Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Bayesian Networks for censored data: A Survival Analysis

Anuradha Ramsajiwan Vishwakarma
Supervisor: Dr. Bahman Honari

August 19, 2022

A dissertation submitted in partial fulfilment
of the requirements for the degree of
MSc (Computer Science - Data science)

Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Signed: Anuradha Ramsajiwan Vishwakarma
Date: 19/08/2022

Permission to Lend and/or Copy

I, the undersigned, agree that the Trinity College Dublin Library may lend or copy this thesis upon request.

Signed: Anuradha Ramsajiwan Vishwakarma
Date: 19/08/2022

Acknowledgements

Firstly I would like to thank my supervisor Dr. Bahman Honari for his guidance throughout the journey. Without this guidance and encouragement, it would not have been easy to finish the thesis in a well-planned and accurate manner.

I would also like to thank my family and friends who always supported and believed in me. My father always believed in me and motivated me to complete all my dreams. I will always be very thankful to them for their continuous support.

I am very glad to be part of such a renowned institution, Trinity College Dublin. I always dreamed of learning with world-class professors, and Trinity allowed me to fulfill my dreams.

Abstract

In the healthcare system, consider having a system that could provide information regarding the survival time for a particular patient. Such as, in the case of cancer, if we have the risk prediction of a particular patient surviving for X days, suitable measured can be taken into account to improve the patient's health. This information can be obtained using Survival analysis. However, we need a complete dataset to have accurate results to perform the survival analysis. Unfortunately, in the healthcare system, obtaining the result for each patient undergoing study is very difficult because individuals tend to leave the study for many reasons. Such individuals for whom the exact outcome for a disease is not available are called censored data, accurately right censored data. Hence one of the biggest challenges while obtaining the survival analysis is dealing with censored data.

In this research, we will explore the methods to deal with censored data. Furthermore, obtain their comparative analysis. We will work with IPCW weights and weighted censored instances to check which methods work best to deal with censored data. Along with this, we will also look into the parameter and structure learning for the Bayesian network and see which Bayesian network algorithm is better when modeling such data.

We will present the comparative analysis of both methods to deal with censored data and the Bayesian network algorithms. we will compare the results obtained from each model.

Keywords : Bayesian network, Survival analysis, Inverse probability censoring weight (IPCW), weighted censored instances.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

In the healthcare system, if there can be a way of predicting a patient's health condition, like the probability of a patient surviving a disease, it could be incredibly beneficial. Having such a system in place will help identify the healthcare workers if they need to take special care of the patient in any specific time range. The information about the risk probability and precise classification will help healthcare workers implement better practices to take care of the patient. "Survival Analysis" is a method that allows us to obtain the survival probability of a patient undergoing a condition or a patient after receiving a particular treatment. If a new treatment is launched for any disease using survival analysis, it can be tested whether or not the treatment has any significance on patience.

There are a considerable number of methods to perform survival analysis, and there are many Machine learning algorithms that could be used for the same purpose. However, machine learning algorithms work best on complex nonlinear data, and the conventional ML models are not the best fit for survival analysis. Moreover, the ML model fails to predict time-to-event occurrences [1]. Hence In this project, we are using Bayesian networks to perform the survival analysis. Bayesian networks are best when it comes to dealing with such data. Bayesian networks are very well structured and informative. They handle model complexity well, capture knowledge, and are very flexible. We can even model dependencies among risk factors[2][3].

In this project, we will learn how to perform survival analysis, deal with censored data in the dataset, and implement a Bayesian network for such data with a conditional probability between covariates. We will also analyze a few techniques to deal with censored data, their advantages, and their limitations.

1.2 Problem Statement

The main motive behind this study is to obtain the survival analysis for Lung cancer patients so that we could understand the expected wait time for a particular patient before the event

could happen, in this case, the patient's death.

In the survival analysis, our main motive is to obtain the time-to-event prediction. In healthcare data, there is high variability in the time between the event occurrence depending on each subject. Because of this, there is always not enough information present regarding a few of the subjects. For example, in Clinical data, we are not sure if the event has occurred after the end of the observation period or not. Such data is called right censored data, which states our next motive is to implement the methods to deal with censored data and understand which method works best for the data under consideration.

In this thesis, we are going to demonstrate the two methods to deal with censored data, which are "Inverse probability censoring weights" and "weighted Censored instances". Afterward, we will obtain the Bayesian inference using data from both strategies and discuss which techniques provide better risk prediction.

1.3 Motivation

Survival analysis is a study of evaluating the effect of the covariates on the time until an event occurs. It is mainly used for studies in which the time until an event occurs is of interest to them. In our case, we are trying to obtain the event time in case of lung cancer. In this thesis, we are focusing on lung cancer patients, so our motive is to obtain the time until the death of a person occurs, considering his age, gender, and other factors.

The motivation behind this study is to analyze the different techniques available to deal with censored data, analyze which method works best for our data, and determine the different issues faced when dealing with censored data. The Survival analysis can be used in many different scenarios. Below are a few examples of it.

Hardware failure: Hardware failure is the time before hardware is considered to fail. In this, we can obtain the waiting period, depending on many factors like when it was sold, for what purpose the hardware is being used, and many more. Here as we are interested in obtaining the time to and event hardware failure, the survival analysis can be a good approach in this case.

Human resources: There are many applications of survival analysis in human resources. We can obtain information regarding which policies contribute more to employee satisfaction, eventually resulting in employees being with the Organization for more time. Another application could be to see how much time it will take to fill a particular position, depending on the supply on the market for the skills required for the position. This will help the Organization plan its hiring process and keep extra time in hand to avoid employee shortages.

Issues resolution: Often, we see in customer care that there is always a long queue

in ticket resolution. In many scenarios, because of some deadlock condition or a dependent ticket, some tickets may not be solved, or it could be because the customer care people are waiting for user response, the ticket is not solved. In such cases, survival analysis can be used to obtain the time to event analysis.

Loan repayment: This use case for survival analysis is most interesting because it involves a direct money transaction. The lenders are mainly interested in loan repayments; however, this study has often been modeled as a binary case, such as whether they pay the loan or not. In my opinion, its use full to model the complete repayment data or the portion of the amount paid over the duration. Using survival analysis can help us predict based on the different factors if an individual is most likely to pay back the loan in T duration or not and can make the like easier for the lenders. Having the risk information regarding the loan repayment will help the lender decide whether they should grant a loan to an individual or not.

Product warehouse management: One of the many applications of survival analysis could be obtaining the information that is most likely how much time a product is considered to be sold out. This information will help the warehouse manager's demand and supply chain.

write more

1.4 Thesis Structure

Below is a brief idea about the structure of the thesis.

Chapter 1: Introduction In this chapter, we will go through the introduction of topics, motivation, and objective of the thesis and lastly discuss the chapter overview.

Chapter 2: Literature Review and Background In this chapter, we have a literature review for the topic and a background study on the Bayesian network. We will go through the important aspects of the Bayesian network and discuss parameter and structure learning in the Bayesian network.

Chapter 3: Survival Analysis In this chapter, we will look at the survival analysis and where it can be used, then we have a few plots describing the survival functions obtained using Kaplin-Meier.

Chapter 4: Data Analysis In this chapter, we have explored our data and obtained the key attributed from the data. We have obtained different distribution plots, which help us understand the data distribution.

Chapter 5: Implementation In this chapter, first, we go through the method to deal with censored data, then we discuss the implementation in detail, and then we have the results

obtained. The same sequence is followed for both methods.

Chapter 6: Conclusion and Future Work In this chapter, we have presented our conclusion and the result comparison. Then we discussed future works.

2 Literature Review and Background

2.1 Literature Review

The most popular method used in the healthcare system to analyze medical economics and behavioral studies in survival analysis. Performing a survival analysis is typically done to determine the risk analysis. The survival analysis is most suitable in longitudinal investigations where the death or the event of interest is reported when it occurs[4]. The Kaplin-Meier technique is the most popular approach for dealing with censored data. However, Murray S. and Tsiatis AA have presented a novel approach that uses prognostic to recover the information lost during censoring in [5]. According to some, this approach is more efficient than the Kaplin-Meier approach. However, according to their research, many scientists employ parametric models because they are more effective.

To handle right-censored, time-to-event prediction, a huge number of methodologies and procedures have been proposed. As the dependent variable is prone to censoring due to events like a hardware failure, death, hospital admission, the appearance of disease, and many more. These data are also known as censored data [6]. There has been previous research on a related method that has suggested how to deal with event statuses that are rightly suppressed and provide no information about the event state. Research has suggested specific actions to eliminate the study or impute the missing data. However, eliminating the data can cause discrepancies in the results and may not provide an accurate analysis. According to certain studies, adapting specific machine learning approaches to deal with censored data is possible, However, they might not be as reliable as the parametric and non-parametric model strategies.

Several statistical techniques can handle survival data, but they struggle to make predictions about future points in time as just a small number of individuals will continue living after the period of observation or research and avoid experiencing failure. It implies that some research participants have undoubtedly lived through the observational period. Thus, we may conclude that two-time classifications are engaged in studying survival. First, the incident or failure occurs within the observation period, and the individual dies or fails within time t. Secondly, the subject remains alive up to a point even before the research's conclusion. It is afterward lost to even further follow-up, leaves the study, or the observational period/study is

over before the individual fails/experiences death. Consequently, this intricate and imprecise observation of the event or failure time t is referred to as censored observation, which is called "Censored data."[7]

Researchers differentiate between data that have been "simple-censored" and "complex-censored." One or more measures may be censored at a single limit-of-detection or multiple limit-of-detection, but they are all at the low end in a simple-censored dataset. On In contrary, a complex-censored dataset includes measurements that have been censored at two or more limit-of-detection and are interspersed with non-censored data[8]. However, usually, in practice, the censored data usually is of a simple-censored type[7]. It is necessary to have a suitable machine learning algorithm that can handle these complexities adequately for reliable predictions of these censored data. Furthermore, information is available for a particular period before censoring occurs in survival analysis for censored data. Therefore, the model should also be fitted with incomplete data to improve outcomes. As a result, in this situation, the traditional semi-supervised ML approaches are not appropriate [7].

Due to the research participants' unwillingness to cooperate, who quit the study before the observation time is up or until they experience the event, such assessments are difficult and complicated. Additionally, they may have seen the incident or passed away, but we are unaware of it since we lost contact with them in the middle of the research. The crucial thing is that these observations shouldn't be disregarded since they may include some or just a little information on survival, which is a crucial aspect. Even though we only have incomplete data on these issues, we are aware that if the event hasn't happened yet, it will eventually happen after the date of the previous follow-up. Because although we presume they lived beyond a certain point in time, we are unable to pinpoint the precise moment of the incident. Another situation that makes the research challenge is the fact that only a small number of people participate in it after a sizable period of time has passed[9]. As a result, we will only observe these people for a limited length of time, and they may or may not pass away or suffer an incident within that time. However, we are unable to eliminate these participants from the research since there is a high likelihood that similar situations would arise in the real world, and doing so would result in a sample size that is too small for us to draw any meaningful conclusions. The Kaplan-Meier estimate is one way to manage such unbiased information. It offers a computing technique for enduring through time despite difficulties with censored data. And another approach is Using the cox proportional hazard model, which makes dealing with censored data easier[9].

2.1.1 Inverse Probability Censoring Weights

Different types of data that may be analyzed are now included in the procedures of survival analysis. [12] Authors proposed the Inverse Probability of Censoring Weighted (IPCW) method

to deal with censored data technique was proposed to deal with right censored data. Although other techniques have been established to analyze data in the situation of informative censoring, the IPCW estimator is the only one that has been created to account for dependent or causal censoring. Authors [10] Discovered that although the Kaplan-Meier estimator is useful for analyzing censored survival data, it produces biased findings when factors are linked with both lifespan and the censoring method. They have examined whether IPCW is appropriate and described how they work to eliminate bias. A family of weighted estimation methods for censored data was presented in [11], and it was shown that these weight estimators are consistent and asymptotically normal for nominal variances. They looked at the effectiveness of the strategy for estimating medical costs, which is difficult since the follow-up data is often insufficient. Their findings demonstrate that the weight estimator approach may also be used to estimate the mean medical expense.

2.1.2 weighting censored instances

When we are dealing with censored data in clinical trials there are many cases in data that need to be taken into account while obtaining the survival analysis. Consider we have cancer patients' data which is of the study period of around 5 years. In this data, we have a few cases where the patients are in the study from starting and their observation period is 6-7 years. As we already know cancer can be treated in 6-7 years so have to take into account that the person is alive because he is treated and he will not be dying because of cancer. Hence in order to deal with his case, a technique of using the weighting censored instances was proposed in [13]. The same technique was also further used by Ivan Štajduhara, and Bojana Dalbelo-Bašićb [14] with Bayesian networks. From their research, they obtained that the method can be used if the data is highly censored otherwise it is better to use the traditional methods to deal with the censored data.

2.1.3 Bayesian networks in Survival Analysis

Bayesian networks are one of the best tools for knowledge representation[14]. They can represent the factors affecting a particular decision using probabilities. Bayesian networks are used in healthcare systems for research for medicine to treatment authentication[32]. [14] build a Bayesian network model using the weighted censored instances technique to deal with censored data. In a Bayesian network, it is easy to manage dependencies. They allow estimation of the probability on the partial data also, so technically, we can build a Bayesian model with censored data also. However, the results will be biased; hence it is better to deal with censored data first and then use them for survival analysis. There are mainly two approaches to building a Bayesian network for risk prediction[33]. In[34] the author created a structure-based Bayesian network model for patients. The bayesian network can also perform the risk assessment for a specific time, like 1 month, 3 months. 180 days, 365 days, and more.

Another approach is to build a dynamic Bayesian network model, which is a bit complex to achieve[35]. [36] implement a dynamic Bayesian network model for risk prediction among patients having mid-size tumors. The dynamic Bayesian model, rather than predicting the risk separately, Provides information on how the patient's health will change over time. Having such model help keep better track of patient's health; however, it is complicated and complex to maintain and requires huge data[36].

Bayesian networks provide many functions to model the clinical data and obtain the risk analysis on the data because so many applications of the Bayesian network is used in clinical research. [38] used the Bayesian network along with the censored instances weights to obtain the survival analysis for patients as the Bayesian network is easy to interpret and less complex to model. It is prevalent for obtaining survival analysis.

In this Thesis, we will use IPCW weights and weighted censored instances to deal will censored data. Using there these two methods, we will obtain the weights for patients and use those weights in order to obtain the best method to deal with censored data. We will also look at different Bayesian network algorithms and obtain which algorithm works better for lung cancer data.

2.2 Bayesian Networks

Bayesian networks can be defined as a type of graphical model. Digital networks are models in which we can perform probability computation using Bayesian interference[15]. Bayesian networks are represented as Directed Acyclic graph (DAG). Such graphs consist of nodes and edges, where nodes are the covariates, and the edges represent the relationship/dependency between them. As the definition of a Directed cyclic graph says, there are no cycles in the Bayesian network. For example, consider three notes A, B, and C. If node A is connected to B and B is connected to C, this connection is only represented in one way, and we can understand that C is a child of B and B is a child of A. Hence, we can conclude that all the notes in the vision network point in One Direction, and if we start from node A it will be impossible to return to node A in the graph.

In the Bayesian network for nodes X = X1, X2....., Xn, the joint probability can be obtained as below[18].

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i|parents(X_i)) \tag{1}$$

Bayesian networks are very significant in the case if we need to obtain the relation between the variables. They are significant in understanding the relationship and can even identify if it does not exist. Bayesian networks can easily calculate complex probabilities, making them

Figure 2.1: Bayesian network graph

easier to use. They can even identify with the help of graphs that, at the time of calculation, the results of what variables were considered and can even force the model to add the variable if it is essential. The Bayesian network representation is easy to read and understand by machines and individuals both. The Bayesian networks are part of the graphical models[16].

2.2.1 Bayesian network as knowledge representation

Probability theory has a way of combining the correlated features, which requires the joint probability distribution of the event, and providing this information costs a lot. Providing the probability distribution makes it very difficult to use the probability theory for knowledge representation[16].

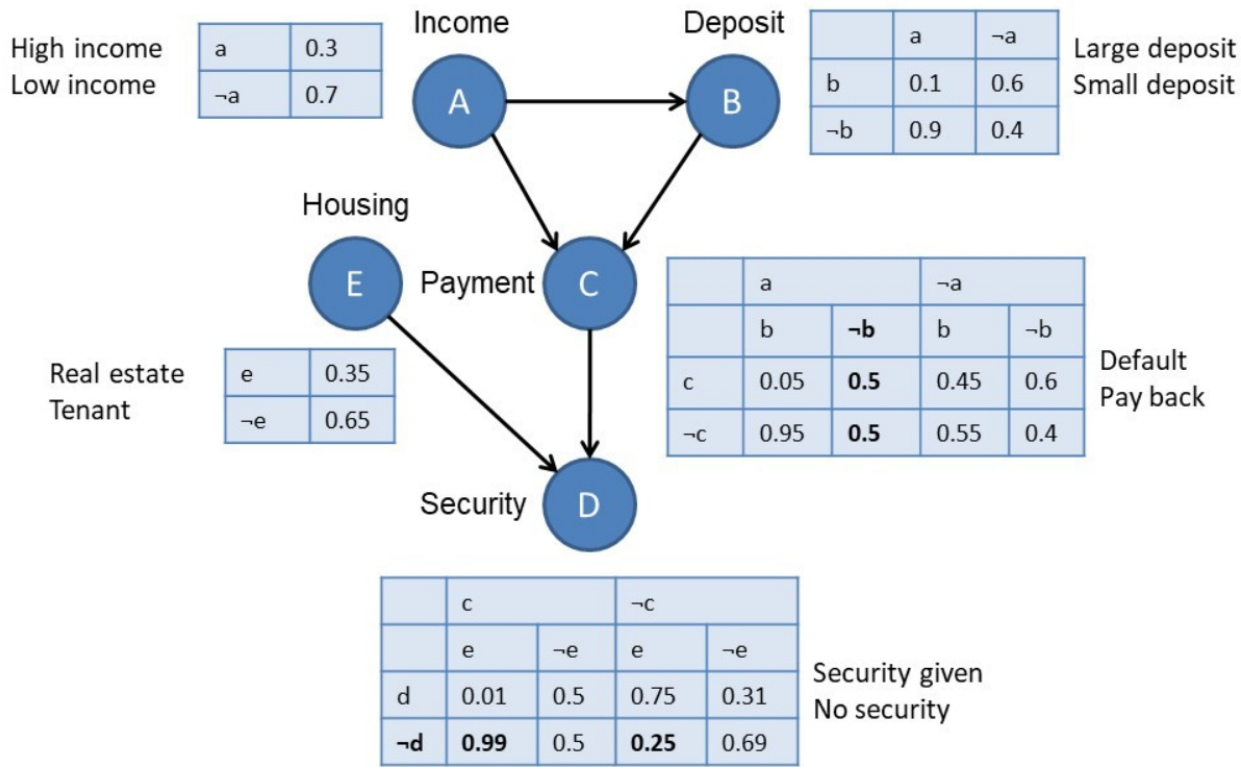

Figure 2.2: [31]Bayesian network representation

Bayesian networks solve the problem in probability theory by maintaining the joint distribution of the event into smaller connected plots as shown in the above figure. As the representation by the Bayesian network is compact and less complex it makes it easier to obtain conditional probability.

2.2.2 Bayesian network as joint probability distribution

For every variable V having parents P1, P2,.....Pn the conditional probability can be obtained such as P(V| P1, P2,.....Pi) this provides quantitative information and hypothesis variable. In practice, it is easier to calculate these conditional probabilities [17]. A Bayesian network consists of both quantitative and qualitative parts. Hence, the DAG representation is called qualitative parts. Moreover, the quantitative parts are the parameters used to obtain the relationship and dependencies between the features present in the structure [16]. In order to represent joint probability, let us first understand a few technical terms. The V is a selected node having parents P = (P1, P2,.....Pn). In a Bayesian network node, V corresponds to a node Vi. And in the Bayesian network, each parent P1, P2,.....Pn is considered parent nodes. Hence, we can say that of a given node Vi its parent node is Pi.

So, for Bayesian network B = (P, θ) and vector V = (V1, ..., Vn), the joint probability distribution can be obtained as[16],

$$P(\mathbf{V}|\mathbf{B}) = \prod_{i=1}^{n} P(V_i|P_i = VP_i, \theta_i) \tag{2}$$

2.2.3 Bayesian networks as generative models

Bayesian networks are also useful when it comes to obtaining the value for data features and generating their coordinate values. In order to have the data of parents at the time of obtaining the child values, it always generates the parent node's value first and then the child node's value. Below is the algorithm proposed in [16] to generate the values.

2.2.4 Bayesian networks parameters and structure[19]

When dealing with Bayesian networks, the most challenging task is to understand the network structure. The important this is to select the best-optimized version of the model. In order to do that, we must first understand how the Bayesian networks work. One of the approaches, in order to find the best Bayesian network, uses constraint-based learning. Constraints are independent conditional statements that are obtained by the data. If the BN structure is dependent on data, one obtains the conditional probability table for the data. From the network, we can make predictions about the target variable.

Algorithm 1: Gendata(B,$topolorder$):

input : Bayesian network $B = (G, \theta)$,
topological ordering $topolorder$ of indices {1 ... n} by G
output: data vector X
n $\leftarrow$ length(G)
$X \leftarrow$ vector of n numbers all -1
for i **in** $topolorder$ **do**
  $j \leftarrow X_{G_i}$
  $X_i \leftarrow$ random_sample_by(θ_{ij})
end
return X

Figure 2.3: [16] Bayesian networks as generative models Algorithm

Another approach is to implement a scoring function that will be responsible for calculating scores for all possible different network structures and then selecting the best structure which will give the best results. This is one of the basic strategies to obtain the best network[19].

There are many other methods for implementing BN learning; however, they are just variations of constraint-based learning and score-based learning.

we will not discuss the BN learning parameters.

Learning Parameters for BN

While learning the Bayesian network, if we consider that we have a dataset D which is complete having structure G, which created data. Even though we have this information, Learning parameters for BN is not very easy.

Maximum Likelihood parameter

One of the most common methods to obtain information about the data generation parameters is obtaining Maximum likelihood. Maximum likelihood estimate is a process of obtaining the parameters which give D a similar probability as any other parameter. In a Bayesian network obtaining Maximum likelihood is relatively easy. Consider for parameter θij we have at least one vector having config j in feature Gi, and then we obtain the maximum likelihood function by the below equation.

Bayesian Learning of Parameter[20]

Bayesian parameter learning states that, if we have distribution is a not known parameter a final set C of the observed dataset, Now consider θ is a random variable having prior distribution p(θ), the modifications in θ which are p(θ|C), can be calculated using prior knowledge p(θ) which is a uniform distribution. Hence p(θ|c) and be known as the posterior probability of θ. Now we need to calculate posterior probability, which is the foundation of

parameter learning.

$$\hat{\theta_{ijk}} = \frac{N_{ijk}}{\sum_{k=1}^{r_i} N_{ijk}} \tag{3}$$

Hence, we can say that p(θ) is Dirichlet distribution[7]. which is given as below.

$$p(\theta) = Dir(\theta|\alpha_1, ..., \alpha_n) = \frac{\gamma(\alpha)}{\prod_{k=1}^{r} \gamma(\alpha_k)} \prod_{k=1}^{r} \theta_k^{\alpha_k^{-1}} \tag{4}$$

here, α = hyper parameter $\gamma(.)$ = gamma function

Hence in Bayesian learning, is it assumed that prior distribution is not a normal distribution. Which is the same as the maximum entropy principle in information theory, which maximizes the entropy of random variables. Hence is there is no information which is used to obtain prior distribution we set is $\alpha^1 = \alpha^1 = ... = \alpha^r$

In the above explanation, the Dirichlet distribution was used as a prior distribution, and now we can explore the Bayesian learning process in the below steps.

1. Non- conditional probability

$$p(\theta|D) = \frac{P(\theta)P(D|\theta)}{P(D)} = \frac{\gamma(\alpha + n)}{\prod_i \gamma(\alpha_i + n_i)} \tag{5}$$

$$\prod_i \theta K^{\alpha_i - n_i} = Dir(\alpha_1 + N_1, ..., \alpha_N + n_N) \tag{6}$$

Therefore,

$$p(\theta_i|D) = \frac{\alpha_i + n_i}{\alpha + N} \tag{7}$$

Here, n^i = count of times the i-th value for x-i was obtaining in D, N = count of time x-I's all possible values were obtained in D

2. Below is the condition probability calculation for nodes having individual parent nodes. Here the assumption is that the parent child relation can be represented as X-> Y where parent and child both have discrete values.

$$p(y|x_i, \theta) = (\alpha_{i1} + n_{i1}, \alpha_{i2} + n_{i2}, ..., \alpha_{ik} + n_{ik}) \tag{8}$$

3. The condition probability calculation for nodes having many parents. Here the first

assumption is parameter independence which says that, if parameters have different distribution, they are independent. Now θ^{ijk} is the condition probability representation for p(a)=j and $X^i = k$. r^i ri specifies the different outcomes for xi. q^i specifies all the possible outcome for their parent nodes.

$$p(\theta_{ij1}, \theta_{ij2}, ..., \theta_{ijl}|c) = c \prod_k \theta_{ijk}^{\alpha_{ijk}-1} \tag{9}$$

$$\theta_{ijk} = \frac{\alpha_{ijk} + n_{ijk}}{\alpha_{ij} + n_{ij}} (\alpha_{ij} = \sum \alpha_{ijk}, n_{ij} = \sum n_{ijk}) \tag{10}$$

Bayesian network Structure learning[21]

As we have discussed how parameter learning works in a Bayesian network, let us discuss how the BN structure learning works. When dealing with data, we can construct the BN structure using a constraint-based search, and another is score based search. Let us discuss them in detail.

Constrain based search

One of the best constraint-based algorithms was proposed in 1993 called the PC algorithm[21]. Constraint-based learning is basically achieved in two steps: a conditional test step and an edge detection phase. The PC algorithm works backward direction. So basically, a unidirectional graph connecting all nodes is created. Once we have the graph, the algorithm selects two nodes and tests if they have any relation or not. If not, the edge is removed, and it keeps entry of the nodes not having a connection. All the edges are mostly checked one or more times. The independence between notes is tested for all possible combinations of note connections.

These tests are performed on each discrete variable to check if they are independent of each other or not. Many diffident methods such as chi-square and degree of freedom are used to determine a node's independence. The assumed null hypothesis states that the two given features are independent at condition S, where S can be null or empty. For this test, we use X^2 and G^2, which are only slightly different from each other. A contingency table is created in order to obtain the value X^2.

$$x^2 = \sum_{i=1,j=1}^{N_{ij}} \frac{c_{ij} - \hat{c_{ij}^2}}{\hat{c}_{ij}} \tag{11}$$

We obtain the value of X^2 to reject or accept the null hypothesis.

Now, as the values of the attribute have been obtained, the next step in the algorithm is

to obtain the edges. To obtain the edges, the algorithm initially searches for V- structure. A V-structure is identified as a structure involving 3 nodes considered A, B, and C, where A and B are connected, and B and C are connected.

On obtaining all the V-structure, Algorithm c checks all the nodes and edges with the below rules. Once a rule is satisfied, the structure is updated.

1. The first rule says if node A and B are connected through an arc and node B and node C has an edge, and there is no edge between node A and node C, node B and node C will be connected through an arc unless creating the arc creates a V-structure.
2. If there is an edge and arc both between two nodes, node A and node B, then the arc is created between node A and node B.

Once all the nodes in the graphical structure are created, the algorithms finalize the structure and return the final structure obtained. This algorithm may not return a finalized BN graph; however, it does provide a partially directed acyclic graph which is also known as a pattern. In case the graph obtained is not complete, a random algorithm is used to complete the graph and obtain a Bayesian network structure.

Score based search

In the score-based method, basically, the algorithm is to define a state space, a bunch of operations for transformation between states, and a score function. Each stare represents a BN network structure. The algorithm has three transition operators. First, Adding and direct arc in between two nodes second is to remove an arc between two nodes last is to invert the arc between nodes.

Finally, the most important thing is the score function. The score function is mostly based on BN. The algorithm defines distributions for all possible BN structures, and to score a structure, the likelihood of the structure given the data is calculated, which is obtained by the below function.

$$P(S|D) = \frac{P(D|S)P(S)}{P(D)} \tag{12}$$

Here,

P(S) = structure prior

P(D|S) = Likelihood for a dataset given a structure

p(D) = Prior distribution of dataset

Below are a few assumptions in order to reduce the cost of calculation to obtain marginal likelihood.

1. The dataset D should be dearest.
2. There is no independence in data which can not be represented by Bayesian network.
3. The dataset should not have any missing values.

Once we obtain the score function, the next step is to obtain the best Bn structure. Practically is not viable to search through the complete set in order to obtain the optimal network structure. The most common approach is to randomly create a structure and start to score the network from there. One of the common algorithms is to use a greedy hill-climbing algorithm to obtain the best network based on score. This process of creating structure and scoring them is repeated until we reach a state that has the highest score or when we reach the local maxima. Usually, the score function mat has too many local maxima, and in this case, the greedy hill climbing algorithm can get stuck. There are a few strategies to break this condition.

1. Random restart: If the algorithm stops at a point and randomly starts from another point and repeats the operation till N times to obtain the structure.
2. Tabu Search: The last N number of states are managed and not allowed to visit again. in case the algorithm stops, the limited number of steps that will deduce the structure are executed. If the algorithm is not able to obtain a structure in the given steps, the last best scoring network is returned.

3 Survival analysis

Survival analysis is a statistical analysis study, also called time-to-event analysis. It takes the dependent covariates as input and tries to obtain the estimated time before the event in interest might occur. English statistician "John Graunt" in 1662, developed the first ever survival analysis by designing the first life table. Initially, survival analyses were only used to obtain the mortality rate. However, in the last few decades, the application of survival analysis has increased in many fields[22].

We will now understand the survival analysis in detail, understand different survival distributions, and go through the factors that impact obtaining survival analysis.

3.1 Censored data

In survival analysis, the event means the event of interest. In our thesis, the event is the patient's death, and the time means the observation time before the event occurred. In survival analysis, the time is not always the same for all the subjects, and it can vary from patient to patient. All subjects did not always enter the study at the start. They can also enter the study anytime during the data collection process. So consider that a subject enters the study sometime between the data collection process, so it is not always that we will have the event observed for each subject. Subjects for which we do not have the event data can be seen as censored data because we do not know the event's status for the subject after the study is completed. Search subjects could experience the event once the study is completed[22]. In survival analysis, missing event data can even be called this censored data; the censored data usually occurs if the study has ended and few subjects have not experienced the event of interest. It could be because of any reason that this study is finished or they are the subject left the study or died because of some other disease. So while performing survival analysis, we cannot drop such data for which we do not have the event because dropping search data would cause the results to be biased. Hence the censored observations are kept in the dataset, but with the label censored, as we can see in our dataset, status 1 in the dataset censored data for lung cancer patients.

As in our data, the event has never been experienced for some patients, and the study has

been completed, so we can say that we are dealing with the **Right Censored data**.

Types of Censored data

3.1.1 Right censored data

Consider the total observation time for the study is T. The right censored data is the data for which we have not experienced the event "death" until Time T. It can be considered that the subjects are still alive. There is no data explaining if the subject will experience the event in the future or not. For example, consider we are running a hardware failure test on 5 components for which we are running to study for T = 1000 seconds. In time T 3 components failed, add 2 are still working, so we record the time for these 2 working components, and they are called "Right censored" as we do not know if they will fail after time T or not.

Figure 3.1: Right Censored data

3.1.2 Left censored data

The left censored data can be said to be the data that are dead before the study. It can also be called interval-censored data for which the first interval is 0. For example, consider the same example as above of the hardware failure data. In the left Censored data, the 2 units failed before the start of the study.

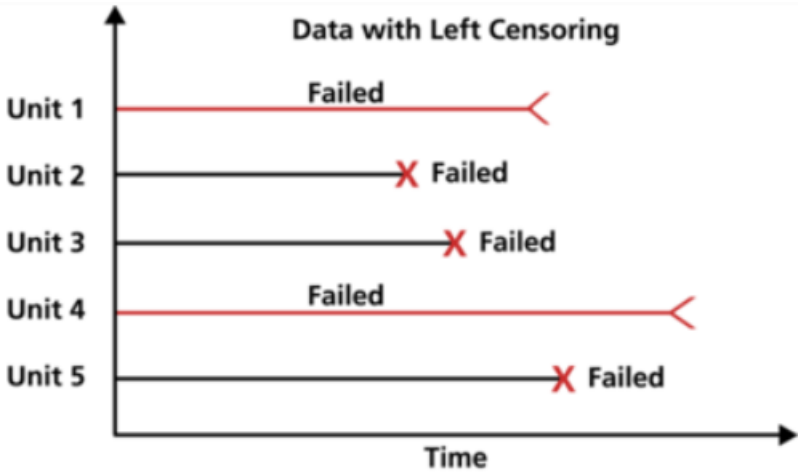

Figure 3.2: Left Censored data

3.1.3 Interval censored data

The interval censoring is when there is no information regarding the actual death time, rather we know about the range of time when the death could have occurred. for example, in case of hardware failure if we make an observation every 100th second and see that the item was not failed at the 300th second but failed at the 400th second. hence there is no information regarding at which exact time the hardware failed.

all other types of censoring can be considered as interval censoring where for right censored data the lower time bound is the end of the study and the upper time bound is unknown/infinity. for left censored data the lower time bound is = 0 and the upper time bound is observation start time.

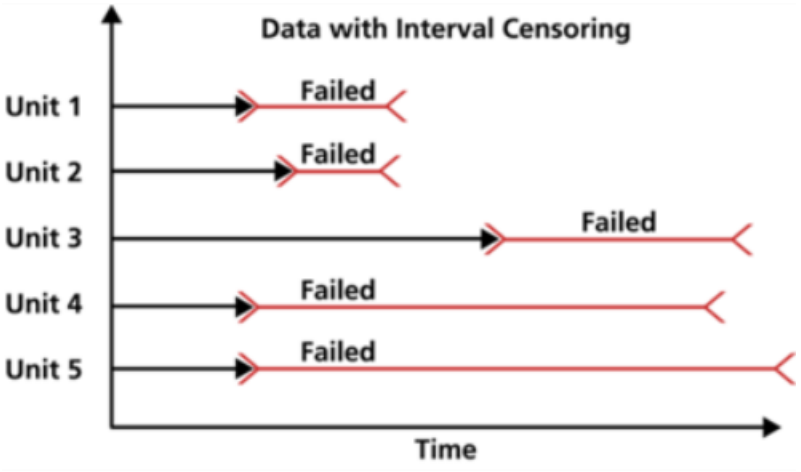

Figure 3.3: Interval Censored data

3.2 Survival function

In order to understand the survival analysis one way could be to understand the math behind it. The survival function can be defined by the survival probability. Consider the T is the time that event has occurred, so survival probability would be that event for interest has not occurred up to time t where T>t. Below is the functional representation of the survival function.

$$\mathrm{S(t) = Pr(T{>}t)\ ,\ 0{<}t{<}\infty} \tag{1}$$

As the survival function value is obtained using the probability of T >t the value for s(t) lies between 0 and 1 it can never be less than 0 or more than 1.

we can also obtain the survival function with respect to the hazard function which is said to be the rate of death. the hazard function is usually denoted by denoted λ or h. Consider a subject has survived till time t and we want to obtain the probability that it will not survive for another instance of time dt. Below is the equation for the hazard function.

$$h(t) = \lim_{dt \to 0} \frac{\Pr(t \leq T < t + dt)}{dt \cdot S(t)} = \frac{f(t)}{S(t)} = -\frac{S'(t)}{S(t)}. \tag{2}$$

let's consider the below example in order to understand the relationship between the survival function and hazard function.

from the below image, we can see in plot a that the hazard rate with respect to time is decreasing quickly and slowly the survival probability in the plot b is also decreasing very quickly on the other hand for plot c the hazard rate is increasing gradually and hence in the plot d, the survival probability is also decreasing slowly[23].

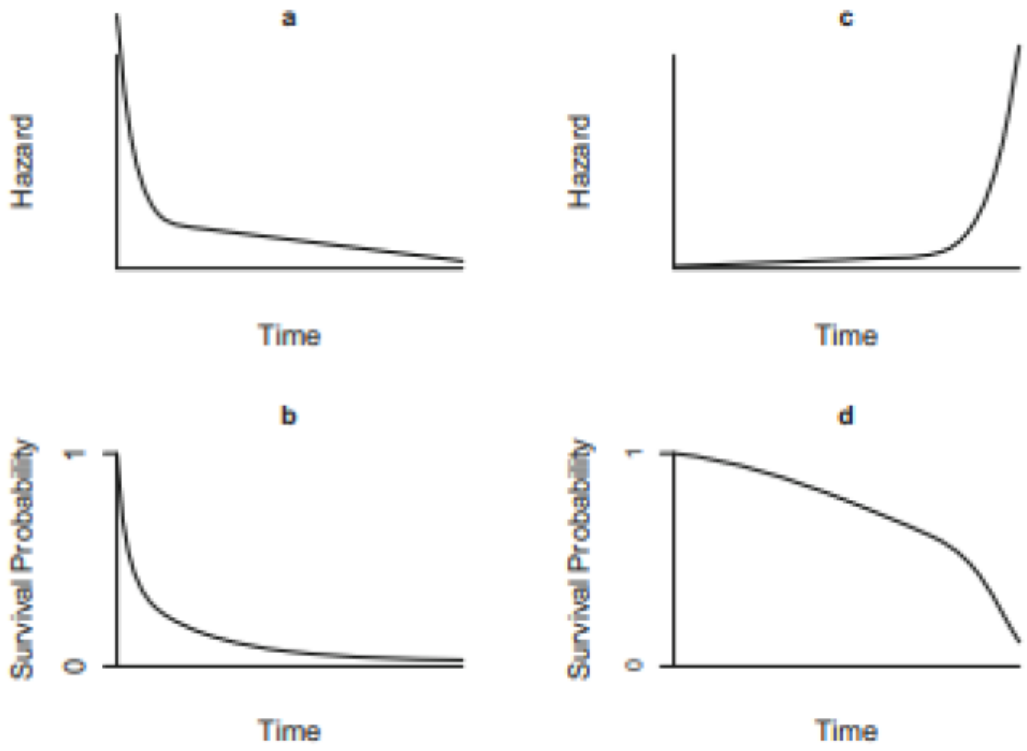

Figure 3.4: Relationship between the hazard and survival function[23]

3.3 Cumulative Hazard function

The cumulative hazard function is usually denoted by H, below is the cumulative hazard function.

$$H(t) = \int_0^t h(u)du \tag{3}$$

3.4 survival plots

The survival plots are obtained using the data discussed in chapter 4. to know more about the data plese have a look at chapter 4.

3.4.1 Survival plots IPCW

The below figure provides the overall survival probability respect to status and time using Keplan Meier. as we can see from ,the figure the survival probability ate the start is at 100% and decreases up to around 5%.

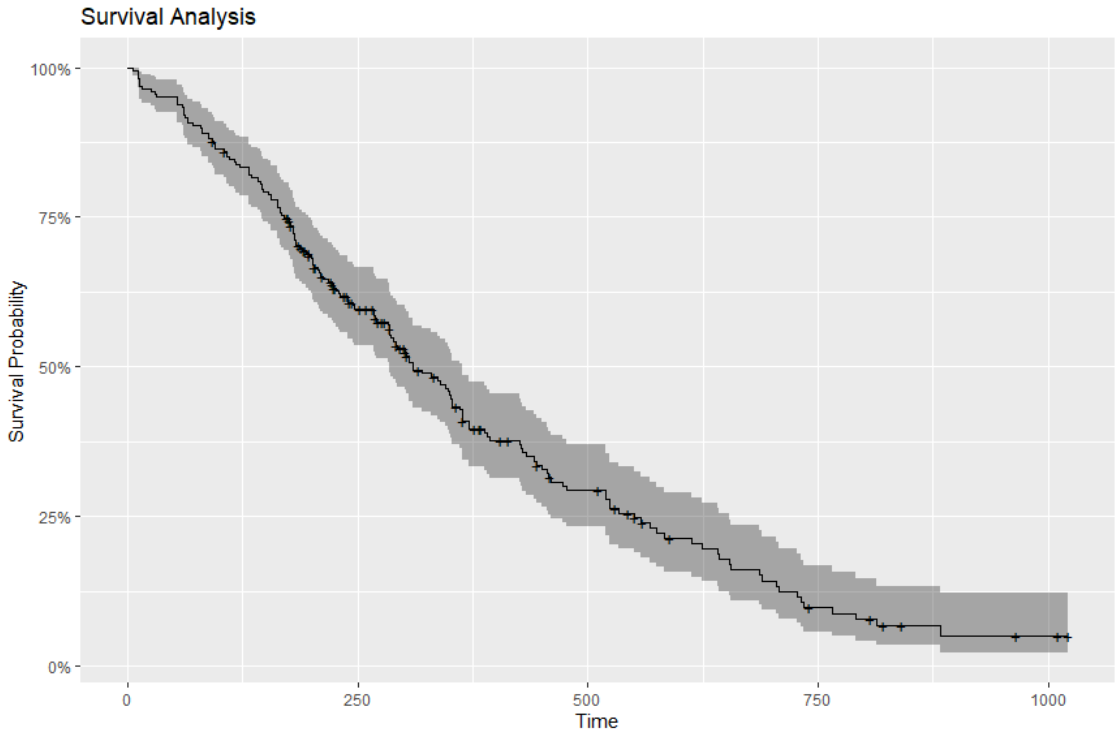

Figure 3.5: survival analysis plot IPCW

From figure 3.6 we can see the impact of age on the results as we can see at a given instance of time the survival probability for the age group over 70 is lower than the age group below 70. Hence we can say that the survival probability is more in the group age lower than 70.

The image 3.7 provides the survival probability for age and with the p-value obtained with the log-rank test. from the figure we can see that the median survival time for the age group over 70 is 270 and for less than 70 is 305. The p-value is, 0.044 which shows that there is little significance.

From figure 3,.8 we can see the impact of gender on the results as we can see at a given instance of ti,me the survival probability for the female is more than the male patients. Hence

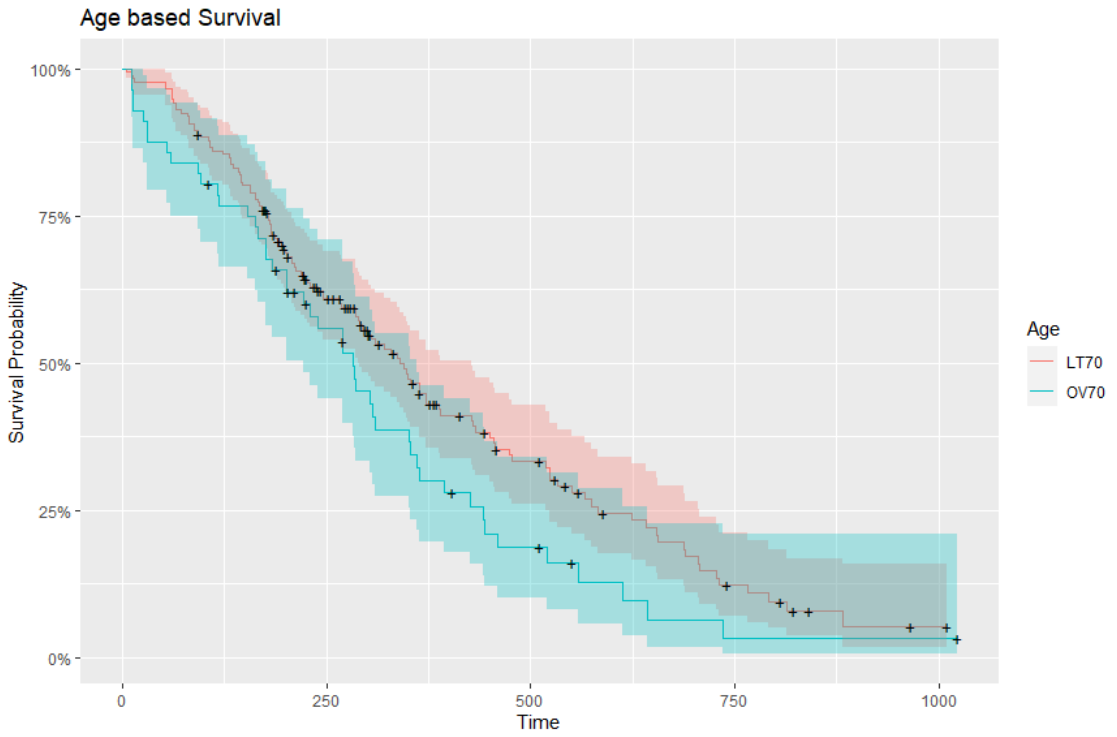

Figure 3.6: survival probability plot for age indicator

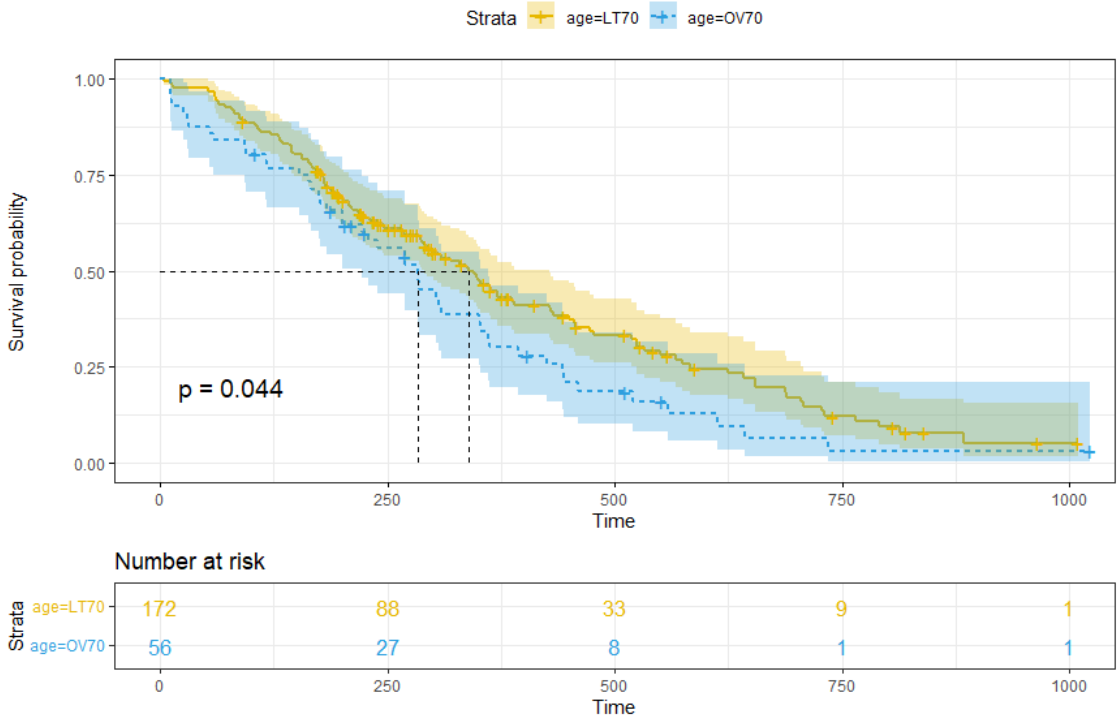

Figure 3.7: survival probability plot for age indicator log-rank test

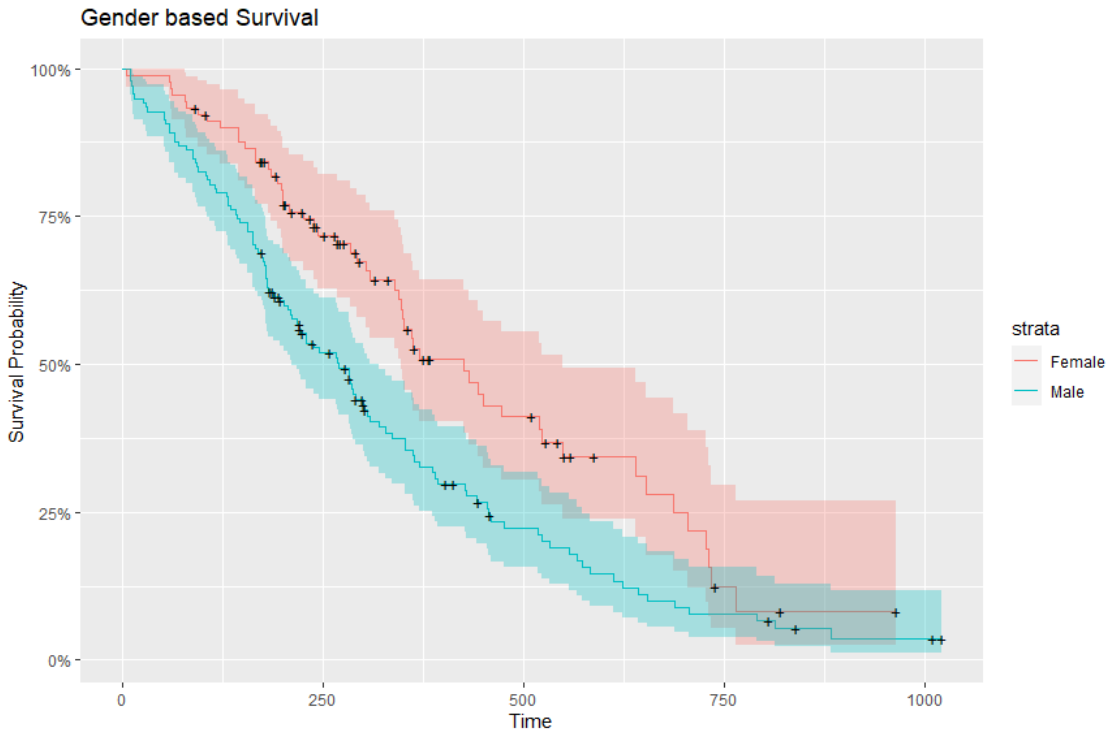

Figure 3.8: survival probability plot for gender indicator

we can say that the survival probability is less for male patients.

Image 3.9 provides the survival probability for gender and the p-value obtained with the log-rank test. from the figure we can that the median survival time for the Male is 265 and for females is 420. The p-value is 0.00,13 which shows that there is little significance.

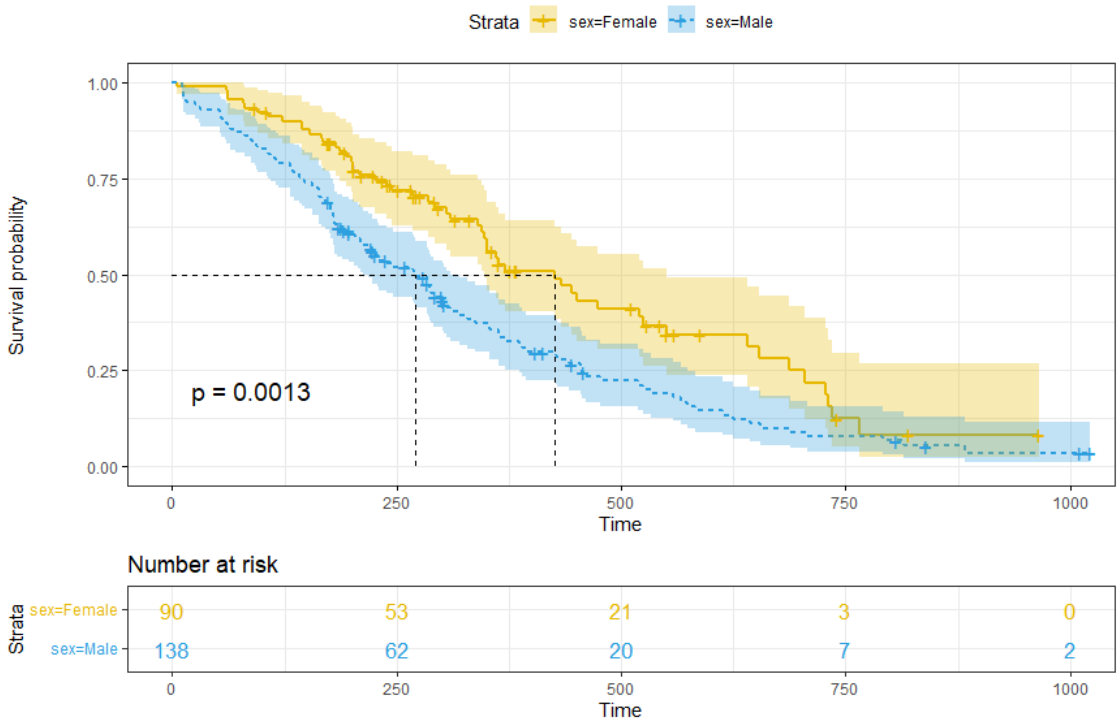

Figure 3.9: survival probability plot for gender indicator Log-rank test

3.4.2 Survival plots COXph

Figure 3.10 show the survival probability obtained with respect to status age gender and ph.ecog using coxph. We can infer from the figure that the survival probability decreases to 12.87%.

we can also obtain hazard ratio of the covariets using the coxph. below images shows the predictors survival probability and the hazard ratio obtained for each of the variables. as we can see from the hazard rato graph the sex and ph.ecog values p value around 0.001 hence we can say that they are significant for hazard ratio.

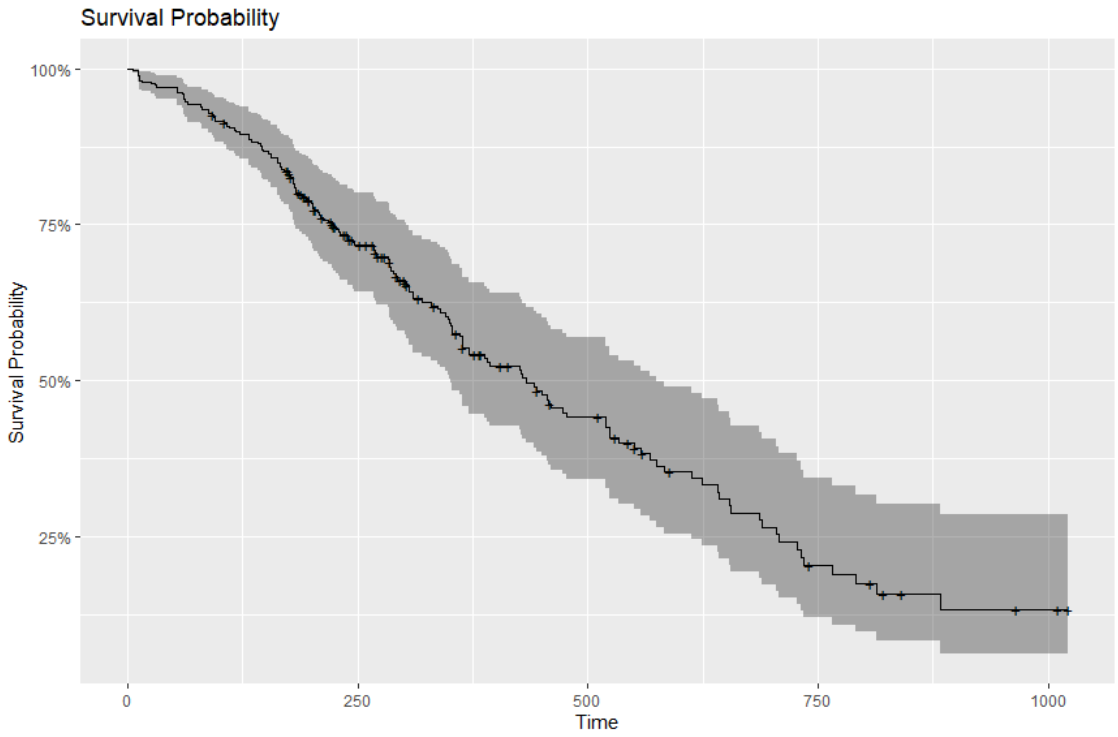

Figure 3.10: survival probability plot COXph

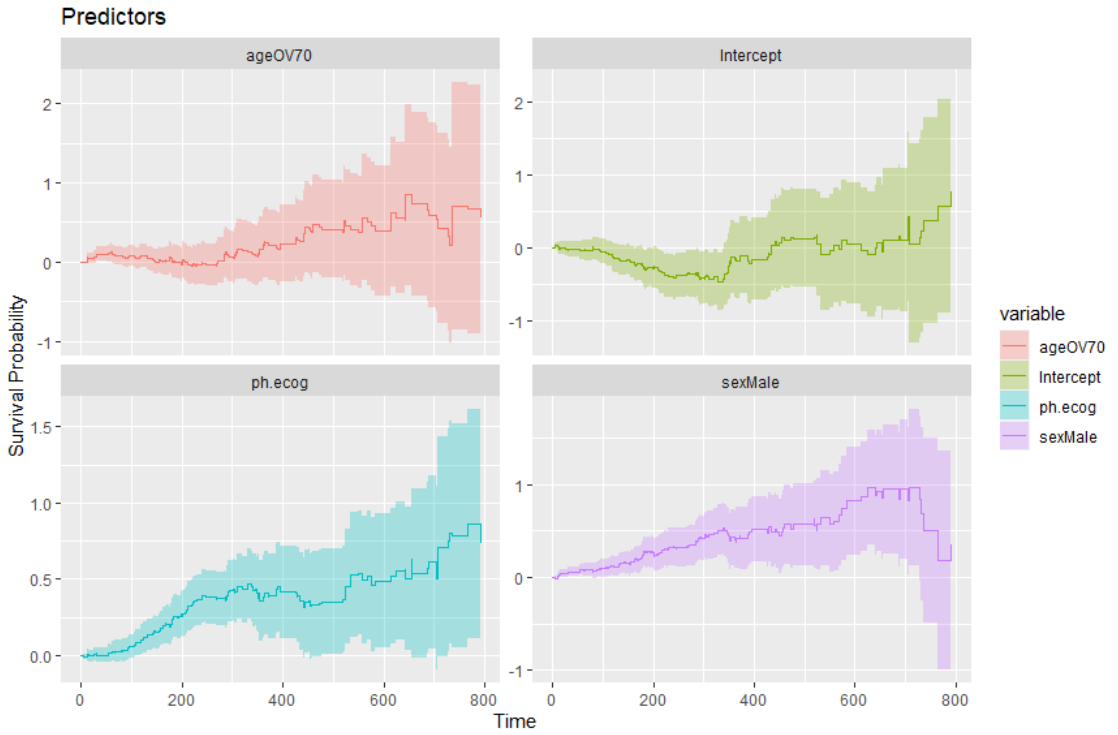

Figure 3.11: Predictors

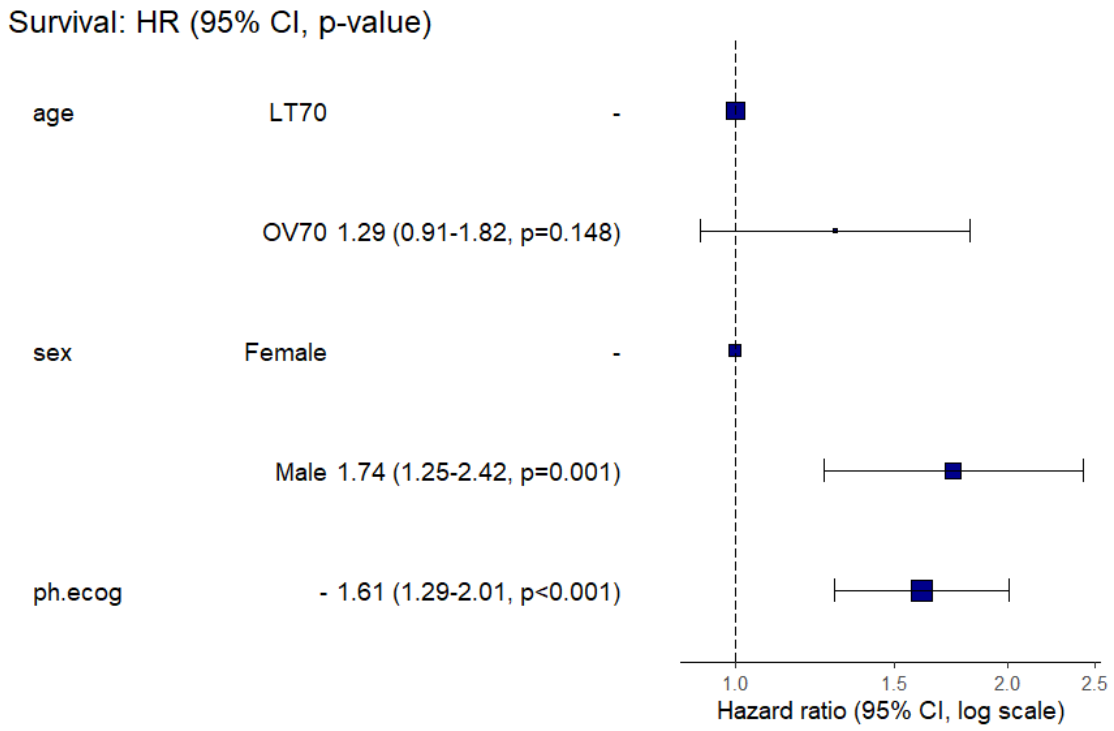

Figure 3.12: Hazard ratio

4 Data Analysis

4.1 Data

The data we choose for this study is "Survival in patients with advanced lung cancer from the North Central Cancer Treatment Group." The data was generated under a study to identify if prognostic information (which does not depend on the data provided by the patient's physician) can be obtained from the user-answered questionnaire[24].

The below tables shows the first ten rows of the dataset, which we are using along with all the column information.

inst	time	status	age	sex	ph.ecog	ph.karno	pat.karno	meal.cal	wt.loss
3	306	2	74	1	1	90	100	1175	NA
3	455	2	68	1	0	90	90	1225	15
3	1010	1	56	1	0	90	90	NA	15
5	210	2	57	1	1	90	60	1150	11
1	883	2	60	1	0	100	90	NA	0
12	1022	1	74	1	1	50	80	513	0
7	310	2	68	2	2	70	60	384	10
11	361	2	71	2	2	60	80	538	1
1	218	2	53	1	1	70	80	825	16
7	166	2	61	1	2	70	70	271	34

Table 4.1: Dataset records: lung Cancer data

we have obtained the from the Kaggle website. Below is the explanation for each column in the data.

inst: Institution code
time: time in days
status: censoring status 1=censored, 2=dead
age: Age in years
sex: Male=1 Female=2
ph.ecog: ECOG performance score as rated by the physician. 0=asymptomatic, 1= symptomatic but completely ambulatory, 2= in bed <50% of the day, 3= in bed > 50% of the

day but not bedbound, 4 = bedbound
ph.karno: Karnofsky performance score (bad=0-good=100) rated by physician
pat.karno: Karnofsky performance score as rated by patient
meal.cal: Calories consumed at meals
wt.loss: Weight loss in last six months (pounds)

In our dataset, the status is our event which is censored here the status 1 indicates that the data is censored, and 2 indicates that the patient is dead. This pattern of using 1/2 was used in the study because of the clinical data requirement.

Note: We have change the status in the dataset to 0 and 1 to standardise the process. 0 being censored and 1 being the event have been experienced.

4.2 Data Analysis

On our primary analysis, when we obtained the data summary in R, we saw some missing values for columns ph.ecog, ph.karno, pat.karno, meal.cal, and wt.loss. One of the approaches to dealing with the missing value is to ignore all the data which has missing values. However, our data sample size is 228 patients, and the max missing value for a column was 47, so ignoring these many sample sizes will cause data loss of around 25%. Hence, we did data imputation for missing values.

4.2.1 Data Imputation: MICE

MICE is called as Multivariate Imputation by chained equation. In this method we aim to impute missing value for a column using existing values of the other column. We basically use values from other column fir a regression model and them impute the value for the missing column[1].

4.2.2 Key features

For survival analysis, we need to understand the relationship between all the features present in the data. Below are the p values obtained for each feature.

	age	sex	ph.ecog	ph.karno	pat.karno	meal.cal	wt.loss
coef	0.018720	-0.5310	0.4759	-0.0164	-0.0198	-0.00012	0.001319
exp(coef)	1.018897	0.5880	1.6095	0.983686	0.980346	0.9998762	1.001320
se(coef)	0.009199	0.1672	0.1134	0.005854	0.005467	0.0002316	0.006079
z	2.035	-3.176	4.198	-2.81	-3.631	-0.535	0.217
Pr(>|z|)	0.0419	0.00149	2.69e-05	0.00496	0.000282	0.593	0.828

Table 4.2: coefficient value for features

When we look at the table first we can see that the P value for "meal.cal" and "wt.loss" is very high and does not show any significance, and hence we can say that these features do not provide any significant information if the subject is going to be alive or not and their coefficient value is also very low. On the other hand, if we look for the features age, sex, ph.ecog pat.krno and ph.krno we can see that their coefficient value are quite good and the p-value is also less than 0.05 which means that these values are significant in order to find out the status of the subject over the year.

4.2.3 Data exploration

In order to understand the features better, we will explore the data here. Below is the time-to-event distribution for our data set.

As we can see from the below figure that the censored data appears after 100 days of observation however the dead cases started appearing from the 0th day. for both the cases, the data distribution is higher at the start and then it gradually decreases.

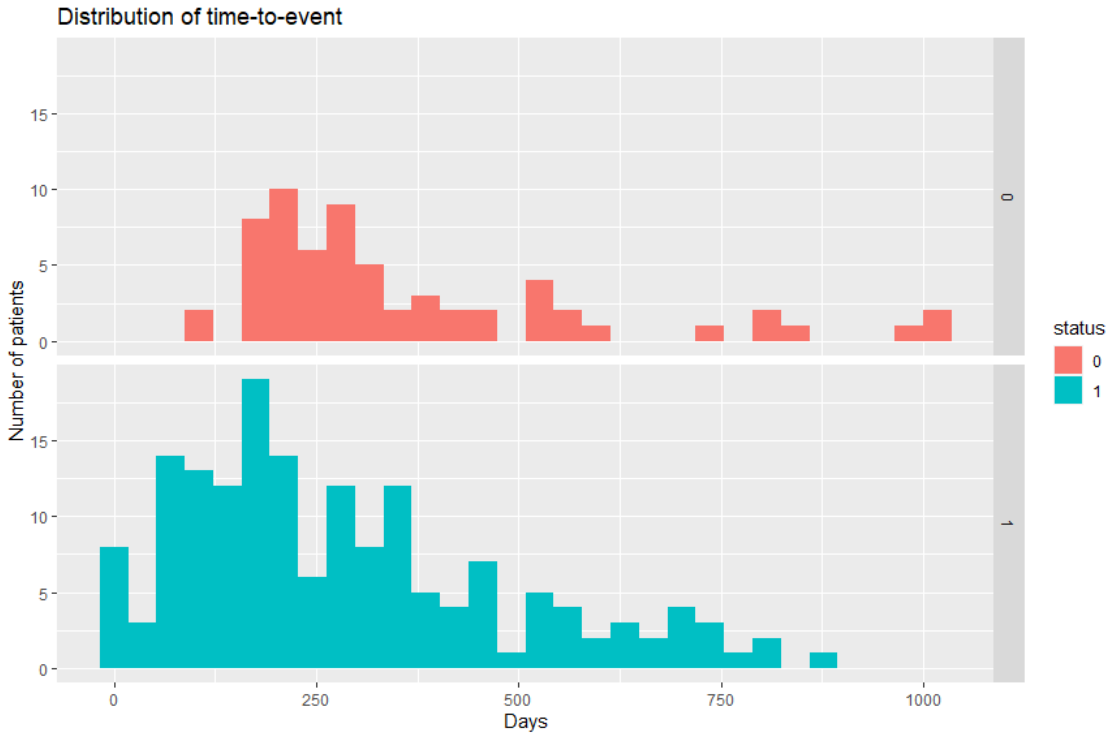

Figure 4.1: Time to event distribution

The below figure is of the age histogram as age is continuous. Here we can see that the patient's age lies between ages 30 to 90, and the maximum number of subjects is from the age range 60-70. For further analysis, we have divided the into two age groups first is the age group over 70, and the second is the age group below 70.

As we can see from the below plot 4.3 the patients in the age group over 70 are less than

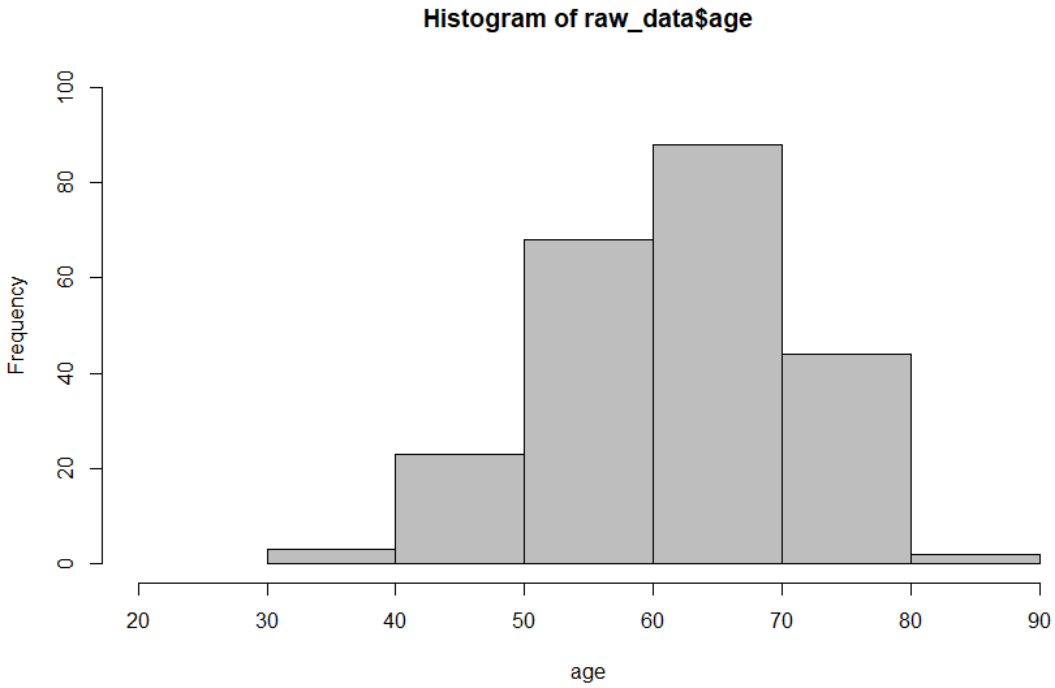

Figure 4.2: Histogram of age

the patients in the less than 70.

For the distribution 4.4, we can obtain that the number of female subjects is more than male subjects. for both events both gender follows the same distribution.

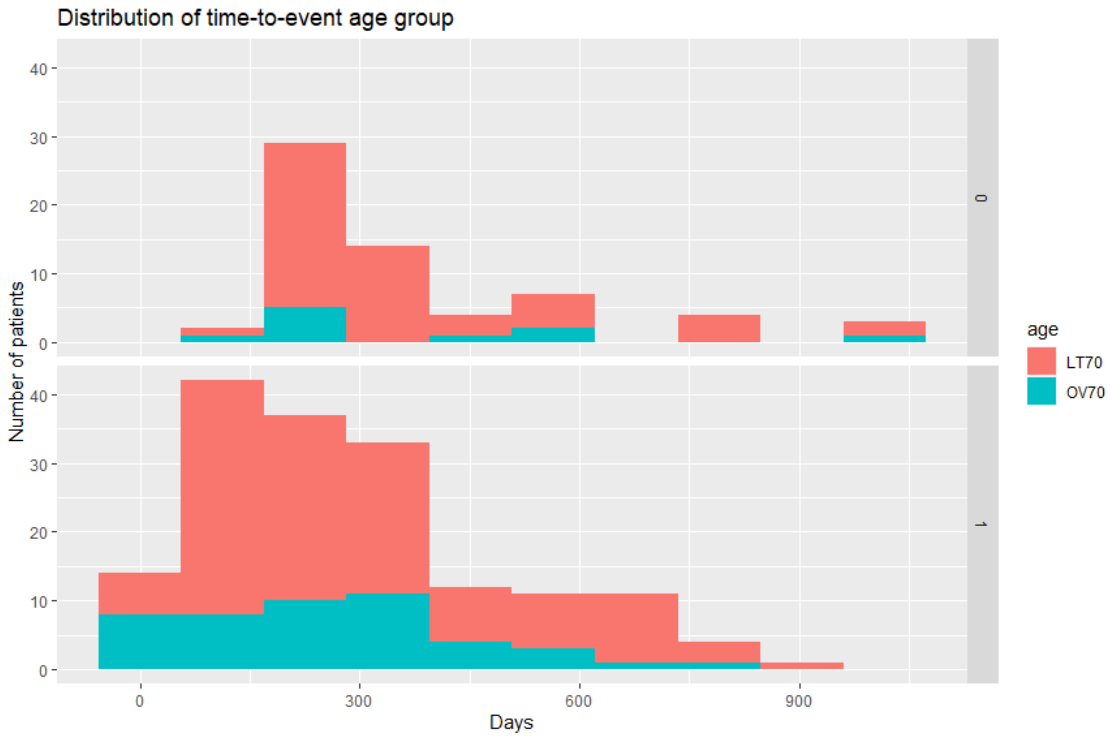

Figure 4.3: time to event distribution with respect to age

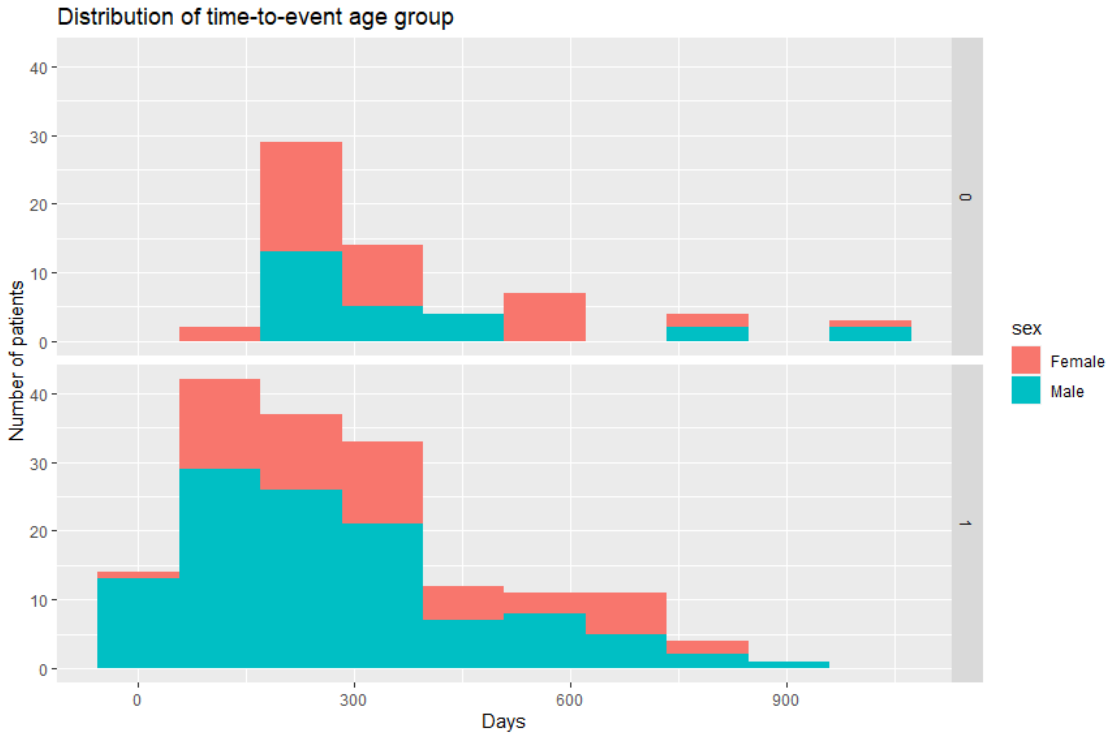

Figure 4.4: time to event distribution with respect to sex

Correlation plot for all features

From the below figure, we can see that the status variable is highly correlated to all the features except meal.cal and wt.loss. the coefficient and correlation show the same characteristic. As we can see that the staus are negatively correlated to time, sex, ph.karno and pat.karno, which means that if these values decrease the chances of survival will increase. whereas there is a positive high correlation between status and age and ph.ecog. We can use this structure when creating the Bayesian network to obtain the dependency.

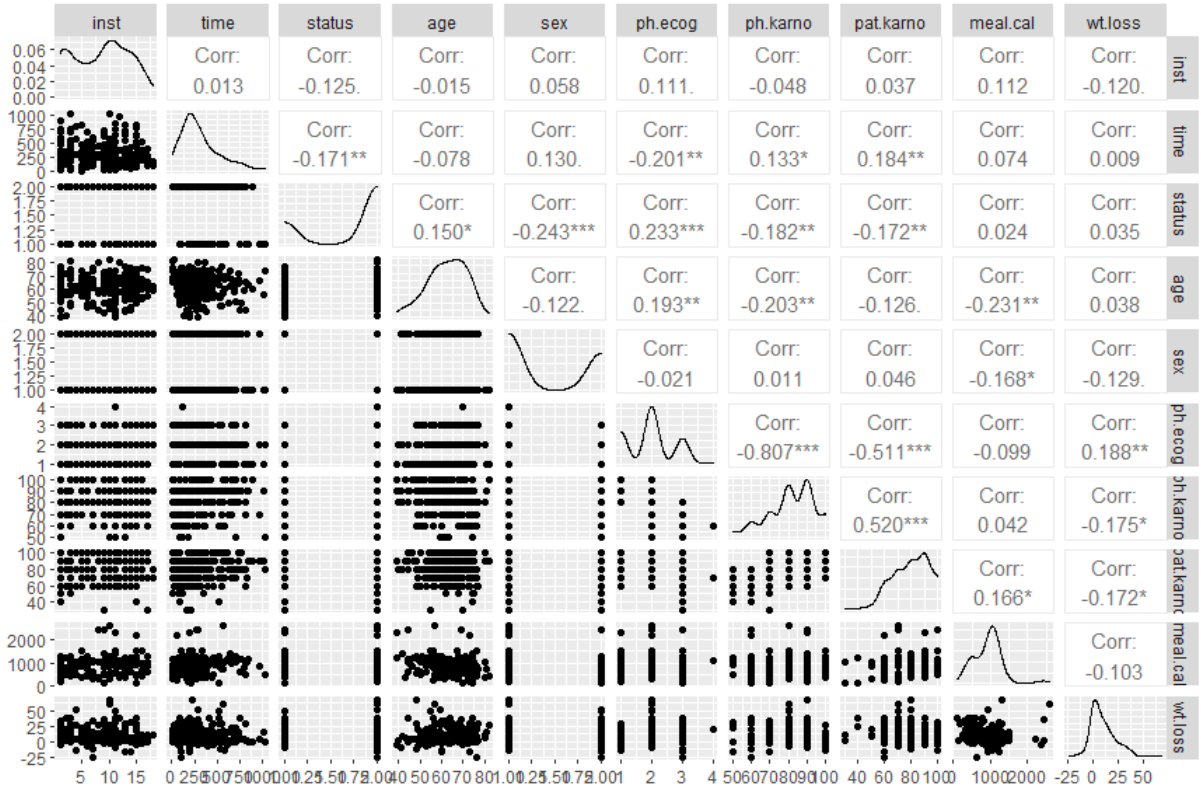

Figure 4.5: Correlation plot for all features in dataset

5 Methodology and Results

5.1 Bayesian network with IPCW weights

Analyses of datasets with censored frequently conducted using inverse probability censoring weighting. A complete case analysis is one way to deal with censored data. Moreover, another approach is excluding the censored data from the observation. If the excluded data vary from the included ones, the findings will be skewed. This bias could be lessened if complete instances are weighted according to their inverse chance of becoming full cases. Hence, for our survival analysis, rather than removing the censored data from the dataset and ending up with a biased model. So, we will be calculating the IPCW weights, and then using those weights, we will fit the model. in this section, we will understand how we calculate the IPCW weights for the right censored data.

When data is not censored, a maximum likelihood estimator is often used to calculate the probability of the classes. For instance, in logistic regression, the coefficients are created using maximum likelihood, and the log odds are parameterized in terms of a linear combination of x. The log odds are coupled to a linear combination of z and an x basis expansion in generalized additive logistic models, and the regression coefficients are computed using penalized maximum likelihood. In Bayesian networks, the continuous components are represented using maximum likelihood using Gaussian densities. In contrast, the discrete components are modeled after the conditional sample average. In order to construct the probability estimates, sample averages from inside nodes are gathered; these averages are non-parametric maximum likelihood estimators. When the neighbors in k-nearest neighbors or the terminal nodes in binary trees are found, this is done. When the sample size reaches infinity, and other regularity conditions are fulfilled, maximum likelihood estimators are routinely used to estimate the population parameters and ensuing risk probability[25].

from weak law of large numbers,

$$\sum_{i=1}^{n} I_i(\beta; E_i, Z_i) \underset{p}{\rightarrow} \epsilon\{I_i(\beta; E_i, Z_i)\} \tag{1}$$

here epsilon E(·) = expectation p$\rightarrow$ = convergence in probability.

For IPCW estimators, we can maximize the likelihood as

$$\sum_{i=1}^{n} \omega_i \; l_i(\beta; E_i, Z_i) \tag{2}$$

where

$$\omega_i \;=\; \Pi[min(T_i, \tau) \;<\; C_i]/\hat{G}min(T_i, \tau) \;\underset{p}{\rightarrow}\; \Pi[min(T_i, \tau) \;<\; C_i]/Gmin(T_i, \tau) \tag{3}$$

hence

$$\epsilon[1/n\sum_{i=1}^{n} \omega_i \; l_i(\beta; X_i, F_i)] = \epsilon[l_i(\beta; X_i, E_i)] \tag{4}$$

We can observe from the aforementioned equations that the IPCW log likelihood converges to the same value as the likelihood for uncensored data for large data sets. Due of the relatively minor differences in both likelihoods for large datasets, IPCW gains all the properties of ML estimators on complete data. According to IPCW, we presume that the censoring time C is independent of either the event's time T or any other variable in data[27]. In our data, the majority of patients are censored because the research came to an end or because they withdrew for one reason or another. Because of this, we make the assumption that the data we are working with in this circumstance complies with our presumptions. another assumption in IPCW is that the data is considerd to be exchangeable and are correctly specified[26]. If in the the dataset we are working with does not follow the assumptions the IPCW will not be able to correct the bias in the data.

So before discussing how we can calculate the IPCW weights, we will first explain the math behind it.

Let us consider that E denotes our event of interest, and it occurs in between τ years/days. Consider that the τ, in this case, is 100o days. So, a person involved in the study leaves the study before τ years/days, say 700 days, and he has not experienced the event at that time. We now do not know if the person will experience the event in the remaining 300 days or not; hence for such individuals, the τ will not be known. Now let us assign Ti as the time between the start and the event time for a person i. Here, Ci is the start time and the time he left the study for any reason. We observe Vi = min(Ti, Ci) and δi= I(Ti<Ci). This is to obtain if the event occurred between the study or not. If δi= 0, then we can say that the data is right censored. Coming back to the previous example, Vi = 700 days and δi= 0, and as we do not know about the event, the patient's information is right censored here[18].

One of the advantages of using IPCW is that this method can be used for an ML model. One can obtain the IPCW weights and then use any ML algorithm to provide better risk

prediction results. Below are the steps followed in obtaining the IPCW weights.

- Using the Kaplin Meier estimator function, the first estimate of the probability that censoring has been observed after time t using function G(t) = P(Ci>t). where G(t) is obtained using the below function.

$$\hat{G}(t) = \prod_{j:V_j<t} \frac{n_j - d_j^*}{n_j} \tag{5}$$

  Where,

  dj = count of patients censored at Vj

  nj = count of patients not censored and not experienced an event at Vj

  Mostly Kaplin mire is used to estimate the event distribution, whereas here, it has been used to estimate the censoring time.

- Now for each patient, we will obtain an IPCW weight using the below equation.

$$\hat{G}(t) \tag{6}$$

  For the subjects, we know the status for them the weight will be Wi=0 . For others, we assign the weight inversely proportional to their calculated probability.

- Now on new data, we can apply any machine learning algorithm of our choice which can give the best results. The model should treat a record having weight 4 as if that record has been seen 4 times in the dataset. Only then will the model be able to predict the correct results without any bias. There are few machine learning algorithms that explicitly ask for weights, and in order to use the IPCW data in such a model, we may need to add a few more steps while modeling so that this weight will be considered differently. One of the approaches to deal with such a problem would be to regenerate the record and add the "shadow" copies of the data in the dataset.

Calculating the IPCW with and example[28]

We state that the IPCW accurately handles the censoring, and helps us obtain the correct results, and handles the bias in data. Here is an example. Consider we obtain that the $\frac{1}{4}$ of the censored observation has the observation time of 500 days, which means G(500) = $\frac{1}{4}$. Now we that the subject "i" subject in our data has the event status as 1(death) at time 500, so δi=1 and Vi= 500. For this patient, we know the event status is 1 the weight is 4. This patient has been assigned weight 4. They represent 4 patients, 3 patients who could be called "shadow" subjects censored before time 500, and itself.

The IPCW weights modify the existing dataset with new weights and increase the number of records. Then we can use this new dataset to train the Bayesian network model.

5.1.1 Methodology Bayesian Network with IPCW weights

Below are the steps we followed to build a Bayesian network using IPCW weights.

1. We Obtained the dataset "Survival in patients with advanced lung cancer from the North Central Cancer Treatment Group." on our first analysis, we obtained that the data has some missing values. Then we performed Multiple Imputation by Chained Equations (MICE) in order to obtain the missing values for the columns.
2. Once we have all the data obtained, we convert them into the numeric format. The status in the dataset corresponds to the event of interest, and it had status 1 for censored data and 2 for dead patients. We converted the status to 0 and 1, o being the censored data and 1 being the dead patients, to standardize the process.
3. In order to calculate the IPCW weights, we first obtained if censored events have after censored before the observation time or censored time. If it was censored after their time we are assigning value 1 for the record and otherwise value 0 is saved in data with the name V.
4. Once we know when the censored event was censored, we then used Kaplin-Meire function in R to fit the model. Then depending on the survival probability obtained for each patient from the model, we assign the weights to records.
5. Now we use these weights in order to obtain the IPCW weights. We inverse the weight from the above step and obtain the IPCW weights for those records having V as 1. So, the observation which has a v value "0" for them, weight is not calculated for them. We store these weights in the table for further use.
6. After obtaining the weights, we will then use them to fit the Bayesian model. Again, we have used python to perform this operation.
7. First, we read the data from the file And shuffle the data to have random samples.Then convert the continuous values into discrete values as the Bayesian network only works for desecrate values. We converted age and time into discrete values.
8. We now select the target and features variable for the model. The target variable is the score, and the feature variable is time, age, gender, and all other features.
9. Now we split our data into training and test data. The training data is of size 75%, and the test data is 25%. Now we use the training data to train the Bayesian network model using the weights calculated.

10. Once the model is the training, we used the test set to test the model prediction and obtain the model accuracy, classification report, and f1 scores. We have also plotted the Bayesian network graph and conditional probability. We will discuss conditional probability and the result in the next section of the report.

5.1.2 Results Bayesian Network with IPCW weights

We built the Bayesian network model in Python. The pomegranate library in Python offers the facility to build Bayesian networks very easily. We have implemented BN using two methods first is exact, and another is greedy search. Below is the structure obtained using the exact algorithm. The surv and V variables are generated while IPCW weight calculation.

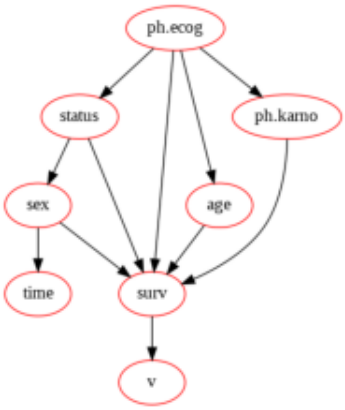

Figure 5.1: Bayesian network plot using IPCW weights and exact algorithm

Model Accuracy The model accuracy is obtained once the model is created and a prediction is created from the test data. Accuracy tells us how many accurate predictions have been made concerning total samples. Accuracy can be calculated using the number of correct predictions divided by the total number of predictions. The Accuracy of the model Bayesian network model with IPCW weight and exact algorithm is **65.78%**.

		Actual values	
		1	0
predicted values	1	12	20
	0	19	63

Table 5.1: Confusing matrix for BN model with IPCW weights exact algoritham

A confusion matrix is one way to measure the classification models' performance. A confusion matrix shows the number of values correctly predicted and for which the prediction was wrong. Here the count of prediction is plotted. The box having actual value 1 and predicted value 1 is called True positive here. The value for class 1 is correctly predicted. In

our case, we have 12 true positives. The next is having an actual value 0 and a predicted value 0, and this is called a false positive here. Values are predicted incorrectly. The actual value should have been 1. However, the model predicted it 0, and we have 20 false positive values. Now the next is called false negative, and here the actual value was supposed to be 1. however model predicted it as 0 in our model. We have 19 false negative values. Finally, the last one is called a True negative, and here the actual and predicted values are 0, which is expected. We have total 63 true negative values.

	precision	recall	f1-score	support
0	0.38	0.39	0.38	31
1	0.77	0.76	0.76	83
accuracy			0.66	114
macro avg	0.57	0.57	0.57	114
weighted avg	0.66	0.66	0.66	114

Table 5.2: Classification matrix for BN model with IPCW weights exact algorithm

The classification matrix is one of the most important evaluation metricx for a model. The precision tells us what proportion of predicted positive value is truly positive. It can is obtained as precision = True positive / (True positive + false positive). we can obtain the precision value for our model in the matrix. Recall tells us what proportion of actual positive are correctly classified. The F1 score is the harmonic mean of precision and recall value. As we can see from the above table, the precision-recall and f1-score values for the weighted average are almost the same as the model performance.

Below is the structure obtained using the greedy search algorithm.

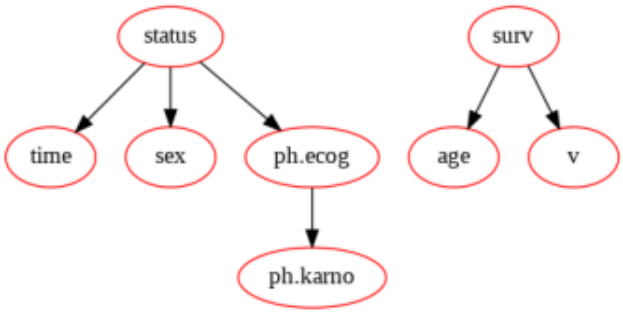

Figure 5.2: Bayesian network plot using IPCW weights and greedy search algorithm

The Accuracy of the model Bayesian network model with IPCW weight and greedy search algorithm is **71.92%**.

		Actual values	
		1	0
predicted values	1	13	15
	0	16	79

Table 5.3: Confusing matrix for BN model with IPCW weights greedy search algorithm

	precision	recall	f1-score	support
0	0.46	0.43	0.45	30
1	0.80	0.82	0.81	84
accuracy			0.72	114
macro avg	0.63	0.63	0.63	114
weighted avg	0.71	0.72	0.72	114

Table 5.4: Classification matrix for BN model with IPCW weights greedy search algorithm

As we can from above evaluation the Bayesian network performs well with greedy search algorithm for IPCW weights.

5.2 Bayesian network with weighting censored instances

One of the most important aspects when dealing with censored data in clinical trials is that we have to consider that if an individual is in a study for a longer time and he has a censored event, then it does not always mean that the individual is going to die because of the disease in the future[13]. The reason is that nowadays, we do have well-advanced treatments, and people do get cured of the disease. And if a patient has been observed for a long time, then there are more chances that he has been cured and may not die because of the given disease. Hence while calculating the weight should take this case into account for censored having a long observation time. To deal with such a scenario, the authors in [13] have provided an alternate method to deal with such data. The method is to divide the data into three groups. First, the people who are in the study for a long time and are censored assign a negative weight (considering they are cured of disease). Second, the people who are dead are assigned then positive weight, and third, last, the censored observation has a lesser observation time. We can calculate the weights for this group using the Kaplin-Meier weights.

So in this method, we will divide the data into three groups

1. Positive - Individuals who have experienced the event
2. Negative - Individual in the study for longer than T* period and not experience the event, T* being the threshold time
3. Last, the people who are in the study for less than the T* period and are censored are assigned both positive and negative weights resulting in the increased data record in the data set and then calculating the Kaplin-Meier weights for them.

Consider that we have six individuals in our study (A, B, C, D, E, F). Now out of these 2 are positive (A, D), one is negative (E), and others are censored before time T*, so according to the split, we will have 9 records to work with (A+, B+, B-, C+, C-, D+, E-, D+, F-) which will ultimately increase the size of our data.

The below function provides the Kaplin-Meier product limit estimate of inherent survival function[29]

$$\hat{S}(t) = \prod_{j:V_j<t} \frac{n_j - d_j}{n_j} = \hat{S}(t-1)(1 - \frac{d_t}{n_t} \tag{7}$$

where

dj = count of events that occurred at tj

nj = count of subjects still at tj

Considering the survival time is not similar to censoring time. [30]

Each doubled record is assigned weight, taking their observation time T into account. For the negative record, we give weight s(T*)/S(T); for positive, the weights are calculated using w+ = 1-w-. If we talk about these to 2 individual censored segments, then we need to consider the higher probability for the one surviving after T* than the one who is censored before the time. T* can we obtain from the field expert. However, for our case, we have considered that the individuals serving after 750 days of observation should have a higher probability of serving as Cancer can be treated in 2 years. Hence in our case, the T* value which we assumed for implementation is 750 days.

5.2.1 Methodology Bayesian Network with weighting censored instances

We have performed the below steps in order to build a Bayesian network model using weighted censored instances.

1. We have used the same dataset as the IPCW weights methods for this implementation. The initial analysis of the data and the imputation method are the same as IPCW.
2. In order to calculate the weights for this model, we have used the R code. First, we defined the T* time as 750 days assuming that the study is conducted for 1000 days and two years is a sufficient amount of time after detecting the canter to treat it.
3. Then we split the data into three sections, as discussed above. We assigned weight 0 to the negative group and weight 1 to the positive group and divided the third group, and created the duplicated records for that group.
4. We have saved the T* 0 and 1 for records before and after the time, respectively. We have also saved the weights 0,1 the duplicated records in a column and called it split. So now, we calculated the IPCW weights for the duplicated third group records. Moreover, we can save these weights in column weights.
5. Now using these updated weights, we can fit the Bayesian network model and obtain the results. To fit the Bayesian network model, we have used the python steps for fitting this model are the same as the above done in IPCW weights.
6. We will discuss the results obtained from the model in the results section.

5.2.2 Results Bayesian Network with weighting censored instances

The Accuracy of the model Bayesian network model with weighted censored instances and exact algorithm is **80.55%**.

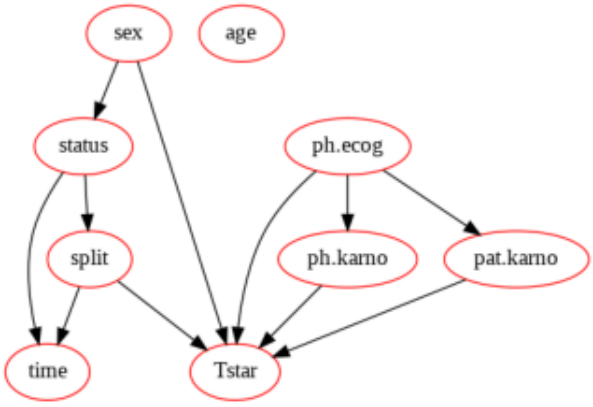

Figure 5.3: Bayesian network plot using weighted censored instances and exact algorithm

		Actual values	
		1	0
predicted values	1	3	19
	0	2	84

Table 5.5: Confusing matrix for BN model with weighted censored instances and exact algorithm

	precision	recall	f1-score	support
0	0.14	0.60	0.22	5
1	0.98	0.82	0.89	103
accuracy			0.81	108
macro avg	0.56	0.71	0.56	108
weighted avg	0.94	0.81	0.86	108

Table 5.6: Classification matrix for BN model with weighted censored instances and exact algorithm

The Accuracy of the model bayesisn network model with weighted censored instances and greedy search algorithm is **87.96%**.

		Actual values	
		1	0
predicted values	1	7	12
	0	1	88

Table 5.7: Confusing matrix for BN model with weighted censored instances and greedy algorithm

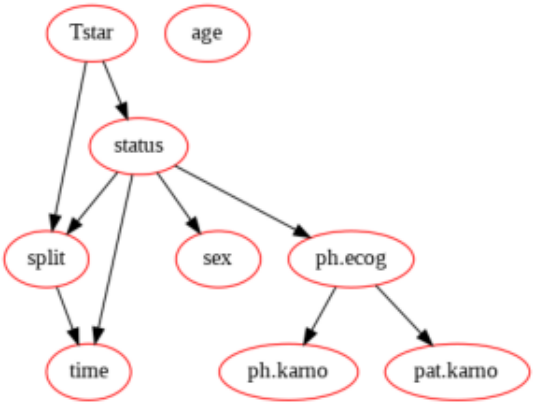

Figure 5.4: Bayesian network plot using weighted censored instances and greedy search algorithm

	precision	recall	f1-score	support
0	0.37	0.88	0.52	8
1	0.99	0.88	0.93	100
accuracy			0.88	108
macro avg	0.68	0.88	0.72	108
weighted avg	0.94	0.88	0.90	108

Table 5.8: Classification matrix for BN model with weighted censored instances and greedy search algorithm

6 Conclusion and Future Works

The table below shows the model accuracy obtained for both the algorithms and both methods to deal with censored data. As we can see from the below table, the accuracy for the model using the weighted censored instances is better than the IPCW weights. Hence we can say that for our data, the weighted censored instances method is better for dealing with censored data. Now when we compare the performance of both algorithms, the greedy search algorithm works best for both weights. The lowest performing model is the Bayesian network using IPCW weights and the exact algorithm having an accuracy 65.78%. And the best performing algorithm is the Bayesian network using weighted censored instances with greedy min max having an accuracy 87.96%.

Model Bayesian network	Algoritham	Accuracy
2*BN IPCW weight	Exact	65.78
	Greedy search	71.92
2*BN weighted censored instances	Exact	80.55
	Greedy search	87.96

Table 6.1: Model Accuracy for all the model with different weight and algorithm

Now we can say that using IPCW weights and weighted censored instances. We successfully deal with censored data. As per our analysis, weighted censored instances methods work best for the given dataset. In order to check the performance of different Bayesian network algorithms, we also used exact and greedy search algorithms. We obtained that the greedy search algorithm works best for our data. We also used imputation methods to deal with the null data so that we could obtain good results.

We have achieved the maximum prediction accuracy for Bayesian networks with weighted censored instances using a greedy search algorithm which is 87.96

6.1 Future Works

Our research is focused on obtaining the survival analysis in the Lung cancer patient. Yet this same research can also be used for many different objectives like testing the effectiveness of a new treatment in the field or in order to obtain using different procedures has what impact on an individual. The research is focused on the healthcare domain. However, this can be used in many domains. We can even use survival analysis to obtain a hardware failure time, given the recent trend in how much time an employee could leave an organization and many more.

In this research, we have discussed two nonparametric methods to deal with censored data. In the future, a few other methods, such as semi-parametric and parametric methods, can be used to deal with censored data, and then we can draw a better analysis of which methods work best for dealing with the right censored data.

As we have used Bayesian networks in this research, another approach would be to work with different models and check if there are any other methods that are best suited to deal with such data.

Bibliography

1. https://pubmed.ncbi.nlm.nih.gov/19833488/

2. Hanna AA, Lucas PJ, Prognostic models in medicine – AI and statistical approaches, Methods of Information in Medicine 40 (2001) 1–5.

3. Husmeier D, Dybowski R, Roberts S, Probabilistic Modeling in Bioinformatics and Medical Informatics, Springer, 2005.

4. https://www.proquest.com/openview/c2228f2c569d315e211f350f9e1fdf37/1?pq-origsite=gscholarcbl=6764

5. Murray S, Tsiatis AA. Nonparametric survival estimation using prognostic longitudinal covariates. Biometrics. 1996 Mar;52(1):137-51. PMID: 8934589.

6. Lee, E.T., Wang, J.: Statistical methods for survival data analysis, vol. 476. John Wiley Sons

7. A Comparison of Several Methods for Analyzing Censored Data. (2007). The Annals of Occupational Hygiene. doi:10.1093/annhyg/mem045.

8. Bullock WH, Ignacio JS. , A strategy for assessing and managing occupational exposures, 20063rd ednFairfax, VAAmerican Industrial Hygiene Association

9. Bewick, V., Cheek, L. and Ball, J. (2004). Statistics review 12: Survival analysis. Critical Care, [online] 8(5), pp.389–394. doi:10.1186/cc2955.

10. Rotnitzky AG, Robins JM (2004) Inverse probability weighted estimation in survival analysis. In: Armitage P, Colton T (eds) The encyclopedia of biostatistics, 2nd edn. Wiley, Hoboken, NJ

11. H Bang, AA Tsiatis, Estimating medical costs with censored data, Biometrika, Volume 87, Issue 2, June 2000, Pages 329–343,

12. Gerds TA, Schumacher M. Consistent estimation of the expected Brier score in general survival models with right-censored event times. Biometrical Journal. 2006;48(6):1029–1040

13. Zupan, B., Demšar, J., Kattan, M.W., Beck, J.Robert. and Bratko, I. (2000). Machine learning for survival analysis: a case study on recurrence of prostate cancer. Artificial Intelligence in Medicine, 20(1), pp.59–75. doi:10.1016/s0933-3657(00)00053-1.

14. Štajduhar, I. and Dalbelo-Bašić, B. (2010). Learning Bayesian networks from survival data using weighting censored instances. Journal of Biomedical Informatics, 43(4), pp.613–622. doi:10.1016/j.jbi.2010.03.005.

15. Stephenson, T.A. ed., (2000). An Introduction to Bayesian Network Theory and Usage. [online] Infoscience. IDIAP. Available at: https://infoscience.epfl.ch/record/82584?ln=en [Accessed 19 Aug. 2022].

16. Silander, T. (n.d.). The Most Probable Bayesian Network and Beyond. [online] Available at: https://core.ac.uk/download/pdf/14916911.pdf [Accessed 19 Aug. 2022].

17. Using Bayesian Networks in reliability evaluation for an (r, s)-out-of-(m, n):F distributed communication system Bahman Honaria,b,, John Donovana, Eamonn Murphyb

18. T. A. Stephenson, "An introduction to bayesian network theory and usage," tech. rep., Idiap, 2000

19. G.F. Cooper, E. Herskovits A Bayesian method for the induction of probabilistic networks from data Mach Learn, 9 (4) (1992), pp. 309-347

20. Ji, Z., Xia, Q. and Meng, G. (2015). A Review of Parameter Learning Methods in Bayesian Network. Lecture Notes in Computer Science, pp.3–12. doi:10.1007/978-3-319-22053-6_1.

21. De Jongh, M. (2003). ALGORITHMS FOR CONSTRAINT-BASED LEARNING OF BAYESIAN NETWORK STRUCTURES WITH LARGE NUMBERS OF VARIABLES. [online] Available at: https://core.ac.uk/download/pdf/20535888.pdf.

humboldt-wi.github.io. (n.d.). Deep Learning for Survival Analysis. [online] Available at: https://humboldt-wi.github.io/blog/research/information$_s ystems_1$920/$group2_s urvivalanalysis/motivation$[$Accessed19Aug$.202

23. Moore, Dirk F. Applied Survival Analysis Using R. Springer Science+Business Media, 2016.

24. Azur, M.J., Stuart, E.A., Frangakis, C. and Leaf, P.J. (2011). Multiple imputation by chained equations: what is it and how does it work? International Journal of Methods in Psychiatric Research, 20(1), pp.40–49. doi:10.1002/mpr.329.

25. E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," Journal of the American statistical association, vol. 53, no. 282, pp. 457–481, 1958.

26. Howe, C.J., Cole, S.R., Chmiel, J.S. and Muñoz, A. (2011). Limitation of Inverse Probability-of-Censoring Weights in Estimating Survival in the Presence of Strong Selection Bias. American Journal of Epidemiology, 173(5), pp.569–577. doi:10.1093/aje/kwq385.

27. Luis Jiménez-Moro, J. and Gómez, J. (n.d.). Inverse Probability of Censoring Weighting for Selective Crossover in Oncology Clinical Trials. [online] Available at: https://www.lexjansen.com/phuse/2014/sp/SP02.pdf [Accessed 19 Aug. 2022].

28. Vock, D.M., Wolfson, J., Bandyopadhyay, S., Adomavicius, G., Johnson, P.E., Vazquez-Benitez, G. and O'Connor, P.J. (2016). Adapting machine learning techniques to censored time-to-event health record data: A general-purpose approach using inverse probability of censoring weighting. Journal of Biomedical Informatics, 61, pp.119–131. doi:10.1016/j.jbi.2016.03.009.

29. E.L. Kaplan, P. Meier Nonparametric estimation from incomplete observations J Am Stat Assoc, 53 (1958), pp. 457-481

30. E.T. Lee, J.W. Wang Statistical methods for survival data analysis (3rd ed.), John Wiley Sons, Hoboken, NJ, USA (2003)

31. Readthedocs.io. (2022). What is censored data — reliability 0.8.6 documentation. [online] Available at: https://reliability.readthedocs.io/en/latest/What%20is%20censored%20data.html: :text=There%20are%20t [Accessed 19 Aug. 2022].

32. P.J.F. Lucas, L.C. van der Gaag, A. Abu-Hanna Bayesian networks in biomedicine and health-care Artif Intell Med, 30 (3) (2004), pp. 201-214

33. Kraisangka, J. and Druzdzel, M.J. (2018). A Bayesian Network Interpretation of the Cox's Proportional Hazard Model. International Journal of Approximate Reasoning: Official Publication of the North American Fuzzy Information Processing Society, [online] 103, pp.195–211. doi:10.1016/j.ijar.2018.09.007.

34. Loghmanpour NA, Kanwar MK, Druzdzel MJ, Benza RL, Murali S, Antaki JF, A new Bayesian network-based risk stratification model for prediction of short-term and long-term LVAD mortality, ASAIO Journal 61 (2015) 313–323.

35. Murphy KP, Dynamic Bayesian networks: Representation, inference and learning, Doctoral dissertation, University of California, Berkeley, 2002.

36.Van Gerven MA, Taal BG, Lucas PJ, Dynamic Bayesian networks as prognostic models for clinical patient management, Journal of Biomedical Informatics 41 (2008) 515–529.

37. Bandyopadhyay et al. [18] Under 4 77. Robins and Finkelstein 2000; Bang and Tsiatis 2000, 2002; Rotnitzky and Robins 2004; Tsiatis 2006

A1 Appendix

A1: Data analysis obtained in R

A2: Obtained IPCW weights in R

A3: created Bayesian network model for IPCW weight using exact and greedy search algorithm in python

A4: Obtained weighted censored instances in R

A5: created Bayesian network model for weighted censored instances using exact and greedy search algorithm in python