B.Sc. in Computer Science and Engineering Thesis

An Intelligent Diabetes Prediction System Using Machine Learning Technique

Submitted by

Nabila Shahnaz Khan
201414105

Mehedi Hasan Muaz
201414102

Anusha Kabir
201414081

Supervised by

Major Muhammad Nazrul Islam, PhD

Instructor Class B

Department Of Computer Science & Engineering

Military Institute of Science & Technology

Department of Computer Science and Engineering
Military Institute of Science and Technology

December 2017

CERTIFICATION

This thesis paper titled **"An Intelligent Diabetes Prediction System Using Machine Learning Technique"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in December 2017.

Group Members:

1. Nabila Shahnaz Khan
2. Mehedi Hasan Muaz
3. Anusha Kabir

Supervisor:

_____

Major Muhammad Nazrul Islam, PhD
Instructor Class B
Military Institute of Science & Technology

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis paper, titled, **"An Intelligent Diabetes Prediction System Using Machine Learning Technique"**, is the outcome of the investigation and research carried out by the following students under the supervision of Major Dr. Muhammad Nazrul Islam, PhD, CSE Department, Military Institute of Science & Technology.

It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Nabila Shahnaz Khan
201414105

Mehedi Hasan Muaz
201414102

Anusha Kabir
201414081

ACKNOWLEDGEMENT

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor, **Major Dr. Muhammad Nazrul Islam**, PhD, CSE Department, Military Institute of Science & Technology, for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advices throughout the study was of great help in completing thesis.

We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work.

We also thank Adnan Sharif (CSE-14) for his assistance in developing the mobile application based on our research. Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

Dhaka
December 2017

1. Nabila Shahnaz Khan
2. Mehedi Hasan Muaz
3. Anusha Kabir

ABSTRACT

With the advancement of information technologies, mobile health (mHealth) technologies can be leveraged for patient self-management, patient diagnosis and determining or predicting the probability of being affected by some disease. Diabetes mellitus is a chronic and lifestyle disease and millions of people from all over the world fall victim to this disease. There are only few mobile apps keeping track of calories, sugar taken, medicine doses, lifestyle, blood glucose, blood pressure and giving suggestion about food, exercises to prevent or control diabetes. Also the number of studies done to predict diabetes or reveal the factors influencing diabetes are handful. Still application that is explicitly developed to analyze the risk of being a diabetic patient using any Artificial Intelligence (AI) or Machine Learning (ML) technique was not found. Therefore, the objective of this work is to understand the underlying factors influencing an individual's risk of being affected by diabetes and to develop an intelligent mHealth system to assess his/her possibility of being diabetic, prediabetic or nondiabetic. For this work, *firstly*, data was collected using a survey method and analyzed using the Pearson correlation and Chi-squared test; *secondly*, we proposed three algorithms for prediction of diabetes: one is based on prior clustering, second one is developed using Naive Bayes and the third one used Naive Bayes with Prior Clustering; *finally*, after evaluating the three approaches for the best performance, an application has been developed taking the best approach into consideration for predicting the possibility of being diabetic using this algorithm.

Contents

CERTIFICATION ii

DECLARATION iii

ACKNOWLEDGEMENT iv

ABSTRACT 1

List of Abbreviation 7

1 Introduction 8

1.1 Preface 8

1.2 Problem Statement 9

1.3 Thesis Objective 10

1.4 Methodological Overview 10

1.5 Thesis Scope 11

1.6 Thesis Organization 11

2 Theoretical Background and Related works 13

2.1 Related Predicitve Models, Algorithms and Techniques 13

2.2 Related Applications 14

2.3 Chapter Summary 16

3 Research Methodology 17

3.1 Working Procedure 17

3.2 Phases of Research Methodology 18
3.2.1 Data Collection and Analysis 18
3.2.2 Developing and Evaluating Intelligent Predictive Algorithms 18
3.2.3 Developing Mobile Application 19

4 Data Collection and Analysis **20**
4.1 Data Collection 20
4.2 Analyzing Data 20
4.2.1 Profile of the Respondents 21
4.2.2 Factor Analysis 21
4.3 Using Larger Dataset to Check Efficiency of the Algorithms 25

5 Developing Intelligent Predictive Algorithm **26**
5.1 Algorithm Based on Proximity 26
5.2 Algorithm Based on Naive Bayes 29
5.2.1 Training Phase 31
5.2.2 Testing Phase 32
5.3 Algorithm Based on Naive Bayes with Prior Clustering 33
5.3.1 Clustering Using BSAS 35
5.3.2 Naive Bayes on Specific Cluster 36

6 Evaluating Intelligent Predictive Algorithm **37**
6.1 Result Analysis Using Survey Dataset 38
6.1.1 Result Analysis of Algorithm 1(based on proximity) 38
6.1.2 Result Analysis of Algorithm 2 (Naive Bayes) 38
6.1.3 Result Analysis of Algorithm 3 (Naive Bayes with prior clustering) 39
6.2 Result Analysis Using Pima Indian Diabetes Dataset 40
6.2.1 Result Analysis of Algorithm 1 (based on proximity) 40
6.2.2 Result Analysis of Algorithm 2 (Naive Bayes) 41

6.2.3 Result Analysis of Algorithm 3 (Naive Bayes with prior clustering) 41
6.3 Comparison of Accuracy of the Algorithms 42

7 Developing Mobile Application **45**
7.1 Development Architecture 45
7.2 Development Platform 46
7.3 Implementing Application 46
7.4 Features 47

8 Discussion and Conclusion **49**
8.1 Main Outcomes 49
8.2 Practical Implications 49
8.3 Study Limitations 50
8.3.1 Lack of Data 50
8.3.2 Lack of Features 50
8.3.3 Usability Issues 50
8.4 Future Expansion 50
8.4.1 Evaluating with Larger Dataset 50
8.4.2 Adding More Features 51
8.4.3 Enhancing Accuracy 51
8.4.4 Exploring Other Techniques of Prediction 51
8.4.5 Evaluate Usability and UX of the App 51
8.5 Concluding Remarks 52

APPENDIX A **58**

APPENDIX B **62**

APPENDIX C **71**

List of Figures

1.1 Methodological Overview 10

3.1 Detailed Overview of the Working Procedure Followed 17

4.1 Survey Questionnaire 21

4.2 Correlation between continuous variables age and BMI 23

5.1 Classification of Machine Learning 30

5.2 Classification of Clustering 35

5.3 Dividing Data into Clusters 36

6.1 Accuracy of Implemented Naive Bayes Algorithm 39

6.2 Accuracy of Implemented Improved Naive Bayes Algorithm 39

6.3 Comparison of Accuracy of Three Algorithms Applied on Both Datasets . . 43

7.1 Development Architecture of the Mobile Application 45

7.2 Sample User Interface of the Mobile Application 47

List of Tables

2.1 Diabetes Related Mobile Applications 15

4.1 Value of Observed Data for Gender and Ancestral Diabetic History 24

4.2 Value of Estimated Data for Gender and Ancestral Diabetic History 24

4.3 Results of Factor Analysis 24

5.1 Mean and Variance Calculation For Continuous Features 33

6.1 Error Estimation 38

6.2 Confusion Matrix 40

6.3 Confusion Matrix 41

6.4 Confusion Matrix 42

6.5 Comparison of Performance Among three Algorithms Using Survey Dataset 43

6.6 Comparison of Performance Among three Algorithms Using Pima Indian Diabetes Dataset 43

LIST OF ABBREVIATION

mHealth : Mobile Health
eHealth : Electronic Health
AI : Artificial Intelligence
ML : Machine Learning
NN : Neural Network
PDA : Personal Digital Assistant
WHO : World Health Organization
BMI : Body Mass Index
HbA1c : Hemoglobin A1c
FTA : Few Touch Application)
MAP : Maximum A Posteriori
BSAS : Basic Sequential Algorithmic Scheme
MBSAS : Modified Basic Sequential Algorithmic Scheme
ROC : Receiver Operating Characteristic
IDE : Integrated Development Environment
UX : User Experience
HTTP : HyperText Transfer Protocol
HTTPS : HyperText Transfer Protocol Secure

CHAPTER 1
INTRODUCTION

The two prime concerns of our work namely mHealth and Diabetes are briefly presented in this chapter. Alongside the chapter also contains methodological overview, thesis scope and the organization of the rest of the book.

1.1 Preface

Mobile Health (mHealth) can be defined as the access, provision and delivery of healthcare interactions anywhere, anytime which is facilitated by mobile and wireless technologies. The mHealth includes wireless technologies to provide various data contents and health related services through devices such as mobile phones, smart phones, PDAs, laptops and tablet PCs [1]. The mHealth field has emerged as a sub-segment of eHealth. The mHealth field is remarkably dynamic and the range of applications being designed is constantly expanding, which includes, for example providing clinical assistance, health monitoring, diet plans or healthy life suggestions, general facility information, patients portal and the like.

The increase in the usage of mobile phone technology and internet access facilities is giving a huge bump to increase of mobile health technology. World Health Organization (WHO) in 2011 has found that over 83% of member states reported implementation of at least four or more mHealth solutions [2]. By 2015, number of apps concerning health issues increased to 165,000 [3]. The developing countries like Bangladesh is also holding on to this advancement. According to WHO [4], Bangladesh has been listed as one of the 15 countries using mHealth in order to raise awareness regarding health issues. A recent study [5] shows that about 180 mHealth applications have been Preliminarily developed specifically for the users from Bangladesh.

Though the growing rate seems to be petty high, still mHealth technology has not been embraced by many healthcare providers and patients inspite of the fact that this technology has been proven to be reliable and can be really helpful in solving the healthcare challenges. The mhealth tools can help people self-manage their health, keep track of their daily routine, to-do list, medication time table, alert doctors in case of sudden or alarming changes in patient's condition. So, it is really important to accept and understand mHealth technology

and take benefit of it to its fullest.

Diabetes- An Epidemic Of Future Diabetes is an alarming health issue which is currently reaching an epidemic proportion. According to WHO [6], by 2012 an estimated 422 million people were living with diabetes and by 2030, this number is estimated to increase to about 552 million [7,8]. Even in Bangladesh about 8.4 million of the adult population are carrying this deadly disease [9] and nearly 51.2% of the population are unaware of it [10]. So, globally diabetes has become a primary health concern.

Among many other initiatives, mHealth service for diabetic patients is very popular. A systematic review conducted by Ahn [11] found a total 656 diabetes related apps including both iOS and Android app. Basar et al [12] filtered 43 diabetes related apps.

There are mainly three forms of diabetes, type 1, type 2 and gestational diabetes. According to Raha [13], 5% of total people suffering from diabetes have type 1 diabetes (juvenile diabetes), children and young adults. The possibility of being affected by this kind of diabetes is not predictable [13]. Gestational diabetes is diagnosed during pregnancy. The most common form is the type 2 diabetes; the symptoms of which can be predicted and this can be prevented through proper care and knowledge. A prediction system depicted in study [14] takes into count 23 features and applies regression technique to predict if a person can be diabetic and at what age. Investigation of the performances of different machine learning approaches for predicting diabetes from medical records is presented in [15]. Unhealthy life style, lack of exercise, absence of balanced food habit, and inheritance from ancestors are the main reasons influencing diabetes. This disease being silent is not observed mostly at initial stages due to which the population being affected by diabetes and the death rate caused by the disease is on an increase. Regular checkup being an expensive way out cannot be afforded by mass population of a developing country like ours. Thus mHealth the sector that underwent an unexpected evolution over the past years is expected to play a positive role in the prediction of diseases like diabetes efficiently.

1.2 Problem Statement

To conduct a research on a topic it is very important to find a problem in that domain and then the problem is approached using the information and knowledge gathered during the research period. At present, diabetes is the common disease globally and it's known as a hidden disease as it remains hidden in the body of the affected person and starts showing its symptoms very slowly. In most of the cases when the affected person becomes aware of his condition, he already has a high sugar level which is very difficult to control. If he had been alarmed at an early stage he might have been able to control his sugar level easily and thus prevent diabetes.

mHealth, being the most prominent and easy to access field of health care does not provide any service to reduce this frightening spread of diabetes and deaths caused by it by predicting the possibility of growing this disease, taking individual profiles into consideration. But the advent of an application that can detect the state of diabetes a person is going through or predict possibility of being diabetic explicitly will be very helpful for the underprivileged domain of developing and under developed countries and also for creating awareness about diabetes globally.

1.3 Thesis Objective

The objective of this thesis was firstly to analyze the risk factors of diabetes and decide on the risk factors which common people are well acquainted with. Secondly to propose an algorithm having better accuracy than the existing algorithm for developing an intelligent predictive system. Finally to build a mHealth application having the algorithm incorporated with it to help people predict their state of diabetes.

1.4 Methodological Overview

This section provides a quick over view of the entire thesis work.

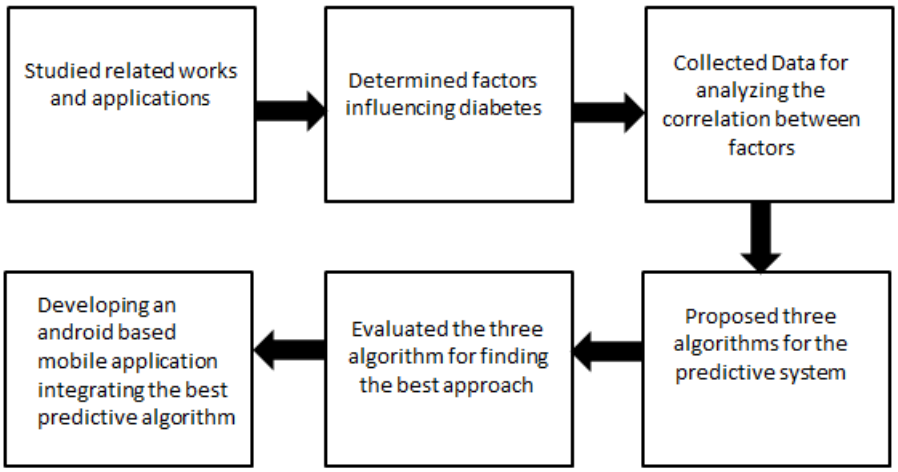

Figure 1.1: Methodological Overview

Initially a good number of research works were studied that either focused on finding the risk factors of diabetes or implemented machine learning for determining the possibility of becoming diabetic to understand the background theory. We then used survey method to collect empirical data for the factors related to diabetes aiming to analyze the factors.

The survey questionnaire was distributed both through social media and in person. The collected data has been analyzed for factor analysis and it has been found that the factors were independent of each other. For developing the intelligent system that predicts diabetes we we proposed three algorithms. Firstly a predicting algorithm based on proximity was developed. Then basing on the independency of factors Naive Bayes algorithm, a machine learning technique was implemented on survey data set to decide on the state of diabetes, the person with respective physical conditions belong to. Finally, with a view to improving the accuracy we proposed a new algorithm that combines prior clustering of data set and Naive Bayes technique. We evaluated all the three systems using our existing data set collected through survey. An android based application was developed integrating the best predictive algorithm.

1.5 Thesis Scope

The thesis scope can be defined as machine learning (supervised approach) based mHealth application for predicting the state of diabetes or possibility of growing diabetes of any individual. Diabetes being a common disease with high unpredictability, the need for an intelligent system predicting the possibility of being diabetic without doing any kind of medical check-up is ever increasing. So this thesis aims at building an intelligent mobile application that can provide the preliminary view of the level of diabetes. The application does not provide any added recommendation or suggestion except for the output obtained from the system with given parameters. Here for classifier first a predictive algorithm, then Naive Byes algorithm and later for better accuracy an improved approach combining of Naive Bayes and clustering was used. No other algorithm was taken into consideration during the research.

1.6 Thesis Organization

In Chapter 2, related works done in this field have been discussed. different diabetes related mHealth apps which are mostly popular have been highlighted. At the same time, different research works and case studies using predictive algorithms and machine learning techniques have been conferred in this part.

In chapter 3, the overall research methodology followed throughout this time have been explained. A clear idea has been given about the development phases which are collecting and analyzing data, proposing, implementing and improving the algorithm, evaluation of the overall process and finally developing the mobile application.

In chapter 4, we have described about the data collection and factor analysis procedure to

get insight of the factors influencing diabetes.

In chapter 5, the total process of developing the intelligent predictive system by implementing three different approaches that predetermine the possibility of being diabetic is described.

In chapter 6, we have discussed about the overall system evaluation which includes the analysis of the results for different types of algorithms, main limitations and challenges faced while fulfilling the goal and also the further plans to make the system a real successful one.

In chapter 7, the development process of the application has been discussed which illustrates the main architecture and developemnt platform of the application and the features it offers along with use case scenarios.

Finally the conclusion of our thesis work has been drawn in chapter 8.

CHAPTER 2
THEORETICAL BACKGROUND AND RELATED WORKS

This section contains the works relevant to the thesis. Firstly algorithms, models or machine learning approaches for the purpose of predicting diseases are discussed. Then digital solutions or applications dealing with diabetes or other diseases are mentioned with their features. Finally a summary of the section is given that highlights the existing gap of the researches.

2.1 Related Predicitve Models, Algorithms and Techniques

Some studies have been carried out for analyzing the risk factors of diabetes and calculating probabilities of developing different diseases.

Chawla and Davis [16] have proposed a patient-centered framework using Big Data driven approach along with collaborative filtering. They depicted correlations between lifestyle, molecular and environmental factors of individuals which is used to create a personalized disease risk profile. Their study concentrates on finding disease risk factors of heart diseases, hepatitis, cancer and few more. Sarwar and Sharma [17] analyzed the efficiency of algorithm for diagnosing diabetes taking 10 factors that influence the disease into consideration. The prediction model was developed for Naive Bayes, Artificial Neural Network(ANN), and K-Nearest Neighbours(KNN). A software model for predicting blood glucose level that occur due to regimen changes is discussed in. The intelligent system based on neural network, expert system and fuzzy logic consisting of treatment algorithm was developed and evaluated. Cafazzo et al [18],developed an app that showed the rise in the blood glucose monitoring rate in adolescent having type 1 diabetes. SMARTDIAB- a platform for the purpose of management, monitoring and treatment of type 1 diabetes mellitus is describes in [19] . It combines database technologies, communications, simulation algorithms, and data mining. The system has 2 units a. Patient Unit and b. Patient management unit.

Another study [20] lays its spotlight on diabetes, considering factors like HbA1c, age, sex, hypertension and uses data mining technique. An attempt was made to diagnose diabetes based on patient's habitual data and A1c sugar level in [21].Here the assessment of agree-

ment between Chronic Condition Warehouse diabetes algorithm incase of Medicare claims and self-report measures regarding diabetes was done. Marx in his study [22] found Gene factors, age, excess weight gain, unhealthy diet and lifestyle, environmental pollution as common factors that contribute in the development of type 2 diabetes.

A novel fuzzy expert system approach for diabetes decision support is depicted in [23]. For describing knowledge with uncertainity five layer fuzzy ontology was developed in fuzzy expert system. For the diagnosis of diabetes an intelligent representation of support vector machines (SVMs) has been proposed in [24], which produces comprehensive rule set that matches the results of other related medical studies. Study [25] performs diagnosis of diabtes using Generalized Discriminant Analysis (GDA) and Least Square Support Vector Machine (LS-SVM) along with proposing a new cascade learning system.

2.2 Related Applications

Till now, several studies have been conducted to highlight the importance of mHealth technology in health care and self-management [26]. A systematic review that has been conducted by Blaya et al [27] found that mobile phones improved efficiency of health services immensely and expanded the scope of delivering treatment to thousands of patients in an effective manner. FTA (Few Touch Application) tool, which is a combination of 10-app based systems is described in [28] has been designed to make life easier for diabetic patients. It is a combination of 10-app based systems that have been designed to analyze the effectiveness of mHealth apps. The prime focus of a study conducted in [29] is diabetes management and monitoring. They have presented a METABO that records dietary, physical activity, medication and medical information. In [30] an application is developed for personalized health care of diabetic patients that provides decision on drug doses for maintaining stable glucose level.

Many other studies have been carried out on the usefulness of mHealth apps available in the market. In [31], a centralized resource is developed which contains information about more than 60,000 mHealth apps from both Google Play and Apple app store. In another study, Comstock [32] have predicted that by 2018, there will be 24 million diabetes app users.Table 2.1 provides a summary of few most downloaded apps with greater positive review (see the apps numbered 1-6 in Table 2.1) and the few other apps (see the apps numbered 7-9 in Table 2.1) that are developed in Bangladesh,which are available in Google Play Store [33].

Table 2.1: Diabetes Related Mobile Applications

Name	Features
1. DiabetesM	Used for type-1, type-2, gestational diabetes, etc. Keeps track of every aspect of diabetes treatment. Generates reports, graphs and statistics based on stored records. Reports can be further sent as emails to physicians. Can find trend of blood glucose level.
2. Glucose buddy	Record user's blood glucose level, food intake, blood pressure, medication etc. Provides reminder for blood sugar testing after certain intervals
3. Diabetes Monitor	Easy and intuitive recording, analyzing and better control of diabetes. Measures blood sugar level and other related factors. Deals with the activities that have impact on blood sugar.
4. Diabetes Predict Prevent	Assesses the likelihood of developing diabetes by a 3 minute questionnaire. On a positive result of the test, provides prevention methods.
5. Diabetes and Me	Two domains: Informative and Interactive Informative domain includes information, guidance to control diabetes. Interactive domain records data of cholesterol level, blood sugar reading, BMI etc.
6. Diabetes Plus	Store readings related to diabetes of individual and send those directly to doctors. Defines a blood glucose target range, generates graphs and calculates statistics.
7. Mx of Diabetes Mellitus	Calculates Daily calorie requirement for patients Calculates Insulin dose
8. Diabetes Control	Informative app providing answers to frequently asked questions about diabetes.
9. Diabetes Doctor	Informative app for diabetic patients giving necessary information about all aspects of diabetes.

2.3 Chapter Summary

Based on this study done on related works and to the best of our knowledge, we have found that: a) Though diabetes related mHealth applications have been developed, but from the study none were found to produce clear-cut prediction of diabetes. b) Few studies have been conducted to propose prediction systems for diabetic patients c) no study has been found that explicitly revealed the factors of being affected by diabetes. Therefore, this work focuses on developing an intelligent prediction application based on machine learning technique which will tell the users about their (users) probability of becoming a diabetic patient.

CHAPTER 3
RESEARCH METHODOLOGY

Research is a logical and systematic search for new and useful information on a particular topic [34] and a research method is a systematic plan for doing research. It can be defined as a scientific and systematic search for relevant information on a specific topic and trying to solve. This Chapter focuses on the overall working procedure that has been followed throughout the research. Here, firstly the working procedure followed throughout the research has been illustrated along with a block diagram (Figure 3.1). Later on, the different stages of the research methodology have been pointed out in brief.

3.1 Working Procedure

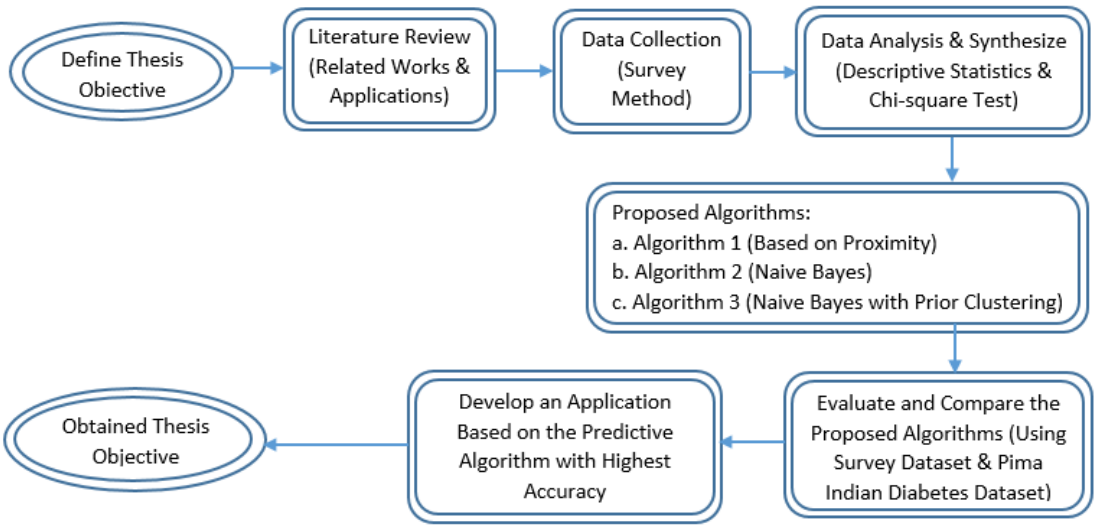

Figure 3.1: Detailed Overview of the Working Procedure Followed

To reach the goal of this research at the very beginning, a working procedure had been followed that has been presented using a block diagram in Figure 3.1. As presented in Figure 3.1, primarily the thesis objective was decided and defined. Then related works and applications were gone through to decide the steps to be followed. Later on data was collected using survey method based on the common features extracted. Next, the correlation between the features were tested to check their interdependency. Based on the data collected three

different algorithms were used to implement the system and their accuracy were checked. The one with the highest accuracy was considered to be the most legitimate one and was used to develop a mobile application to make the system accessible to general mass for their regular use.

3.2 Phases of Research Methodology

The overall working procedure stated above can be divided in the three following phases:

- Data Collection and Analysis
- Developing and Evaluating the Intelligent Predictive Algorithm
- Developing Mobile Application

A brief idea about these phases has been given below:

3.2.1 Data Collection and Analysis

For developing a predictive system we needed true data to train the machine. So data was collected using survey methodology based on the selected factors which are commonly known to mass people. Later on, these factors were analyzed to check their interdependency using Pearson Correlation and Chi-square test. The data collection procedure and the analysis result will be discussed in detail in Chapter 4.

3.2.2 Developing and Evaluating Intelligent Predictive Algorithms

To give a predictive output predictive algorithms have been developed and their accuracy have been tested in the following manner:

Developing Algorithm

Initially a predictive algorithm was proposed but it had some limitations such as the machine was not completely learning from the dataset, calculative range was predefined. So later on, machine learning was adapted and supervised learning was the prime choice. After studying various types of supervised learning algorithms like Nearest Neighbor, Naive Bayes, Decision Trees, Linear Regression, Support Vector Machines (SVM), Neural Networks [35] Naive Bayes was found to be highly feasible with a higher accuracy and is commonly used

for predictive machines [36–38]. So, Naive Bayes algorithm was proposed for implementing the intelligent diabetes predictive system. Later on, an improvement was performed on the basic Naive Bayes algorithm to gain even a higher accuracy. The algorithms have been explained elaborately in Chapter 5.

Checking Accuracy

The accuracy of the initial predictive algorithm was calculated. And for both the Naive Bayes and Improved Naive Bayes, error was tested using Fractional Counting along with Leave-one-out technique and their accuracy also calculated. To check if the algorithms work better using a larger dataset, all three were implemented and evaluated on the Pima Indian Diabetes dataset. The whole accuracy analysis method has been demonstrated in Chapter 6

3.2.3 Developing Mobile Application

Finally, a mobile application has been developed which is right now up and running. By giving one's BMI, Age, Gender and ancestral diabetic history, that person will get a result saying if he has the possibility of being diabetic, prediabetic or nondiabetic. This phase has been discussed in detail in Chapter 7

CHAPTER 4
DATA COLLECTION AND ANALYSIS

This Chapter will present the data collection approach and how we have analyzed and synthesized the data:

4.1 Data Collection

As the application should be usable by general mass so the decision was to use common features which are known to everyone. Such four common features were selected which highly affect one's possibility of being diabetic, these features are:

1. Age
2. BMI
3. Gender
4. Ancestral Diabetic History

For this research, survey method has been used to collect the empirical data. A survey is a systematic method for collecting information from entities for the purpose of gathering quantitative descriptors of attributes [39]. To attain the objectives of this work, the survey questionnaires were related to people's age, weight, gender, HbA1c sugar level, and ancestral diabetic history as shown in Figure 4.1. Survey questions were distributed through Facebook and E-mail to departmental staffs, students of the educational institution, acquaintances and other associated persons. As most of the people weren't aware of their sugar level so we also did volunteer check up to collect data. Around 191 usable responses were received during May to July, 2017 which were used later to train the machine. Respondents were assured of anonymity and confidentiality.

4.2 Analyzing Data

After receiving the data the profile of the respondents were analyzed and the relation between the factors were inspected as depicted below:

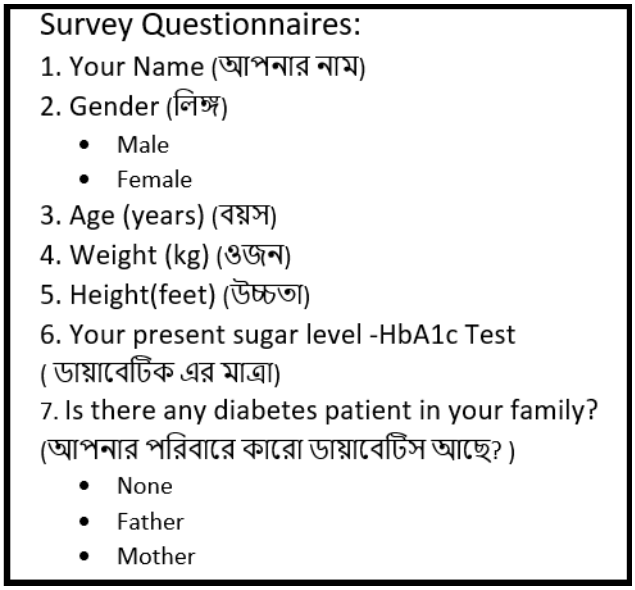

Figure 4.1: Survey Questionnaire

4.2.1 Profile of the Respondents

Among the total 191 respondents 36.8% were female and 63.2% were male with average age of 36 ± 15 (m $\pm$ sd) and an average weight of 64 ± 16 (m $\pm$ sd). Among them, 68.5% of the respondents were unaware of their sugar level (HbA1c value) but the respondents who were diabetic patients were well aware of their sugar level. This clearly indicates that if the nondiabetic people enter the prediabetes stage or is currently standing at the border level or become a diabetic patient, their chances to know the fact is extremely low. This increases their risk factor for this disease. Such lack of knowledge, ignorance and avoidance towards diabetes can play a vital role in leading it to the extreme stage that results in failure of organs and may cause death.

4.2.2 Factor Analysis

The features that have been taken into consideration to train the machine are age, BMI, gender, ancestral diabetic history as they are commonly known to people and have impact on diabetes. To apply some machine learning algorithms like Naive Bayes, these features must be independent of each other and this check was done in the following manner:

Checking Independence of Continuous Features

Continuous variables are numeric variables that have an infinite number of values between any two values. A continuous variable can be numeric or represent date or time [40]. Features BMI and age are continuous features. To test their dependency Pearson Correlation test

was performed. In statistics, the Pearson correlation is the covariance of the two variables divided by the product of their standard deviations. Pearson correlation coefficient (r) is a measure of the linear correlation between two variables X and Y. It has a value between $+1$ and -1, where 1 is total positive linear correlation, 0 is no linear correlation and -1 is total negative linear correlation. So if we have one dataset $\{x_1, x_2, ..., x_n\}$ containing n values and another dataset $\{y_1, y_2, ..., y_n\}$ containing n values then that formula for r is:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where:

- n is the sample size
- x_i, y_i are the single samples indexed with i
- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (the sample mean); and analogously for $\bar{y}$

Positive correlation indicates that both variables increase or decrease together whereas negative correlation indicates that as one variable increases, so the other decreases. The t-test is used to establish if the correlation coefficient is significantly different from zero and hence, that there is evidence of an association between the two variables. If the coefficient value lies between 0.50 and 1, then it is said to be a strong correlation. If the value lies between 0.30 and 0.49, then it is said to be a medium correlation. When the value lies below + .29, then it is said to be a small correlation and it indicates no correlation when the value is 0.

To calculate the Pearson correlation coefficient of the two features BMI and age, we used the online tool 'Pearson Correlation Coefficient Calculator' [41] and the value of co-efficient was found to be 0.13 which is almost equal to 0. So, it could be easily said that the variables BMI and age are independent.

Even from Figure 4.2, it is quite clear that there is no correlation between these two features as the points are scattered and do not form the shape of a line with positive or negative tangent.

Checking Independence of Nominal Features

The Chi-square test is used to determine whether an association (or relationship) between two categorical variables in a sample is likely to reflect a real association between these two variables in the population. In the case of two variables being compared, the test can also be interpreted as determining if there is a difference between the two variables. The sample data is used to calculate a single number (or test statistic), the size of which reflects the

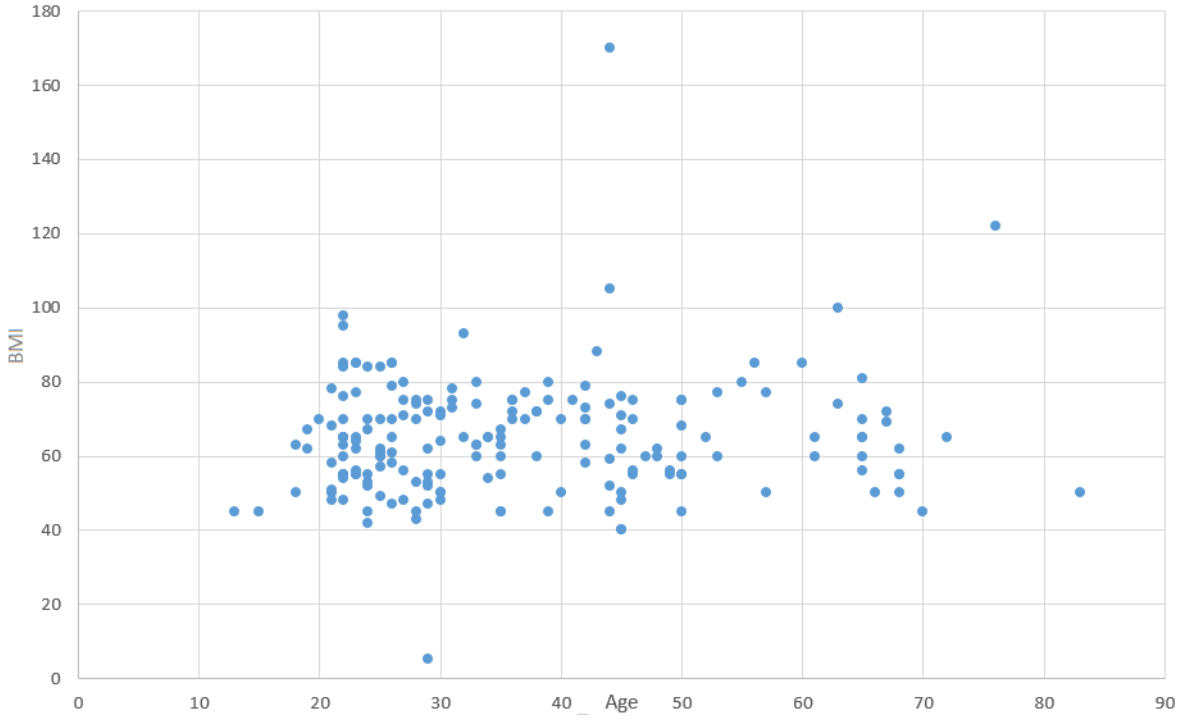

Figure 4.2: Correlation between continuous variables age and BMI

probability (p-value) that the observed association between the two variables has occurred by chance, ie due to sampling error.

There are two types of chi-square tests. Both use the chi-square statistic and distribution for different purposes:

- A chi-square goodness of fit test determines if a sample data matches a population. Example: Goodness of Fit Test.
- A chi-square test for independence compares two variables in a contingency table to see if they are related. In a more general sense, it tests to see whether distributions of categorical variables differ from each another.
  - A very small chi-square test statistic means that your observed data fits your expected data extremely well. In other words, there is a relationship.
  - A very large chi-square test statistic means that the data does not fit very well. In other words, there isnt a relationship.

The formula for the chi-square statistic used in the chi square test is given in Equation 4.1 where O = Observed data, E = Estimated data and i represents ith value.

$$\chi^2 = \Sigma \frac{(O_i - E_i)^2}{E_i} \tag{4.1}$$

For our dataset, the calculation is shown below where Table 4.1 shows the values of observed data and Table 4.2 shows the values of estimated data for features ancestral diabetic history with respect to gender.

Table 4.1: Value of Observed Data for Gender and Ancestral Diabetic History

	Male	Female
Yes	62	52
No	59	19

Table 4.2: Value of Estimated Data for Gender and Ancestral Diabetic History

	Male	Female
Yes	71.84	42.16
No	49.16	28.84

After calculation, the value of Chi-square statistics was found to be 8.98. For a degree of freedom value of 1 and confidence interval of 95% the Critical value was found to be 3.84. As Chi-square statistics $>>$ Critical value, it can be clearly stated that the features gender and ancestral diabetic history are independent of each other.

Summary of Factor Analysis

The summary of the factor analysis has been drawn in Table 4.3.

Table 4.3: Results of Factor Analysis

Factors	Test	Result	Interpretation
Age, BMI	Pearson Correlation	r=0.13	Pearson Correlation Coefficient, r=0.13 $\approx$ 0; independent
Gender, Ancestral diabetic history	χ^2 test	$\kappa = 8.98$ $\alpha = 0.05$	As test statistic, κ >critical value for α; hypothesis rejected; independent

The analysis result shows that there is no strong correlation among the factors in any of the cases. So, the variables are independent. Thus later we could easily move forward to proposing the Naive Bayes Classifier algorithm for the implementation of diabetes prediction system.

4.3 Using Larger Dataset to Check Efficiency of the Algorithms

The amount of data collected through survey method is meager. So to check the accuracy of the developed algorithms more precisely, the algorithms were also implemented on the well known 'Pima Indians Diabetes Data Set' [42] which contains a total of 768 data. It contains data of females who are at least 21 years old and are of Pima Indian heritage. It contains the following attributes:

- Number of times pregnant
- Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Diastolic blood pressure (mm Hg)
- Triceps skin fold thickness (mm)
- Check accuracy of algorithm
- Improve algorithm
- implementing application if enough accuracy is achieved

This dataset was used for a more rigid evaluation of the algorithms but this dataset contained only data of female participants which will make the system bias. Also, the features used here are not commonly known to general folk. So the 'Pima Indians Diabetes Data Set' is not suitable for developing the application and the research has been done based on the dataset collected in the survey method.

CHAPTER 5
DEVELOPING INTELLIGENT PREDICTIVE ALGORITHM

Algorithms that have been considered to develop the predictive system have been introduced in this Chapter. Initially, a predictive algorithm was improvised which used to predict a sugar level for the given user data based on the data stored in the knowledge base. After that, the well known Naive Bayes algorithm was implemented to predict if the user resides in diabetic, prediabeitc or nondiabetic class. Lastly, the Naive Bayes algorithm was modified to get a better result by performing clustering along with the Naive Bayes. The three algorithms that have been consecutively used to implement the system are demonstrated in the following Sections:

5.1 Algorithm Based on Proximity

Initially we developed a predictive algorithm based on supervised learning which compares the user input with the data available in the knowledge base. Then the probable sugar level of user is calculated using data having similar type of attributes. The proposed model used for this purpose contains four phases which are stated below:

1. Building the Knowledge Base
2. Assessment of New User (Test Case)
3. Processing the Knowledge Base
4. Logic and Prediction

Phase 1: Knowledge Base

In Phase 1 of this algorithm, all the relevant data of the users are properly formatted and put into a separate file. Parsing program takes input from this dataset and stores only relevant data in specific data structure. The percentage calculation for each genetic relation is also done in this phase. The pseudocode used for implementing this part is given below [Algorithm 1]:

Algorithm 1 Knowledge Base ($Phase1$)

$ages, weights, sugarlevels, genes, genders \leftarrow newList$
$relations \leftarrow list\ of\ relations$
$weightage \leftarrow empty\ map$
for each patient in times **do**
    ages.add($patient.age$)
    weights.add($patient.weight$)
    sugarlevels.add($patient.sugarlevel$)
    genes.add($patient.gene$)
    genders.add($patient.gender$)
end for
for each unique relation in relations **do**
    $weightage[relation] \leftarrow \frac{count(relation)\ in\ genes}{total\ patients}$
end for

Phase 2: Assessment of New User

Necessary information are taken from a new volunteer in phase 2 of Algorithm 1. In this scope, we will be working on age, weight, gender and ancestral diabetic history of the volunteer. These metric values will be used to compare with the collected data during analysis in next phase.

Algorithm 1 Assess new user ($Phase2$)

$userage \leftarrow age\ of\ new\ user$
$userweight \leftarrow weight\ of\ new\ user$
$usergene \leftarrow genetic\ factor\ of\ new\ user$
$uergender \leftarrow weight\ gender\ of\ new\ user$

Phase 3: Processing the Knowledge Base

Different degree of tolerance are set for different metric values in Phase 3 of Algorithm 1. For example, the sugar level of diabetic patients having an age difference of C_1 from the user will be considered while determining the sugar level of the user. In the same manner, the sugar level of people having an weight difference of C_2 from the users weight or having the same gender as the user will also be evaluated while predicting the sugar level of the user. If one or more ancestors have diabetes in the users family then the sugar level of other diabetic patients whose same family relations have diabetes will be considered according to the percentage weight value of that ancestor relation which was been calculated in phase 1. The value of C_1, C_2 are determined from statistical study of data related to diabetic patients and these values may vary.

Algorithm 1 Process knowledge base ($Phase3$)

ages, weights, sugarlevels, genes, genders $\leftarrow 0$
for each age in ages **do**
  if $|age - userage| < C_1$ **then**
    $sumAge \leftarrow sumAge + correspondingsugarlevel$
    $ageCount \leftarrow ageCount + 1$
  end if
end for
for each weight in weights **do**
  if $|weight - userweight| < C_2$ **then**
    $sumWeight \leftarrow sumWeight + correspondingsugarlevel$
    $weightCount \leftarrow weightCount + 1$
  end if
end for
for each gene in genes **do**
  $sumGene \leftarrow 0$
  for each relation in gene **do**
    $sumGene \leftarrow sumGene + Expected\ avg\ sugarlevel$
  end for
end for
for each gender in genders **do**
  if $gender = userGender$ **then**
    $sumGender \leftarrow sumGender + correspondingsugarlevel$
    $enderCount \leftarrow genderCount + 1$
  end if
end for

Phase 4: Logic and Prediction

At phase 4 of Algorithm 1, based on data having similar parameters an average sugar level will be calculated. For age, the sum of all the sugar levels of the people having an age difference of C_1 from the user will be divided by the number of people who fall into this category. In this way an average sugar level will be calculated for the age factor. In the same way, avg sugar level will be calculated for weight, gender and ancestor factors. These average values will be summed and the divided by the number of factors to get the probable mean average value of sugar level. Based on that assumed sugar value, we will be able to tell the user if he falls in the safe zone or prediabetes category or risk zone or he already might be a diabetic patient with high diabetes value and based on these ranges it will also suggest the user to go to doctor or to test his sugar level or just remain alert.

Algorithm 1 Logic and prediction ($Phase4$)

1: $avgAge \leftarrow \frac{sumAge}{ageCount}$
2: $avgWeight \leftarrow \frac{sumWeight}{weightCount}$
3: $avgGender \leftarrow \frac{sumGender}{genderCount}$
4: $pridcedLevel \leftarrow avgAge * ageFactor + avgWeight * weightFactor + sumGene * geneFactor + avgGender * genderFactor$

Thus based on these four fields, the probable diabetic level of a person can be anticipated to an approximate range. For this, at first all the collected data are put in a file and the user is prompt for his/her data. After taking data from the user, each data is compared with the base dataset collected through survey. Firstly, a mean blood sugar level is calculated for each four criteria where values nearer to the user data are considered while calculating the users sugar level. Then those mean values are combined to get the final estimated sugar level of the user. For example, suppose a 30 years old male user has weight of 60 kg and his father and uncle have diabetes. Then his approximate average AC1 sugar level will be calculated by comparing with sugar level of diabetic male patients who have age in the range [30 C_1, 30 + C_1], weight in the range [60 C_2, 60 + C_2] and those whose father or uncle or both have diabetes. Based on the sugar approximate sugar level the system will recommend him about his probable diabetes status and steps he should take in this situation.

5.2 Algorithm Based on Naive Bayes

The initially developed algorithm has some limitations to it. So, machine learning was considered next to improve efficiency and accuracy. Machine learning is a type of artificial intelligence (AI) that allows software applications to become more accurate in predicting outcomes without being explicitly programmed [43]. In this process an algorithm is de-

veloped where based on given input and previously stored data, machine tries to predict the output as correctly as possible. The main classification of machine learning is given in Figure 5.1.

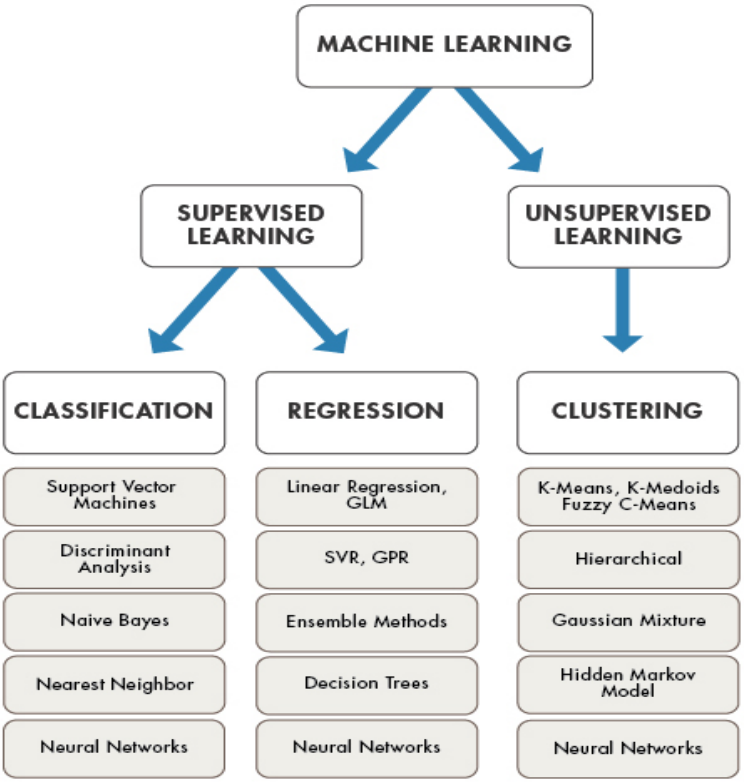

Figure 5.1: Classification of Machine Learning

Primarily, we choose Naive Bayes classifier for classification. In machine learning, Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) which are also mutually independent. The features considered in this research were found to be mutually independent as stated in Section 4.2. So, Naive Bayes algorithm is surely a feasible one for this research.

Abstractly, Naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector x = $\{x_1, x_2, ..., x_n\}$ representing some n features (independent variables), it assigns to this instance probabilities $p\{C_k|x1, ..., xn\}$ for each of K possible outcomes or classes. Using Bayes theorem, the conditional probability can be decomposed as

$$p(C|x) = \frac{p(Ck)p(x|Ck)}{p(x)} \tag{5.1}$$

The equation can be written as:

$$posterior = \frac{prior * likelihood}{evidence} \tag{5.2}$$

In Naive Bayes [37], if C is the estimated class and $\{x_1, x_2, ..., x_n\}$ are the features, the joint probability is defined as

$$P(C|x_1, x_2, ..., x_n) = P(C)\Pi_{i=1}^{n} P(x_i|C) \tag{5.3}$$

Here, $P(C|x_1, x_2, ..., x_n)$ represents the probability of being in class C having features $\{x_1, x_2, ..., x_n\}$. In this research, Gaussian Naive Bayes has been used for continuous features and Multinomial Naive Bayes has been used for the categorical features. The algorithm used here [Algorithm 2] contains two phases: training phase and testing phase; these two phases have been illustrated below:

Algorithm 2 Predicting the Possibility of Diabetes

1: $features \leftarrow \{age, weight, gender, inheritance\}$
2: $classes \leftarrow \{diabetic, prediabetic, nondiabetic\}$
3: **for** each c_i in classes **do**
4: calculate μ and σ of age and $weight$
5: **end for**
6: calculate $P(gender|c)$ where $gender \in \{male, female\}$ and $c \in classes$
7: calculate $P(inheritance|c)$ where $inheritance \in \{yes, no\}$ and $c \in classes$
8: **for** each case in test data **do**
9: **for** each c_i in classes **do**
10: $posterior(c_i) \leftarrow \frac{P(c_i)\Pi P(x_i|c_i)}{evidence}$ where $x_i \in features$
11: **end for**
12: $max\_posterior \leftarrow max(posterior(c_i \in classes))$
13: $resulting\_class \leftarrow c_i$ such that $c_i \in classes$ and $posterior(c_i) = max\_posterior$
14: **end for**

5.2.1 Training Phase

The first phase of this algorithm is for training the system [line number 3-7 in Algorithm 2]. The steps followed in this phase are:

- The system begins by taking the related training data on features (age, gender, weight, ancestral diabetic history) and classes (diabetic, prediabetic, nondiabetic) as input and enhances its knowledge base.
- After that, the mean (μ) and variance (σ) for age and weight have been calculated for each class [Gaussian Naive Bayes]

- Next, the conditional probability for categorical test data gender and ancestral diabetic history is calculated [Multinomial Naive Bayes] using equation 5.4 [38]. Here f represents categorial value for feature x (e.g., 'male' is a categorial value of feature 'gender') then for class C_i.

$$P(x|C_i) = \frac{k}{\text{No. of training data in } C_i} \tag{5.4}$$

where, k = No. of occurrence of f for feature x in training data of C_i.

5.2.2 Testing Phase

The second phase is for testing the system [line 8-14 in Algorithm 2]. The steps are given below:

- For test data, the conditional probability for age and weight $P(x|C)$ are calculated using equation 5.5, since factors age and weight are assumed to exhibit normal [Gaussian Naive Bayes] distribution in the data set.

$$P(x|C) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5.5}$$

- For each set of test case, their posterior probability $P(C|x)$ of belonging to each predefined class is calculated using equation no 5.3. The division by evidence is intentionally ignored as it scales the result for all the classes by same value.

- After that, the class with the maximum posterior value is taken to be the resultant class for that particular data set using the MAP (maximum a posteriori) decision rule.

An example is presented here to provide an overview on how the algorithm works to anticipate a probable condition of the patient of being diabetic, prediabetic or nondiabetic and classify accordingly. Suppose, for a particular feature (age or weight) from training set, mean is μ_1 and variance is σ_1 in case of diabetic, similarly consider μ_2 and σ_2 for class prediabetic and μ_3 and σ_3 for class nondiabetic. Taking the 1st data as test data and the rest as training data from the main dataset, the calculated values of mean and variance are given in Table 5.1.

Here in testing phase, the values of the features of a test case are age = '53' years, weight = '77' kg, gender = 'female', ancestral diabetic history='no'. Now, for class diabetic, putting these values in equation 5.5 we get, $P(age|diabetic) = 0.008$, $P(weight|diabetic) = 0.0061$. Similarly for prediabetic, $P(age|prediabetic) = 0.0081$, $P(weight|prediabetic) =$

Table 5.1: Mean and Variance Calculation For Continuous Features

Feature	μ_1	σ_1	μ_2	σ_2	μ_3	σ_3
Age	47.80	2459.64	46.50	2391.79	30.55	985.38
Weight	62.07	4051.96	69.29	4925.00	65.74	4586.36

0.0057 and for nondiabetic $P(age|nondiabetic) = 0.0098$, $P(weight|nondiabetic) = 0.0058$.

For categorial factors gender and inheritance, using equation 5.4 for class diabetic, it was found, $P(gender|diabetic) = 0.56$, $P(inheritance|diabetic) = 0.6$. Similarly for prediabetic, $P(gender|prediabetic) = 0.179$, $P(inheritance|prediabetic) = 0.93$, for nondiabetic, $P(gender|nondiabetic) = 0.254$, $P(inheritance|nondiabetic) = 0.817$. The prior probabilities are $P(diabetic) = 0.313$, $P(prediabetic) = 0.22$ and $P(nondiabetic) = 0.49$. Now using equation 5.3 yields the following posterior probabilities which will help to predict the class in which the test case belongs. After calculation, $P(diabetic|x) = 0.404$, $P(prediabetic|x) = 0.133$ and $P(nondiabetic|x) = 0.464$. As the posterior probability of being diabetic is high, the case is assumed to be under class diabetic. This example gives a complete picture of the overall working procedure of this algorithm.

5.3 Algorithm Based on Naive Bayes with Prior Clustering

The accuracy of the Naive Bayes algorithm over existing data was not very satisfying. To improve the accuracy and efficiency of the algorithm, unsupervised classification has been fused with the existing Naive Bayes' Classifier (supervised learning) as shown in Algorithm 3. The Algorithm 3 shows the improved Naive Bayes where the symbols represent the following meaning:

Dataset, $X = \{x_1, x_2, ..., x_n\}$,
$d(x, C)$ = dissimilarity between feature vector x and cluster C,
θ = threshold of dissimilarity,
q = maximum allowable clusters,
m = current cluster no. after each step,
N = total no. of training data,
x_{input} = input feature

This algorithm works in two stages. **Firstly**, BSAS clustering algorithm was implemented to cluster similar datasets. **Secondly**, Naive Bayes algorithm was run over the specific cluster to which the test case belongs. These two stages are explained below:

Algorithm 3 Improved Algorithm for Predicting the Possibility of Diabetes

$features \leftarrow \{age, weight, gender, inheritance\}$
$classes \leftarrow \{diabetic, prediabetic, nondiabetic\}$
$m = 1$
$C_m = \{x_1\}$
for $i = 2$ to N **do**
  Find $C_k : d(x_i, C_k) = min_{1 \leq j \leq m} d(x_i, C_j)$
  if $d(x_i, C_k) > \theta$ AND $m < q$ **then**
    $m = m + 1$
    $C_m = \{x_i\}$
  else
    $C_k = C_k \cup \{x_i\}$
  end if
end for
Find $C_k : d(x_{input}, C_k) = min_{1 \leq j \leq m} d(x_{input}, C_j)$
$C_{input} = C_k$
for each c_i in classes **do**
  calculate μ and σ of age and $weight$ for data contained in cluster C_{input}
end for
calculate $P(gender|c)$ for data contained in cluster C_{input} where $gender \in \{male, female\}$ and $c \in classes$
calculate $P(inheritance|c)$ for data contained in cluster C_{input} where $inheritance \in \{yes, no\}$ and $c \in classes$
for each case in test data of cluster C_{input} **do**
  for each c_i in classes **do**
    $posterior(c_i) \leftarrow \frac{P(c_i) \Pi P(x_i|c_i)}{evidence}$ where $x_i \in features$
  end for
  $max\_posterior \leftarrow max(posterior(c_i \in classes))$
  $resulting\_class \leftarrow c_i$ such that $c_i \in classes$ and $posterior(c_i) = max\_posterior$
end for

5.3.1 Clustering Using BSAS

Clustering is a type of unsupervised learning where the machine is not provided with any kind of training data, here the machine divides the given input data into different clusters based on the similarity of the features, it deals with finding a structure in a collection of unlabeled data. So, similar types of input data are expected to remain in the same cluster while data having distinguished features are contained in different clusters. A cluster is therefore a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters [44]. The classification of clustering algorithms is shown in Figure 5.2. There are different types of sequential clustering such as: BSAS, MBSAS, K-means. For now BSAS has been used in this algorithm to cluster the dataset as it's the basic one.

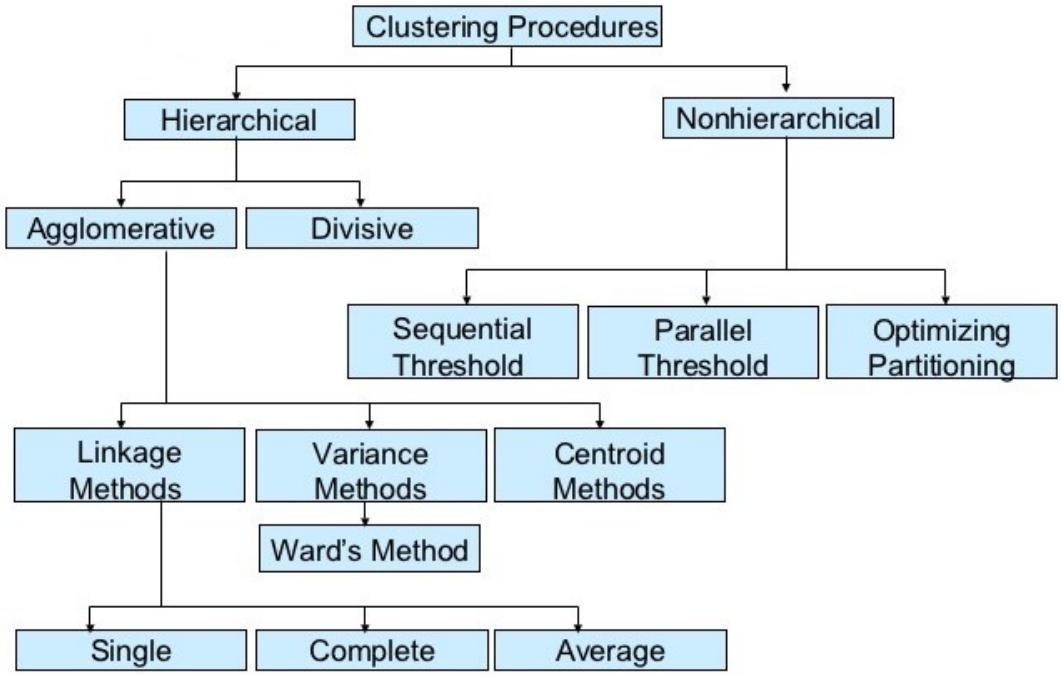

Figure 5.2: Classification of Clustering

In Algorithm 2 all the data present in the dataset were used to train the machine and then the machine would give output for the current input. But with the improved version, the machine will first divide all the data present in the dataset into different clusters as shown in Figure 5.3. In this algorithm, q=5 has been used which means the maximum number of clusters can be 5. Then based on similarity, machine will assign the user input data to anyone of the clusters. In Figure 5.3, 'Cluster 0' contains the user input (test) data.

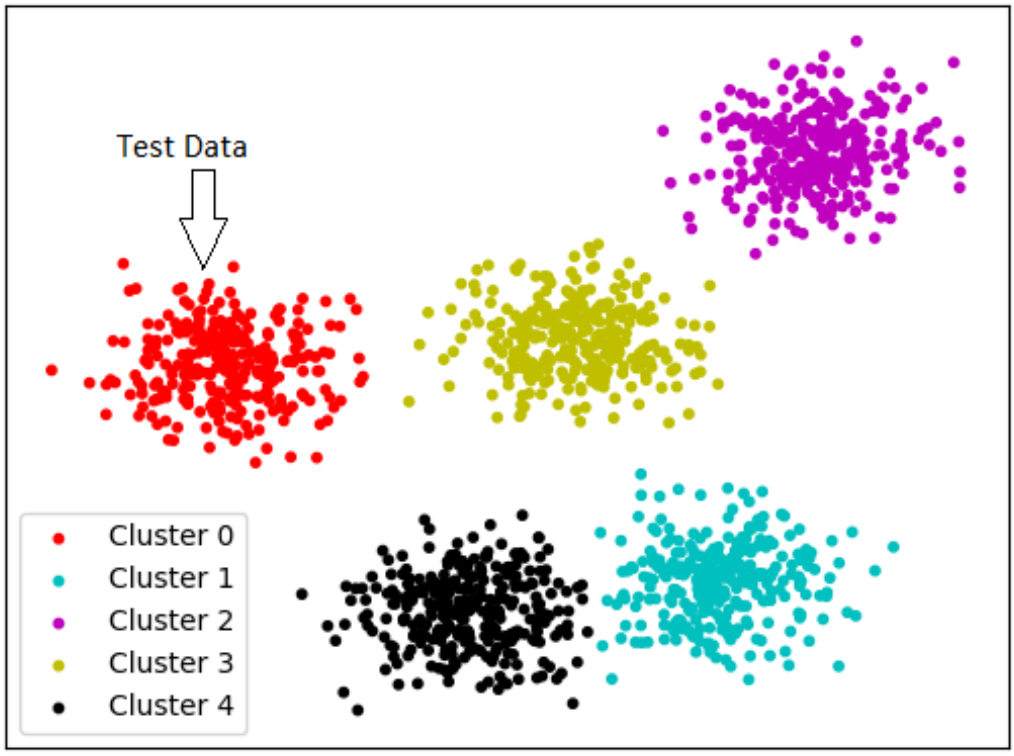

Figure 5.3: Dividing Data into Clusters

5.3.2 Naive Bayes on Specific Cluster

Then Naive Bayes will be applied to the dataset belonging to only that cluster containing the user input, according to this figure the Naive Bayes will be implemented only of the data contained in 'Cluster 0'. In this way the machine will be trained using only the similar types of data and possibility of misleading the machine will decrease. Just like Algorithm 2, Naive Bayes here will work in two stages, training and testing, in the similar manner as before but this time only on the specific cluster containing test data.

CHAPTER 6
EVALUATING INTELLIGENT PREDICTIVE ALGORITHM

The outcome of this analysis based on survey method could be used to suggest non-diagnosed persons on possible diabetes threat. The system could suggest how much a person can be potentially prone to diabetes using three categories:

- Non-diabetic, out of danger, risk is very low for now
- Pre-diabetic stage, need to remain alert and take precautions
- Diabetic, in danger (diagnosis needed)

The accuracy of the system is defined by how correctly the machine predicts the condition of the users. At each phase of developing the algorithms, the accuracy of the system was tested using various methodologies to check if the system being developed is factual using the data collected through survey. But due to limitation of the dataset acquired through survey, we looked for other datasets to check the accuracy of the algorithm. And in this process we found 'Pima Indian Diabetes Dataset', having considerably more number of records. Therefore, all three algorithms were tested using it. This dataset classifies all the subjects into two categories:

- Non-diabetic, out of danger, risk is very low for now
- Diabetic, in danger (diagnosis needed)

The evaluation method applied on both the 'Survey Dataset' and 'Pima Indian Diabetes Dataset' for all three algorithms stated in Chapter 5 has been discussed in this Chapter. Section 6.1 discusses the evaluation method of the algorithms based on 'Survey Dataset' while Section 6.2 focuses on the 'Pima Indian Diabetes Dataset'. At the same time, the results of the evaluations have been compared to show the overall progress of the research work.

6.1 Result Analysis Using Survey Dataset

6.1.1 Result Analysis of Algorithm 1(based on proximity)

The accuracy of the initially developed algorithm was examined using true positive-true negative method. To test the system, the algorithm was run over the dataset collected through survey. Later, their real conditions were compared to the prediction provided by the system. As the number of classes were more than two, conventional binary classification test was not done in this case. Rather, The number of correct assumptions by the classifier was assumed as the parameter for evaluating the system. Among 188 test cases, the number of correct prediction was 102. Which corresponds to an accuracy of 54.26%.

The probability of error of the classifier was not calculated for this algorithm as this algorithm does not follow any probabilistic approach for classification.

As this accuracy seemed very low, it was clear that the algorithm needed much more improvement. Also, in this algorithm, the range for considering age and BMI was predefined which is a limitation of the algorithm. So, machine learning technique was adopted in place of this algorithm.

6.1.2 Result Analysis of Algorithm 2 (Naive Bayes)

To estimate the probability of error of the system, we used the *Leaving-One-out-Technique* on the data set. Firstly, we used the first sample as test data and the rest $n-1$ samples were used as training data. Then again the second sample is used as test data while rests were used as training data. Thus we repeated the process $n-1$ times.

For each case, while estimating the error of the test sample we used the fractional counting which uses estimated probability of class membership of sample. If $\hat{P}(Ci|x)$ is the largest of probabilities for C_i class and sample x, then its probability of being erroneous is $\hat{P}(\varepsilon|x) = 1 - \hat{P}(Ci|x)$. For example, for a test case where a person's age = '30' years, weight = '60' kg, gender = 'male', ancestral diabetic history = 'no' and by representing class Diabetic by 'A', Prediabetic by 'B', Nondiabetic by 'C', we get $\hat{P}(\varepsilon|x) = 1 - 0.6 = 0.4$. Some similar cases are shown in Table 6.4

Table 6.1: Error Estimation

Test Case	True Class	$P(A\|x)$	$P(B\|x)$	$P(C\|x)$	$\hat{P}(\varepsilon\|x)$
1	A	0.69	0.03	0.28	0.31
2	B	0.37	0.12	0.52	0.49
3	C	0.34	0.11	0.55	0.45

Therefore, total estimated error for n samples has been calculated using the equation 6.1,

$$\hat{P}(\varepsilon) = \Sigma_i^n \hat{P}_i(\varepsilon|x) \tag{6.1}$$

Thus the resultant probability of error of the classifier, $\hat{P}(\varepsilon) = 0.303$ as shown in Figure 6.1. The number of correct assumptions are higher than that found using the previous algorithm. The accuracy calculated by the number of correct assumptions is 63.83%. The accuracy had increased evidently but still there was room for further improvement. So another approach was taken to improve the accuracy of the algorithm to some extent.

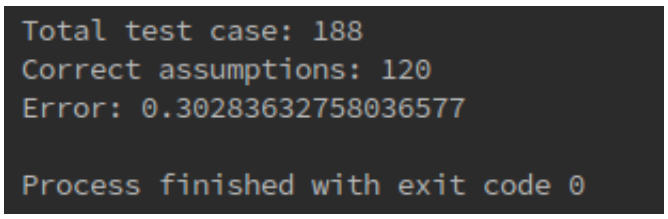

Figure 6.1: Accuracy of Implemented Naive Bayes Algorithm

6.1.3 Result Analysis of Algorithm 3 (Naive Bayes with prior clustering)

The Naive Bayes algorithm has been improved by clustering the training dataset along with the test case as stated before. The probability of error of the system was again checked using the *Leaving-One-Out-Technique* and *Fractional Counting* in the same manner as before. But this time the result was found to be better than before. It was seen that the algorithm gave the best result when maximum number of clusters, $q = 5$ and minimum distance, $\theta = 17$. The average probability of error of the system is now 0.263 as shown in Figure 6.2. The system now had an accuracy of 67.55% that is significantly higher than before. As we know that the dataset is very small, it is expected that with a larger dataset the accuracy of the system will rise even higher making it a reliable intelligent system to predict diabetes.

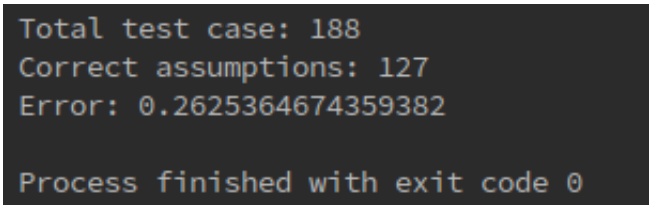

Figure 6.2: Accuracy of Implemented Improved Naive Bayes Algorithm

6.2 Result Analysis Using Pima Indian Diabetes Dataset

6.2.1 Result Analysis of Algorithm 1 (based on proximity)

As the dataset acquired by survey was very small for making an effective classifier with sufficient accuracy. Therefore, 'Pima Indian Diabetes Dataset' from UCI machine learning repository was used for training the classifier for better performance. 625 records out of 768 were found to be usable in this research.

The initially developed algorithm was evaluated using true positive-true negative method. To test the system, the algorithm was run over the 188 collected data and. Later, their real conditions were compared to the prediction provided by the system. For evaluating the system, we assumed 4 cases. Namely,

- True positive (TP) = the number of cases correctly identified as patient: 4 out of 625
- False positive (FP) = the number of cases incorrectly identified as patient: 15 out of 625
- True negative (TN) = the number of cases correctly identified as healthy: 393 out of 625
- False negative (FN) = the number of cases incorrectly identified as healthy: 213 out of 625

A Confusion Matrix is shown in Table 6.2 to discuss the cases more clearly:

Table 6.2: Confusion Matrix

	Actual Diabetic	Actual Non-diabetic	Total
Test Diabetic	True Positive=4	False Positive=15	19
Test Nondiabetic	False Negative=213	True Negative=393	606
Total	217	408	625

Finally, Accuracy = The ability to differentiate the diabetic and the non-diabetic cases correctly = (TN + TP)/(TN+TP+FN+FP) = 63.52%

Since the dataset contains only two classes, namely diabetic and non-diabetic, we were able to conduct binary classification test on the algorithm. Because of this, we calculated sensitivity and specificity of the system in addition to accuracy of the system.

Sensitivity = The ability to determine the diabetic cases correctly = TP/(TP + FN) = 1.84%
Specificity = The ability to determine the non-diabetic cases correctly = TN/(TN + FP) = 96.32%

6.2.2 Result Analysis of Algorithm 2 (Naive Bayes)

The error estimation for this dataset was done in the same way as the previous case. To test the system, the algorithm was run over the 188 collected data and . Later, their real conditions were compared to the prediction provided by the system. For evaluating the system, we assumed 4 cases. Namely,

- True positive (TP) = the number of cases correctly identified as patient: 5 out of 625
- False positive (FP) = the number of cases incorrectly identified as patient: 2 out of 625
- True negative (TN) = the number of cases correctly identified as healthy: 406 out of 625
- False negative (FN) = the number of cases incorrectly identified as healthy: 212 out of 625

Table 6.3: Confusion Matrix

	Actual Diabetic	Actual Non-diabetic	Total
Test Diabetic	True Positive=5	False Positive=2	7
Test Nondiabetic	False Negative=212	True Negative=406	618
Total	217	408	625

Finally, Accuracy = (TN + TP)/(TN+TP+FN+FP) = 65.76%
Sensitivity = TP/(TP + FN) = 2.3%
Specificity = TN/(TN + FP) = 99.51%

The probability of error of the classifier calculated by fractional counting and leaving one out technique was 0.201.

6.2.3 Result Analysis of Algorithm 3 (Naive Bayes with prior clustering)

The error estimation for this dataset was done in the same way as previous two cases. To test the system, the algorithm was run over the 188 collected data and . Later, their real

conditions were compared to the prediction provided by the system. For evaluating the system, 4 cases were assumed like before. They are,

- True positive (TP) = the number of cases correctly identified as patient: 68 out of 625
- False positive (FP) = the number of cases incorrectly identified as patient: 20 out of 625
- True negative (TN) = the number of cases correctly identified as healthy: 388 out of 625
- False negative (FN) = the number of cases incorrectly identified as healthy: 149 out of 625

Table 6.4: Confusion Matrix

	Actual Diabetic	Actual Non-diabetic	Total
Test Diabetic	True Positive=68	False Positive=20	88
Test Nondiabetic	False Negative=149	True Negative=388	537
Total	217	408	625

Finally, Accuracy = (TN + TP)/(TN+TP+FN+FP) = 72.96%
Sensitivity = TP/(TP + FN) = 31.34%
Specificity = TN/(TN + FP) = 95.1%

The probability of error of the classifier calculated by fractional counting and leaving one out technique was 0.227.

It can be seen that the performance of the classifier has been significantly better in every aspects when applied to the Pima Indian Diabetes Dataset. The increased number of data has resulted in improvements in different metrics.

6.3 Comparison of Accuracy of the Algorithms

Comparison of performance of algorithms on survey dataset are shown in table 6.5.

Comparison of performance of algorithms on 'Pima Indian Diabetes Dataset' are shown in table 6.6.

It is quite evident from the comparison that for all the three algorithms the result is significantly better when used on a larger dataset (Pima Indian Diabetes Dataset) in terms of

Table 6.5: Comparison of Performance Among three Algorithms Using Survey Dataset

Algorithm	Accuracy	Probability of Error
Algorithm 1 (calculated based on proximity) 1	54.26%	-
Naive Bayes 2	63.83%	0.303
Naive Bayes with prior clustering 3	67.55%	0.263

Table 6.6: Comparison of Performance Among three Algorithms Using Pima Indian Diabetes Dataset

Algorithm	Accuracy	Sensitivity	Specificity	Probability of Error
Algorithm 1 (based on proximity) 1	63.52%	1.84%	96.32%	-
Naive Bayes 2	65.76%	2.3%	99.51%	0.201
Naive Bayes with prior clustering 3	72.96%	31.34%	95.1%	0.227

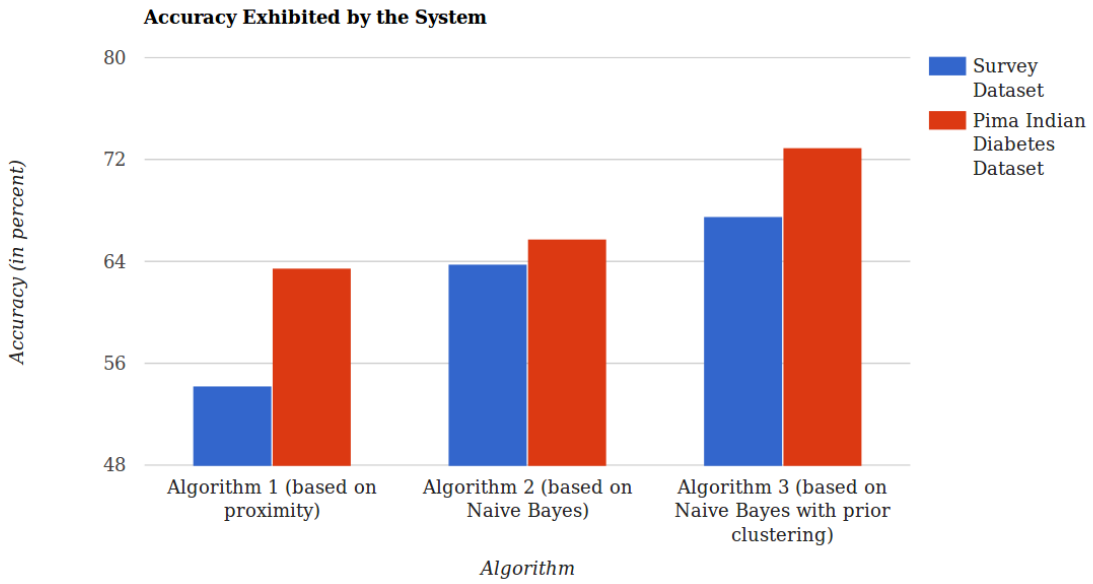

Figure 6.3: Comparison of Accuracy of Three Algorithms Applied on Both Datasets

accuracy. Also, for both Survey Dataset and Pima Indian Diabetes Dataset, the accuracy is significantly better for the updated predictive algorithm based on Naive Bayes with prior clustering 3 compared to other two algorithms. These findings are depicted on Figure: 6.3. However, the probability of error of the classifier was found to increase for the algorithm based on Naive Bayes with prior clustering while checked on Pima Indian Diabetes Dataset and decrease while checked on survey dataset.

CHAPTER 7
DEVELOPING MOBILE APPLICATION

This Chapter describes the system architecture of the application to be developed as well as the steps of developing the mobile application. The development platforms and features of the application have also been discussed in this section.

7.1 Development Architecture

The system is developed basing on sequential architecture. The development architecture has been depicted in Fig 7.1. Here data is stored in a database through a remote server. Using different transmission and security protocols data are sent from input fields to the remote server. After that, the data is is stored as training data in the database. This dataset is used as the training dataset for predicting diabetes.

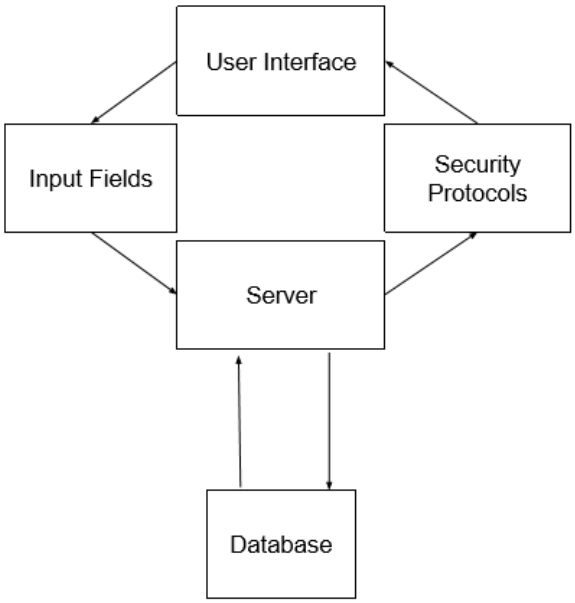

Figure 7.1: Development Architecture of the Mobile Application

7.2 Development Platform

The development specifications are here:

- The mobile application was developed on Android Studio (version 2.3.3).
- Apache server (version 2.4.18) was selected as the primary server for the application.
- The language used for designing the user interface was XML. Java was used as the language for handling the connections and application logic. PHP has been been used as the server side language.
- MySQL database was chosen for storing the knowledge base.
- Data transmission was done through Volley, a library provided by Google for asynchronous data transmission.

7.3 Implementing Application

To make this system easily usable by general people, a mobile application has been developed based on the classifier implemented using Naive Bayes classifier algorithm with prior clustering. The user can easily check user's diabetes risk with the help of this application. The application has been developed on Android Studio. Data collected through survey is stored in the MySQL database. Data is passed from the UI of the application using Volley in JSON format. The JSON data is received in the server. This acts as the training dataset for the system. The test data is taken from the user and sent to the server in a similar way. Then analyzing the data in the training dataset along with user input and using the predictive algorithm, it is decided whether the user lies in the diabetic, prediabetic or nondiabetic class. The application consists of two activities- a) User Input b) Result. The screenshot of the application has been shown in Figure 7.2.

In the user input page the application takes four inputs from the user with the help of a user interface. The four required inputs are:

- BMI (in kg/m^2)
- Age (in years)
- Gender (Male/ Female)
- Family members with diabetes (Father/ Mother/ Both/ None)

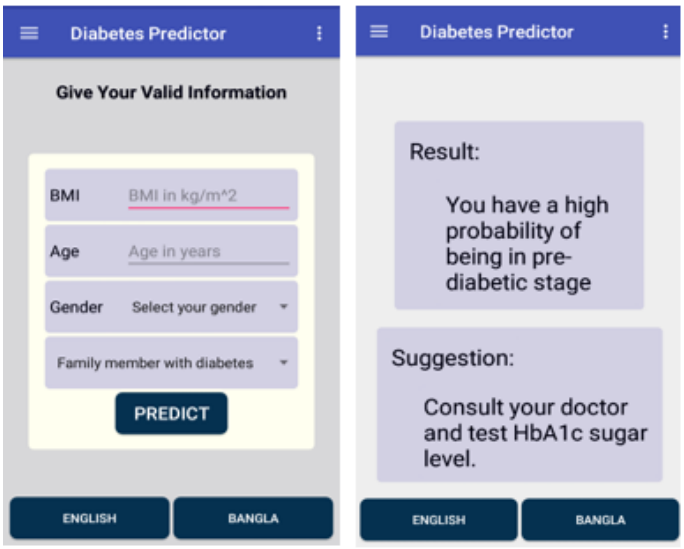

Figure 7.2: Sample User Interface of the Mobile Application

After providing the input in all the four fields, the user has to press the 'PREDICT' button to submit the input. Then based on this input and data stored in knowledge base, the classifier finds out the most probable condition of the user using the predictive algorithm. This predicted result is shown to the user along with a suggestive message using the resultant page of the application.

7.4 Features

The current features of the application are as follows:

- **Take input data from the user:** The BMI, age, gender and diabetic history of the family is taken as input from the user. These factors are easily known to anyone making this application usable by general mass. These inputs won't be submitted for analysis until the user fills up all the four required fields and press 'PREDICT' button.
- **Analyze the data and predict the result with the help of classifier in the server:** The classifier code snippet works in the back-end of the application in an Apache server. The information submitted by the user is processed and the algorithm is run to find the prediction result.
- **Show the prediction result to the user:** After giving user input the result page shows the user whether he/she lies in the diabetic, prediabetic or nondiabetic class.
- **Suggest the user to take necessary measures accordingly:** The user will be suggested to take one of the preset measures based on the predicted result. For example,

if the user is found to be in diabetic class, the application will suggest the user to consult a doctor and test his sugar level as early as possible.

CHAPTER 8
DISCUSSION AND CONCLUSION

This Chapter discusses about the main outcomes and practical implications of the system developed. It also points out to the limitations of the system and possible solutions to overcome these limitations and lastly draws a conclusion.

8.1 Main Outcomes

According to the outcomes, the study firstly shows that factors like age, weight, gender and ancestral diabetic history are the key determinants of individual's risk of being affected by diabetes. Secondly, data obtained through survey have served as knowledge base for the development of the intelligent diabetes prediction system that predicts whether the patient is in diabetic, prediabetic or nondiabetic class. Primarily a predictive algorithm, then Naive Bayes was applied on knowledge base. But for achieving an elevated level of accuracy we then implemented the system using an improved version of Naive Bayes that combines Naive Bayes and Basic Sequential Clustering algorithm. A mobile health application is developed so that users can easily see prediction of their possibility of being affected by diabetes.

8.2 Practical Implications

Diabetes is a silent disease and tests for its diagnosis is hardly done frequently in developed or in developing countries. People do not tend to visit hospitals on regular basis for diabetes tests. The intelligent prediction system will help any individual to know his/her probability of being diabetic even when at home without consulting a doctor. This reduces the efforts required to meet a physician in person. The proposed system is cost effective, yet gives the result right away and provides users with enough time to prevent and control diabetes by making them aware of their present condition. It can also be used by doctor as a mean of preliminary checkup. The domain of researchers can also be benefited from this work as it provides room for working on improvement of algorithm, modification and investigation of the application.

8.3 Study Limitations

The system has the following major limitations which are yet to be overcome:

8.3.1 Lack of Data

The primary limitation of the study is the lack of sufficiently large dataset which eventually lowered the accuracy. As we know, in supervised learning the machine learns by itself from the given training data. If the amount of data is really small, machine can't get familiar enough with all the possible cases and thus the tendency of making wrong assumptions tends to increase.

8.3.2 Lack of Features

The more independent features are used to teach the machine the more easily it can distinguish between the classes. In our system we only provided four basic features to the machine which can never be enough, specially in case of diabetes as we know clearly there are many other factors that impact the possibility of being affected by diabetes greatly. So, undoubtedly while keeping the features simple enough for the general mass, we also need to consider other features that have high impact factor on diabetes.

8.3.3 Usability Issues

Another limitation is the usability issues that are not considered explicitly to develop the mHealth system. The app needs to be made more feasible and attractive so that anyone can easily use this application.

8.4 Future Expansion

Following are the tasks that have been pointed out for further implementation:

8.4.1 Evaluating with Larger Dataset

As dataset constructs the knowledge base of the system which is utilized in learning stage, a larger dataset will enrich the knowledge base and provide better accuracy. Our main focus for future expansion is the enhancement of knowledge base by collecting sufficiently large amount of data that will improve the accuracy of the system to a reliable extent. For

larger amount of data we have already applied to authors who have worked in related field and looking forward to their reply. We also plan to collect as much data as we can using survey method. We will also incorporate the mechanism to enhance the knowledge base by considering the user (user of the mHealth app) data.

8.4.2 Adding More Features

We plan to include features like daily exercise, place of living, dietary habits, smoking habit, drinking habit as they have been found to be risk factors of diabetes in [45] and other studies also. Medical factors like hypertension, blood pressure, high cholesterol, polycystic ovary syndrome (for female) will also be considered in the predictive analysis of diabetes in our further research work as per the suggestion of some diabetes specialists.

8.4.3 Enhancing Accuracy

We intend to improve the algorithm even more by applying MBSAS, K-means instead of BSAS and also may try other approaches for improving it.

8.4.4 Exploring Other Techniques of Prediction

It has been found in several studies including [46] that the accuracy of Artificial Neural Network is considerably better than that of Naive Bayes classifier in case of diabetes prediction. Accuracy of Naive Bayes is high but still less than Artificial Neural Network. Therefore we intend to implement the prediction system in Artificial Neural Network during further research to see if a better accuracy can be achieved.

8.4.5 Evaluate Usability and UX of the App

It is expected that a feature will be included in the succeeding system which will provide medical advice to the ones who are under the threat of diabetes to make the application even more helpful. After making the above mentioned improvements, an extensive usability evaluation study will be caved out taking into account even a larger number of people to evaluate the usability and user experience (UX) of the application. Finally, proper advertisement of the application will be needed to make it familiar with the general mass.

8.5 Concluding Remarks

In the context of the global report on diabetes, the necessity of a system easily comprehensible and vastly reachable for people is indispensable. With a view to creating mass awareness on diabetes beforehand and reducing the death rate caused due to undiagnosed presence of diabetes, this research was conducted to develop an intelligent diabetes predictive system. In this reserach, survey method and machine learning techniques have been used for developing the predictive system with a view to support diabetic patients. Considering the present explosion rate of diabetes across the world, it is expected that the proposed system will play a vital role in creating awareness among people about diabetes as well as reducing the outspread of diabetes. This study thus will contribute to provide effective health services as well as to create awareness about diabetes.

Bibliography

[1] A. Iluyemi *et al.*, "5 community-based health workers in developing countries and the role of m-health," *Telehealth in the developing world*, p. 43, 2009.

[2] N. Kiongo, "A framework for mobile health adoption in developing countries: Case study kenya," *A framework for mobile health adoption in developing countries: Case study Kenya*, 2014.

[3] iMedicalApps.com, *More than 165,000 mobile health apps now available*, September 18, 2015 [Accessed August 01, 2017]. [Online]. Available: https://www.imedicalapps.com/2015/09/ims-health-apps-report/.

[4] M. Kay, J. Santos, and M. Takane, "mhealth: New horizons for health through mobile technologies," *World Health Organization*, vol. 64, no. 7, pp. 66–71, 2011.

[5] M. M. Karim, M. N. Islam, A. T. Priyoti, W. Ruheen, N. Jahan, P. L. Pritu, T. Dewan, and Z. T. Duti, "Mobile health applications in bangladesh: A state-of-the-art," in *Electrical Engineering and Information Communication Technology (ICEEICT), 2016 3rd International Conference on*. IEEE, 2016, pp. 1–5.

[6] W. H. Organization *et al.*, *Global report on diabetes*. World Health Organization, 2016.

[7] M. Arnhold, M. Quade, and W. Kirch, "Mobile applications for diabetics: a systematic review and expert-based usability evaluation considering the special requirements of diabetes patients age 50 years or older," *Journal of medical Internet research*, vol. 16, no. 4, 2014.

[8] WebCite.org, *International Diabetes Federation Diabetes Atlas*, 2013 [Accessed July 22, 2017]. [Online]. Available: https://www.webcitation.org/6JG2EAtt4.

[9] S. Akter, M. M. Rahman, S. K. Abe, and P. Sultana, "Prevalence of diabetes and prediabetes and their risk factors among bangladeshi adults: a nationwide survey," *Bulletin of the World Health Organization*, vol. 92, no. 3, pp. 204–213A, 2014.

[10] FutureStartup.com, *The State Of Diabetes In Bangladesh*, October 05, 2016 [Accessed August 05, 2017]. [Online]. Available: http://futurestartup.com/2016/07/27/the-state-of-diabetes-in-bangladesh/.

[11] M. A. David Ahn, *Diabetes Mobile Apps are many, but still lacking*, 2014 [Accessed August 01, 2017]. [Online]. Available: https://www.imedicalapps.com/2014/06/diabetes-mobile-apps-features/.

[12] M. A. Basar, H. N. Alvi, G. N. Bokul, M. S. Khan, F. Anowar, M. N. Huda, and K. A. Al Mamun, "A review on diabetes patient lifestyle management using mobile application," in *Computer and Information Technology (ICCIT), 2015 18th International Conference on*. IEEE, 2015, pp. 379–385.

[13] O. Raha, S. Chowdhury, S. Dasgupta, P. Raychaudhuri, B. Sarkar, P. V. Raju, and V. Rao, "Approaches in type 1 diabetes research: A status report," *International journal of diabetes in developing countries*, vol. 29, no. 2, p. 85, 2009.

[14] K. M. Orabi, Y. M. Kamal, and T. M. Rabah, "Early predictive system for diabetes mellitus disease," in *Industrial Conference on Data Mining*. Springer, 2016, pp. 420–427.

[15] M. Alghamdi, M. Al-Mallah, S. Keteyian, C. Brawner, J. Ehrman, and S. Sakr, "Predicting diabetes mellitus using smote and ensemble machine learning approach: The henry ford exercise testing (fit) project," *PLoS One*, vol. 12, no. 7, p. e0179805, 2017.

[16] N. V. Chawla and D. A. Davis, "Bringing big data to personalized healthcare: a patient-centered framework," *J. of general internal medicine*, vol. 28, no. 3, pp. 660–665, 2013.

[17] A. Sarwar and V. Sharma, "Comparative analysis of machine learning techniques in prognosis of type ii diabetes," *AI & society*, vol. 29, no. 1, pp. 123–129, 2014.

[18] J. A. Cafazzo, M. Casselman, N. Hamming, D. K. Katzman, and M. R. Palmert, "Design of an mhealth app for the self-management of adolescent type 1 diabetes: a pilot study," *Journal of medical Internet research*, vol. 14, no. 3, 2012.

[19] S. G. Mougiakakou, C. S. Bartsocas, E. Bozas, N. Chaniotakis, D. Iliopoulou, I. Kouris, S. Pavlopoulos, A. Prountzou, M. Skevofilakas, A. Tsoukalis *et al.*, "Smartdiab: a communication and information technology approach for the intelligent monitoring, management and follow-up of type 1 diabetes patients," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 3, pp. 622–633, 2010.

[20] J. L. Breault, C. R. Goodall, and P. J. Fos, "Data mining a diabetic data warehouse," *Artificial intelligence in medicine*, vol. 26, no. 1, pp. 37–54, 2002.

[21] J. W. Sakshaug, D. R. Weir, and L. H. Nicholas, "Identifying diabetics in medicare claims and survey data: implications for health services research," *BMC health services research*, vol. 14, no. 1, p. 150, 2014.

[22] J. Marx, "Unraveling the causes of diabetes," 2002.

[23] C.-S. Lee and M.-H. Wang, "A fuzzy expert system for diabetes decision support application," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 139–153, 2011.

[24] N. Barakat, A. P. Bradley, and M. N. H. Barakat, "Intelligible support vector machines for diagnosis of diabetes mellitus," *IEEE transactions on information technology in biomedicine*, vol. 14, no. 4, pp. 1114–1120, 2010.

[25] K. Polat, S. Güneş, and A. Arslan, "A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine," *Expert systems with applications*, vol. 34, no. 1, pp. 482–487, 2008.

[26] M. J. Handel, "mhealth (mobile health)using apps for health and wellness," *EXPLORE: The Journal of Science and Healing*, vol. 7, no. 4, pp. 256–261, 2011.

[27] J. A. Blaya, H. S. Fraser, and B. Holt, "E-health technologies show promise in developing countries," *Health Affairs*, vol. 29, no. 2, pp. 244–251, 2010.

[28] E. Årsand, D. H. Frøisland, S. O. Skrøvseth, T. Chomutare, N. Tatara, G. Hartvigsen, and J. T. Tufano, "Mobile health applications to assist patients with diabetes: lessons learned and design implications," *Journal of diabetes science and technology*, vol. 6, no. 5, pp. 1197–1206, 2012.

[29] E. Georga, V. Protopappas, A. Guillen, G. Fico, D. Ardigo, M. T. Arredondo, T. P. Exarchos, D. Polyzos, and D. I. Fotiadis, "Data mining for blood glucose prediction and knowledge discovery in diabetic patients: The metabo diabetes modeling and management system," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pp. 5633–5636.

[30] D. Preuveneers and Y. Berbers, "Mobile phones assisting with health self-care: a diabetes case study," in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. ACM, 2008, pp. 177–186.

[31] W. Xu and Y. Liu, "mhealthapps: a repository and database of mobile health apps," *JMIR mHealth and uHealth*, vol. 3, no. 1, 2015.

[32] MobiHealthNews.com, *Report: 24M people will use diabetes apps in 2018*, January 21, 2014 [Accessed August 10, 2017]. [Online]. Available: http://www.mobihealthnews.com/29079/report-24m-people-will-use-diabetes-apps-in-2018.

[33] *Android Apps on Google Play*. [Online]. Available: https://play.google.com/store?hl=en.

[34] S. Rajasekar, P. Philominathan, and V. Chinnathambi, "Research methodology," *arXiv preprint physics/0601009*, 2006.

[35] *Types of Machine Learning Algorithms You Should Know*, June 15, 2017 [Accessed August 1, 2017]. [Online]. Available: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861

[36] S. A. Pattekari and A. Parveen, "Prediction system for heart disease using naïve bayes," *International J. of Advanced Computer and Mathematical Sciences*, vol. 3, no. 3, pp. 290–294, 2012.

[37] D. Lowd and P. Domingos, "Naive bayes models for probability estimation," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 529–536.

[38] S. Theodoridis and K. Koutroumbas, "Pattern recognition and neural networks," *Machine Learning and Its Applications: Advanced Lectures*, vol. 2049, p. 169, 2003.

[39] R. M. Groves, F. J. Fowler Jr, M. P. Couper, J. M. Lepkowski, E. Singer, and R. Tourangeau, *Survey methodology*. John Wiley & Sons, 2011, vol. 561.

[40] *What are categorical, discrete, and continuous variables?*, 2016 [Accessed November 12, 2017]. [Online]. Available: http://support.minitab.com/en-us/minitab-express/1/help-and-how-to/modeling-statistics/regression/supporting-topics/basics/what-are-categorical-discrete-and-continuous-variables/

[41] *Pearson Correlation Coefficient Calculator*. [Online]. Available: http://www.socscistatistics.com/tests/pearson/

[42] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[43] M. Rouse, *Machine Learning*, 2017 [Accessed September 03, 2017]. [Online]. Available: http://whatis.techtarget.com/definition/machine-learning

[44] *Clustering: An Introduction*. [Online]. Available: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/1

[45] K. Asadollahi, P. Asadollahi, M. Azizi, and G. Abangah, "A self-assessment predictive model for type 2 diabetes or impaired fasting glycaemia derived from a population-based survey," *Diabetes Research and Clinical Practice*, vol. 131, 2017.

[46] N. Nai-arun and R. Moungmai, “Comparison of classifiers for the risk of diabetes prediction,” *Procedia Computer Science*, vol. 69, pp. 132–142, 2015.

APPENDIX A
Source Code of Predictive Algorithm Based on Proximity (Algorithm 1)

```
import java.util.ArrayList;
import java.util.Scanner;

public class Main {
   public static int geneConvert(String str){
      if(str.equals("father")){
         return 1;
      }
      else if(str.equals("mother")){
         return 2;
      }
      else if(str.equals("grandfather")){
         return 3;
      }
      else if(str.equals("grandmother")){
         return 4;
      }
      else if(str.equals("brother")){
         return 5;
      }
      else if(str.equals("sister")){
         return 6;
      }
      else if(str.equals("uncle")){
         return 7;
      }
      else if(str.equals("aunt")){
         return 8;
      }
      else {
         return 0;
      }
   }

   public static double avg(Double sum, int count){
```

```
        if(count == 0) return 0;
        return sum / (double) count;
    }

    public static void main(String args[]){
        Scanner input = new Scanner(System.in);

        // Taking input from user
        System.out.println("Age in years in years: ");
        int userAge = Integer.valueOf(input.nextLine());
        System.out.println("Weight in kg: ");
        int userWeight = Integer.valueOf(input.nextLine());
        System.out.println("Who has diabetic among your close
            relatives?");
        System.out.println("None: 0, Father: 1, Mother: 2, Grand
            Father: 3, Grand Mother: 4, Brother: 5, Sister: 6,
            Uncle: 7, Aunt: 8");
        int userGene = Integer.valueOf(input.nextLine());

        // Take input from csv file
        ArrayList age = new ArrayList();
        ArrayList weight = new ArrayList();
        ArrayList gene = new ArrayList();
        ArrayList sugarLevel = new ArrayList();
        // Headline
        String temp1 = input.nextLine();
        // Actual data
        int count = 0;
        while(input.hasNextLine()){
            String temp = input.nextLine();
            String store[] = temp.split(",");
            age.add(store[3]);
            weight.add(store[4]);
            sugarLevel.add(store[5]);
            gene.add(geneConvert(store[6]));
            count++;
        }
        // Calculate level from age
        int countAge = 0;
        double sumAge = 0;
        for(int i = 0; i < count; i++){
```

```
    if(Double.valueOf(age.get(i).toString()) < userAge + 5
        && Double.valueOf(age.get(i).toString()) > userAge -
        5){
        countAge++;
        sumAge +=
            Double.valueOf(sugarLevel.get(i).toString());
    }
}

// Calculate level from weight
int countWeight = 0;
double sumWeight = 0;
for(int i = 0; i < count; i++){
    if(Double.valueOf(weight.get(i).toString()) < userWeight
        + 8 && Double.valueOf(weight.get(i).toString()) >
        userWeight - 8){
        countWeight++;
        sumWeight +=
            Double.valueOf(sugarLevel.get(i).toString());
    }
}

// Calculate level from Genetic factor
int countGene = 0;
double sumGene = 0;
for(int i = 0; i < count; i++){
    if(Integer.valueOf(gene.get(i).toString()) == userGene){
        countGene++;
        sumGene +=
            Double.valueOf(sugarLevel.get(i).toString());
    }
}

double avgAge = avg(sumAge, countAge);
System.out.println("Avg sugar level from age: " + avgAge);
System.out.println();
double avgWeight = avg(sumWeight, countWeight);
System.out.println("Avg sugar level from weight: " +
    avgWeight);
System.out.println();
double avgGene = avg(sumGene, countGene);
```

```
        System.out.println("Avg sugar level from genetic factor: "
            + avgGene);
        System.out.println();

        double realSum = 0;
        int realCount = 0;
        if(countAge > 0){
            realSum += avgAge;
            realCount++;
        }
        if(countWeight > 0){
            realSum += avgWeight;
            realCount++;
        }
        if(countGene > 0){
            realSum += avgGene;
            realCount++;
        }

        double realAvg = avg(realSum, realCount);
        System.out.print("Probable sugar level: ");
        if(realCount > 0){
            System.out.println(realAvg);
        }
        else{
            System.out.println("Not enough base data");
        }
    }
}
```

APPENDIX B
Source Code of Predictive Algorithm Based on Naive Bayes (Algorithm 2)

```
package GaussianNaiveBayes;

import java.util.ArrayList;
import java.util.Scanner;


public class MainNaiveBayesPima {
   public static int calculateInheritanceFactor(String string){
      String[] array = string.split(";");
      for(int i = 0; i < array.length; i++){
         if(array[i].equalsIgnoreCase("father") ||
            array[i].equalsIgnoreCase("mother")){
            return 1;
         }
      }
      return 0;
   }

   public static int calculateGenderFactor(String string){
      if(string.equalsIgnoreCase("male")){
         return 0;
      }
      return 1;
   }

   public static double calculateConditionalProbability(double
      value, double mean, double variance){
      double ans = 1 / Math.sqrt(2 * Math.PI * variance);
      ans *= Math.exp((-1) * (value - mean) * (value - mean) / (2
         * variance));
      return ans;
   }

   public static double square(double a){
      return a * a;
```

```
}

public static double updateAvg(double avg, double n, double
    num){
    return (avg * (n - 1) + num) / n;
}

public static void main(String[] args) throws Exception{
    ArrayList ageList = new ArrayList();
    ArrayList bmiList = new ArrayList();
    ArrayList pregnantList = new ArrayList();
    ArrayList glucoseList = new ArrayList();
    ArrayList pressureList = new ArrayList();
    ArrayList classList = new ArrayList();

    Scanner input = new Scanner(System.in);

    input.nextLine();

    while(input.hasNextLine()){
        String temp = input.nextLine();
        String store[] = temp.split(",");
        double age = Double.valueOf(store[0]);
        double bmi = Double.valueOf(store[1]);
        double pregnant = Double.valueOf(store[2]);
        double glucose = Double.valueOf(store[3]);
        double pressure = Double.valueOf(store[4]);
        if(age == 0 || bmi == 0 || pregnant == 0 || glucose == 0
            || pressure == 0){
            continue;
        }
        String className = store[5];
        ageList.add(age);
        bmiList.add(bmi);
        pregnantList.add(pregnant);
        glucoseList.add(glucose);
        pressureList.add(pressure);
        classList.add(className);
    }
    int totalCount = classList.size();
```

```
int correctCount = 0;
double errorSum = 0.0;
for(int i = 0; i < totalCount; i++){ // leaving one out at
   each iteration
   //Clustering parameters
   int q = 5; // Maximum number of clusters
   double theta = 17; // Maximum distance to be in same
      cluster;
   int m = 1; // Initialize the number of clusters
   int testCase = i;
   int diabeticCount = 0;
   int preDiabeticCount = 0;
   int nonDiabeticCount = 0;

   // Metrices for diabetic
   double diabeticAgeSum = 0;
   double diabeticBmiSum = 0;
   double diabeticPregnantSum = 0;
   double diabeticGlucoseSum = 0;
   double diabeticPressureSum = 0;

   double diabeticAgeSqSum = 0;
   double diabeticBmiSqSum = 0;
   double diabeticPregnantSqSum = 0;
   double diabeticGlucoseSqSum = 0;
   double diabeticPressureSqSum = 0;

   // Metrices for non diabetic
   double nonDiabeticAgeSum = 0;
   double nonDiabeticBmiSum = 0;
   double nonDiabeticPregnantSum = 0;
   double nonDiabeticGlucoseSum = 0;
   double nonDiabeticPressureSum = 0;

   double nonDiabeticAgeSqSum = 0;
   double nonDiabeticBmiSqSum = 0;
   double nonDiabeticPregnantSqSum = 0;
   double nonDiabeticGlucoseSqSum = 0;
   double nonDiabeticPressureSqSum = 0;

   for(int it = 0; it < totalCount; it++){
```

```
int curIndex = it;
if(curIndex == testCase){
   continue;
}
double age =
   Double.valueOf(ageList.get(curIndex).toString());
double bmi =
   Double.valueOf(bmiList.get(curIndex).toString());
double pregnant =
   Double.valueOf(pregnantList.get(curIndex).toString());
double glucose =
   Double.valueOf(glucoseList.get(curIndex).toString());
double pressure =
   Double.valueOf(pressureList.get(curIndex).toString());
if(classList.get(curIndex).toString().equalsIgnoreCase("0")){
   nonDiabeticCount++;
   nonDiabeticAgeSum += age;
   nonDiabeticAgeSqSum += (age * age);
   nonDiabeticBmiSum += bmi;
   nonDiabeticBmiSqSum += (bmi * bmi);
   nonDiabeticPregnantSum += pregnant;
   nonDiabeticPregnantSqSum += (pregnant * pregnant);
   nonDiabeticGlucoseSum += glucose;
   nonDiabeticGlucoseSqSum += (glucose * glucose);
   nonDiabeticPressureSum += pressure;
   nonDiabeticPressureSqSum += (pressure * pressure);
}

else{
   diabeticCount++;
   diabeticAgeSum += age;
   diabeticAgeSqSum += (age * age);
   diabeticBmiSum += bmi;
   diabeticBmiSqSum += (bmi * bmi);
   diabeticPregnantSum += pregnant;
   diabeticPregnantSqSum += (pregnant * pregnant);
   diabeticGlucoseSum += glucose;
   diabeticGlucoseSqSum += (glucose * glucose);
   diabeticPressureSum += pressure;
   diabeticPressureSqSum += (pressure * pressure);
}
```

```
}
double diabeticAgeMean = (diabeticAgeSum /
   diabeticCount);
double diabeticAgeVariance = (diabeticAgeSqSum /
   diabeticCount) - (diabeticAgeMean);
double diabeticBmiMean = (diabeticBmiSum /
   diabeticCount);
double diabeticBmiVariance = (diabeticBmiSqSum /
   diabeticCount) - (diabeticBmiMean);
double diabeticPregnantMean = (diabeticPregnantSum /
   diabeticCount);
double diabeticPregnantVariance = (diabeticPregnantSqSum
   / diabeticCount) - (diabeticPregnantMean);
double diabeticGlucoseMean = (diabeticGlucoseSum /
   diabeticCount);
double diabeticGlucoseVariance = (diabeticGlucoseSqSum /
   diabeticCount) - (diabeticGlucoseMean);
double diabeticPressureMean = (diabeticPressureSum /
   diabeticCount);
double diabeticPressureVariance = (diabeticPressureSqSum
   / diabeticCount) - (diabeticPressureMean);

double nonDiabeticAgeMean = (nonDiabeticAgeSum /
   nonDiabeticCount);
double nonDiabeticAgeVariance = (nonDiabeticAgeSqSum /
   nonDiabeticCount) - (nonDiabeticAgeMean);
double nonDiabeticBmiMean = (nonDiabeticBmiSum /
   nonDiabeticCount);
double nonDiabeticBmiVariance = (nonDiabeticBmiSqSum /
   nonDiabeticCount) - (nonDiabeticBmiMean);
double nonDiabeticPregnantMean = (nonDiabeticPregnantSum
   / nonDiabeticCount);
double nonDiabeticPregnantVariance =
   (nonDiabeticPregnantSqSum / nonDiabeticCount) -
   (nonDiabeticPregnantMean);
double nonDiabeticGlucoseMean = (nonDiabeticGlucoseSum /
   nonDiabeticCount);
double nonDiabeticGlucoseVariance =
   (nonDiabeticGlucoseSqSum / nonDiabeticCount) -
   (nonDiabeticGlucoseMean);
double nonDiabeticPressureMean = (nonDiabeticPressureSum
```

```
    / nonDiabeticCount);
double nonDiabeticPressureVariance =
    (nonDiabeticPressureSqSum / nonDiabeticCount) -
    (nonDiabeticPressureMean);

System.out.println("Diabetic: " + diabeticCount + ",
    Pre: " + preDiabeticCount + ", Non: " +
    nonDiabeticCount);
System.out.println("Feature\tDiabetic mean\tDiabetic
    variance\tPre mean\tPre variance\tNon mean\tNon
    variance");
System.out.println("Age\t" + diabeticAgeMean + "\t" +
    diabeticAgeVariance + "\t" + nonDiabeticAgeMean +
    "\t" + nonDiabeticAgeVariance);
System.out.println("Bmi\t" + diabeticBmiMean + "\t" +
    diabeticBmiVariance + "\t" + nonDiabeticBmiMean +
    "\t" + nonDiabeticBmiVariance);
System.out.println("Pregnant\t" + diabeticPregnantMean +
    "\t" + diabeticPregnantVariance + "\t" +
    nonDiabeticPregnantMean + "\t" +
    nonDiabeticPregnantVariance);
System.out.println("Glucose\t" + diabeticGlucoseMean +
    "\t" + diabeticGlucoseVariance + "\t" +
    nonDiabeticGlucoseMean + "\t" +
    nonDiabeticGlucoseVariance);
System.out.println("Pressure\t" + diabeticPressureMean +
    "\t" + diabeticPressureVariance + "\t" +
    nonDiabeticPressureMean + "\t" +
    nonDiabeticPressureVariance);

int totalCountInCluster = diabeticCount +
    preDiabeticCount + nonDiabeticCount;


double diabeticPrior = (double) diabeticCount / (double)
    totalCountInCluster;
double preDiabeticPrior = (double) preDiabeticCount /
    (double) totalCountInCluster;
double nonDiabeticPrior = (double) nonDiabeticCount /
    (double) totalCountInCluster;
```

```
double testAge = (double) ageList.get(testCase);
double testBmi = (double) bmiList.get(testCase);
double testPregnant = (double)
   pregnantList.get(testCase);
double testGlucose = (double) glucoseList.get(testCase);
double testPressure = (double)
   pressureList.get(testCase);
String testClass = classList.get(testCase).toString();
double error = 0;

System.out.println("Test #" + (testCase + 1) + ":");
System.out.println();
System.out.println("Age: " + testAge + ", Bmi: " +
   testBmi + ", Pregnant: " + testPregnant + ", Glucose:
   " + testGlucose + ", Pressure: " + testPressure );

// Test for diabetic
double diabeticAgeConditional =
   calculateConditionalProbability(testAge,
   diabeticAgeMean, diabeticAgeVariance);
double diabeticBmiConditional =
   calculateConditionalProbability(testBmi,
   diabeticBmiMean, diabeticBmiVariance);
double diabeticPregnantConditional =
   calculateConditionalProbability(testPregnant,
   diabeticPregnantMean, diabeticPregnantVariance);
double diabeticGlucoseConditional =
   calculateConditionalProbability(testGlucose,
   diabeticGlucoseMean, diabeticGlucoseVariance);
double diabeticPressureConditional =
   calculateConditionalProbability(testPressure,
   diabeticPressureMean, diabeticPressureVariance);
double diabeticPosteriorNumerator = diabeticPrior *
   diabeticAgeConditional * diabeticBmiConditional *
   diabeticPregnantConditional *
   diabeticGlucoseConditional *
   diabeticPressureConditional;
System.out.println("Diabetic: " + diabeticPrior + " " +
   diabeticAgeConditional + " " + diabeticBmiConditional
   + " " + diabeticPregnantConditional + " " +
   diabeticGlucoseConditional + " " +
```

```
        diabeticPressureConditional);

    // Test for non diabetic

    double nonDiabeticAgeConditional =
        calculateConditionalProbability(testAge,
        nonDiabeticAgeMean, nonDiabeticAgeVariance);
    double nonDiabeticBmiConditional =
        calculateConditionalProbability(testBmi,
        nonDiabeticBmiMean, nonDiabeticBmiVariance);
    double nonDiabeticPregnantConditional =
        calculateConditionalProbability(testPregnant,
        nonDiabeticPregnantMean, nonDiabeticPregnantVariance);
    double nonDiabeticGlucoseConditional =
        calculateConditionalProbability(testGlucose,
        nonDiabeticGlucoseMean, nonDiabeticGlucoseVariance);
    double nonDiabeticPressureConditional =
        calculateConditionalProbability(testPressure,
        nonDiabeticPressureMean, nonDiabeticPressureVariance);
    double nonDiabeticPosteriorNumerator = nonDiabeticPrior
        * nonDiabeticAgeConditional *
        nonDiabeticBmiConditional *
        nonDiabeticPregnantConditional *
        nonDiabeticGlucoseConditional *
        nonDiabeticPressureConditional;
    System.out.println("non diabetic: " + nonDiabeticPrior +
        " " + nonDiabeticAgeConditional + " " +
        nonDiabeticBmiConditional + " " +
        nonDiabeticPregnantConditional + " " +
        nonDiabeticGlucoseConditional + " " +
        nonDiabeticPressureConditional);

    double probabilitySum = diabeticPosteriorNumerator +
        nonDiabeticPosteriorNumerator;
    System.out.println("P(diabetic): " +
        diabeticPosteriorNumerator / probabilitySum);

    System.out.println("Diagnosis (real) result: " +
        testClass);
    String testResult;
    if(nonDiabeticPosteriorNumerator >=
```

```
            diabeticPosteriorNumerator){
            testResult = "0";
            error = (diabeticPosteriorNumerator / probabilitySum);
         }

         else{
            testResult = "1";
            error = (nonDiabeticPosteriorNumerator /
               probabilitySum);
         }
         errorSum += error;

         System.out.println("Test Result: " + testResult);
         if(testResult.equalsIgnoreCase(testClass)){
            System.out.println("Prediction: Correct!");
            correctCount++;
         }
         else{
            System.out.println("Prediction: Incorrect!");
         }
         System.out.println();
         System.out.println();
      }
      System.out.println("Total test case: " + totalCount);
      System.out.println("Correct assumptions: " + correctCount);
      System.out.println("Error: " + (errorSum / totalCount));

   }
}
```

Source Code of Predictive Algorithm Based on Naive Bayes with Prior Clustering (Algorithm 3)

```
package GaussianNaiveBayes;

import java.util.ArrayList;
import java.util.Scanner;

public class MainBSASNaiveBayesPima {
   public static int calculateInheritanceFactor(String string){
      String[] array = string.split(";");
      for(int i = 0; i < array.length; i++){
         if(array[i].equalsIgnoreCase("father") ||
            array[i].equalsIgnoreCase("mother")){
            return 1;
         }
      }
      return 0;
   }

   public static int calculateGenderFactor(String string){
      if(string.equalsIgnoreCase("male")){
         return 0;
      }
      return 1;
   }

   public static double calculateConditionalProbability(double
      value, double mean, double variance){
      double ans = 1 / Math.sqrt(2 * Math.PI * variance);
      ans *= Math.exp((-1) * (value - mean) * (value - mean) / (2
         * variance));
      return ans;
   }

   public static double square(double a){
      return a * a;
   }
```

```
public static double updateAvg(double avg, double n, double
   num){
   return (avg * (n - 1) + num) / n;
}

public static void main(String[] args) throws Exception{
   ArrayList ageList = new ArrayList();
   ArrayList bmiList = new ArrayList();
   ArrayList pregnantList = new ArrayList();
   ArrayList glucoseList = new ArrayList();
   ArrayList pressureList = new ArrayList();
   ArrayList classList = new ArrayList();

   Scanner input = new Scanner(System.in);

   input.nextLine();

   while(input.hasNextLine()){
      String temp = input.nextLine();
      String store[] = temp.split(",");
      System.out.println("bingo");
      double age = Double.valueOf(store[0]);
      double bmi = Double.valueOf(store[1]);
      double pregnant = Double.valueOf(store[2]);
      double glucose = Double.valueOf(store[3]);
      double pressure = Double.valueOf(store[4]);
      if(age == 0 || bmi == 0 || pregnant == 0 || glucose == 0
         || pressure == 0){
         continue;
      }
      String className = store[5];
      ageList.add(age);
      bmiList.add(bmi);
      pregnantList.add(pregnant);
      glucoseList.add(glucose);
      pressureList.add(pressure);
      classList.add(className);
   }
   int totalCount = classList.size();
```

```
int correctCount = 0;
double errorSum = 0.0;
for(int i = 0; i < totalCount; i++){ // leaving one out at
   each iteration
   //Clustering parameters
   int q = 5; // Maximum number of clusters
   double theta = 17; // Maximum distance to be in same
      cluster;
   int m = 1; // Initialize the number of clusters

   int testCase = i;
   ArrayList[] clusters = new ArrayList[10];
   double[] clusterAgeAvg = new double[10];
   double[] clusterBmiAvg = new double[10];
   double[] clusterPregnantAvg = new double[10];
   double[] clusterGlucoseAvg = new double[10];
   double[] clusterPressureAvg = new double[10];
   for(int j = 0; j < 10; j++){
      clusters[j] = new ArrayList();
      clusterAgeAvg[j] = 0;
      clusterBmiAvg[j] = 0;
      clusterPregnantAvg[j] = 0;
      clusterGlucoseAvg[j] = 0;
      clusterPressureAvg[j] = 0;
   }

   int testCaseCluster = -1;
   clusters[0].add(0);
   clusterAgeAvg[0] = (double)ageList.get(0);
   clusterBmiAvg[0] = (double)bmiList.get(0);
   clusterPregnantAvg[0] = (double)pregnantList.get(0);
   clusterGlucoseAvg[0] = (double)glucoseList.get(0);
   clusterPressureAvg[0] = (double)pressureList.get(0);
   if(testCase == 0){
      testCaseCluster = 0;
   }
   for(int it = 1; it < totalCount; it++){
      double minDistance = 9999999.0;
      int minIndex = -1;
      double age = (double)ageList.get(it);
      double bmi = (double)bmiList.get(it);
```

```
double pregnant = (double)pregnantList.get(it);
double glucose = (double)glucoseList.get(it);
double pressure = (double)pressureList.get(it);
for(int it1 = 0; it1 < m; it1++){
   double distance = Math.sqrt(square(age -
      clusterAgeAvg[it1]) + square(bmi -
      clusterBmiAvg[it1]) + square(pregnant -
      clusterPregnantAvg[it1]) + square(glucose -
      clusterGlucoseAvg[it1]) + square(pressure -
      clusterPressureAvg[it1]) );
   if(distance < minDistance){
      minDistance = distance;
      minIndex = it1;
   }
}

if(minDistance > theta && m < q){ // create a new
   cluster
   clusters[m].add(it);
   clusterAgeAvg[m] = (double)ageList.get(it);
   clusterBmiAvg[m] = (double)bmiList.get(it);
   clusterPregnantAvg[m] =
      (double)pregnantList.get(it);
   clusterGlucoseAvg[m] = (double)glucoseList.get(it);
   clusterPressureAvg[m] =
      (double)pressureList.get(it);
   if(it == testCase){
      testCaseCluster = m;
   }
   m++;
}
else{
   clusters[minIndex].add(it);
   int clusterSize = clusters[minIndex].size();
   clusterAgeAvg[minIndex] = updateAvg
      (clusterAgeAvg[minIndex], clusterSize,
      (double)ageList.get(it));
   clusterBmiAvg[minIndex] = updateAvg
      (clusterBmiAvg[minIndex], clusterSize,
      (double)bmiList.get(it));
   clusterPregnantAvg[minIndex] = updateAvg
```

```
            (clusterPregnantAvg[minIndex], clusterSize,
            (double)pregnantList.get(it));
         clusterGlucoseAvg[minIndex] = updateAvg
            (clusterGlucoseAvg[minIndex], clusterSize,
            (double)glucoseList.get(it));
         clusterPressureAvg[minIndex] = updateAvg
            (clusterPressureAvg[minIndex], clusterSize,
            (double)pressureList.get(it));
         if(it == testCase){
            testCaseCluster = minIndex;
         }
      }
   }

   for(int it = 0; it < m; it++){
      System.out.println(clusters[it]);
   }
   System.out.println(testCaseCluster);

   int diabeticCount = 0;
   int preDiabeticCount = 0;
   int nonDiabeticCount = 0;

   // Metrices for diabetic
   double diabeticAgeSum = 0;
   double diabeticBmiSum = 0;
   double diabeticPregnantSum = 0;
   double diabeticGlucoseSum = 0;
   double diabeticPressureSum = 0;

   double diabeticAgeSqSum = 0;
   double diabeticBmiSqSum = 0;
   double diabeticPregnantSqSum = 0;
   double diabeticGlucoseSqSum = 0;
   double diabeticPressureSqSum = 0;

   // Metrices for non diabetic
   double nonDiabeticAgeSum = 0;
   double nonDiabeticBmiSum = 0;
   double nonDiabeticPregnantSum = 0;
   double nonDiabeticGlucoseSum = 0;
```

```
double nonDiabeticPressureSum = 0;

double nonDiabeticAgeSqSum = 0;
double nonDiabeticBmiSqSum = 0;
double nonDiabeticPregnantSqSum = 0;
double nonDiabeticGlucoseSqSum = 0;
double nonDiabeticPressureSqSum = 0;

for(int it = 0; it < clusters[testCaseCluster].size();
   it++){
   int curIndex =
      Integer.valueOf(clusters[testCaseCluster].get(it).toString());
   if(curIndex == testCase){
      continue;
   }
   double age =
      Double.valueOf(ageList.get(curIndex).toString());
   double bmi =
      Double.valueOf(bmiList.get(curIndex).toString());
   double pregnant =
      Double.valueOf(pregnantList.get(curIndex).toString());
   double glucose =
      Double.valueOf(glucoseList.get(curIndex).toString());
   double pressure =
      Double.valueOf(pressureList.get(curIndex).toString());
   if(classList.get(curIndex).toString().equalsIgnoreCase("0")){
      nonDiabeticCount++;
      nonDiabeticAgeSum += age;
      nonDiabeticAgeSqSum += (age * age);
      nonDiabeticBmiSum += bmi;
      nonDiabeticBmiSqSum += (bmi * bmi);
      nonDiabeticPregnantSum += pregnant;
      nonDiabeticPregnantSqSum += (pregnant * pregnant);
      nonDiabeticGlucoseSum += glucose;
      nonDiabeticGlucoseSqSum += (glucose * glucose);
      nonDiabeticPressureSum += pressure;
      nonDiabeticPressureSqSum += (pressure * pressure);
   }

   else{
      diabeticCount++;
```

```
            diabeticAgeSum += age;
            diabeticAgeSqSum += (age * age);
            diabeticBmiSum += bmi;
            diabeticBmiSqSum += (bmi * bmi);
            diabeticPregnantSum += pregnant;
            diabeticPregnantSqSum += (pregnant * pregnant);
            diabeticGlucoseSum += glucose;
            diabeticGlucoseSqSum += (glucose * glucose);
            diabeticPressureSum += pressure;
            diabeticPressureSqSum += (pressure * pressure);
        }
    }
    double diabeticAgeMean = (diabeticAgeSum /
        diabeticCount);
    double diabeticAgeVariance = (diabeticAgeSqSum /
        diabeticCount) - (diabeticAgeMean);
    double diabeticBmiMean = (diabeticBmiSum /
        diabeticCount);
    double diabeticBmiVariance = (diabeticBmiSqSum /
        diabeticCount) - (diabeticBmiMean);
    double diabeticPregnantMean = (diabeticPregnantSum /
        diabeticCount);
    double diabeticPregnantVariance = (diabeticPregnantSqSum
        / diabeticCount) - (diabeticPregnantMean);
    double diabeticGlucoseMean = (diabeticGlucoseSum /
        diabeticCount);
    double diabeticGlucoseVariance = (diabeticGlucoseSqSum /
        diabeticCount) - (diabeticGlucoseMean);
    double diabeticPressureMean = (diabeticPressureSum /
        diabeticCount);
    double diabeticPressureVariance = (diabeticPressureSqSum
        / diabeticCount) - (diabeticPressureMean);

    double nonDiabeticAgeMean = (nonDiabeticAgeSum /
        nonDiabeticCount);
    double nonDiabeticAgeVariance = (nonDiabeticAgeSqSum /
        nonDiabeticCount) - (nonDiabeticAgeMean);
    double nonDiabeticBmiMean = (nonDiabeticBmiSum /
        nonDiabeticCount);
    double nonDiabeticBmiVariance = (nonDiabeticBmiSqSum /
        nonDiabeticCount) - (nonDiabeticBmiMean);
```

```
double nonDiabeticPregnantMean = (nonDiabeticPregnantSum
    / nonDiabeticCount);
double nonDiabeticPregnantVariance =
    (nonDiabeticPregnantSqSum / nonDiabeticCount) -
    (nonDiabeticPregnantMean);
double nonDiabeticGlucoseMean = (nonDiabeticGlucoseSum /
    nonDiabeticCount);
double nonDiabeticGlucoseVariance =
    (nonDiabeticGlucoseSqSum / nonDiabeticCount) -
    (nonDiabeticGlucoseMean);
double nonDiabeticPressureMean = (nonDiabeticPressureSum
    / nonDiabeticCount);
double nonDiabeticPressureVariance =
    (nonDiabeticPressureSqSum / nonDiabeticCount) -
    (nonDiabeticPressureMean);

System.out.println("Diabetic: " + diabeticCount + ",
    Pre: " + preDiabeticCount + ", Non: " +
    nonDiabeticCount);
System.out.println("Feature\tDiabetic mean\tDiabetic
    variance\tPre mean\tPre variance\tNon mean\tNon
    variance");
System.out.println("Age\t" + diabeticAgeMean + "\t" +
    diabeticAgeVariance + "\t" + nonDiabeticAgeMean +
    "\t" + nonDiabeticAgeVariance);
System.out.println("Bmi\t" + diabeticBmiMean + "\t" +
    diabeticBmiVariance + "\t" + nonDiabeticBmiMean +
    "\t" + nonDiabeticBmiVariance);
System.out.println("Pregnant\t" + diabeticPregnantMean +
    "\t" + diabeticPregnantVariance + "\t" +
    nonDiabeticPregnantMean + "\t" +
    nonDiabeticPregnantVariance);
System.out.println("Glucose\t" + diabeticGlucoseMean +
    "\t" + diabeticGlucoseVariance + "\t" +
    nonDiabeticGlucoseMean + "\t" +
    nonDiabeticGlucoseVariance);
System.out.println("Pressure\t" + diabeticPressureMean +
    "\t" + diabeticPressureVariance + "\t" +
    nonDiabeticPressureMean + "\t" +
    nonDiabeticPressureVariance);
```

```
int totalCountInCluster = diabeticCount +
   preDiabeticCount + nonDiabeticCount;

double diabeticPrior = (double) diabeticCount / (double)
   totalCountInCluster;
double preDiabeticPrior = (double) preDiabeticCount /
   (double) totalCountInCluster;
double nonDiabeticPrior = (double) nonDiabeticCount /
   (double) totalCountInCluster;

double testAge = (double) ageList.get(testCase);
double testBmi = (double) bmiList.get(testCase);
double testPregnant = (double)
   pregnantList.get(testCase);
double testGlucose = (double) glucoseList.get(testCase);
double testPressure = (double)
   pressureList.get(testCase);
String testClass = classList.get(testCase).toString();
double error = 0;

System.out.println("Test #" + (testCase + 1) + ":");
System.out.println();
System.out.println("Age: " + testAge + ", Bmi: " +
   testBmi + ", Pregnant: " + testPregnant + ", Glucose:
   " + testGlucose + ", Pressure: " + testPressure );

// Test for diabetic
double diabeticAgeConditional =
   calculateConditionalProbability(testAge,
   diabeticAgeMean, diabeticAgeVariance);
double diabeticBmiConditional =
   calculateConditionalProbability(testBmi,
   diabeticBmiMean, diabeticBmiVariance);
double diabeticPregnantConditional =
   calculateConditionalProbability(testPregnant,
   diabeticPregnantMean, diabeticPregnantVariance);
double diabeticGlucoseConditional =
   calculateConditionalProbability(testGlucose,
   diabeticGlucoseMean, diabeticGlucoseVariance);
double diabeticPressureConditional =
   calculateConditionalProbability(testPressure,
```

```
        diabeticPressureMean, diabeticPressureVariance);
    double diabeticPosteriorNumerator = diabeticPrior *
        diabeticAgeConditional * diabeticBmiConditional *
        diabeticPregnantConditional *
        diabeticGlucoseConditional *
        diabeticPressureConditional;
    System.out.println("Diabetic: " + diabeticPrior + " " +
        diabeticAgeConditional + " " + diabeticBmiConditional
        + " " + diabeticPregnantConditional + " " +
        diabeticGlucoseConditional + " " +
        diabeticPressureConditional);

    // Test for non diabetic

    double nonDiabeticAgeConditional =
        calculateConditionalProbability(testAge,
        nonDiabeticAgeMean, nonDiabeticAgeVariance);
    double nonDiabeticBmiConditional =
        calculateConditionalProbability(testBmi,
        nonDiabeticBmiMean, nonDiabeticBmiVariance);
    double nonDiabeticPregnantConditional =
        calculateConditionalProbability(testPregnant,
        nonDiabeticPregnantMean, nonDiabeticPregnantVariance);
    double nonDiabeticGlucoseConditional =
        calculateConditionalProbability(testGlucose,
        nonDiabeticGlucoseMean, nonDiabeticGlucoseVariance);
    double nonDiabeticPressureConditional =
        calculateConditionalProbability(testPressure,
        nonDiabeticPressureMean, nonDiabeticPressureVariance);
    double nonDiabeticPosteriorNumerator = nonDiabeticPrior
        * nonDiabeticAgeConditional *
        nonDiabeticBmiConditional *
        nonDiabeticPregnantConditional *
        nonDiabeticGlucoseConditional *
        nonDiabeticPressureConditional;
    System.out.println("non diabetic: " + nonDiabeticPrior +
        " " + nonDiabeticAgeConditional + " " +
        nonDiabeticBmiConditional + " " +
        nonDiabeticPregnantConditional + " " +
        nonDiabeticGlucoseConditional + " " +
        nonDiabeticPressureConditional);
```

```
            double probabilitySum = diabeticPosteriorNumerator +
               nonDiabeticPosteriorNumerator;
            System.out.println("P(diabetic): " +
               diabeticPosteriorNumerator / probabilitySum);
            System.out.println("P(non diabetic): " +
               nonDiabeticPosteriorNumerator / probabilitySum);

            System.out.println("Diagnosis (real) result: " +
               testClass);
            String testResult;
            if(nonDiabeticPosteriorNumerator >=
               diabeticPosteriorNumerator){
               testResult = "0";
               error = (diabeticPosteriorNumerator / probabilitySum);
            }

            else{
               testResult = "1";
               error = (nonDiabeticPosteriorNumerator /
                  probabilitySum);
            }
            errorSum += error;

            System.out.println("Test Result: " + testResult);
            if(testResult.equalsIgnoreCase(testClass)){
               System.out.println("Prediction: Correct!");
               correctCount++;
            }
            else{
               System.out.println("Prediction: Incorrect!");
            }
            System.out.println();
            System.out.println();
         }
         System.out.println("Total test case: " + totalCount);
         System.out.println("Correct assumptions: " + correctCount);
         System.out.println("Error: " + (errorSum / totalCount));

      }
   }
```