# A HYBRID MODEL FOR PREDICTING MALARIA USING DATA MINING TECHNIQUES

**BY**

**Aminu Aliyu**

**(A00018729)**

**SCHOOL OF IT & COMPUTING, AMERICAN UNIVERSITY OF NIGERIA, YOLA, ADAMAWA STATE**

**Fall, 2017**

# A HYBRID MODEL FOR PREDICTING MALARIA USING DATA MINING TECHNIQUES

**BY**

**Aminu Aliyu**

**(A00018729)**

In partial fulfillment of the requirements for the award of degree of Master of Science (M.Sc.) in Computer Science submitted to the School of Graduate Studies (SGS), American University of Nigeria, Yola.

Fall, 2017

# DECLARATION

I, Aminu Aliyu, declare that the work presented in this thesis entitled ***'A Hybrid Model for Predicting Malaria Using Data Mining Techniques'*** submitted to the School of Graduate Studies, American University of Nigeria, in partial fulfillment for the award of the Master of Science (M.Sc.) in Computer Science. I have neither plagiarized nor submitted the same work for the award of any other degree. In case this undertaking is found incorrect, my degree may be withdrawn unconditionally by the University.

Date: 28/11/2017 Aminu Aliyu

Place: Yola ID.No: A00018729

# CERTIFICATE

I certify that the work in this document has not been previously submitted for a degree nor neither has it been submitted as a part of a requirement for a degree except fully acknowledged within this text.

---------- ----------
Student Date
Aminu Aliyu

---------- ----------
Supervisor Date
Dr. Rajesh Prasad

---------- ----------
Dean SITC Date
Dr. Mathias Fonkam

---------- ----------
Dean SGS Date
Dr. Charles Nche

# ACKNOWLEDGEMENTS

My gratitude goes to the Almighty Allah for giving me the strength, knowledge, and courage to carry out this thesis successfully. My profound gratitude goes to my supervisor **Dr. Rajesh Prasad** for his patience, advice and time to guide me throughout the thesis work, Dr. Mathias Fonkam, Dean SITC for his guide on the paper prepared for publication from the thesis, Dr. Charles Nche, Dean Graduate School and the entire staff of School of IT & Computing, American University of Nigeria, Yola, Nigeria.

My sincere appreciation goes to the entire Management of AUN especially faculty member Dr. Rao Narasimha Vajjhala and library officials for the kind support through making the AUN environment a learning one.

Also, I have to seize this opportunity to acknowledge the endless effort of my parents Hajiya Astajam Aliyu and the entire family, including my brothers and sisters for their cooperation and tireless concern for me all through.

Finally, I say a big thanks to my friends and course mates. I wish you success in all your future undertakings. I am most sincerely grateful to all that have contributed in one way or the other to the success of this thesis.

# ABSTRACT

Data mining is used in extracting rules to predict certain information in many areas of Information Technology, medical science, biology, education, and human resources. Data mining can be applied on medical data to foresee novel, useful and potential knowledge that can save a life, reduce treatment cost, increases diagnostic and prediction accuracy as well as save human resources. Data mining involve several techniques such as anomaly detection, classification, regression, clustering, time series analysis, association rule, and summarization. Classification is the most important application of data mining. In this thesis, we use a classification technique called Naïve Bayes (a supervised learner) to build a hybrid framework for classifying and predicting the status of only malaria and their complications in a suspect patient using their clinical presentation. For the purpose of this study, we considered the parameter: *fever, headache, nausea, vomiting, respiratory distress, convulsion,* and *coma* as the main distinct clinical symptom. This method has the relative advantage of easy to construct, can classify categorical data, and occurrences of an event (attributes) are independent, and work better on high dimensional data. The framework developed was divided into two phases *Classification Phase 1, Classification Phase 2* and is implemented using Java built on Weka library version 3.8.0. The framework was trained using data acquired from hospital and tested for performance accuracy using Receiver Operating Characteristic (ROC) and Confusion Matrix (CM). The results demonstrated that the system predicted accurately with performance accuracy of 90%, 98% on confusion matrix and 92%, 99% on ROC-Area under Curve (ROC-AUC) for *Classification Phase 1* and *Classification Phase 2* respectively. This means that ROC presented more optimal result than confusion matrix and such system should be useful for rural area where clinician or medical equipment are is not available to assist in predicting malaria is suspected malaria patient.

# TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

# LIST OF ABBREVIATIONS

| Abbreviation | Description |
|---|---|
| ANN | Artificial Neural Network |
| AUC | Area Under Curve |
| CART | Classification and Regression Tree |
| CHAID | Chi-Square Automatic Interaction Detection |
| CM | Confusion Matrix |
| CRISP-DM | Cross-Industry Standard Process for Data Mining |
| ECG | Electrocardiogram |
| IRB | Institutional Review Board |

| | |
|---|---|
| K-NN | K-Nearest Neighbor |
| LFT | Liver Function Test |
| ML | Machine Learning |
| MYSQL | My Structured Query Language |
| NBN | Naïve Bayes Network |
| ODANB | One Dependency Augmented Naïve Bayes |
| OOP | Object - Oriented Programming |
| *P.falciparum* | Plasmodium falciparum |
| REPTRee | Reduced Error Pruning Tree |
| ROC | Receiver Operating Characteristic |
| SVM | Support Vector Machine |
| UCI | Universal Client Identifier |

# CHAPTER ONE

# INTRODUCTION

This chapter introduced the basic idea about malaria, data mining, classification, problem statement which involves severity of malaria, our research objective and contribution, then finally chapterization of the entire thesis.

## 1.1 Background of the Study

Malaria can be regarded as a life-threatening parasite which is contained in spittle of mosquitoes and is transmitted through a bite (Pirnstill & Coté, 2015; Razzak, 2015). Thereafter biting human, it takes about 45 minutes to spread across entire human blood (Bartoloni & Zammarchi, 2012). Afterward, the infection would start confronting the body's red blood cells together with liver cells, altering the body's biochemistry and attributes of cells built - structure (Pirnstill & Coté, 2015). The four most common species of malaria parasites found in Sub-Saharan Africa includes *P.falciparum, P. vivax, P. ovale* and *malariae* ( Calderaro, Piccolo, Gorrini, Rossi, Montecchini, Dell'Anna, & Arcangeletti, 2013). Among them, *P.falciparum* was regarded as the most common causes of malaria and its severe cases (Gomes, Vitorino, Costa, Mendonça, Oliveira, & Siqueira-Batista, 2011). Therefore, historically, *P.falciparum* was attributed to the causes of malaria disease in Sub-Saharan Africa (Howes *et al*., 2015).

The numerous figures from assessing the hazard of this infection have been overstressed, with first initial score estimates of 300-500 million clinical cases annually leading to 1-3 million deaths globally (Chotivanich, Silamut, & Day, 2007). The parasite of *P.falciparum* cases of malaria alone was responsible for an approximate estimate of 40% (2.4 billion population) of the world that was affected by malaria (Gomes *et al.*, 2011). With reference to World malaria report of (2010), out of 225 million recorded incidence in the world, 781000 death in 2009 (Mohapatra, Jangid, & Mohanty, 2014), and from a total of 198 million incidences of malaria documented in 2013, 584000 were stated death (Gu, Chen, & Yang, 2015). The most devastating effect of malaria infection is that its most targets are children <5 years of age giving an annual estimate of >300 million population globally and >3000 pediatric per day (Stauffer & Fischer, 2003). However, many levels of this infection from even clinical point of view were present in patients,

in which at some degree may cause neurological complication in human brain called as cerebral malaria (Idro, Marsh, John, & Newton, 2010).

Cerebral malaria can be defined as any anomaly of mental status in a patient with malaria and has a case death rate between 15% and 50%. Cerebral malaria is a quickly progressive possibly lethal complication of *P.falciparum* infection. It is categorized by unarousable and persistent coma along with regular motor signs. The most vulnerable groups of people are pregnant women, children, and adults with weak immune system. Many scholars agreed to the stated term that the most occurring severity complications of malaria are severe anemia and cerebral malaria. Cerebral malaria is the most evident cause of neurological complication of malaria infection with *P.falciparum*. Its syndrome is clinically characterized by the presence of a sexual forms of the parasite and coma caused by no any other concomitant disease of these features (Idro *et al.*, 2010).

**1.2 Data Mining**

Data mining is a process of scrutinizing data from a different viewpoint and collecting the knowledge from such data (Hemanth, Vastrad, & Nagaraju, 2011; Dangare & Apte, 2012). Data mining involves the use of numerous techniques in pattern recognition and knowledge presentation. In fact, data mining task is accomplished by using*: Association, class description, classification, prediction, clustering,* and *time series analysis* (Srinivas, Rani & Govrdhan, 2010).

The healthcare industry in this 21st century is rich with data, and this data is an ingredient in data mining and knowledge discovery (Dangare & Apte, 2012). Knowledge discovery is a well-defined process of distinct phases and data mining is the important phase in the discovery of useful hidden knowledge in large databases (Soni, Ansari, Sharma, & Soni, 2011; Srinivas *et al*., 2010). For example, the huge amount of medical data that are gathered in hospital on daily basis contains hidden information. This hidden information could be extracted using various data mining technique and used for decision making (Taneja, 2013). Data mining provides a mechanism for novelty and discovering of unobserved patterns in data (Srinivas *et al*., 2010). Data mining can also be referred to as a process of discovering pattern and mining of this pattern from large datasets (Jothi, Rashid, & Husain, 2015).

Data mining has been used in the healthcare sector to provide an assistive tool for early detection, prediction systems of various diseases that can be used for decision making (Jothi *et*

*al*., 2015). It is sometimes used as the technique of both classification and clustering to achieve a common goal (Patil, Chopade, Mishra, Sane, & Sargar, 2016). Data mining can be useful in answering many vital and critical questions about health care (Srinivas *et al.*, 2010). For example, data mining has been used in prediction of malaria outbreak using a large data set from Maharashtra state, India. The prediction was achieved using two data mining classification Support Vector Machine (SVM) and Artificial Neural Networks (Sharma, Kumar, Panat, & Karajkhede, 2015). It is used to improve quality of service in the healthcare industries and assist the medical practitioner in reducing the number of adverse effect on drug in order to recommend cheap medicinally equivalent substitutes ( Srinivas *et al*., 2010).

**1.3 Classification**

Classification is the process of defining a fitting model which describes and differentiates class label with the aim of providing the ability to use the model to predict the class of tuples whose class label is unidentified (Soni, Ansari, Sharma, & Soni, 2011). This imitative model is constructed on the analysis of training data i.e., data tuples whose class label is identified (Tribhuvan, Tribhuvan, & Gade, 2015).

One key area of data mining that demonstrates its application in the healthcare sector is the use of classification techniques in classifying and predicting various diseases (Srinivas *et al*., 2010). Classification is supervised learning methods that extract models, labeling significant data classes or predicting upcoming trends (Soni *et al.*, 2011). It is the process of assigning a class to find formerly unseen records as correctly as possible by using a collection of records called a training dataset, where each tuple in the training set comprises a set of attributes, and then one of the attributes is called a class. The objective is to provide a classification model for the class elements, then devise a validation mechanism using test data set in order to determine the accuracy of the model ( Kuar & Wasan, 2006). This technique has been used in the healthcare environment to automatically diagnosis a patient's disease in order to choose immediate treatment while awaiting lab-test results (Nikam, 2015).

Several classification techniques exist in data mining which includes: Support Vector Machine (SVM), K-nearest neighbor, Naive Bayes, IB3, Artificial Neural Network (ANN), Decision Tree and J48, C4.5 version of decision tree classification (Nikam, 2015). Each of these techniques can be used to classify record depends on the nature of pattern in data and the phenomena to investigate ( Kriegel, Borgwardt, Kröger, Pryakhin, Schubert, & Zimek, 2007).

### 1.4 Problem Statement

The clinical presentation of malaria in a patient is the symptomatic features presented by patients. This feature is an indication of disease course and therefore, has direct significance in guiding clinicians about the decision to take. The combination of symptoms and signs has made tremendous achievement in predicting disease (Lubezky, Ben-Haim, Nakache, Lahat, Blachar, Brazowski, & Klausner, 2010). Even the standard diagnostic criteria developed by clinicians and researchers was based on clinical manifestation that assists with an integrated approach in treatment and management of disease (Laishram, Sutton, Nanda, Sharma, Sobti, Carlton, & Joshi, 2012; Patil *et al.*, 2016).

The main challenging issue confronting the healthcare industry is lack of quality of service at minimal cost implying from diagnosing to predicting patients correctly (Chaurasia & Pal, 2013) or administering therapy that is effective, and sometimes even understanding the complications that may result from diseases (Srinivas *et al*., 2010; Dangare & Apte, 2012). This issue can sometimes lead to an unfortunate clinical decision that can result in devastating consequences that are unacceptable (Dangare & Apte, 2012).

The availability of patients medical data has derived the need for clinicians, payers, and patients for an alternative computer-based assessment tool that can assist in decision making (Soni *et al*., 2011). For example, the physicians can compare analytical information of numerous patients with the matching condition and physicians can equally confirm their results with the conformity of other physicians dealing with a matching case from another part of the country (Srinivas *et al*., 2010). For a disease that can be complicated like malaria, the patient is first classified as either have the presence of malaria i.e. positive or negative before further classify the severity of the disease as either uncomplicated (mild) or complicated (severe) based on clinical manifestation (Bartoloni & Zammarchi, 2012). In each case, it's highly important to understand the clinical features of these classes (Gomes *et al*., 2011). In the case of classifying positive malaria, the most clinical features are *fever, headache, vomiting,* and *loss of appetite* in accordance with the report for predicting malaria (Ndyomugyenyi, Magnussen, & Clarke, 2007). Based on this assumption, we consider these features in predicting positive malaria in phase 1. Further classifications of complicated or uncomplicated are in phase 2. Following are the classes considered in the study:

i. **Positive Class (P):** Patient can be confirmed to have positive malaria when the patient has one or more of the above symptoms and has also been confirmed by laboratories (Mutanda, Cheruiyot, Hodges, Ayodo, Odero, & John, 2014). Patients found to be positive from diagnosis can confirm the clinical suspicion of malaria (Bartoloni & Zammarchi, 2012). Therefore, in the course of this study, we considered fever, headache, nausea, and vomiting as the most occurring symptoms in a patient with malaria.

ii. **Negative Class (N):** Patient may have some of the parameters (symptoms) of positive malaria, but after several trying of diagnostic tests to confirm, the malaria is undetectable (Cdc, 2013). This means that the existence of the signs may be as a result of other concomitant disease but not really caused by malaria (Rai & Abraham, 2012).

iii. **Mild or Uncomplicated Class (U)** of malaria is the presence of one or few signs of clinical manifestation such as mile fever, sweating, weakness, chills, loss of appetite coupled with a headache or recent history of malaria but no signs of severity (Arévalo-Herrera, Lopez-Perez, Medina, Moreno, Gutierrez, & Herrera, 2015). The core central features, in this case, are fever and headache (Rai & Abraham, 2012).

iv. **Severe or Complicated Class (C)** manifest with one or more of the following features; repeated generalized convulsions, impaired consciousness / coma or circulatory collapse/shock, acute respiratory distress syndrome, severe anemia, renal failure, metabolic acidosis, hypoglycemia, hyperthermia, abnormal bleeding and hyperparasitaemia (Bartoloni & Zammarchi, 2012; Laishram *et al.*, 2012). Headache and high fever are centered across all class of suspected patients with malaria (Rai & Abraham, 2012).

## 1.5 Research Aim & Objectives

The aim of this study is to design a hybrid model for predicting malaria which utilize large data obtained from hospital.

### 1.5.1 Research Objectives

The major objectives of this research are:

i. To model malaria prediction using Naïve Bayes classifier in order to enhance the *accuracy* of the prediction model.
ii. To support the understanding of various stages of malaria infection causing complications such as cerebral malaria (malaria which causes edema in a human brain).

### 1.6 Limitation of the Study

This research study is limited to predicting only malaria and no any other disease. The study highlighted the significance of Naïve Bayes classifier in supervised learning considering its independent assumption. Performance of the system was checked using confusion matrix and ROC only.

### 1.7 Chapterization

The work in this thesis is about building a hybrid model for predicting malaria using data mining approach. The entire thesis is distributed over five chapters. Each chapter highlighted different topics and subtopics.

Chapter 1 contains introduction about malaria, data mining, classification, problem statement, objective and our thesis contribution.

Chapter 2 discusses the basic concept of machine learning, data mining, classification, and critically review literature.

Chapter 3 explain choosing the methodology of Naïve Bayes classification algorithm, the material used in the design, experimental setup and performance evaluation criteria, particularly, the use of confusion matrix and Receiver Operating Characteristics (ROC) in demonstrating accuracy measure of a classifier.

In Chapter 4, we presented a demo of design, result discussion, and findings of the stated methodology.

Finally, in Chapter 5, we draw conclusion, recommendation, and future work.

# CHAPTER TWO

# LITERATURE REVIEW

In this chapter, we discuss basic concepts of machine learning, data mining, classification, clustering, and prediction. We highlight related techniques such as Decision tree, SVM, and Naïve Bayes classifiers with their algorithms. We, finally review related literature in the data mining especially Naïve Bayes classification.

## 2.1 Basic Concept of Machine Learning, Data Mining, and Classification

Machine Learning is a study which existed from the area of artificial intelligence that uses a variety of probabilistic, statistical, and optimization techniques to train computer system in order to scrutinize and "learn" discern and hard patterns in complex, large and noisy data (Vihinen, 2012). It is about learning how to do better in upcoming based on experience learned in the past (Cruz & Wishart, 2006). For example, learns to act as an intelligent or predict disease accurately based on some number of observations (Raj & Prasanna, 2013). The objective is to develop learning algorithms that can learn automatically without human assistance or intervention. Machine learning can be applied when people are susceptible to making mistakes at the time of analysis or, perhaps, when trying to create relationships between many features. It is used for improving the efficiency of a system as well as designs of machines (Archana & Elangovan, 2014). Machine learning provides an alternative solution to a medical problem by using different techniques such as clustering and classification applied on previous real data to predict current disease**.** This approach was found stimulating by many researchers trying to use medical data to predict disease (Durairaj & Ranjani, 2013; Razzak, 2015)**.**

## 2.2 Data Mining

Data mining is the most important application of machine learning (Srinivas *et al*., 2010). It can be defined as an extraction of information from huge amount of datasets (Sala al-Din Abdullah, 2016)**.** In another word, data mining is simply defined as "Knowledge mining in data". The key focus area of data mining is pattern recognition (Jothi *et al.,* 2015). Data mining is used in extracting rules and predicting certain performances in many areas of information technology, science medicine, biology, education, and human resources (Al-radaideh & Nagi, 2012). Data mining can be applied on medical data to foresee useful, new and potential knowledge that can

save life, reduce treatment cost, increase diagnostic accuracy as well as save human resources (Manjusha, Sankaranarayanan & Seena, 2015). The knowledge acquired from data mining can be applied in fraud detection, customer retention, and market analysis, to science exploration and production control (Baby & Priyanka, 2012). Data mining involve several techniques such as anomaly detection, classification, regression, clustering, time series analysis, association rule and summarization (Ameta & Jain, 2017).

### 2.2.1 Clustering

Clustering also called data segmentation is the process of an alliance of data items that are similar or dissimilar in some sense into one cluster (Soni *et al.,* 2011). It divides data into clusters according to certain similarities measure (Shinde, Arjun, Patil & Waghmare, 2015). Clustering is the act of determining structures and groups in data that are in one way or the other similar, without the use of known structures in the data (Baby & Priyanka, 2012). There are many clustering techniques which include K-Nearest Neighbor, K-Means, K-medsiod etc.

### 2.2.2 Classification

Classification, a data mining technique, is the process of classifying and predicting the value of class attribute based on the values of predictors (Romero, Ventura, Espejo & Hervás, 2008). Predictors are attributes that are used to predict new dataset e.*g. fever, headache, nausea, vomiting, respiratory distress, convulsion,* and *coma* etc. There are two main categories of classification model used in prediction: *Descriptive* and *predictive* classification model.

Descriptive model detects relationships or pattern in data and explore even the properties of the scrutinized data. Example of such technique which support this includes summarization, clustering, association rule etc. While, predictive model conducts prediction of unknown data values by using supervised learning function applied on known values (Jothi, Rashid & Husain, 2015). The known data is historical in nature. Example of such techniques includes Time series analysis, Prediction, Classification, Regression etc. Our interest in this study lies in predictive classification model, where the model is constructed based on a feature of historical data and are used to predict future trend (Al-radaideh & Nagi, 2012). Many classification algorithms are used for classifying categorical data e.g. *Decision tree, K-Nearest Neighbor, Naïve Bayes, SVM, J48, Random Forest* etc. In this study, we dwell on Naïve Bayes classification technique. Naïve Bayes classifier provides an analysis tool that defined a set of pattern rule which categorizes data into different classes using probabilistic approach. Initially, it would first construct a model for

each of the class attributes as a function of other remaining attributes in datasets. Then, tries to correlate the class of every record using a previously designed model on unseen and even new data set (Manjusha *et al.,* 2015). This analysis aids with a good understanding of the data set and predicting future trend (Ameta & Jain, 2017).

### 2.2.3 Decision Tree Classifier

Decision tree is another classification algorithm that uses an organized hierarchical structure of a set of conditions to classify an instance. Decision tree is associated with a number of drawbacks such as the inability to have a representative object in real world, lack of quality measuring mechanism for attributes cost and value, ability to fail to classify in some instances e.g. when a class is dependent on a high number of attributes and such set of conditions are not set. Decision tree supports a predictive approach in machine learning and data mining. The leaves represent the classes while branches represent junctions leading to class label (Zorman, Štiglic, Kokol, & Malčić, 1997). The initial challenge in decision tree is the lack of a distinct method for selecting attributes to be used for constructing the tree from root node to a leave. This provides means for a various construct in which some construct can provide the opportunity for failing a given instance. Decision tree classifies data by following some path of stated satisfied conditions that start from the root of a tree to the leaf called class label (Romero *et al.,* 2008). For example, suppose we have the Table 2.1 as training data set and we want to classify new data set using decision tree classifier. The decision tree is shown in Figure 2.1.

**Table 2.1: Training data set**

| ID | Refund | Marital Status | Taxable Income | Cheat |
|---|---|---|---|---|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

Adopted from lecture note by Predrag Radivojac (2017).

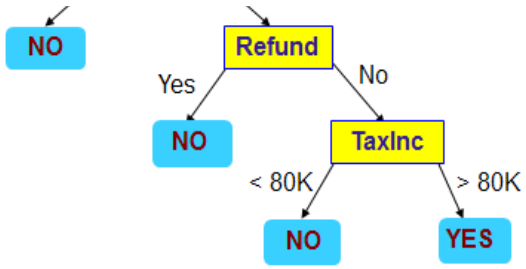



Figure 2.1 Decision tree for Table 2.1**.** Adopted from Predrag Radivojac (2017).

Supposes we have a test data set of tuple X with selections *Refund= Yes, Marital Status= single, Taxable Income = 80.* We could see that the decision tree above will classify the test data to "**No"** class.

### 2.2.4 Support Vector Machine (SVM)

SVM is a learning method that used the concept of computer science and statistics to analyze data and support pattern recognition. This approach is used in classification problem and nonlinear regression analysis. SVM is a non-probabilistic linear classifier which makes a prediction based on the set of accepted input, in which for every given input, there are two feasible classes that form the inputs (Raj & Prasanna, 2013). SVM was designed based on the principle of "Structural Risk Minimization principle" with the basic idea of finding hypothesis with the lowest minimum error e.g. error rate of a learner on data say training data set is restricted by the summation of the training-error rate (Ghumbre, Patil, & Ghatol, 2011). However, the drawback of this learner is that its computation is highly expensive thereby running slow on high data set and the classifier is also a binary classifier, therefore performing multi-class classification is done pair-wise (Madzarov, Gjorgjevikj & Chorbev, 2009), and similarly, SVM provides inability to present result in a transparent manner on high dimensional data (Auria & Moro, 2008; Karamizadeh, Abdullah, Halimi, Shayan & Rajabi, 2014).

**SVM Algorithm**

**C**onsider the training sample $\sum_{i=1}^{n}(x_i, y_i)$ in which $x_i$ is the input pattern for instance $i^{th}$, and $y_i$ is the matching target output. In pattern represented with the subset $y_i$ = +1 and the one represented by the subset $y_i$ = -1 is linearly separable. The equation in the form of a hyperplane which does the separation is

$$w^T x + b = 0 \quad (1)$$

Where,

x is an input vector,

w is the weight vector,

b is a bias. Thus,

$$w^T x_i + b \geq 0 \; \forall \; y_i = +1 \quad (2)$$

$$w^T x_i + b < 0 \; \forall \; y_i = -1 \quad (3)$$

For a given weight vector w and a bias b, the separation between the hyperplane defined in equation 3 and point with the closest data is said to be margin of separation and is represented with $\rho_0$ as shown in Figure 2.2, the construction of optimal geometrical hyperplane for a 2-dimensional input space (Ghumbre *et al.,* 2011).

.

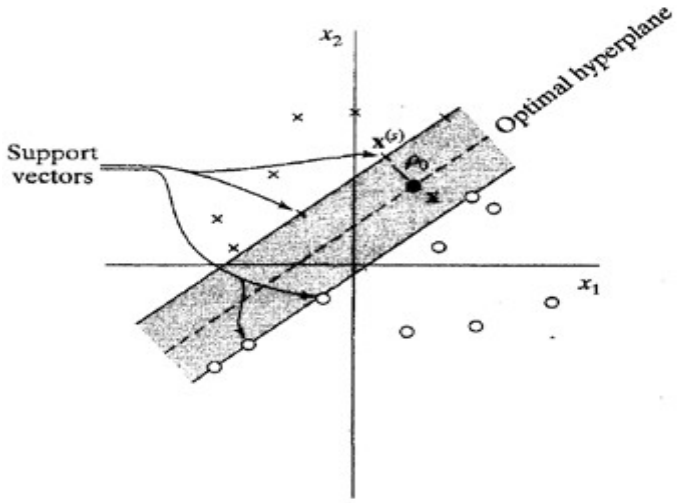



Figure 2.2 Ideal Hyperplane for a two (2) dimensional input space. Adopted (Ghumbre *et al.*, 2011).

The discriminant equation provides an arithmetical measure of distance from x to the ideal hyperplane for the optimum values of the weight vector and bias, accordingly.

$$g(x)=W_o^T x+b_o \quad (4)$$

## 2.3 Model of Prediction

Prediction is the process of studying the present and past states of the attribute in order to predict its future state (Medhekar, Bote & Deshmukh, 2013). Prediction consists of evaluating classification, pattern matching, trends, and relation through analyzing data of past instances or events to forecast future event using developed predictive model (Srinivas *et al.,* 2010). Data prediction involves two-step processes: first, the training of the model using a classifier in order to predict class label (predicted attribute) of a test data. Then secondly, evaluating performance accuracy of a predictor by calculating the error based on the differences between actual and predicted values for each tuple in test data (Baby & Priyanka, 2012). Predictive model of data mining tasks includes regression, classification, prediction and time series analysis (Tribhuvan *et al*., 2015). In this study, we focus on Naïve Bayes classification to predict malaria classes and its severe cases.

## 2.4 Review of Literature

Classification, a data mining technique, is the process of classifying and predicting the value of class attributes based on the values of predictors. These predictors are attributes that are used to predict new dataset (Sharma, Kumar, Panat & Karajkhede, 2015). In predictive systems, we always emphasize the extract of two group attributes: Predictors and target class attribute. Example of predictor attributes include Patient Name, Patient Id, age, blood group, gender, and Protein sequence, which are used to predict heart disease target attribute (Vijayarani & Deepa, 2014).

### 2.4.1 Prediction Systems Using Naïve Bayes technique

Usually, a predictive system is obtained through building a model by extracting rules and pattern in training data set and to use those extracted rule to predict the class of records whose class label is unidentified. In the early years, many researchers in the area of data mining have demonstrated a means of providing a system for enhancing accuracy and precision through developing automated predictive and diagnosis system of disease using various classification technique. To mention few related ones.

Implementation of HIV predictive model which helps to facilitate the knowledge of HIV among peoples in Addis Ababa (Zewdu & Beshah, 1998). The system uses J48, CART, and Naïve Bayes classification algorithms on data collected from Voluntary Counseling and Testing centers (VCT) Addis Ababa – Ethiopia. Raj and Prasanna (2013) proposed an automatic disease

identification model which could be converted into an integrated model to improve on text classification based on Machine Learning principle. The approach uses Natural Language Processing and Naïve Bayes technique of Machine Learning (ML) and the disease considered in the study includes; Malaria, Typhoid, Dengue, Tuberculosis, and Hepatitis B. However, this system dwell on Medline text as target parameter and does not consider checking accuracy as means of authenticating model. Similar study although from different phenomena by Vijayarani and Dhayanand (2015), were conducted on a predictive system of liver diseases such as Hepatitis, Cirrhosis, Bile Duct, and Liver Cancer using SVM and Naïve Bayes classification algorithm. The classifier utilizes large data set of Liver Function Test (LFT) from UCI database and compared their result based on classification, accuracy and execution time of both classifiers. However, some of the study conducted in classification focuses on biting issues in hospitals management such as models that predict patient hospitalization and discharge from emergency department based on the early medical record using text mining approach (Lucini *et al.,* 2017).

In the area of heart disease prediction, Masethe and Masethe (2014) conducted an experiment on early detection and prediction of heart disease using different classifications techniques such as Naïve Bayes, J48, CART, Bayesian Network and REPTREE. The study uses k-means clustering on a dataset from South Africa and demonstrates the prototype using Naïve Bayes to predict the chances of the patient getting a heart attack. The result shows a prediction accuracy of 97% using confusion matrix. Similarly, Indhumathi and Vijaybaskar (2015) designed an intelligence computer-based information and decision support system which can diagnose and predict Heart Disease using Naïve Bayes technique. The system works based on user response to complex queries and can be used to helps medical practitioners to make a clinical decision that cannot be made using the traditional system through providing effective treatment and reduces cost. However, the contribution made by Masethe and Masethe (2014), was only on a comparative study of four classification technique chosen.

Vembandasamy, Sasipriya and Deepa (2015) conducted a similar study by providing an analysis tool for predicting heart disease using Naïve Bayes algorithm. The tool was trained using 500 records of patients obtained from Chennai diabetic research institute considering attributes like Trestbps, Cholesterol, Fbs, Cp, Sex, and Age. The novelty in this study was the

fact that the tool can be used to reduce the number of tests that can be done in the detection of cardiovascular diseases.

Recently, Cherian and Bindu (2017) presented a technique for predicting the presence of heart disease using detail medical record of patient-supplied. The techniques used are Laplace smoothing and Naïve Bayes classification. The system assists through avoiding redundant diagnosis test conducted on a patient and some delay caused in starting proper treatment by speedily diagnosing heart disease in a patient. The system provides an alternative opinion concerning the patient's condition just like an experienced doctor since it is predicting from a historical database with a large number of heart patient records. The system can provide quality service at affordable cost and patient can equally use it if they have their medical test with them.

Predictive system using Naïve Bayes classification technique has demonstrated knowledgeable contributions even in the area of social and education events. A classification model to predict employees' performance as well as predicting new candidate performance using Cross Industry Standard Process for Data Mining (CRISP-DM), Naïve Bayes, and Decision tree technique was developed (Al-radaideh & Nagi, 2012). Similarly, a model for predicting student performance in order to differentiate between low learners and high learners using Naïve Bayes was developed (Bhardwaj & Pal, 2011). The necessity of predicting higher learner's students can assist in identifying slow learners so that teachers can assist the slow learners in improving their performances. This kind of models was used in a school as a tool for training medical student and nurses in diagnosing patients with heart diseases only (Patil, 2014). This type of analysis tools was equally devised for predicting students performance taking advantage of attributes such as aptitude test, student attendance, assignment, submission, GPA, class test marks, grade etc from student test record Database using Decision tree and Naïve Bayes algorithms (Tribhuvan *et al.*, 2015).

Recently, Tarekegn and Sreenivasarao (2016) designed a system to predict student placement in a higher institution using three data mining techniques: Random Forest, J48, and Naïve Bayes. The study uses student's entry data to build the model of prediction, and the model was evaluated using various cross-fold validations. Similarly, Saa (2016) discovered a qualitative model that classify and predict students' performance in a higher learning environment based on the correlation among social, personal and academic performance factors of students'. The research designed was done using Weka and RapidMiner on combined classifications of CART,

Naïve Bayes, CHAID, ID3 Decision tree, and C4.5 Decision tree techniques. The study provides novelty that can help lecturers/ instructors and student in executing improved higher educational quality through identifying students with a possible low performance at the beginning of the semester.

Naïve Bayes technique has been used in predicting several diseases such as the chance of getting Heart attack using blood pressure, sex, blood sugar, and age. This system was designed to work better with few dataset using an optimized version of Naïve Bayes called *One Dependency Augmented Naïve Bayes classifier (ODANB)* (Srinivas *et al.*, 2010). Some of these systems can actually predict the classes of heart attack such as low, average and high with an accuracy of prediction (Medhekar *et al*, 2013), while others take in test medical parameters as its input and used a trained model to predict classes using the same method.

### 2.4.2 Hybrid model of Prediction systems

Many developed predictive system has used combined techniques of preprocessing, clustering and classification called hybrid model to achieve a common goal of prediction. This is because research has shown the weakness of single model in providing an efficient model of prediction (Hakizimana, Wilson, Cheruiyot, Stephen & Nyararai, 2017). A study on a hybrid model for predicting dengue disease outbreak in Malaysia using ANN on attributes: *time series from dengue incidence data, proximity location,* and *rainfall data*. The result demonstrates that hybrid model provides more optimal result than the single model (Husin, Mustapha, Sulaiman & Yaakob, 2012). In Support of this confronting argument with a scholarly work, a predictive model for cost operative therapy information system was used in easing of diagnosing and decision support using three diverse supervised machine learning algorithms i.e., Bayesian Classifier, Decision Tree, and Neural Network to predict heart disease (Taneja, 2013) Recently, Langarizadeh and Moghbeli (2016) conducted a review of the application of Naïve Bayes Network (NBN) in predicting and improving disease diagnosis by physicians. The review demonstrates performance power of NBNs along with another classifier for papers between 2005 and 2015. The results have shown that NBN performed better than other classifiers in terms of accuracy, specificity, sensitivity, Receiver Operating Characteristic (ROC) and Area Under Curve (AUC).

Similarly, in the area of malaria disease prediction system, which is the narrowed phenomenon of interest in this thesis, Sharma *et al.* (2015) developed a model of malaria

outbreak prediction using machine learning which can serve as an early flagging tool to areas that are susceptible to malaria. This model was test-run using two classification algorithm: SVM and Artificial Neural Network (ANN). The model was tested for performance measure using Receiver Operating Characteristic (ROC).

Some of the contributions made in malaria disease were not actually on model development but reviewing all contributions made on the computer-based model of prediction and diagnosis of malaria fever (Oguntimilehin, Adetunmbi & Abiola, 2013). This review aims at studying present and future requirements to produce better viable classifiers in treatment and diagnosis of malaria. The study was categorized based on authors, methodology, approach, the image of parasites or symptoms consideration and motivation for the research. However, the study indicates the need for a better system such as the hybrid model of predicting malaria and also failed to comprehend on performance accuracy of each of the reviewed study.

Predictive system for malaria and typhoid co-infection based on symptom using Support Vector Machine (SVM) was developed by (Aminu, Ogbonnia & Shehu, 2016). Despite the drawback of SVM which includes: computation is highly expensive thereby running slow on high data set, and the classifier is also a binary, therefore performing multi-class classification is to be done pair-wise (Karamizadeh, Abdullah, Halimi, Shayan & Rajabi, 2014). The study investigates how malaria and typhoid were influenced by the same symptom; this means that the system predicts patients to either have malaria or typhoid based on same supplied parameters. The system achieved 80%-90% accuracy on cross-validation technique. However, this system considered influencing symptom, because some of the symptom used were fever and temperature together.

A recent research study by Iroezindu, Agaba, Okeke, Daniyam, and Isa (2012) and Kokori,

Inuwa, Babakura, and Garba (2016) have shown that fever is usually influenced by temperature. Hence, the need for a novel predictive system that predicts only malaria with its severe classes based on symptom supply using the method that supports independent assumption like Naïve Bayes (Hemanth *et al.,* 2011). Moreover, aligned with this assumption, Patil *et al*. (2016) proposed disease prediction framework using Naïve Bayes without any implantation and use of data set to experiment whether the proposed system will provide an optimal solution or not. This framework was designed to be non-specific to peculiar disease. The framework can be used in analyzing some diseases that can have different phases of classification like malaria.

In view of the same reasoning, Bohra, Arora, Gaikwad, Bhand, and Patil (2017) presented the same conceptual framework for predicting non-specific disease too based on symptoms supplied by the users with an even briefed explanation on how the implementation can be achieved using Naïve Bayes. Both contribution by Patil *et al*.(2016) and Bohra *et al*. (2017) presented only a conceptual framework for predicting non-specific disease based on symptoms supplied. However, the system proposed by Bohra *et al.* (2017) even requires Internet to be functional. Therefore, such system cannot be used in rural areas where there is no adequate medical expert and Internet facilities. Therefore, there is need for a system which delivers prediction services of most challenging diseases like malaria, hence such system should have devise mechanism for checking performance accuracy of prediction and perhaps be acceptable in healthcare domain.

### 2.4.3 Model Evaluation

Model is defined as an abstraction tool in research which reproduce the provisional image of the real object of study (Volkova, Kozlov, Mager & Chernenkaya, 2017). In disease predictive system, model is a machine learning tool build using a supervised learner (algorithm with automatic learning ability), which learned a pattern from experience and then use the pattern to predict a future event (Raj & Prasanna, 2013). This model is usually evaluated on performance measure using several mechanisms dedicated for evaluating the accuracy of classification model such as Confusion Matrix (CM), Receiver Operating Characteristic (ROC), Gain and Lift Chart, Gini Coefficient, Kolmogorov Smirnov Chart, Concordant, Matthews correlation coefficient etc (Vihinen, 2012). In this study, we choose to demonstrate the accuracy of our model using Confusion matrix and ROC. This is because confusion matrix is a standard matric and provides accuracy of model with other associated attributes like sensitivity, specificity, precision and F-measure (Hemanth *et al.,* 2011), while ROC was designed specifically to evaluate model performance of health-related study with the hope of providing cutoff value that minimizes the number of False Positive and False Negative which is the same as minimizing sensitivity and specificity (Vihinen, 2012; SigmaPlot, 2014).

The gap established in this thesis, the implementation of hybrid framework for predicting malaria using data mining techniques, is to the best of my knowledge novel, and the idea is in accordance with principle organized by giants in the area of data mining. The research has strictly targeted the symptoms of *P. falciparum* as it is most common and dangerous causes of

death (Al-Hassan & Roberts, 2002; Bartoloni & Zammarchi, 2012). The proposed framework has used the concept of data preprocessing techniques like cleaning, transformation, and Naïve Bayes classifier. The proposed framework has been tested for accuracy using Confusion Matrix (CM) and Receiver Operating Characteristic (ROC).

# CHAPTER THREE

# MATERIALS AND METHODS

This section explained the concept of classification as needed in the thesis. Elaborates on the discussion of Naïve Bayes methodology and highlighted its great advantage, requirement elicitation of the system (both hardware and software), use case diagram, the framework of prediction, data collection, sample size determination, and experimental setup demonstrating the methodology with an example. Finally, we highlight the mechanism of confusion matrix and ROC for checking the performance of the classifier.

## 3.1 Concept of Classification Technique

One of the techniques used in machine learning and data mining for prediction of disease is classification. Classification provides predictive data mining approach which makes a prediction about values of data using known results found from different data. The predictive models have the precise goal of allowing us to predict the unknown values of attributes of a target given known values of other attributes. Predictive modeling can be understood of as a learning function of mapping from an input set of vector measurements to a scalar output (Bhardwaj & Pal, 2011).

Several research studies have been conducted using classification algorithms. Recently in malaria, symptom-based predictive system of malaria and typhoid co-infection were developed using Support Vector Machine classifier (Aminu *et al.*, 2016). The study feels the need to investigate malaria and typhoid disease together because of their similarity in clinical manifestation. However, the study considers influencing symptoms like fever and temperature. Temperature is influenced by fever (Kokori *et al.*, 2016). In this study, we have developed a model for predicting malaria and it severe classes using Naïve Bayes Classifier. This classifier support independent assumption. The model has been trained and tested using large data set obtained from Federal Medical Center, Yola.

## 3.2 Naïve Bayes Classification

Naïve Bayes classifier works as both supervised learning and statistical based technique for classification (Hemanth *et al.,* 2011). It works based on Bayes' theorem through finding the probability of an event occurring given the probability of another event that has already occurred. It assumes a model which relies on probability to calculate uncertainty of future events

in such a principled mechanized way through estimating the probabilities of the events. Such mechanism has been widely used in prediction and diagnosis of diseases (Medhekar *et al.,* 2013). Naïve Bayes classification is simple and particularly suited when the dimensionality of the input is high. Despite its simplicity, it can outperform more sophisticated classification method. It provides perspective for understanding many learner algorithms and works on the assumptions that: is easy to construct, classifying categorical data, occurrences of an event (attributes) are independent and can be trained in a supervised manner (Patil *et al*., 2016). The major advantage of Naïve Bayes in classification is its simplicity and its ability to approximate probabilities for a class on any given instance (Kononenko, 1991).

### 3.3 Software Design Phase

The system was designed using Object-Oriented Programming (OOP) approach and use case diagram was used to present requirement in more details.

#### 3.3.1 Requirement Elicitation

The system is intended to serve as an assistive tool to clinicians in rural area hospitals where there are no adequate health services. It can also be used by any other individual who wishes to inquire the status of their malaria. Therefore, the users are expected to supply the value of their parameters (*fever, headache, nausea, vomiting, respiratory distress, convulsion,* and *coma*). The system would automatically predict the status of a user's.

#### 3.3.2 Outline of the Requirement

i. User opens the system.
ii. A dashboard appears with different menus and sub-menus.
iii. User can add record to database, view existing record, and view the model.
iv. The user can equally predict his/her malaria status after supplying their parameter and view performance of the two classification phase using confusion matrix.
v. User can also view graphical performance of the classifier using ROC curve.

#### 3.3.3 Software Requirement

The software is designed using Java programming language since it supports Weka build-library of all the package needed e.g. Naïve Bayes classifier library, Confusion matrix, and ROC. This language was chosen over other in order to have maximum control on interface design, better presentation of statistical result and flexibility.

### 3.3.4 Hardware Requirement

The minimum hardware requirements include the following:

1. Windows 8.7 or 10, 64 bits (PC or Mac computers).
2. All CPU (Intel family or Xeon).
3. 4 GB RAM or above, 20GB HDD Free Space.

## 3.4 Use Case of the System

This illustrates the list of action or event that normally defined the interaction between the actors and use cases which occur in the system. In this case, the actors are the clinicians or any individual user. Figure 3.1 shows the use case diagram of the system.


**Figure 3.1:** ***Use Case Diagram: Architect of the System***

## 3.5 Naïve Bayes Algorithm

Let B be the training data set. Suppose each tuple is represented by n-dimensional attribute vector $Y= (Y_1, Y_2, . . ., Y_n)$, which represents 'n' measurement on the tuple from 'n' attribute $A_1, A_2, A_n$.

Suppose that there are N classes: $C_1$, $C_2$, $C_{3...}$ $C_n$. Given a tuple Y, the classifier will predict that Y belongs to the class having the highest probability (P) condition on X. i.e.

X belong to class $C_1$ if and only if

$$P(C_k|Y) > P(C_i|Y) \quad \text{for } 1 \leq i \leq n, i \neq k \tag{5}$$

The algorithm will maximize (6) and (7)

$$P(C_1|Y)=P(Y|C_1)*P(C_1) / P(Y) \tag{6}$$

$$P(C_2|Y) = P(Y|C_2)*P(C_2) / P(Y) \tag{7}$$

Hence,

$P(C_1|Y)> P(C_2|Y)$ if and only if

$P(Y|C_1)*P(C_1)> P(Y|C_2)*P(C_2)$, since P(Y) is same in both the cases.

Given dataset with many attributes, it will be expensive to compute $P(Y|C_1)$. Therefore, we assume that the values of the attribute are conditional independent. Thus,

$P(Y|C_1) = \Pi P(Y_i|C_1)$. Therefore,

$$P(Y|C_1) = P(Y_1|C_1) \times P(Y_2|C_2) \times P(Y_3|C_3) \times \ldots \times P(Y_n|C_n) \tag{8}$$

### 3.6 The Proposed Framework of Prediction

This study proposed a model that would mimic the trends of predicting malaria in a suspected patient. The major steps involved in the prediction system will include; general data collection, data cleaning and transformation, classification and prediction, and finally, interpretation, evaluation and knowledge discovery. Fig. 3.2 shows the proposed framework of our study.

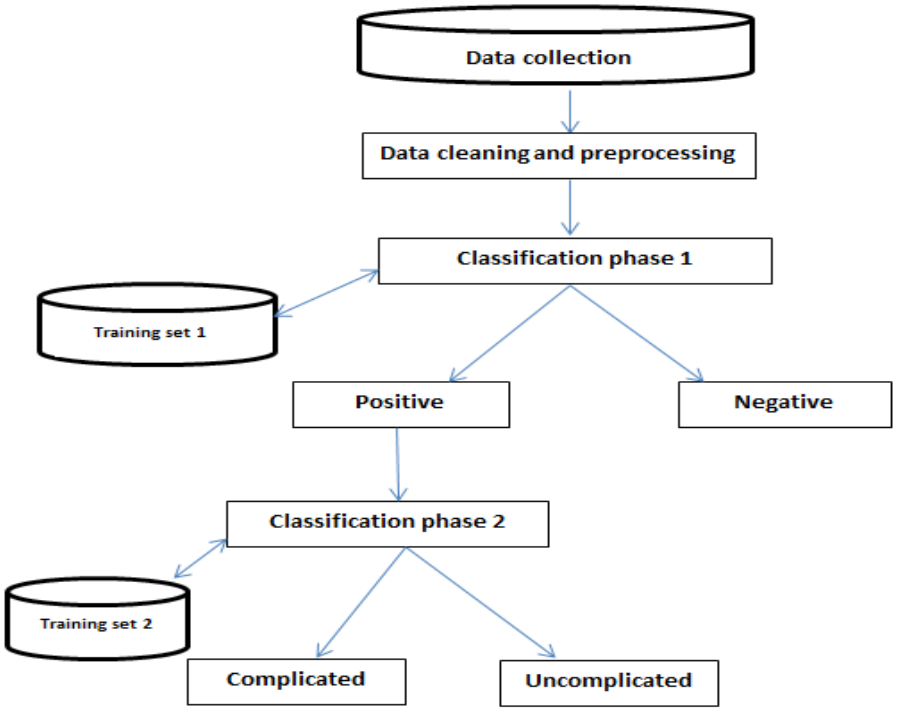



Figure *3.2 Prediction frameworks*

### 3.6.1 Prediction Procedure

**Data Collection:** The data collected would be divided into two portions; one portion of the data is extracted as a training set, while the other portion would be used for testing. The training portion is taken from a table stored in a database (say Table 1) is called as data1 which is *training set1*, while the training portion taking from another table stored in a database (called Table 2) is called as data 2 which is *training set2*.

**Data Preprocessing:** Data preprocessing was done to remove noise, and outlier.

**Transformation**: the data was transformed from analog to electronic record.

**Classification and prediction:** Based on the nature of variable in our dataset, we will use Naïve Bayes classification techniques twice; *Classification phase 1* and *Classification phase 2*. Working of the framework is illustrated as follows:

i. Data collection and preprocessing are done.
ii. Preprocessed data is stored in a training set 1 and training set 2. These datasets are used during classification.
iii. Test data set is stored in database test data set.
iv. Part of test data set is compared for classification using classifier 1 and rest part is classified using classifier 2 as follows:

*Classifier phase 1:* classify into positive or negative class label. If the patient is having malaria, then the patient is classified as positive (P), while a patient is classified as negative (N) if the patient does not have malaria.

*Classifier phase 2:* classify only that data set that has been classified as positive by classifier 1, and then further classify them into complicated and uncomplicated class label.

The system would be designed in such a way that the core parameters as a determining factor should be supplied their value. The *nature* of the data collected from the hospital is shown in Table 4.1 (reference Chapter 4).

### 3.7 Data Collection and Sample Size technique

Before collecting the data, the researcher was guided with all ethical training certification on data collection, right to confidentiality and privacy reserved called Institutional Review Board (IRB). Data was collected from the manual archive of the Hospital using stratified sampling technique then transformed the data to electronic form and stored in MYSQL database called *malaria*. Each patient file was extracted and reviewed for signs and symptoms of malaria then check for laboratory confirmation result from diagnosis. The data is divided into two tables: the first table is called *data* which contain data used in *phase 1* of the classification, while the second table called *data2* which contain data used in *phase 2* of the classification.

#### 3.7. 1 Sample Size Determination

Naïve Bayes technique requires that data should be as high as possible because its accuracy depends on how high the volume of the data (Qian, Zhou, Yan, Li, & Han, 2015). We understand that the hospital has ~10,000 manually archived records of malaria cases. We used the formula developed by Krejcie and Morgan, (1970) to understand minimum sample size required for the study.

$$\mathbf{s} = X^2NP\ (1 - P) \div \boldsymbol{d^2}\ \boldsymbol{(N} - 1) + X^2P\ (1 - \boldsymbol{P)}, \quad (9)$$

**s** = the required sample size.

$X^2$ = is the table value of chi-square for 1 degree of freedom at confidence level (3.841).

*P* = is population proportion (assumed to be .50 since this would give the max. sample size).

***N*** = is population size.

***d*** = is a degree of accuracy expressed as a proportion (.05).

*Sample size*

(S) = (3.841) x (10000) x 0.50(1-0.50) ÷ $(0.05)^2$ x (10000-1) + (3.841) x 0.50(1-0.50) = 385.6

A total of 700 cleaned preprocessed records were collected and stored in database say Table1. According to a popular scholar, Sordo & Zeng (2005) which states that a sample size of ~150 -~8500 should be adequate for training while testing set of 10 - 60 should be adequate for a classifier performance measure (Indira, Vasanthakumari, Jegadeeshwaran, & Sugumaran, 2015). In this study, out of 700 records collected, 650 was used for training the model in *classification phase1*, out of which 414 were found positive and was used for training the model in *classification phase2*. During performance testing of the classifier, 50 records sample was drawn from the initial 700 population as a validation set. Out of the 414 records that were found positive, also 50 records were drawn for validating the performance of phase 2 model in accordance (Indira *et al.,* 2015).

### 3.8 Experimental Setup

A portion of real data was used for training the model. We have two training set. First training set for classifying patient to either have malaria (positive) or not (negative). The second training set would further classify those that were positive in the first classification as either having complicated or uncomplicated malaria. Using naïve Bayes classification technique, we have predicted a new patient having a set of parameter. Naïve Bayes works as follows.

Using,

$$Naive\ Bayes; P(t/y) = \frac{P(y/t)P(t)^{\square}}{P(y)} \tag{10}$$

Where t: is the class

$P(t|y)$: is a posterior prob. of *class* given *predictor*.

$P(t)$: is the past (prior) prob. of *class*.

$P(y|t)$: the prob. of *predictor* given *class*.

$P(y)$: past prob. of the *predictor*.

**Positive class label (N):** Patient may have malaria (positive) if the probability of selected features points out that the probability of positive class is greater than negative class.

$$P(Positive/patient)=\frac{P(patient/P)P(P)^{\square}}{P(patient/P)*p(P)+p(patient/N)*p(N)}$$

(11)

**Negative class label (N):** Patient may not have malaria (negative) if the probability of selected features points out that the probability of negative class is greater than positive class.

$$P(Nagetive/patient)=\frac{P(patient/N)P(N)^{\square}}{P(patient/P)*p(P)+p(patient/N)*p(N)}$$

(12)

**Complicated class label (C):** Patient may have a complicated case of malaria if the probability of selected features point out that the probability of complicated class is greater than uncomplicated class.

$$P(Complicated/patient)=\frac{P(patient/c)P(C)^{\square}}{P(patient/C)*p(C)+p(patient/U)*p(U)}$$

(13)

**Mild or uncomplicated class label (U)**: patients may have an uncomplicated case of malaria if the probability of selected features points out that the probability of uncomplicated class is greater than complicated class.

$$P(Uncomplicated/patient)=\frac{P(patient/U)P(U)^{\square}}{P(patient/C)*p(C)+p(patient/U)*p(U)}$$

(14)

### 3.9 Example 1

Using a given training dataset like the one in Table 3.1 and Table 3.2, suppose we have a new patient with the following parameter;

Test dataset X = (*fever=1, headache=1, nausea=1, vomiting= 0, & respiratory dist. =1, convulsion=1, coma=0).*

After computing the probabilities of event with reference to Table 3.1, we get:

P(X | positive) = 0.0384 and P(X | negative) = 0.01992.

Therefore, X is classified as having malaria since the P(X | positive) > P(X | negative).

And likewise,

P(X | complicated) = 0.09 and P(X | Uncomplicated) = 0.024.

Since P(X | complicated) > P(X | complicated). Then X is classified as having complicated malaria.

Table 3.1 Sample of Training data1

| #ID | Fever | Headache | Nausea | Vomiting | Class Label |
|---|---|---|---|---|---|
| Fmc/01 | 1 | 1 | 0 | 0 | P |
| Fmc/02 | 1 | 0 | 1 | 0 | P |
| Fmc/03 | 0 | 1 | 0 | 1 | P |
| Fmc/04 | 1 | 0 | 1 | 0 | P |
| Fmc/05 | 1 | 0 | 0 | 1 | P |
| Fmc/06 | 1 | 0 | 0 | 0 | N |
| Fmc/07 | 0 | 1 | 0 | 1 | N |
| Fmc/08 | 1 | 0 | 1 | 0 | N |
| Fmc/09 | 0 | 1 | 0 | 1 | N |
| Fmc/10 | 0 | 0 | 1 | 0 | N |

*#ID: Identity of each tuple*
*Attributes: Fever, Headache, Nausea, and Vomiting*
*Class Label: p* = Positive, N* = Negative*

Table 3.2 Sample of Training data 2

| #ID | Respiratory Distress | Convulsion | Coma | Class Label |
|---|---|---|---|---|
| Fmc/01 | 1 | 0 | 1 | C |
| Fmc/02 | 0 | 1 | 0 | U |
| Fmc/03 | 0 | 0 | 1 | U |
| Fmc/04 | 1 | 1 | 0 | C |
| Fmc/05 | 0 | 1 | 1 | C |
| Fmc/06 | 1 | 0 | 0 | U |
| Fmc/07 | 0 | 0 | 1 | U |
| Fmc/08 | 0 | 1 | 0 | U |
| Fmc/09 | 1 | 1 | 0 | C |
| Fmc/10 | 0 | 1 | 1 | C |

#ID: Identity of each tuple
Attributes: Respiratory Distress, Convulsion, and Coma
Class Label: C* = Complicated, U* = Uncomplicated.

### 3.10 Performance Measure of Classifier

#### 3.10.1 Accuracy Check

The accuracy of Naïve Bayes algorithm can be checked using confusion matrix (Kohavi and Provost, 1998) and Receiver Operating Characteristic (ROC) analysis.

#### 3.10.2 Confusion Matrix

$$Confusion¿\begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad (15)$$

Where,

"a" is the *N0* of accurate predictions that a given instance is "-",

"b" is the *N0* of incorrect predictions that a given instance is "+",

"c" is the *N0* of incorrect of predictions that a given instance "-",

"d" is the *N0* of accurate predictions that a given instances "+".

A confusion matrix is a table that describes the performance of a classifier. Confusion matrix demonstrates the accuracy of a solution to a given classification. It contains information about the predicted and actual classifications done by a classifier system. The performance of the model is normally evaluated using the data in the confusion matrix. For a given number of classes say M, the confusion matrix is R×Y matrix such that $C_{ik}$ indicates the number of records from F which were assigned to class $C_{ik}$ instead of assigning the correct class $C_i$. A perfect solution would have zeros (0) through the diagonal of the matrix. However, below are few parameters of interest in the matrix.

Accuracy: is the ability of a model to appropriately predict the class label of previously unseen data or new data. It is a measure of how well the classifier makes a prediction on average. A good classification algorithm will try to minimize the number of times it makes the wrong prediction.

$$\text{Accuracy} = (TP + TN)/n. \quad (16)$$

Where n is the number of observation.

True Positive Rate (TPR) or Sensitivity (Recall): A true positive is when the outcome of a prediction is said 'P' and the classifier have actually predicted the value to be same 'P'. It is a measure of comprehensiveness or magnitude.

True Positive Rate (TPR) or sensitivity = ∑ true positive / ∑ conditional positive

True positive rate (TPR) = TP/(FN + TP). (17)

True Negative (TNR) or Specificity: is also called *specificity*, it indicates the number of tuples classified as false while they were actually false. It is also the ability of the classifier to classify those that were false correctly:

True negative rate (TNR) = TN/ (TN + FP). (18)

False Positive (FPR): is when a record is classified to be false while is actually supposed to be predicted as true.

False positive rate (FPR) = FP/ (TN + FP). (19)

False Negative (FNR): It signifies the number of tuples classified as false while they were actually true.

False negative rate (FNR) = FN/ (FN + TP). (20)

Precision: Precision is the portion of retrieved cases that are relevant. It is the measure of correctness or excellence.

Precision = TP/ (TP + FP). (21)

Kappa Statistic: Is a multivariate discrete method for conveying total accuracy through relating two sources of data: How far is the classification different from a random matrix. Kappa statistic is used as a mechanism for measuring *interrater reliability* (Freeman & Moisen, 2008). It measures the rate at which the data used in the study are a true or correct representation of their variance measured. Kappa ranges value between -1 and +1. Research study suggested that a bench score *0.41* and above should be accepted for any health-related research (McHugh, 2012). However, a range of values has been provided with their respective grade. Values ≤ 0 as signifying no agreement, 0.01–0.20 as none to slight, 0.21–0.40 - fair, 0.41– 0.60 - moderate, 0.61–0.80 - substantial, while 0.81–1.00 nearly perfect agreement.

$$K = Pr(a) – Pr(e) /1- Pr(e) \quad (22)$$

Where,

Pr(a) = Probability of success classification ( accurate)

Pr(e) = Probability of success due to chance

Mean Absolute Error (MAE): is another mechanism that is used to measure the performance of a model**.** Both RMSE and MAE are appropriate means of evaluating model performance because MAE also provides the same attributes to all error in the system as it describes an aspect of error in model performance. MAE always provide smaller value than RMSE (Chai & Draxler, 2014). MAE is better in describing uniformly distributed error.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} ¿ e_i \vee ¿ \quad (23)$$

Where,

N = assumed to have n sample error

$e_i$ = is calculated as i= 1, 2, 3, 4,…n and assume error sample to be unbiased.

Root Mean Square Error (RMSE): is a standard statistical metric that is used to measure the performance of a model. For example: meteorology, climate change, and air quality research to measure model performance. The only difference between RMSE and MAE is that RMSE Penalizes variance because it provides more weight of errors to larger absolute values than smaller absolute values (Chai & Draxler, 2014).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \quad (24)$$

Relative Absolute Error (RAE): this can be used to compare between models in which errors are estimated in the different units.

$$RAE = \frac{\sum_{i=1}^{n} ¿ P_i - a_i \vee ¿}{\sum_{i=1}^{n} ¿ \bar{a} - a_i \vee ¿ ¿} ¿ \quad (25)$$

Where,

a = actual class

p = predicted class

Root Relative Square Error: this is an analogy of Root Absolute Error (RAE) and is estimated by MAE divided by dividing with the error of the Naïve Bayes classifier (i.e. classifier which disregard every predictor and simply chooses the most recurrent value).

$$RAE = \sqrt{(P_1 - a_1) + \ldots + (P_n - a_1)} / (a_1 - \tilde{a})^2 + \ldots + (a_n - \tilde{a})^2 \quad (26)$$

### 3.10.3 ROC and Area under Curve

This is a graphical display of a measure of performance in machine learning (Vihinen, 2012). It is used in medicine to decide the limit value of a clinical test. ROC is simply a plot of true positive rate (TPR) on the vertical axis and false positive rate (FPR) on the horizontal axis (Hemanth *et al.*, 2011). This signifies the relationship between true positive and true negative rates of a classifier through Minimizing sensitivity (false positive) and specificity (false negative) (Vihinen, 2012; Patil & Sherekar, 2013). ROC curves are assessed by either smearing the classification rule on test dataset with known classes or by using a sample of re-used method e.g. cross-validation (Hand & Till, 2001). It presents better performance in a number of ways through decreasing standard error as both the number of test sample and Area Under Curve (AUC) increases, and increases sensitivity when performing analysis of variance test (Lobo, Jiménez-valverde, & Real, 2008). ROC provides the cost of a different kind of misclassification in classifying data using classification rule (Hemanth *et al.*, 2011). A good classification rule design by a classifier is reflecting on the ROC curve by lying at the upper left triangle of the square area ( Vihinen, 2012). Therefore, in this study, we use ROC as an additional means of presenting detail clarity in demonstrating the performance of the classifier.

The AUC is a term that is defined with respect to ROC. It simply refers to the area under ROC curve. The AUC is a measure of how good prediction performed (Vihinen, 2012). It presents common index summary of the information contained within the curve. AUC can be regarded as a gauge for the quality of separation (Taneja, 2013). For example, illustrated in Figure 3.3, the point (0, 1) signifies perfect classifier: it means that the classifier classifies every negative and positive case correctly. It is referred as (0, 1) because the FPR is 0 (zero), and the TPR is 1 (perfect), and the point (0, 0) signifies a classifier which predicts every case as negative, while the point (1, 1) relates to the classifier which predicts all case to be positive. Point (1, 0) is the classifier that is not correct in every classification.

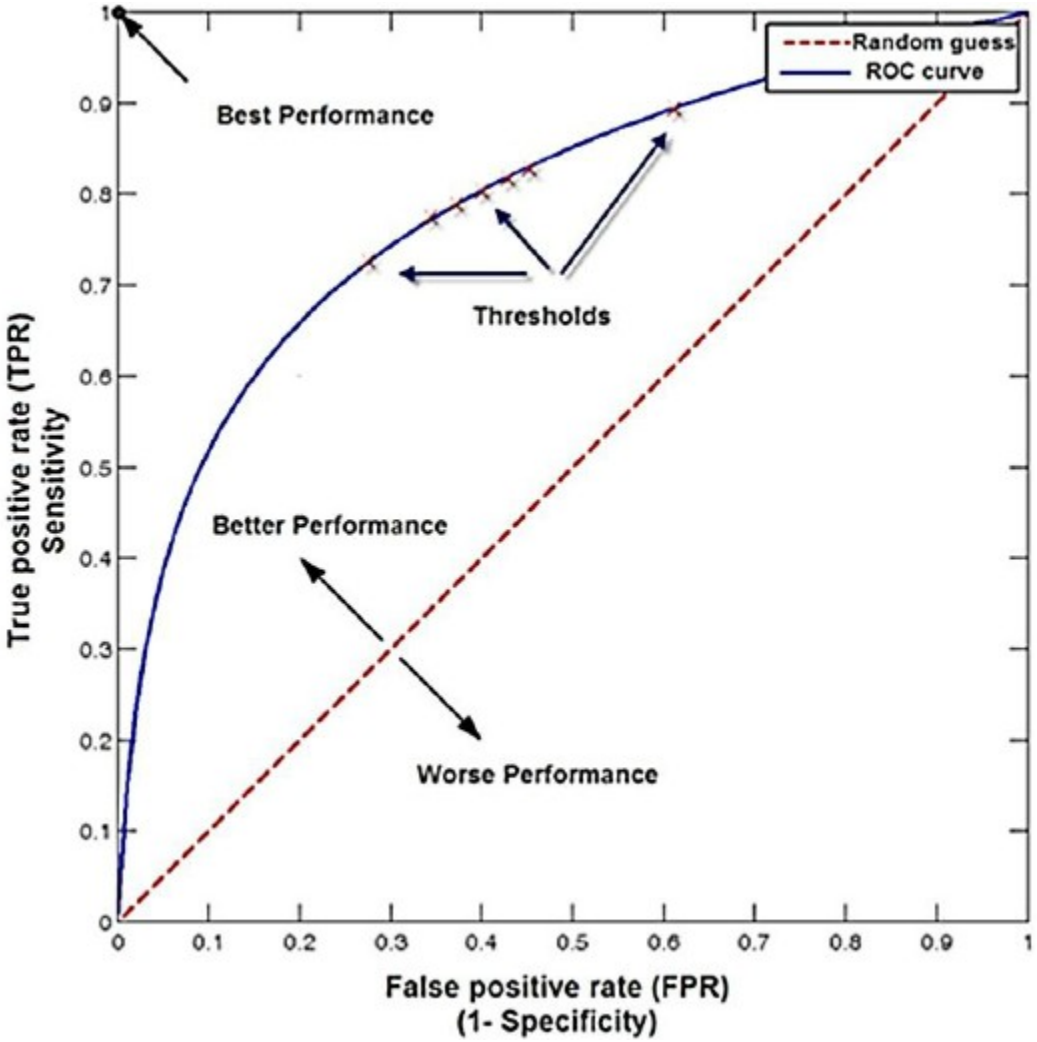



Figure *3.3 Example of ROC.* Adopted from Hassouna, Tarhini, Elyas and AbouTrab (2016)

# CHAPTER FOUR

# RESULTS AND DISCUSSION

This chapter discusses the result of our findings, used a screenshot to present the design of our implementation which includes the models used, training data, prediction interface, data entry form and confusion matrix along with ROC-AUC for performance evaluation of the classifier.

## 4.1 Presentation of Results

The prediction framework has been implemented using Java programming language built on Weka 3.8.0 (Software code is attached as Appendix I). Two Naïve Bayes model *Data1 model classifier* and *Data2 model classifier* using Training set1 and Training Set2 respectively was used. A total sample record of 700 was taken from the hospital and used in the training. The designed system is user-friendly in which users can supply their parameters, and then the system would automatically predict the status of their malaria and their complication if necessary. Figure 4.1 shows the main window, Table 4.1 shows training data set 1& 2, Figure 4.2 shows form for adding data (training data), Figure 4.3 and 4.4 show the two Naïve Bayes models, Figure 4.5 prediction interface with confusion matrix alongside, while Figure 4.6 and 4.7 shows the result of ROC. Table 4.2 represent summary of the results.

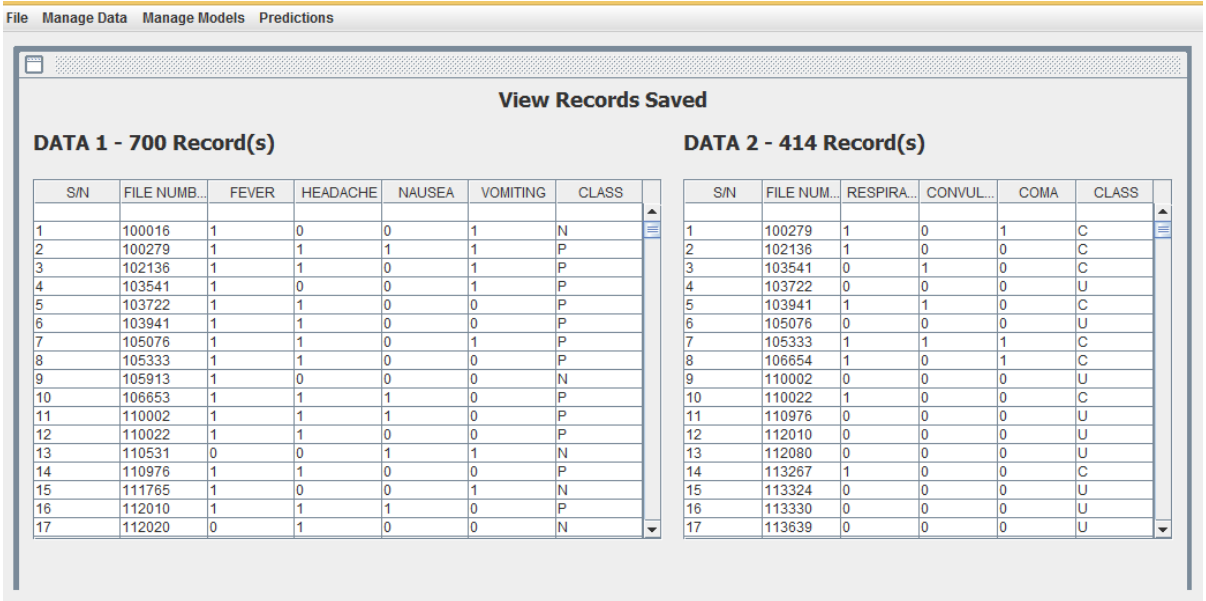

Figure *4.1: Main Window*

The system is designed in such a way that ser is required to click on proceed button to log in to the main menu.

Table 4.1 Training set 1 & 2

***Table 1:***
***#File Number****: Identity of each tuples / patient*
***Attributes****: fever, headache, Nausea and Vomiting*
***Class Label:*** *P*= Positive, N*= Negative.*

***Table2:***
*#**File Number:** Identity of tuples*
***Attributes:*** *Respiratory distress, convulsion, and Coma*
***Class Label:*** *C*= Complicated, U*= Uncomplicated*




| S/N | FILE NUMB... | FEVER | HEADACHE | NAUSEA | VOMITING | CLASS |
|---|---|---|---|---|---|---|
| 1 | 100016 | 1 | 0 | 0 | 1 | N |
| 2 | 100279 | 1 | 1 | 1 | 1 | P |
| 3 | 102136 | 1 | 1 | 0 | 1 | P |
| 4 | 103541 | 1 | 0 | 0 | 1 | P |
| 5 | 103722 | 1 | 1 | 0 | 0 | P |
| 6 | 103941 | 1 | 1 | 0 | 0 | P |
| 7 | 105076 | 1 | 1 | 0 | 1 | P |
| 8 | 105333 | 1 | 1 | 0 | 0 | P |
| 9 | 105913 | 1 | 0 | 0 | 0 | N |
| 10 | 106653 | 1 | 1 | 1 | 0 | P |
| 11 | 110002 | 1 | 1 | 1 | 0 | P |
| 12 | 110022 | 1 | 1 | 0 | 0 | P |
| 13 | 110531 | 0 | 0 | 1 | 1 | N |
| 14 | 110976 | 1 | 1 | 0 | 0 | P |
| 15 | 111765 | 1 | 0 | 0 | 1 | N |
| 16 | 112010 | 1 | 1 | 1 | 0 | P |
| 17 | 112020 | 0 | 1 | 0 | 0 | N |

| S/N | FILE NUM... | RESPIRA... | CONVUL... | COMA | CLASS |
|---|---|---|---|---|---|
| 1 | 100279 | 1 | 0 | 1 | C |
| 2 | 102136 | 1 | 0 | 0 | C |
| 3 | 103541 | 0 | 1 | 0 | C |
| 4 | 103722 | 0 | 0 | 0 | U |
| 5 | 103941 | 1 | 1 | 0 | C |
| 6 | 105076 | 0 | 0 | 0 | U |
| 7 | 105333 | 1 | 1 | 1 | C |
| 8 | 106654 | 1 | 0 | 1 | C |
| 9 | 110002 | 0 | 0 | 0 | U |
| 10 | 110022 | 1 | 0 | 0 | C |
| 11 | 110976 | 0 | 0 | 0 | U |
| 12 | 112010 | 0 | 0 | 0 | U |
| 13 | 112080 | 0 | 0 | 0 | U |
| 14 | 113267 | 1 | 0 | 0 | C |
| 15 | 113324 | 0 | 0 | 0 | U |
| 16 | 113330 | 0 | 0 | 0 | U |
| 17 | 113639 | 0 | 0 | 0 | U |

### 4.1.1 Add Record

The clinician can decide to add more training dataset using the Register New Medical Record form. Patient's parameter should be supplied and click save record button to add the record to the database.


Figure *4.2 Add Record Form*

### 4.1.2 Designed Naïve Bayes Model

The Naïve Bayes Model built using training data set 1. This model is used in *classification Phase1* to predict patient as Positive (P) or Negative (N) during prediction.

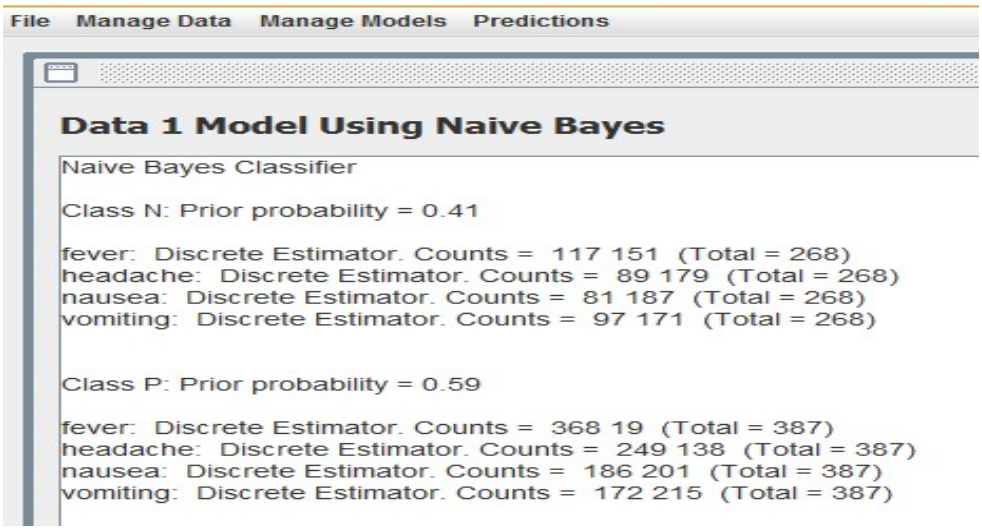



Figure *4.3 Data 1 Model Built using training set1*

Data 2 Model is designed using training dataset 2 and is used in further predicting those classified as positive from Data 1 model.

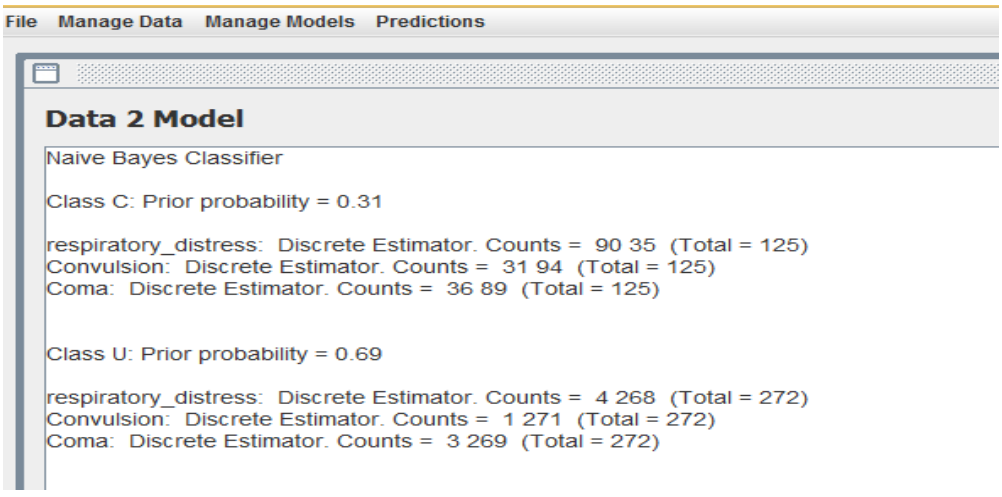



Figure 4.4 *Data 2 model built using training set2*

### 4.1.3 Prediction Interface

In prediction, the user is first required to supply the status of their parameters. The first parameters are whether the patient has the following parameter; *fever, headache, nausea*, and *vomiting,* then click on predict. The system would use the designed Naïve Bayes models to learn and predict the class label of the patient automatically based on the training and as well as displaying the confusion matrix. If the patient is classified as "Positive" then the system will further request for more parameter: *respiratory distress, coma,* and *convulsion* in order to classify the complication of the patient.

Here, let's say the patient has selected that he has fever, headache, vomiting, but does not have nausea. The system predicted the status to be Positive (P). Then, further selected that he has respiratory distress, coma, but no convulsion. The system predicted the status as Complicated (C).

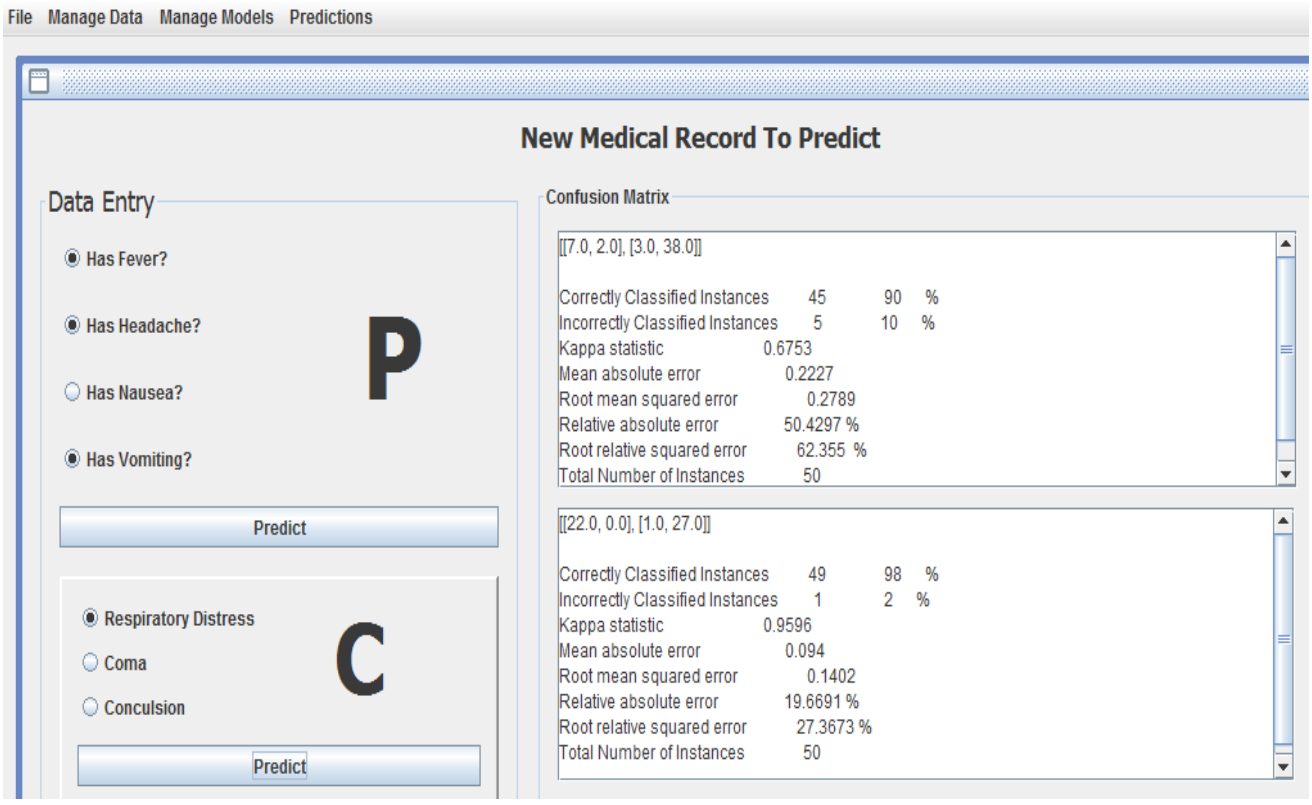



Figure *4.5: Predicting interface*

### 4.1.4 Performance Result

To demonstrate the accuracy of the system, we divide the data into 650 records for training and 50 records for the validation set in phases. The system has achieved 90% accuracy of a correctly classified instance during prediction in *classification Phase1* and 98% accuracy of a correctly classified instance during *classification Phase2, while* incorrectly classified instances of 10% (5) and 2 % (1) respectively.

*Confusion Matrix*

Classification phase1 Model $confusion¿\begin{vmatrix} 7.0 & 2.0 \\ 3.0 & 38.0 \end{vmatrix}$ (27)

Where,

7.0 is the *N0* of accurate predictions that a given instance is "-",

3.0 is the *N0* of incorrect predictions that a given instance is "+",

2.0 is the *N0* of incorrect of predictions that a given instance "-",

38.0 is the *N0* of accurate predictions that a given instances "+".

Accuracy = (TP + TN)/n. (28)

= (38.0+7.0)/50 = 90%

Sensitivity = (TPR) = TP/ (FN + TP). (29)

= 38/ (38 + 3.0) = 92.6%

Specificity = (TNR) = TN/ (TN + FP). (30)

= 7.0/ (7.0 + 2.0) = 77.7%

False positive rate (FPR) = FP/ (TN + FP). (31)

= 2.0 / (7.0 + 2.0) = 22.2%

False negative rate (FNR) = FN/ (FN + TP). (32)

= 3.0/ (3.0 + 38.0) = 7.3%

Precision = TP/ (TP + FP). (33)

= 38.0 / (38.0 + 2.0) = 95%

Classification phase2: $confusion¿\begin{vmatrix} 22.0 & 0.0 \\ 1.0 & 27.0 \end{vmatrix}$ (34)

Accuracy = (TP + TN)/n. (35)

$$(22.0 + 27.0)/50 = 98\%$$

Sensitivity = (TPR) = TP/ (FN + TP). (36)

$$= 27.0/ (1.0 + 27.0) = 96.4\%$$

Specificity = (TNR) = TN/ (TN + FP). (37)

$$= 22.0/ (22.0 + 0.0) = 100\%$$

False positive rate (FPR) = FP/ (TN + FP). (38)

$$= 0.0 / (22.0 + 0.0) = 0.0\%$$

False negative rate (FNR) = FN/ (FN + TP). (39)

$$= 1.0/ (1.0 + 27.0) = 3.5\%$$

Precision = TP/ (TP + FP). (40)

$$= 27.0 / (27.0 + 0.0) = 100\%$$

Table 4.2 Performance Summary

| **Classifier** | **Correctly classified Accuracy (in %)** | **Incorrectly classified (in %)** | **Kappa Statistic (in %)** | **Mean Absolute Error (in %)** | **Root Mean Sq. Error (in %)** | **Relative Absolute Error (in %)** | **Root Relative Sq. Error (in %)** | **Number Instance (in %)** |
|---|---|---|---|---|---|---|---|---|
| NB Data1 model | 90 | 10 | 67 | 22 | 27 | 50 | 62 | 50 |
| NB Data2 model | 98 | 2 | 95 | 9 | 14 | 19 | 27 | 50 |

*ROC and AUC Result*


Figure *4.6 Accuracy using ROC for classification phase 1*

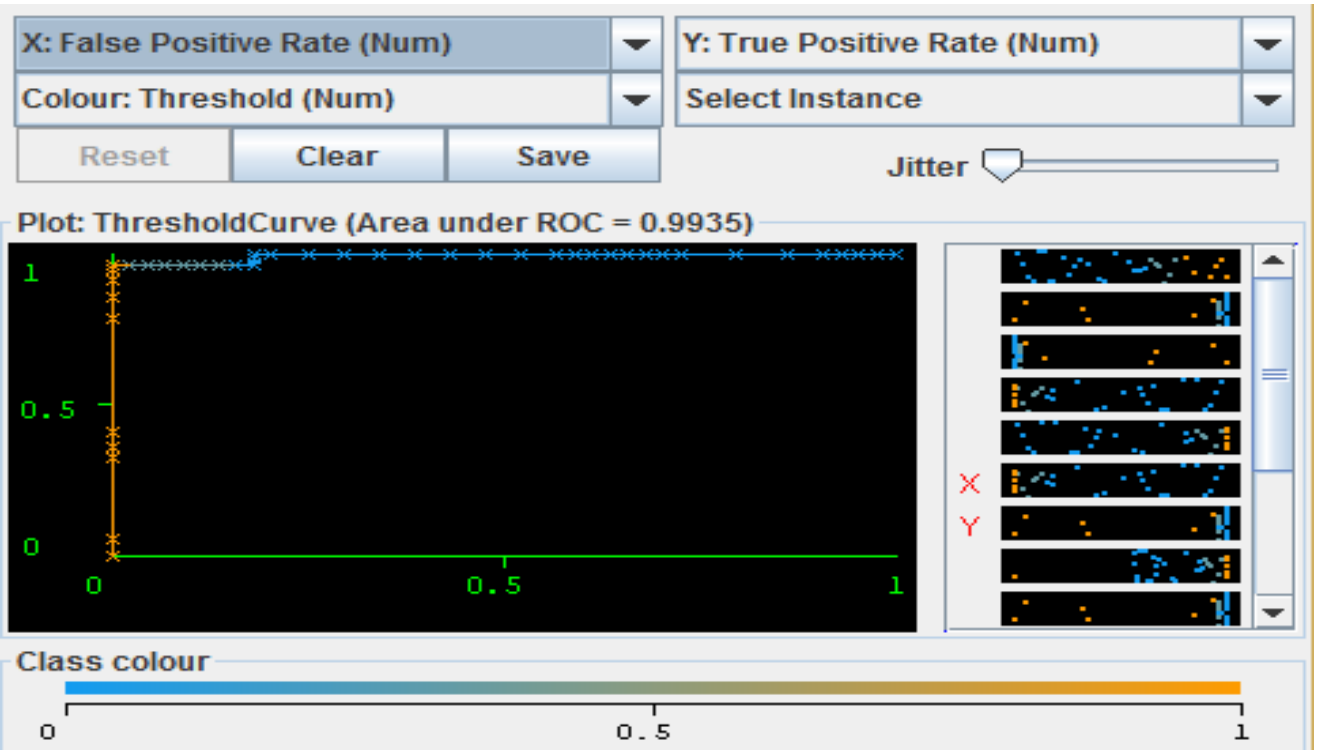



Figure *4.7 Accuracy using ROC for classification phase* 2

## 4.2 Our Contribution

This research uses a supervised learning algorithm on large data sets of 700 patients (approximately) obtained from Federal Medical Center, Yola (FMC, Yola) hospital to predict malaria in any suspected patient using various data processing task and Naive Bayes classification technique. The experiment is conducted in Java programming environment built on Weka version 3.8.0 library.

# CHAPTER FIVE

# SUMMARY, CONCLUSION, AND RECOMMENDATIONS

Finally, this chapter discusses the report of the entire thesis, highlighted the importance of our finding, recommendation, limitation of the study and the future work.

## 5.1 Summary

Classification is indeed a reliable technique in data mining for classifying and predicting the class label of unknown data. Naïve Bayes classification technique is one of the classification algorithms that predict well on high dataset like the one used in this study. In this thesis, we apply Naïve Bayes Learning algorithm to train a model using malaria dataset obtained from hospital. This technique considered each attribute to be independent. The data used constitute an attribute of *fever, headache, nausea, vomiting, respiratory distress, convulsion* and *coma.*

The study built two model, the first model was built for predicting the presence of malaria, while the second model was built for predicting the severity of those found with the parasite from first model. Both model predict using clinical manifestation by suspected malaria patient. Performance accuracy of the two model was checked using confusion matrix and Receiver Operating Characteristic (ROC).

## 5.2 Conclusion

The techniques used in this research demonstrated a better performance. The parameters used in this study are distinct, non-influencing by one another, and support the independent assumption of Naïve Bayes. For better understanding, the classification is divided into *Classification Phase1* and *Classification Phase2*. The first four parameters are used in *classification phase1* to predict the status of malaria in patients, while the remaining three are used in classification Phase 2 for predicting the complication of those found with malaria from Phase 1.

The system is designed to accept input of parameters from the user and automatically predict the class label based on the training data. This data was collected from medical record of patients suspected and diagnosed in hospital. A total of 700 records were collected, out of which 414 were diagnosed as positive. Two Naïve Bayes model was built, one in each classification phase.

In *Classification Phase 1*, 650 records were used in training the model while 50 records were used in testing the accuracy. In *classification phase 2*, 364 records were used in training model while 50 were used as a validation set. During prediction, the system will also automatically display the performance accuracy of the model.

Performance accuracy of the model was checked using confusion matrix, ROC and other derived evaluation attributes such as sensitivity, specificity, and precision. The performance accuracy of the two models was summarized in Table 4.2 and Figure 4.6, 4.7. From the result of the findings, the first model achieved better accuracy of 90% and 98% correctly classified instances, 10% and 2% incorrectly classified instance respectively using confusion matrix, while ROC demonstrated a more optimal result than confusion matrix. In general, the entire result demonstrated that the system can be efficient and reliable in predicting malaria.

### 5.3 Recommendations

a. The system can be used in rural area (environment) where there is limited or inadequate medical facility.
b. The system would assist the clinician in understanding the various complications that may result from malaria cases. Therefore it can serve as an early flagging tool in examining the severity of malaria in patients such as cerebral malaria signs.
c. In the area where there are clinicians but no adequate medical facilities, the system can serve as an assistive tool to clinician through guiding them on decision making during malaria prediction using clinical manifestation.
d. It can equally be used by an individual like you and I that are willing to predict the likelihood or status of their malaria.

### 5.4 Future Work

This study can further be experimented on other similar diseases and another classification technique as well as on more attributes for better performance. The design can be achieved using different programming language. The performance of the classifier could be check using other performance evaluation method design for checking the accuracy of a model such as *Gain and Lift Chart, Gini Coefficient, Kolmogorov Smirnov Chart, Concordant (Discordant Ratio), Cross Validation.*

# REFERENCES


Al-Hassan, N. A., & Roberts, G. T. (2002). Patterns of presentation of malaria in a tertiary care institute in Saudi Arabia. *Saudi Medical Journal*, *23*(5), 562–567.

Al-Radaideh, Q. A., & Al Nagi, E. (2012). Using data mining techniques to build a classification model for predicting employees performance. *International Journal of Advanced Computer Science and Applications*, *3*(2), 144-151.

Ameta, M. A., & Jain, M. K. (2017). Data mining techniques for the prediction of kidney diseases and treatment : *A Review*, *6*(2), 20376–20378.

Aminu, F, Ogbonnia, E. O., & Shehu, I. S. (2016). A predictive symptoms-based system using support vector machines to enhanced classification accuracy of malaria and typhoid coinfection. *International Journal of Mathematical Sciences and Computing*, *2*(4), 54–66.

Archana, S., & Elangovan, K. (2014). Survey of classification techniques in data mining. *International Journal of Computer Science and Mobile Applications*, *2*(2), 65-71.

Arévalo-Herrera, M., Lopez-Perez, M., Medina, L., Moreno, A., Gutierrez, J. B., & Herrera, S. (2015). Clinical profile of Plasmodium falciparum and Plasmodium vivax infections in low and unstable malaria transmission settings of Colombia. *Malaria Journal*, *14*(1), 154.

Auria, L., & Moro, R. A. (2008). Support Vector Machines (SVM) as a technique for solvency analysis. *DIW Berlin*, (August), 1–16.

Baby, M. N., & Priyanka, L. T. (2012). Customer classification and prediction based on data mining technique, *2*(12).

Bartoloni, A., & Zammarchi, L. (2012). Clinical aspects of uncomplicated and severe malaria. *Mediterranean Journal of Hematology and Infectious Diseases*, *4*(1).

Bhardwaj, B. K. (2011). Data Mining : A prediction for performance improvement using classification. *(IJCSIS) International Journal of Computer Science and Information Security, 9*(4).

Bohra, H., Arora, A., Gaikwad, P., Bhand, R., & Patil, M. R. (2017). Health prediction and medical diagnosis using Naive Bayes. *Ijarcce*, *6*(4), 32–35.

Calderaro, A., Piccolo, G., Gorrini, C., Rossi, S., Montecchini, S., Dell'Anna, M., … Arcangeletti, M. (2013). Accurate identification of the six human Plasmodium spp. causing

imported malaria, including Plasmodium ovale wallikeri and Plasmodium knowlesi. *Malaria Journal*, *12*(1), 321.

Cdc, C. F. D. C. and P. (2013). Treatment of Malaria ( Guidelines For Clinicians ). *Treatment of Malaria (Guidelines for Clinicians)*, (July), 1–8.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247-1250.

Chaurasia, V., & Pal, S. (2013). Early prediction of heart diseases using data mining techniques. *Caribbean Journal of Science and Technology*, *1*, 208-217.

Cherian, V., & Bindu, M. S. (2017). Heart disease prediction using Naïve Bayes algorithm and laplace smoothing technique, *5*(2), 68–73.

Chotivanich, K., Silamut, K., & Day, N. P. (2007). Laboratory diagnosis of malaria infection-A short review of methods. *New Zealand Journal of Medical Laboratory Science*, *61*(1), 4.

Cruz, J. A., & Wishart, D. S. (2006). Applications of machine learning in cancer prediction and prognosis. *Cancer informatics*, *2*, 59.

Dangare, C. S., & Apte, S. S. (2012). Improved study of heart disease prediction system using data mining classification techniques. *International Journal of Computer Applications*, *47*(10), 44-48.

Hemanth, K. S., Vastrad, C. M., & Nagaraju, S. (2011). Data mining technique for knowledge discovery from engineering materials data sets. *Advances in Computer Science and Information Technology*, 512-522.

Durairaj, M., & Ranjani, V. (2013). Data mining applications in healthcare sector: a study. *International Journal Of Scientific & Technology Research*, *2*(10), 29-35.

Freeman, E. A., & Moisen, G. G. (2008). A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa. *Ecological Modelling*, *217*(1), 48-58.

Ghumbre, S., Patil, C., & Ghatol, A. (2011). Heart disease diagnosis using Support Vector Machine. *International Conference on Computer Science and Information Technology (ICCSIT'2011)*, 84–88.

Gomes, A. P., Vitorino, R. R., Costa, A. D. P., Mendonça, E. G. D., Oliveira, M. G. D. A., & Siqueira-Batista, R. (2011). Severe Plasmodium falciparum malaria. *Revista Brasileira de Terapia Intensiva*, *23*(3), 358-369.

Gu, X., Chen, H., & Yang, B. (2015, October). Heterogeneous data mining for planning active surveillance of malaria. In *Proceedings of the ASE BigData & SocialInformatics 2015* (p. 34). ACM.

Hakizimana, L., Cheruiyot, W. K., Kimani, S., & Nyararai, M. (2017). A hybrid based classification and regression model for predicting diseases outbreak in datasets. *International Journal of Computer (IJC)*, *27*(1), 69-83.

Hand, D. J., & Till, R. J. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine learning*, *45*(2), 171-186.

Howes, R. E., Reiner Jr, R. C., Battle, K. E., Longbottom, J., Mappin, B., Ordanovich, D., ... & Smith, D. L. (2015). Plasmodium vivax transmission in Africa. *PLoS neglected tropical diseases*, *9*(11), e0004222.

Hassouna, M., Tarhini, A., Elyas, T., & AbouTrab, M. S. (2016). Customer Churn in Mobile Markets A Comparison of Techniques. *arXiv preprint arXiv:1607.07792*.

Husin, N. A., Mustapha, N., Sulaiman, M. N., & Yaakob, R. (2012, September). A hybrid model using genetic algorithm and neural network for predicting dengue outbreak. In *Data Mining and Optimization (DMO), 2012 4th Conference on* (pp. 23-27). IEEE.

Idro, R., Marsh, K., John, C. C., & Newton, C. R. (2010). Cerebral malaria: mechanisms of brain injury and strategies for improved neurocognitive outcome. *Pediatric research*, *68*(4), 267-274.

Indira, V., Vasanthakumari, R., Jegadeeshwaran, R., & Sugumaran, V. (2015). Determination of minimum sample size for fault diagnosis of automobile hydraulic brake system using power analysis. *Engineering Science and Technology, an International Journal*, *18*(1), 59-69.

Indhumathi, S., & Vijaybaskar, G. (2015). Web based health care detection using naive Bayes algorithm. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *4*(9), 3532-36.

Iroezindu, M. O., Agaba, E. I., Okeke, E. N., Daniyam, C. A., Isa, S. E., & Akindigh, M. T. (2012). Relationship Between Fever and Malaria Parasitaemia in Adults: Does HIV Infection Make any Difference?. *Journal of Medicine in the Tropics*, *14*(2), 103-108.

Jothi, N., Rashid, N. A., & Husain, W. (2015). Data mining in healthcare - A Review. *Procedia Computer Science*, *72*, 306–313.

Srinivas, K., Rani, B. K., & Govrdhan, A. (2010). Applications of data mining techniques in healthcare and prediction of heart attacks. *International Journal on Computer Science and*

*Engineering (IJCSE)*, *2*(02), 250-255.

Karamizadeh, S., Abdullah, S. M., Halimi, M., ShaJothyan, J., & javad Rajabi, M. (2014, September). Advantage and drawback of support vector machine functionality. In *Computer, Communications, and Control Technology (I4CT), 2014 International Conference on* (pp. 63-65). IEEE.

Kokori, M., Inuwa, A. M., Babakura, M., & Garba, A. M. (2016). Body temperature trends and fever risk in the parasitaemia of Plasmodium Falciparum treated children at lake-alau , borno state , *North Eastern Nigeria, 5*(9), 1684–1689.

Kononenko, I. (1991). Semi-naive Bayesian classifier. In *Machine Learning—EWSL-91* (pp. 206-219). Springer Berlin/Heidelberg.

Krejcie, R. V., & Morgan, D. W. (1970). Determining sample size for research activities. *Educational and psychological measurement*, *30*(3), 607-610.

Kaur, H., & Wasan, S. K. (2006). Empirical study on applications of data mining techniques in healthcare. *Journal of Computer science*, *2*(2), 194-200.

Kohavi, R., & Provost, F. (1998). Glossary of terms. *Machine Learning*, *30*(2-3), 271-274.

Kriegel, H. P., Borgwardt, K. M., Kröger, P., Pryakhin, A., Schubert, M., & Zimek, A. (2007). Future trends in data mining. *Data Mining and Knowledge Discovery*, *15*(1), 87-97.

Laishram, D. D., Sutton, P. L., Nanda, N., Sharma, V. L., Sobti, R. C., Carlton, J. M., & Joshi, H. (2012). The complexities of malaria disease manifestations with a focus on asymptomatic malaria. *Malaria Journal*, *11*(1), 29.

Langarizadeh, M., & Moghbeli, F. (2016). Applying Naive Bayesian networks to disease prediction: a systematic review. *Acta Informatica Medica*, *24*(5), 364.

Lobo, J. M., Jiménez-Valverde, A., & Real, R. (2008). AUC: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, *17*(2), 145-151.

Lubezky, N., Ben-Haim, M., Nakache, R., Lahat, G., Blachar, A., Brazowski, E., ... & Klausner, J. M. (2010). Clinical presentation can predict disease course in patients with intraductal papillary mucinous neoplasm of the pancreas. *World journal of surgery*, *34*(1), 126.

Lucini, F. R., S. Fogliatto, F., C. da Silveira, G. J., L. Neyeloff, J., Anzanello, M. J., de S. Kuchenbecker, R., & D. Schaan, B. (2017). Text mining approach to predict hospital admissions using early medical records from the emergency department. *International Journal of Medical Informatics*, *100*, 1–8.

Madzarov, G., Gjorgjevikj, D., & Chorbev, I. (2009). A multi-class SVM classifier utilizing binary decision tree. *Informatica*, *33*(2).

Manjusha, K. K., Sankaranarayanan, K., & Seena, P. (2015). Data mining in dermatological diagnosis: A method for severity prediction. *International Journal of Computer Applications*, *117*(11).

Masethe, H. D., & Masethe, M. A. (2014, October). Prediction of heart disease using classification algorithms. In *Proceedings of the world Congress on Engineering and computer Science* (Vol. 2, p. 2224).

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia medica*, *22*(3), 276-282.

Medhekar, D. S., Bote, M. P., & Deshmukh, S. D. (2013). Heart disease prediction system using naive Bayes. *Int. J. Enhanced Res. Sci. Technol. Eng*, *2*(3).

Mohapatra, B. N., Jangid, S. K., & Mohanty, R. (2014). GCRBS score: a new scoring system for predicting outcome in severe falciparum malaria. *The Journal of the Association of Physicians of India*, *62*(1), 14–17.

Mutanda, A. L., Cheruiyot, P., Hodges, J. S., Ayodo, G., Odero, W., & John, C. C. (2014). Sensitivity of fever for diagnosis of clinical malaria in a Kenyan area of unstable, low malaria transmission. *Malaria Journal*, *13*(1), 163.

Ndyomugyenyi, R., Magnussen, P., & Clarke, S. (2007). Diagnosis and treatment of malaria in peripheral health facilities in Uganda: findings from an area of low transmission in south-western Uganda. *Malaria Journal*, *6*(1), 39.

Nikam, S. S. (2015). A comparative study of classification techniques in data mining algorithms. *Oriental Journal of Computer Science & Technology*, *8*(1), 13-19.

Oguntimilehin, A., Adetunmbi, A. O., & Abiola, O. B. (2013). A Machine Learning Approach to clinical diagnosis of typhoid fever. *A Machine Learning Approach to Clinical Diagnosis of Typhoid Fever*, *2*(4), 671-676.

Patil, R. R. (2014). Heart disease prediction system using Naive Bayes and Jelinek-mercer smoothing. *International Journal of Advanced Research in Computer and Communication Engineering*, *3*(5), 2278-1021.

Patil, R., Chopade, P., Mishra, A., Sane, B., & Sargar, Y. (2016). Disease prediction system using data mining hybrid approach. *Communications on Applied Electronics Published by Foundation of Computer Science (FCS), NY, USA*, *4*(9), 48–51.

Patil, T. R., & Sherekar, S. S. (2013). Performance analysis of Naive Bayes and J48 classification algorithm for data classification. *International Journal of Computer Science and Applications*, *6*(2), 256-261.

Pirnstill, C. W., & Coté, G. L. (2015). Malaria diagnosis using a mobile phone polarized microscope. *Scientific reports*, *5*, 13368.

Qian, Y., Zhou, W., Yan, J., Li, W., & Han, L. (2014). Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery. *Remote Sensing*, *7*(1), 153-168.

Rai, N., & Abraham, J. (2012). Different Clinical Features of Malaria. *Asian Journal of Biomedical and Pharmaceutical Sciences*, *2*(12), 28.

Raj, T. F. M., & Prasanna, S. (2013). Implementation of ML using naïve bayes algorithm for identifying disease-treatment relation in bio-science text. *Research Journal of Applied Sciences, Engineering and Technology*, *5*(2), 421–426.

Srinivas, K., Rani, B. K., & Govrdhan, A. (2010). Applications of data mining techniques in healthcare and prediction of heart attacks. *International Journal on Computer Science and Engineering (IJCSE)*, *2*(02), 250-255.

Razzak, M. I. (2015). Automatic detection and classification of malarial parasite. *International Journal of Biometrics and Bioinformatics (IJBB)*, *9*(1), 1–12.

Romero, C., Ventura, S., Espejo, P. G., & Hervás, C. (2008, June). Data mining algorithms to classify students. In *Educational Data Mining 2008*.

Saa, A. A. (2016). Educational Data Mining & Students' Performance Prediction. *International Journal of Advanced Computer Science & Applications*, *1*, 212-220.

Sala al-Din Abdullah, A. (2016). Using Data mining techniques to identify the causes of deaths in al-gedaref hospital. *European Journal of Computer Science and Information Technology*, *4*(2), 1–8.

Sharma, V., Kumar, A., Lakshmi Panat, D., & Karajkhede, G.(2015). Malaria outbreak prediction model using Machine Learning. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume*, *4*.

Shinde, R., Arjun, S., Patil, P., & Waghmare, P. J. (2015). An intelligent heart disease prediction system using K-Means Clustering and Naïve Bayes Algorithm. *International Journal of Computer Science and Information Technologies*, *6*(1), 637–639.

SigmaPlot. (2014). ROC Curves Analysis. *SigmaPlot*, 1–20. Retrieved from

http://www.sigmaplot.com/products/sigmaplot/ROC_Curves_Analysis.pdf

Soni, J., Ansari, U., Sharma, D., & Soni, S. (2011). Predictive data mining for medical diagnosis: An overview of heart disease prediction. *International Journal of Computer Applications*, *17*(8), 43-48.

Sordo, M., & Zeng, Q. (2005). On sample size and classification accuracy: a performance comparison. *Biological and medical data analysis*, 193-201.

Stauffer, W., & Fischer, P. R. (2003). Diagnosis and treatment of malaria in children. *Clinical infectious diseases*, *37*(10), 1340-1348.

Taneja, A. (2013). Heart disease prediction system using data mining techniques. *Oriental Journal of Computer science and technology*, *6*(4), 457-466.

Tarekegn, G. B. (2016). Application of data mining techniques to predict students placement in to Departments. *International Journal of Research Studies in Computer Science and Engineering*, *3*(2), 10–14.

Tribhuvan, A. P., Tribhuvan, P. P., & Gade, J. G. (2015). Applying Naive Bayesian classifier for predicting performance of a student using Weka. *Advances in Computational Research*, *7*(1), 239.

Vembandasamy, K., Sasipriya, R., & Deepa, E. (2015). Heart diseases detection using Naive Bayes algorithm. *International Journal of Innovative Science, Engineering & Technology*, *2*(9), 441–444.

Vihinen, M. (2012). How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. *BMC genomics*, *13*(4), S2.

Vijayarani, S., & Deepa, S. (2014). Naïve Bayes Classification for Predicting Diseases in Haemoglobin Protein Sequences. *International Journal of Computational Intelligence and Informatics*, *3*(4).

Vijayarani, S., & Dhayanand, S. (2015). Liver disease prediction using SVM and Naïve Bayes algorithms. *International Journal of Science, Engineering and Technology Research (IJSETR)*, *4*(4), 816-820.

Volkova, V. N., Kozlov, V. N., Mager, V. E., & Chernenkaya, L. V. (2017, May). Classification of methods and models in system analysis. In *Soft Computing and Measurements (SCM), 2017 XX IEEE International Conference on* (pp. 183-186). IEEE.

Zewdu, T. (1998). Prediction of HIV Status in Addis Ababa using Data Mining Technology.

Zorman, M., Štiglic, M. M., Kokol, P., & Malčić, I. (1997). The limitations of decision trees and

automatic learning in real world medical decision making. *Journal of Medical Systems*, *21*(6), 403-415.

# APPENDIX I

**DB Connection**

```
public class DBConnect {

   public static Statement stmt = null;
   private static Connection con = null;

   public static Statement connectDB() throws ClassNotFoundException, SQLException {
      if (stmt == null) {
         try {
            Class.forName("com.mysql.jdbc.Driver");
            con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/malaria", "root", "");
            stmt = (Statement) con.createStatement();
         } catch (Exception e) {
            System.out.println(e.getMessage() + " During Connection");
         }
      }
      return stmt;
   }

   public static int saveNewRecord(String fileno, String fever, String headache, String nausea,
String vomiting, String class1) {
      int result = -1;
      //String classed = "N";
      try {
         Statement connectDB = connectDB();
         String sql = "insert into data (file_no,fever, headache, nausea,vomiting,class1) values"
               + " ('" + fileno + "','" + fever + "','" + headache + "','" + nausea + "','" + vomiting +
"','" + class1 + "')";
         result = connectDB.executeUpdate(sql);
      } catch (Exception ex) {
         ex.printStackTrace();
      }
      return result;
   }

   public static void saveNewDataRecord(String fileno, String rd, String conv, String comm,
String cla) {
      try {
         Statement connectDB = connectDB();
```

```
            String sql = "insert into data2 (file_no,respiratory_distress, convulsion, coma,class2)
values"
                    + " ('" + fileno + "','" + rd + "','" + conv + "','" + comm + "','" + cla + "')";
            int executeUpdate = connectDB.executeUpdate(sql);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    public static ResultSet loadData1() {
        ResultSet executeQuery = null;
        try {
            Statement connectDB = connectDB();
            String sql = "select * from data";
            executeQuery = connectDB.executeQuery(sql);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        return executeQuery;
    }

    public static ResultSet loadData2() {
        ResultSet executeQuery = null;
        try {
            Statement connectDB = connectDB();
            String sql = "select * from data2";
            executeQuery = connectDB.executeQuery(sql);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        return executeQuery;
    }

}
```

**Naïve Bayes data model1**

```
package FramesPanel;
public class NaiveBayesData1Model extends javax.swing.JInternalFrame {
            public NaiveBayesData1Model() {
                    try {
                        initComponents();
                        jTextArea1.setText(UtilityClass.getNaiveBayesData1Model().toString());
                        } catch (Exception ex) {
                        Logger.getLogger(NaiveBayesData1Model.class.getName()).log(Level.SEVERE,
            null, ex);
                    }
                }
```

**Naïve Bayes data model 2**

```
package FramesPanel;
```

```
public class NaiveBayesData2Model extends javax.swing.JInternalFrame {
        public NaiveBayesData2Model() {
              try {
                 initComponents();
                 jTextArea1.setText(UtilityClass.getNaiveBayesData2Model().toString());
        } catch (Exception ex) {
        Logger.getLogger(NaiveBayesData2Model.class.getName()).log(Level.SEVERE, null,
ex);
      }
   }
private javax.swing.JLabel jLabel1;
   private javax.swing.JScrollPane jScrollPane1;
   private javax.swing.JTextArea jTextArea1;
}
```

**View Records**

```
public class ViewRecords extends javax.swing.JInternalFrame {
        public ViewRecords() {
                    initComponents();
                    loadAllData();
           }

        private javax.swing.JTable data1tabled;
                  private javax.swing.JTable data2table;
                  private javax.swing.JLabel jLabel1;
                  private javax.swing.JLabel jLabel2;
                  private javax.swing.JLabel jLabel3;
                  private javax.swing.JScrollPane jScrollPane1;
                  private javax.swing.JScrollPane jScrollPane2;

private void loadAllData() {
      try {
         ResultSet data1 = DBConnect.loadData1();
         DefaultTableModel model = (DefaultTableModel) data1tabled.getModel();
         int i =1;
         while(data1.next()){
            model.addRow(new Object[]{i++, data1.getString("file_no"), data1.getString("fever"),
data1.getString("headache"),
                data1.getString("nausea"), data1.getString("vomiting"),data1.getString("class1")});
         }
         jLabel3.setText(jLabel3.getText()+" - "+i+" Record(s)");
         data1 = DBConnect.loadData2();
         model = (DefaultTableModel) data2table.getModel();
         i=1;
         while(data1.next()){
            model.addRow(new Object[]{i++, data1.getString("file_no"),
data1.getString("respiratory_distress"),
                data1.getString("convulsion"), data1.getString("coma"),data1.getString("class2")});
         }
```

```
            jLabel2.setText(jLabel2.getText()+" - "+i+" Record(s)");
        } catch (SQLException ex) {
            Logger.getLogger(ViewRecords.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

**Visualize ROC1**

```
public class ROC1 implements RevisionHandler {
        public static final String RELATION_NAME = "ThresholdCurve";
        public static final String TRUE_POS_NAME = "True Positives";
        public static final String FALSE_NEG_NAME = "False Negatives";
        public static final String FALSE_POS_NAME = "False Positives";
        public static final String TRUE_NEG_NAME = "True Negatives";

public static final String FP_RATE_NAME = "False Positive Rate";
public static final String TP_RATE_NAME = "True Positive Rate";
public static final String PRECISION_NAME = "Precision";
public static final String RECALL_NAME = "Recall";
public static final String FALLOUT_NAME = "Fallout";
public static final String FMEASURE_NAME = "FMeasure";
public static final String SAMPLE_SIZE_NAME = "Sample Size";
public static final String LIFT_NAME = "Lift";
public static final String THRESHOLD_NAME = "Threshold";

        public Instances getCurve(FastVector predictions) {

            if (predictions.size() == 0) {
               return null;
            }
            return getCurve(predictions,
                  ((NominalPrediction) predictions.elementAt(0))
                        .distribution().length - 1);
          }
public Instances getCurve(FastVector predictions, int classIndex) {

      if ((predictions.size() == 0)
            || (((NominalPrediction) predictions.elementAt(0))
                  .distribution().length <= classIndex)) {
         return null;
      }

      double totPos = 0, totNeg = 0;
      double[] probs = getProbabilities(predictions, classIndex);
for (int i = 0; i < probs.length; i++) {
         NominalPrediction pred = (NominalPrediction) predictions.elementAt(i);
         if (pred.actual() == Prediction.MISSING_VALUE) {
            System.err.println(getClass().getName()
```

```
            + " Skipping prediction with missing class value");
        continue;
    }
    if (pred.weight() < 0) {
        System.err.println(getClass().getName()
            + " Skipping prediction with negative weight");
        continue;
    }
    if (pred.actual() == classIndex) {
        totPos += pred.weight();
    } else {
        totNeg += pred.weight();
    }
}

Instances insts = makeHeader();
int[] sorted = Utils.sort(probs);
TwoClassStats tc = new TwoClassStats(totPos, totNeg, 0, 0);
double threshold = 0;
double cumulativePos = 0;
double cumulativeNeg = 0;

for (int i = 0; i < sorted.length; i++) {

    if ((i == 0) || (probs[sorted[i]] > threshold)) {
        tc.setTruePositive(tc.getTruePositive() - cumulativePos);
        tc.setFalseNegative(tc.getFalseNegative() + cumulativePos);
        tc.setFalsePositive(tc.getFalsePositive() - cumulativeNeg);
        tc.setTrueNegative(tc.getTrueNegative() + cumulativeNeg);
        threshold = probs[sorted[i]];
        insts.add(makeInstance(tc, threshold));
        cumulativePos = 0;
        cumulativeNeg = 0;
        if (i == sorted.length - 1) {
            break;
        }
    }

    NominalPrediction pred = (NominalPrediction) predictions.elementAt(sorted[i]);

    if (pred.actual() == Prediction.MISSING_VALUE) {
        System.err.println(getClass().getName()
            + " Skipping prediction with missing class value");
        continue;
    }
    if (pred.weight() < 0) {
        System.err.println(getClass().getName()
            + " Skipping prediction with negative weight");
```

```
            continue;
          }
          if (pred.actual() == classIndex) {
            cumulativePos += pred.weight();
          } else {
            cumulativeNeg += pred.weight();
          }
}
          if (tc.getFalseNegative() != totPos || tc.getTrueNegative() != totNeg) {
          tc = new TwoClassStats(0, 0, totNeg, totPos);
          threshold = probs[sorted[sorted.length - 1]] + 10e-6;
          insts.add(makeInstance(tc, threshold));
        }

        return insts;
    }
          public static double getNPointPrecision(Instances tcurve, int n) {

        if (!RELATION_NAME.equals(tcurve.relationName())
              || (tcurve.numInstances() == 0)) {
          return Double.NaN;
        }
        int recallInd = tcurve.attribute(RECALL_NAME).index();
        int precisInd = tcurve.attribute(PRECISION_NAME).index();
        double[] recallVals = tcurve.attributeToDoubleArray(recallInd);
        int[] sorted = Utils.sort(recallVals);
        double isize = 1.0 / (n - 1);
        double psum = 0;
        for (int i = 0; i < n; i++) {
          int pos = binarySearch(sorted, recallVals, i * isize);
          double recall = recallVals[sorted[pos]];
          double precis = tcurve.instance(sorted[pos]).value(precisInd);

while ((pos != 0) && (pos < sorted.length - 1)) {
              pos++;
              double recall2 = recallVals[sorted[pos]];
              if (recall2 != recall) {
                double precis2 = tcurve.instance(sorted[pos]).value(precisInd);
                double slope = (precis2 - precis) / (recall2 - recall);
                double offset = precis - recall * slope;
                precis = isize * i * slope + offset;
          break;
              }
          }
          psum += precis;
        }
        return psum / n;
    }
```

```
        public static double getPRCArea(Instances tcurve) {
      final int n = tcurve.numInstances();
      if (!RELATION_NAME.equals(tcurve.relationName())
            || (n == 0)) {
         return Double.NaN;
      }

      final int pInd = tcurve.attribute(PRECISION_NAME).index();
      final int rInd = tcurve.attribute(RECALL_NAME).index();
      final double[] pVals = tcurve.attributeToDoubleArray(pInd);
      final double[] rVals = tcurve.attributeToDoubleArray(rInd);

      double area = 0;
      double xlast = rVals[n - 1];
for (int i = n - 2; i >= 0; i--) {
         double recallDelta = rVals[i] - xlast;
         area += (pVals[i] * recallDelta);

         xlast = rVals[i];
      }

      if (area == 0) {
         return 1;
      }
      return area;
   }
public static double getROCArea(Instances tcurve) {

      final int n = tcurve.numInstances();
      if (!RELATION_NAME.equals(tcurve.relationName())
            || (n == 0)) {
         return Double.NaN;
      }
      final int tpInd = tcurve.attribute(TRUE_POS_NAME).index();
      final int fpInd = tcurve.attribute(FALSE_POS_NAME).index();
      final double[] tpVals = tcurve.attributeToDoubleArray(tpInd);
      final double[] fpVals = tcurve.attributeToDoubleArray(fpInd);

      double area = 0.0, cumNeg = 0.0;
      final double totalPos = tpVals[0];
      final double totalNeg = fpVals[0];
      for (int i = 0; i < n; i++) {
         double cip, cin;
         if (i < n - 1) {
            cip = tpVals[i] - tpVals[i + 1];
            cin = fpVals[i] - fpVals[i + 1];
         } else {
            cip = tpVals[n - 1];
```

```
            cin = fpVals[n - 1];
          }
          area += cip * (cumNeg + (0.5 * cin));
          cumNeg += cin;
        }
        area /= (totalNeg * totalPos);

        return area;
    }
public static int getThresholdInstance(Instances tcurve, double threshold) {

        if (!RELATION_NAME.equals(tcurve.relationName())
              || (tcurve.numInstances() == 0)
              || (threshold < 0)
              || (threshold > 1.0)) {
          return -1;
        }
        if (tcurve.numInstances() == 1) {
          return 0;
        }
        double[] tvals = tcurve.attributeToDoubleArray(tcurve.numAttributes() - 1);
        int[] sorted = Utils.sort(tvals);
        return binarySearch(sorted, tvals, threshold);
    }
private static int binarySearch(int[] index, double[] vals, double target) {

        int lo = 0, hi = index.length - 1;
        while (hi - lo > 1) {
          int mid = lo + (hi - lo) / 2;
          double midval = vals[index[mid]];
          if (target > midval) {
            lo = mid;
          } else if (target < midval) {
            hi = mid;
          } else {
            while ((mid > 0) && (vals[index[mid - 1]] == target)) {
              mid--;
            }
            return mid;
          }
        }
        return lo;
    }

    /**
     *
     * @param predictions the predictions to use
     * @param classIndex the class index
```

```
     * @return the probabilities
     */
    private double[] getProbabilities(FastVector predictions, int classIndex) {

        // sort by predicted probability of the desired class.
        double[] probs = new double[predictions.size()];
        for (int i = 0; i < probs.length; i++) {
            NominalPrediction pred = (NominalPrediction) predictions.elementAt(i);
            probs[i] = pred.distribution()[classIndex];
        }
        return probs;
    }
private Instances makeHeader() {

        FastVector fv = new FastVector();
        fv.addElement(new Attribute(TRUE_POS_NAME));
        fv.addElement(new Attribute(FALSE_NEG_NAME));
        fv.addElement(new Attribute(FALSE_POS_NAME));
        fv.addElement(new Attribute(TRUE_NEG_NAME));
        fv.addElement(new Attribute(FP_RATE_NAME));
        fv.addElement(new Attribute(TP_RATE_NAME));
        fv.addElement(new Attribute(PRECISION_NAME));
        fv.addElement(new Attribute(RECALL_NAME));
        fv.addElement(new Attribute(FALLOUT_NAME));
        fv.addElement(new Attribute(FMEASURE_NAME));
        fv.addElement(new Attribute(SAMPLE_SIZE_NAME));
        fv.addElement(new Attribute(LIFT_NAME));
        fv.addElement(new Attribute(THRESHOLD_NAME));
        return new Instances(RELATION_NAME, fv, 100);
    }
private Instance makeInstance(TwoClassStats tc, double prob) {

        int count = 0;
        double[] vals = new double[13];
        vals[count++] = tc.getTruePositive();
        vals[count++] = tc.getFalseNegative();
        vals[count++] = tc.getFalsePositive();
        vals[count++] = tc.getTrueNegative();
        vals[count++] = tc.getFalsePositiveRate();
        vals[count++] = tc.getTruePositiveRate();
        vals[count++] = tc.getPrecision();
        vals[count++] = tc.getRecall();
        vals[count++] = tc.getFallout();
        vals[count++] = tc.getFMeasure();
        double ss = (tc.getTruePositive() + tc.getFalsePositive())
                / (tc.getTruePositive() + tc.getFalsePositive() + tc.getTrueNegative() +
tc.getFalseNegative());
        vals[count++] = ss;
```

```
      double expectedByChance = (ss * (tc.getTruePositive() + tc.getFalseNegative()));
      if (expectedByChance < 1) {
         vals[count++] = 0;//Utils.missingValue();
      } else {
         vals[count++] = tc.getTruePositive() / expectedByChance;

      }
      vals[count++] = prob;
      return new Instance(1.0, vals);
   }
public String getRevision() {
      return RevisionUtils.extract("$Revision: 8034 $");
   }
public void generateROC(){
      try {
         Instances inst = new Instances(new BufferedReader(new FileReader("data\\
input30.txt")));
         if (false) {
            System.out.println(ROC1.getNPointPrecision(inst, 11));
         } else {
            inst.setClassIndex(inst.numAttributes() - 1);
            ROC1 tc = new ROC1();
            EvaluationUtils eu = new EvaluationUtils();
            Classifier classifier = new weka.classifiers.functions.Logistic();
            FastVector predictions = new FastVector();
            for (int i = 0; i < 2; i++) { // Do two runs.
               eu.setSeed(i);
               predictions.appendElements(eu.getCVPredictions(classifier, inst, 10));
}
            Instances result = tc.getCurve(predictions);
ThresholdVisualizePanel vmc = new ThresholdVisualizePanel();
            vmc.setROCString("(Area under ROC = "+
Utils.doubleToString(tc.getROCArea(result), 4) + ")");
            vmc.setName(result.relationName());
            PlotData2D tempd = new PlotData2D(result);
            tempd.setPlotName(result.relationName());
            tempd.addInstanceNumberAttribute();
boolean[] cp = new boolean[result.numInstances()];
            for (int n = 1; n < cp.length; n++) {
               cp[n] = true;
            }
            tempd.setConnectPoints(cp);
         vmc.addPlot(tempd);
         String plotName = vmc.getName();
                  final javax.swing.JFrame jf
                        = new javax.swing.JFrame("Weka Classifier Visualize: " + plotName);
                  jf.setSize(500, 400);
                  jf.getContentPane().setLayout(new BorderLayout());
```

```
                jf.getContentPane().add(vmc, BorderLayout.CENTER);
                jf.addWindowListener(new java.awt.event.WindowAdapter() {
                   public void windowClosing(java.awt.event.WindowEvent e) {
                      jf.dispose();
                   }
                });
                jf.setVisible(true);
             }
          } catch (Exception ex) {
             ex.printStackTrace();
          }
       }
       public static void main(String[] args) {

          try {
             Instances inst = new Instances(new BufferedReader(new FileReader("data\\
input30.txt")));
             if (false) {
                System.out.println(ROC1.getNPointPrecision(inst, 11));
             } else {
                inst.setClassIndex(inst.numAttributes() - 1);
                ROC1 tc = new ROC1();
                EvaluationUtils eu = new EvaluationUtils();
                Classifier classifier = new weka.classifiers.functions.Logistic();
                FastVector predictions = new FastVector();
                for (int i = 0; i < 2; i++) { // Do two runs.
                   eu.setSeed(i);
                   predictions.appendElements(eu.getCVPredictions(classifier, inst, 10));
      }
                Instances result = tc.getCurve(predictions);
      ThresholdVisualizePanel vmc = new ThresholdVisualizePanel();
                vmc.setROCString("(Area under ROC = "+
Utils.doubleToString(tc.getROCArea(result), 4) + ")");
                vmc.setName(result.relationName());
                PlotData2D tempd = new PlotData2D(result);
                tempd.setPlotName(result.relationName());
                tempd.addInstanceNumberAttribute();
      boolean[] cp = new boolean[result.numInstances()];
                for (int n = 1; n < cp.length; n++) {
                   cp[n] = true;
                }
                tempd.setConnectPoints(cp);
      vmc.addPlot(tempd);
      String plotName = vmc.getName();
                final javax.swing.JFrame jf
                      = new javax.swing.JFrame("Weka Classifier Visualize: " + plotName);
                jf.setSize(500, 400);
                jf.getContentPane().setLayout(new BorderLayout());
```

```
                jf.getContentPane().add(vmc, BorderLayout.CENTER);
                jf.addWindowListener(new java.awt.event.WindowAdapter() {
                    public void windowClosing(java.awt.event.WindowEvent e) {
                        jf.dispose();
                    }
                });
                jf.setVisible(true);
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

**Declaring and setting data from utility class**

```
public class UtilityClass {

    static ArrayList<Attribute> attInfo = new ArrayList<>();
    private static Instances train_data1;
    private static Instances train_data2;
    private static Instances test_data;
    private static Instances test_data2;
```

**Preparing data**

```
    public static void prepareData() {
        BufferedReader reader = null;
        try {
            reader = new BufferedReader(new FileReader("data\\data70.txt"));
            setTrain_data1(new Instances(reader));
            getTrain_data1().setClassIndex(getTrain_data1().numAttributes() - 1);

            reader = new BufferedReader(new FileReader("data\\data270.txt"));
            setTrain_data2(new Instances(reader));
            getTrain_data2().setClassIndex(getTrain_data2().numAttributes() - 1);
        } catch (Exception ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```

**Building model1 using training data**

```
    public static NaiveBayes getNaiveBayesData1Model() throws Exception {
        NaiveBayes nb = null;
        try {
            prepareData();
            nb = new NaiveBayes();
            nb.buildClassifier(getTrain_data1());

        } catch (IOException ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }
        return nb;
```

```
    }

    public static NaiveBayes getNaiveBayesData70Model() throws Exception {
        NaiveBayes nb = null;
        try {
            prepareData();
            nb = new NaiveBayes();
            nb.buildClassifier(getTrain_data1());

        } catch (IOException ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }
        return nb;
    }
```

**Building model2 using training data 2**

```
    public static NaiveBayes getNaiveBayesData270Model() throws Exception {
        NaiveBayes nb = null;
        try {
            prepareData();
            nb = new NaiveBayes();
            nb.buildClassifier(getTrain_data2());

        } catch (IOException ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }
        return nb;
    }
```

**Preparing data 2**

```
    public static Instances getTest30Data() throws Exception {
        Instances instances = null;
        try {
            BufferedReader reader = new BufferedReader(new FileReader("data\\input30.txt"));
            instances = new Instances(reader);
            instances.setClassIndex(instances.numAttributes() - 1);

        } catch (IOException ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }

        return instances;
    }

    public static Instances getTest230Data() throws Exception {
        Instances instances = null;
        try {
            BufferedReader reader = new BufferedReader(new FileReader("data\\input230.txt"));
            instances = new Instances(reader);
            instances.setClassIndex(instances.numAttributes() - 1);
```

```
        } catch (IOException ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }

        return instances;
    }
```

**Building model 2 using training set 2**

```
    public static NaiveBayes getNaiveBayesData2Model() throws Exception {
        NaiveBayes nb = null;
        try {
            BufferedReader reader = new BufferedReader(new FileReader("data\\data270.txt"));
            train_data2 = new Instances(reader);
            getTrain_data2().setClassIndex(getTrain_data2().numAttributes() - 1);
            nb = new NaiveBayes();
            nb.buildClassifier(getTrain_data2());

        } catch (IOException ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }
        return nb;
    }
```

**Classification phase 1**

```
    public static double getInstanceClassInput1(Instances train_data, NaiveBayes nb) throws
IOException {
        double res = 0;
        try {
            ArffLoader loader2 = new ArffLoader();
            BufferedReader reader = new BufferedReader(new FileReader("data\\input.txt"));
            test_data = new Instances(reader);
            test_data.setClassIndex(test_data.numAttributes() - 1);

            Evaluation eval = new Evaluation(train_data);
            res = eval.evaluateModelOnce(nb, test_data.firstInstance());

        } catch (Exception ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null,
ex.getStackTrace());
        }
        return res;
    }
```

**Classification phase 2**

```
    public static double getInstanceClassInput2(Instances train_data, NaiveBayes nb) throws
IOException {
        double res = 0;
        try {
            BufferedReader reader = new BufferedReader(new FileReader("data\\input2.txt"));
            test_data2 = new Instances(reader);
```

```
            test_data2.setClassIndex(test_data2.numAttributes() - 1);

            Evaluation eval = new Evaluation(train_data);
            res = eval.evaluateModelOnce(nb, test_data2.firstInstance());
            System.out.println("Result: " + res);
        } catch (Exception ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }
        return res;
    }
```

**Accepting data from interface for classification phase1**

```
    public static void writeToInput1File(int[] data) {
        try {
            BufferedWriter bw = new BufferedWriter(new FileWriter(new File("data\\input.txt")));
            String text = "@relation\tMalaria\n"
                    + "@attribute fever\t{1, 0}\n"
                    + "@attribute headache\t{1, 0}\n"
                    + "@attribute nausea\t{1, 0}\n"
                    + "@attribute vomiting\t{1, 0}\n"
                    + "@attribute class1\t{N, P}\n"
                    + "@data\n"
                    + data[0] + "," + data[1] + "," + data[2] + "," + data[3] + ",?";
            bw.write(text);
            bw.close();
        } catch (IOException ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```

**Accepting data from interface for classification phase 2**

```
    public static void writeToInput2File(int[] data) {
        try {
            BufferedWriter bw = new BufferedWriter(new FileWriter(new File("data\\input2.txt")));
            String text = "@relation\tMalaria\n"
                    + "@attribute respiratory_distress\t{1, 0}\n"
                    + "@attribute convulsion\t{1, 0}\n"
                    + "@attribute coma\t{1, 0}\n"
                    + "@attribute class2\t{C, U}\n"
                    + "@data\n"
                    + data[0] + "," + data[1] + "," + data[2] + ",?";
            bw.write(text);
            bw.close();
        } catch (IOException ex) {
            Logger.getLogger(UtilityClass.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
public static Instances getTrain_data() {
        return getTrain_data1();
```

```
    }

    public static void setTrain_data(Instances aTrain_data) {
        setTrain_data1(aTrain_data);
    }

    public static Instances getTest_data() {
        return test_data;
    }

    public static void setTest_data(Instances aTest_data) {
        test_data = aTest_data;
    }

    public static Instances getTrain_data2() {
        return train_data2;
    }

    public static void setTrain_data2(Instances aTrain_data2) {
        train_data2 = aTrain_data2;
    }

    public static Instances getTrain_data1() {
        return train_data1;
    }

    public static void setTrain_data1(Instances aTrain_data1) {
        train_data1 = aTrain_data1;
    }
}
```