ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

University of Piraeus

Department of Digital Systems

Postgraduate Programme (M.Sc.) in Big Data & Analytics

Title: Analytics on clinical and gene expression data to provide risk stratification predictions for cancer subtypes.

Marinos George - AM: ME1817

<u>**Supervisor**</u>

Dimosthenis Kyriazis

Assistant Professor

Table of contents

Abstract

Survival analysis is not only applicable to biomedical problems for time until death estimation and recent studies have shown that it is a powerful tool for risk stratification that can be used in various sectors. Survival analysis is a well-established statistical technique and last years there have been many studies that combined survival analysis with machine learning algorithms in order to capture non linear relationships among features and to obtain better performances. The usage of different data modalities has been proven to be effective for the enhancement of the performance of machine learning models. In this study even though our main purpose was not a classification task, we utilize two different data modalities (clinical and genes expression data) for risk stratification of a group of patients with cancer according to their latent cancer subtypes. Firstly, we utilize clinical data to extract features that are associated with survival time and with the presence or absence of the event of death. Next use these features selected to identify latent groups between patients and when this is done, we used the labels as ground truth to identify a subset of survival-associated genes. We tested three different features selection and dimensionality reduction techniques for genes expression data to examine if this will cause any differences in our results. We finally applied classifiers to the genes subset identified, and we tried to predict in which sub cancer group category would a future patient belong to. Implementing this approach, we expect that the identified subgroups are biologically meaningful or in other words they differ in terms of survival. The two contributions of the proposed approach were a) the discovery of a meaningful subset of genes that are associated with the survival of the patient and b) with the usage of this approach future patients will be able to be accurately categorized in a survival risk group even if the available data are not labeled from the very beginning.

Acknowledgements

Firstly, I would like to express my thanks to my supervisor Dr. Dimosthenis Kyriazis for the inspiration, support, and his illuminating instructions he offered during writing my thesis.

I would also express my deepest sincere gratitude to my family and friends for their unconditional support and encouragement through both the highs and lows of my academic pursuits.

List of Tables

List of figures

Chapter 1

1.1 Introduction

In the era of digital disruption, the daily production of giant amounts of data has led computer scientists to develop many efficient algorithms to control those data and extract valuable knowledge from them. Computers science subfield, namely machine learning, is about the scientific study and development of algorithms which are empowered with math and statistical power and perform a specific task without using straightforward instructions counting on patterns and inference instead (Mitchell 2017).

Machine learning algorithms vary from more naive algorithms to algorithms with high complexity and additionally are utilized in a good sort of applications from recommendation systems for advertisements to computer vision and healthcare. They are able to access large amount of data, learn complex associations and therefore result to accurate predictions (or even decisions) without being explicitly programmed to perform the task.

Because of their effective performance machine learning has been widely utilized to solve tasks like regression that statistical science has already handled. Survival analysis or time to event analysis is one among these tasks (Liu 2012). Due to the incompleteness of the data collected, in many studies missing or incomplete data are often excluded from the phase of training of a machine learning algorithm. However, using survival analysis we are able to handle data with incomplete information (*censored data*).

In terms of biomedical research Survival analysis is an extremely useful tool. Computer scientists and medical researchers don’t only consider about the survival outcome of a patient but also utilize a rich repository of machine learning and deep learning techniques to stratify level of morbidity in each subject in a clinical study and estimate survival probability (Singh et al. 2011).

Additionally, a big chapter of data analysis methodology generally is data preprocessing. Data scientists have to handle every data peculiarity that they possibly face. Last decades with widely usage of gene expression data in medical research, analysts have to deal with high dimension data (Pölsterl et al. 2016). One can find affluent literature for scientific methods in the field of data analysis for the selection of most relevant features in a data set in order to keep valuable information and increase model accuracy. Too many studies focus on genes expression data analysis and especially in dimensionality reduction and feature selection for gene expression data.

1.2 Motivation

Survival analysis plays an important role in clinical research. When someone is diagnosed with cancer, several parameters are used to assess the patient's survival profile. An accurate estimation of patients' survival probability may potentially have a big number of utilities for clinicians such as better hospitalization (Bussy et al. 2019), more accurate diagnosis etc.

Having a certain form of cancer is often thought as a single disease but recent studies have been proven that multiple subtypes of a specific cancer may exist in a patients' cohort suffering i.e., from breast cancer. In addition, clinical research has been shown that two seemingly alike tumors may be completely different diseases at the molecular profile of the tumor (Kittaneh et al. 2013).

Consequently, disease subtypes identification may be very useful in clinical domain in order to improve clinical management, obtain different prognosis and responses to certain therapies even to patients that have diagnosed with a certain type of disease (i.e., breast cancer).

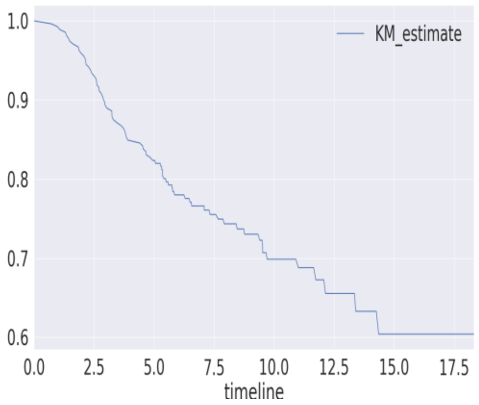


Figure 1.1.a

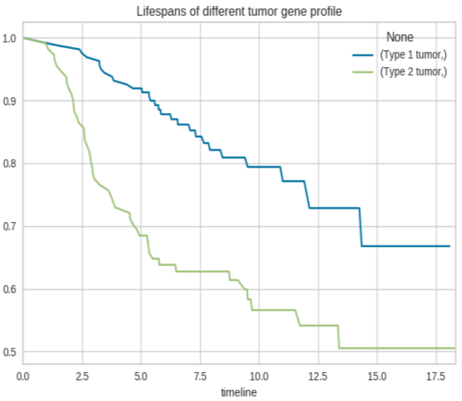


Figure 1.1.b

In the Figure 1.1.a above is shown a Kaplan Meier Curve of a patients' cohort that have diagnosed with breast cancer. In (KM) Curve we can obtain patient's survival probability as the time passing by. Figure 1.1.b illustrates the (KM) Curve of the same patients' cohort after some of them have been identified with sub cancer type 1 and sub cancer type 2. As we can obtain there is a significantly difference between two survival curves and this denotes the importance of accurate estimation of survival probability of patients and further research of subtype existence.

Cancer subtypes identification provide clues for differences in patients survival profile (Yang et al. 2019; Sinkala et al. 2020). Last decades there have been conducted plenty of studies related to accurate patient diagnoses of cancer subtypes with the subtypes to be known in advance. However, in real world this is not a very common case and there is the potential neither the cancer subtypes nor the number of subtypes to be already known.

Scientific community has been using various data modalities to identify cancer subtypes in the diagnosis of future patients (Begg et al. 2010; Kuijjer et al. 2018) not only clinical data but also gene expression data, DNA methylation, miRNA sequence data, histopathological images etc. However, this problem remains a largely open research question and further research is required.

A lot of scientific papers can be found in literature trying to solve this problem, many of them fall into a class of statistical procedures **(**Witten et al. 2010; Lee et al. 2019) and others that utilize a machine learning approach. Statistical procedures have achieved varying degrees of success, however their effectiveness is limited in terms of handling high dimensional data and selecting important features. Additionally, the large amounts of data collected in such kind of research in parallel with high dimensionality of gene expression data or even more clinical images (Cheerla et al. 2019) that must be processed have become an obstacle for scientists that have been utilized only statistical techniques.

Nevertheless, machine learning approaches in identification of subtypes in a certain cancer are divided into three main categories: supervised, fully unsupervised and semi-supervised (Roder et al. 2019; Cheerla et al. 2019; Livieris et al. 2018) techniques. Relevant scientific studies for these categories are mentioned in Chapter 3 but generally, the necessity of application of supervised techniques are obtained when the cancer subtypes have been already identified from clinicians and the main purpose of analyst is to build a classifier that have the ability to accurately classify a new patient in the right subtype of cancer. Fully unsupervised approaches are applied in biomedical data when there is no prior knowledge of certain cancer's subcategories and research. This kind of studies focuses on identify groups of patients that their clinically stage is more relevant than others that belong to different group. Semi-supervised learning field fall between supervised and unsupervised learning since utilizes both supervised and unsupervised techniques. Semi-supervised approaches are limited in literature, but there seem to be very promising in solving such kink of problems and thus this thesis explores the performance of a semi-supervised approach for cancer subtypes identification without prior knowledge of the existence of cancer subtypes or the number of them.

This thesis describes a procedure that utilizes both gene expression data and clinical data to conduct dimensionality reduction and survival prediction utilizing machine learning and deep learning techniques. Also, we analyze breast cancer dataset in order to conduct accurate identification of cancer subtypes, if they exist, to ensure the robustness of our approach. Finally, a complement goal for us is to explore a range of possibilities for future work on designing a more powerful tool for diagnosing and stratifying cancer risk utilizing both statistical and machine learning techniques.

1.3 Research Questions

This study focuses on utilizing both clinical and genes expression data for constructing a pipeline procedure statistical and machine learning algorithms that can accurately identify subgroups in a patients' cohort that have statistically significant differences with regards of survival.

How can we extract meaningful (survival) clusters not only in terms of feature characteristics but also in terms of lifetime distribution of each subject?

Censoring is a well-known concept in Survival Analysis. Right and left censoring are types of data incompleteness that Survival Analysis models and able to handle unlike other statistical techniques (e.g.: regression). Utilizing those type of data and trying to perform clustering is not a trivial task because of this peculiarity and thus conventional clustering algorithms like K-means, DB-scan etc., will not result in clusters that would be meaningful in terms of survivability. In this thesis we study a semi-supervised approach to efficiently cluster our data and obtain clusters of “high risk”, and “low risk” patients.

How can we discover genes that are associated with the lifetime distribution of each patient?

Significant genes identification is a very important task in bioinformatics. Despite the fact that genes identification is important the vast majority of studies in literature try to identify genes that are associated with a specific type of disease (e.g., cancer). In our study we try to identify genes that are associated with the survival outcome of the patient and not with a specific type of disease. This is a challenging task since information about the survival of the patient is not directly included in genes datasets.

How can we utilize statistical and machine learning techniques for risk stratification of various subtypes of a certain cancer in unlabeled data?

Since cancer affected patients risk stratification is a main challenge in biomedical research there are numerous statistical procedures in literature trying to solve this problem utilizing unsupervised learning. Recently Zhang (Zhang et. al 2016) in his study outperformed existing statistical techniques in cancer subtypes identification in a semi-supervised manner. He also suggested an interesting approach that has the potential for major improvements if other more sophisticated machine learning techniques be used for the same purpose in a similar manner. During the development of this thesis, we tested several machine learning approaches to provide a slightly improved solution. We present and explain those that returned the most significant results.

1.4 Approach

Cancer subtypes identification involves the discovery of meaningful group of objects that hold vital intimation for survival time. The objective in this kind of research is to identify a set of latent class membership that are associated with the phenotype of interest.

In contrast with many existing methods that the initial step was to select a relevant subset of features in this study we apply a diagnostic procedure that make use of both clinical and genes expression data. Having done so, then it is applicable to predict survival of future patients.

Though many studies utilize few labeled data and thus have restricted knowledge about patients' cancer category or subtype. Since this is not common in real world cases and due to the data availability, in this study such kind of information was not included. Since information about class label was hidden, existing feature selection methods were not applicable. We tackled this problem by firstly selecting a significant subset of clinical features with respect to survival time and the presence or absence of the event of interest (death). After we tried to find the optimal number of clusters utilizing well-established techniques for optimal number of clusters identification with the usage of the previous selected clinical features which are representative and significantly associated with the survivability of the individuals.

Then, unsupervised learning was applied to the subgroup of clinical features selected in the first step with the optimal number of groups that discovered in the second step. Next using the labels relied on clinical data along with feature selection or dimensionality reduction techniques and taking into account that identification of genes that are associated with the survival outcome is a very important task in bioinformatics, we were able to identify a subset of genes that are significantly associated with survival outcome of the patient.

Finally in order to validate our approach, different classifiers were applied to this subset of genes that was identified in order to check if using only those genes we are able to accurately predict the cancer subtype for a future group of patients. A similar approach was introduced in 2016 by Zhang (Zhang 2016), and on this occasion we applied various and more sophisticated feature selection and dimensionality reduction techniques to obtain if any improvements in the final results will occur. Thanks to rich repository of feature selection and dimensionality reduction techniques, we slightly improved the outcome. In this study we present these techniques that had the best performance.

1.5 Thesis Organization

This thesis is organized as follows:

In Chapter 2 fundamental of survival analysis will be presented. This includes basic concept and basic terminology and notation. Furthermore, chapter 3 is about related work in the literature and various approaches for dealing with high dimensional data in terms of cancer subgroups discovery. In Chapter 4 are analyzed all methods, techniques and algorithms that are utilized in this study. In Chapter 5 we present the application of our method a real-world dataset (NKI breast cancer dataset) and discuss the results. Chapter 6 is the conclusion of this thesis and suggestions for future work

Chapter 2

2.1 Survival Analysis Overview

Survival analysis is a field in statistics that is used to predict the time until a particular event of interest happens. The field was first created in terms of medical research and the purpose was to model a patient's survival, hence the term "survival analysis".

It is a type of regression problem (one wants to predict a continuous value), but with a twist. It differs from traditional regression by the fact that parts of the training data can only be partially observed – they are censored.

Even though survival analysis started for medical research purposes, it is broadly used in several domains e.g., computer science for predict when a Devices will failure, in Healthcare for Rehospitalization, Disease recurrence and disease survival, at marketing and sales domain for Customer Lifetime Value etc.

For example, in health-related studies, typical research questions are like:

- What is the impact of certain clinical characteristics on patients' survival?
- What is the probability that a person survives 3 years?
- Are there differences in survival between groups of patients?

It is often used to identify different subgroups of subjects in a study and how they are alternate under various circumstances.

Table 2.1 shows a few examples of real-world applications that survival analysis is used.

Application	**Event of Interest**	**Estimation**
Healthcare	1) Disease Survival 2) Rehospitalization	**e.g.,** likelihood of death within t days from the time someone got sick.
Reliability	Device failure	**e.g.,** Likelihood of a device being failed within t days.
Financial Industry	Purchase Behavior	**e.g.,** Likelihood of a customer purchasing from a given service supplier within t days
Economics	Unemployment duration	**e.g.,** Likelihood of a person finding a new job within t days
Banking	Credit scoring	**e.g.,** Likelihood of a customer pay back a loan

Insurance Industry	Claims occurring	**e.g.,** An Insurance Company interest is when their customers will die

Survival analysis differs from both classification and regression problems. In classification tasks (e.g., logistic regression), we were interested in studying how risk factors were associated with the presence or absence of disease.

Standard regression is the statistical process of estimating the relationships between a dependent variable and one or more independent variables. Common types of regression are not able to handle censored data and furthermore rely on some basic statistical assumptions such as homoscedasticity etc.

As was mentioned before Survival analysis is a type of regression problem but with the difference that in survival analysis, we are interested **in how a risk factor or treatment affects the time to disease or some other event considering censored data.**

2.2 Introduction to Survival Analysis

As it was mentioned before during the study of a survival analysis problem it is possible some events of interest are not observed for some subjects. This concept is widely known as censoring (Klein and Moeschberger 2005).

2.3 Censoring

When we have some information about a subject's event time, but we don't know the exact event time, this is called **censoring**. There are two types of censoring **Right censoring** and **Left censoring** (Lee and Wang 2003).

Therefore, the time to the event of interest is known only for those instances who have the event occurred. The reasons why censoring might occur are:

- A person of the population doesn't experience the event before the study ends
- A person of the study is lost to follow-up during the study period

- A person of the whole study population withdraws from the study

2.3.1 Right Censoring

Right censoring occurs when a person leaves the study before the event of interest occurs, or the study ends before the event has occurred. For example, we consider customers in a company to study Customer churn/attrition, a.k.a the percentage of customers that stop using a company's products or services and the study ends after 5 years. Those patients who have had no churn by the end of the study are censored.

2.3.2 Types of Right Censoring

- **Fixed type I censoring** occurs when a study is designed to end after a predefined number of years of follow-up. In this case, everyone who doesn't have experienced the event of interest during the course of the study is censored.
- In **random type I censoring**, the study is designed to end after a predefined number of years, but censored subjects do not all have the same censoring time. This is the main type of right-censoring that exists in most of survival analysis studies.
- In **type II** censoring, a study ends when there's a prespecified number of events.

2.3.3 Left Censoring

Left censoring is when the event of interest has already occurred before enrolment. This is very rarely encountered.

Regardless of the type of censoring, we must assume that it is non-informative about the event; that is, the censoring is caused by something aside from the approaching failure.

The below example it is given for better understanding of censoring.

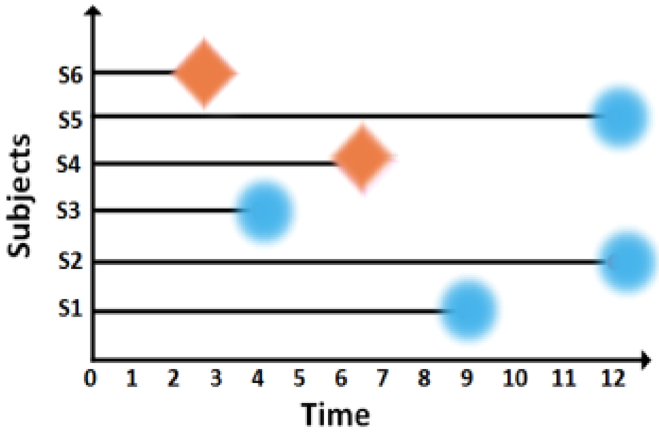


Figure 2.1

Figure 2.1 Six instances are observed in this study over a 12-month period, and the event occurrence information during this time period is recorded. From Figure 1, we can see that only subjects S4 and S6 actually experienced the event (marked by orange rhombus) during the follow up time, and the observed time for them will be the event time. As the event did not occur within the 12-month monitoring period for subjects S1, S2, S3, and S5, these are considered to be censored and are thus marked as blue circles in the figure. More specifically, subjects S2 and S5 are censored since no event occurred during the study period, while subjects S1 and S3 are censored due to withdrawal, or the follow-up being lost within the study time period.

2.4 Truncation

Truncation (Lee and Wang 2003) is another factor which affects the survival data by giving rise to incomplete observations. Truncation is the interval over which the subject was not observed but is not failed as well. The difficulty when having truncated data is that if a person of the study has failed, he or she has never been observed. In truncated survival time data, survival times are excluded systematically from one's sample. There are three types of truncated data:

2.4.1 Left Truncation

The period of ignorance in left truncation starts before the beginning of the study (starting point denoted by t=0) to some future time point t=0. The subject is not observed for some time after the start time but come under observation. Later if they have not had the event. Therefore, left truncation arises as we confront a subject who enrolled sometimes after the onset of risk. This subject is only added to the study if he or she has not failed earlier before the threshold. For example, only those individuals who survive the initial stage of myocardial infarction and reach the hospital will be included in the study. If an individual has been admitted to the hospital and is added to the study where the time t=0 is the time of infarction. For the different patient it may happen at different times, but those patients will never be entered into the study if they die before reaching the hospital. Delayed entry is sometimes used for left truncated data.

2.4.2 Interval Truncation

Interval truncation is just an adoption of left truncation where an individual enters in the study at time zero but disappear for some time and report back to the study generating a gap in between observation. This is what the issue is that an individual could have died when he or she disappears and can never report back.

2.4.3 Right Truncation

In this case, only those individuals are added to the study who have experienced the exit event by some specific date but there is a point after which the subject who hasn't experienced an exit event is not observed anymore and consequently, long survival times are excluded systematically.

2.5 Terms and notation

Table 2.2 This table is about terms and notations used for survival analysis problem formulation.

Notations	**Descriptions**
P	The number of features
N	The number of instances
X	R^{NxP}feature vector
X_i	R^{1xP}covariate vector of instance *i*
T	R^{Nx1}vector of event times
C	R^{Nx1}vector of last follow up times
y	R^{Nx1}vector of observed time which is equal to *min (T,C)*
δ	*Nx1* binary vector for event status
β	R^{Px1}coefficient vector
f(t)	Death density function
F(t)	Cumulative event probability function
S(t)	Survival probability function
h(t)	Hazard function
h_0*(t)*	Baseline hazard function
H(t)	Cumulative hazard function

2.6 Problem Formulation

For a given observation i in our dataset represented by a triplet (X_i,y_i,δ_i), where $\underline{X_i \in R^{1xP}}$is the feature vector, $\delta_i$is the binary event indicator and $y_i$is the observed time and is equal to the survival time T_i if the given observation is uncensored otherwise C_i, if the given observation is censored. The purpose of survival analysis is to estimate the time to the event of interest for a new instance k with feature predictors denoted by a new feature vector X_k.

2.7 Survival Function

The **survival function** (Lee and Wang 2003; Klein and Moeschberger 2005) represents the probability that the time to the event of interest is not earlier than a specified time *t*.

Often survival function is referred as: the *survivor function* or *survivorship function* in problems of biological survival, and as *reliability function* in mechanical survival problems. Reliability function is denoted *R*(*t*). Survival function is represented as follows:

$$S(t) = P(T \geq t) = P\ (\textit{an individual survives longer than t}) \qquad (1)$$

Survival function decreases when the t increases. Its starting value is 1 for t=0 which represents that in the beginning of the observation all subjects survive. From the definition of c*umulative death distribution function F(t),*

$$S(t) = 1 - P\ (\textit{an individual fails before t}) = 1 - F(t) \qquad (2)$$

Cumulative death function represents the probability that the event of interest occurs earlier than time t.

The survival function is therefore associated with a continuous probability density function by

$$S(t) = P(T>t) = \int_t \ P(t')\,dt', \qquad (3)$$

Similarly, the survival function is related to a discrete probability P(t) by

$$S(t) = P(T>t) = \sum_{T>t} P(t) \qquad (4)$$

2.8 Probability density function

Survival time *T* has a probability density function defined as the limit of the probability that an individual fails in the short interval t to *t+Δt* per unit width *Δt.* This can be expressed as:

$$f(t) = \frac{\lim_{\Delta t \to 0} P[an\ individual\ dying\ in\ the\ interval\ (t, t+\Delta t)]}{\Delta t} \qquad (5)$$

In real world examples if there are no censored observations the probability density function f(t) is estimated as the proportion of subjects having the experience of the event of interest (e.g. event of death in clinical studies) in an interval per unit width:

$$\hat{f}(t) = \frac{Number\ of\ patients\ dying\ in\ the\ interval\ beginning\ at\ time\ t}{(Total\ number\ of\ patients) x (interval\ width)} \qquad (6)$$

Here it is important to note that similar to S(t) when censored observations are present is not applicable.

2.9 Hazard Function

In survival analysis, another commonly used function is the hazard function *h(t)*, which is also called the *force of mortality,* the *instantaneous death rate*, or the *conditional failure rate (*Dunn and Clark 2009).

The hazard function t (Lee and Wang 2003; Klein and Moeschberger 2005) does not indicate the prospect or probability of the event of interest, but it is the rate of event at time *t* as long as no event occurred before time *t*. Specifically is the ratio of the probability density function to the survival function.

Hazard function is defined as:

$$h(t) = \lim_{\Delta t \to 0} \frac{Pr(t \leq T < T + \Delta t | T \geq t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{F(T + \Delta_t) - F(t)}{\Delta t \cdot S(t)} = \frac{f(t)}{S(t)}$$

$$\frac{\lim_{\Delta t \to 0} P[\text{ an individual fails in the time interval } (t, t + \Delta t) \text{ given the individual has survived to } t]}{\Delta t} \qquad (7)$$

It is defined as of failure during a very small-time interval assuming that the individual has survived to the beginning of the interval.

The hazard function can also be defined in terms of the cumulative distribution function *F(t)* and the probability density function *f(t)* as:

$$h(t)\text{:} \ \frac{f(t)}{1 - F(T)} = \frac{f(t)}{S(t)} \qquad (8)$$

In real world cases when there are no censored instances the hazard function is estimated as the proportion of patients dying in an interval per unit time, given that they have survived to the beginning of the interval:

$$\hat{f}(t) = \frac{\text{number of patients dying in the interval beginning at time } t}{(\text{number of patients surviving at } t) \ x \ (\text{interval width})} =$$

$$\frac{\text{number of patients dying per unit time in the interval}}{\text{number of patients surviving at } t} \qquad (9)$$

The cumulative hazard function *H(t)* is the sum of the individual hazard rates from time zero to time *T*. The formula of cumulative hazard function is:

$$H(t) = \int_0^T h(u) du \qquad (10)$$

For any two instances the X_1 and $X_2$the hazard ratio is given by

$$\frac{h(t,X_1)}{h(t,X_2)}=\frac{h_0(t)exp(X_1\beta)}{h_0(t)exp(X_2,\beta)}= exp[(X_1 - X_2)\beta] \tag{11}$$

which means that the hazard ratio is independent of the baseline hazard function.

2.10 Traditional Statistical Methods for Survival Analysis

There are three different types of statistical methods to estimate survival and hazard function. They are divided into three main subcategories: parametric, semi-parametric and non-parametric. Each category includes some methods. In the array below we give a summary of different categories of statistical methods.

Table 2.3 This table is categorization of models into parametric, semi-parametric and non-parametric categories.

Category	**Methods**
Non-parametric	Kaplan-Meier Nelson-Aalen Life-Table

Semi-Parametric	Cox model Regularized Cox Cox Boost Time-Dependent Cox
Parametric	Tobit Buckley -James Penalized regression Accelerated Failure Time

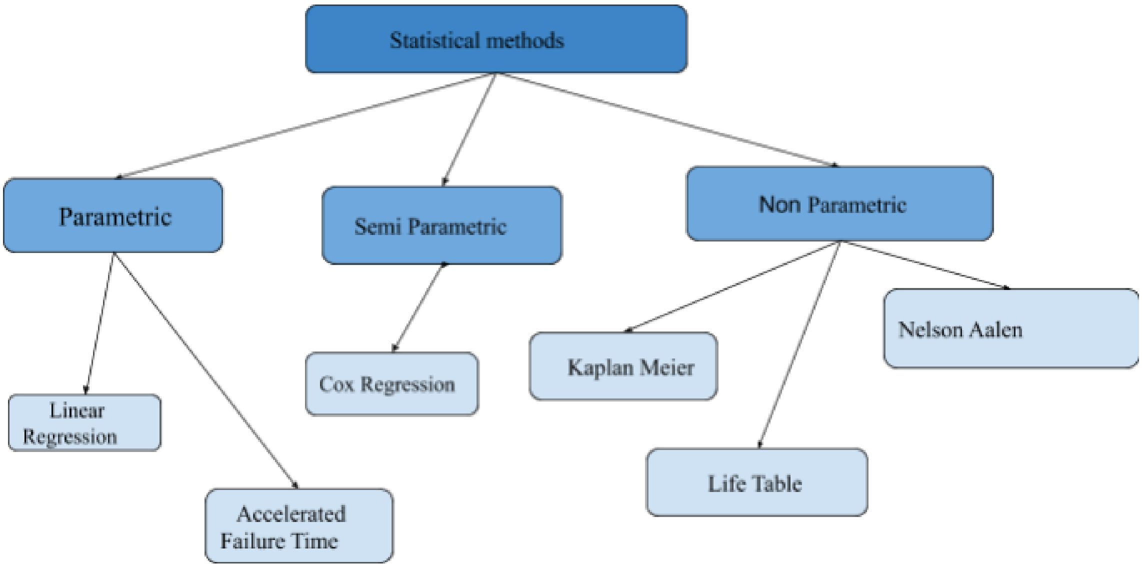


Figure 2.2 A tree structured representation of statistical methods that have been used in Survival Analysis.

2.11 Survival analysis using machine learning methods

There have been a lot of scientific studies that use machine learning methods in order to apply survival analysis in a set of data. Statistical methods and machine learning approaches have the same purpose (share the common goal) to make predictions of the time the event of interest will occur (Wang 2019).

The difference is that statistical methods focus more on both the distribution of the event times and statistical properties of the parameter estimation in contrast to machine learning methods that are usually used for high - dimensional problems. The main goal of applying machine learning methods for survival analysis is that efficient machine learning and deep learning algorithms can learn the dependencies between covariates and survival times in different ways.

Machine learning is effective when there are a large number of instances in a reasonable dimensional feature space and there have been used algorithms that have the ability to discover non linear dependencies between the covariates (Wang 2019).

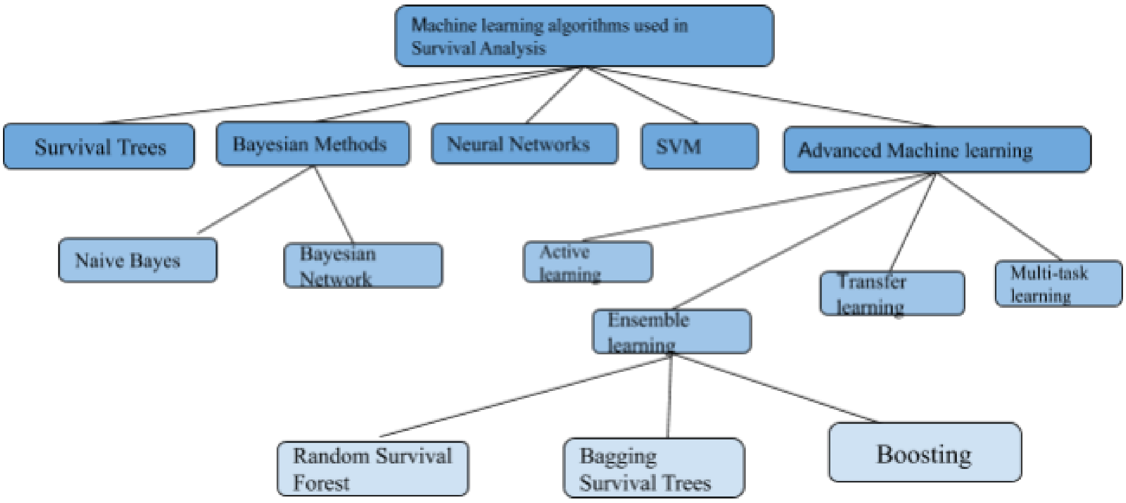


Figure 2.3 A tree structured representation of machine learning algorithms that have been used in Survival Analysis.

2.12 Performance Evaluation Metrics

Due to the uniqueness of the survival data, instead of standard evaluation metrics used at different machine learning tasks, such as root mean square error (R^2), more specialized evaluation metrics are utilized.

2.12.1 Concordance Index

The *concordance index* or *c-index* is a metric to investigate if the predictions made by an algorithm are accurate. It is defined as the proportion of concordant pairs divided by the entire number of possible evaluation pairs. The survival times of two observations can be ordered either if both are uncensored if the event time of an uncensored observation is smaller than the censoring time of the censored observation.

So, concordance index cares about the order that the event of interest will happen, between two (or more) observations. For two given observations $X_1$and X_2 and their predicted values $\widehat{X_1}$, $\widehat{X_2}$ the concordance probability between them can be computed as

$$c = P(\widehat{X_1} > \widehat{X_2} | X_1 \geq X_2) \qquad (12)$$

Because of the different output predictions between survival algorithms/models there are multiple ways of calculating the c-index.

2.12.2 Brier Score

Brier score can only be used for models whose predictions are probabilities e.g., **probability of survival in a given time interval.** Consequently, Brier's Score outcome must remain within the range [0,1]. When the outcome of a prediction is binary with a sample of N observations and of each X_i the prediction at time t is $\widehat{y}_i$(t) and the actual value is $y_i(t)$ Brier's score formula is:

$$BS(t) = \frac{1}{N} \sum_{I=1}^{n} [\widehat{y}_i(t) - y_i(t)]^2 \qquad (13)$$

In 1999 Graf extended this measure for survival data, so Brier Score could also care about censored information. Below is the formula of Brier Score for survival data:

$$BS(t) = \frac{1}{N} w_i(t) \sum_{I=1}^{n} [\widehat{y}_i(t) - y_i(t)]^2 \qquad (14)$$

2.12.3 Mean Absolute Error

Mean absolute error is a very common measure for the evaluation of many tasks in machine learning. In terms of survival analysis is defined as the average of the differences between the predictions and the actual values:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}(\delta_i|y_i - \widehat{y_i}|) \qquad (15)$$

where $y_i$are the actual values of time and $\widehat{y}_i$ are the predictions.

It should be noted that only the samples for which the event occurs are being considered in this metric since if δ i = 0, the corresponding term will become zero. MAE can only be used for the evaluation of survival models which can provide the event time as the predicted target value such as AFT models.

2.12.4 Log - Rank Test

The Log rank is the most commonly used non-parametric statistical test for comparing the survival distributions of two or more groups (such as different treatment groups in a clinical trial). It is often used in clinical trials to compare survival experience for two groups of individuals. Other tests for differences, like the two-sample t-test are not appropriate for this type of data, because the data is usually highly skewed (Stephanie 2018).

Log-Rank Test hypothesis are:

H_0 : There is no difference between survival probabilities between two groups

H_1 : There is difference between survival probabilities between two groups

The probability is calculated for some event, which could be death or another significant event. The test compares estimates of the hazard functions of the two groups at each observed event time. The observed and expected number of events is calculated in one of the groups at each observed event time, then these results are added to get an overall summary for all points in time when an event happened.

Assumptions

The assumptions for the log-rank test are:

- Censoring (which happens when you don't know the exact survival time) must be unrelated to prognosis,
- Survival probabilities are equal for subjects recruited at any time in the study,
- The events happened at the specified time.

Chapter 3

3.1 Survival analysis in high dimensional data

Recent years biomedical tools and technologies has changed the way of biomedical research (Benam et al. 2019). Biologists and analysts in the field of computational biology are now able to use a wide variety of biomedical and genetic data from different sources in order to conduct more experiments. Increasing availability of this kind of data drives biomedical research at better understanding of the biological mechanisms of particular diseases such as cancer.

In the context of survival estimation and usage of genomics data produced, existing statistical techniques are not the best way to analyze and produce useful insights. The number of features in biological data such as gene expression data and pathological images are significantly large. Often the number of features in this kind of data exceeds the number of observations (patients) that are available.

Many studies have resulted in the successful identification of previously unknown subtypes of cancer as well as stratifying newly diagnosed patients into subtypes based on short- or long-term prognoses and predicting survival time (Chen et al. 2019; Koestler et al. 2010)

There have been many research studies that have apply machine learning and statistical techniques to handle biomedical data and stratify the risk of death in a patients' cohort. Furthermore, many studies attempt to identify subtypes of a certain cancer in patients suffering from the same disease. Various studies have merged different data modalities i.e., clinical data, gene expression data, pathological images, DNA methylation, miRNA etc. in order to have a better representation of patients.

3.2 Unsupervised Approach

Unsupervised techniques in machine learning aim to find previously unknown patterns in data set without pre-existing labels. They are also known as self-organization methods and allows modelling of probability densities of given inputs. In terms of survival analysis from high dimensional data unsupervised learning attempts to discover gene expression profile structure that led to potential different cancer subtypes.

Since survival information is not taken into account during the identification of subgroup membership (i.e., subgroups are identified using only the gene expression data or only the clinical data and are not associated with survival outcome). Usually after the label extraction, data are splitting into training and test set and after that a classifier can be applied into training set using as ground truth the labels that have been extracted in previous step. The classifier is then validating in test set in order to test its performance.

To name a few, unsupervised approaches include hierarchical clustering (Murtagh et al. 2011), K-means (Macqueen et al. 1964) and model-based clustering (Tjaden et al. 2006).

Although there have been some research studies that used unsupervised learning to discover cancer subtypes focusing on association of survival outcome with cancer subtypes.

Many of these studies have been follow a methodology that use a metric of similarity between the individual observation to group observations into different clusters (Young et al. 2017). The similarity metric used in such types of methodology is based on features that are selected independently from survival outcome. Though this approach results to objects in the same cluster tend to be more similar and objects in different cluster tend to be more dissimilar, there is no guarantee that subgroups identified have some biological meaning. This is because survival information has been ignored in features selection step.

The identification of the subtypes has to be associated with survival outcome in order to have a biological meaning.

3.3 Supervised Approach

Supervised techniques in machine learning infers a function from labeled training data consisting of a set of training examples. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples (Vijver et al. 2002; Veer et al 2002; Gao et al. 2019;). In terms of survival analysis, and more specifically in identification of cancer subtypes in a patients' cohort, there are two options. The first one is when analyzed data are labeled i.e., in breast cancer dataset that has labels in each individual like "high risk", "low risk", "medium risk" etc. and the classifier in then trained to these labeled examples.

The second option is not to have labels in advance and in the context in survival time observations can be partitioned into a "low risk" or a "high risk" subgroup based on their median survival time. The subgroup identification in this case is set by taking account a relevant clinical threshold. Determined subgroups are then used to train a classifier and then predict subgroup membership in of a future patient.

However, in the second approach there is also no guarantee that labels have been correctly assigned to their actual subgroups so the trained classifier. For example, there are not relevant clinical thresholds for all types of diseases in order to take into account when there is a need to stratify risk in a patients' cohort. This situation leads classifiers to bad generalization and when a future patient with cancer needs to be categorized there is a chance of wrong categorization.

For example, if there are two pre-specified subtypes associated with the same cancer diagnosis, and patients with subtype 1 live somewhat longer than those with subtype 2, there is a chance of significant overlap between the two subgroups with respects to survival time. By simply assigning observations to the low- risk or high-risk group based on the median survival time would result in an incorrect determination of the subgroups for unseen patients. Therefore, the diagnosis of any future patients based on this model would be questionable.

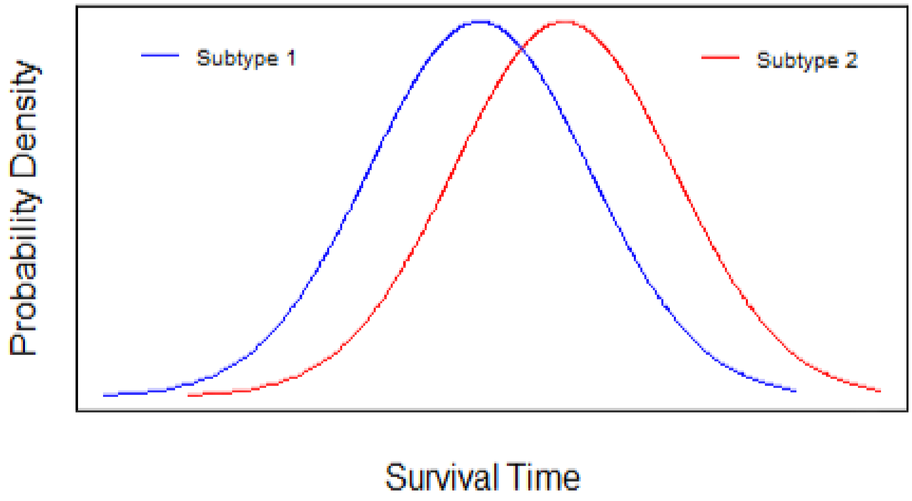


Figure 3.1 Cancer subtypes probability density with significant overlap

3.4 Semi-Supervised Approach

Semi-supervised approaches in machine learning falls between supervised and unsupervised learning and use both unsupervised and supervised techniques. Since subgroups of a certain type of cancer is not predefined in the vast majority of datasets, semi supervised techniques facilitating the identification cancer subtypes (Koestler et al. 2010; Wasito et al. 2012; Wei et al. 2018). Additionally semi-supervised techniques take the advantage by taking into consideration the survival outcome and in this way identify biologically meaningful classes.

Below we describe the four universal steps in semi-supervised learning:

1. **Data splitting**: Since different modalities data combined together from different sources and the datasets usually have not been splitted into training and test set there is a need to efficiently preprocess the raw data and divide them into training and test set. Training and test splitting is a very important stage of data preprocessing in order to build a dataset that is not imbalanced and includes several training examples as well as testing cases. Random splitting the data may lead to a bad balance between the

proportion of samples allocated to training and validation set. In order to avoid bad predicting performance and overfitting it is strongly suggested carefully splitting training and validation set.

2. **Subgroup assignment based on supervision**: In this phase a list of significant features that are associated with the phenotype of interest i.e., survival outcome is identified. Then based on these subsets of features an unsupervised technique is applied for identification of subgroups of a certain disease (e.g., cancer).

3. **Determining class subgroup in the testing set:** After the identification of latent subgroups (labels) a classifier is applied to training set. Then test set is used for mapping "unseen" examples into identified subgroups.
   There have been two canonical semi-supervised techniques in literature, the Clustering Cox method and the Risk index approach.

4. **Testing the association with the phenotype of interest:** Provided that previous steps were successful trained classifier should correctly assign new examples to the actual subgroups. So, when testing the trained classifier and get predictions, patients that have been predicted to have cancer subtype 1 should be significantly different in terms of survival probability. In order to test the association of identified subgroups with the survival outcome the log-rank test can be used.

3.5 Related Work

As it was mentioned before the identification of subtypes of a certain cancer is crucial for various aspects on biomedical industry. Several works had been done to integrate multiple types of genomics data to investigate cancer subtypes. Guo (Guo et. al 2019) proposed a hierarchical deep learning framework to named *HI-SAE* to integrate gene expression and transcriptomic alternative splicing profiles data as well. They stacked autoencoder neural network to learn high level representation in each data type respectively. After that the data representation goes through another learning layer for more complex representation learning. In their final step used unsupervised techniques to group patients into similar subgroups that indicates cancer subtypes.

Chen in his recent research (Chen et al. 2019) proposed a deep learning framework namely *Deep Type t*hat performs joint supervised classification, unsupervised clustering and dimensionality reduction to learn cancer-relevant data representation with cluster structure.

Shen (Shen et. al 2009) introduced a model for integrative clustering namely *iCluster*. Their framework models the associations between different data types and the variance-covariance structure within data types while in parallel reducing the dimensionality of the datasets.

In supervised learning approaches Gao (Gao et. al 2019) proposed a novel supervised cancer classification framework namely *DeepCC*. It has the ability to capture biological characteristics associated with distinct molecular subtypes and in this way to learn deep features. These deep features are enabling more compact within-subtype distribution and between-subtype separation of patient samples.

Vasudevan (Vasudevan et. al 2018) used max-flow/min-cut graph clustering in order to identify molecular mechanisms of cancer and discover novel biomedical targets.

He in his research (He et. al 2019) integrate gene expression and clinical data in order to accurately discover breast cancer subtypes. They generally utilized two phases, the first one is gene selection and the second one clustering. Specifically, they utilized maximum relevance minimum redundancy for gene selection and k-means for unsupervised learning (clustering).

Gene expression data are high dimensional so in order to use them we have to preprocess them. Final purpose in this stage of analysis often is to improve model predictions. There have been many studies that utilize several known machine learning techniques for accurate feature selection or dimensionality reduction in biological studies using gene expression data. In one of these studies Souza (Souza et. al 2019) prosecuted a detailed comparison of two reduction methods, attribute selection and principal component analysis. He introduced a combination of consistency-based subset evaluation and minimum redundancy maximum relevance technique to improve his final model predictions.

Since gene expression data includes large quantities of variables with unknown correlation structures Wang (Wang et. al 2011) proposed a dimension reduction procedure based on the variable importance measurement (VIM) in the framework of targeted maximum likelihood estimation.

Furthermore, unlike other studies that focus on feature selection from high dimensional data and especially from gene expression data Lee (Lee et. al 2017) produced research that explore different analysis techniques for microarray data in order to create a more effective predictor of age from DNA methylation level. In this study several known models such as principal component regression and supervised principal component regression are compared to elastic net regression, and it is found that elastic net regression performs better than the other model when considering less than ten principal components for each method

Paul in his study (Paul et. al 2014) highlights the need of dimension reduction of gene expression data for developing a robust classifier to predict patients with cancerous genes. In this study it is constructed a fuzzy rule-based classifier the gene expression matrix is discretized. The importance factor of each gene is then evaluated representing the degree of presence of a unique linguistic value of the gene both in disease and non-disease classes. In other words, gene selection algorithm evaluates fuzzy importance factor of each gene that signifies relevance of the gene in classifying the diseased patients using microarray gene expression data. Finally, only a subset of important genes are selected.

Chapter 4

This chapter describes the procedure that have been followed and the techniques that have been used.

4.1 Selecting clinical parameters

Our purpose in this phase is to select clinical variables that have a strong association with the phenotype of interest which is patient survival time. For that reason, Cox proportional Hazard model was used.

The Cox proportional-hazards model is essentially a regression model commonly used statistical in medical research for investigating the association between the survival time of patients and one or more predictor variables.

It is the most widely used statistic model for survival analysis (David 1972). The reason that this model is very much known is that the knowledge of the underlying distribution of time to event of interest is not required although the attribute 's influence on the outcome assumed to be exponential. In other words, it does assume that the effects of the predictor variables upon survival are constant over time and are additive in one scale. In other words, it does assume that the effects of the predictor variables upon survival are constant over time and are additive in one scale.

Basic Cox Model: If an observation of our data is denoted as *i* and it is represented by a triplet (X, y_i, δ_i), the hazard function $h(t,X_\iota)$ in the Cox model follows the proportional hazards assumption given by

$$h(t,X_i)=h_0(t)exp(X_i\beta), \text{ for } i = 1,2,.....,N \qquad (16)$$

where the baseline hazard function, $h_0(t)$can be an arbitrary non negative function of time

X_i=$(x_{i1}, x_{i2}, \dots, x_{iP})$ is the corresponding covariate vector for instance i, and β^T=$(\beta_1,\beta_2, \dots,\beta_p)$ is the coefficient vector. Cox model belongs to semi parametric techniques because the baseline hazard function, $h_0(t)$, is unspecified and for that reason it is not possible to fit the model using the standard likelihood function. In other words, the hazard function h_0(t) is a nuisance function while the coefficients β are the parameters of interest in the model. Let us assume we have two instances, X_1 and X_2. The hazard ratio is given by:

$$\frac{h(t,X_1)}{h(t,X_2)} = \frac{h_0(t)exp(X_1\beta)}{h_0(t)exp(X_2,\beta)} = exp[(X_1 - X_2)\beta] \qquad (17)$$

The survival function then can be computed as follows:

$$S(t) = exp(-H_0(t)exp(X\beta)) = S_0(t)^{exp(X\beta)} \qquad (18)$$

Where $H_0(t)$is the cumulative baseline hazard function, and $S_0(t)$= $exp(-H_0(t))$ represents the baseline hazard function.

The Breslow’s estimator (Breslow 1972) is the most widely used method to estimate $H_0(t)$, which is given by:

$$\widehat{H_0}(t) = \sum_{t_i \le t} \widehat{h_0}(t_i) \qquad (19)$$

where $\widehat{h_0}(t_i)$=$\frac{1}{\sum_{j \in R_i}} e^{X_j\beta}$ if t_i is an event time, otherwise $\widehat{h_0}(t_i)$= 0.

Furthermore, a variety of different penalty functions have been introduced in literature in order to apply Cox regression model but in parallel identify most relevant features in high dimensional datasets. Such penalty functions are lasso (Tibshirani 1996), group lasso, fused lasso (Tibshirani et al. 2005), and graph lasso.

In this thesis we implemented standard Cox proportional hazard regression and obtain what is the ranking Cox model gives to all clinical covariates. In order to complete the identification of clinical variables that are significant both for survival and for death we also apply Penalized logistic Regression.

When having too many variables or there is a need to select a subset of the features Penalized logistic regression can be used in terms of classification. Penalized Logistic Regression imposes a penalty to the logistic model and in shrinks the coefficients of the less contributive variables toward zero. This is also known as regularization (Francis 2018).

The most commonly used penalized regression include:

- **ridge regression**: variables with minor contribution have their coefficients close to zero. However, all the variables are incorporated in the model. This is useful when all variables need to be incorporated in the model according to domain knowledge.
- **lasso regression**: the coefficients of some less contributive variables are forced to be exactly zero. Only the most significant variables are kept in the final model.
- **elastic net regression**: the combination of ridge and lasso regression. It shrinks some coefficients toward zero (like ridge regression) and set some coefficients to exactly zero (like lasso regression).

In this thesis we used lasso penalized logistic regression.

4.2 Latent class membership identification

4.2.1 Search optimal number of groups

In unsupervised learning and especially in subgroups discovery, except from decide which unsupervised algorithm to choose it is a need to predefine the number of expected groups. Consequently, there have been a number of appropriate metrics and methods such as silhouette, elbow rule and gap method to evaluate clustering. Different number of groups may have different evaluation. As it concerns which metric to use in order to better identify the optimal number of clusters, this is something not well established in literature so there is not a clear answer. We utilized two of the most widely used methods for discover the optimal number of clusters, elbow and silhouette method.

4.2.1.1 Elbow method

In Elbow method the idea is to run k-means clustering on the dataset for a range of values of k, for example from 1 to 10, and for each of value of k calculate the sum of squared errors

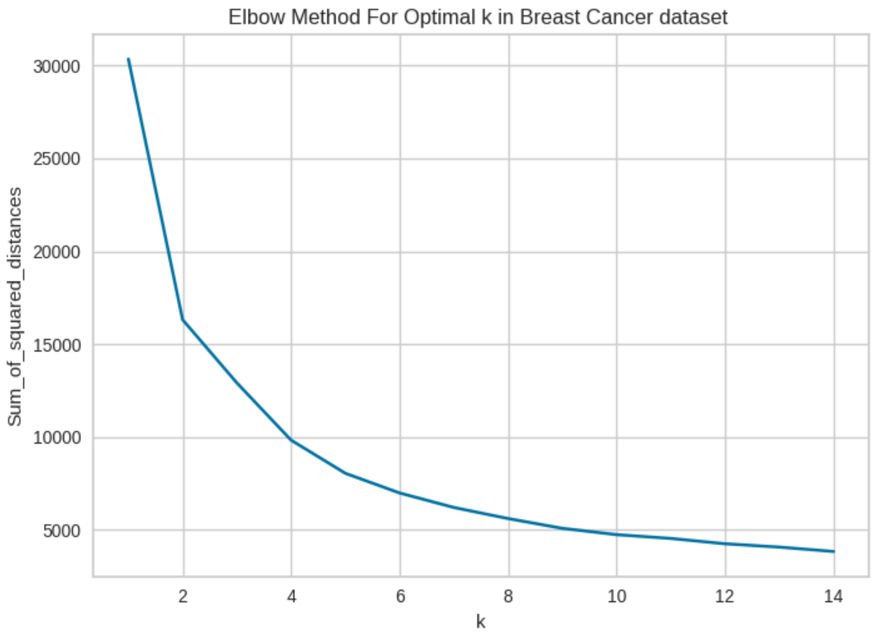


Figure 4.1 Elbow rule example in which we can decide that the optimal number of clusters are 2.

(SSE). Elbow rule typically uses the percentage of unexplained variance. This number is 100% when the number of clusters are 0, and it decreases (initially sharply, then more modestly) as the number of clusters increasing. When each point constitutes a cluster, this number drops to 0. Somewhere in between, the curve that displays your criterion, exhibits an elbow, and that elbow determines the number of clusters.

4.2.1.2 Silhouette method

Silhouette is another method to study the separation distance between the identified groups. Like elbow rule belongs to visual techniques for optimal number of groups determination. Silhouette plot displays a measure of how close each point in one group is to points in the neighboring groups and thus provides a way to assess the number of groups visually. Measure displayed has a range of [-1,1]. A value near to 0 denotes that sample is very close to the decision boundary between two neighboring groups, so it is possible this sample to be assigned in a wrong group. In the other hand, coefficient near to 1 indicate that the sample is far away from the neighboring so most probably that sample may be assigned in the wright cluster.

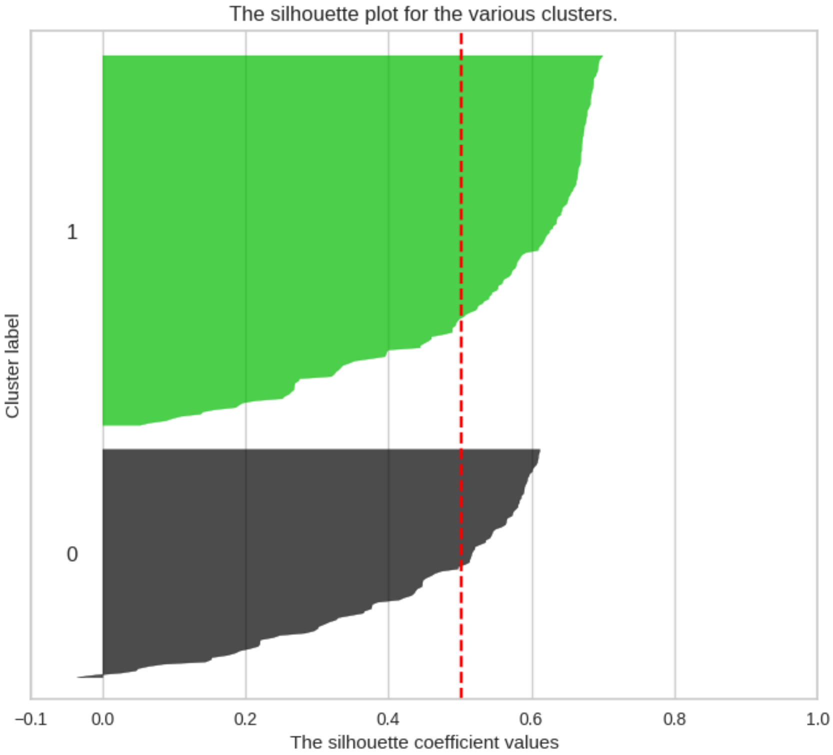


Figure 4.2 Silhouette method example for optimal number of clusters identification

Table 4.1 This table gives an explanation about the interpretation of silhouette-coefficient.

Silhouette-Coefficient	Interpretation
0.70 - 1.00	Identification of a very clear structure
0.50 - 0.69	Identification of a reasonable structure between

0.26 - 0.49	Identification of weak structure and further investigation is needed
<= 0.25	No substantial structure has been found

4.2.1.3 Unsupervised learning

In unsupervised clustering K-means is one of the simplest and popular unsupervised machine learning algorithms. In order to identify clusters K-means algorithm starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids

It stops creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.

4.3 Feature Selection for high dimensional Survival data

Feature selection is the process of selecting a subset of relevant features for use in model construction. Necessity of feature selection as a preprocessing technique is proved by a number of reasons. Some of them may be that researchers want to simplify their models and want more interpretable models to utilize, they maybe want to achieve better training times when cases that training time matters, they often want to reduce overfitting etc. The main reason for applying a feature selection algorithm as a preprocessing step before developing a machine learning model is that usually high dimensional data include redundant or irrelevant features and thus they can be moved without incurring much loss of information.

Even though relevancy and redundancy are two distinct notions, a relevant feature may be redundant if within the same dataset another feature that have strong correlation with, exists.

A naive approach of feature selection might be to test every possible feature subset in order to discover this one that minimizes the error rate. This is an exhaustive search of the feature space and is computationally intractable. Consequently, there have been developed many efficient feature selection techniques. According to the choice of the evaluation metric each one use, feature selection techniques can be discriminated in three main categories:

- Wrapper methods: They rely on greedy search algorithms as they evaluate all possible combinations of the features and keep the combination that produces the best performance on the selected machine learning algorithm. As is obvious those kinds of approaches might be very computationally expensive due to high dimensions of the dataset.

- Filter methods: Filter methods use a proxy measure to score a feature subset. Common measures include mutual information, pointwise mutual information and relief-based algorithms. Filter methods do not produce a feature subset according to a specified classifier. This means that these methods usually produce a feature subset that is more general that a feature subset discovered by a wrapper method. Some feature methods provide feature ranking instead of explicit best feature subset and the cutoff point in the ranking is chosen via cross-validation.

- Embedded methods: Embedded methods are similar to wrapper methods since they are also used to optimize the objective function of a specific learning algorithm. Although they are much less computationally expensive than wrapper methods. Most widely known example of embedded method is Lasso in regression model that uses L1 function to penalize coefficients shrinking many of them to zero. Features that have non-zero regression coefficients are selected by this procedure.

4.3.1 Feature selection and dimensionality reduction algorithms used in this thesis

4.3.1.1 Relief algorithm

Relief algorithm was first introduced in 1992 by Kira and Rendell (Kira et. al 1992). This algorithm belongs to filter-based approaches in the field of feature selection. In his initially implementation was designed for binary problems and could handle both numerical and discrete features.

Next was introduced ReliefF that was an extension of basic Relief algorithm for multiclass problems and additionally RReliefF that was adapted for continuous class problems (regression). The basic idea behind Relief algorithm remains simple and that is to estimate the quality of attributes on the basis of how well the attribute can distinguish between instances that near to each other.

Basic Relief algorithm is described below:
Input is a vector of attribute values and the label for each training instance and output is the vector of the weights of attributes that denote their quality.

Steps for basic Relief algorithm implementation

```
1: set all weights equal to 0
2: for each attribute from 1 to number of attributes ($n$):
        a. randomly select an instance ($i$)
        b. find nearest instance that has the same label and also find the nearest instance
           that has the opposite label
3: for all attributes (a):
        c. calculate the weight
4: End
```

Weights are calculated by the following formula:

diff (a, i_1,i_2) = if {value(a,i_1) = value(a,i_2): 1}
otherwise: 1, where instances are denoted with i. This formula consists for nominal attributes.

$$\text{diff (a,}i_1,i_2\text{)} = \frac{|value(a,i_1) - value(a,i_2|}{max(a) - min(a)}$$

Weights on attributes are updated based on idea:

- if instance i and instance their nearest attribute's instance (this attribute with the same label that was mentioned before) have large difference on value that means that this attribute separates two instances with the same class which is not desirable. Otherwise, if instance i and attribute's instance (this attribute with the opposite label) has a large difference on value that means the attribute separates the two instances with different class which is desirable.

4.3.1.2 MultiSurf algorithm

In his study Urbanowicz (Urbanowicz et. al 2018) made a benchmarking for all relief-based feature selection methods. In this study all relief-based algorithms were implemented and were evaluated in a large number of datasets. Also, it was introduced a new algorithm namely MultiSurf that inherits the majority of Multisurf* algorithm (Granizo-Mackenzie et. al 2013). MultiSurf adopts all aspects of MultiSurf* but eliminates the "far"scoring introduced in Surf* (Greene et al. 2010).

More specifically Surf* algorithm introduced the concept of "instances that were near vs instances that were far from target". Applying a distance threshold T, it determines any instance within the threshold was considered as "near" and those outside as "far". Surf* proceeds to weight 'far' instance differences in an opposite manner than 'near' instances. Specifically, feature value differences in hits differently receive a (+1) while feature value differences in misses differently receive a (−1).

The dead-band boundary $Tnear_i$in MultiSurf is equal to T_i- $\sigma_i/2$.

Pseudo-code for MultiSURF algorithm

```
**Require** for each training instance a vector of feature values and the class value
$n \leftarrow$number of training instances
$a \leftarrow$number of attributes (i.e., features)

**#STAGE 1**
preprocess dataset {$\approx a \cdot n$ time complexity}
**#STAGE 2**
pre-compute distance array {$\approx 0.5 \cdot a \cdot n^2$time complexity}
**for** i:=1 **to** $n$ **do**
    set $T_i$ to mean distances between instance $i$ and all others
    set $\sigma_i$ to standard deviation of those distances
**end for**
**#STAGE 3**
initialize all feature weights W[A]:=0.0
**for** $i$:=1 **to** $n$ **do**
    **# IDENTIFY NEIGHBORS**
    initialize hit and miss counters $h$:=0.0 and $m$:=0.0
    **for** j:=1 **to** $n$ **do**
        **if** distance between $i$ and $j$ is < $T_i$- $\sigma_i/2$(using distance array) **then**
            **if** $j$ is a hit **then**
                $h$+= 1 {and identify instance as hit}
            **else if** $j$ is a miss **then**
                $m$+=1 {and identify instance as miss}
            **end if**
        **end if**
    **end for**
    **# FEATURE WEIGHT UPDATE**
    **for all** hits and misses **do**
        **for** A:= **to** a **do**
            W[A]:=W[A] - $diff(A,R_i,H) / (n \cdot h) + diff(A,R_i, M) /(n \cdot m)$
        **end for**
    **end for**
**end for**
**return** the vector W of feature scores that estimate the quality of features
```

For far instanced the update would look like Equation *(3)*

$$W[A]:=W[A]-diff(A,R_i,H)/(n*k)+\sum_{C\neq class(target)}[\frac{P(C)}{1-P(class(target))}diff(A,R_i,M(C))]/(n\cdot k) \quad (1)$$

$$W[A]:=W[A]-diff(A,R_i,H)/(n\cdot k)+\sum_{C\neq class(target)}[\frac{mc}{m}diff(A,R_i,M(C))]/(n\cdot k) \quad (2)$$

$$W[A]:=W[A]-diff(A,R_i,H)/(n\cdot h)+\sum_{C\neq class(target)}[\frac{mc}{m}diff(A,R_i,M(C))]/(n*m) \quad (3)$$

4.3.1.3 Autoencoders

Autoencoders are neural networks that are trained to attempt to copy its inputs to its outputs. An Autoencoder may be viewed as consisting of two parts: an encoders function *h = f(x)* and a decoder that produces a reconstruction *r = g(h)*. Autoencoders are designed to be unable to learn to copy perfectly and if an autoencoder succeeds in simply learning to set *g(f(x)) = x* everywhere then it is not especially useful. Often, they are restricted in ways that allow them to copy only approximately and to copy only input that resembles the training data. In this sense because the autoencoder is forced to prioritize which aspects of the input should be copied, it often learns useful properties of the data. Autoencoders are traditionally used for dimensionality reduction or feature learning (Goodfellow et al. 2017). They may be thought of as being a special case of feedforward networks and may be trained with all of the same techniques, typically minibatch gradient descent following gradients computed by back-propagation.

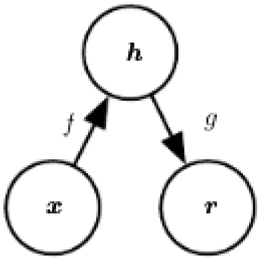


Figure 4.3 The general structure of an autoencoder, mapping an input x to an output (called reconstruction) *r* though an internal representation or code *h.* The autoencoder has two components: the encoder *f* (mapping *x* to *h)* and the decoder *g* (mapping *h* to *r).*

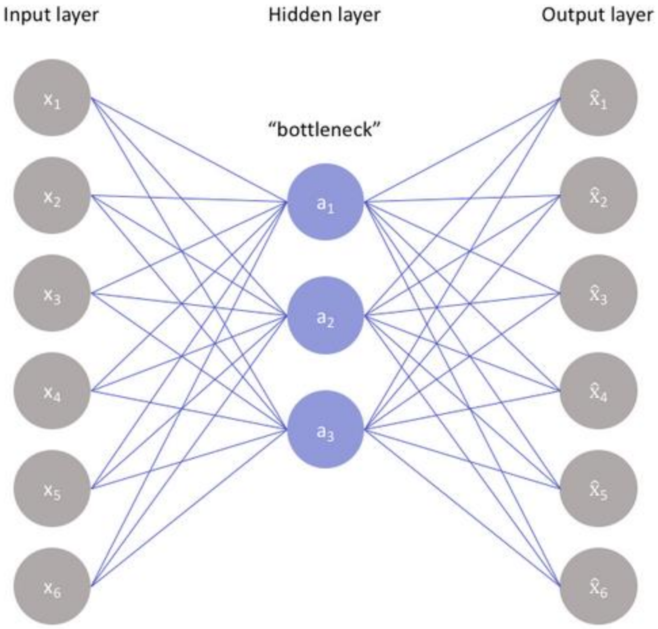


Figure 4.4 This figure shows an autoencoder with one hidden layer.

As visualized above, the task is the inputs x be accurately reconstructed (x). This network can be trained by minimizing the *reconstruction error*, which measures the differences between our original input and the consequent reconstruction.

A bottleneck constrains the amount of information that can traverse the full network, forcing a learned compression of the input data. It is true that if we were to construct a linear network *(i.e., without the use of nonlinear activation functions at each layer)* we would observe a similar dimensionality reduction as observed in PCA.

The ideal autoencoder model balances the following:

- Sensitive to the inputs enough to accurately build a reconstruction.
- Insensitive enough to the inputs that the model doesn't simply memorize or overfit the training data

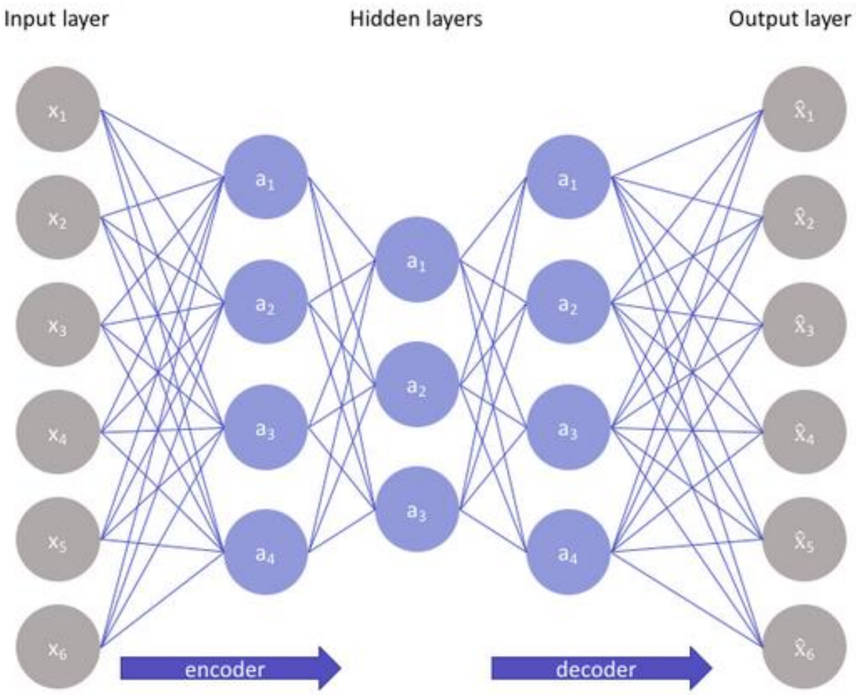


Figure 4.5 This figure shows encoder and decoder parts of an autoencoder architecture with more than one hidden layer.

The simplest architecture for constructing an autoencoder is to constrain the number of nodes present in the hidden layer(s) of the network, limiting the amount of information that can flow through the network. By penalizing the network according to the reconstruction error, our model can learn the most important attributes of the input data and how to best reconstruct the original input from an "encoded" state. Ideally, this encoding will learn and describe latent attributes of the input data.

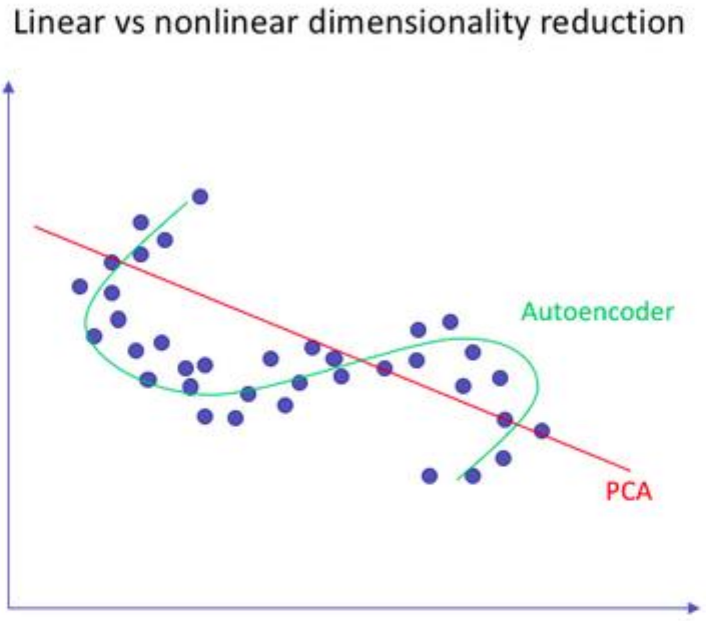


Figure 4.6 This figure shows an optical representation of the difference between a linear and a nonlinear dimensionality reduction.

Because neural networks are capable of learning nonlinear relationships, this can be thought of as a more powerful (nonlinear) generalization of PCA Whereas PCA attempts to discover a lower dimensional hyperplane which describes the original data, autoencoders are capable of learning nonlinear manifold (a manifold is defined in *simple* terms as a continuous, non-intersecting surface). The difference between these two approaches is visualized below.

4.4 Classifiers

4.4.1 Naive Bayes Classifier

Naive Bayes is a probabilistic classifier using Bayes theorem. It calculates the probability a certain instance belonging to each label and chooses the label with the highest probability. Let *n* be the number of classes, C = $\{C_1, C_2,\ldots..C_n\}$ and a given instance X = $\{X_1,X_2,\ldots.,X_n\}$. The posterior probability that this instance belongs to class C_i according to Bayes Theorem can be calculated as:

$$P(C_i | X) = \frac{P(X \mid C_i)\, P(C_i)}{P(X)}$$

Bayes criterion therefore is equal to classifying X in the class the maximizes $P(X | C_i)\, P(C_i)$

4.4.2 Decision Tree Classifier

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

A decision Tree represents classification rules. It is like a flowchart diagram with the terminal nodes representing classification outputs/decisions. The goal is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data).

Each internal node corresponds to an attribute and each leaf node is a class. Algorithm iterations starts with dataset as the root node and then the “best” attribute is chosen at each iteration to split the dataset into subsets in a way that each subset contain data with the same value for an attribute. Homogeneity of one attribute within the subset can be measured with many metrics for example information gain of that attribute. Algorithm’s iterations stops when all objects at a node have an identical class label. Alternatively, algorithm stops when there are no more attributes to be selected in the subset.

Chapter 5

5.1 Experiments on breast cancer dataset

This chapter presents experiments designed to evaluate the performance of proposed method on survival dataset that include both clinical and genes expression information.

5.2 Description of dataset

Due to uniqueness of survival data and the difficulty to collect them because of the follow-up procedure, many datasets are not applicable. However, we have chosen a publicly available dataset that includes clinical information about patients and their genes.

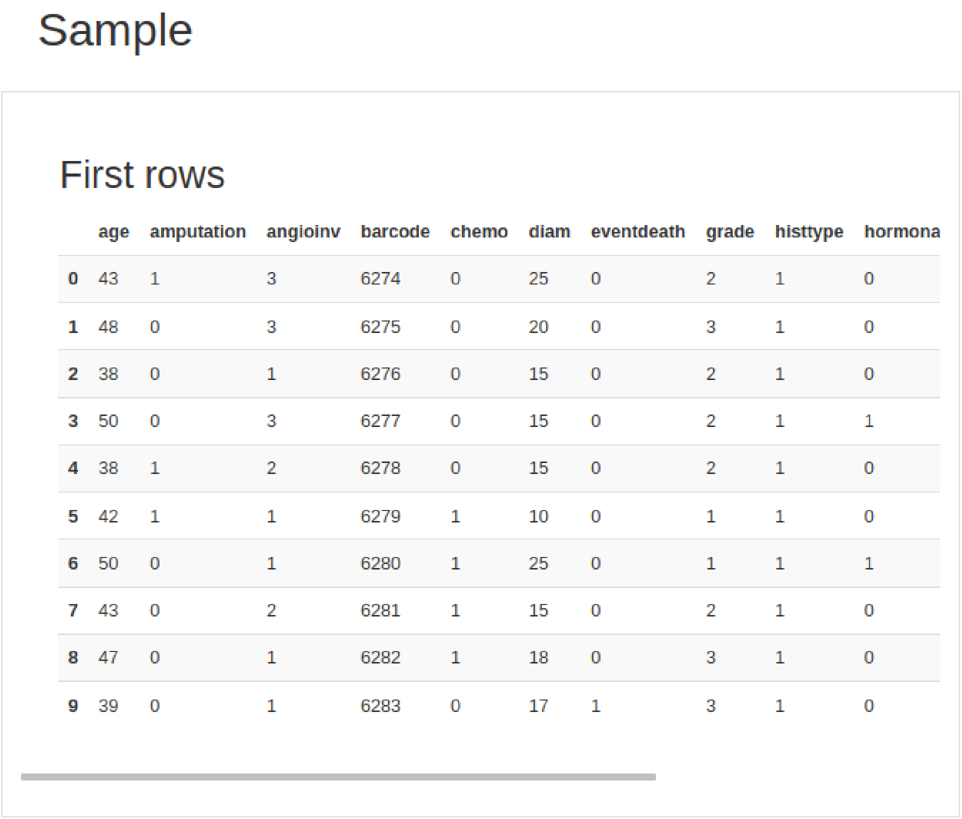
5.2.1 N.K.I. Breast Cancer Dataset

The first dataset is breast cancer dataset. It consists of 272 patients, with 16 features of clinical data including their survival time and the event of death for those that this have already happened. Additionally, their genes expression data consists of 1570 dimensions.
Breast Cancer clinical data include:

- age, integer variable
- chemo, binary variable
- hormonal, binary variable

- amputation, binary variable
- histtype, binary variable
- diam, integer variable
- posnodes, nominal variable
- grade, nominal variable
- angioinv, nominal variable
- lymphinfil, nominal variable
- barcode, integer variable (it was excluded)
- ID, integer variable (it was excluded)
- Patient, integer variable (it was excluded)
- time recurrence, float variable (it was excluded)

Below we collocate a sample of the breast cancer dataset.



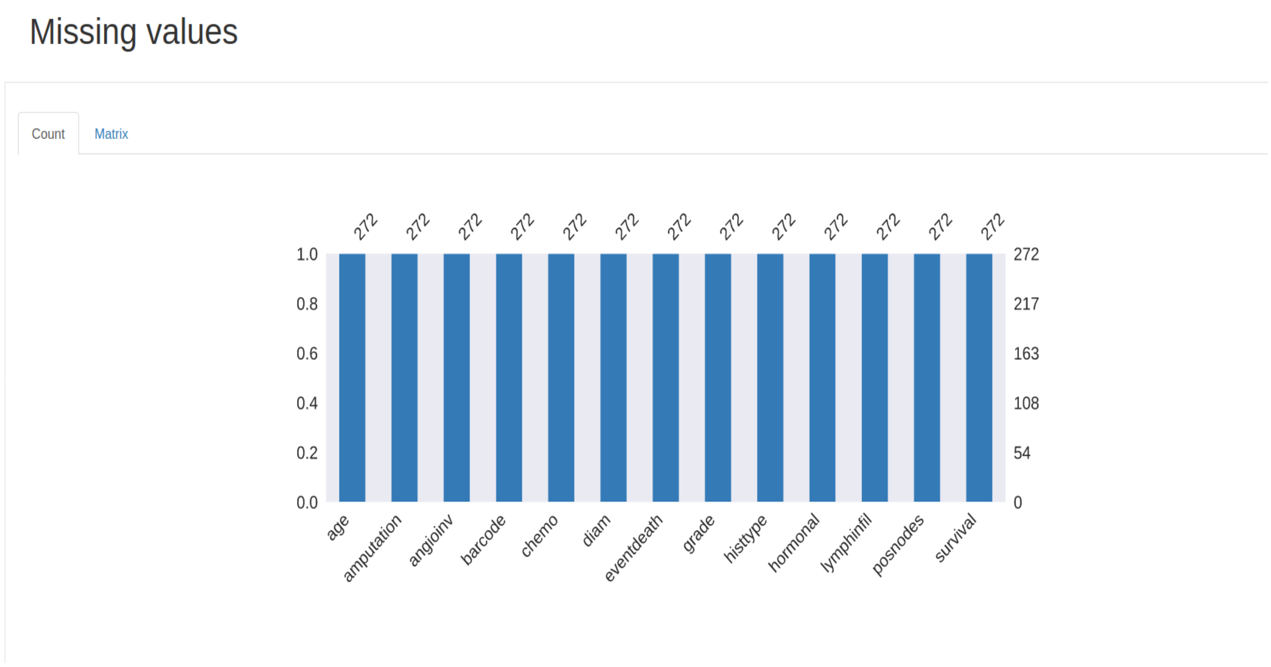
Sample

First rows

	age	amputation	angioinv	barcode	chemo	diam	eventdeath	grade	histtype	hormona
0	43	1	3	6274	0	25	0	2	1	0
1	48	0	3	6275	0	20	0	3	1	0
2	38	0	1	6276	0	15	0	2	1	0
3	50	0	3	6277	0	15	0	2	1	1
4	38	1	2	6278	0	15	0	2	1	0
5	42	1	1	6279	1	10	0	1	1	0
6	50	0	1	6280	1	25	0	1	1	1
7	43	0	2	6281	1	15	0	2	1	0
8	47	0	1	6282	1	18	0	3	1	0
9	39	0	1	6283	0	17	1	3	1	0

Figure 5.1 This figure shows first rows of clinical part of NKI breast cancer dataset.

Dataset was first checked for existence of missing values but were not found any. Below diagram shows that there are not any missing values.



Missing values

Count Matrix

Figure 5.2 This figure shows that this dataset is empty of missing values.

Next graph is about the proportion of censored patients where patients that already have experienced the event of interest are denoted with label “1” otherwise with label “0”.

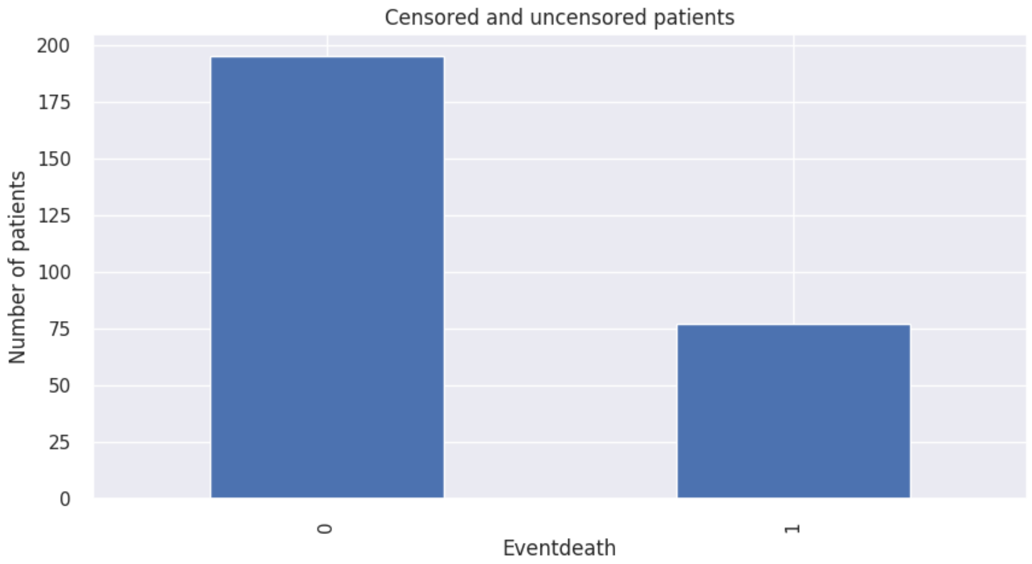


Figure 5.3 This figure shows how many of the patients have experienced the event of interest.

5.3 Selecting Significant Clinical Parameters

As discussed before in this phase we select the most significant clinical variables in terms of survivability but also those ones that participates most at the existence of the event of interest (death). Consequently, we used two algorithms one that selects features that play important role at survival namely Cox proportional hazard and the second one that selects significant features for the event of death namely Penalized logistic Regression algorithm. Below we collocate two diagrams presenting variables that have identified to be important for both algorithms.

Table 5.1 This table shows the significant features that identified from both Cox proportional hazard model and also Penalized Logistic Regression. Last column shows that the final selected features are the union of the two set of selected features.

Clinical Parameters	Variable Importance Based on event of death	Variable Importance Based on Survival variable	Final variables selected
Age	x	x	x
Chemo			
Hormonal			
Amputation			
Histtype		x	x
Diam	x		x
Posnodes		x	x
Grade	x	x	x
Angioinv	x		x
Lymphinfil			

As we can obtain in the table above the final clinical variables selected are the sum union of variables denoted as significant from Cox model and from Penalized logistic Regression model.

5.4 Review of feature selection - dimension reduction techniques

This semi-supervised approach was initially introduced by Bair (Bair et al. 2004) after that Zhang (Zhang et al. 2016) outperformed his algorithm by using a different feature selection technique namely Fast Correlation Based Filter. He obtained significantly improved results when tested his approach in the same datasets.

Zhang’s approach has the potential of further improvement and due to the existence of such a rich repository of feature selection techniques we did apply many of them, and we will present those with the better performance.

We validated our approach in a publicly available breast cancer dataset. Firstly, we selected the most significant clinical features using both penalized logistic regression and Cox proportional hazard model. Significant features have been marked with red in the previous section. Then we test two (and use one at each time) techniques in order to decide what is the optimal number of clusters that we will choose. Next, we apply clustering to identify potential cancer subtypes. In our first approach we used Relief algorithm and in the second we used MultiSURF feature selection technique to choose a subset of genes.

Furthermore, we designed a third approach in which we apply autoencoders to genes expression data. Autoencoder network is trained with the genes dataset having as labels the labels identified in the clustering phase. The figure below describes its architecture i.e., the number of layers and the number of input and output neurons. After autoencoders training and when the reconstruction error reduced satisfactorily a custom clustering layer will be used.

The extra custom clustering layer will be used as final layer. Consequently, the network learns a reduced representation of dataset from 1554 to 10 dimensions. After autoencoders training phase we use those 10 encoded feature vectors to feed them to the custom clustering layer and produce the probability each observation belongs to one of the predefined number of clusters. As stated earlier we use as ground truth the labels that we observed in the clustering phase.

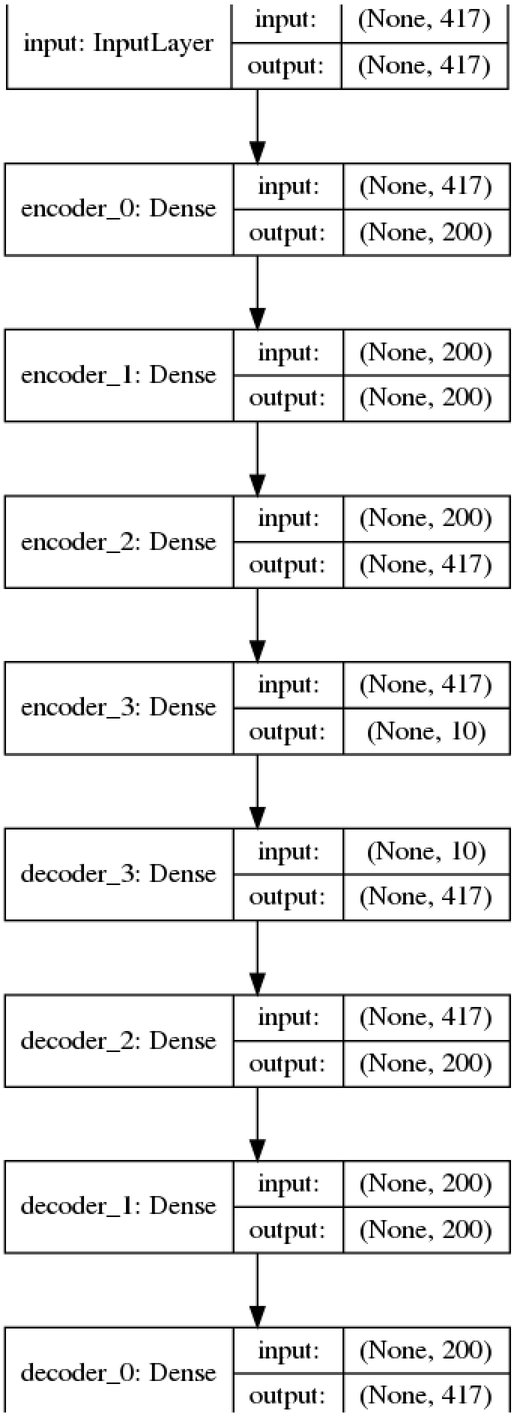


Figure 5.4 This figure shows the overall architecture of autoencoder used for dimensionality reduction

In the custom clustering layer weights were initialized based on K-means initial weights. Also, in this custom layer the probability for each observation was computed using t-distribution to measure similarity between each point and the centroid. After adding custom layer, we trained it using an auxiliary distribution as target $p_{ij} = \frac{q_{ij}^{2}/f_i}{\sum_{j'} q_{ij'}^{2}/f_{j'}}$ based on Xie's (Xie et al. 2016) approach. In this procedure we used KL divergence loss function which is a measure of how one probability distribution is different from a second, reference probability distribution. In this sense we want clustering layer to be trained on its confidence predictions.

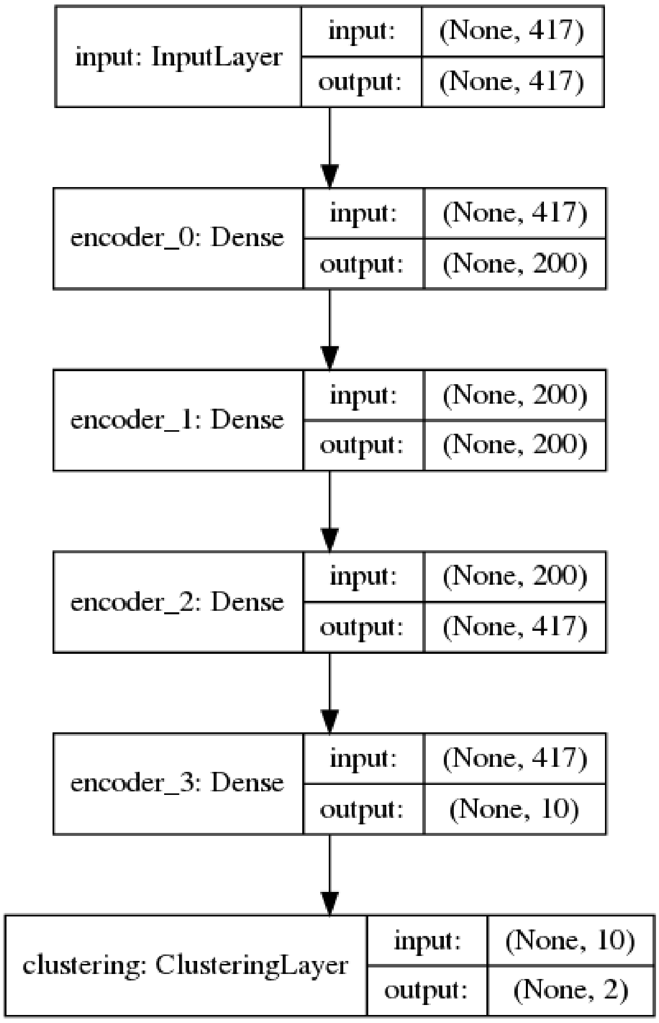


Figure 5.5 This figure shows the structure of final neural network used. As it is shown in the figure it has been used the encoder part of the autoencoder with a custom clustering layer as final layer.

Naive Bayes classifier and Decision Trees were applied in the encoded genes and the ground truth were the labels identified in the previous step.

In validation phase we applied log-rank test in predictions in order to obtain if the subgroup of patients that have been categorized as “patients with cancer subtype 1” differs in terms of survival from patients that have been categorized as “patients with cancer subtype 2”.

5.5 Clustering Results

In this section we present the Kaplan Meier curves for subgroups identified from clustering. First figure shows Kaplan Meier curves of subgroups discovered using significant clinical features and the second figure shows Kaplan Meier curves of two subgroups discovered from neural network and its clustering layer.

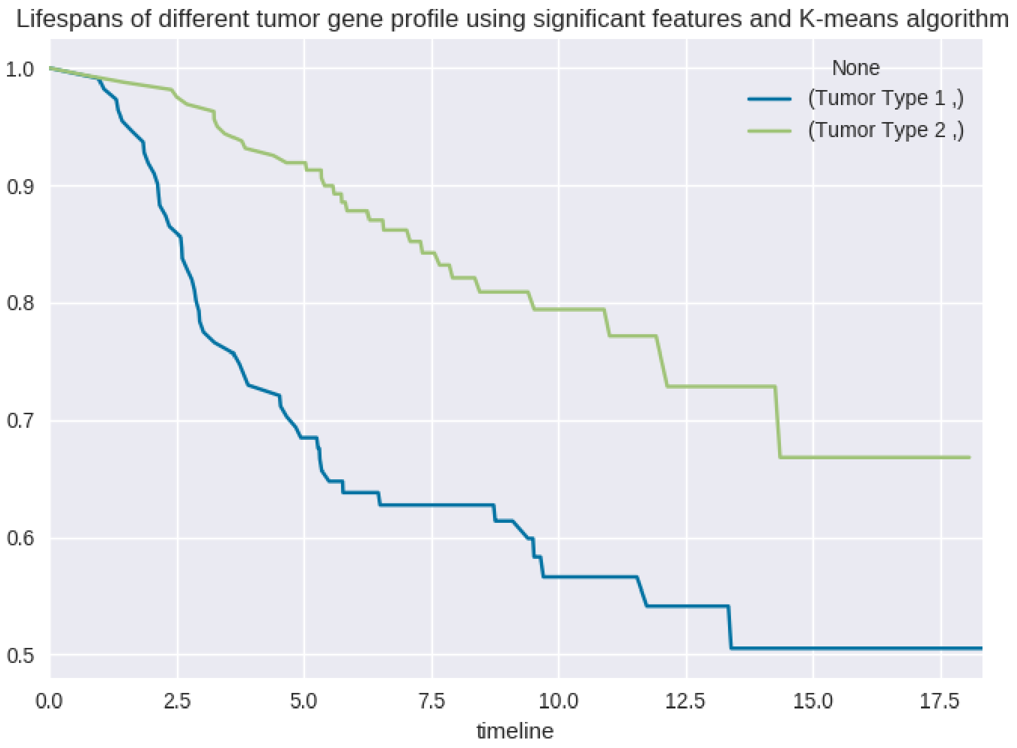


Figure 5.6 This figure shows the lifespan of two identified groups when we applied clustering with K-means (k=2) in the significant variables selected by Cox proportional model and penalized logistic Regression. The p-value of log-rank test for these two subgroups is 0.0003809.

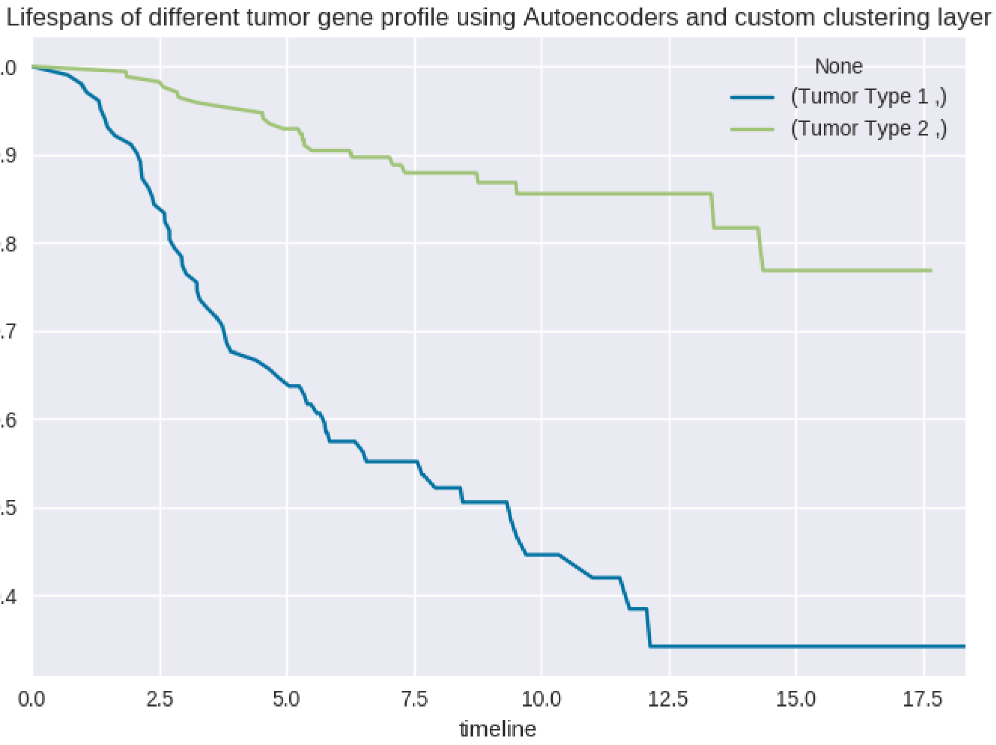


Figure 5.7 This figure describes the lifespans of two identified groups when we used together clinical parameters and genes expression data, in the autoencoder neural network which has a custom clustering layer as final layer in its architecture. The p-value of log-rank test for these two subgroups is 0.00004584.

5.6 Classification Accuracy

In our first experiment we applied Relief algorithm as feature selection technique. After feature selection we used Decision Tree and Naive Bayes classifier in order to obtain if our selected features can force a classifier to accurate classify a patient “never seen before”. Classifiers were trained with the most significant features selected each time and as targets for the feature selection were used labels extracted in clustering phase. After important features selection, a combined dataset was created with selected genes and the clustering labels and then the dataset was splitted into training and test set. Classifiers were trained with a number of training samples and then tested on test set. Results as presented below.

We present classification report and roc curves for both classifiers when features selected with Relief algorithm firstly and then for features selected by MultiSurf.

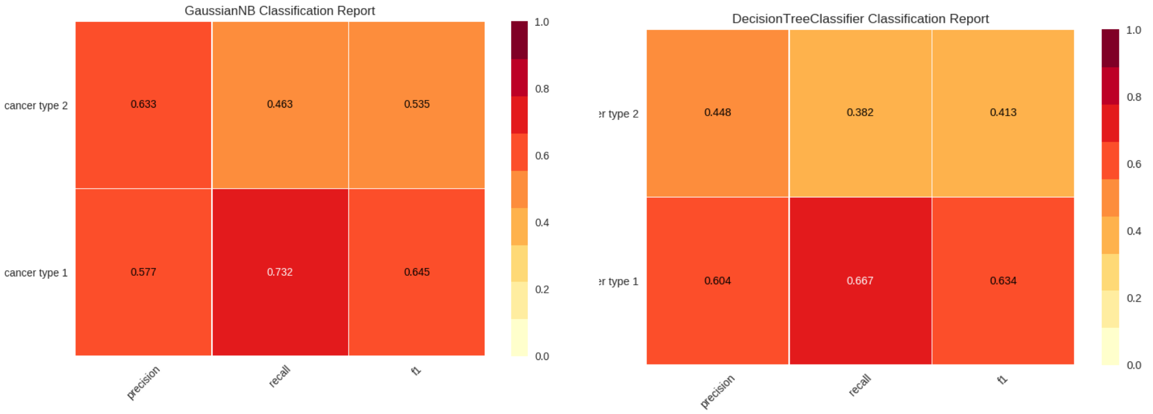


Figure 5.8.a

Figure 5.8.b

Figure 5.8 Classification report of Naive Bayes and Decision Tree classifiers when Relief algorithm have been used as feature selection technique for genes expression data.

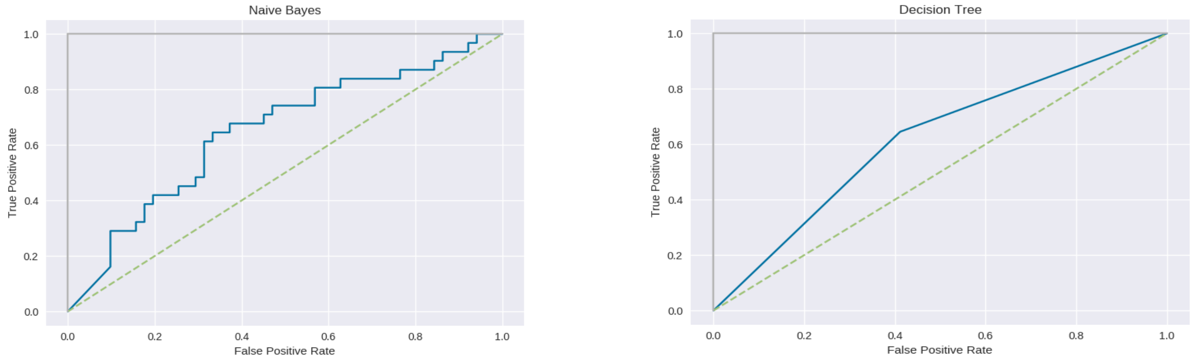


Figure 5.9.a

Figure 5.9.b

Figure 5.9 Roc curves for Naive Bayes and Decision Tree classifiers.

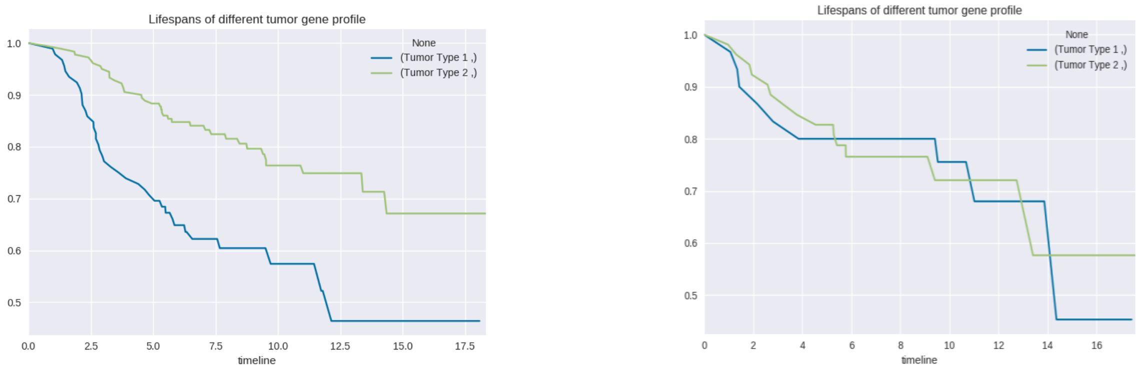


Figure 5.10.a

Figure 5.10.b

Figure 5.10 Kaplan Meier curves for Naive Bayes and Decision Tree classifiers.

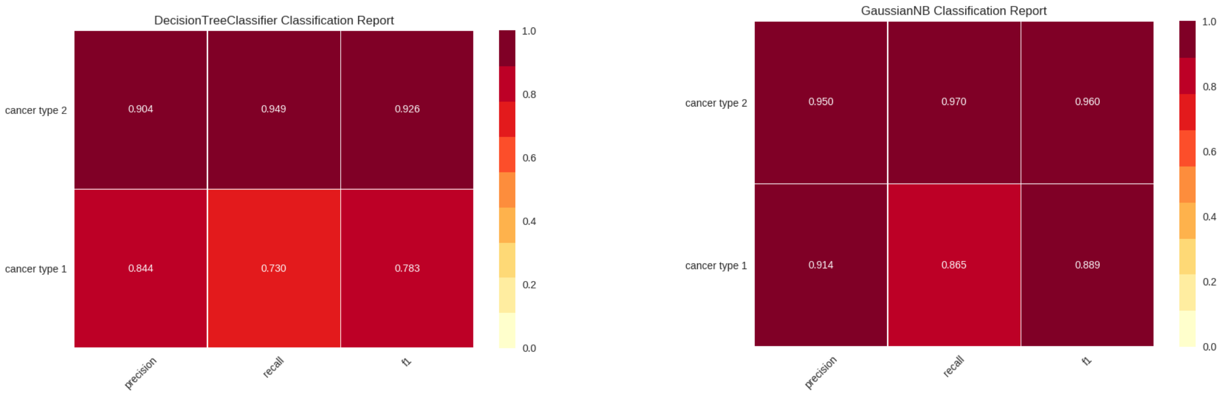


Figure 5.11.a

Figure 5.11.b

Figure 5.11 Classification Report for Decision Tree classifier and Naive Bayes classifier. MultiSurf has been used as Feature selection algorithm.

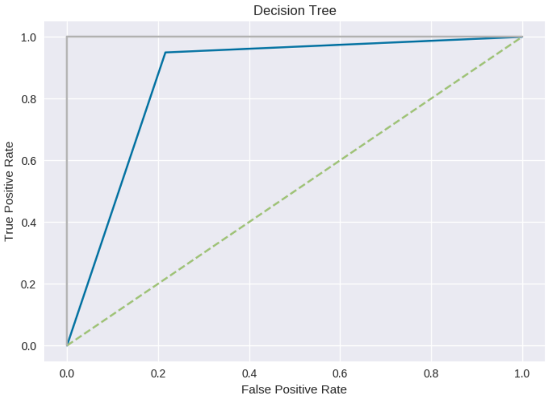


Figure 5.12.a

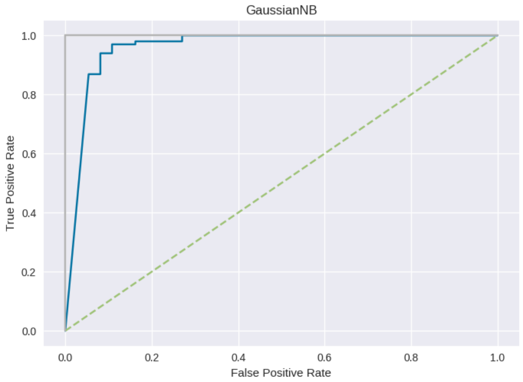


Figure 5.12.b

Figure 5.12 Roc Curve for Decision Tree and Naive Bayes. MultiSurf has been used as feature selection algorithm.

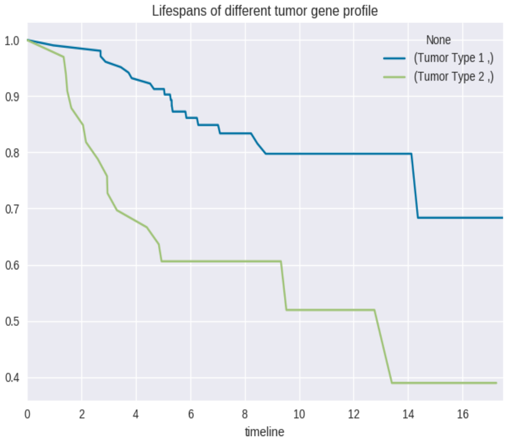


Figure 5.13.a

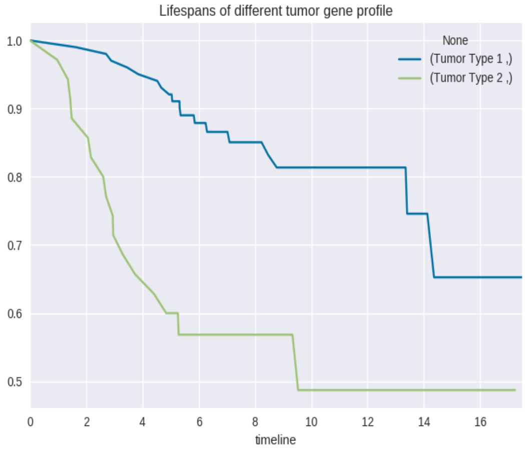


Figure 5.13.b

Figure 5.13 Kaplan Meier plot for predictions made for Decision Tree and Naive Bayes classifier. MultiSurf has been used as feature selection algorithm.

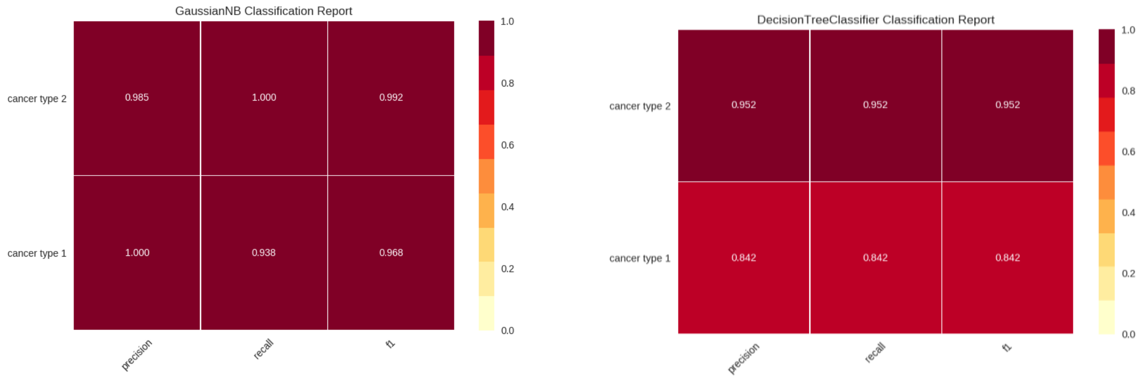


Figure 5.14.a

Figure 5.14.b

Figure 5.14 Classification Report for Decision Tree classifier and Naive Bayes classifier when autoencoder neural network has been used for dimension reduction.

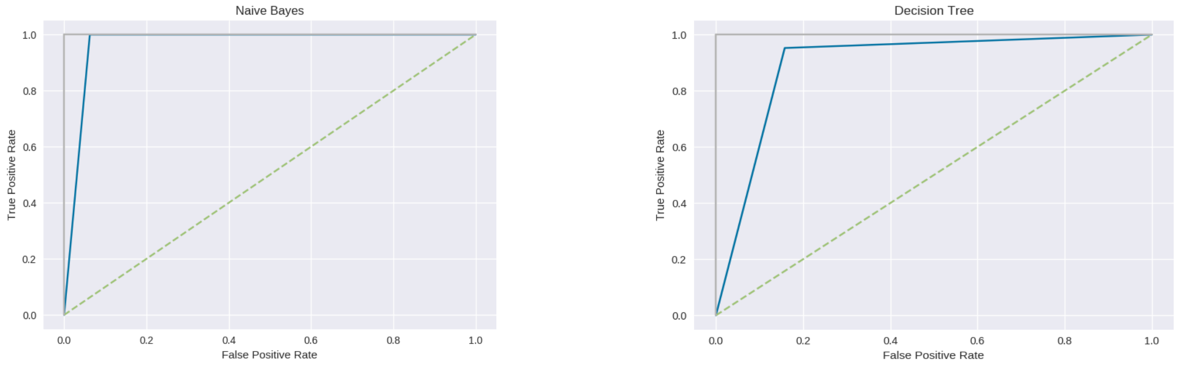


Figure 5.15.a

Figure 5.15.b

Figure 5.15 Roc Curve for Decision Tree and Naive Bayes when autoencoder neural network has been used for dimension reduction.

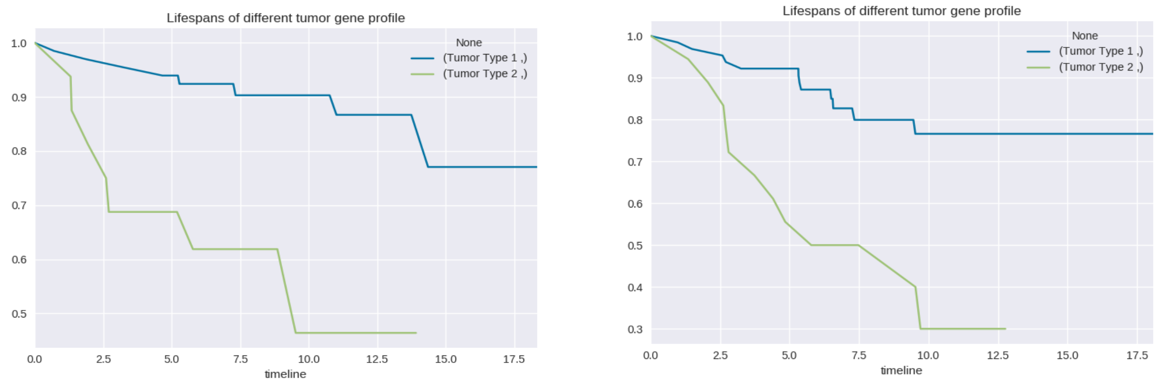


Figure 5.16.a

Figure 5.16.b

Figure 5.16 Kaplan Meier curves for predictions made for Decision Tree and Naive Bayes classifier when autoencoders neural network has been used for dimension reduction.

Table 5.2 In the table below it is shown the log-rank test p-values for every classifier and the feature selection or dimension reduction technique that have been used before each run.

Feature selection Method for clinical data	Feature Selection/ Dimensionality Reduction for genes expression data	Classifier	log-rank test (p-value)
Parameter selection using Cox Model and Penalized logistic Regression	Relief	Naive Bayes	0.02799961813
		Decision Tree	0.04820644890
	MultiSurf	Naive Bayes	0.00022567500
		Decision Tree	0.00068432300
	Autoencoders	Naive Bayes	0.00000458901
		Decision Tree	0.00000564654

5.7 Experimental Setup

All experiments were developed using Pycharm as the integrated development environment for Python language and functions used were taken from packages: pandas, numpy, scikit-learn, keras, tensorflow, scipy, lifelines, py-survival, scikit-survival, matplotlib, pandas_profiling, yellowbrick. All experiments were carried out in a laptop computer with the following characteristics: CPU: Intel® Core™ i7-8750H and RAM: 16GB.

Chapter 6

6.1 Conclusions and future work

The main objective of this study was to utilize unlabeled clinical variables and genes expression data in order to identify risk groups that may indicate cancer subtypes. We presented three approaches to achieve our goal. The main idea was based on Bair's idea (Bair et al. 2004) and also Zhang's approach (Zhang et al. 2016). We used clinical data in order to identify biologically and also clinically relevant clusters. In this sense we applied variable selection in clinical data using most significant variables that Cox proportional hazard model and penalized logistic Regression suggested. After this phase using only significant clinical features we applied K-means and we identified two clusters. Number of k was discovered using silhouette method. Labels extracted from K-means algorithms were then used for variable selection of the genes data for the same patients' cohort. In our early approaches we chosen two feature selection techniques (Relief and then MultiSurf) for genes expression data having as label the clustering label observed in the previous phase. Finally, the most significant genes were utilized to train classifiers and to predict the cluster (which in real world life would be a cancer subtype) for future groups of patients. In our third approach we constructed an autoencoder neural network using a custom clustering layer as final layer which was used instead of feature selection techniques after the significant clinical features were selected in the two previous approaches. The encoded dimensions produced by the autoencoder were used as inputs to the clustering layer and finally each observation was categorized in one of the two predefined clusters. We compare those methods by measuring the log-rank test of the two groups discovered (in each experiment) in order to check if the applied clustering technique has created two groups that differs in terms of survival.

Our methodology contributions are a) the meaningful clustering in terms of lifetimes distribution on the clinical data b) significant genes subset identification which is also associated with the survival outcome of the patient. Finally, we argue that there is a lot of space for further experimentation a) using different machine learning approaches for feature selection or dimensionality reduction techniques in order to select more informative clinical and genes expression features that can be used for cancer subtype identification of any future patient or

b) explore the usage of clustering algorithms that will be able to consider survival information along with the conventional features.

References

Greene, C. S., Himmelstein, D. S., Kiralis, J., & Moore, J. H. (2010). The Informative Extremes: Using Both Nearest and Farthest Individuals Can Improve Relief Algorithms in the Domain of Human Genetics. Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics Lecture Notes in Computer Science, 182-193. doi:10.1007/978-3-642-12211-8_16

Kira, K., & Rendell, L. A. (1992). A Practical Approach to Feature Selection. Machine Learning Proceedings 1992, 249-256. doi:10.1016/b978-1-55860-247-2.50037-1

Wei, W., Sun, Z., Silveira, W. A., Yu, Z., Lawson, A., Hardiman, G., . . . Chung, D. (2018). Semi-supervised identification of cancer subgroups using survival outcomes and overlapping grouping information. Statistical Methods in Medical Research, 28(7), 2137-2149. doi:10.1177/0962280217752980

Wasito, I., & Veritawati, I. (2012). Subtype of Cancer Identification for Patient Survival Prediction Using Semi Supervised Method. Journal of Convergence Information Technology, 7(14), 215-222. doi:10.4156/jcit.vol7.issue14.25

Koestler, D. C., Marsit, C. J., Christensen, B. C., Karagas, M. R., Bueno, R., Sugarbaker, D. J., . . . Houseman, E. A. (2010). Semi-supervised recursively partitioned mixture models for identifying cancer subtypes. Bioinformatics, 26(20), 2578-2585. doi:10.1093/bioinformatics/btq470

Zhang, W., Tang, J., & Wang, N. (2016). Using the machine learning approach to predict patient survival from high-dimensional survival data. 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). doi:10.1109/bibm.2016.7822695

Veer, L. J., Dai, H., Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., . . . Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. Nature, 415(6871), 530-536. doi:10.1038/415530a

Vijver, M. J., He, Y. D., Veer, L. J., Dai, H., Hart, A. A., Voskuil, D. W., . . . Bernards, R. (2002). A Gene-Expression Signature as a Predictor of Survival in Breast Cancer. New England Journal of Medicine, 347(25), 1999-2009. doi:10.1056/nejmoa021967

Gao, F., Wang, W., Tan, M., Zhu, L., Zhang, Y., Fessler, E., . . . Wang, X. (2019). DeepCC: A novel deep learning-based framework for cancer molecular subtype classification. Oncogenesis, 8(9). doi:10.1038/s41389-019-0157-8

Tjaden, B., & Cohen, J. (2006). A Survey of Computational Methods Used in Microarray Data Interpretation. Applied Mycology and Biotechnology, 161-178. doi:10.1016/s1874-5334(06)80010-9

Young, J. D., Cai, C., & Lu, X. (2017). Unsupervised deep learning reveals prognostically relevant subtypes of glioblastoma. BMC Bioinformatics, 18(S11). doi:10.1186/s12859-017-1798-2

Macqueen, J. (1964). A Mathematical Approach to the Problem of Achieving Selective Biological Effects. Biometrics, 20(1), 130. doi:10.2307/2527622

Murtagh, F., & Contreras, P. (2011). Algorithms for hierarchical clustering: An overview. WIREs Data Mining and Knowledge Discovery, 2(1), 86-97. doi:10.1002/widm.53

Koestler, D. C., Marsit, C. J., Christensen, B. C., Karagas, M. R., Bueno, R., Sugarbaker, D. J., . . . Houseman, E. A. (2010). Semi-supervised recursively partitioned mixture models for identifying cancer subtypes. Bioinformatics, 26(20), 2578-2585. doi:10.1093/bioinformatics/btq470

Cheerla, A., & Gevaert, O. (2019). Deep Learning with Multimodal Representation for Pan Cancer Prognosis Prediction. doi:10.1101/577197

Chen, Yang, Steve, Sun, Yijun, & Goodison. (2019, October 11). Deep-learning approach to identifying cancer subtypes using high-dimensional genomic data. Retrieved from http://doi.org/10.1093/bioinformatics/btz769

The Cancer Genome Atlas - Cancers Selected for Study. (n.d.). Retrieved from https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga/studied-cancers

Benam, K. H., Gilchrist, S., Kleensang, A., Satz, A. B., Willett, C., & Zhang, Q. (2019). Exploring new technologies in biomedical research. Drug Discovery Today, 24(6), 1242-1247. doi:10.1016/j.drudis.2019.04.001

Livieris, I. E. (2018). Identification of Blood Cell Subtypes from Images Using an Improved SSL Algorithm. Biomedical Journal of Scientific & Technical Research, 9(1). doi:10.26717/bjstr.2018.09.001755

Roder, H., Oliveira, C., Net, L., Linstid, B., Tsypin, M., & Roder, J. (2019). Robust identification of molecular phenotypes using semi-supervised learning. BMC Bioinformatics, 20(1). doi:10.1186/s12859-019-2885-3

Lee, S., & Lim, H. (2019). Review of statistical methods for survival analysis using genomic data. Genomics & Informatics, 17(4). doi:10.5808/gi.2019.17.4.e41

Witten, D. M., & Tibshirani, R. (2009). Survival analysis with high-dimensional covariates. Statistical Methods in Medical Research, 19(1), 29-51. doi:10.1177/0962280209105024

Kuijjer, M. L., Paulson, J. N., Salzman, P., Ding, W., & Quackenbush, J. (2018). Cancer subtype identification using somatic mutation data. British Journal of Cancer, 118(11), 1492-1501. doi:10.1038/s41416-018-0109-7

Begg, C. B. (2010). A strategy for distinguishing optimal cancer subtypes. International Journal of Cancer, 129(4), 931-937. doi:10.1002/ijc.25714

Sinkala, M., Mulder, N., & Martin, D. (2020). Machine Learning and Network Analyses Reveal Disease Subtypes of Pancreatic Cancer and their Molecular Characteristics. Scientific Reports, 10(1). doi:10.1038/s41598-020-58290-2

Yang, S. X., & Polley, E. C. (2019). Systemic treatment and radiotherapy, breast cancer subtypes, and survival after long-term clinical follow-up. Breast Cancer Research and Treatment, 175(2), 287-295. doi:10.1007/s10549-019-05142-x

Kittaneh, M., Montero, A. J., & Glück, S. (2013). Molecular Profiling for Breast Cancer: A Comprehensive Review. Biomarkers in Cancer, 5. doi:10.4137/bic.s9455

Associations between Two Seemingly "Contradictory" Diseases: Hereditary Hemorrhagic Telangiectasia and Pulmonary Hypertension. (2016). Med One. doi:10.20900/mo.20160025

Goodfellow, I., Bengio, Y., & Courville, A. (2017). Deep learning. The MIT Press.

Bussy, S., Veil, R., Looten, V., Burgun, A., Gaïffas, S., Guilloux, A., . . . Jannot, A. (2019). Comparison of methods for early-readmission prediction in a high-dimensional heterogeneous covariates and time-to-event outcome framework. BMC Medical Research Methodology, 19(1). doi:10.1186/s12874-019-0673-4

Pölsterl, S., Conjeti, S., Navab, N., & Katouzian, A. (2016). Survival analysis for high-dimensional, heterogeneous medical data: Exploring feature extraction as an alternative to feature selection. Artificial Intelligence in Medicine, 72, 1-11. doi:10.1016/j.artmed.2016.07.004

Singh, R., & Mukhopadhyay, K. (2011). Survival analysis in clinical trials: Basics and must know areas. Perspectives in Clinical Research, 2(4), 145. doi:10.4103/2229-3485.86872

Liu, & Xian. (2012). Survival Analysis. John Wiley & Sons.

Mitchell, T. M. (2017). Machine learning. McGraw Hill.

Guo, Y., Shang, X., & Li, Z. (2019). Identification of cancer subtypes by integrating multiple types of transcriptomics data with deep learning in breast cancer. Neurocomputing, 324, 20-30. doi:10.1016/j.neucom.2018.03.072

Chen, R., Yang, L., Goodison, S., & Sun, Y. (2019). Deep-learning approach to identifying cancer subtypes using high-dimensional genomic data. Bioinformatics. doi:10.1093/bioinformatics/btz769

Shen, R., Olshen, A. B., & Ladanyi, M. (2009). Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. Bioinformatics, 25(22), 2906-2912. doi:10.1093/bioinformatics/btp543

Gao, F., Wang, W., Tan, M., Zhu, L., Zhang, Y., Fessler, E., . . . Wang, X. (2019). DeepCC: A novel deep learning-based framework for cancer molecular subtype classification. Oncogenesis, 8(9). doi:10.1038/s41389-019-0157-8

Vasudevan, P., & Murugesan, T. (2018). Cancer Subtype Discovery Using Prognosis-Enhanced Neural Network Classifier in Multi Genomic Data. Technology in Cancer Research & Treatment, 17, 153303381879050. doi:10.1177/1533033818790509

He, Z., Zhang, J., Yuan, X., Xi, J., Liu, Z., & Zhang, Y. (2019). Stratification of Breast Cancer by Integrating Gene Expression Data and Clinical Variables. Molecules, 24(3), 631. doi:10.3390/molecules24030631

Souza, J. T., Francisco, A. C., & Macedo, D. C. (2019). Dimensionality Reduction in Gene Expression Data Sets. IEEE Access, 7, 61136-61144. doi:10.1109/access.2019.2915519

Wang, H., & Laan, M. J. (2011). Dimension reduction with gene expression data using targeted variable importance measurement. BMC Bioinformatics, 12(1). doi:10.1186/1471-2105-12-312

Urbanowicz, R. J., Olson, R. S., Schmitt, P., Meeker, M., & Moore, J. H. (2018). Benchmarking relief-based feature selection methods for bioinformatics data mining. Journal of Biomedical Informatics, 85, 168-188. doi:10.1016/j.jbi.2018.07.015

Lee, J., Ciccarello, S., Acharjee, M., & Das, K. (2017). Dimension reduction of gene expression data. Journal of Statistical Theory and Practice, 12(2), 450-461. doi:10.1080/15598608.2017.1413456

Paul, A., Sil, J., & Mukhopadhyay, C. D. (2014). Dimension Reduction of Gene Expression Data for Designing Optimized Rule Base Classifier. Recent Advances in Information Technology Advances in Intelligent Systems and Computing, 133-140. doi:10.1007/978-81-322-1856-2_15

Bair, E., & Tibshirani, R. (2004). Semi-Supervised Methods to Predict Patient Survival from Gene Expression Data. PLoS Biology, 2(4). doi:10.1371/journal.pbio.0020108

Stephanie. (2018, March 21). Log-Rank Test , Weighted LRT:, Stratified LRT: Definitions, Examples. Retrieved from https://www.statisticshowto.datasciencecentral.com/log-rank-test/

Francis, & Don. (2018, March 11). Penalized Logistic Regression Essentials in R: Ridge, Lasso and Elastic Net. Retrieved from http://www.sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-lasso-and-elastic-net/

Heagerty, P. J., & Zheng, Y. (2005). Survival Model Predictive Accuracy and ROC Curves. Biometrics, 61(1), 92-105. doi:10.1111/j.0006-341x.2005.030814.x

John P. Klein and Melvin L. Moeschberger. (2005). Survival Analysis: Techniques for Censored and Truncated Data. Springer Science & Business Media.

Wang, P., Li, Y., & Reddy, C. K. (2019). Machine Learning for Survival Analysis. ACM Computing Surveys, 51(6), 1-36. doi:10.1145/3214306

Elisa T. Lee and John Wang. (2003). Statistical Methods for Survival Data Analysis. Vol. 476. John Wiley & Sons.

Survival Function. (n.d.). Retrieved from http://mathworld.wolfram.com/SurvivalFunction.html

Dunn, O. J., & Clark, V. A. (2009). Basic Statistics. doi:10.1002/9780470496862

Kaplan, E. L., & Meier, P. (1958). Nonparametric Estimation from Incomplete Observations. Journal of the American Statistical Association, 53(282), 457-481. doi:10.1080/01621459.1958.10501452

Cutler, S. J., & Ederer, F. (1958). Maximum utilization of the life table method in analyzing survival. Journal of Chronic Diseases, 8(6), 699-712. doi:10.1016/0021-9681(58)90126-7

Nelson, W. (2000). Theory and Applications of Hazard Plotting for Censored Failure Data. Technometrics, 42(1), 12-25. doi:10.1080/00401706.2000.10485975

Aalen, O. (1978). Nonparametric Inference for a Family of Counting Processes. The Annals of Statistics, 6(4), 701-726. doi:10.1214/aos/1176344247

Cox, D. R. (1972). Regression Models and Life-Tables. Journal of the Royal Statistical Society: Series B (Methodological), 34(2), 187-202. doi:10.1111/j.2517-6161.1972.tb00899.x

Breslow, N., & Crowley, J. (1981). Discussion of Paper by D. Oakes. International Statistical Review / Revue Internationale De Statistique, 49(3), 255. doi:10.2307/1402608

Wang, P., Li, Y., & Reddy, C. K. (2019). Machine Learning for Survival Analysis. ACM Computing Surveys, 51(6), 1-36. doi:10.1145/3214306

Tobin, J. (1958). Estimation of Relationships for Limited Dependent Variables. Econometrica, 26(1), 24. doi:10.2307/1907382

Buckley, J., & James, I. (1979). Linear regression with censored data. Biometrika, 66(3), 429-436. doi:10.1093/biomet/66.3.429

Ciampi, A., Bush, R. S., Gospodarowicz, M., & Till, J. E. (1981). An approach to classifying prognostic factors related to survival experience for non-Hodgkin's lymphoma patients: Based on a series of 982 patients: 1967–1975. Cancer, 47(3), 621-627. doi:10.1002/1097-0142(19810201)47:33.0.co;2-0

Bou-Hamad, I., Larocque, D., & Ben-Ameur, H. (2011). A review of survival trees. Statistics Surveys, 5(0), 44-71. doi:10.1214/09-ss047

Lisboa, P., Wong, H., Harris, P., & Swindell, R. (2003). A Bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer. Artificial Intelligence in Medicine, 28(1), 1-25. doi:10.1016/s0933-3657(03)00033-2

Li, Y., Vinzamuri, B., & Reddy, C. K. (2016). Regularized Weighted Linear Regression for High-dimensional Censored Data. Proceedings of the 2016 SIAM International Conference on Data Mining. doi:10.1137/1.9781611974348.6

Fard, M. J., Wang, P., Chawla, S., & Reddy, C. K. (2016). A Bayesian Perspective on Early Stage Event Prediction in Longitudinal Data. IEEE Transactions on Knowledge and Data Engineering, 28(12), 3126-3139. doi:10.1109/tkde.2016.2608347

Faraggi, D., & Simon, R. (1995). A neural network model for survival data. Statistics in Medicine, 14(1), 73-82. doi:10.1002/sim.4780140108

Biganzoli, E., Boracchi, P., Mariani, L., & Marubini, E. (1998). Feed forward neural networks for the analysis of censored survival data: A partial logistic regression approach. Statistics in Medicine, 17(10), 1169-1186. doi:10.1002/(sici)1097-0258(19980530)17:103.0.co;2-d

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC Medical Research Methodology, 18(1). doi:10.1186/s12874-018-0482-1

Zhu, X., Yao, J., & Huang, J. (2016). Deep convolutional neural network for survival analysis with pathological images. 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). doi:10.1109/bibm.2016.7822579

Yao, J., Zhu, X., Zhu, F., & Huang, J. (2017). Deep Correlational Learning for Survival Prediction from Multi-modality Data. Lecture Notes in Computer Science Medical Image Computing and Computer-Assisted Intervention − MICCAI 2017, 406-414. doi:10.1007/978-3-319-66185-8_46

Ching, T., Zhu, X., & Garmire, L. X. (2018). Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data. PLOS Computational Biology, 14(4). doi:10.1371/journal.pcbi.1006076

Gensheimer, M. F., & Narasimhan, B. (2019). A scalable discrete-time survival model for neural networks. PeerJ, 7. doi:10.7717/peerj.6257

Giunchiglia, E., Nemchenko, A., & Schaar, M. V. (2018). RNN-SURV: A Deep Recurrent Model for Survival Analysis. Artificial Neural Networks and Machine Learning – ICANN 2018 Lecture Notes in Computer Science, 23-32. doi:10.1007/978-3-030-01424-7_3

Shivaswamy, P. K., Chu, W., & Jansche, M. (2007). A Support Vector Approach to Censored Targets. Seventh IEEE International Conference on Data Mining (ICDM 2007). doi:10.1109/icdm.2007.93

Khan, F. M., & Zubek, V. B. (2008). Support Vector Regression for Censored Data (SVRc): A Novel Tool for Survival Analysis. 2008 Eighth IEEE International Conference on Data Mining. doi:10.1109/icdm.2008.50

Kiaee, F., Sheikhzadeh, H., & Mahabadi, S. E. (2016). Relevance Vector Machine for Survival Analysis. IEEE Transactions on Neural Networks and Learning Systems, 27(3), 648-660. doi:10.1109/tnnls.2015.2420611

Hothorn, T., Lausen, B., Benner, A., & Radespiel-Tröger, M. (2003). Bagging survival trees. Statistics in Medicine, 23(1), 77-91. doi:10.1002/sim.1593

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests. The Annals of Applied Statistics, 2(3), 841-860. doi:10.1214/08-aoas169

Hothorn, T. (2005). Survival ensembles. Biostatistics, 7(3), 355-373. doi:10.1093/biostatistics/kxj011

Ishwaran, H., Kogalur, U. B., Chen, X., & Minn, A. J. (2011). Random survival forests for high-dimensional data. Statistical Analysis and Data Mining, 4(1), 115-132. doi:10.1002/sam.10103

Chen, Y., Jia, Z., Mercola, D., & Xie, X. (2013). A Gradient Boosting Algorithm for Survival Analysis via Direct Optimization of Concordance Index. Computational and Mathematical Methods in Medicine, 2013, 1-8. doi:10.1155/2013/873595

Wieczorkowska, A., & Jarmulski, W. (2020). Optimizing C-Index via Gradient Boosting in Medical Survival Analysis. Complex Pattern Mining Studies in Computational Intelligence, 33-45. doi:10.1007/978-3-030-36617-9_3

Mayr, A., Hofner, B., & Schmid, M. (2016). Boosting the discriminatory power of sparse survival models via optimization of the concordance index and stability selection. BMC Bioinformatics, 17(1). doi:10.1186/s12859-016-1149-8

Li, H., & Luan, Y. (2005). Boosting proportional hazards models using smoothing splines, with applications to high-dimensional microarray data. Bioinformatics, 21(10), 2403-2409. doi:10.1093/bioinformatics/bti324

Vinzamuri, B., Li, Y., & Reddy, C. K. (2014). Active Learning based Survival Regression for Censored Data. Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14. doi:10.1145/2661829.2662065

Li, Y., Vinzamuri, B., & Reddy, C. K. (2016). Regularized Weighted Linear Regression for High-dimensional Censored Data. Proceedings of the 2016 SIAM International Conference on Data Mining. doi:10.1137/1.9781611974348.6

Li, Y., Wang, L., Wang, J., Ye, J., & Reddy, C. K. (2016). Transfer Learning for Survival

Analysis via Efficient L2,1-Norm Regularized Cox Regression. 2016 IEEE 16th International Conference on Data Mining (ICDM). doi:10.1109/icdm.2016.0034

Li, Y., Wang, J., Ye, J., & Reddy, C. K. (2016). A Multi-Task Learning Formulation for Survival Analysis. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16. doi:10.1145/2939672.2939857

ANNEX 1

Parametric Survival Models

Parametric censored regression models for survival data assume that the survival times follow a particular theoretical distribution (Lee and Wang 2003). Most commonly used distributions in parametric censored regression models are: **normal, exponential, weibull, logistic, log-logistic and log-normal.**

Under the hypothesis that survival times of all instances in the data follow on of these distributions, the model is referred as linear regression model.

The **maximum likelihood estimation (MLE)** method (Lee and Wang 2003) can be utilized in order to estimate the parameters for these models. If the number of instances is N with c censored observations and *(N-c)* uncensored observations and use $\beta = (\beta_1, \beta_2, \ldots, \beta_P)^T$ as a general notation to denote the set of all parameters then the death density function *f(t)* and the survival function *S(t)* of the survival time can be represented as *f(t,β)* and *S(t,β)*, respectively.

For a given censored instance i, the actual survival time will not be available but we can conclude that the instance i did not experience the event of interest before the censoring time C_i therefore the value of the survival function $S(C_i,\beta)$ will be a probability closed to 1. In contrast the death density function $f(T_i, \beta)$ will have a high probability value, if the event occurs for instance *i* at T_i . Thus, we can denote $\prod_{\delta_i=1} f(T_i, \beta)$ as the joint probability of the *c* censored observations and $\prod_{\delta_i=1} S(T_i, \beta)$ to represent the joint probability of all the c censored observations. Therefore, we can estimate the parameters β by optimizing the likelihood function of all N instances in the form of

$$L(\beta) = \prod_{\delta_i=1} f(T_i, \beta) \prod_{\delta_i=1} S(T_i, \beta) \qquad (12)$$

Below table shows that the death density function and its corresponding survival function *S(t)* and hazard function *h(t)* for these commonly used distributions.

Distribution	PDF *f(t)*	Survival *S(t)*	Hazard *h(t)*
Exponential	$\lambda exp(-\lambda t)$	$exp(-\lambda t)$	λ
Weibull	$\lambda\kappa t^{\kappa-1} exp(-\lambda t^{k})$	$exp(-\lambda t^{k})$	$\lambda\kappa t^{k-1}$
Log- logistic	$\frac{\lambda\kappa t^{k-1}}{(1+\lambda t^{k})^2}$	$\frac{1}{1+\lambda t^{k}}$	$\frac{\lambda k t^{k-1}}{1+\lambda t^{k}}$
Logistic	$\frac{e^{-(t-\mu)/\sigma}}{\sigma(1+e^{-(t-\mu)/\sigma})^2}$	$\frac{e^{-(t-\mu)/\sigma}}{1+\varepsilon^{-(\tau-\mu)/\sigma}}$	$\frac{1}{\sigma(1+e^{-(t-\mu)/\sigma)}}$
Normal	$\frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(t-\mu)^2}{2\sigma^2})$	$1-\Phi(\frac{t-\mu}{\sigma})$	$\frac{1}{\sqrt{2\pi\sigma}\ (1-\Phi((t-\mu)/\sigma))} exp(-\frac{(t-\mu)^2}{2\sigma^2})$
Log-Normal	$\frac{1}{\sqrt{2\pi}\sigma t} exp(-\frac{(log(t)-\mu)^2}{2\sigma^2}$	$1-\Phi(\frac{log(t)-\mu)}{\sigma})$	$\frac{\frac{1}{\sqrt{2\pi}\sigma t}\ exp(-(log(t)-\mu)^2/2}{1-\Phi\,\frac{(log(t)-\mu)}{\sigma}}$

A fundamental statistical technique is linear regression and it is one of the most commonly used approach when we want to make continuous predictions. Although we cannot apply linear regression in survival analysis problems because we have censored observations in our data. As it is mentioned before censored observations stand for observations that the actual event times are missing.

There have been a lot of different approaches to extend linear regression in order to handle censored data. In **Tobit Regression** (Tobin 1954; Wang 2019) there is a latent variable y^* and there is the assumption that it linearly depends on X via the parameter β as $y^* = X\beta + \varepsilon$, $\varepsilon \sim N(0, \sigma^2)$, where ε is a normally distributed error term. Then, for the i_{th}instance, the observable variable $y_i$will be $y_i{}^*$if > 0, otherwise it will be 0. Consequently, if the latent variable is above zero, the observed variable equals to the latent variable and zero otherwise. Based on the latent variable, the parameters in the model can be estimated with maximum likelihood estimation (MLE) method that is was mentioned before.

At **Buckley-James Regression** (Buckley & James 1979; Wang 2019) survival time of censored instances is estimated based on the Kaplan - Meier estimation method. Then it is fitted a linear (AFT) model by simultaneously considering the survival times of uncensored instances and the approximate survival time of the uncensored instances.

Yan Li (Li et al. 2016**)** in his paper proposed a regression model with weighted least squares to handle the survival prediction in the presence of censored instances. He also employs the elastic net penalty term for inducing sparsity into the linear model in order to handle high dimensional data.

Under the hypothesis that the logarithm of the survival times of all instances follow these distributions, the problem can be analyzed using the **accelerated failure time model (AFT)**.

Accelerated failure time model (AFT) assumes that the variable can affect the time to the event of the interest of an instant by some constant factor (Lee and Wang 2003).

In (AFT) model it is assumed that the relationship of the logarithm of survival time T and the covariates is linear and can be written in the following form:

$$ln(T) = X\beta + \sigma\varepsilon \qquad (13)$$

Where X is the covariate matrix, β represents the coefficient vector, $\sigma(\sigma>0)$ is an unknown scale parameter, and ε is an error variable that follows a similar distribution to $ln(T)$. There is the assumption that ε follow any of the distributions mentioned in table (**above)**. Consequently, survival depends on the covariate and the underlying distribution. Accelerated failure time (AFT) model differs from regular linear regression because considers censored information in the survival analysis problem.

Non-Parametric Survival Models

As it is known in real world cases a theoretical distribution does not suits perfect to data. In these cases non parametric methods are more efficient. Most popular among all non parametric models is Kaplan Meier’s model. Kaplan and Meier (Kaplan and Meier 1958) developed the Kaplan-Meier (KM) Curve or the product-limit (PL) estimator.

The Kaplan-Meier estimator is used to estimate the survival function using the actual length of the observed time. Kaplan - Meier curve represents this function. At a certain time, interval Kaplan-Meier curve shows what the probability of an event is. If the sample size is large enough, the curve should approach the true survival function for the population under investigation.

Let $T_1 < T_2 < ... < T_k$ be a set of distinct ordered event times observed for $N(K \leq N)$ instances. In addition to these event times there are also censoring times for instances. For a specific event time $T_j (j = 1,2,....,K)$, the number of observed events is $d_j \geq 1$and r_j instances will be considered to be “at risk” since their event time or censored time is greater

than or equal to T_j. The conditional probability of surviving beyond time $T_j$can be defined as:

$$\mathrm{p}(T_j) = \frac{r_j - d_j}{r_j} \qquad \textit{(14)}$$

Based on this conditional probability the product- limit estimate of survival function

S(t) = P(T≥ t)is given as follows:

$$\hat{S}(\mathrm{t}) = \prod\nolimits_{j:T_j<t} p(T_J) = \prod\nolimits_{J:T_j<t} (1 - \frac{d_j}{r_j}) \qquad \textit{(15)}$$

Other common non parametric techniques are Life Table analysis (Cutler and Ederer 1958) and Nelson-Aalen (Nelson 1972; Aalen 1978). The first is more commonly used when we have to deal with large sample of data or when data are grouped into some interval periods. The second of is a method to estimate the cumulative hazard function for censored data based on counting process approach.

ANNEX 2

Survival analysis using machine learning methods

There have been a lot of scientific studies that use machine learning methods in order to apply survival analysis in a set of data. Statistical methods and machine learning approaches have the same purpose (share the common goal) to make predictions of the time the event of interest will occur (Wang 2019).

The difference is that statistical methods focus more on both the distribution of the event times and statistical properties of the parameter estimation in contrast to machine learning methods that are usually used for high - dimensional problems. The main goal of applying machine learning methods for survival analysis is that efficient machine learning and deep learning algorithms can learn the dependencies between covariates and survival times in different ways.

Machine learning is effective when there are a large number of instances in a reasonable dimensional feature space and there have been used algorithms that have the ability to discover non linear dependencies between the covariates (Wang 2019).

Survival Trees

Decision trees are very common in the field of machine learning. They are used both in classification and regression problems. Ciampi (Ciampi 1981) in his study made the first attempt to build a decision tree that could handle censored data. This kind of trees are called Survival trees. One of the differences between a survival tree and a standard decision tree is the splitting criterion. Standard decision trees perform recursive partitioning on the data by setting a threshold for each feature. The splitting criteria for survival trees can be grouped into two categories (i) maximizing between node heterogeneity and (ii) minimizing within node homogeneity.

As applicable to standard decision trees a major pursuit is the selection of the final tree and procedures such as backward selection or forward selection can be followed for choosing the optimal tree (Bou-Hamad et al. 2011; Wang 2019)

Bayesian Methods

In statistical theory, Bayesian theorem is very well known. Bayes theorem describes the probability of an event, based on prior knowledge of conditions which may be associated to the event. In (Lisboa et al. 2003), the authors proposed a Bayesian neural network framework to perform model selection for longitudinal data using automatic relevance determination.

Naive Bayes and Bayesian Network algorithms are based on Bayes theorem and recently, the authors in (Fard et al. 2016) effectively integrate Bayesian methods with an AFT model by extrapolating the prior event probability to implement early-stage prediction on survival data for the future time points.

One drawback of approaches based on Bayes theorem is the independence assumption between all the features. There is a big chance this may be not true for many real world scenarios.

Artificial Neural Networks

A fundamental category of algorithms in computer science and especially in the field of machine learning are Neural Networks. Neural Networks are inspired by biological neural systems, and they simulate the way biological neurons in our brain system work.

An ANN is predicated on set n of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, just like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and may signal neurons connected to it. In ANN implementations, the "signal" at a connection may be a real number, and therefore the output of every neuron is computed by some linear or nonlinear function of the sum of its inputs. This is called activation function.

There have been various approaches of neural network architectures to solve survival analysis problems. A first naive approach was to predict the survival time of a subject directly from the given inputs. Then at 1995 Faraggi and Simon's approach was to extend the Cox PH model to non-linear ANN prediction (Faraggi & Simon 1995). Other studies **(**Lisboa et al. 2003) take the survival status of a subject as the neural network's output. In these studies survival status is considered to be the survival or hazard probability. Also feed-forward neural networks are used to obtain a more flexible non-linear model by considering the censored information in the data using a generalization of both continuous and discrete time models **(**Biganzoli et al. 1998**)**.

State of the art approaches use deep learning architectures for survival analysis. In this category belongs Yao's study (Yao et al. 2017) to efficiently learn the complex interactions of multimodal data using correlational learning. Convolutional neural networks are also used for survival analysis with pathological images (Yao 2016). Survival analysis also have been used in order to provide personalized treatment recommendations, help clinicians make more accurate decisions (Katzman et al. 2016).

Recently in 2018 (Ching 2018) constructed an artificial neural network named Cox-nnet that its outputs from the hidden layer node provide an alternative approach for survival-sensitive dimension reduction. In this study transcriptomics data have been used for accurate survival estimation. Gensheimer at his recent paper (Gensheimer 2019) proposed another approach for predicting probability survival in patients' cohort. His model is made up neural networks that are designed to use discrete time survival and is called Nnet-survival. It is trained with the maximum likelihood method using mini batch stochastic gradient descent (SGD). Main advantages of this model are the usage of gradient descent that improves the time needs of the model so it can be fitted into large datasets and its flexibility to vary hazard rate over time accordingly to the inputs.

Unlike feedforward neural networks recurrent neural networks are widely used in deep learning applications thanks to their special architecture. They are extensively used in tasks like handwriting recognition or speech recognition and recently Giunchiglia (Giunchiglia 2019) in his paper proposed recurrent neural networks for personalized survival outcomes. Her model at each time step, takes as input the features characterizing the patient and the identifier of the time step, creates an embedding, and outputs the value of the survival function in that time step and finally, the values of the survival function are linearly combined to compute the unique risk score.

Support Vector Machines

Support Vector Machines belongs to supervised algorithms that used in the machine learning both for regression and classification tasks. An SVM model is a representation of the examples as points in space, mapped in order that the samples of the separate categories are divided by a transparent gap that is as wide as possible. New examples are then mapped into that very same space and predicted to belong to a category based on the side of the gap on which they fall. So an SVM training algorithm builds a model that assigns new examples to one category or the other.

There have been many studies that use support vector machines in survival analysis problems. Some proposed studies have been implement support vector machine approaches but with the disadvantage that they ignored the censored instances of the dataset. Such approaches are (Shivaswamy et al. 2007). Others tried to design a predictive model which handles independently right censored survival data and others like Khan and Zubek (Khan and Zubek 2008) tried to build a support vector regression model by building an asymmetric loss function.

In 2017 Relevance Vector Machine (Kiaee 2017) was used for survival analysis and is based on Weibull AFT model that enables the use of kernel framework that has the ability to learn automatically the possible nonlinear effects of the input explanatory variables on target (survival times).

Ensemble learning

As it is known ensemble learning is an advanced machine learning technique that consists of a set of classifiers that each classifier make prediction for the class label then all predictions from all classifiers that are used are collected and the result is coming from a weighted vote among all the prediction results. This technique was developed in order to overcome instability of a single method. Such approaches have also adapted to survival analysis. Larsen (Larsen 2019) in his research introduced an algorithm named Grand Forest, an ensemble learning method that extends random forests and integrates experimental data with molecular interaction networks to discover relevant endophenotypes and their defining gene modules. The latest has two main purposes, the first is to discover modules associated with the outcome of interest and second to identify patient subgroups.

Bagging Survival Trees

One of the most widely used ensemble technique is bagging. Bagging technique is generating multiple versions of a predictor and using these to get an aggregated predictor. It reduces the variance of the base models that are used. Many studies in literature have proposed applications of bagging with survival trees. Breiman at 1996 proposed bagging predictors to achieve better performance of one single classifier and then performed ensemble based algorithm. Another example of bagging technique for survival analysis was introduced in 2003 by Hothorn using survival trees is (Hothorn 2003). Instead of many predictors he used a survival tree to improve the survival time predictions. Also, Ishwaran (Ishwaran et al. 2008) propose a forest of relative risk trees using the tree–building method introduced in Leblanc and Crowley (Crowley et al. 1992), a model which assumes proportional hazards. In this study the relative risk it is computed for any covariate x and finally an ensemble relative risk estimation it is computed.

Random Survival Forest

The basic idea of Random Forest in machine learning was to make predictions using tree structured models without using all different attributes in each node but a random subset instead. The selection of attributes is based on splitting criterion. Hothorn (Hothorn et al. 2005) build a survival random forest tree using the log–survival time. In this framework in which the estimated inverse probability of censoring weights is used as sampling weights to draw each bootstrap sample and a tree is built for each of them. The ensemble prediction of the mean log–survival time is then obtained as a weighted average, over all trees, of those predictions.

In 2008 (Ishwaran et al. 2008) was implemented an extension of Breiman’s Random Forest by using the already known random forest technique for survival data.In addition he enriched his approach by implementing also an algorithm for missing value imputation. Furthermore (Ishwaran et al. 2011) in his study proposed an effective way to apply RSF for high-dimensional survival analysis problems by regularizing forests.

Variable importance (VIMP) was introduced by Breiman (Breiman 2001) in parallel with the implementation of random forest algorithm. As it is described in the paper (VIMP) importance measure increase when noise is added to a covariate. This approach was extended for survival data by (Ishwaran et al. 2011) and it is based on the concept of minimal depth of a maximal subtree. It assesses the importance of a single variable by measuring how deep in a tree the first split based on it occurs.

Kȩretowska (Kȩretowska 2006) studied forests of dipolar survival trees. As in Hothorn (Hothorn et al. 2004), the final Kaplan–Meier estimate is computed by using the set of aggregated observations from all individual trees.

(Eckel et al. 2008) combines the predictor from Cox model with a tree predictor and finally the predictions are estimated from aggregated trees.

Boosting Technique

Boosting in machine learning technique of combining many weak learners in order to build a strong one. A learner is either a classifier or a regression algorithm. A learner that is slightly correlated with actual labels/values is denoted as weak and strong classifier is denoted a classifier that is strongly correlated with the actual labels/values. Hothorn (Hothorn et al. 2006) proposed an extended version of ensemble algorithm based on boosting technique for right-censored data. Also, Ridgeway (Ridgeway 1999) and Benner (Benner 2002) proposed a boosting algorithm with different base learners. Yifei Chen (Chen 2013) proposed a nonparametric model for survival analysis that utilizes an ensemble of regression trees to work out how the hazard function varies in keeping with the associated covariates. His ensemble model is trained using a gradient boosting method to optimize a smoothed approximation of the concordance index. Wojciech Jarmulski (Jarmulski 2019) in his paper present a new modeling approach for survival analysis based on gradient boosting. He used bagged trees as base learners.The resulting models consist of additive components of single variable models and their pairwise interactions. Hongzhe Li (Li 2005) developed a boosting procedure using smoothing splines for estimating the general proportional hazards models utilizing high dimensional genomic data. Finally, Andreas Mayr (Mayr 2016) tried to build a model for prognosis prediction. He constructed a model utilizing gene signature scores for time-to-event outcomes. He suggests a combined approach to automatically select and fit sparse discrimination models for potentially high-dimensional survival data based on boosting a smooth version of the concordance index (C-index). The gradient boosting algorithm is combined with the steadiness selection approach to reinforce and control its variable selection properties.

Active learning

Active learning is a sophisticated case of machine learning during which a learning algorithm is in a position to interactively query the user so as to induce more accurate predictions in new data points. Since Survival Analysis has many real-world applications, where expert’s opinion is advantageous the idea to adapt active learning techniques for survival analysis is powerful.

Active learning algorithms allows the survival model to select a subset of subjects by learning from a limited set of examples first and then query the expert to get the label of survival status before considering it within the training set. Then the algorithm gets feedback from the expert and by this interaction improves itself. (Vinzamuri et al. 2014) in his study proposed an integration of regularized Cox model with active learning technique. His model uses a discriminative gradient based sampling scheme. He Developed a unified ARC framework which encapsulates three regularized Cox regression algorithms which include the kernel elastic net Cox, elastic net Cox and LASSO-COX regression algorithms.

Transfer learning for survival analysis

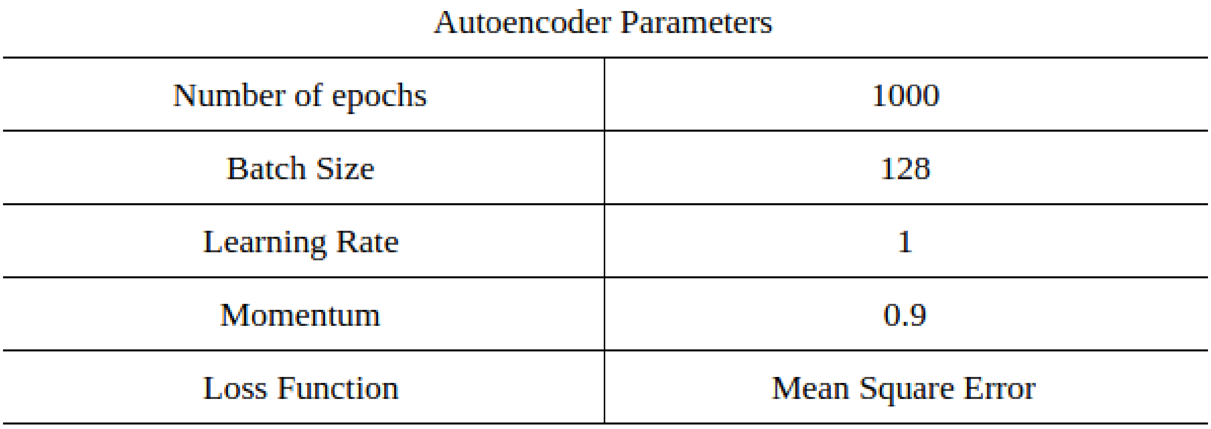
Transfer learning is the technique that focuses on storing knowledge gained while solving one problem and applying it to a distinct but related problem so as to create more accurate predictors/models. This technique has been widely used to solve problems such as regression and classification. In study (Li et al. 2016) uses the $l_{2,1}$-norm penalty to encourage multiple predictors to share similar sparsity patterns. He used $l_{2,1}$ -norm to penalize the sum of the loss functions.

Multi-task learning for survival analysis

Multi-task learning is the subfield of machine learning that aims to solve multiple different tasks at the same time, by taking advantage of the similarities between different tasks. (Li et al. 2016) used the$l_{2,1}$-norm penalty to learn a shared representation across related tasks and hence select most significant features and reduce overfitting. In this study Li propose an indicator matrix to enable the multi-task learning algorithm to handle censored instances and incorporate a number of the important characteristics of survival problems.

Below graph is a presentation of all machine learning algorithms that have been implemented for survival analysis.

ANNEX 3



Autoencoder Parameters

Number of epochs	1000
Batch Size	128
Learning Rate	1
Momentum	0.9
Loss Function	Mean Square Error

The above figure shows the parameters chosen for autoencoder neural network and the below figure shows information related to custom clustering layer (weights, kernel, loss function, auxiliary target distribution definition and further notations).

Clustering Layer Optimization		
Weights	K-means centroids	
Kernel	t-distribution	$q_{ij} = \frac{(1+\lVert z_i - \mu_j \rVert^2/a)^{-\frac{a+1}{2}}}{\sum_j (1+\lVert z_i - \mu_j \rVert^2/a)^{-\frac{a+1}{2}}}$
Loss Function	KL divergence min	$L = KL(P \Vert Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$
Target distribution		$p_{ij} = \frac{q_{ij}{}^2 / f_i}{\sum_{j'} q_{ij'}{}^2 / f_{j'}}$
Notations		
$f_j = \sum_i q_{ij}$		
q_{ij} = the probability of assigning sample *i* to cluster *j*		
$z_i = f_\theta(x_i) \in Z$		
a = degrees of freedom		
μ_i = centroids		