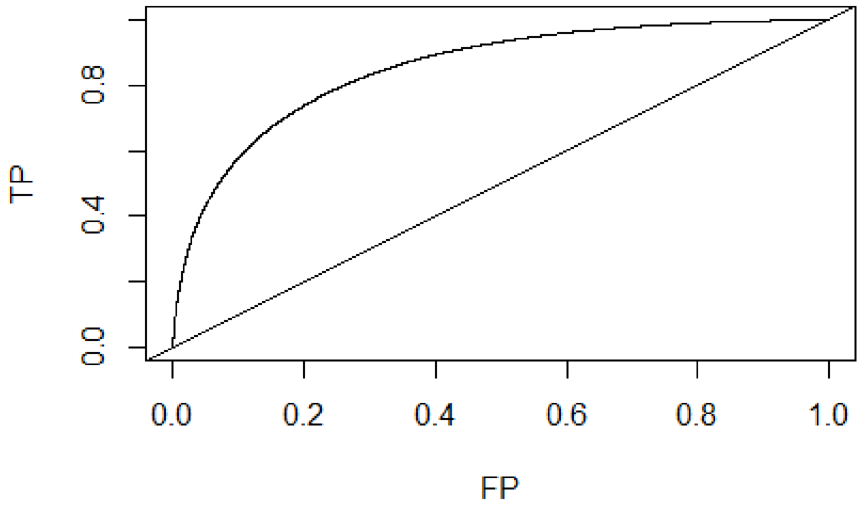


Figure 20: ROC Chart : Area Under Curve of Paired

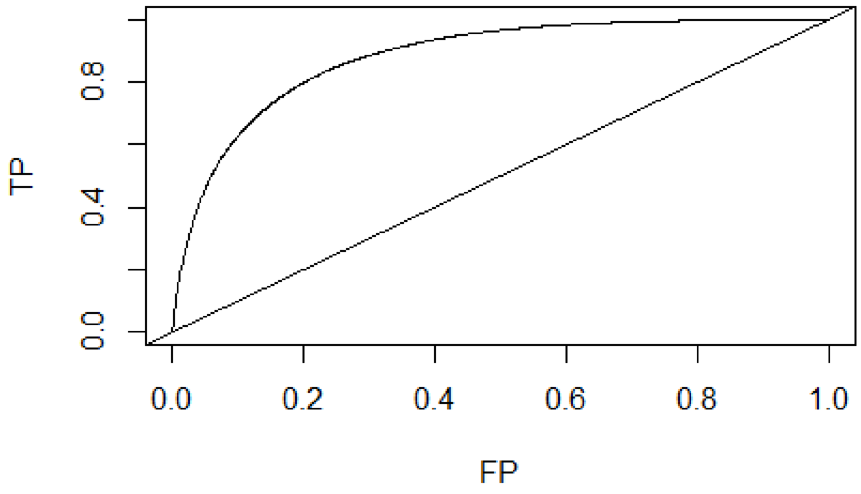


Figure 21: ROC Chart : Area Under Curve of Paired Long

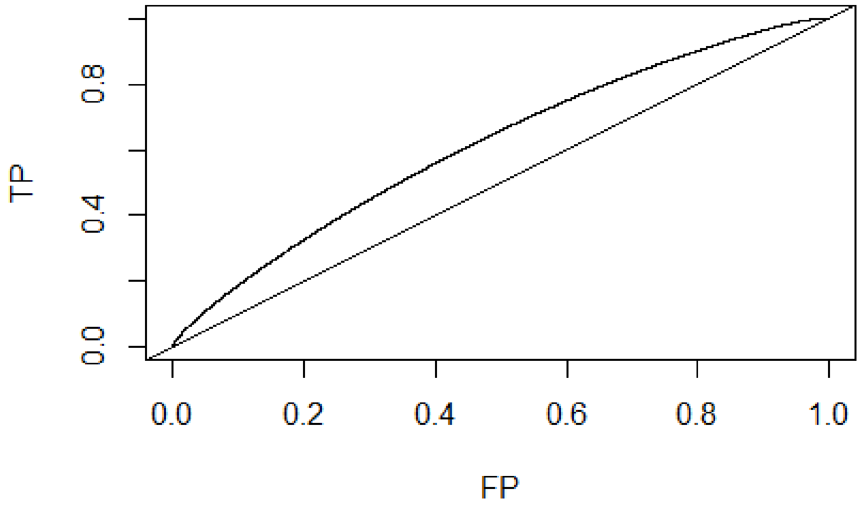


Figure 22: ROC Chart : Area Under Curve of Squared

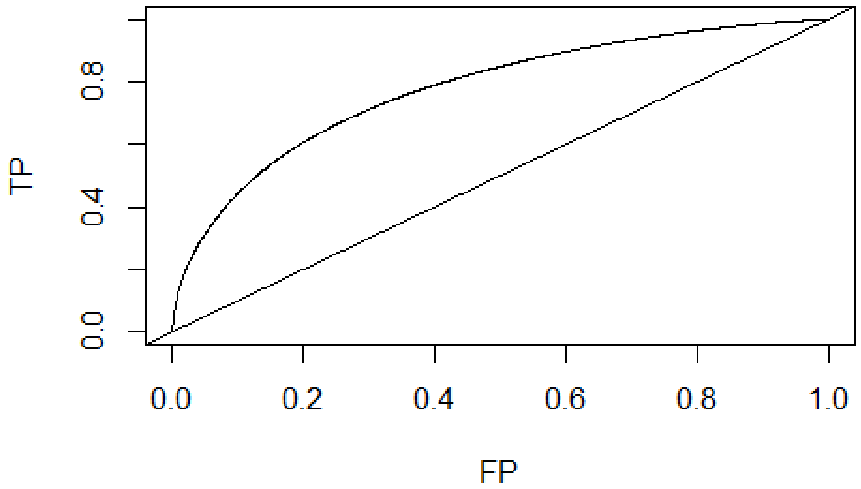


Figure 23: ROC Chart : Area Under Curve of Squared Long

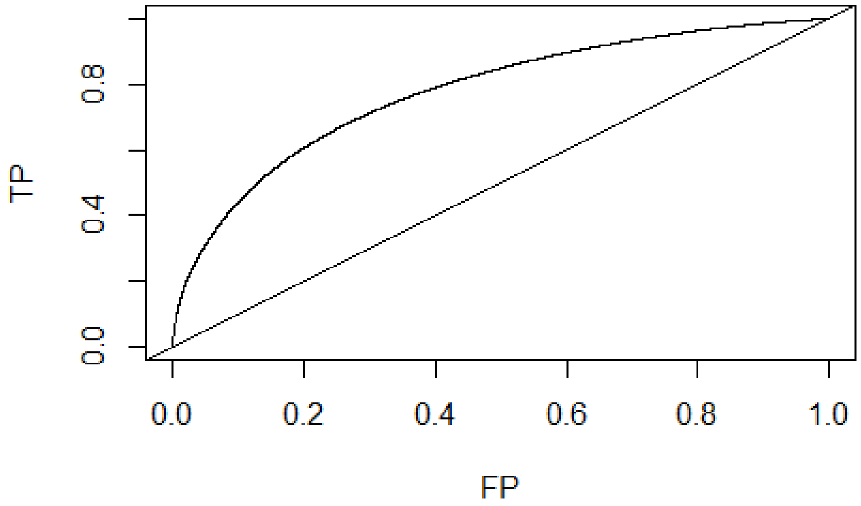


Figure 24: ROC Chart : Area Under Curve of Paired and Squared

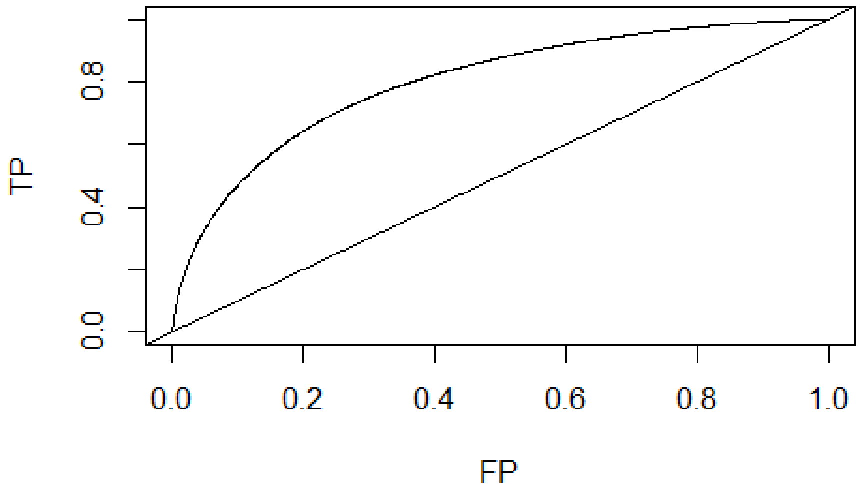
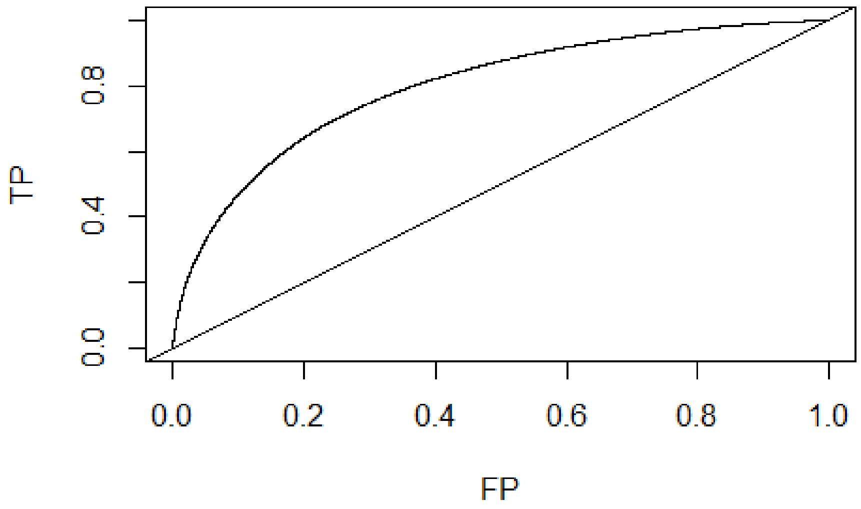
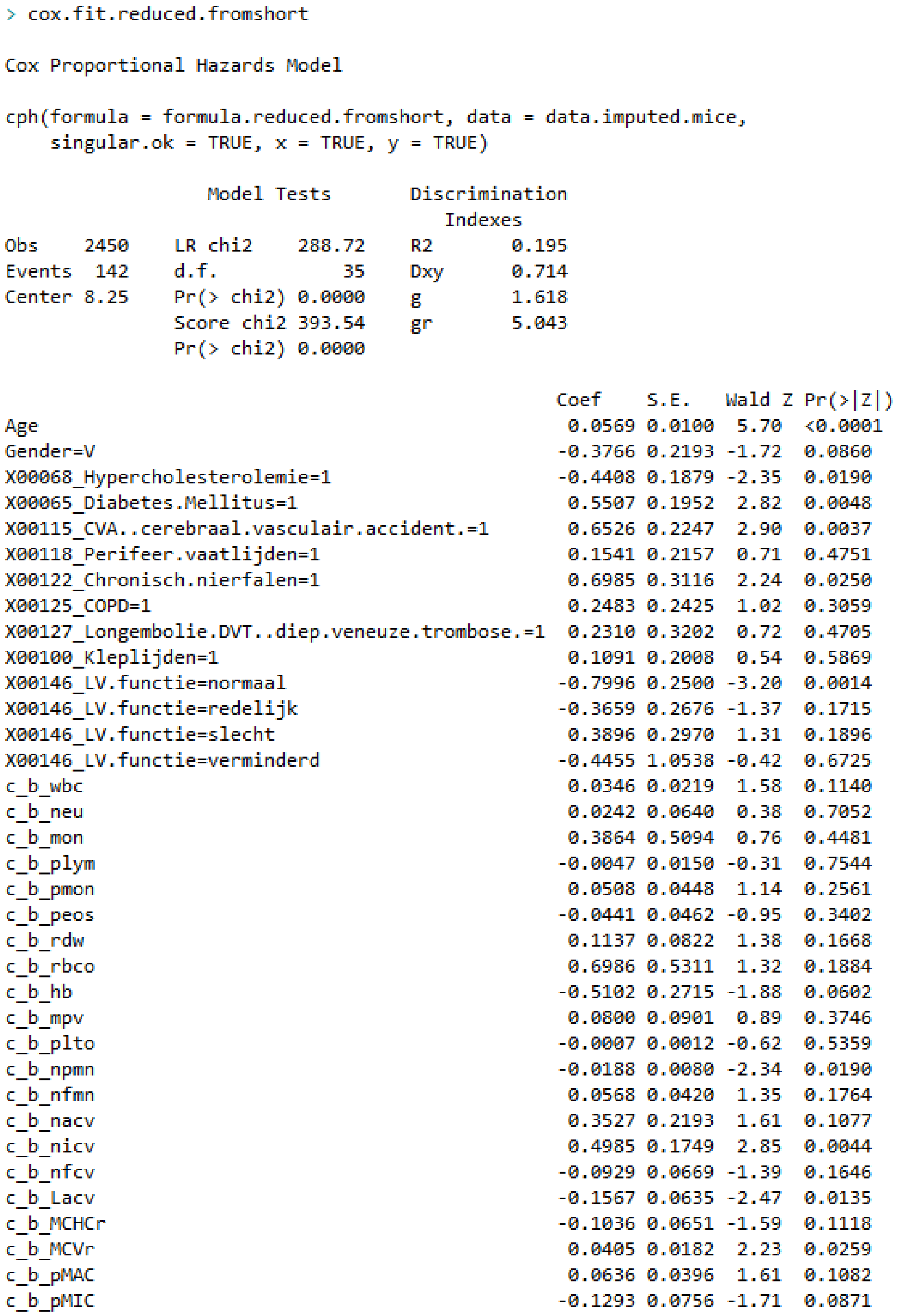


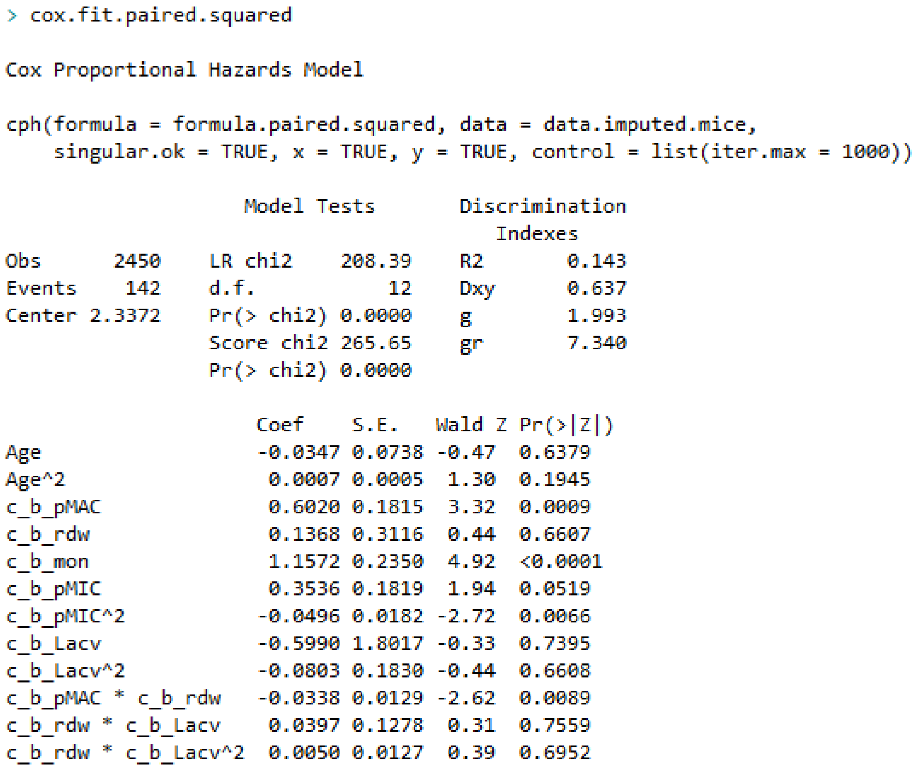
Figure 25: ROC Chart : Area Under Curve of Area Under Curve of Paired and Squared Long





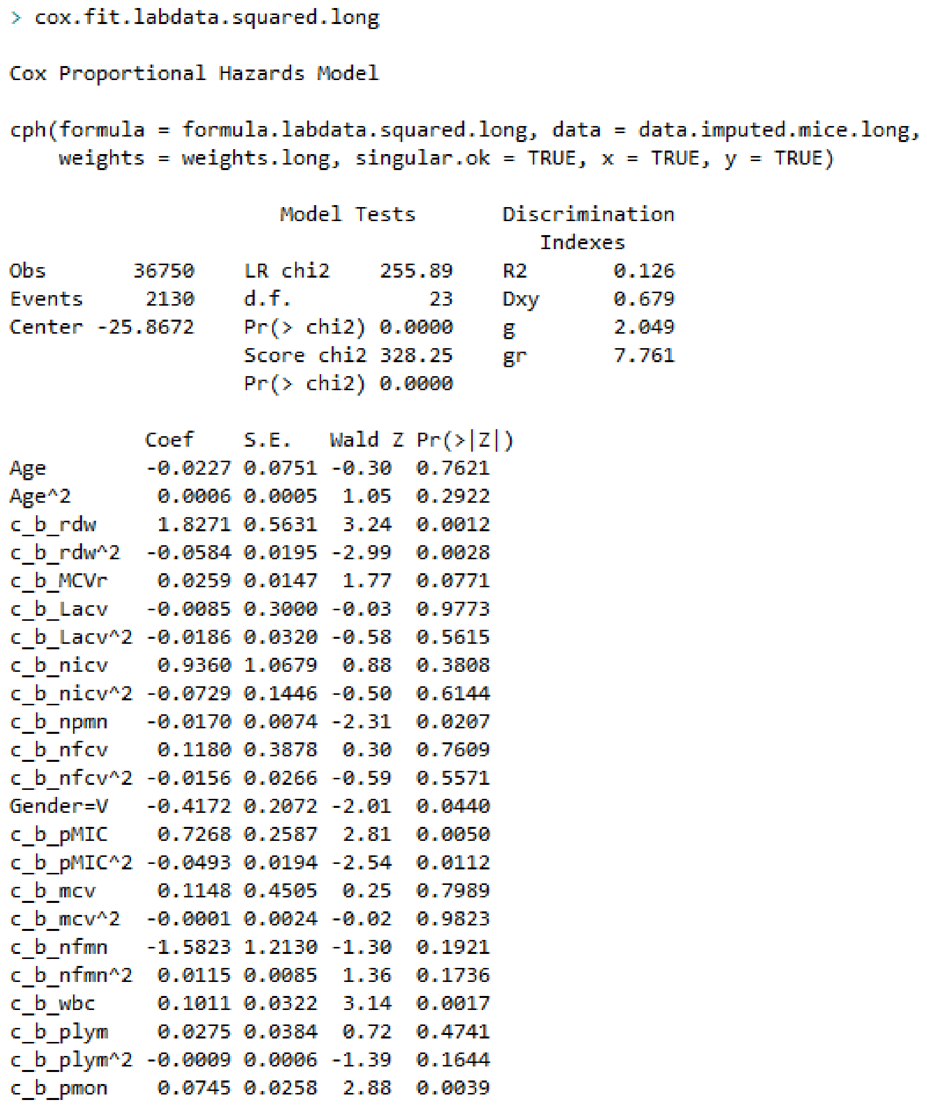
```
> cox.fit.reduced.fromshort

Cox Proportional Hazards Model

cph(formula = formula.reduced.fromshort, data = data.imputed.mice,
    singular.ok = TRUE, x = TRUE, y = TRUE)

                  Model Tests       Discrimination
                                       Indexes
Obs      2450   LR chi2     288.72   R2       0.195
Events    142   d.f.            35   Dxy      0.714
Center 8.25     Pr(> chi2) 0.0000    g        1.618
                Score chi2 393.54    gr       5.043
                Pr(> chi2) 0.0000

                                                  Coef    S.E.   Wald Z Pr(>|Z|)
Age                                               0.0569 0.0100  5.70  <0.0001
Gender=V                                         -0.3766 0.2193 -1.72  0.0860
X00068_Hypercholesterolemie=1                    -0.4408 0.1879 -2.35  0.0190
X00065_Diabetes.Mellitus=1                        0.5507 0.1952  2.82  0.0048
X00115_CVA..cerebraal.vasculair.accident.=1       0.6526 0.2247  2.90  0.0037
X00118_Perifeer.vaatlijden=1                      0.1541 0.2157  0.71  0.4751
X00122_Chronisch.nierfalen=1                      0.6985 0.3116  2.24  0.0250
X00125_COPD=1                                     0.2483 0.2425  1.02  0.3059
X00127_Longembolie.DVT..diep.veneuze.trombose.=1  0.2310 0.3202  0.72  0.4705
X00100_Kleplijden=1                               0.1091 0.2008  0.54  0.5869
X00146_LV.functie=normaal                        -0.7996 0.2500 -3.20  0.0014
X00146_LV.functie=redelijk                       -0.3659 0.2676 -1.37  0.1715
X00146_LV.functie=slecht                          0.3896 0.2970  1.31  0.1896
X00146_LV.functie=verminderd                     -0.4455 1.0538 -0.42  0.6725
c_b_wbc                                           0.0346 0.0219  1.58  0.1140
c_b_neu                                           0.0242 0.0640  0.38  0.7052
c_b_mon                                           0.3864 0.5094  0.76  0.4481
c_b_plym                                         -0.0047 0.0150 -0.31  0.7544
c_b_pmon                                          0.0508 0.0448  1.14  0.2561
c_b_peos                                         -0.0441 0.0462 -0.95  0.3402
c_b_rdw                                           0.1137 0.0822  1.38  0.1668
c_b_rbco                                          0.6986 0.5311  1.32  0.1884
c_b_hb                                           -0.5102 0.2715 -1.88  0.0602
c_b_mpv                                           0.0800 0.0901  0.89  0.3746
c_b_plto                                         -0.0007 0.0012 -0.62  0.5359
c_b_npmn                                         -0.0188 0.0080 -2.34  0.0190
c_b_nfmn                                          0.0568 0.0420  1.35  0.1764
c_b_nacv                                          0.3527 0.2193  1.61  0.1077
c_b_nicv                                          0.4985 0.1749  2.85  0.0044
c_b_nfcv                                         -0.0929 0.0669 -1.39  0.1646
c_b_Lacv                                         -0.1567 0.0635 -2.47  0.0135
c_b_MCHCr                                        -0.1036 0.0651 -1.59  0.1118
c_b_MCVr                                          0.0405 0.0182  2.23  0.0259
c_b_pMAC                                          0.0636 0.0396  1.61  0.1082
c_b_pMIC                                         -0.1293 0.0756 -1.71  0.0871
```

Figure 26: Model After regularization started from the short dataset, implemented on the short dateset.



```
> cox.fit.paired.squared

Cox Proportional Hazards Model

cph(formula = formula.paired.squared, data = data.imputed.mice,
    singular.ok = TRUE, x = TRUE, y = TRUE, control = list(iter.max = 1000))

                    Model Tests       Discrimination
                                         Indexes
Obs       2450    LR chi2     208.39    R2       0.143
Events     142    d.f.            12    Dxy      0.637
Center 2.3372     Pr(> chi2) 0.0000    g        1.993
                  Score chi2 265.65    gr       7.340
                  Pr(> chi2) 0.0000

                      Coef    S.E.   Wald Z Pr(>|Z|)
Age                   -0.0347 0.0738 -0.47  0.6379
Age^2                  0.0007 0.0005  1.30  0.1945
c_b_pMAC               0.6020 0.1815  3.32  0.0009
c_b_rdw                0.1368 0.3116  0.44  0.6607
c_b_mon                1.1572 0.2350  4.92  <0.0001
c_b_pMIC               0.3536 0.1819  1.94  0.0519
c_b_pMIC^2            -0.0496 0.0182 -2.72  0.0066
c_b_Lacv              -0.5990 1.8017 -0.33  0.7395
c_b_Lacv^2            -0.0803 0.1830 -0.44  0.6608
c_b_pMAC * c_b_rdw    -0.0338 0.0129 -2.62  0.0089
c_b_rdw * c_b_Lacv     0.0397 0.1278  0.31  0.7559
c_b_rdw * c_b_Lacv^2   0.0050 0.0127  0.39  0.6952
```

Figure 27: Model After regularization started from the long dataset, after Stepwise Regression for interaction and squared terms



```
> cox.fit.labdata.squared.long

Cox Proportional Hazards Model

cph(formula = formula.labdata.squared.long, data = data.imputed.mice.long,
    weights = weights.long, singular.ok = TRUE, x = TRUE, y = TRUE)

                      Model Tests       Discrimination
                                           Indexes
Obs       36750    LR chi2     255.89    R2       0.126
Events     2130    d.f.            23    Dxy      0.679
Center -25.8672    Pr(> chi2) 0.0000    g        2.049
                   Score chi2 328.25    gr       7.761
                   Pr(> chi2) 0.0000

           Coef    S.E.   Wald Z Pr(>|Z|)
Age        -0.0227 0.0751 -0.30  0.7621
Age^2       0.0006 0.0005  1.05  0.2922
c_b_rdw     1.8271 0.5631  3.24  0.0012
c_b_rdw^2  -0.0584 0.0195 -2.99  0.0028
c_b_MCVr    0.0259 0.0147  1.77  0.0771
c_b_Lacv   -0.0085 0.3000 -0.03  0.9773
c_b_Lacv^2 -0.0186 0.0320 -0.58  0.5615
c_b_nicv    0.9360 1.0679  0.88  0.3808
c_b_nicv^2 -0.0729 0.1446 -0.50  0.6144
c_b_npmn   -0.0170 0.0074 -2.31  0.0207
c_b_nfcv    0.1180 0.3878  0.30  0.7609
c_b_nfcv^2 -0.0156 0.0266 -0.59  0.5571
Gender=V   -0.4172 0.2072 -2.01  0.0440
c_b_pMIC    0.7268 0.2587  2.81  0.0050
c_b_pMIC^2 -0.0493 0.0194 -2.54  0.0112
c_b_mcv     0.1148 0.4505  0.25  0.7989
c_b_mcv^2  -0.0001 0.0024 -0.02  0.9823
c_b_nfmn   -1.5823 1.2130 -1.30  0.1921
c_b_nfmn^2  0.0115 0.0085  1.36  0.1736
c_b_wbc     0.1011 0.0322  3.14  0.0017
c_b_plym    0.0275 0.0384  0.72  0.4741
c_b_plym^2 -0.0009 0.0006 -1.39  0.1644
c_b_pmon    0.0745 0.0258  2.88  0.0039
```

Figure 28: Model After regularization started from the long dataset, after Stepwise Regression for interaction and squared terms