PREDICTION OF HEART DISEASE USING BAYESIAN NETWORK MODEL

A Thesis Presented to the Department of

Computer Science

African University of Science and Technology

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

By

Muibideen Mistura Adebimpe

Abuja, Nigeria

June, 2019

CERTIFICATION

This is to certify that the thesis titled “PREDICTION OF HEART DISEASE USING BAYESIAN NETWORK MODEL” submitted to the School of postgraduate studies, African University of Science and Technology (AUST), Abuja, Nigeria for the award of the Master's degree is a record of original research carried out by Muibideen Mistura Adebimpe in the Department of Computer Science.

PREDICTION OF HEART DISEASE USING BAYESIAN NETWORK MODEL

By

Muibideen Mistura Adebimpe

A THESIS APPROVED BY THE COMPUTER SCIENCE DEPARTMENT

RECOMMENDED: _____

Supervisor, Dr. Rajesh Prasad

_____

Head, Department of Computer Science

APPROVED: _____

Chief Academic Officer

_____

Date

ABSTRACT

The Heart Disease according to the survey is the leading cause of death all over the world. The health sector has a lot of data, but unfortunately, these data are not well utilized. This is as a result of lack of effective analysis tools to discover salient trends in data. Data Mining can help to retrieve valuable knowledge from available data. It helps to train model to predict patients' health which will be faster compared to clinical experimentation. A lot of research has been carried out using the Cleveland heart datasets. Different Implementation of machine learning algorithms such as K-Nearest Neighbor, Support Vector Machine, Logistic Regression, Naïve Bayes, etc. have been applied but there has been limit to modeling using Bayesian Belief Network. This research tackles this drawback. This research applied Bayesian network (BN) modeling to discover the relationship between the 14 relevant attributes of the Cleveland heart data set from University of California, Irvine. The BN produce a reliable and transparent graphical representation between the attributes with the ability to predict new scenarios which makes it an artificial intelligent tool. The model has an accuracy of 85%, precision of 86%, recall of 85% and f1-score of 85%. It was concluded that the model outperformed Naïve Bayes classifier which have accuracy of 80%, precision of 81%, recall of 80% and f1-score of 80%.

Abstract: Naïve Bayes Classifier, Bayesian network, machine learning, data mining, artificial intelligence

DEDICATION

I dedicate this thesis to my mother, Alhaja Wulemot Muyibi, for all her encouragement to push through with this degree. May Almighty Allah in his infinite mercy reward you in this life and in the hereafter (Amin).

ACKNOWLEDGEMENT

All praise belongs to Almighty Allah, the most beneficent, the most merciful for sparing my life and giving me the wisdom, and good health to complete this program.

My deepest gratitude goes to my supervisor Dr. Rajesh Prasad whose guidance and encouragement has made this thesis possible. Thanks for your suggestions and motivations to see research as an adventure. May the good God reward you sir.

My appreciation goes to all my amiable faculties especially Professor Amos David, Professor Lehel Csato, Professor Mohamed Hamada, and Dr. Victor Odumuyiwa. God bless you all for the knowledge you impacted on me.

I want to use this opportunity to appreciate the African Development Bank (AfDB) for considering me worthy of a scholarship for my Master of Science degree program. I promise to be a good ambassador of the society and give back to Africa.

To my husband, Mallam Ibrahim Tijani, my parent, in-laws and my wonderful brothers and sister, I do appreciate all your prayers and support. May Almighty Allah reward you abundantly.

I wish to thank my friends and senior colleagues: Zulihat Hassan, Abdul Rasheed Rukoyah, Odedina Omolade, Akubo Patricia, Shettima Nafisah, Maduakor Francis, Hamid Lawal, Ismail Lukman, Mohammed Habib, for their support, words of encouragement and prayers throughout the duration of my program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 Research Background

The heart is a vital organ in the human body. It is responsible for pumping blood through the blood vessels of the circulatory system. The blood helps to convey oxygen which is needed for the functioning of the body cells. The heart beats for about 100,000 times per day. Heart diseases are also called cardiovascular diseases (CVDs). Heart diseases happen to be the most common cause of death globally. According to WHO, both men and women are equally affected by heart disease. WHO estimated that 17.9 million people are dead due to heart disease in 2016 which represent 31% of all global deaths. 85% of these deaths are caused by stroke and heart attack (WHO, 2016).

Cardiovascular diseases result when the heart and blood vessels are not working normally. Other problems do exist along with the cardiovascular disease. Arteriosclerosis which generally means hardening of arteries, the arteries, in this case, becomes thicker and inflexible. Atherosclerosis means narrowing of arteries, so less blood flow through the buildups (Varun, Mounika, Sahoo, & Eswaran, 2019). Heart attacks occur generally when the blood clots or there is a blockage to blood flow from the heart.

To buttress the importance of overcoming deaths of cardiovascular diseases, WHO launched a new program on 22nd September 2016 called the Global Hearts (WHO, 2017).

Some factors that tend to prone heart diseases are smoking, high cholesterol, high blood pressure, physical inactivity, unhealthy diet, obesity, and poorly controlled diabetes. Diagnosis of heart disease is usually done by taking of medical history, the use of a stethoscope, Ultrasound, and ECG.

Data mining helps to identify useful trends in a large set of data. As a result of the increase in the amount of health data gathered through the electronic health record (EHR) systems, it is believed that strong analysis tools are important. With a huge amount of data, health care providers are now optimizing the efficiency of their organization using data mining. Data mining has helped the health care industry to specifically reduce costs by increasing efficiencies, improving patient's quality of life, and most importantly saving the lives of more patients. In healthcare, data mining has proven effective in areas such as predictive medicine, customer relationship management, detection of fraud and abuse, management of healthcare and measuring the effectiveness of certain treatments (USF, 2019). Data mining can be applied to health data for many different purposes and investigations. These applications can roughly be grouped into the four main categories as discussed below (Tekieh & Raahemi, 2015).

Clinical Decision Making

Patients are normally examined by clinicians to diagnose their diseases. This process is experimental in nature and there is a possibility of the diagnosis being wrong. Data mining gives the experts in the field a second opinion for most diagnoses, especially to make sure the disease is not under-estimated during diagnosis. This information can help the clinicians to make more accurate decisions. It also helps the providers to deliver higher quality services.

Biomedicine and Genetics

This is another application of data mining in health care. Some diseases are studied in the biomedical and molecular level in addition to the clinical level. The effects of genetics on different diseases in micro-level can be investigated as the amount of retrieved biomedical data is increasing. In microarray data analysis, clustering techniques have received more attention compared to classification and association as there is not a lot of information available about genes, in contrast to health conditions and disease symptoms that a lot of information is known (Yoo, et al., 2011).

Population Health

Epidemiologists and other health analysts focused on the prevalence of diseases are interested in identifying the patterns, trends, and causes of spreading a specific disease across a population. For these studies, they consider different risk factors and health determinant, including early-life, lifestyle, and socio-demographic (Tekieh & Raahemi, 2015).

Health Administration and Policies:

Handling insurance plans is a big challenge in health administration. There is always a problem of insurance fraud. Data mining has been applied to detect insurance fraud in which the doctors, patients or hospitals claim drugs that were not necessary or procedures that did not actually happen. This can lead the insurance company to bankruptcy. The solution to this is a built predictive model that is real-time that can help to detect what type of drug is necessary for every diagnosis.

1.2 Problem Statement

As a result of some risks identified with clinical treatments such as the delay in the result and the non-availability of the medical facilities to the people, the prediction model is recommended. Although prediction model is not alternative to clinical treatments, but it can serve as first hand tool to be aware of any type of disease and be prepared for it.

1.3 Research Aim and Objectives

The aim of this research is to build a probabilistic graphical model- Bayesian Network to understand the relationship between the attributes of Cleveland heart disease dataset. Given the aim of this research, the objectives to achieve the aim are:

i. To transform the Cleveland dataset to a form suitable to model a probabilistic graphical model.
ii. To learn the structure and parameter of the model from the dataset.
iii. To make inference from the constructed model.
iv. To compare the model with Naïve Bayes algorithm.

1.4 Expected Contributions

It is expected that the Bayesian network model will assist in making inference about heart diseases, thereby serving as a diagnostic tool to support the medical practitioners.

1.5 Thesis Structure

The thesis contains five basic chapters.

Chapter 1 discusses the introductory part of heart diseases, problem statement, aim, objectives, the expected contribution and the thesis structure.

Chapter 2 gives an insight into an overview of machine learning, Bayesian Network, and critically review the literature.

Chapter 3 discusses the research methodology used as well as the network design. The Cleveland Heart Disease Dataset was also described, the data preprocessing steps and the tools used for the study.

Chapter 4 provides a detailed discussion on the results and system implementation.

Chapter 5 rounds off the research by giving the conclusion.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter discusses some basic concepts and terminologies such as Machine Learning, Classification and Bayesian Network. Related and existing works on heart disease prediction using different machine learning techniques such as Naïve Bayes, KNN, Decision Tree, SVM, etc. are reviewed, thereby looking at what was done, how it has been done, the classification technique used, the data set used for the implementation, the tools used and the result and accuracy of the system.

2.2 Machine Learning

The field of Machine Learning has been in existence since 1959. Arthur Samuel while working for IBM defined Machine Learning as a field of study that enables the computer to learn without being explicitly programmed.

A formal definition of ML was proposed by Tom Mitchell (1998) using a well-posed learning problem, stating that A is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T as measured by P, improves with experience E(Mitchell, 2006).

To relate this definition to this research, we want to develop a heart disease prediction system. The task T of this system is to predict the occurrence of heart disease. The performance measure P is the prediction accuracy of our model. The system learns if we have more clinical

data on heart disease status in patients. Here the experience E refers to the set of already processed clinical datasets. Hence as more records of data are added to the system, we achieve a higher precision as regards to its accuracy.

Machine Learning is a recurrent area of Artificial Intelligence (AI) with active research and applications in the past decades. Self-driven cars, speech recognition, robotic controls, effective web search, face detection are only a few of the areas where Machine Learning is being used.

There are four types of machine learning Algorithm:

1. Supervised Learning
2. Unsupervised Learning
3. Semi-Supervised
4. Reinforcement Learning.

2.2.1 Supervised Learning

Supervised learning refers to the training of data sample from a data source and then using a test dataset also derived from the data sample to forecast or predict (Sathya & Abraham, 2013). We have two supervised learning: Regression models and Classification models.

Regression

Regression model tries to predict the result in a continuous (real-valued) output. An example of a regression problem is trying to predict the price of a given house based on a number of parameters.

Classification

A classifier model tries to map an input space onto discrete output. An example of classification is a prediction of the presence of heart disease in a patient (Witten & Frank, Practical Machine Learning Tools and Techniques., 2005).

2.2.2 Unsupervised Learning

In unsupervised learning, the goal is to find hidden structure in unlabeled data which means finding which example are similar to each other, and then bringing them together as a cluster. The lack of direction for the learning algorithms in unsupervised learning can sometimes be gainful since it enables the algorithm to look back for patterns that have not been earlier considered (Kohonen, Oja, Simula, & Visa, 1996).

2.2.3 Semi-Supervised Learning

Semi-supervised learning is similar to supervised except that it uses both labeled and unlabeled data. It uses a small amount of labeled data with a large amount of unlabeled data due to the fact that unlabeled data is less expensive and takes less effort to acquire. This type of learning can be used with methods such as classification, regression and prediction. Semi-supervised learning is useful when the cost associated with labeling is too high to allow for a fully labeled training process. Early examples of this include identifying a person's face on a webcam.

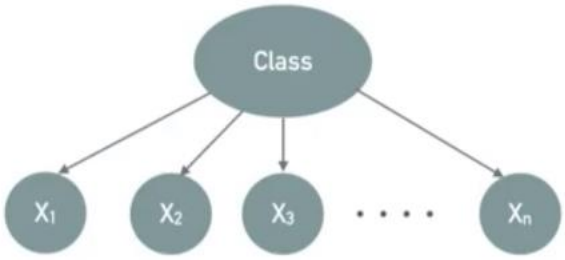
2.2.4 Reinforcement Learning

Reinforcement learning occurs when an agent learns through trial and error interactions with a dynamic environment (Giryes & Elad, 2011). It is closely related to the fields of decision theory in statistics, game theory and control theory in Engineering. The machine’s goal is to learn to act

in a way that minimizes the punishments or maximizes the future rewards it receives over its lifetime.

2.3 Naïve Bayes

Naïve Bayes is the most commonly used Bayesian network. There is a very strong independence assumption in a naïve Bayes. All the random variables are independent of each other given the class. Structure of Naïve Bayes is depicted in Figure 2.1.



$$(X_i \perp X_j \mid Class) \quad \forall \quad X_i, X_j$$

$$P(C, X_1, \ldots . X_n) = P(C) \prod_{i=1}^{n} P(X_i \mid C)$$

Figure 2. 1. Naïve Bayes Structure

Bayes theorem gives a way of calculating posterior probability P(c|x), from the prior probability of class P(c), prior probability of predictor P(x) and the likelihood which is the probability of predictor given class P(x|c).

$$P(C|\mathrm{X}) = \frac{P(X|\mathrm{C})\,P(C)}{P(X)}$$

2.4 Bayesian Belief Network

Bayesian networks are the best-known classifier that is able to provide the probability distribution concisely and comprehensibly (Witten & Frank, 2005).

Bayesian Network Structure

It is a type of probabilistic graphical model. It is a directed acyclic graph that consists of nodes and edges. The nodes represent the random variables while the edges represent the causality (Spirites, Glymour, & Scheines, 2001).

Each node has a conditional probability distribution (CPD) that shows the relationship of a node with its parents.

The joint probability distribution factorized as the product of several conditional distribution denotes the dependency/ independency structure by a direct acyclic graph (DAG).

$$P(\mathrm{X1},\ldots,\mathrm{Xn}) = \prod_{i=1}^{n} P(X1|\ \mathrm{Pa(Xi)})$$

where Pa (Xi) denotes the parent nodes of *Xi*. This equation is also called the *chain rule* Bayesian Networks (Parra, et al., 2015). An example structure of Bayesian belief network is as shown in Figure 2.2.

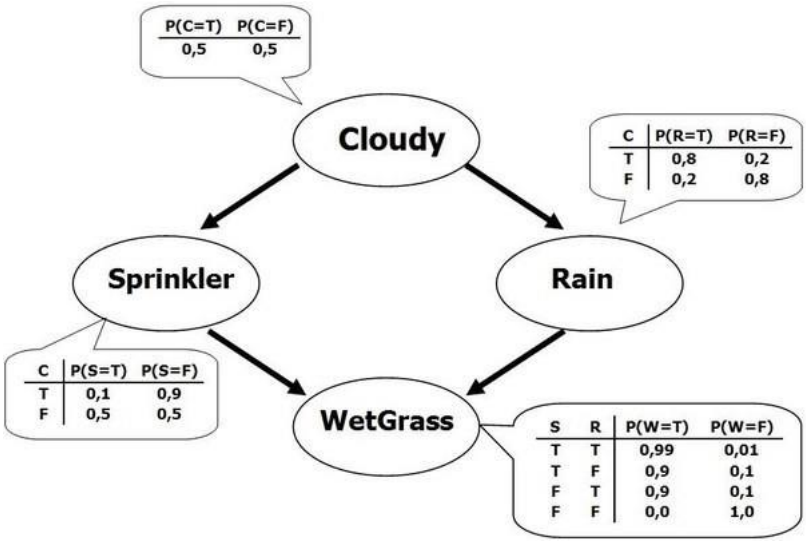


Figure 2. 2. A Bayesian Network Structure

The conditional independence assumption made by naïve Bayes is sometimes rigid, especially in situations where the attributes are correlated. Hence the need for a more flexible way of modeling- Bayesian Belief Network. In BBN, we specify which pair of the attribute are conditionally independent. Bayesian network is an established framework for uncertainty management in Artificial Intelligence, and it uses graph theory and probability theory to represent the relationship between nodes. BN modeling has its origin within data mining and machine learning and captures probabilistic influences induced out of big data sets. They constitute a strong knowledge representation and an efficient tool under uncertainty condition (Parra, et al., 2015).

2.4.1 Some Basic Definition in BB Network

BLOCKED:

A path between two vertices X and Y in a BN is blocked if it passes through a vertex Z such that either:

I. The connection is serial ((X→Z→Y) or (X←Z ←Y)) or diverging (X ← Z→Y) and Z is conditioned on
II. The connection is converging (X→Z←Y) and neither Z or any of its descendant have received influence

D-separation:

X and Y are d-separated by Z if all the paths from a vertex of X to a vertex of Y are blocked. If X and Y are d-separated by Z, then X is independent of Y given Z ($X \perp Y \mid Z$).

Active Trail:

If there is a flow of influence from a node X to Y through a node Z, then it is said that the trail is active. A causal trail (X→Z→Y), an evidential trail (X←Z ←Y), or a common causal trail (X ← Z→Y) is active if and only if Z is not observed. A common effect trail (X→Z←Y) is active if and only if either *Z* or one of *Z*'s descendants is observed

Markov's Blanket:

The Markov property states that any node N is conditionally independent of any other node given its Markov blanket. A node's Markov blanket includes all its parents, children and children's parents. Therefore, if a node is absent from the class attribute's Markov blanket, its value is totally irrelevant to the classification (Elsayad & Fakhr, 2015).

Any node in the BN would be *d-separated* of the nodes belonging to the non-Markov blanket given its Markov blanket.

Learning Bayesian Network

There are two major tasks in learning Bayesian network:

1. Structural Learning
2. Parametric Learning

Structure Learning

This is the identification of the topology of the BN. Three popular structure learning algorithms are discussed below.

Constraint-based algorithms: This involves analyzing the probabilistic relations using the Markov property of BN with conditional independence tests. This can result in causal models even when learned from observational data (Pearl, 1998). In this algorithm, an undirected graph could be generated and can be transformed into a BN using additional independence test (Parra, et al., 2015).

Score-based algorithms: Algorithms assigns a number(score) to each candidate Bayesian network and then maximize it with some search algorithm choosing the model with the highest score.

Hybrid algorithms: This combines both the constraint-based and the score-based, they leverage on conditional independence test to minimize the search space and the network score to find the optimal network.

Parametric Learning:

This is the calculation of the conditional probabilities in a network topology. Parameter learning is of two main types:

Maximum Likelihood Estimation:

This is the natural estimate for the CPDS. It simply uses the relative frequencies with which the variable states have occurred. According to MLE, we should fill the CPDs such that P(data | model) is maximized. This is achieved when using relative frequencies (Koller., 2009).

Bayesian Estimation:

The Bayesian Parameter estimator starts with the already existing prior conditional probability tables that express our beliefs about the variables before the data was observed.

2.5 Application of Bayesian Network Model.

Typical use of Bayesian networks includes to model and explain a domain, to update beliefs about states of certain variables when some other variables were observed, i.e., computing conditional probability distributions, e.g, P(X2|X3= yes, X5= no) , to find most probable configurations of variables , to support decision making under uncertainty , to find good strategies for solving tasks in a domain with uncertainty (Vomlel, 2019).

The Best ten real-world applications of Bayesian Network is in different domains such as Gene Regulatory Networks, System Biology, Turbo Code, Spam Filter, Image Processing, Semantic

Search, Medicine, Biomonitoring, Document Classification, Information Retrieval etc. (Data-flair, 2019) .

2.6 Some Programming Modules for Bayesian Network Programming

R and Python are one of the most important programming languages that contain packages for probabilistic programming. We will use the bnlearn package from R to generate the Bayesian model. This network will then be built using the pgmpy library in python. The model will be fit and used to make the necessary predictions.

Pgmpy is a python library for working with graphical models. It allows the user to create their own graphical models and answer inference or map queries over them. Pgmpy has the implementation of many inference algorithms like Variable Elimination, Belief Propagation etc (Ankan & Panda, 2015). Pgmpy was made by Ankur Ankan and Abinash Panda and was presented on Scipy Conference 2015 in Austin, Texas.

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) (Scutari, 2019)

2.7 Review of Literature

There are many prediction systems that have been designed for different diseases diagnosis using different techniques. Examples of these diseases are breast cancer, diabetics, heart diseases, flu, cold, uterine fibroid diseases, etc.

Prediction of heart diseases has been going on for two decades now. Most of the papers have implemented several techniques such as Decision Tree, Naïve Bayes, Neural Network, Support Vector Machine each showing different levels of accuracy. Some papers used data set from the UCI repository while others use data from local source hospital in their immediate environment.

Kumar, Koushi and Deepak (2018) used a number of algorithms including Decision Tree, J48 algorithm, Logistic Model Tree algorithm, Naïve Bayes, KNN, Support Vector Machine to predict heart diseases. They presented a new model that enhanced the decision tree accuracy in heart disease prediction. The Waikato Environment for Knowledge Analysis (WEKA) and the UCI dataset was used in the implementation. Out of the four, the Naïve Bayes classifier was the best in performance followed by Support Vector Machine, the Decision Tree and then the K-nearest neighbor.

X Liu et al.,(2017) used a hybrid classification system based on Relief F and Rough Set (RFRS) method. The system is made up of two subsystems: the RFRS feature selection system and a classification system with an ensemble classifier. Data discretization, feature extraction and Relief F algorithm are the stages of the first system. An ensemble classifier is proposed for the second system. The dataset used is the stat log dataset obtained from the UCI repository. The accuracy was 92.59% with MATLAB as the Implementation tool.

Zriqat, Altamimi, Azzeh(2017) used five data mining algorithm which are Naïve Bayes, Decision Tree, Discriminant, Random Forest and Support Vector Machine. The Algorithm was implemented using MATLAB with two datasets: Cleveland and stat log. Results showed that all algorithms are predictive with Decision Tree outperforming other classifiers with an accuracy of 99.0% followed by Random Forest of 98.15%. The Similarity between the Decision Tree and

Random Forest is responsible for their close accuracy. Generally, ensemble learning has proved to be superior but in their case, the Decision Tree Outperformed its ensemble version.

Khateeb and Usman (2017) used algorithms such as Naïve Bayes, K- Nearest Neighbor, Decision Tree and bagging technique. KNN was found to be the best technique with an accuracy of 79.2%. WEKA tool was used for the implementation and the datasets used was Cleveland dataset.

J.Patel et al.,(2016) compares the different algorithm of decision tree classification for better performance in heart disease diagnosis. The algorithm tested are the J48 algorithm, Logistic model tree algorithm and Random Forest algorithm. Cleveland datasets from UCI data repository was used. It contains 303 instances and 76 attributes. The tool used for implementation was WEKA. J48 tree technique came out as the best classifier because it is more accurate and took the least time to build. This was followed by the logistic model tree and the Random Forest algorithm respectively. Application of reduced error pruning to J48 results in higher performance compared to where it was not applied. J48 has an accuracy of 56.76% and build time of 0.04 seconds while the LMT algorithm has the lowest accuracy of 55.77% and build time of 0.39 seconds.

Wiharto, Kusnanto and Herianto(2015) used a multiclass performance classification Support Vector Machine to diagnose the level of heart disease. The study used multiclass SVM algorithm namely: One-against-one(OAO), One-against-all(OAA), Binary Tree Support Vector Machine(BTSVM), Decision Direct Acyclic Graph(DDAG) and Exhaustive Output Error Correction Code(EOECC). The dataset used was the UCI Cleveland dataset with BTSVM accuracy of 61.8%. It was concluded that BT-SVM multiclass classification, OAO-SVM and SVM

–DDAC provide better performance than the binary classification approach. Also, these algorithms provides a relatively stable performance for all levels. The occurrence of the imbalance dataset for sick-low, sick-medium, sick-serious results in low performance of the system.

Vembandasamy,Sasipriya and Deepa(2015) used the Naïve Bayes Algorithm in heart disease detection. The dataset used was the clinical dataset gotten from one of the top research institutes of diabetics in Chennai. The dataset contains a record of about 500 patients with 11 attributes. The experiments were conducted using the WEKA data mining tool with 70% of percentage split. The proposed model was able to classify with 86.4% accuracy. It exhibited a recall of 74% in average, precision of 71% on average and F measure of 71.2% on average.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

In this chapter, we describe the various methods and materials used for this study: The research design, Cleveland Heart Disease Dataset, data preprocessing and the tools used for the study.

3.2 Network design

This section illustrates the network design. It describes the actual flow of the entire network building. A flowchart is shown below that explains the sequence involved in the network design. In principle, data is collected through download from the UCI site. It is then preprocessed to solve for the problem of missing values, discretized and label encoded. The network is then built, trained and tested. The flow diagram of the network is as shown in Figure 3.1.

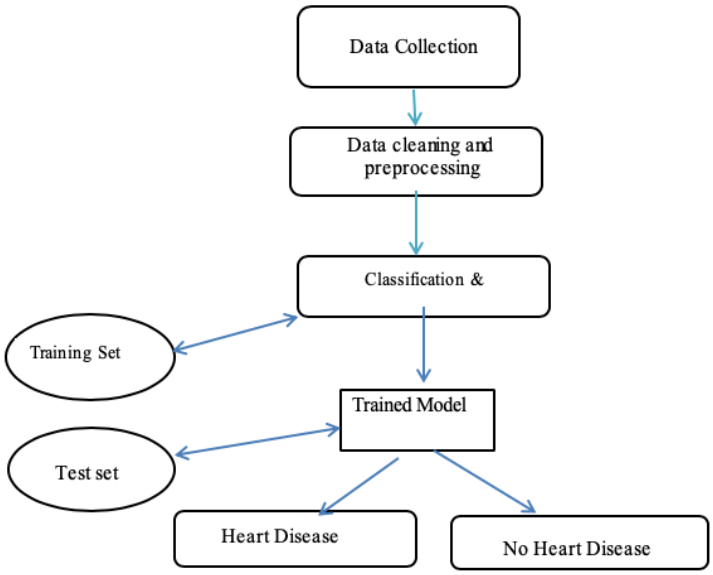


Figure 3. 1. Flow Diagram of Network Design

3.3 Cleveland Heart Disease Data set

This research uses the dataset provided by the University of California, Irvine Machine Learning Repository. The data is a Heart Disease dataset obtained from the V.A Medical Center, Long Beach and Cleveland Clinic Foundation. It consists of 303 samples with 14 attributes; 13 numeric input named age, sex, chest pain type, cholesterol, fasting blood sugar, resting ecg, maximum heart rate, exercise induced angina, old peak, slope, number of vessels colored and thal and one output attribute named target. Target contains 5 classes indicating either healthy or one of the four sick types. Table 3.1 shows the detailed description of the Cleveland dataset.

Table 3. 1. Attributes of Cleveland heart disease dataset Attribute Description

Attribute	Description	Domain of Values
Age	Age in years	29-79
Sex	Sex	0→ Female, 1→ Male
Cp	Chest pain type	1→Typical angina, 2→Atypical angina 3→Non-angina, 4→Asymptomatic
Trestbps	Resting blood sugar	94 to 200 mm Hg
Chol	Serum cholesterol	126 to 564 mg dL^{-1}
Fbs	Fasting blood	sugar >120 mg dL^{-1}: 0→ False, 1→True
Restecg	Resting ECG result	0→Normal, 1→ST-T wave abnormality2→LV hypertrophy
Thalach	Maximum heart rate achieved	71 to 202
Exang	Exercise induced angina	0→No, 1→Yes
Oldpeak	ST depression induced by exercise relative to rest	0 to 6.2
Slope	Slope of peak exercise ST segment	1→ Upsloping, 2→Flat, 3→Downsloping
Ca	Number of major vessels colored by fluoroscopy	0-3

Thal	Defect type	3→ Normal, 6→Fixed defect, 7→Reversible defect
Target	Heart diseases	0-4

3.4 Preprocessing data

Data preprocessing is also known as cleaning data. It is one of the most important steps in order to achieve the best from the dataset. This is a process whereby data inconsistencies such as missing values, out of range values, unformatted data, noise are removed from the data. The process is usually time-consuming because it involves a lot of experimentation trying out various data analysis tools.

Our preprocessing involves data retrieval, handling missing values, target class transformation and data discretization.

3.4.1 Data Retrieval

The first step is usually to get some data. It can be gotten from various sources. It can be as easy as someone handing over a file on a drive for you to analyze them directly. Or you need to download it or issue a database query to collect the data. Our dataset is downloaded from the UCI Machine Repository.

3.4.2 Handling Missing Values

In real-world data, there are some instances where a particular element is absent because of various reasons, such as corrupt data, failure to load the information, or incomplete extraction. Handling the missing values is one of the greatest challenges faced by analysts,

because making the right decision on how to handle it generates robust data models (KISHAN, 2019).

These are 5 ways of handling missing data:

1. Deleting Rows
2. Replacing with mean/median/mode
3. Assigning a unique category
4. Predicting the missing values
5. Using algorithms which supports missing values

We adopted Deleting rows because complete removal of data with missing values results in a highly accurate model especially when we have few missing values.

3.4.3 Target Class Transformation

As stated in the data set description, the target class contains values (0, 1, 2, 3, 4). Where 0 means healthy (no heart disease) and (1, 2, 3, 4) means the presence of sickness of varying degrees. Interest is in the presence or absence of heart disease, so the need to limit the class to (0, 1). Level (1, 2, 3, 4) was converted to 1.

3.4.4 Data Discretization

In our dataset, 5 out of the 14 attributes are continuous. Variables in Bayesian network models are discrete in nature, and therefore we need to make this continuous data categorical. We rely on expert knowledge to discretize our data. The attributes that are continuous are age, trestsbp, chol, thalach, and oldpeak.

3.5 Performance Metrics

Performance metrics are used to evaluate how different algorithms perform based on various criteria such as accuracy, precision, recall etc. Different performance metrics are discussed below.

Confusion Matrix

The confusion matrix shows the performance of the algorithm. It depicts how the classifier is confused while predicting. The rows indicate the actual instance of the class label while the columns indicate the predicted class instances. The Table 3.2 show a confusion matrix for binary classification.

Table 3. 2.Confusion Matrix

Actual Label	**Predicted Label**	
	+(1)	-(0)
+(1)	True Positive	False Negative
-(0)	False Positive	True Negative

True Positive value means the positive value is correctly predicted, false positive means the positive value is falsely classified, false negative means the negative value is falsely predicted while the true negative means the negative value is correctly classified.

Confusion matrix table is used to calculate different performance metrics as discussed below.

Accuracy

Accuracy is defined as the ratio of the number of correctly classified instances to all the cases. It is equal to the sum of TP and TN divided by the total number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

Precision is defined as the proportion of true positive instances which are classified as positive. It shows how close predicted values are to each other (Max, 2013).

$$Precision = \frac{TP}{TP + FP}$$

Recall

Recall is defined as the proportion of positive instances are that correctly classified as positive. Recall is also often called sensitivity.

$$Recall = \frac{TP}{TP + FN}$$

F1 Score

F1 score is a measure that combines both precision and recall and tries to find a balance between both.

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall}$$

3.6 Tools Used

Various tools are used for this study. All of them are free and open source.

1. Python 3.6.5
2. Pgmpy
3. Pandas 0.23.0
4. NumPy 1.14.3
5. Matplotlib 2.2.2
6. SciPy and Scikit-learn 0.19.1
7. Seaborn 0.8.1
8. Bnlearn in R

Python is a powerful and fast programming language. It is friendly and easy to learn. It runs everywhere and popularly use in applications such as Web and Internet Development, Database Access, Desktop GUIs, Scientific & Numeric, Network Programming, Software &Game Development etc. (Python Programming Documentation, 2019).

Python is used for this project because it is very open source and easy to use. Its documentation and community support is also very good.

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. It provides complete set of data analysis tools for python and is best competitor for R programming language. Operations like reading data-frame, reading csv and excel files, slicing, indexing, merging,

handling missing data etc., can be easily performed with Pandas (Pandas Documentation, 2019).

NumPy is very powerful package used for scientific computing. It comes with sophisticated functions and is able to perform N-dimensional array, algebra, Fourier transform etc. NumPy is used in data analysis, image processing. Different other libraries are built above NumPy and NumPy acts as a base stack for those libraries (Numpy Documentation , 2019).

Seaborn is a library for making statistical graphics in Python. It is used for data visualization. It is high level library stacked on top of matplotlib. Seaborn is more attractive and informative than matplotlib. It is easier to use and is tightly integrated with NumPy and Pandas. Seaborn and matplotlib can be used essentially side by side to derive conclusions from the datasets (Waksom, 2018).

SciPy consists of mathematical functions and is built on top of Numpy. Scikit-learn is a popular library for machine learning, it is third party extension to SciPy. Scikit-learn includes all the tools and algorithms needed for most of machine learning tasks. Scikit-learn supports regression, classification, clustering, dimensionality reduction and data pre-processing(Pedregosa, 2011).

CHAPTER FOUR

IMPLEMENTATION

4.1 Introduction

In this chapter, we discuss the various implementations done in data preprocessing, structure and parameter learning to achieve our desired model.

4.2 Data Preprocessing

As discussed earlier, data preprocessing is one of the most important steps to achieve the best from data set. It is very crucial in Machine Learning. We will be focusing on data retrieval, handling missing values, data transformation and data discretization.

4.2.1 Data retrieval

The first step of constructing our model is to retrieve the data from the UCI repository. The data set was named processed.cleveland.data on the UCI website. The data was downloaded and renamed as processed.cleveland.txt. Microsoft excel was then used to convert it to a CSV file named processed.csv. Attributes names are added as stipulated in the documentation. The CSV format was chosen for easy data preprocessing using Pandas. Figure 4.1 shows the processed.cleveland.data and Figure 4.2 shows the processed.csv.

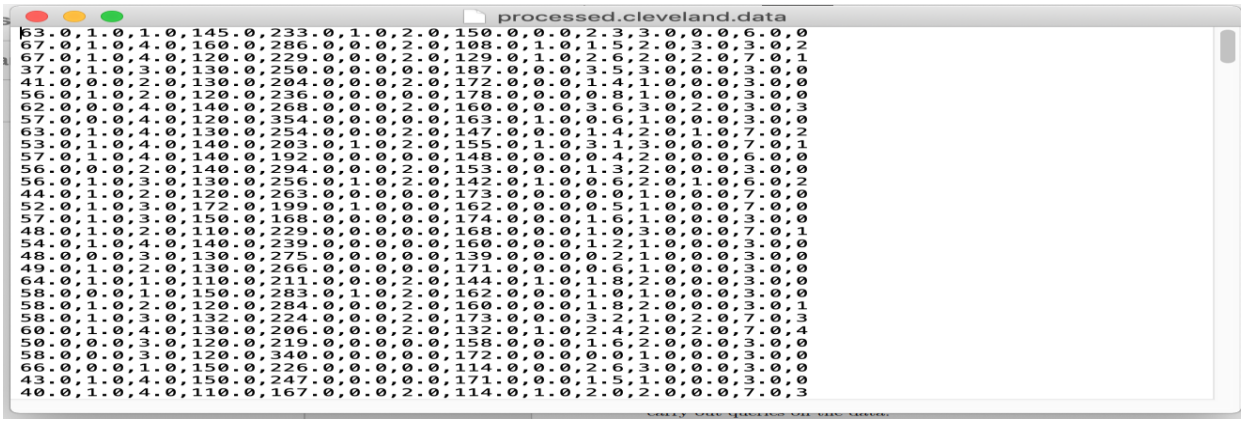
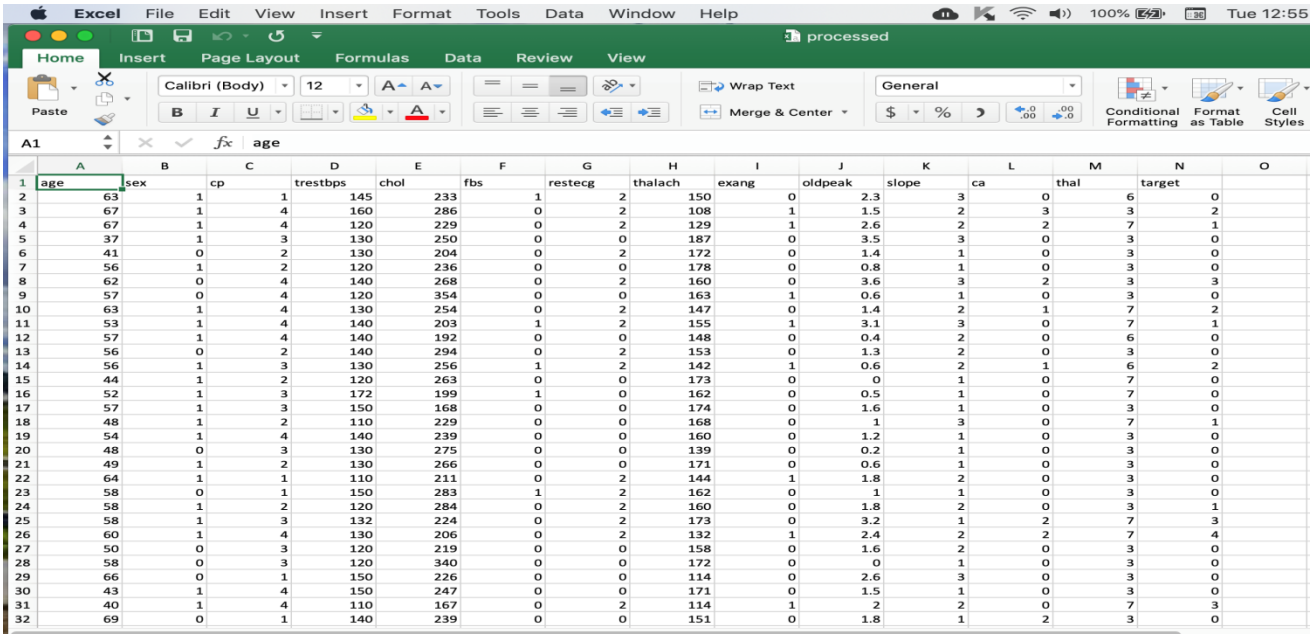


Figure 4. 1. Preprocessed.cleveland.data



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	
2	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0	
3	67	1	4	160	286	0	2	108	1	1.5	2	3	3	2	
4	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1	
5	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0	
6	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0	
7	56	1	2	120	236	0	0	178	0	0.8	1	0	3	0	
8	62	0	4	140	268	0	2	160	0	3.6	3	2	3	3	
9	57	0	4	120	354	0	0	163	1	0.6	1	0	3	0	
10	63	1	4	130	254	0	2	147	0	1.4	2	1	7	2	
11	53	1	4	140	203	1	2	155	1	3.1	3	0	7	1	
12	57	1	4	140	192	0	0	148	0	0.4	2	0	6	0	
13	56	0	2	140	294	0	2	153	0	1.3	2	0	3	0	
14	56	1	3	130	256	1	2	142	1	0.6	2	1	6	2	
15	44	1	2	120	263	0	0	173	0	0	1	0	7	0	
16	52	1	3	172	199	1	0	162	0	0.5	1	0	7	0	
17	57	1	3	150	168	0	0	174	0	1.6	1	0	3	0	
18	48	1	2	110	229	0	0	168	0	1	3	0	7	1	
19	54	1	4	140	239	0	0	160	0	1.2	1	0	3	0	
20	48	0	3	130	275	0	0	139	0	0.2	1	0	3	0	
21	49	1	2	130	266	0	0	171	0	0.6	1	0	3	0	
22	64	1	1	110	211	0	2	144	1	1.8	2	0	3	0	
23	58	0	1	150	283	1	2	162	0	1	1	0	3	0	
24	58	1	2	120	284	0	2	160	0	1.8	2	0	3	1	
25	58	1	3	132	224	0	2	173	0	3.2	1	2	7	3	
26	60	1	4	130	206	0	2	132	1	2.4	2	2	7	4	
27	50	0	3	120	219	0	0	158	0	1.6	2	0	3	0	
28	58	0	3	120	340	0	0	172	0	0	1	0	3	0	
29	66	0	1	150	226	0	0	114	0	2.6	3	0	3	0	
30	43	1	4	150	247	0	0	171	0	1.5	1	0	3	0	
31	40	1	4	110	167	0	2	114	1	2	2	0	7	3	
32	69	0	1	140	239	0	0	151	0	1.8	1	2	3	0	

Figure 4. 2. Processed.csv

4.2.2 Handling Missing Values

There are six missing values in our data. Since this is relatively small compared to the whole dataset, we handled the missing data by deleting the rows containing them. The code implementation is shown in Figure 4.3.

```
# To import important libraries
import numpy as np
import pandas as pd
# To read our data
df = pd.read_csv("processed.csv")
# To view the six missing values
df.thal.value_counts()
3    166
7    117
6     18
?      2
Name: thal, dtype: int64
df.ca.value_counts()
0    176
1     65
2     38
3     20
?      4
Name: ca, dtype: int64
# To delete the six rows with missing values
df.drop([87,166,192,266,287,302],axis=0,inplace=True)
```

Figure 4. 3. Python code for handling missing values.

4.2.3 Target Class Transformation

To make target class values (0 and 1) to depict absence of heart disease(0) and presence of heart disease(1). The code implementation is shown in Figure 4.4

```
# To replace (2,3,4) with 1
df['target'].replace([2,3,4],[1,1,1],inplace=True)
```

Figure 4. 4. Python code for Target class transformation

4.2.4 Label Encoding

In our dataset, there are three categorical variables Cp, Slope and Thal that is not well represented. Cp is represented as 1, 2 , 3 and 4. 1, 2, 3, 4 labels for a categorical variable of size four will throw an index error when applied directly to machine learning algorithm. Similarly is the case of thal that has values 3,6 and 7 for a variable of size 3 and slope that has values 1,2 and 3 for a variable of size 3. We then converted the Cp values of (1,2,3,4) to (0,1,2,3) , slope values of (1,2, 3) to (0, 1, 2) and the thal values of (3,6,7) to (0,1,2). The code implementation is shown in Figure 4.5.

```
# To change  Cp(1,2,3,4) to (0,1,2,3)
df['cp'].replace([1,2,3,4],[0,1,2,3],inplace=True)
# To change  slope (1,2,3) to (0,1,2)
df['slope'].replace([1,2,3],[0,1,2],inplace=True)
# To change  thal(13,6,7) to (0,1,2)
df['thal'].replace([3,6,7],[0,1,2],inplace=True)
```

Figure 4. 5. Python code for label encoding

4.2.5 Data Discretization

In our dataset, 5 out of the 14 attributes are continuous. Variables in Bayesian network models are discrete in nature, and therefore we need to make this continuous data categorical.

We rely on expert knowledge to discretize out data. The attributes that are continuous are age, trestsbp, chol, thalach, and oldpeak.

Age attribute's range is from 29 to 77. Range 29-45 was assigned to 0(young), range 46-64 was assigned to 1(middle) and 64-77 assigned to 2(old).). The code implementation is shown in Figure 4.6.

```
# To discretize age attribute

bins = [28,45,64,79]

names = [0,1,2]

df['AgeC'] = pd.cut(df['age'], bins, labels=names)
```

Figure 4. 6. Python code to discretize age attribute

Trestbps attribute's range from 94mmHg to 200 mmHg. According to the expert classification (Blood Pressure UK, 2008), range 90-120 was assigned to 0(ideal), Range 120-140 was assigned to 1(pre-high) and 140-200 assigned to 2(high).). The code implementation is shown in Figure 4.7.

```
# To discretize trestbps attribute

bins = [90,120,140,201]

names = [0,1,2]

df['trestbpsC'] = pd.cut(df['trestbps'], bins, labels=names)
```

Figure 4. 7. Python code to discretize trestbps attribute

Chol attribute's range from 126mg/dL to 564 mg/dL. According to the expert classification (Fletcher, 2017), range 125-200 was assigned to 0(normal), range 200-239was assigned to

1(borderline high) and 240-565 assigned to 2(high). The code implementation is shown in Figure 4.8.

```
# To discretize chol attribute

bins = [125,200,240,565]

names = [0,1,2]

df['cholC'] = pd.cut(df['chol'], bins, labels=names)
```

Figure 4. 8. Python code to discretize chol attribute

To discretize thalach, we use the expert knowledge that the approximate maximum heart rate is 220 – age (Wood, 2019). We first create a column that holds values(220- age). We then compare this column with our thalach attribute. If thalach is less than (220 – age), it is categorized as 0, else as 1. The code implementation is shown in Figure 4.9.

```
# To discretize thalach attribute

df['220-age'] = 220 - df['age']

df['thalachC'] =df['200-age'] < df['thalach']

df['thalachC'].replace([True,False],[1,0],inplace=True)
```

Figure 4. 9. Python code to discretize thalach attribute

According to Wikipedia, ST depression induced by exercise relative to rest can be categorized 0(irreversible ischaemia) for range 0 – 2, and 1(reversible ischaemia) for range 2 – 6.2. The code implementation is shown in Figure 4.10.

```
# To discretize oldpeak attribute

bins = [-1,2,6.3]

names = [0,1]

df['oldpeakC'] = pd.cut(df['oldpeak'], bins, labels=names)
```

Figure 4. 10. . Python code to discretize oldpeak attribute

We then drop the unneeded columns to ensure our variables are still 14. The attributes that were categorized was renamed adding C at their end, i.e ‘age’ becomes ‘ageC’.

4.3 Structure Learning and Parameter Learning

4.3.1 Structure Learning using Hill Climbing Algorithm

After the preprocessing of our data to a form suitable to model, we use bnlearn in R to construct the belief network of the attributes. We will be using the hill climbing implementation provided in the library. The code implementation is shown in Figure 4.11. The generated model is shown in Figure 4.12.

```r
# import bnlearn library
library(bnlearn)
# read in the CSV file
d <- read.csv("categorized1.csv",TRUE, ",")
# view the datatype
class(d)
# discretize data
d[,1:14]= lapply(d[,1:14],as.factor)
class(d[,13])
# view the structure of datd
str(d)
# apply hill climbing
t3 <- hc(d)
#construct the network
plot(t3)
#view the no of tests
ntests(t3)
```

Figure 4. 11. R Code for Structure Learning

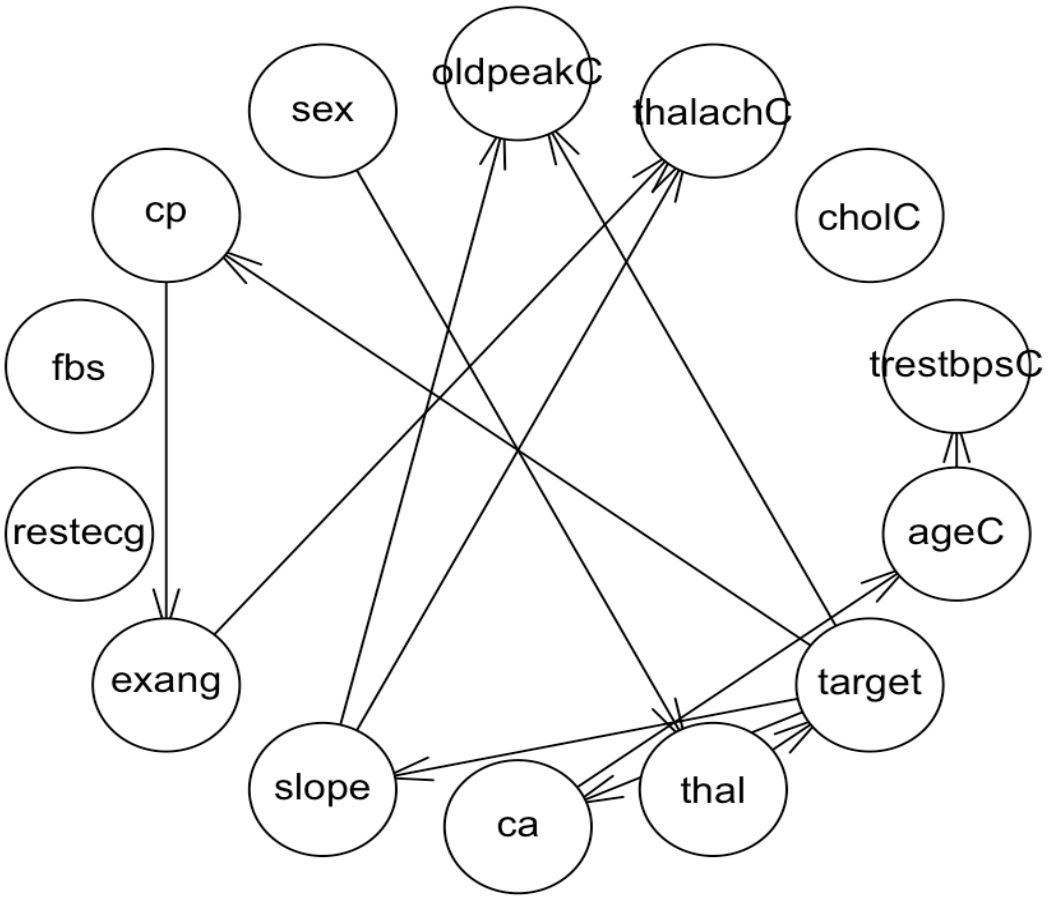


Figure 4. 12. The Belief Network of the attributes

4.3.2 Parameter Learning

The next thing is to learn the conditional probability table of each variable. The code implementation is shown in Figure 4.13.

```
18  # fit for parameter learning
19  fitted = bn.fit(t3,d)
20  fitted
```

Figure 4. 13. R Code for Parameter Learning

Parameters of node sex

Conditional probability table of attribute sex is as shown in Table 4.1. Female is represented as 0 and male 1.

Table 4. 1 Conditional probability table of attribute sex

0	1
0.3232323	0.6767677

Parameters of node cp

Conditional probability table of attribute cp is as shown in Table 4.2. The table depicts P(cp|target).

Table 4. 2. Conditional probability table of attribute cp

target	0	1
CP		
0	0.10000000	0.05109489
1	0.25000000	0.06569343
2	0.40625000	0.1313868
3	0.24375000	0.75182482

Parameters of node fbs

Conditional probability table of attribute fbs is as shown in Table 4.3. It gives the P(fbs).

Table 4. 3. Conditional probability table of attribute fbs

0	1
0.8552189	0.1447811

Parameters of node restecg

Conditional probability table of attribute restecg is as shown in Table 4.4. It gives the P(restecg).

Table 4. 4. Conditional probability table of attribute restecg

0	1	2
0.49494949	0.01346801	0.49158249

Parameters of node exang

Conditional probability table of attribute exang is as shown in Table 4.5. The table depicts P(exang|cp).

Table 4. 5. Conditional probability table of attribute exang

Cp	0	1	2	3
Exang				
0	0.82608696	0.91836735	0.86746988	0.45070423
1	0.17391304	0.08163265	0.13253012	0.54929577

Parameters of node slope

Conditional probability table of attribute slope is as shown in Table 4.6. The table depicts P(slope|target).

Table 4. 6. Conditional probability table of attribute slope

Target	0	1
Slope		
0	0.64375000	0.26277372
1	0.30000000	0.64963504
2	0.05625000	0.08759124

Parameters of node ca

Conditional probability table of attribute ca is as shown in Table 4.7. The table depicts P(ca|target).

Table 4. 7. Conditional probability table of attribute ca

Target	0	1
Ca		
0	0.8062500	0.3284672
1	0.1312500	0.3211679
2	0.0437500	0.2262774
3	0.0187500	0.1240876

Parameters of node thal

Conditional probability table of attribute thal is as shown in Table 4.8. The table depicts P(thal|sex).

Table 4. 8. Conditional probability table of attribute thal

sex	0	1
thal		
0	0.83333333	0.41791045
1	0.01041667	0.08457711
2	0.15625000	0.49751244

Parameters of node target

Conditional probability table of attribute target is as shown in Table 4.9. The table depicts P(target|thal).

Table 4. 9. Conditional probability table of attribute target

thal	0	1	2
target			
0	0.7743902	0.3333333	0.2347826
1	0.2256098	0.6666667	0.7652174

Parameters of node ageC

Conditional probability table of attribute ageC is as shown in Table 4.10. The table depicts P(ageC|ca).

Table 4. 10. Conditional probability table of attribute ageC

ca	0	1	2	3
agec				
0	0.31034483	0.07692308	0.02631579	0.05000000
1	0.60344828	0.75384615	0.73684211	0.65000000
2	0.08620690	0.16923077	0.23684211	0.30000000

Parameters of node trestbpsC

Conditional probability table of attribute trestbpsC is as shown in Table 4.11. The table depicts P(trestbpsC|ageC).

Table 4. 11. Conditional probability table of attribute trestbpsC

ageC	0	1	2
trestbpsC			
0	0.54098361	0.25641026	0.34146341
1	0.39344262	0.51794872	0.21951220
2	0.06557377	0.22564103	0.43902439

Parameters of node cholC

Conditional probability table of attribute cholC is as shown in Table 4.12. It gives the P(cholC).

Table 4. 12. Conditional probability table of attribute cholC

0	1	2
0.1649832	0.3265993	0.5084175

Parameters of node thalachC

Conditional probability table of attribute thalachC is as shown in Table 4.13. The table depicts P(thalachC|slope,exang).

Table 4. 13. Conditional probability table of attribute thalachC

slope = 0

exang	0	1
thalachC		
0	0.1504425	0.3461538
1	0.8495575	0.6538462

slope = 1

exang	0	1
thalachC		
0	0.4133333	0.7580645
1	0.5866667	0.2419355

slope = 2

exang	0	1
thalachC		
0	0.1666667	0.7777778
1	0.8333333	0.2222222

Parameters of node oldpeakC

Conditional probability table of attribute oldpeakC is as shown in Table 4.14. The table depicts P(oldpeakC|slope,target).

Table 4. 14. Conditional probability table of attribute oldpeakC

target = 0

slope	0	1	2
oldpeakC			
0	0.990291262	0.958333333	0.555555556
1	0.009708738	0.041666667	0.444444444

target = 1

slope	0	1	2
oldpeakC			
0	0.944444444	0.651685393	0.166666667
1	0.055555556	0.348314607	0.833333333

4.4 Training the Network

After obtaining our model as shown in Figure 4.4, we need to fit our training data to this model. Our model was named heart model and was created using pgmpy as shown below. The code implementation is shown in Figure 4.14.

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from pgmpy.models import BayesianModel

heart_model = BayesianModel([('cp', 'exang'),('target', 'cp'), ('sex',
'thal'),('slope', 'oldpeakC'), ('target', 'oldpeakC'), ('exang',
'thalachC'),('slope', 'thalachC'),('ageC', 'trestbpsC'),('ca', 'ageC'),
('target', 'slope'), ('target', 'ca'), ('thal', 'target'), ('thal', 'ca')])

df1 = pd.read_csv("categorized1.csv")

df2 = df1.drop(['fbs','restecg','cholC'], axis = 1)

y =
df2.drop(['sex','cp','exang','slope','ca','thal','ageC','trestbpsC','thalachC
','oldpeakC'], axis = 1)

x = df2.drop(['target'], axis = 1)

x_train, x_test, y_train, y_test = train_test_split(x,y,test_size =
0.2,random_state=0)

data_train= pd.concat([x_train, y_train], axis=1)

heart_model.fit(data_train, estimator=MaximumLikelihoodEstimator)
```

Figure 4. 14. Python code to train the network.

4.5 Testing

We have successfully fitted our model to the data and need to test our model now. We will now be using our test dataset and would be making predictions on it. The code implementation is shown in Figure 4.15.

```
# Obtain prediction
target_predict_values= heart_model.predict(x_test)
target_predict_values_array = target_predict_values.target.values
target_predict_values_array
array([0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0,
       0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0])
y_test_array = y_test.target.values
y_test_array
array([0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,
       0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0,
       0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0])
```

Figure 4. 15. Python code to test the network

4.6 Performance Evaluation

This is to show how well our model performed. Refer to section 3.5 to get indepth knowledge on what each performance metrics means. The model was able to predict 51 out of the 60 test samples correctly thereby achieving an accuracy of 85%. Other metrics are as shown below. The code implementation is shown in Figure 4.16.

```
from sklearn.metrics import precision_score,recall_score, f1_score,
confusion_matrix

confusion_matrix(y_test_array, target_predict_values_array)

array([[28,  2],

       [ 7, 23]])

from sklearn import metrics

from sklearn.metrics import classification_report

print('accuracy %s' %metrics.accuracy_score(y_test_array,
target_predict_values_array))

print(classification_report(y_test_array, target_predict_values_array))

accuracy 0.85

             precision    recall  f1-score   support

          0       0.80      0.93      0.86        30

          1       0.92      0.77      0.84        30

avg / total       0.86      0.85      0.85        60
```

Figure 4. 16. Python code for performance evaluation of the model

4.6.1 Comparison with Naïve Bayes

The performance evaluation of the Naïve Bayes model is shown below. It can be seen that it is poorer than the belief network model. It has accuracy of 80%. The code implementation is shown in Figure 4.17.

```
from sklearn.metrics import precision_score,recall_score, f1_score,
confusion_matrix

confusion_matrix(y_test.T,y_pred)

array([[27,  3],

       [ 9, 21]])

from sklearn import metrics

from sklearn.metrics import classification_report

print('accuracy %s' %metrics.accuracy_score(y_test,y_pred))

print(classification_report(y_test,y_pred))

accuracy 0.8

             precision    recall  f1-score   support

          0       0.75      0.90      0.82        30

          1       0.88      0.70      0.78        30

avg / total       0.81      0.80      0.80        60
```

Figure 4. 17. Python code for performance evaluation of Naïve Bayes model

CHAPTER FIVE

CONCLUSION

5.1 Conclusion

The research work developed a Bayesian network model for heart diseases prediction in a human being. This model was built using the hill climbing algorithm deployed in bnlearn package in R.

The main goal of the study is to show the effectiveness of the Bayesian classifiers in the prediction of heart diseases. We used two different implementations of Bayesian classifier: the Bayesian Belief Network and the Naïve Bayes.

The Bayesian Belief Network produced an intuitive graphical representation of the dependency. The obtained model helps us to easily identify the relationships of probabilistic causal dependencies and conditional independencies between attributes.

Dataset was collected from the University of California, Irvine machine learning repository. The dataset is obtained from V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. It consists of 303 instances of heart disease data each having 14 attributes; 13 numeric input attributes and one class output.

The performances of Bayesian classifiers are benchmarked against each other with the Bayesian Belief Network Outperforming the Naïve Bayes in the prediction of heart diseases.

This research will assist in making inference about heart diseases, thereby serving as a diagnostic tool to support medical practitioners.

Bibliography

Ankan, A., & Panda, A. (2015). Probabilistic Graphical Models using Python. *Proc. of the 14th Python in Science Conf. (SciPy 2015)*, (p. 11).

Blood Pressure UK. (2008). Retrieved from http://www.bloodpressureuk.org/BloodPressureandyou/Thebasics/Bloodpressurechart

Data-flair. (2019). Retrieved from https://data-flair.training/blogs/bayesian-network-applications/

Elsayad, A., & Fakhr, M. (2015). Diagnosis of Cardiovascular Diseases with Bayesian Classifiers. Journal of Computer Sciences 2015, 11 (2): 274.282 . DOI:10.3844/jcssp.2015.274.282

Fletcher, J. (2017, Feb 20). *What should my cholesterol level be at my age?* Retrieved from Medical News Today: https://www.medicalnewstoday.com/articles/315900.php

Giryes, R., & Elad, M. (2011). Reinforcement Learning: A Survey. (pp. 1475 -1479). Eur. Signal Process. Conf. https://doi.org/10.1613/jair.301

Kaur, B., & Singh, W. (2014). Review on Heart Disease Prediction System using Data Mining Techniques. *International Journal on Recent and Innovation Trends in Computing and Communication*(10), 3003 - 3008.

KISHAN, M. (2019, Feb 9). *Analyticsindia*. Retrieved from Analyticsindia: https://www.analyticsindiamag.com/5-ways-handle-missing-values-machine-learning-datasets/

Kohonen, T., Oja, E., Simula, O., & Visa, A. K. (1996). Engineering applications of the selforganizing map. . *Proceedings of the IEEE, 84(10),*, (pp. 1358 - 1384).

Koller., N. F. (2009). *Probabilistic Graphical Models: Principles and Tech- niques.* 1st ed. Adaptive Computation and Machine Learning series. The MIT Press, 2009. isbn: 0262013193,9780262013192. url: http://gen.lib.rus.ec/book/index. php?md5=8ac4fc1b: The MIT Press.

Max, B. (2013). *Principles of Data Mining.* 2nd ed. Springer.

Numpy Documentation . (2019). Retrieved from http://www.numpy.org/

Pandas Documentation. (2019). Retrieved from http://pandas.pydata.org/

Parra, P., Tauler, P., Veng, M., Ligeza, A., Gonzalez, A., & Aguilo, A. (2015). Bayesian Network modeling:A case sudy of an epidemiologic system analysis of cardiovascular risk. *Elsevier*. DOI:10.1016/j.cmpb.2015.12.010

Pearl. (1998).

Pedregosa, F. (2011). *Scikit-learn: Machine Learning in Python*. Retrieved from http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Python Programming Documentation. (2019). Retrieved from https://www.python.org/about/

Ray, S. (2017, September 11). *Analytics Vidya*. Retrieved from https://www.analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Sathya,R.,& Abraham, A. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence*, *2*(2), 34–38. https://doi.org/10.14569/IJARAI.2013.020206

Scutari, M. (2019). Retrieved from www.bnlearn.com

Spirites, P., Glymour, C., & Scheines, R. (2001). *Causation,Prediction and Search, Adaptive Computation and Machine Learning.* MIT Press.

Tekieh, M. H., & Raahemi, B. (2015). Importance of Data Mining in Healthcare. Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015. DOI: http://dx.doi.org/10.1145/2808797.2809367

USF, H. (2019). *Datamining in HealthCare*. Retrieved from https://www.usfhealthonline.com/resources/key-concepts/data-mining-in-healthcare/

Varun, S., Mounika, G., Sahoo, P., & Eswaran, K. (2019). Efficient System for Heart Disease Prediction by applying Logistic Regression 1. International Journal of Computer Science and Technology, Vol.10. Issue 1, Jan-March 2019.

Vomlel, V. (2019). Retrieved from http://staff.utia.cas.cz/vomlel/slides/presentace-karny.pdf

Waksom, M. (2018). *An Introduction to Seaborn*. Retrieved from http://seaborn.pydata.org/introduction.html

WHO. (2016). Retrieved from https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases- (cvds) 2016

WHO. (2017, July 03). Retrieved from http://www.who.int/cardiovascular_diseases/global-hearts/Global_hearts_ initiative/en/.03-Jul-2017

Witten, I., & Frank, E. (2005). *Practical Machine Learning Tools and Techniques.* Amsterdam.

Wood, H. T. (2019). *What Happens When You Reach Your Max Heart Rate?* Retrieved from Live Strong: https://www.livestrong.com/article/331250-what-happens-when-you-reach-your-max-heart-rate/

Yoo, I., Alafaireet, P., Marinov, M., Pena-Hernandez, K., Gopidi, R., Chang, F., & Hua, L.(2011).Data Mining in Healthcare and Biomedicine : A Survey of the Literature., (pp. 2431–2448). DOI: 10.1007/s10916-011-97105