University of Macau

Faculty of Science and Technology

澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Visualizing Decision Rules and Relations for Medical Data Analysis: Modeling

by

CHEANG WENG HEI, Student No: DB625417

Final Project Report submitted in partial fulfillment
of the requirements of the Degree of
Bachelor of Science in Computer Science

Project Supervisor

Prof. Simon FONG & Prof. Shirley SIU

05 June 2020

DECLARATION

I sincerely declare that:

1. I and my teammates are the sole authors of this report,
2. All the information contained in this report is certain and correct to the best of my knowledge,
3. I declare that the thesis here submitted is original except for the source materials explicitly acknowledged and that this thesis or parts of this thesis have not been previously submitted for the same degree or for a different degree, and
4. I also acknowledge that I am aware of the Rules on Handling Student Academic Dishonesty and the Regulations of the Student Discipline of the University of Macau.

Signature : ___________________________

Name : CHEANG WENG HEI

Student No. : DB625417

Date : 05 June 2020

ACKNOWLEDGEMENTS

The author would like to express his utmost gratitude to UM for providing the opportunity to carry out a project as a partial fulfillment of the requirement for the degree of Bachelor of Science.

Throughout this project, the author was very fortunate to receive the guidance and encouragement from his supervisor Prof. Simon FONG & Prof. Shirley SIU.

ABSTRACT

Visual technique is very useful in data exploration because of the phenomenal abilities of the human visual system to detect structures and relations in images. This is sometimes called visual data mining or visual mining that makes learning from visually presented information faster and is quite useful in exploration stage when the exact prediction target may be not very well known or needs to be confirmed preliminarily via studying the abstract data graphically. In this project, we are going to apply this technology into the medical diagnosis process. The advantage of visualization can benefit and accelerate the medical diagnosis process. We build graphical model of relationship between different diseases and symptoms by analysing the medical data set. The analysis process involves Bayesian network and Decision tree model learning. This paper shows how we can use these techniques to analyse medical data sets and provide visualization results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. Introduction

1.1 Overview

1.1.1Background

Medical diagnosis is a main part of in medicine. It is the first and important procedure to save the patient’s life and improve health. As many possible reasons are involved, it is not easy for medical diagnosis and identify the cause of the illness. Traditionally, a doctor can only give a diagnosis result based on his/her knowledge and experience. Mentioned in [1], many researchers combine computer science and medical science in recent years. By taking advantage of computer science, medical research can be efficiently. Although computers cannot completely replace doctors for medical diagnosis, computers provide doctors with useful tools to help doctors make diagnosis efficiently and accurately.

1.1.2Research problem

Using computer technology in medical data analysis accelerate medical progress. Nowadays, artificial intelligence and mechanical learning are already mature. By taking advantage machine learning, computer can generally provide a reliable and accuracy result. The traditional machine learning model is a black box learning process, and this technology sometime lacks interpretability. In medicine, interpretability is important for doctors to analyse and interpret data. It enables doctors to determine causal relationship between diseases, living habit, and other features.

Finding causal relationship is not enough, an intuitive representation of causal relationship is also required. A graphical representation is preferred because humans have the strong power to analyse graph. Data and graph are two different representation of information, converting data to graph is a challenge. No universal visualization tool is existed. Depending on the application, special visualization tool needs to be developed.

1.1.3Importance

Visualizing data is important in data analysis. Humans have the strong power to analyse structures and relationships by studying graphical data. It may facilitate data mining that learning from visually presented information may faster. It is useful in the stage of exploration if exploring goal is not well known. It may also help people to confirm preliminarily knowledge quickly by studying the graphical data.

1.1.4Motivation

This final year project is focused on analysing medical data. Our motivation is providing useful information for doctor or medical research by taking the advantage of computer power. Data information is not intuitive enough, providing a better result by using visualizing technology is also our motivation. By the advantage of the phenomenal abilities of the human visual system, visualizing result can improve understandability, usability, and efficiency.

1.2 Objectives

1.2.1Goal

This approach is quite opposite to formal methods of model building and testing, but it is ideal for searching through data to find unexpected or unusual relationships. Therefore, the objective of this FYP is focused on a vertical solution for analysing medical data, which joins advantages of several data mining techniques in one system, which consist of following parts: Data recognition-this subsystem transforms raw data to a form suited for further data processing. Additionally, noise and redundant data are removed based on a statistical analysis. Feature subset selection which is responsible for selecting an optimal set of attributes for a clean generation of decision rules Rule induction subsystem which uses both classical machine learning algorithms as well as new ones to be proposed. Visualization of the collected knowledge in a form easily understandable by humans. In this FYP we extend some initial research on the data visualization on both cancer and liver malfunction testing data, which are real from a local and international hospital.

In addition, many of the computer technologies we learned in our bachelor's degree programs are useful for analysing medical data, including data processing, machine learning and visualization. Therefore, we would like to apple those technologies in this final year project. Data processing can deal with raw data and transform it into research-worthy data. Machine learning enables us to learn data and generate useful prediction. Visualization provides a concise, understandable, and user-friendly representation to demonstrate the final result. We want to design a system that combines the advantages of different computer technologies which we have already learned to produce useful medical results.

1.2.2How to achieve

In this project, we divide it into two part: machine learning & visualizing. To find out and understand relationship between behaviours and the disease, we try to use algorithms like Bayesian network, decision tree etc. Besides, to make the result more ‘user friendly’ we are going to make some improvement through existed visualizing tool.

For the machine learning part, the main workflow is analysing the data to generate a useful result for the visualization part. A set of medical data is given. The data was collected by other medical researchers and published on the Internet. The data includes patients living habits, diseases, and other factors. Then, data processing methods can be used to filter out the invalid data. By using machine learning method, the relationships between different attributes can be found.

For the visualization part, the main task is to develop a visualization software to show the result of the machine learning. The result of the machine learning part is given by another sub team. Based on the result, a concise, understandable, and user-friendly graphical representation should be provided. Since this software is designed for doctors, useful functions for doctors should be considered.

1.2.3Problems we may face

However, there are so many machine learning algorithms that we have to develop an own judgement method to choose the suitable model. After choosing the best model to implement, we also have to choose the rules generated from the model and decide to reduce the inappropriate ones. Moreover, the most difficult part is how to design the visualizing and interface to help user understand the result better.

Integration of system may one of the problems because this project includes two different parts. We are divided into two sub teams, and work on different parts respectively. Finally, we should integrate them to provide the result. In the real world, integration is generally a difficult part because our software design may conflict with each other. Therefore, we should have good communication to reduce conflict of software design.

Medical data analysis is extremely difficult because there are many unstable and random factors in medicine. Some medical researchers have spent years and a lot of energy, and they still cannot get a perfect result. Having an unsatisfactory result is common in medical research. Besides, we are just computer students, not medical students. Regardless of whether the results are perfect, we should focus on the application of computer technology in different fields.

CHAPTER 2. Literature Survey/Related Work

2.1 Machine learning for medical diagnosis

Medical diagnosis is an important part in medical situation. A precise diagnosis can save the human life and improve their health. However, a wrong diagnosis may lead to serious consequences.

As described in [1], many medical institutions use computer information tools for diagnosis now. Due to technical limitations, they are not a substitute for a doctor to make a direct medical diagnosis now. However, they support doctors by selecting or generate related data for the medical diagnosis. They make medical diagnosis procedure to be more efficient and accurate.

Mentioned in [1], machine learning is an effective technology for medical diagnostic systems. Bayesian networks are one of these methods. They are a graphical model that includes a set of variables and their probabilistic independences. They are always used to handle uncertain problems. They can be used for medical decision-making process. In medical diagnosis procedure, they can predict the likelihood of a patient's illness based on detected symptoms.

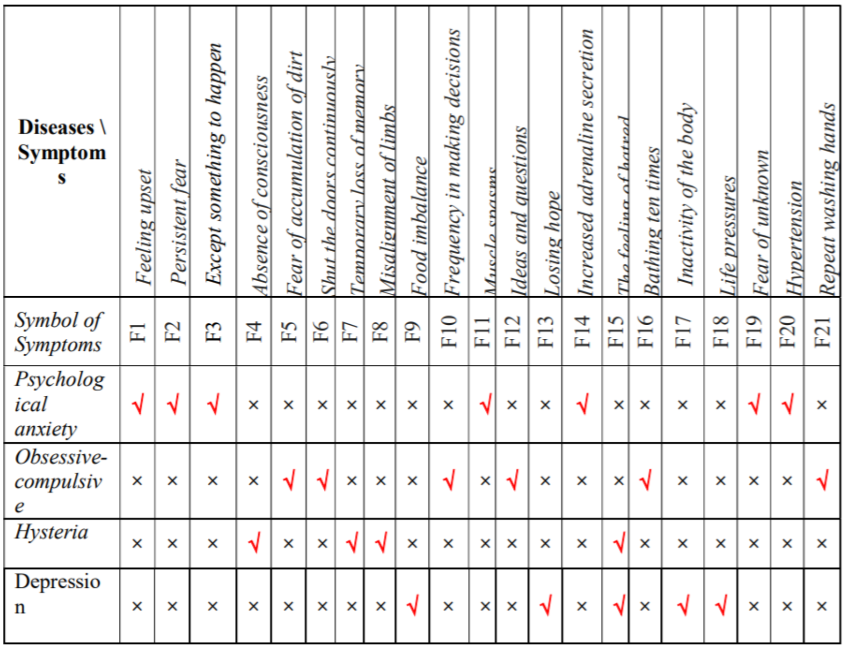
Also mentioned in [2], Bayesian networks are graphical presentation of relationships between different variables. In Bayesian networks, the nodes represent variables. The arcs between different variables represent causal, influential, or correlated relationships. The structure of the Bayesian networks is very suitable for medical decision making.

In [1], the author introduced the four case studies based on recorded medical evidence and statistics from Lugoj Municipal Hospital. Author proved that Bayesian networks are an efficient tool for the doctors to predict and treat disease.

Author of [2] said that most of the real-world problems involve uncertain variables and data, especially for medical related problems. These problems are perfectly represented by Bayesian network. Thus, Bayesian networks caught people's attention. Application of Bayesian networks are now popular in solving medicine problem. They are been known as a powerful tool for risk analysis and decision support in real-world.

2.2 Existing medical diagnosis systems and their deficiencies

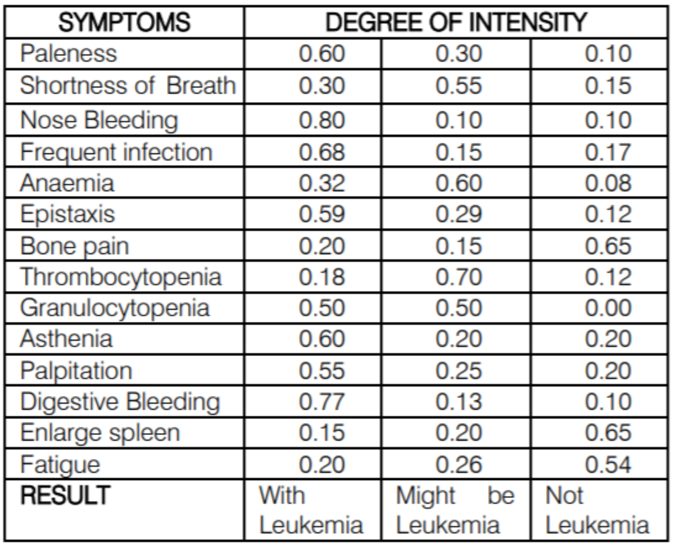
Ahmad A. Al-Hajji and partners [17] developed a rule-based expert system models for psychological diseases diagnosis and classification. This system can diagnose several types of psychiatric diseases, such as depression, anxiety disorder, obsessive-compulsive disorder, and hysteria. Their model is based on the backward chaining, also called goal-driven reasoning. The knowledge is represented by a set of IF-THEN rules. The system allow user to enter the symptoms. The program analysis the symptoms by using a set of IF-THEN rules and classify the disease.



Diseases \ Symptoms	Feeling upset	Persistent fear	Except something to happen	Absence of consciousness	Fear of accumulation of dirt	Shut the doors continuously	Temporary loss of memory	Misalignment of limbs	Food imbalance	Frequency in making decisions	Muscle spasms	Ideas and questions	Losing hope	Increased adrenaline secretion	The feeling of hatred	Bathing ten times	Inactivity of the body	Life pressures	Fear of unknown	Hypertension	Repeat washing hands
Symbol of Symptoms	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21
Psychological anxiety	√	√	√	×	×	×	×	×	×	×	√	×	×	√	×	×	×	×	√	√	×
Obsessive-compulsive	×	×	×	×	√	√	×	×	×	√	×	√	×	×	×	√	×	×	×	×	√
Hysteria	×	×	×	√	×	×	√	√	×	×	×	×	×	×	√	×	×	×	×	×	×
Depression	×	×	×	×	×	×	×	×	√	×	×	×	√	×	√	×	√	√	×	×	×

Figure 1: IF-THEN rules decision table of Ahmad A. Al-Hajji's project

J C Obi and A. A. Imianvan [18] developed a hybrid Neuro–Fuzzy Expert System. Its logic is similar to neural networks to finds the parameter of a fuzzy system. It helps in diagnosis of Leukemia using a set of symptoms. Their system checks the number of symptoms the patient has. It is an interactive system that can tell the patient his current condition as regards Leukemia.



SYMPTOMS	DEGREE OF INTENSITY		
Paleness	0.60	0.30	0.10
Shortness of Breath	0.30	0.55	0.15
Nose Bleeding	0.80	0.10	0.10
Frequent infection	0.68	0.15	0.17
Anaemia	0.32	0.60	0.08
Epistaxis	0.59	0.29	0.12
Bone pain	0.20	0.15	0.65
Thrombocytopenia	0.18	0.70	0.12
Granulocytopenia	0.50	0.50	0.00
Asthenia	0.60	0.20	0.20
Palpitation	0.55	0.25	0.20
Digestive Bleeding	0.77	0.13	0.10
Enlarge spleen	0.15	0.20	0.65
Fatigue	0.20	0.26	0.54
RESULT	With Leukemia	Might be Leukemia	Not Leukemia

Figure 2: Diagnosis of Leukemia in J C Obi and A. A. Imianvan's project

There are many people work on the related projects, many successful medical diagnosis systems are built. However, most of them just focus on text-based result and black-box model. They just provide the diagnosis result to the user. User cannot know the relationship between different attributes, including diseases, symptoms and living habits. Also, they lack graph-based diagnosis result.

2.3 Bayesian network

Bayesian networks are a kind of graphical model. They can be used for representing the dependence relationship between a given set of random variables as a directed acyclic graph. According to the definition in the book "Bayesian Networks with Examples in R" [4], Bayesian networks graph can be defined as

```
G = (V,A)
```

They include two parts, a set of nodes V and a set of arcs A. An arc is either an ordered pair or a non-ordered pair. Generally, an arc is called directed arc if it is an ordered pair. Otherwise, it is called undirected arc. Each arc can be defined as

```
a = (u,v)
```

It represents that the arc is outgoing for u and it is incoming for v (u→v) if it is a directed arc. Otherwise, it is represented with a simple line (u-v). Depend on characterization of arcs, a graph can be defined as directed graphs if all arcs are directed, undirected graphs if all arcs are undirected, or partially directed graph if it includes both directed and undirected arcs (see Figure 3 and Figure 4) (Figures are from [3]).

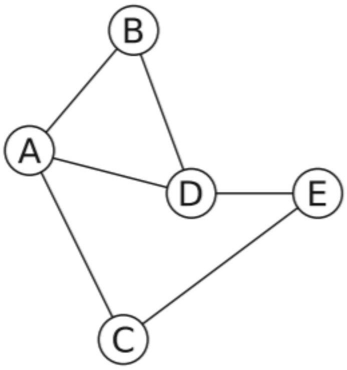


Figure 3: Example of undirected graph

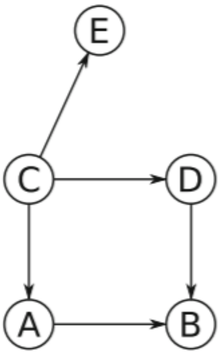


Figure 4: Example of directed graph

Different to other type of graph, Bayesian networks focuses on probability. They assign a probability to each measurable set of events. For discrete case, it is represented as conditional probability tables. This is the most common in the real-world application. For continuous case, it is represented as linear models [3].

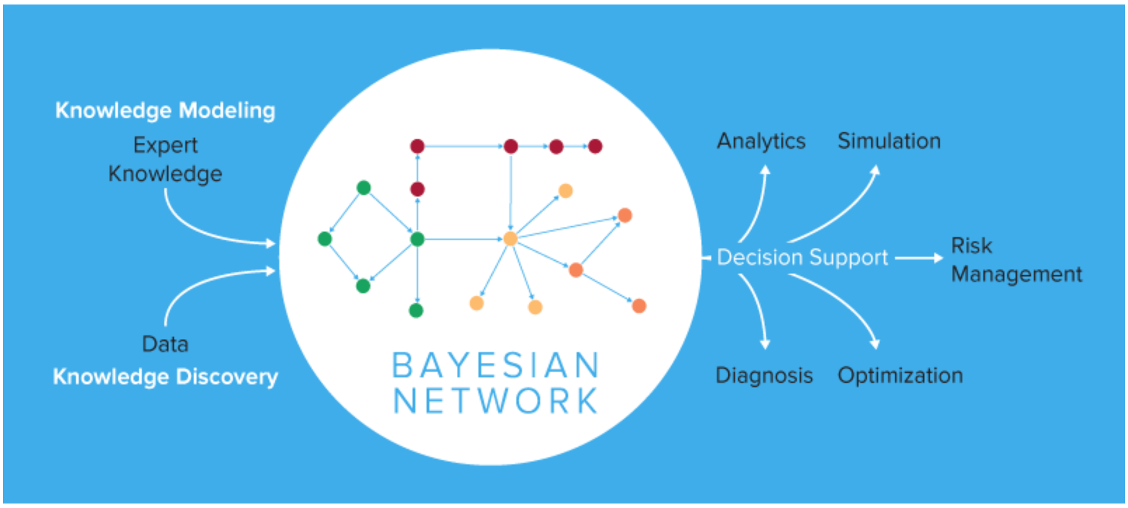


Figure 5: Example of Bayesian Network workflow

2.4 Advantages and difficulty of Bayesian network

Bayesian networks are a powerful method to model uncertain variables and data. Thus, they become a popular tool for the analysis of uncertain problems. They provide a lot of many advantages to solve scientific problems. However, nothing is perfect, Bayesian networks also have some disadvantages.

They are the key points of the Bayesian networks. Author of [5] said that they provide a probabilistic representation of the interaction between different nodes. They make use of probability to measure uncertain data. If the uncertainty is higher, probability distribution is wider. As the amount of information increases, probability distribution become narrower, it means uncertainty become lower. They allow users to better estimate risks and uncertainties. They have strong data analysis capabilities in areas full of uncertainty. They have the ability to convert uncertain problems to be possible. Therefore, they are often used in medicine and medical research because this type of research is always full of uncertainty.

According to [6], the second advantage of Bayesian networks is that they create a causal relationship between variables. Traditionally, people use black-box machine learning methods to do the research. It generally accepts a set of input variables, then it returns a result. There is hidden layer in the black-box machine learning methods. It is difficult to know what the hidden layer is doing. Thus, causal relationship between variables cannot be provided by black-box methods.

As mentioned in [6], Bayesian networks are white-box methods. They provide a direct presentation of the uncertain interactions between causes and effect. White-box methods allow incorporate domain knowledge in the model. Having representation of the uncertain relationships are useful for solving real world problems, such as including diagnosis, forecasting and information retrieval. For example, this help doctors and research to explain the disease.

In many other areas of research, dataset usually includes continuous data. As shown in [5], Bayesian networks can handle continuous data, but they have less support on continuous variables than other machine learning method. Generally, people solve this problem by converting continuous variables to discrete variables. This means that they discretize variables during the pre-processing data phase.

This is a trade-off. After discretization, the data can only retain the rough features of the original distribution. If too many characteristics of the original data are missing, it is impossible to show the exact relationship between variables. Discretization is an important task that can affect results. Discretization should carefully consider intervals and breakpoints, try different combinations to find the best combination. This is a time-consuming task.

2.5 Available tool for Bayesian network learning

2.5.1R and its libraries

R [9] is one of the programming languages and environments for computing and graphics. Similar to Python, another well-known programming language, it also has great library support for machine learning. It can be used to develop machine learning and analyse data, and to solve problems in the real world. Under the terms of the Free Software Foundation's GNU General Public License, it is free that everyone has the rights to freely use, study, and modify it. In other words, it can be freely used for academic research of students.

According to [9], compared with other programming languages, R provides more extensive support for graphical techniques. Good at providing high-quality graphical solutions. The key to Bayesian networks is to provide a graphical representation of the data analysis. R is a suitable choice for developing Bayesian network learning.

According to [3], many people have developed useful libraries for Bayesian network implementations because of the advantage of the R language. Libraries bnlearn, deal, pcalg and pcalg are popular libraries dealing with Bayesian networks. They respectively provide different functions. The main different of them have been showed in the following table (see Table 1).

Table 1: Comparison of built-in libraries

	bnlearn	**deal**	**pcalg**	**pcalg**
Processing discrete data	Yes	Yes	Yes	Yes
Processing continuous data	Yes	No	Yes	Yes
Constraint-based learning	Yes	No	No	Yes
Score-based learning	Yes	Yes	Yes	No
Parameter estimation	Yes	Yes	Yes	Yes
Prediction	Yes	Yes	No	No

Most of them provide basic structure learning algorithms and parameter learning approaches for both discrete and continuous data. They are the basis for implementing Bayesian networks.

Based on them, implementations can be faster and more efficient. These features can be used to build our own systems instead of writing everything from scratch. According to software engineering principles, save the cost of implementing basic functions as soon as possible, and focus on the main goal, which is to implement Bayesian network learning for medical data analysis. More time resources can be used to optimize the results, more suitable solutions for medical data analysis can be provided.

2.5.2Bnlearn

bnlearn is an R package for learning the graphical structure of Bayesian networks, estimate their parameters and perform some useful inference. It was first released in 2007, it has been under continuous development for more than 10 years (and still going strong). [7] In this period, we use ‘Hill Climbing’, a score-based structure learning algorithm from bnlearn to generate the plot.

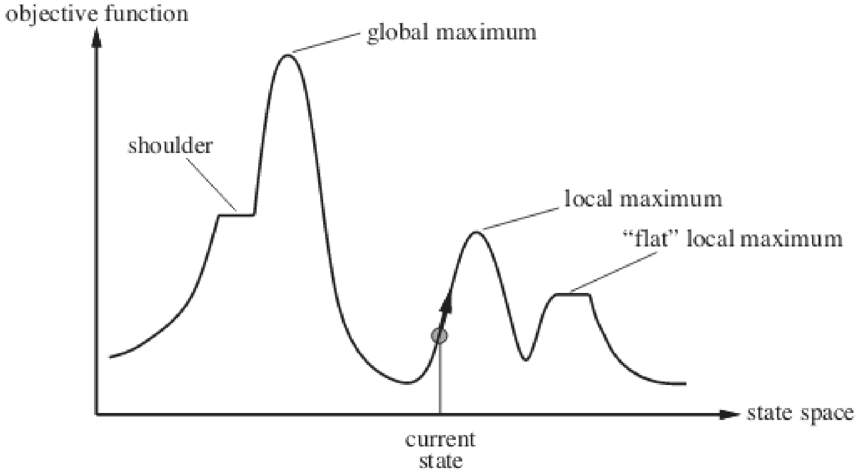


Figure 6: Example of Hill Climbing graph

Algorithm 1: Structural learning of BNs by using a hill climbing (HC) algorithm [8]

```
    Input: D: A dataset defined over variables V = {X1,..., Xn}
    Input: G0: A DAG defined over V used as the starting point for the search
    Output: A DAG G being the graphical part of network B
G←G0;
fG←f(G:D)//*
[r]f: decomposable scoring metric improvement←true;
while improvement do
        improvement ← false;
        //*
        [h]neighbors generated by addition
        For each node Xi and each node X j ∈/ PaG(Xi) such that X j → Xi does not
        introduce a directed cycle in G, compute the difference
        diff = $f(G+\{Xj \rightarrow Xi\}: D) - fG$. Store the change which maximizes diff in
        $\langle change_a, diff_a \rangle$;
        //*
         [h]neighbors generated by deletion
         For each node Xi and each node $X\,j \in PaG(Xi\,)$, compute the difference
        diff = $f(G-\{Xj \rightarrow Xi\}: D) - fG$. Store the change which maximizes diff in
        $\langle change_d, diff_d \rangle$;
        //*
        [h]neighbors generated by reversal
        For each node Xi and each node $X\,j \in PaG(Xi\,)$ such that reversing $X\,j \rightarrow Xi$
        does not introduce a directed cycle in G, compute the difference diff = d1 + d2
        where d1 corresponds to $f(G-\{Xj \rightarrow Xi\}: D) - fG$ andd2 corresponds to
        $f(G+\{Xj \rightarrow Xi\}: D) - fG$.Store the change which maximizes diff in $\langle change_r$ ,
        $diff_r \rangle$;
        //*
        [h]Checking if improvement
        Let $d^* = max_{k= a,d,r} diff_k$ and move* its corresponding change;
        if $d^* > 0$ then
            improvement ← true;
            G ← apply move* over G;
            fG ← fG + d*;
        end
    end
    return G;
```

Hill Climbing is a heuristic search to find out the optimal result in the function, the main idea is to compare the point next to the current point and determine the continuous direction. In our project we use it to learn a network structure to attain relationship of different attribute from the dataset.

2.6 Bayesian network structure learning algorithms

According to [4], structure learning is the process of identifying the graph structure of a Bayesian network by analysing data. This process can be done manually from existing human knowledge. This method is usually used to represent some existing knowledge. It can also be done automatically using computer algorithms. This method is usually used to study some new knowledge. Bayesian network structure learning algorithms can be divided into two categories. They are constraint-based algorithms and score-based algorithms.

As mentioned in [10], constraint-based algorithms learn network structure by analysing the probability relationships brought by Markov property. Constraint-based algorithms are based on Verma and Pearl's inductive causality algorithms. They provide a framework for learning Bayesian network structures using conditional independence tests. Common constraint-based algorithms

According to [10], general constraint-based algorithms can be expressed in the following form:

1. First the skeleton of the network (the undirected graph underlying the network structure) is learned. Since an exhaustive search is computationally unfeasible for all but the simplest data sets, all learning algorithms use some kind of optimization such as restricting the search to the Markov blanket of each node (which includes the parents, the children and all the nodes that share a child with that particular node).
2. Set all direction of the arcs that are part of a v-structure (a triplet of nodes incident on a converging connection Xj → Xi ← Xk).
3. Set the directions of the other arcs as needed to satisfy the acyclicity constraint.

As mentioned in [4], score-based algorithms learn network structure by using heuristic optimization technique. Each candidate network is assigned a score which represent goodness of the current candidate network and try to get the maxima with some heuristic search algorithm. Greedy search algorithms are generally be used, such as hill-climbing search.

According to [4], general score-based algorithms can be expressed in the following form:

1. Choose a network structure, usually empty.
2. Compute the score of the network structure.
3. Set maxscore = score.
4. Repeat the following steps as long as maxscore increases:
    a. For every possible arc addition, deletion or reversal not resulting in a cyclic network,
    b. compute the score of the modified network,
    c. and update maxscore.
5. Return the directed acyclic graph.

2.7 Built-in plotting in Bayesian network and its shortcomings

Although a simple plotting function is already provided by bnlearn [7] package in R, t this function is too simple to meet our requirements. Figure 7 is a simple plotting generated by this simple plotting function. In this case, there are 70 nodes in this graph. We can no longer clearly see these nodes. It is only extreme simple plotting. Even it does not allow us to change the text size, enlarge, or colour important paths. It is only suitable for extreme small data set (may be less than 10 attributes). It is not suitable for displaying the result of our project. That is the reason why our group have visualization part to develop our own plotting program.

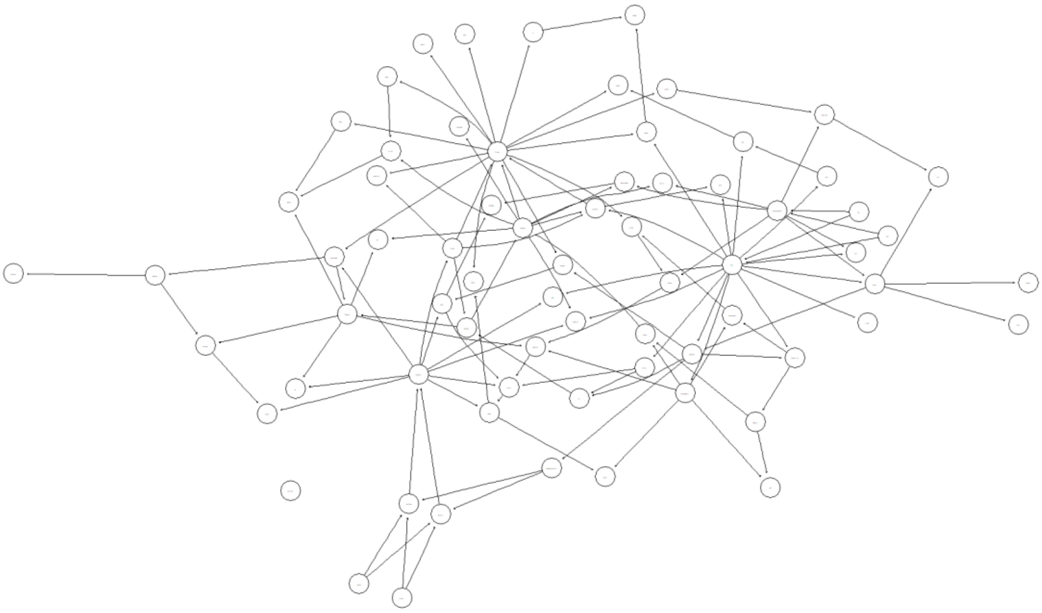
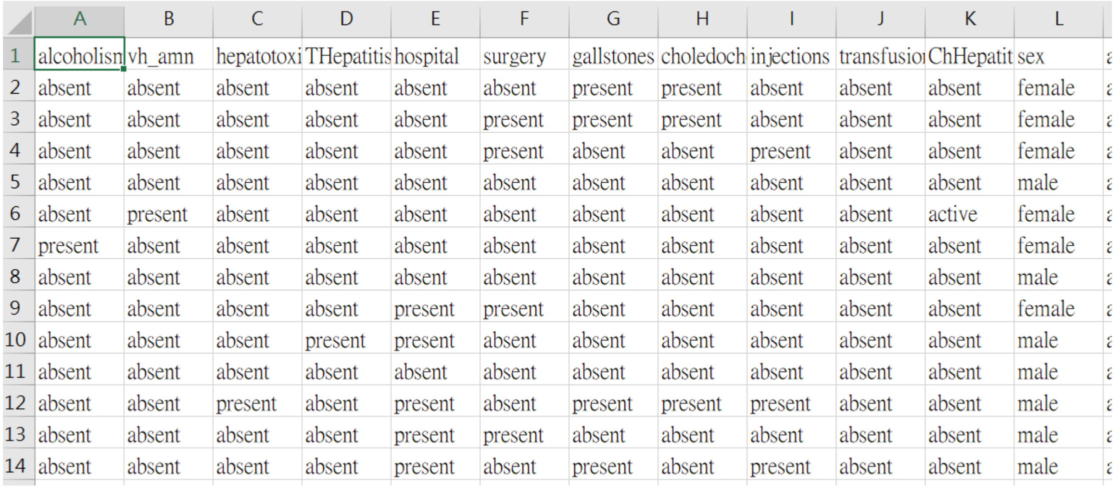


Figure 7: Built-in simple plotting in Bayesian network

2.8 Related data set

Suggested by our supervisors, there are two public data sets are useful for this project. They are "HEPARTWO10k" and "Tox21".

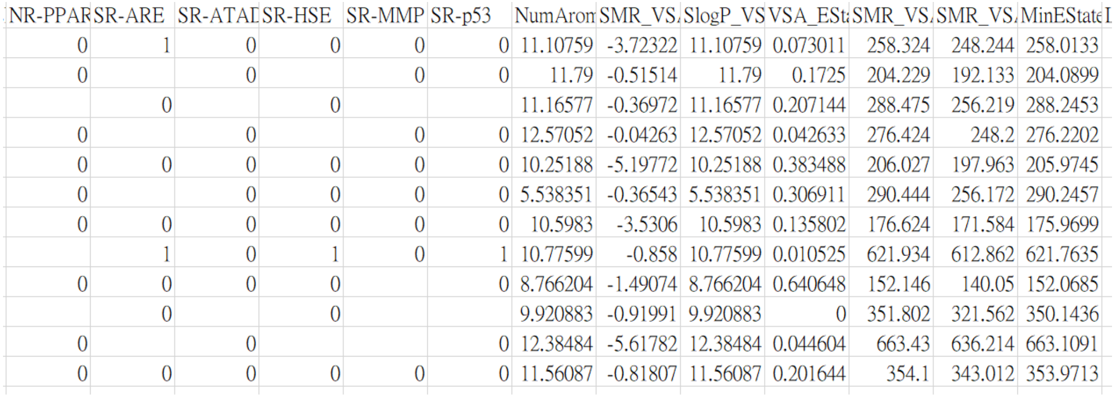
"HEPARTWO10k" is from the book "Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders" [12] A. Onisko. *Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders*, March 2003.[12]. It is a liver cancer data set. It involves 70 attributes and 10,000 rows of sample data. It is a professional data set created by some medical experts. Any research that people can do based on this data is public.



	A	B	C	D	E	F	G	H	I	J	K	L	
1	alcoholisn	vh_amn	hepatotoxi	THepatitis	hospital	surgery	gallstones	choledoch	injections	transfusioı	ChHepatit	sex	a
2	absent	absent	absent	absent	absent	absent	present	present	absent	absent	absent	female	a
3	absent	absent	absent	absent	absent	present	present	present	absent	absent	absent	female	a
4	absent	absent	absent	absent	absent	present	absent	absent	present	absent	absent	female	a
5	absent	absent	absent	absent	absent	absent	absent	absent	absent	absent	absent	male	a
6	absent	present	absent	absent	absent	absent	absent	absent	absent	absent	active	female	a
7	present	absent	absent	absent	absent	absent	absent	absent	absent	absent	absent	female	a
8	absent	absent	absent	absent	absent	absent	absent	absent	absent	absent	absent	male	a
9	absent	absent	absent	absent	present	present	absent	absent	absent	absent	absent	female	a
10	absent	absent	absent	present	present	absent	absent	absent	absent	absent	absent	male	a
11	absent	absent	absent	absent	absent	absent	absent	absent	absent	absent	absent	male	a
12	absent	absent	present	absent	present	absent	present	present	present	absent	absent	male	a
13	absent	absent	absent	absent	present	present	absent	absent	absent	absent	absent	male	a
14	absent	absent	absent	absent	present	absent	present	absent	present	absent	absent	male	a

Figure 8: Part of the data set "HEPARTWO10k"

The “Toxicology in the 21st Century” (Tox21) initiative created a public database measuring toxicity of compounds, which has been used in the 2014 Tox21 Data Challenge [16]. This dataset contains qualitative toxicity measurements for 8k compounds on 12 different targets and 111 features, including nuclear receptors and stress response pathways.



NR-PPAR	SR-ARE	SR-ATAD	SR-HSE	SR-MMP	SR-p53	NumArom	SMR_VSA	SlogP_VS	VSA_ESta	SMR_VSA	SMR_VSA	MinEState	I
0	1	0	0	0	0	11.10759	-3.72322	11.10759	0.073011	258.324	248.244	258.0133	
0		0		0	0	11.79	-0.51514	11.79	0.1725	204.229	192.133	204.0899	
	0		0			11.16577	-0.36972	11.16577	0.207144	288.475	256.219	288.2453	
0		0		0	0	12.57052	-0.04263	12.57052	0.042633	276.424	248.2	276.2202	
0	0	0	0	0	0	10.25188	-5.19772	10.25188	0.383488	206.027	197.963	205.9745	
0		0	0	0	0	5.538351	-0.36543	5.538351	0.306911	290.444	256.172	290.2457	
0	0	0	0	0	0	10.5983	-3.5306	10.5983	0.135802	176.624	171.584	175.9699	
	1	0	1	0	1	10.77599	-0.858	10.77599	0.010525	621.934	612.862	621.7635	
0	0	0	0		0	8.766204	-1.49074	8.766204	0.640648	152.146	140.05	152.0685	
	0		0			9.920883	-0.91991	9.920883	0	351.802	321.562	350.1436	
0		0			0	12.38484	-5.61782	12.38484	0.044604	663.43	636.214	663.1091	
0	0	0	0	0	0	11.56087	-0.81807	11.56087	0.201644	354.1	343.012	353.9713	

Figure 9: Part of the data set "Tox21"

CHAPTER 3. Project Execution Schedule

3.1 Project schedule

In the first semester, we conducted grouping and topic selection. Since we do not have much knowledge related to the project topic, we spent more time on this part to learn to ensure that we have enough knowledge to carry out further work. In addition, we conducted a literature survey to study some related projects. We learned some useful knowledge from them. In addition, we also consider more on our project based on the experience of others. At the same time, we sought out some available software that was useful for our project. We also consider the modifiability and applicability of these software.

The following table showed that our schedule in first semester:

Table 2: Schedule in first semester

Task	Month			
	September 2019	October 2019	November 2019	December 2019
Grouping and selecting topic	🟩			
Learn the basics of machine learning and visualization	🟩	🟩		
Literature survey			🟩	🟩
Finding available software			🟩	🟩

In second semester, we made an interim report at the beginning. The report records all work in the first semester. According to the first semester of research, we already have enough work to do. We began to design and implement our project. Unfortunately, due to the coronavirus, progress in this part has been delayed. Therefore, we spend more time on this part than we expected. After design and implementation, we conducted tests and evaluations to examine our project in detail. Finally, we organized the final report and prepared a presentation at the end of the second semester.

The following table showed that our schedule in second semester:

Table 3: Schedule in second semester

Task	Month

	January 2020	February 2020	March 2020	April 2020	May 2020
Documenting interim report	🟩				
Design and implantation		🟩	🟩	🟩	
Testing and evaluation				🟩	
Documenting final report and preparing presentation					🟩

3.2 Meeting schedule before class suspension

The schedule is based on the meeting with supervisors every two weeks; therefore, we will show the finished items and schedule parts from the meeting notes.

- **25-09-2019.** Discussion At this meeting, our supervisors introduce the background information of our title. Also, they briefly tell me the goal of our project. It makes us more familiar with our project. In the machine leaning part, we need to make a decision that what model we use in the project. This decision is important. It may affect accuracy, efficiency, ... etc. Our supervisors suggest us that we do the research of different models in the following two weeks, such as Bayesian network, decision tree, ... etc. Compare them and make decision at the next meeting. Before next meeting, we will do the research of different models in two weeks. At the next meeting, we will discuss different models. Select the most suitable models for our final year project.

- **23-10-2019.** Did a simple study to understand what Bayesian network is. Teammate in another subgroup showed some demos of Bayesian network visualization. We discussed the advantage and disadvantage of those demos. We asked supervisors that what is division of labor between two subgroups. And how to integrate outputs from two subgroup to achieve our goal. Supervisors suggested some tools which is useful for machine learning. They are Weka, sklearn in Python and bnlearn in R. We will study the tools they suggested in the discussion.

- **13-11-2019.** Learned the R programming language. Studied useful libraries for machine learning in R. Wrote a program to try machine learning in R. Showed the program that written to try machine learning in R. We discussed how to improve this program. Supervisors said that this program is a good start. * But they suggested that we should learn more about the principles of Bayesian networks. They advised us not to use only R language. We should try more options and compare their different. We will try different programming languages and tools. Also, we will improve our existing programs.

- **27-11-2019.** Using Weka to compare different classifier for the big dataset provided by Prof. Fong. Studied useful libraries for machine learning in R. Modify the pervious program in R so that the program can generate different plots of the results. Discussion: Showed modification of the program that generate different plots. We discussed how to improve this program so that to apply to dataset provided by professor than using our small testing dataset. Supervisors said that we should try all different classifiers in Weka to pick a best one rather than use it all. They suggested that we still need to keep learning more about the principles of Bayesian networks and other machine learning. We also need to keep contacting office to get the license of Bayesia software. We plan to have next meeting in January after holiday. We will try all classifiers in Weka to get the most suitable one for the machine learning and to figure out how this classifier is used in the project. We will keep improving our existing programs and use the book 'Bayesian Networks in R' to do background study.

- **15-01-2020.** Made some modifications on the pervious R program with 'averaged. network'. Tested 'Bayesian Network', 'Naive Bayesian Network', 'Logistic', 'Random Forest',' Random Tree' in Weka. We start to have a plan for some interface for this project suggested by supervisors. We will learn and try more on Weka to analysis more different algorithms rather than focus on the ones we tried.

3.3 Meeting schedule during class suspension

After the last meeting on January 15th, we can no longer have physical meetings on campus because of the coronavirus. For this reason, we turn to the online meeting. During the suspension, we tried to use Zoom to organize video conferences. Unfortunately, it was difficult for one of our supervisors to attend our meeting because he encountered some technical problems with the network connection. Due to network connection issues, video conferencing may not be suitable for us.

Based on the experience of failure, we believe that text discussion on WeChat may be a better solution for us. For future discussion, we only did text discussion on WeChat. We no longer organized any video conferences. The format of the meeting was changed to text discussion on WeChat. I send text messages, and others can reply to me when available. We no longer have a formal meeting. We discuss freely on WeChat at any time. However, it is difficult to record a complete formal meeting log.

CHAPTER 4. Design

Basically, our project includes two different parts: machine learning and visualization. I work on machine learning part. For machine learning part, our objective is to learn medical data by using machine learning method and retrieve useful result for visualization from machine learning result. The result is passed to my groupmate for generating a graph to achieve visualization.

My part is mainly done in R [9] language. General machine learning can basically be divided into the following steps: data input, data pre-processing, model training, model testing, new data prediction. However, our project topic is not only focus on machine learning. We also use machine learning results to find relationships in medical data and output useful data for visualization. Thus, there is one more step in project comparing to general machine learning project. The overall flow of our project is shown below:

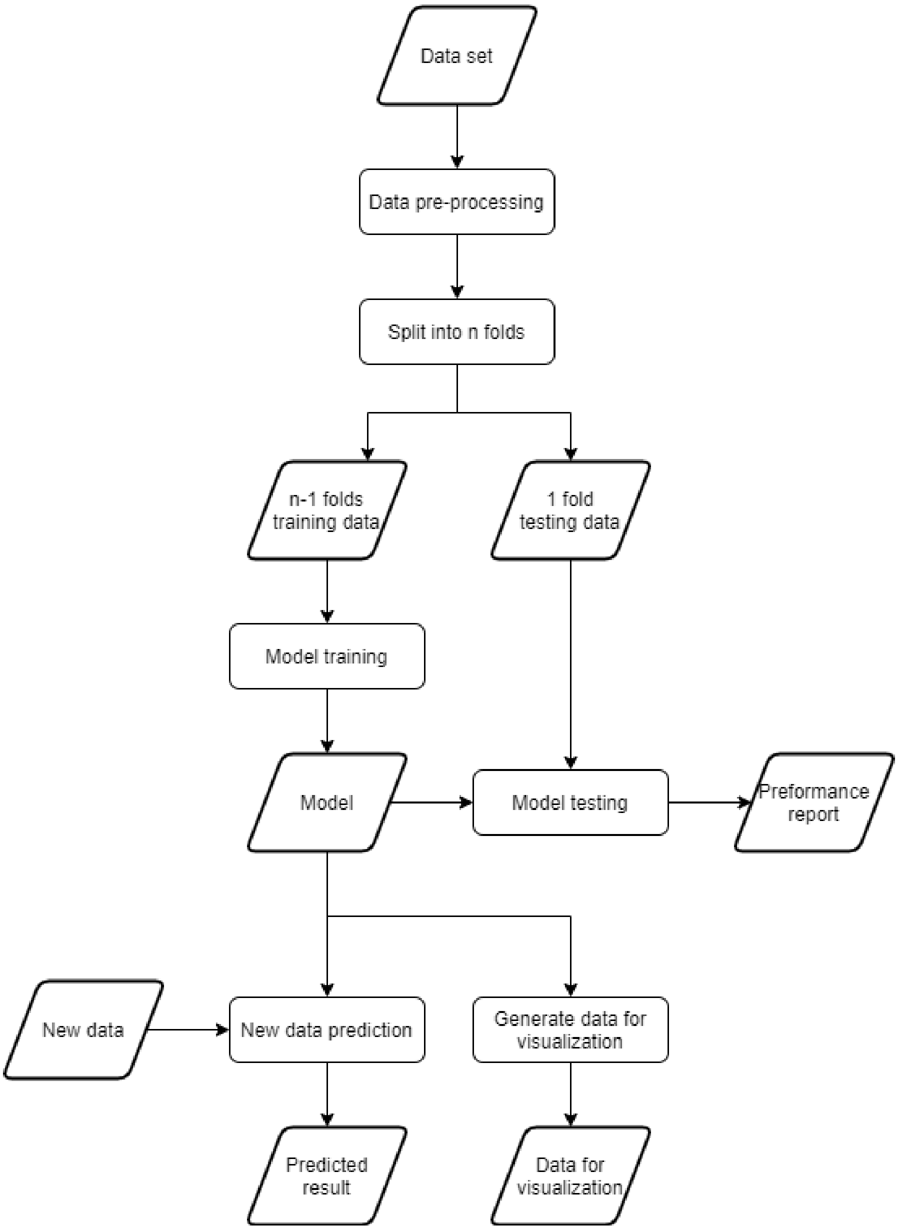


Figure 10: Overall flow of our project

4.1 Data input

Our project is not designed for a specific data set. It should be universal and suitable for different data sets. However, I still need a reliable data set for reference to design and implement our project, as well as for testing and evaluation.

Suggested by one of our supervise Simon, 'HEPARTWO10k' [12] is a suitable data set for us, which is a liver cancer data set. It involves 70 attributes and 10000 rows of sample data. It is professional data set created by some medical experts. It is public that people can do any research based on this data. It is well organized, there is no missing data and strange data type in this dataset. It is quite suitable for us to start a project from the beginning. To facilitate the explanation of my project, this data set will be used as an example to explain the following part in this report.

4.2 Data pre-processing

Data pre-processing is an important step before model training. It may affect the whole performance of the machine learning model. Mentioned in the previous paragraph, our reference data set is well organized, there is no missing data and strange data type in this dataset. I do not need to do anything to pre-process. Remember that our objective is to create universal software which work for different data sets. I much provide a function to check and deal with the missing data, as well as the strange data type.

Beside the missing data and the strange data type, I may also need to consider the imbalanced data set problem. For medical data set, imbalanced data set problem is common, especially for the rare disease. In our 'HEPARTWO10k' [12] dataset, there are totally 10000 rows of record. However, there are only 634 rows of records are diagnosed with cancer, and the reminding 9366 rows of records are diagnosed with no cancer. Too few patient records may affect the performance of the machine learning model.

According to [11], the mainstream solution is to resample the data set by using Synthetic Minority Over-sampling Technique, which also called SMOTE. It is not only duplicating records in the minority class, because those records do not add any new information to the model. New records are synthesized from the existing records. SMOTE can predict new records for the minority class based on the information of existing data. Although SMOTE is an existing algorithm for us to solve imbalanced data set problem, it is also a trade-off. Too much resampling may cause data loss, too little resampling may not solve imbalanced data set problem. Therefore, the parameter setting of SMOTE is very important. I conducted some tests on the parameter settings of SMOTE. For more information on testing, please read testing section in 6.1.

4.3 Model training

After completing all the data pre-processing, the model training part which is the best important can be started. In bnlearn [7] package in R [9], many basic functions for model training have already be provided. Therefore, I use these existing functions in combination with our own code to build our own model training functions.

I use hill climbing algorithm for Bayesian network structure learning. Mentioned in literature review section, hill climbing algorithm is most popular algorithm for

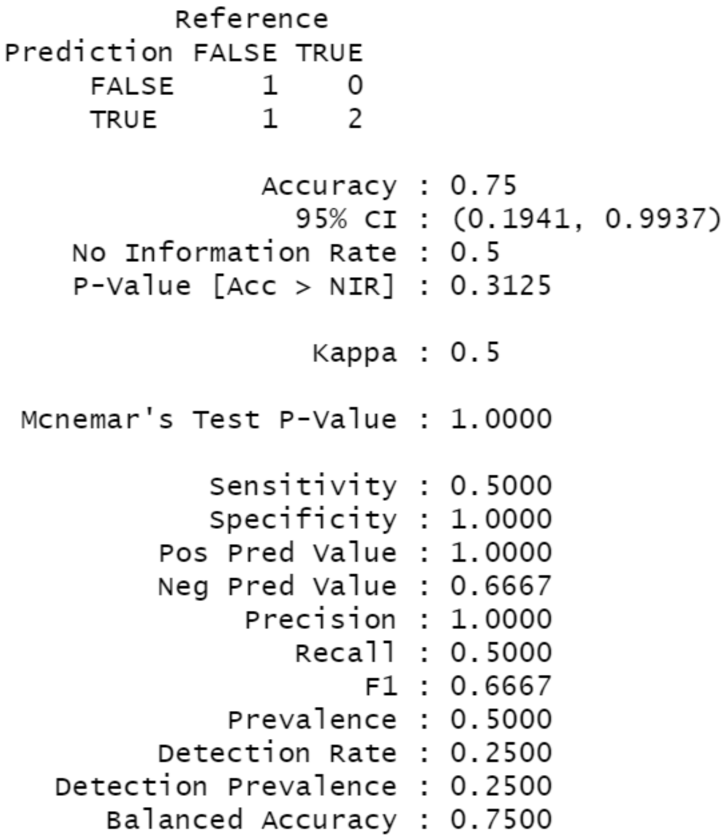
Bayesian network learning because of their good trade-off between computational demands and the quality of the models learned. Hill Climbing is a heuristic search to find out the optimal result, the main idea is to compare the point next to the current point and determine the continuous direction. In our project I use it to learn a network structure to attain relationship of different attribute from the data set.

In order to obtain more stable results, I also incorporate cross-validation in the model training part. Cross-validation splits the data set into n folds. For each iteration, I use n-1 fold for model training, and the remaining 1 fold will be used for the model testing later. Then, n models are generated through cross-validation. To generate the final model, a model averaging method is performed on n models. For example, there are 10 models, 9 of which show that node A is related to node B, and only 1 model shows that node A is not related to node B, so the final average model will show that node A is related to node B.

4.4 Model testing

Since I incorporated cross-validation in the model training part, the training data and testing data are clearly defined in the cross-validation. For each iteration, n-1 fold has already used for model training, and the remaining 1 fold can be used for the model testing now. The testing data can be input into the model. The predicted result can be generated. Then, the performance can be measured of model by comparing the predicted data to the observed data.

However, it is difficult for humans to measure the performance of the model by viewing the list of predicted data and observed data. Thus, I make use of a package ‘caret’ [13]. Within this package, there is a function confusionMatrix(), which can generate a performance report including confusion matrix, accuracy, kappa, F1,etc.



```
          Reference
Prediction FALSE TRUE
     FALSE     1    0
     TRUE      1    2

               Accuracy : 0.75
                 95% CI : (0.1941, 0.9937)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 0.3125

                  Kappa : 0.5

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.5000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.6667
              Precision : 1.0000
                 Recall : 0.5000
                     F1 : 0.6667
             Prevalence : 0.5000
         Detection Rate : 0.2500
   Detection Prevalence : 0.2500
      Balanced Accuracy : 0.7500
```

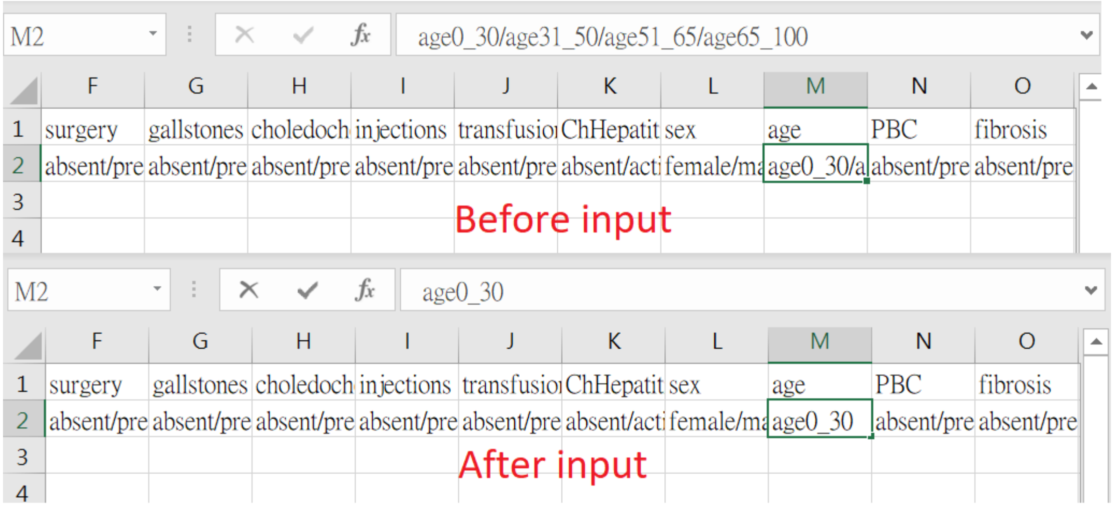
Figure 11: Example of confusionMatrix

4.5 New data prediction

Generally, new data prediction is an essential part for a machine learning project. Since our project forces on generating data for visualization by using machine learning method, rather than a general machine learning project, I may not provide advanced new data prediction function. However, basic new data prediction function is still provided.

Before making a new data prediction, the user is required to input the data that he / she used for prediction. For different data set, its data type may be different. If the user is required to enter data on a blank file without any template, he / she may enter data in the wrong data type. For example, the user enters data for the attribute ‘age’ in the data set ‘HEPARTWO10k’ [12]. User may just enter an integer number. Actually, attribute ‘age’ in this data set is discrete data. The user should enter the age group to which he / she belongs. The input should be ‘age0_30’, ‘age31_50’, ‘age51_65’, or ‘age65_100’.

To avoid user enter data in the wrong data type, an input template should be provided. I wrote a function to generate an input template for user based the original data set automatically. The input template is a csv file. User can edit this file by Excel or other editors. Its columns are the same as the original data set, except for the target attribute, because the target attribute is the data user are about to predict, and it should not be entered. All available input values are displayed in the second row. Using the same example from the previous paragraph, available input values for attribute ‘age’ are ‘age0_30/age31_50/age51_65/age65_100’. If user want to choose age0_30, he / she just need to delete other values in this cell. After input all values, user can go back to our software to open the modified file, software will predict based the existing models.



M2 | age0_30/age31_50/age51_65/age65_100

	F	G	H	I	J	K	L	M	N	O
1	surgery	gallstones	choledoch	injections	transfusio	ChHepatit	sex	age	PBC	fibrosis
2	absent/pre	absent/pre	absent/pre	absent/pre	absent/pre	absent/acti	female/ma	age0_30/a	absent/pre	absent/pre
3										
4										

Before input

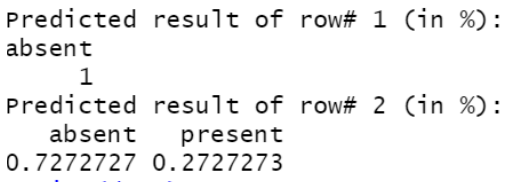
M2 | age0_30

	F	G	H	I	J	K	L	M	N	O
1	surgery	gallstones	choledoch	injections	transfusio	ChHepatit	sex	age	PBC	fibrosis
2	absent/pre	absent/pre	absent/pre	absent/pre	absent/pre	absent/acti	female/ma	age0_30	absent/pre	absent/pre
3										
4										

After input

Figure 12: Example of new data input

Mentioned model training section, there are multiple models since I incorporated cross-validation in 'bnlearn' [7]. Different models may not have different predicted result. If the program just shows all the result predicted by different models, user may have difficulty to understand why there are more than one result. Thus, I combine the results from different model and calculate a percentage to show the result. The following figure show an example of prediction. For this example, predicted result of row# 1 shows that absent:1, it means 100% chance of result is 'absent' value.

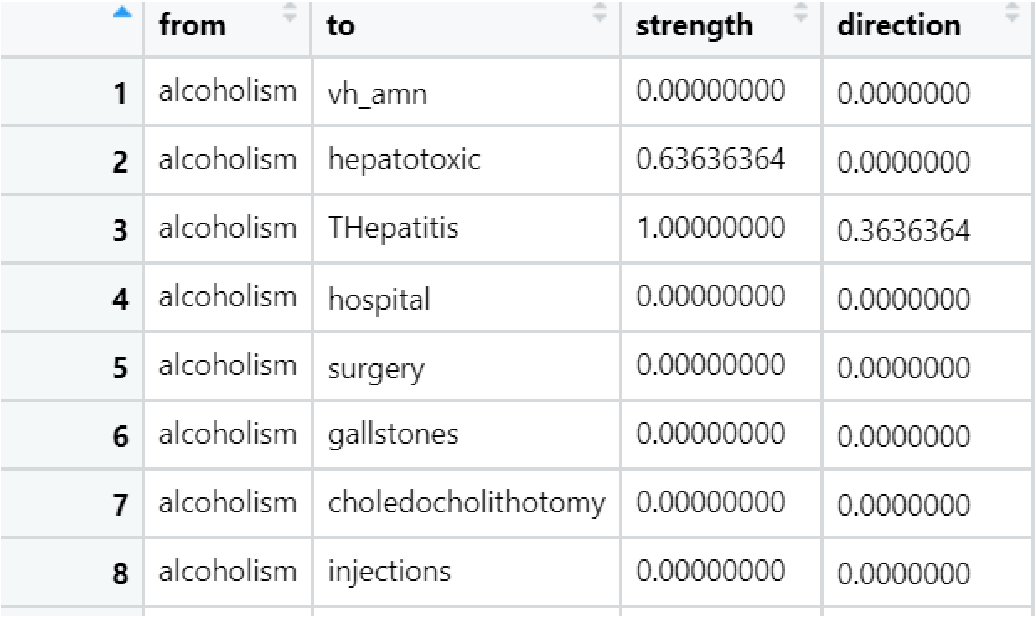


```
Predicted result of row# 1 (in %):
absent
     1
Predicted result of row# 2 (in %):
   absent   present
0.7272727 0.2727273
```

Figure 13: Example of new data prediction result

4.6 Output useful data for visualization

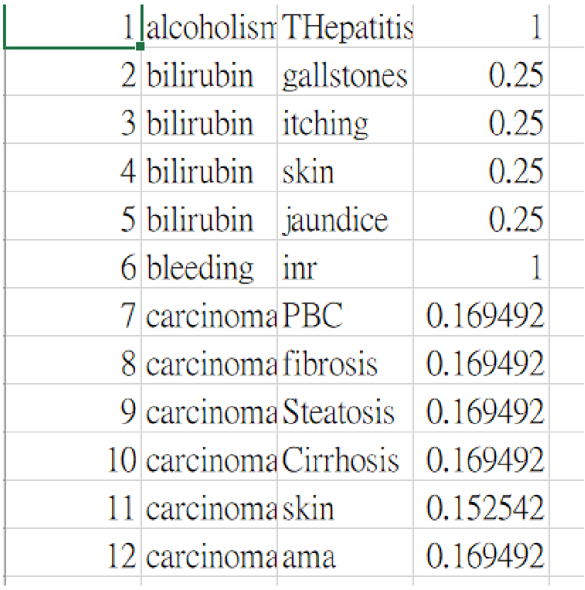
Our project forces on generating data for visualization by using machine learning method. Since our project use Bayesian network model in 'bnlearn' [7], relationship between different nodes is provided in model (see Figure 14). Although it shows the relationship between different nodes, the relationship may be duplicated or too weak. It is just a raw data and cannot be used for visualization directly.



	from	to	strength	direction
1	alcoholism	vh_amn	0.00000000	0.0000000
2	alcoholism	hepatotoxic	0.63636364	0.0000000
3	alcoholism	THepatitis	1.00000000	0.3636364
4	alcoholism	hospital	0.00000000	0.0000000
5	alcoholism	surgery	0.00000000	0.0000000
6	alcoholism	gallstones	0.00000000	0.0000000
7	alcoholism	choledocholithotomy	0.00000000	0.0000000
8	alcoholism	injections	0.00000000	0.0000000

Figure 14: Relationship data in Bayesian network model

Thus, I wrote a function to process this raw data to generate useful data for visualization. This function removes the duplicate or weak relationship. In addition, relationship may cause chicken-and-egg problem. For example, A is the cause of B, B is also the cause of A. It is a chicken-and-egg problem with no solution. Thus, our program also checks the path to avoid it. When this type of relationship is found, it analyses their relationship strength and remove the weaker relationship to avoid this type problem. Finally, the modified data can be used for visualization (see Figure 15).



1	alcoholisn	THepatitis	1
2	bilirubin	gallstones	0.25
3	bilirubin	itching	0.25
4	bilirubin	skin	0.25
5	bilirubin	jaundice	0.25
6	bleeding	inr	1
7	carcinoma	PBC	0.169492
8	carcinoma	fibrosis	0.169492
9	carcinoma	Steatosis	0.169492
10	carcinoma	Cirrhosis	0.169492
11	carcinoma	skin	0.152542
12	carcinoma	ama	0.169492

Figure 15: Example of data for visualization

CHAPTER 5. Implementation

According to the division of the departments in the group, I am mainly engaged in the development of R. Therefore, I implemented my program in R using some existing packages and my own code. Mentioned in CHAPTER 4, my program can be divided into several parts, I will use the similar structure to illustrate my implementation.

5.1 Install and load packages

My program is written in R [9]. Since there are many useful packages for machine learning in R, I make use of them to help me to implement my program. If user's computer is using these packages for the first time, user need to install these packages first. To help users easily install these packages, I wrote a function that contains all the necessary package installations. User only needs to run this function, rather than having to manually install these packages one by one.

```
func.installLibrary = function() {
  install.packages('bnlearn')
  install.packages('caret', dependencies = TRUE)
  install.packages('DMwR')
  install.packages('ROSE')
}
```

Beside packages installation, user is required to load the packages before using the functions provided by these packages. The following function is written by me to help user to load these packages. User just is required to run this function, all required packages will be loaded into R. Someone may notice that some packages is not installed by previous function. The reason is that some packages are built-in R packages, or automatically installed by dependency packages. Thus, they are not required to install again.

```
func.loadLibrary = function() {
  require(bnlearn)
  require(caret)
  require(DMwR)
  require(ROSE)
  require(tcltk)
  require(dplyr)
}
```

5.2 Data input

At the beginning of this program, a data set is required to be inputted to start the learning process. The following function can allow user to select a csv file as an input file, then program will open the file and load data inside this file into a data frame in R. In order to provide a better way for users to choose files, I make use of tclvalue() function from package 'tcltk' (R [9] built-in package). It can pop up a dialog box to open the file, similar to opening files in Word and PowerPoint.

```
func.readData = function() {
  # read csv file
  filename = tclvalue(tkgetOpenFile(title='Open a file', filetypes = '{{CSV File}
{.csv}}'))
  data = read.csv(filename)
  return(data)
}
```

5.3 Data pre-processing

Handling missing data is a common step in data pre-processing. The following function receive two parameters data and replace. Parameter data is the data set to be processed. Parameter is a binary variable that represents how to deal with missing value, either replace missing value with the median value, or delete records that contain missing value.

```
func.checkNA = function(data, replace) {
  if (replace==T) {
    # replace missing data
    for(i in 1:ncol(data)){
      if (is.factor(data[ , i])) {
        # for factor data type
        temp = as.numeric(data[,i])
        data[is.na(data[,i]), i] = levels(data[,i])[median(temp, na.rm = TRUE)]
        data[is.infinite(data[,i]), i]  = levels(data[,i])[median(temp, na.rm = 
TRUE)]
        data[,i] = droplevels(data[,i])
      } else {
        # for numeric data type
        data[is.na(data[,i]), i] = median(data[,i], na.rm = TRUE)
        data[is.infinite(data[,i]), i] = median(data[,i], na.rm = TRUE)
      }
    }
  } else {
    # delete missing data
    data = na.omit(data)
    data = data[apply(data, 1, function(x) all(!is.infinite(x))),]
  }
  return(data)
}
```

Beside missing value, data set may also contain strange data type. The model may not be able to handle strange data types, which may cause model training to fail or create the wrong model. The following function can convert strange data type to be either factor or numeric, which are the most common data type for model training. There is a variable ‘toBeRemove’ in this function. This variable marks attributes which have only one level and deletes them from the dataset because Bayesian network model cannot access an attribute which has only one level. According to the definition of Bayesian network model, each attribute should have at least two levels.

```
func.checkType = function(data) {
  toBeRemove = NULL
  # handle data type
  for(i in 1:ncol(data)){
    if (!is.factor(data[,i])) {
      if (nlevels(factor(data[,i]))<=1) {
        toBeRemove = c(toBeRemove, i)
      } else if (nlevels(factor(data[,i]))<=5) {
        data[,i] = factor(data[,i])
      }
      else {
        data[,i] = as.numeric(data[,i])
        data[,i]  =  cut(data[,i],  breaks=unique(quantile(unique(data[,i]))), 
include.lowest=T)
      }
    }
  }
  data[,toBeRemove] = NULL
  return(data)
}
```

For solving imbalanced data set problem, I apply SMOTE resampling to the data set. Thus, a resampled data set can be used for model training without imbalanced data set problem. Since SMOTE function is provided by package ‘DMwR’ [14], I just use the exiting function rather than writing from scratch. However, this function required user input the over sampling rate and under sampling rate, rather than a number. It is inconvenient for users because it requires some mathematical calculations. Thus, I wrote some code to do calculation according to it formula.

```
overSize = 1000
underSize = 9000
overRate = (overSize-min(table(data[target])))/min(table(data[target]))*100
underRate = underSize/(overSize-min(table(data[target])))*100
data.balanced    =    SMOTE(as.formula(paste(target,'~.')),    data,    k=10,
perc.over=overRate, perc.under=underRate)
```

After resampling, we may want to check whether the data set is correctly resampled to the target size. However, this package does not provide this type of function. Thus, I wrote function that plot a bar chart to show its distribution and it size.

```
func.checkBalance = function(data, target, title) {
  # check balance
  bp = barplot(table(data[, target]))
  text(bp, table(data[, target])/2, table(data[, target]))
  title(title)
}
```

5.4 Model training

For model training function, it requires some parameters, they are data, nFold, algorithm, algorithm.args and target. Parameter data is the data set to be processed. Parameter data nFold is the number of folds of cross-validation. Parameter data algorithm, algorithm.args define algorithm which is used for model training. Parameter target is the targeted attribute in the data set. My program allows user set algorithm by himself / herself rather than a fixed algorithm because I want to increase flexibility of my program. Then, using bn.cv() from package bnlearn [7] to apply model training with cross-validation. It will return a list of items. Each item is one of the training results in each fold. It contains model, training data and testing data.

```
func.learn = function(data, nFold, algorithm, algorithm.args, target) {
  nFold = ifelse(nFold<=0, 1, nFold)
  # do bayesian networks learning and cross-validation
  bn.cv = bn.cv(data, k = nFold,
             bn = algorithm, algorithm.args = algorithm.args,
             loss = "pred", loss.args = list(target = target))
  return(bn.cv)
}
```

5.5 Model testing

To help user test the model, program can generate a performance report based on the cross-validation result. Since 1 fold of testing data is already defined in cross-validation process, program just need to retrieve those data from the result of cross-validation. Because there are more one folds result, I use unlist() and lapply() functions to combine them to be a final result. Finally, using confusionMatrix() from package caret [13] to create a confusion matrix to show its performance.

```
func.performancereport = function(bn) {
  # performance report
  pred = unlist(lapply(bn, `[[`, "predicted"))
  obse = unlist(lapply(bn, `[[`, "observed"))
  pred = factor(pred)
  obse = factor(obse)
  print(confusionMatrix(pred, obse, mode = "everything"))
}
```

Beside the performance testing, ROC AUC value should also be tested because some data set may have the imbalanced problem. The ROC AUC value is a score that can test whether the data set is an imbalanced data set. Using roc.curve() from package ROSE [15] to calculate ROC AUC value. If this value is less than or equal 0.5, it means the data set is imbalanced and program will report an error to ask user to resample it by SMOTE algorithm and run again.

```
func.checkROCAUC = function(bn) {
  pred = unlist(lapply(bn, `[[`, "predicted"))
  obse = unlist(lapply(bn, `[[`, "observed"))
  pred = factor(pred)
  obse = factor(obse)
  rocauc = roc.curve(obse, pred, plotit=F)
  print(rocauc)
  if (rocauc$auc <= 0.5)
    stop('It is an imbalanced dataset, please try to resample it and run again!')
}
```

5.6 New data prediction

When the user wants to use the program to predict new data, he / she is required to input the new data in the same format as the original data. Otherwise, the model may have difficult to read the input. Mentioned in CHAPTER 4, I want to provide an input template in csv format. This function builds an input template according to column and data of the original data set. Then, using tkgetSaveFile and write.table() to ask user to save this template. After user filled in new data, using tkgetOpenFile() and read.csv() to re-open the file. Since cross-validation provides more one models, using lapply(), sapply() and predict() functions to combine models and perform prediction. Finally, using table() to show result in a table format.

```
func.predictNewData = function(data, bn, target) {
  # build input template
  input.template = data.frame(matrix(ncol=ncol(data), nrow = 0))
  colnames(input.template)=colnames(data)
  for(i in 1:ncol(input.template)){
    input.template[1,i] = paste(levels(data[,i]),collapse='/')
  }
  input.template[,target] = NULL
  # save input template
  print('Please save the input template, fill in, and open again!')
  filename = tclvalue(tkgetSaveFile(title='Save a file', filetypes = '{{CSV File}
{.csv}}'))
  if (substr(filename, nchar(filename)-3, nchar(filename)) != '.csv')
    filename = paste(filename, '.csv', sep='')
  write.table(input.template, filename, sep=',', row.names=F, col.names=T)
  # read csv file
  filename = tclvalue(tkgetOpenFile(title='Open a file', filetypes = '{{CSV File}
{.csv}}'))
  newdata = read.csv(filename, stringsAsFactors=F)
  # set proper factor level
  for(i in 1:ncol(newdata)){
    newdata[,i]=factor(newdata[,i],levels = levels(data[,i]))
  }
  # predict the result
  fitted = lapply(bn, `[[`, "fitted")
  pred = sapply(fitted, predict, node=target, data=newdata)
  for(i in 1:nrow(pred)){
    print(table(pred[i,],dnn=paste('Predicted     result     of     row#',i,'(in
%):'))/length(pred[i,]))
  }
}
```

5.7 Output useful data for visualization

Output data for visualization is the most important of my project. At the beginning, using custom.strength() from package bnlearn [7] to retrieve arc relationship in the model. It shows the arc strength between every two attributes, regardless of the arc strength, even 0. Therefore, thresholds should be used to filter less important records. In addition, there may be some contradictory relationships. For example, A is parent of B, B is also parent of A. I wrote the func.adjust() function to identify arc which has potential problem and make adjustment. After that, using group_by() and transmute() functions to calculate the percentage by comparing the arc strength with siblings. Finally, output file is saved as a csv file. It can be passed to visualization program written by my teammate to show the result.

```
func.output = function(bn, target, threshold) {
  arc = custom.strength(bn)
  arc = arc[arc$direction>0.5, c('from', 'to', 'strength')]
  arc = arc[arc$strength>=threshold, ]
  arc$adjusted = F
  arc = func.adjust(arc, target)
  arc       =       group_by(arc,       to)       %>%       transmute(from,
percent=(strength+min(arc$strength))/sum(strength+min(arc$strength)))
  arc = arc[order(arc$to), ]
  rownames(arc) = NULL
  filename = tclvalue(tkgetSaveFile(title='Save a file', filetypes = '{{CSV File}
{.csv}}'))
  if (substr(filename, nchar(filename)-3, nchar(filename)) != '.csv')
    filename = paste(filename, '.csv', sep='')
  write.table(arc, filename, sep=',', row.names=T, col.names=F)
  return(arc)
}

func.adjust = function(arc, parent) {
  index = which(arc$from==parent & arc$adjusted==F)
  if (length(index)>0) {
    arc[index,]$from = arc[index,]$to
    arc[index,]$to = parent
  }
  child = arc[arc$to==parent & arc$adjusted==F,]$from
  arc[arc$to==parent, 'adjusted'] = T
  if (length(child)> 0) {
    for (i in 1:length(child)) {
      arc = func.adjust(arc, child[i])
    }
  }
  return(arc)
}
```

CHAPTER 6. Testing

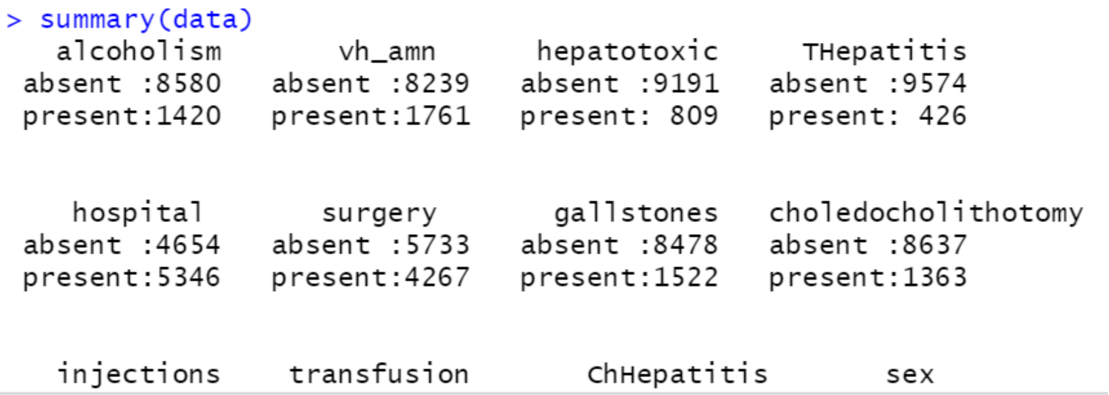
Our project is a machine learning-based project, and the most intuitive forward test method is to test through data. According to the design of the program, the whole testing process can be separated into the following part: data input testing, data pre-processing testing, model testing and visualization testing.

6.1 Data input testing

This part basically tests func.readData() function in the program. This function should popup a dialog to allow user to open a csv file as an input. Then, the program read the file and load the data into the program. To test this function, I try to use this function to open different csv files, those files are different file names, in different folders, different contents. Therefore, use the R [3] built-in function View () to view the contents and check whether the data is loaded into the program correctly.

6.2 Data pre-processing testing

This part mainly tests func.checkNA(), func.checkType() and SMOTE() functions in the program. Since raw data set may include missing data and strange data type which machine learning model cannot handle, func.checkNA() and func.checkType() should detect missing data and strange data type, then remove or convert those data. To test these two functions, I can use the R [3] built-in function summary() to generate a report that it show the distribution of each attribute value (see Figure 16). According to this report, I can simply check whether the missing data and the strange data type exist. For some imbalanced data set, SMOTE() function [14] is required to resample data set. To test this function easily, I wrote a function func.checkBalance() which can plot a bar chart to show the distribution of targeted attribute value (see Figure 17).



```
> summary(data)
   alcoholism       vh_amn       hepatotoxic     THepatitis
 absent :8580   absent :8239   absent :9191   absent :9574
 present:1420   present:1761   present: 809   present: 426


    hospital        surgery        gallstones    choledocholithotomy
 absent :4654   absent :5733   absent :8478   absent :8637
 present:5346   present:4267   present:1522   present:1363


   injections     transfusion        ChHepatitis        sex
```

Figure 16: Using summary() to test data pre-processing part

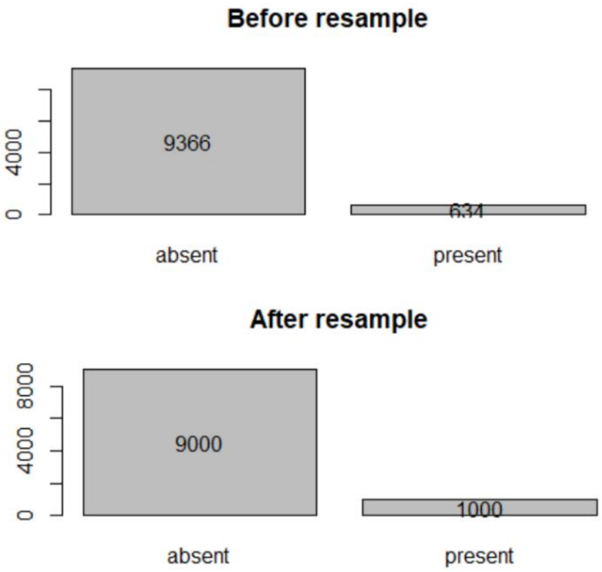
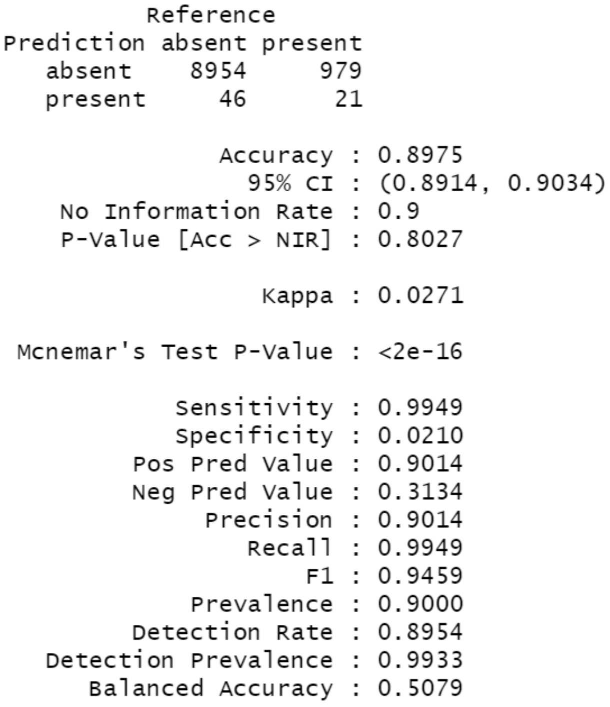


Figure 17 :Using bar chart to test data resampling part

6.3 Model testing

Since I incorporated cross-validation in the model training part, the training data and testing data are clearly defined in the cross-validation. Thus, there is 1 fold can be used for the model testing now. If I input this testing data into the model, the predicted result will be generated. Then, I can test the model by comparing the predicted data to the observed data. I can use function confusionMatrix() from package 'caret' [13] to generate a performance report including confusion matrix, accuracy, kappa, F1,etc. Thus, I can measure the its performance by reading this report (see Figure 18).



```
          Reference
Prediction absent present
   absent    8954     979
   present     46      21

               Accuracy : 0.8975
                 95% CI : (0.8914, 0.9034)
    No Information Rate : 0.9
    P-Value [Acc > NIR] : 0.8027

                  Kappa : 0.0271

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9949
            Specificity : 0.0210
         Pos Pred Value : 0.9014
         Neg Pred Value : 0.3134
              Precision : 0.9014
                 Recall : 0.9949
                     F1 : 0.9459
             Prevalence : 0.9000
         Detection Rate : 0.8954
   Detection Prevalence : 0.9933
      Balanced Accuracy : 0.5079
```

Figure 18: Using performance report to test the model

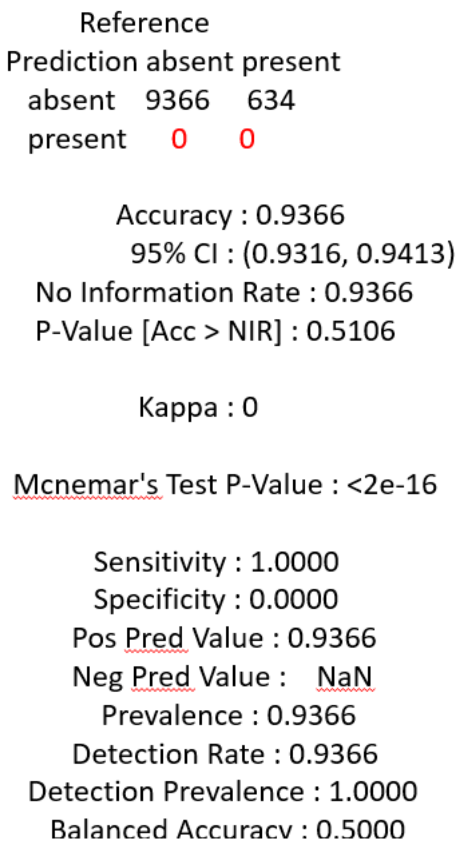
6.4 Visualization testing

The program only uses machine learning methods to generate data for visualization, and the visualization is done by another teammate. This section is only a simple test to check output file is generated correctly. For deeper visualization testing, please read the report written by my teammate.

CHAPTER 7. Evaluation

7.1 Imbalanced data set problem

Mentioned in CHAPTER 4, imbalanced data set problem is a common for medical data. If I ignore this problem and use an imbalanced data set to train a model, I will get a non-reliable result. For example, I use ‘HEPARTWO10k’ [12] dataset to train a model without any resampling. Looking at the result displayed in the following figure, accuracy is 93.66%, it looks good. However, there is a serious problem. Looking at the confusion matrix, there is 0 true negative and 0 false negative. It means that model never predict minority class ‘present’. Imagine that someone always tells you that you do not have the disease, whether you have the disease. It does not provide any information to help diagnose the disease. This kind of result is not our objective. To provide a reliable result, imbalanced data set problem should be solved. Thus, solving imbalanced data set problem is main topic in this evaluation section.



```
          Reference
Prediction absent present
   absent   9366     634
   present     0       0

               Accuracy : 0.9366
                 95% CI : (0.9316, 0.9413)
    No Information Rate : 0.9366
    P-Value [Acc > NIR] : 0.5106

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.9366
         Neg Pred Value :    NaN
             Prevalence : 0.9366
         Detection Rate : 0.9366
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000
```

Figure 19: Performance report of model trained by imbalanced data set

7.2 SMOTE resampling method

In the SMOTE [14] resampling method, there are parameters: k, perc.over and perc.under. K is the number of nearest neighbours that are used to generate the new examples of the minority class. Perc.over drives the decision of how many extra cases from the minority class are generated. Prec.under the decision of how many cases from the majority class are selected. I try to apply SMOTE resampling to the data set and compare evaluation the result. For comparison, I list a table to show the original result and result after resampling.

Table 4: Comparison different between original and resampled data set

	Original data set	After SMOTE resampling
Size of two classes	absent: 9366 present: 634	absent: 9000 present: 1000
Confusion matrix	9366 634 0 0	8954 979 46 21
Accuracy	0.9366	0.8975
Kappa	0	0.0271
Sensitivity	1.0000	0.9949
Specificity	0.0000	0.0210
Pos Pred Value	0.9366	0.9001
Neg Pred Value	NaN	0.1538
F1	0.9366	0.9459
Balanced accuracy	0.5000	0.5079

According to the table, kappa, specificity and neg pred value change the most. Originally, they are 0, 0 and Nan. After SMOTE resampling, they become some regular value. Those value are not high. However, it is an inevitable problem for imbalance data set. I can generate more minority class to improve those value. However, generating more minority class means there are means there are more artificial data and less real data in the dataset. If I just train a lot of artificial data rather than real data, it may not useful for real medical diagnoses, it also not our project objective.

CHAPTER 8. Discussion

For this project, the objective is visualizing the medical data by machine learning method to help medical diagnose. My teammates and I work hard on this project. We not only want to complete the final year project, but also want to contribute to society. I know that this objective is meaningful for human health. However, I know that my project is not perfect because I have insufficient knowledge of applied machine learning and medicine, and I am affected by the suspension of classes.

8.1 What I am satisfied

In my opinion, I am satisfied with the part of outputting data for visualization. I think this part is the most special function in my project. Although there are many similar machine learning projects done by other students, most of them is just black-box process, their project just accepts user input and predict the result, user never knows how the predicted results are generated. My project not only provides the predicted results, but also generate data for visualization. The visualization is done by my teammate. Cooperate with his program, we can show a graph to user. Thus that, user can figure out the relationship between each attribute and understand how the predicted results are generated. Users can get more information through our project, not just know the prediction results.

8.2 What I can do better

I know that the meeting schedule of our group in the second semester is poor. Mentioned in 2.7, meeting schedule is affected by the suspension of classes. We cannot have a physical meeting on campus. Although we try to use Zoom to do an online video meeting, it is not successful because of the computer equipment and network connection problem. After that, we just use text message in WeChat to communicate. I know text message in WeChat is not formal meeting style. Even this type of communication may not be considered a meeting because we send messages from time to time and it is difficult to record meeting logs. However, we have no other better solution during the suspension of classes because of the computer equipment and network connection problem. I understand that this type of meeting may conflict with the original meeting requirement. I hope the coordinator can understand our situation and accept our special meeting schedule.

Beside the meeting schedule, I think I also should do better on the testing different data set. My original goal of this project is to provide a software that have the ability to provide visual results for every medical data set. To achieve this goal, I should test more and more different data sets. This type of testing can help me to which type of data set my program can handle, which type of data set my program cannot handle. Thus, I can improve my program based on the test result. Unfortunately, my progress was delayed by the suspension of class. I do not have enough time to do it before the submission day. Thus, I cannot ensure that my program can handle every data set.

CHAPTER 9. Ethics and Professional Conduct

Since the subject of my project is related to medical data, medical knowledge is required to provide complete software. However, I am just a computer science student without any medical background. Because of my lack of medical knowledge, my project may not have some potential problem, and raise some issues about the ethics and professional conduct. The following potential problems will occur with high probability.

9.1 Impact of false positive diagnosis

My project can provide accurate predictions for medical diagnosis through machine learning methods. For example, the accuracy of the "HEPARTWO10k" dataset is as high as 89.75%. This is already a good performance for general machine learning project. However, accuracy is very important in medical diagnosis. Any wrong medical diagnosis may damage human health. False positive diagnosis means that someone has been diagnosed as a patient, but in fact he / she is healthy. For this type of diagnostic error, he / she received treatment by mistake. Therefore, he / she may be affected by the negative effects of treatment. Especially for cancer, chemotherapy may be required, which may cause side effects such as hair loss, weakness, and vomiting.

9.2 Impact of false negative diagnosis

In addition to false positive diagnosis, there is false negative diagnosis. False negative diagnosis means that someone has been diagnosed as healthy, but in fact he / she is a patient. For this type of diagnostic error, his / her treatment may be delayed. For most diseases, early diagnosis has a greater chance of recovery. If the patient is diagnosed with the disease too late, his / her disease may become more and more serious. He / she may not be cured due to delayed treatment.

For these reasons, I mentioned at the beginning of this report that my project is just a tool to associate a doctor to make medical diagnosis. People should not only rely on our projects for medical diagnosis without any professional medical knowledge. Although I have mentioned this concept many times in this report, I think many people may still misunderstand my project, and they may think that the software can make medical diagnosis without any professional medical knowledge.

CHAPTER 10. Conclusions

In conclusion, my project focuses on visualization of medical data. At the beginning, I do not have any relative knowledge. However, I worked hard on many related projects, and with the help of superiors, I gradually understand more relative knowledge. In addition, my supervisor and I found many useful tools to help me complete the project. These tools have benefited me a lot. They provide many useful functions, so I integrated them into the project to achieve more functions. Based on the help of the above people and tools, I finally completed my project.

10.1 Major accomplishments

My final product can analysis the medical data by using machine learning method. Similar to other machine learning projects, it can perform model training, perform test, new data prediction. However, these functions are not main objective of my project. My main objective is to provide a visual result.

According to the division of labour, my write the program can generate the useful data for visualization, and my teammate work on visualization. My program can input raw medical data set. Then, it can perform data pre-processing, model training, retrieve the relationship between each attribute. The relationship data is processed by my program and generate useful data for visualization. It can be passed to my teammate’s program to provide visual result. Through the cooperation of team members, we successfully completed the goal of providing users with visual results of medical data analysis.

10.2 Future works

My initial goal was to provide software that can provide visual results for any medical data set. Thus, I should test more different data sets as many as possible to ensure my program handle different data sets with different structure and data type. Due to suspension of classes, some progress was delayed. I do not have enough time to do it before the submission day. If I a change to continue work on this project, I will test more data sets. Based on the test results, I continue to improve my program. My ultimate goal is to handle every data set automatically without modifying or adjusting manually.

CHAPTER 11. References

[1] D. Curiac, G. Vasile, O. Banias, C. Volosencu and A. Albu, "Bayesian network model for diagnosis of psychiatric diseases," 2009. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5196055

[2] C. Anthony, F. Norman, M. William and R. Lukasz, "From complex questionnaire and interviewing data to intelligent Bayesian network models for medical decision support," February 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S093336571600004X

[3] R. Nagarajan, M. Scutari and S. Lèbre, *Bayesian Networks in R with Applications in Systems Biology*, 2013.

[4] M. Scutari and J.-B. Denis, *Bayesian Networks with Examples in R*, 2014.

[5] U. Laura, "Advantages and challenges of Bayesian networks in environmental modelling," May 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304380006006089

[6] A. Peter, F. Geert, T. Dirk, M. Yves, M. Bart De, "Bayesian applications of belief networks and multilayer perceptrons for ovarian tumor classification with rejection," 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0933365703000538

[7] bnlearn, "bnlearn - an R package for Bayesian network learning and inference". [Online]. Available: http://bnlearn.com/

[8] J. Gámez, J. Mateo, J. Puerta, "Learning Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighbourhood, " 2001. [Online]. Available: https://link.springer.com/article/10.1007/s10618-010-0178-6

[9] R, "The R Project for Statistical Computing". [Online]. Available: https://www.r-project.org/

[10] S, Marco, "Learning Bayesian Networks with the bnlearn R Package," 2009. [Online]. Available: https://arxiv.org/abs/0908.3817

[11] B. Jason, "SMOTE for Imbalanced Classification with Python," 2020. [Online]. Available: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[12] A. Onisko. *Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders*, March 2003.

[13] caret, "The caret Package". [Online]. Available: https://topepo.github.io/caret/

[14] DMwR, "Functions and data for "Data Mining with R"". [Online]. Available: https://cran.r-project.org/web/packages/DMwR/index.html

[15] ROSE, "Random Over-Sampling Examples". [Online]. Available: https://cran.r-project.org/web/packages/ROSE/index.html

[16] National Center for Advancing Translational Sciences, "Tox21 Data Challenge 2014". [Online]. Available: https://tripod.nih.gov/tox21/challenge/

[17] A. Al-Hajji, F. AlSuhaibani, N. Alharbi, "Online Knowledge-Based Expert System (KBES) for Psychological Diseases Diagnosis," 2019. [Online]. Available: https://www.semanticscholar.org/paper/Online-Knowledge-Based-Expert-System-(KBES)-for-Al-Hajji-AlSuhaibani/10382c6f4596e98fe80aa35efe105fd07a16bca6

[18] J. Obi, A. Imianvan, "Interactive Neuro-Fuzzy Expert System for Diagnosis of Leukemia," 2011. [Online]. Available: https://www.semanticscholar.org/paper/INTERACTIVE-NEURO-FUZZY-EXPERT-SYSTEM-FOR-DIAGNOSIS-Obi-Imianvan/9fa1a9bbebc701e2e055529428469d633c5ad181