CLASSIFICATION OF BANK DEPOSITOR USING ID3 AND NAIVE BAYESIAN CLASSIFIERS

MOE SAN PHYU

M.C.Sc. **SEPTEMBER 2022**

CLASSIFICATION OF BANK DEPOSITOR USING ID3 AND NAIVE BAYESIAN CLASSIFIERS

By

MOE SAN PHYU

B.C.Sc.

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Computer Science (M.C.Sc.)

University of Computer Studies, Yangon

SEPTEMBER, 2022

ACKNOWLEDGEMENTS

First of all, I would like to express my inmost gratitude to **Prof. Dr. Mie Mie Khin**, Rector, the University of Computer Studies, Yangon, for allowing me to develop this research and giving me excellent guidance during the period of my dissertation.

Secondly, I would like to express my deepest gratitude to my supervisor **Dr. Zaw Tun**, Prorector, University of Computer Studies (Sittway), for his caring and encouragement, and providing me with excellent ideas and guidance during the time of writing this dissertation.

Thirdly, I would like to express very special thanks to **Dr. Si Si Mar Win**, **Dr**. **Tin Zar Thaw,** Professors, Faculty of Computer Science, the University of Computer Studies, Yangon, as Dean of Master's Course, for giving me the valuable guidance and suggestions during the development of this thesis.

In addition, I would like to acknowledge and special thanks to **Daw Mya Hnin Mon, Associate Professor,** Department of English, University of Computer Studies, Yangon. I would like to thank her for valuable supports and revising my dissertation from the language point of view.

Finally, I would like to extend my thanks to all my teachers who taught me throughout the master's degree course and my friends for their cooperation. The completion of my dissertation would not have been possible without the support and nurturing of my family.

STATEMENT OF ORIGINALITY

I hereby certify that the work embodied in this thesis is the result of original study and has not been submitted for a higher degree to any other University or Institution.

---------- ----------

Date Moe San Phyu

ABSTRACT

Nowadays, banks are financial institutions whose activities are to collect funds from the public in the form of deposits (saving deposit and time deposit). Deposits are an alternative for customers because the interest offered on deposits is higher than regular savings. So, this system is proposed as the bank depositor classification system by using data mining (DM) methods. Among many DM methods, this system uses the ID3 and Naive Bayesian classifiers to classify bank customer's data. This system predicts which customer will subscribe to a long-term deposit proposed by a bank. Moreover, this system analyses the sensitivity, specificity and accuracy of ID3 and Naive Bayesian classifiers. This system can help the bank for identifying customers who will potentially open a time deposit so that it can be used to assist the performance and operations of the bank.

Keywords: Bank Deposit, Classification, NB Classifier, ID3 Classifier.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

1.1 Introduction of the System

Recently, corporate organizations and the banking industry have been affected by the global economic depression. Banks suffer high attrition as a result of this economic situation, and it becomes impossible to keep customers. As a result, marketing managers must intensify their marketing campaigns, while businesses avoid both costs and growth. Business intelligence is a relatively new concept that refers to the use of intelligent mechanisms and the information space to support managerial decisions. Data mining techniques are utilized extensively in data analysis, data summarization, hidden pattern finding, and data interpretation.

The most effective intelligent mechanisms that can manage such a massive expansion of data set and information are data mining approaches because corporate organizations, including the banking sectors, produce tons of records and transactions every day (DM). Data mining is known as the process of observing fresh and novel information from enormous amounts of data sets by identifying hidden and unrecognized relationships between features that are included in the data records, identifying intriguing occurrences and hidden patterns, summarizing the information space to extract predictive decision rules, differentiating the information space into set of objects, and minimizing the features that describe the information space.

Therefore, by avoiding risky transactions that result in bank attrition and boosting customer retention incentives to increase bank profits, DM can be utilized to help decision-makers in the banking sector tackle the economic presumption. One of the economic improvements in the world can be visible through the emergence of financial institutions, particularly in the banking area. Banking is everything related to banks, institutions, and business activities. Banking is the core of every country's financial system. Banks are places for companies, government and private agencies as well as individuals to store their finances. The goal of banking is to carry out national development in order to boost equity, economic growth and national stability in order to improve overall welfare.

Banks are financial institutions whose fundamental activities are gathering funds from people (funding) and directing these funds back to the community

(lending) and giving other bank services. Generally, banks benefit from customers that can be utilized as a source of funds in the form of checking accounts, savings and time deposits. Furthermore, the form of source of funds that became one of the bank's backbones is deposits. Customers have an alternative in deposits since they offer better interest rates than traditional savings accounts. Deposit itself is a place for customers to make transactions in the form of securities. Time deposits can be an alternative for customer because time deposits have a period of time, but a consideration for customers to choose deposits is interested because the interest offered on deposits is higher than ordinary saving. For the above reasons, this system is proposed as the bank depositor classification system.

1.2 Motivation of the System

In managing customer data, obviously, the amount of information is extremely huge. In this way, a bank customer data classification system is required that can be classified between customers who have the opportunity to open a deposit or not. So, this system is motivated as the bank customer data classification system that helps the operation of the bank.

For bank deposit classification, this system utilizes the ID3 decision tree and Naive Bayesian classifiers. Then, at that point, this system shows the effectiveness of ID3 and Naive Bayesian classifiers. By analyzing the sensitivity, specificity and accuracy, this system compares these two classifiers to know which classifier is more precise than other.

1.3 Objectives of the System

Objectives of the system are as follows:

- To assist the operation of the bank,
- To classify between customers who have the opportunity to open a deposit or who do not have the opportunity to open a deposit,
- To classify bank customer’s data by using data mining methods,
- To show the effectiveness of decision tree (ID3) and Naive Bayesian classifiers and
- To analyze and compare the sensitivity, specificity and accuracy as the performance of these two algorithms.

1.4 Organization of the Thesis

This thesis is organized as five chapters, abstract, acknowledgement and references. They are as follows:

In chapter one, bank deposit classification system is introduced. This chapter also described objectives of the thesis and organization of the thesis.

In chapter two, the fundamental of Data Mining and classification methods are presented.

In chapter three, Naive Bayesian classifier and ID3 decision tree classifier are discussed in detail. Then, NB classification algorithm and ID3 classification algorithm are described in this chapter.

In chapter four, the system design, by using use case diagram of the system, explanation of the system, implementation of the system and experimental results are expressed.

In chapter five, the conclusion of the thesis work is presented. In addition, further extensions of the system are depicted.

CHAPTER 2
BACKGROUND THEORY

2.1 Data Mining

An interdisciplinary area of computer science is data mining. It is a computational process that combines techniques from artificial intelligence, machine learning, statistics, and database systems to find patterns in massive data sets. The main objective of the data mining process is to take information from a data set and organize it so that it may be used later [1].

Process of evaluating data from various angles and distilling it into valuable information, information that can be utilized to enhance income, minimize costs, or both, is known as data mining (also known as data or knowledge discovery). One of the many analytical techniques for examining data is data mining software. Users are able to classify the data, summarize the relationships found, and evaluate it from a variety of various dimensions or angles. In big relational databases, data mining is the process of looking for correlations or patterns among numerous fields [3].

Tasks involving data mining are numerous. Among the more popular ones are association rule mining, sequential pattern mining, supervised learning (also known as classification), and unsupervised learning (also known as clustering). Data analysts (data miners), who first study the application domain, choose the most appropriate data sources and the goal data, to begin a data mining application. Data mining, which typically involves three primary phases, can be done with the data [1]:

- **Pre-processing:** For a variety of reasons, the raw data is typically not suited for mining. To get rid of noises or anomalies, it might need to be cleaned. In this case, data reduction by sampling and attribute selection is necessary. The data may also be overly vast or contain a high number of useless qualities.
- **Data mining:** After being cleaned up, the data is fed into a data mining algorithm, which produces patterns or knowledge.
- **Post-processing:** Not all patterns that are discovered are beneficial in various applications. This stage selects the ones that are applicable for applications. To get at the decision, many evaluation and visualization techniques are employed.

Almost always, the entire procedure—also known as the data mining procedure—is iterative. Typically, it takes several iterations to arrive at the desired results, which are subsequently incorporated into actual operational duties.

Descriptive and predictive data mining are two categories into which data mining tasks can be divided. Without having a preconceived notion, descriptive data mining provides information to comprehend what is happening inside the data. With predictive data mining, users can submit records with blank fields, and the system will fill them in based on past patterns it has identified in the database.

According to the functions they carry out, data mining models can be divided into three categories: association rules, clustering, and classification and prediction. Clustering and association rules are descriptive models, whereas classification and prediction is a predictive model [10].

In data mining, classification is the most frequent action. It detects patterns that characterize the family to which an object belongs. It accomplishes this by looking at already classified items and drawing conclusions about a set of rules. Clustering is akin to classification. The main distinction is that no predefined groups have been created. The act of making a prediction is the creation and use of a model to determine the class of an unlabeled item or to determine the value or ranges of values that a specific object is likely to have. The next application is is predicting. Since this guesses the future value of continuous variables based on patterns in the data, it differs from predictions.

2.2 Classification

In order to utilize the model to predict the class of objects whose class label is unknown, classification is the process of finding a model which explains and differentiates data classes of concepts. It is a method for data mining that forecasts the membership of groups for data instances. There are numerous classification methods. These include the decision tree, the Bayes theorem, the k-nearest neighbor classifier, the case-based reasoning approach, genetic algorithms, the association classifier, fuzzy logic, and others.

In classification, a specific output is predicted based on an input. The algorithm analyzes a training set made up of a number of attributes and the desired output, also known as the goal or prediction attribute, in order to predict the outcome.

The program looks for connections between the attributes that could be used to forecast the result. The algorithm is then given a new data set, known as the prediction set, which has the identical set of attributes as the previous one with the exception of the prediction attribute. After analyzing the input, the algorithm generates a prediction. The algorithm's performance is determined by prediction accuracy. Among the various forms of knowledge expression found in the literature, categorization often expresses knowledge using prediction rules [12].

2.3 Classification Methods

The two categories of classification techniques are supervised and unsupervised techniques. Bayesian classifier, Decision Tree, Artificial Neural Network, and Support Vector Machine are examples of supervised approaches, whereas clustering techniques like a K-means variant are examples of unsupervised methods.

Supervised learning is used in most practical machine learning applications. The mapping function from the input to the output, such as Y = f, is learned using an algorithm in supervised learning, which employs the input variables (x) and an output variable (Y) (x). The objective is to estimate the mapping function as closely as possible so that you can forecast the output variables (Y) for new input data (x).

Because an algorithm learning from the training dataset can be compared to a teacher supervising the learning process, it is known as supervised learning. The algorithm iteratively makes predictions on the training data and is corrected by the teacher thus it is possible to know the right responses. When the algorithm reaches a certain threshold of difficulties, learning stops. Unsupervised learning using the input data (x) but without any associated output variables Unsupervised learning aims to learn more about the data by simulating its underlying structure or distribution. These are referred to as unsupervised learning because, in contrast to the supervised learning described above, there is no right answer and no teacher. Algorithms are allowed to use their own initiative to discover and present the interesting structure in the data.

2.3.1 Bayesian Classifier

Statistical classifiers include Bayesian classifiers. They are able to forecast probabilities of class membership, such as the likelihood that a given tuple belongs to

a specific class. The Bayes theorem is the foundation of Bayesian classification. A straightforward Bayesian classifier known as the naive Bayesian classifier, which is used in studies comparing classification algorithms, has been found to perform on par with decision trees and some neural network classifiers. When used on sizable databases, Bayesian classifiers have also demonstrated great accuracy and speed. The assumption made by naive Bayesian classifiers is that the impact of one attribute's value on a particular class depends only on that attribute's value. The term "class conditional independence" refers to this presumption. It is "naive" in this sense because it is designed to make the calculations involved simpler [11].

Theoretically, compared to all other classifiers, Bayesian classifiers have the lowest error rate. However, due to inaccurate underlying assumptions, such as class conditional independence, and a dearth of readily available probability data, this is not often the case in practice [11].

2.3.2 Decision Tree Induction

The process of learning decision trees from class-labeled training tuples is known as decision tree induction. In a decision tree, each internal node (non-leaf node) symbolizes a test on an attribute, each branch shows the test's result, and each leaf node (or terminal node) stores a class label. The root node is the node at the top of a tree. A typical decision tree is shown in Figure 2.1.

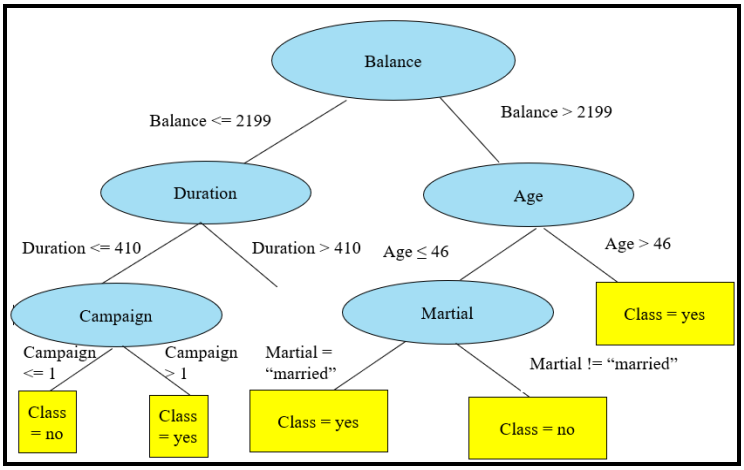

Figure 2.1. Typical Decision Tree

In other words, it forecasts whether a user at bank is likely to deposit. It reflects the concept deposit. Leaf nodes are represented by ovals, while internal nodes

are represented by rectangles. While certain decision tree algorithms can only build nonbinary trees (with each internal node branching to exactly two additional nodes), others can.

2.3.3 Artificial Neural Network

A mathematical model or computational model based on biological neural networks, or a "emulation of biological brain system," is known as an artificial neural network (ANN), which is frequently referred to as a "neural network" (NN). It uses a connectionist method of computation to handle information and is made up of a network of artificial neurons. An ANN is often an adaptive system that modifies its structure in response to information flowing through the network during the learning phase, whether that information is internal or external [10]. The process of artificial neural network is shown in Figure 2.2.

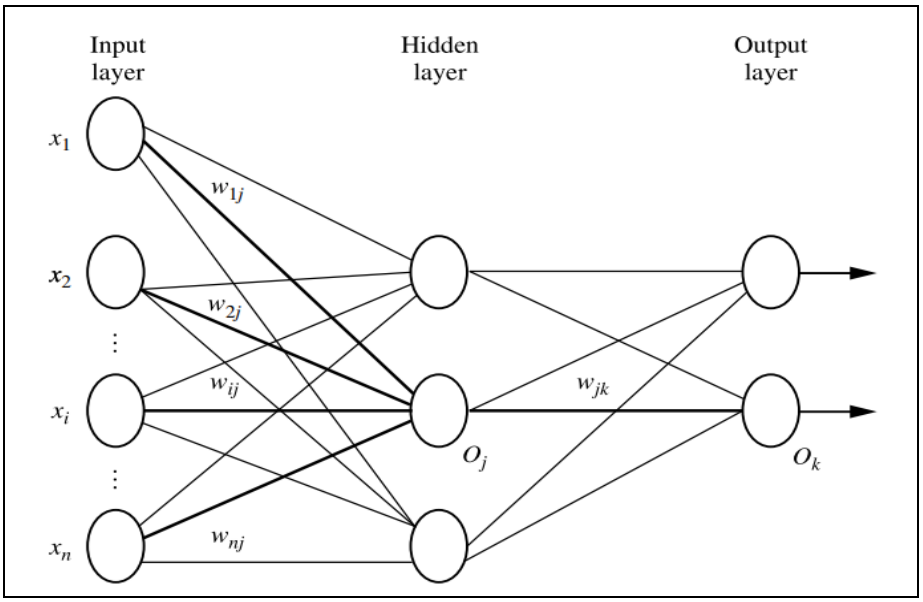

Figure 2.2. Artificial Neural Network

The discovery that human brains' intricate learning mechanisms consisted of networks of densely coupled neurons served as the basis for neural networks. Although a given neuron may have a very straightforward structure, dense networks of interconnected neurons are capable of carrying out challenging learning tasks like classification and pattern recognition [2]. A family of highly parameterized statistical models that have received a lot of interest recently includes artificial neural networks

(ANNs). ANNs are highly parameterized, which makes them quite flexible and allows them to accurately simulate relatively minor function abnormalities [3].

2.3.4 Support Vector Machine

Support vector machines (SVM) in machine learning are supervised learning models with related learning algorithms that examine data used for regression and classification analyses. An SVM training algorithm creates a model that categorizes fresh instances into one of two categories given a series of training examples, converting it into a non-probabilistic binary linear classifier. An SVM model is a representation of the instances as points in space, mapped in a way that creates as much of a clear distance between the examples of the various categories as feasible.

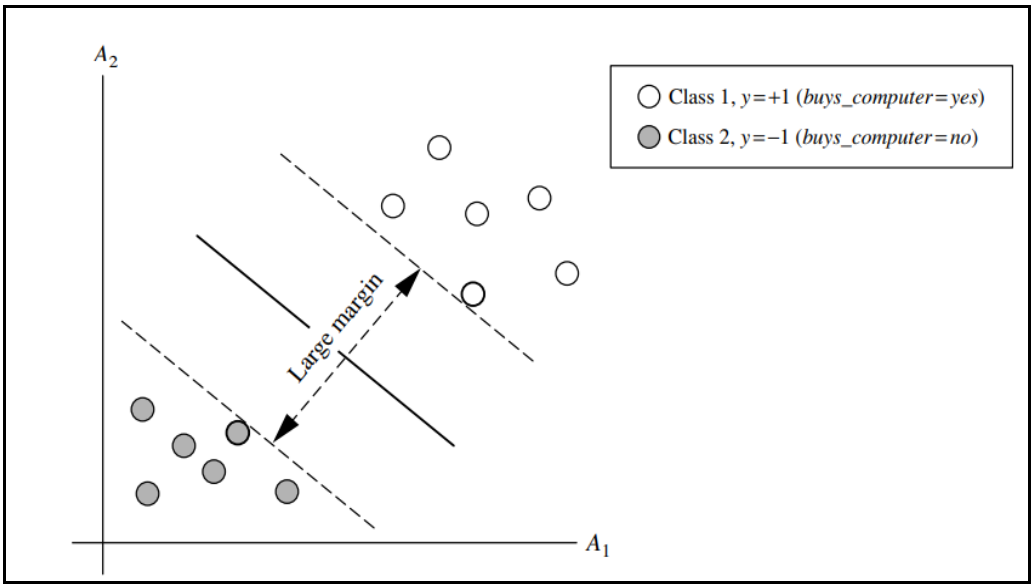

Figure 2.3. Support Vector Machine (SVM)

In Figure 2.3, a support vector machine (SVM) is displayed. In addition to conducting linear classification, SVMs can be effectively carry out non-linear classification by implicitly mapping their inputs into high-dimensional feature spaces. This technique is known as the kernel trick.

2.3.5 Clustering

Clustering analyzes data items without referencing a known class label, unlike classification and prediction, which do so. Because they are unknown to begin with, the class labels are typically not included in the training data. Such labels can be

created using clustering. Based on the maximization of intra-class similarity and minimization of interclass similarity, the items are clustered or classified. In other words, object clusters are created so that objects inside a cluster have a high degree of similarity to one another but have a low degree of similarity to objects in other clusters. Every created cluster can be thought of as a class of objects from which rules can be deduced [2]. Clustering is shown in Figure 2.4.

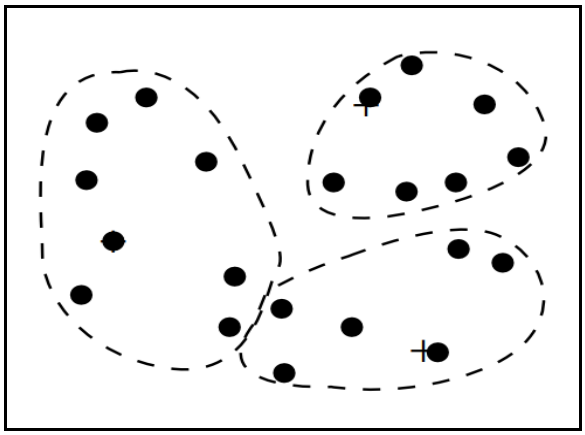

Figure 2.4. Clustering

Clustering is frequently carried out as the first stage in data mining, and the generated clusters are then used as additional inputs into a different technique, such neural networks, further down the line. It is frequently beneficial to use clustering analysis first to decrease the search space for the downstream algorithms because many modern databases are very large [8, 11].

2.3.6 Association Rule

The task of determining which attributes "go together" is known as the association task in data mining. The task of association, also known as affinity analysis or market basket analysis, is most common in the business world and aims to identify guidelines for quantifying the relationship between two or more features. If antecedent, then consequent is how association rules are expressed, along with a measure of the rule's support and confidence.

The underlying statistics and pattern structure serve as the foundation for the confidence and support. Rule support, which measures the proportion of transactions from a transaction database that the provided rule fulfills, is an objective metric for association rules of the form XY [11]. Which is the probability P(X U Y) where X U

Y denotes the union of the item sets X and Y, or the fact that a transaction contains both X and Y. Confidence, which measures the degree of certainty of the identified relationship, is another objective metric for association rules. This is considered to represent the conditional probability (X | Y), or the likelihood that an X and Y transaction will occur. More formally, support and confidence are defined as:

$$Support(X \rightarrow Y) = \mathrm{P}(X \cup Y)$$

$$Confidence(X \rightarrow Y) = P(X|Y)$$

Association varies from classification in two ways: they can "predict" the value of more than one attribute at once, and they can "predict" any attribute, not only the class. Due of this, association rules are much more prevalent than categorization rules, and the issue is to avoid becoming overrun by them [5].

2.4Related Work

In 2021, M. H. Effendy and D. Anggraeni used Naive Bayes classifier (NBC) and K-Nearest Neighbor (KNN) classifier to classify the bank customer data. This system used the bank customer data consisting of 4521 records and 17 variables. The data is divided into 3 types of training-testing processes, namely 70%:30%, 75%:25% and 80%:20%, and the K-fold cross validation method is used with a value of K-10. The results of this study indicate that the KNN method is better than the NBC method. This system helps the bank in identifying customers who will potentially open a time deposit so that it can be used to assist the performance and operations of the bank [15].

In 2021, F. Safarkhani and S. Moro used the J48 decision tree classifier to predict whether a customer will subscribe to a long-term deposit or not. The combination of resampling and feature selection has been applied to 10% of the whole dataset (41188), resulting in 4119 instances from the UCI repository of a Portuguese bank. To evaluate the performance of the classifier, three metrics, specificity, sensitivity and accuracy were calculated. The main goal of this system was to understand the effectiveness of the J48 model at predicting the success of telemarketing calls for selling bank long-term deposits [16].

S. Abbas used genuine marketing data from a Portuguese marketing campaign relating to bank deposit subscription in 2015, as well as rough set theory and decision tree mining techniques. By minimizing the number of variables that define the dataset

and focusing on the most important ones, and by predicting the deposit customer retention criteria based on potential prediction rules, the article intends to increase the effectiveness of marketing campaigns and assist decision makers [17].

CHAPTER 3
NAIVE BAYESIAN AND DECISION TREE CLASSIFIER

A data mining function called classification places objects in a collection into specific groups or classes. To correctly forecast the target class for each case in the data is the aim of classification. A data set with established class assignments serves as the starting point for a classification effort. By comparing the predicted values to the predetermined target values in a collection of test data, classification models are put to the test. The major activities of classification involve data cleaning, relevance analysis, data transformation and reduction such as normalization and generalization. The criteria for classification and prediction are accuracy, speed, robustness, scalability and interpretability.

3.1. Naive Bayesian

Naive Bayes classifiers are a family of straightforward probabilistic classifiers used in machine learning. They are based on the application of Bayes' theorem with strong (naive) independence assumptions across the features.

Since the 1950s, there has been a lot of research done on Naive Bayes. It was first presented to the text retrieval community in the early 1960s under a different name, and it is still a well-liked (baseline) method for text categorization, the challenge of classifying documents as legitimate or spam, sports or politics, etc. using word frequencies as the characteristics. It can compete in this field with more sophisticated techniques like support vector machines with the right preprocessing. It can also be used for automated medical diagnosis.

The number of parameters required for naive Bayes classifiers is linear in the number of variables (features/predictors) in a learning problem, making them extremely scalable. Instead, then using an expensive iterative approximation, which is how many other types of classifiers are trained, maximum-likelihood training can be accomplished by evaluating a closed-form expression, which requires linear time.

Simple Bayes and independent Bayes are two different names for naive Bayes models. Although naive Bayes is occasionally referred to as a Bayesian classifier, a somewhat sloppy usage that has led some Bayesians to refer to it as the idiot Bayes

model, Russell and Norvig remark that all these designations refer to the employment of Bayes' theorem in the classifier's decision process.

Naive Bayes is a straightforward method for building classifiers. These models assign class labels to problem cases, which are represented as vectors of feature values, and the class labels are chosen from a finite set. For training such classifiers, there isn't just one technique, but rather a family of algorithms built on the premise that, given the class variable, the value of one feature is independent of the value of every other feature. For instance, if a fruit is red, round, and roughly 10 cm in diameter, it may be regarded as an apple. Regardless of any potential interactions, a naive Bayes classifier believes that each of these characteristics separately contributes to the likelihood that this fruit is an apple, regardless of any possible correlations between the color, roundness and diameter features.

Naive Bayes classifiers can be taught very effectively in a supervised learning environment for specific kinds of probability models. It is possible to operate with the naive Bayes model without embracing Bayesian probability or applying any Bayesian techniques because parameter estimation for naive Bayes models frequently employs the maximum likelihood method.

Naïve Bayes classifiers have performed admirably in a variety of challenging real-world circumstances despite their naive design and ostensibly oversimplified assumptions. The apparently implausible performance of naïve Bayes classifiers has solid theoretical justifications, as demonstrated by an examination of the Bayesian classification problem in 2004 [4]. However, a thorough comparison with different classification algorithms revealed that other strategies, like boosted trees or random forests, beat Bayes classification [9]. The fact that naive Bayes only needs a modest amount of training data to estimate the classification-related parameters is a benefit.

The naive Bayesian classifier, or simple Bayesian classifier, works as follows:

- Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, $X=(x_1, x_2,\ldots, x_n)$, depicting n measurements made on the tuple from n attributes, respectively, $A_1, A_2,.., A_n$.
- Suppose that there are m classes, $C_1, C_2,\ldots, C_m$. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior

probability, conditioned on X. That is, the naïve Bayesian classifier predicts that tuple x belongs to the class C_i.

$$P(C_i \backslash X) > P(C_j \backslash X) \text{ for } 1 \leq j \leq m, j \neq i \quad (3.1)$$

Thus we maximize $P(C_i|X)$. The class C_i for which $P(C_i|X)$ is maximized is called the maximum posteriori hypothesis.

$$P(C_i \backslash X) = P(X \backslash C_i)\, P(C_i)/\, P(X) \quad (3.2)$$

- As P(X) is constant for all classes, only $P(X|C_i)\, P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1)=P(C_2)=\ldots=P(C_m)$, and we would therefore maximize $P(X|C_i)$. Otherwise, we maximize $P(X|C_i)P(C_i)$. Note thatthe class prior probabilities may be estimated by $P(C_i)=|C_i,D|/|D|$**,** where $|C_i,D|$ is the number of training tuples of class C_i in D.
- Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X|C_i)$. In order to reduce computation in evaluating $P(X|C_i)$, the naïve assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes).

  $$P(X \backslash C_i) = \prod^{n}_{k=1} P(x_k \backslash C_i) \quad (3.3)$$

  The probabilities $P(x_1|C_i)$, $P(x_2|C_i)$,… ,$P(x_m|C_i)$ from the training tuples.
- In order to predict the class label of X, $P(X|C_i)P(C_i)$ is evaluated for each class C_i. The classifier predicts that the class label of tuple X is the class C_i if and only if

  $$P(X \backslash C_i)\, P(C_i) > P(X \backslash C_j)\, P(C_j) \quad \text{for } 1 \leq j \leq m, j \neq i \quad (3.4)$$

  In other words, the predicted class label is the class C_i for which $P(X|C_i)P(C_i)$ is the maximum.

3.2 Decision Tree

Using a series of straightforward decision rules, the decision tree is a structure that may be used to partition a huge collection of records into progressively smaller groups of records [6]. The individuals in the resulting groups resemble one another more and more with each subsequent division.

At the root node, a record is added to the tree. To identify which child node the record will meet next, the root node performs a test. The initial test is chosen using a variety of algorithms, but the objective is always the same: select the test that discriminates between the target classes the best. Repeating this procedure brings the record to a leaf node. The classification of each record that lands at a specific leaf of the tree is the same. Each leaf follows a different route from the root. That route is an expression of the record classification rule [6].

Even while various leaves might classify the same thing, each leaf classifies something for a different reason. For instance, in a tree that categorizes fruits and vegetables by hue, the leaves for apple, tomato, and cherry may all forecast "red," though with varied degrees of certainty, as there are probably some cases of green apples, yellow tomatoes, and black cherries as well [6].

They stated that decision tree classifier building is suitable for exploratory knowledge discovery because it doesn't call for parameter selection or domain knowledge [11]. High dimensional data can be handled via decision trees. Their use of a tree-like representation of learned information makes it clear and typically simple for people to understand. Decision tree induction's learning and classification phases are easy and quick. For linearly separable issues, decision tree classifiers often have acceptable accuracy. Many application domains, including medical, manufacturing and production, financial analysis, astronomy, and molecular biology, have used decision tree induction methods for categorization [7].

Using various attribute values from the available data, a decision tree is a predictive machine-learning model that chooses the target value (dependent variable) of a new sample. In a decision tree, the internal nodes represent the various qualities, the branches between the nodes describe potential values for these attributes in the observed samples, and the terminal nodes describe the dependent variable's classification at its conclusion.

The attribute that has to be predicted is referred to as the dependent variable because the values of all the other qualities influence or determine its value. The other characteristics in the dataset are referred to as the independent variables since they aid in predicting the value of the dependent variable.

The simple algorithm used by the J48 Decision tree classifier is shown below. It must first build a decision tree based on the attribute values of the available training data in order to categorize a new item. As a result, whenever it comes across a set of

items (a training set), it recognizes the attribute that most clearly distinguishes between the numerous occurrences. The feature with the largest information gain is the one that can tell us the most about the data instances in order to classify them most accurately. Now, out of all the potential values for this characteristic, if there is one that has no ambiguity, meaning that all data instances inside its category have the same value for the target variable, then it terminates that branch and assign to the target value that have obtained.

It then searches for a different attribute with the greatest knowledge gain for the remaining cases. So it keeps doing this until it either decides clearly what set of traits gives us a specific target value or runs out of attributes. If it runs out of attributes or is unable to determine an answer clearly given the information at hand, it assigns this branch a goal value that the majority of the objects falling under this branch should have.

3.2.1 Attribute Selection Measure

At each node in the tree, the test attribute is chosen using the information gain metric. An attribute selection measure or a measure of the goodness of split are two terms used to describe this type of measurement. The test attribute for the current node is the one with the maximum information gain (or entropy reduction). This characteristic shows the least randomness or "impurity" in the generated partitions and reduces the amount of information required to categorize the samples in those partitions. A simple (but not necessarily the simplest) tree will be found using this information-theoretic approach, which also reduces the anticipated number of tests required to categorize an object.

Let S be a set consisting of s data sample. Suppose the class label attribute has m distinct values defining m distinct classes, C_i (for i=1,…,m). Let s_i be the number of samples of S in class C_i. The expected information needed to classify a given sample is given by

$$I(s_1,s_2,\ldots,s_m) = -\sum_{i=1}^{m} P_i Log_2(P_i) \qquad (3.5)$$

where p_i is the probability that an arbitrary sample belongs to class C_i and is estimated by s_i/s. Note that a log function to the base 2 is used since the information is encoded in bits.

Let attribute A have distinct values, $\{a_1,a_2,\ldots,a_v\}$. Attribute A can be used to partition S into v subsets, $\{S_1,S_2,\ldots,S_v\}$, where S_j contains those samples in S that have value a_j of A. If A were selected as the test attribute (i.e the best attribute for splitting), then these subsets would correspond to the branches grown from the node containing the set S. Let S_{ij} be the number of samples of class C_i in as subset S_j. The entropy, or expected information based on the partitioning into subsets by A, is given by

$$E(A) = (\sum_{j=1}^{v}(s_{1j}+\ldots+s_{mj})/s)\, I(s_{1j}+\ldots+s_{mj}) \quad (3.6)$$

The term $(s_{1j},s_{2j},\ldots,s_{mj})$ / s acts as the weight of the j^{th} subset and is the number of samples in the subset (i.e having a_j of A) divided by the total number of samples in S. The smaller the entropy value, the greater the purity of the subset partitions. Note that for a given subset S_j,

$$I(s_{1j},s_{2j},\ldots,s_{mj}) = -\sum_{i=1}^{m} P_{ij}\, Log_2(P_{ij}) \quad (3.7)$$

where $p_{ij} = s_{ij} / |s_i|$ and is the probability that a sample in S_j belongs to class C_i.
The encoding information that would be gained by branching on A is

$$\text{Gain (A)} = I(s_1,s_2,\ldots,s_m) - E(A) \quad (3.8)$$

In other words, Gain (A) is expected reduction in entropy caused by knowing the value of attribute A.

Each attribute's information gain is calculated using the algorithm. For a given set S, the attribute with the greatest information gain is selected as the test attribute. The samples are divided in accordance with the attribute's values, branches are made for each value of the attribute, and a node is produced and labeled with the attribute.

3.2.2 Decision Tree Algorithm

The decision tree algorithm is utilized in machine learning and data mining. Observations about an object are mapped to conclusions about the item's goal value using a decision tree, which is a predictive model. Classification trees or regression trees are more evocative names for such tree models. In these tree-like structures, the leaves stand for classes, and the branches for the qualities that combine to form those classifications.

The training data set is recursively partitioned in decision tree classification up until the records in the sub-partitions are wholly or primarily from the same classes. In ID3, C4.5, and CART, decision trees are built using a top-down, recursive divide-

and-conquer strategy. A training set of tuples and their corresponding class labels serve as the starting point of the majority of decision tree induction algorithms. As the tree is constructed, the training set is recursively divided into smaller subsets. The general decision tree algorithm for the system is as follows:

Algorithm: Generate_decision_tree. Generate a decision tree from the given training data.

Input: The training samples, samples, represented by categorical valued attributes; the set of candidate attributes, attribute list.

Output: A decision tree.

Method:

(1) create a node N;
(2) **if** samples are all of the same class, C **then**
(3) return N as a leaf node labeled with the class C;
(4) **if** *attribute–list* is empty **then**
(5) return N as a leaf node labeled with the most common class in *samples*;//majority voting
(6) Select *test-attribute*, the attribute among attribute-list with the highest information gain;
(7) label node N with *test-attribute*;
(8) **for each** known value a_i of test attribute// partition the samples
(9) grow a branch from a node N for the condition *test attribute* = a_i;
(10) Let s_i be the set of samples in samples for which *test-attribute*=a_i;
(11) **if** s_i is empty **then**
(12) attach a leaf labeled with the most common class in samples;
(13) **else** attach the node returned by Generate_decision_tree (s_i, *attribute-list-test-attribute);*

3.2.3 Extracting Classification Rules from Decision Trees

Rules are a form of information or knowledge representation. A rule-based classifier classifies data using a collection of IF-THEN rules. An IF-THEN rule is an expression of the form:

IF condition, THEN conclusion.

The rules can be easily extracted from the classification tree. Rules are easier for users to understand.

- One rule is created for each path from the root to a leaf.
- Each attribute-value pair along a path forms a conjunction.
- The leaf node holds the class prediction.

3.2.4 From Tree to Rules

A common classification technique is the use of decision trees, which are noted for their accuracy and their ease of understanding. Decision trees can grow vast and be challenging to understand. Learn how to create a rule-based classifier in this section by removing IF-THEN rules from a decision tree. One rule is constructed for each path from the root to a leaf node in order to retrieve rules from a decision tree. The rule (the "IF" section) is formed by logically ANDing each splitting criterion along a certain path. The class prediction that makes up the rule consequent (the "THEN" component) is held by the leaf node.

The decision tree adheres to the attribute selection order that it has determined to be appropriate for the tree. It is able to assign or anticipate the target value for this new instance by comparing all the relevant attributes and their values with those shown in the decision tree model. With the aid of an example, the aforementioned description will be more understandable and transparent. So, let's look at a J48 decision tree categorization example. a tree of choices.

3.3 Holdout Method

Holdout method is used for accuracy estimation. The holdout approach, which is based on randomly sampled partitions of the provided data, is a popular methodology for evaluating classifier accuracy. The given data are randomly divided into two separate sets, a training set and a test set, in the holdout method. Usually, one third of the data is assigned to the test set, while the remaining two thirds are assigned to the training set as shown in Figure 3.1. Holdout method is shown in Figure 3.2.

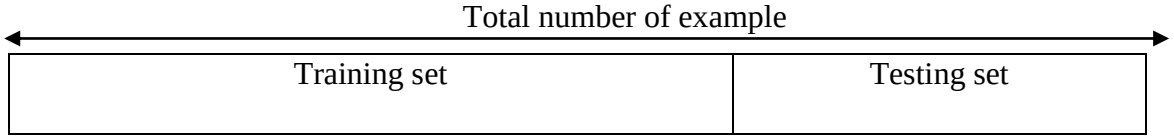

Figure 3.1 Training and Testing Set

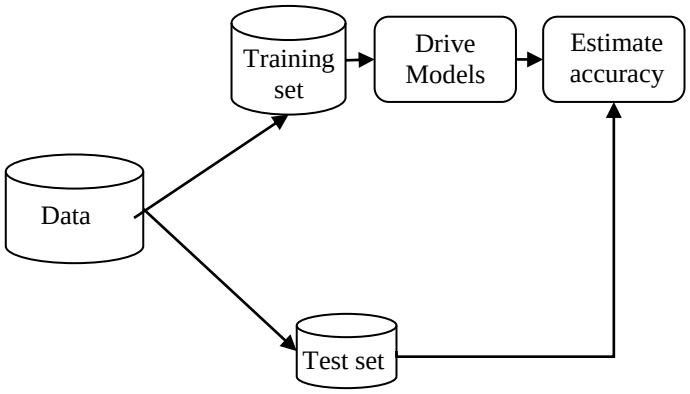

Figure 3.2 Holdout Method

The training set is used to derive the classifier, whose accuracy is estimated with the test set.

3.3.1 Confusion Matrix

A performance indicator for machine learning classification issues when the output can be two or more classes is the confusion matrix. It is a table with combinations of values that were expected and actual. The table that is frequently used to represent how well a classification model performs on a set of test data for which the true values are known is known as a confusion matrix...

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 3.3 Confusion Matrix

Confusion matrix is shown in Figure 3.3. Confusion matrix is useful for measuring sensitivity, specificity and accuracy of the classifier. True positive, true negative, false positive and false negative are as follows:

- True Positive: We predicted positive and it's true.
- True Negative: We predicted negative and it's true.
- False Positive (Type 1 Error): We predicted positive and it's false.
- False Negative (Type 2 Error): We predicted negative and it's false [13].

From the confusion matrix, the sensitivity, specificity and accuracy is evaluated by using the following equation:

$$\text{Sen}sitivity = \frac{TP}{TP + FN} \tag{3.9}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.10}$$

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{3.11}$$

Sensitivity (true positive rate) refers to the probability of a positive test, conditioned on truly being positive. Specificity (true negative rate) refers to the probability of a negative test, conditioned on truly being negative. Accuracy is the proportion of true results, either true positive or true negative [14].

CHAPTER 4
PROPOSED SYSTEM DESIGN

This chapter describes the proposed system design, the system flow diagram and use case diagram about bank depositor classification system. Then, the attributes and its information about bank customer are described. The detail explanation, implementation and experimental results of the system are also described in this chapter.

4.1. Proposed System Design

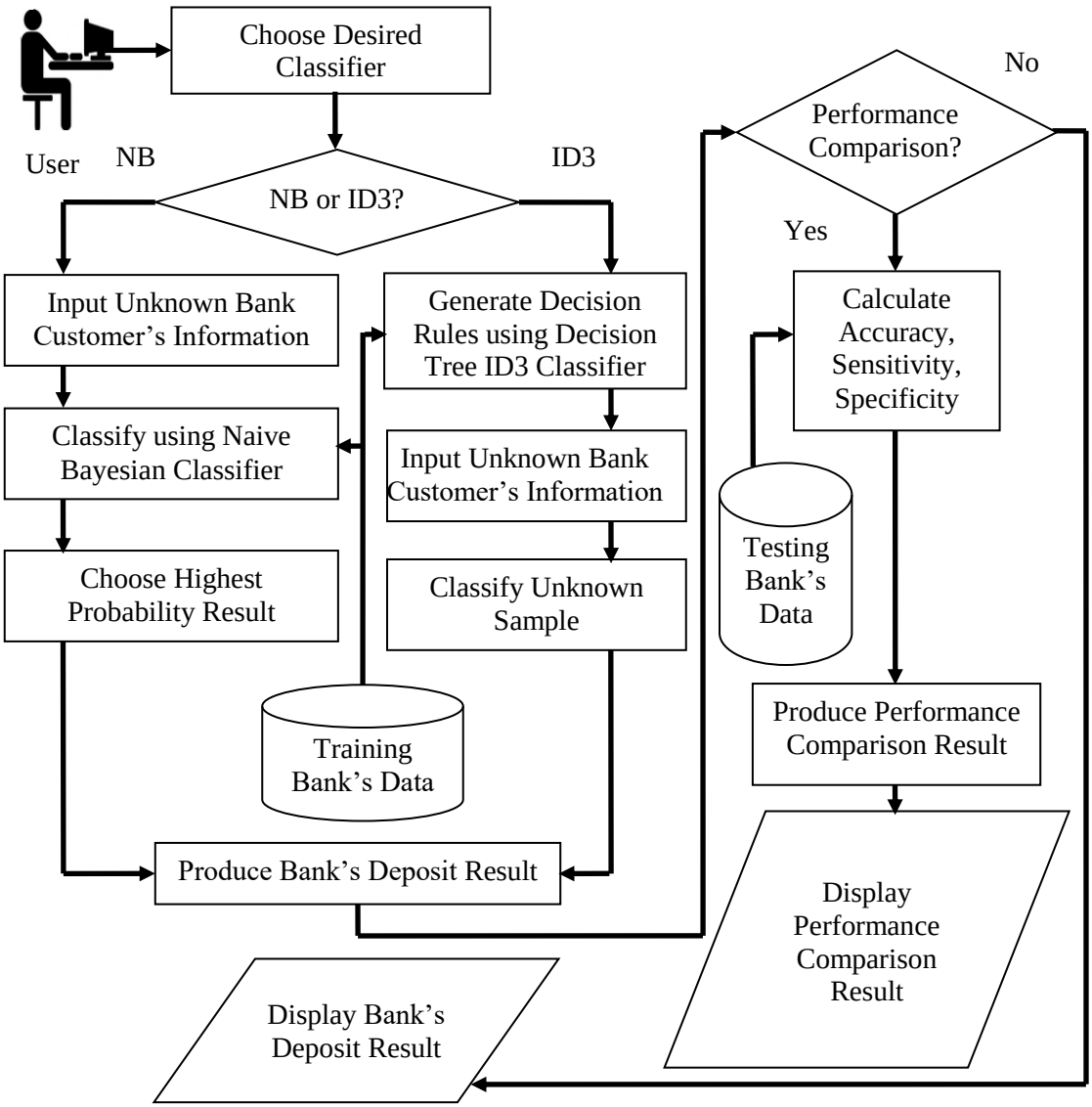

Figure 4.1 Proposed System Design

Proposed system design is shown in Figure 4.1. This system consists of four main processes. The first process is the user chosen desired classifier among ID3 and NB classifiers. The second process is the NB classification process. In this process, this system calculates the probability of each class to choose higher probability bank deposit's result for the unknown sample (user inputted bank's information). In the third process, this system performs the ID3 classification process by generating decision tree and rules. According to the rules, this system produces the bank deposit's result. Finally, this system compares the performance of ID3 and Naive Bayesian classifier to point out which classifier is more precise than another classifier. This system can support the financial domain by classifying bank deposit or not.

4.2. Process Flow Diagram of the System

This system is proposed as the bank depositor classification system by using ID3 (Iterative Dichotomiser) and NB (Naive Bayesian) classifiers. At first of the system, the user can choose the desired classifiers. If the user chooses the ID3 classifier, this system classifies the user inputted unknown sample (unknown bank information) according to the decision rules. For ID3 classification, this system first calculates each information gain for each attribute from the training bank data. If the attribute has highest information gain, this system chooses these attributes as the root node. After choosing root node, this system continues to choose leaf node by calculating the information gain of each attribute. By using root node and leaf nodes, this system generates the decision tree. Then, this system generates the decision rules from the decision tree. By using generated decision rules, this system classifies the user who can be bank's depositor or not.

If the user chooses the NB classifier, the user must first input the unknown sample X (unknown bank information). Because of NB classifier performs the classification process based on unknown sample X. After accepting unknown sample, this system calculates the probability of each class ($P(C_i)$) by using training bank's data. Then, this system also calculates the probability of each attribute of unknown sample X ($P(X\backslash C_i)$). To choose highest probability, this system calculates the multiplication $P(X\backslash C_i)$ and $P(C_i)$. To determine the user who is bank's depositor or not, this system produces the class that has highest probability result. Figure 4.2 shows the process flow diagram of the system.

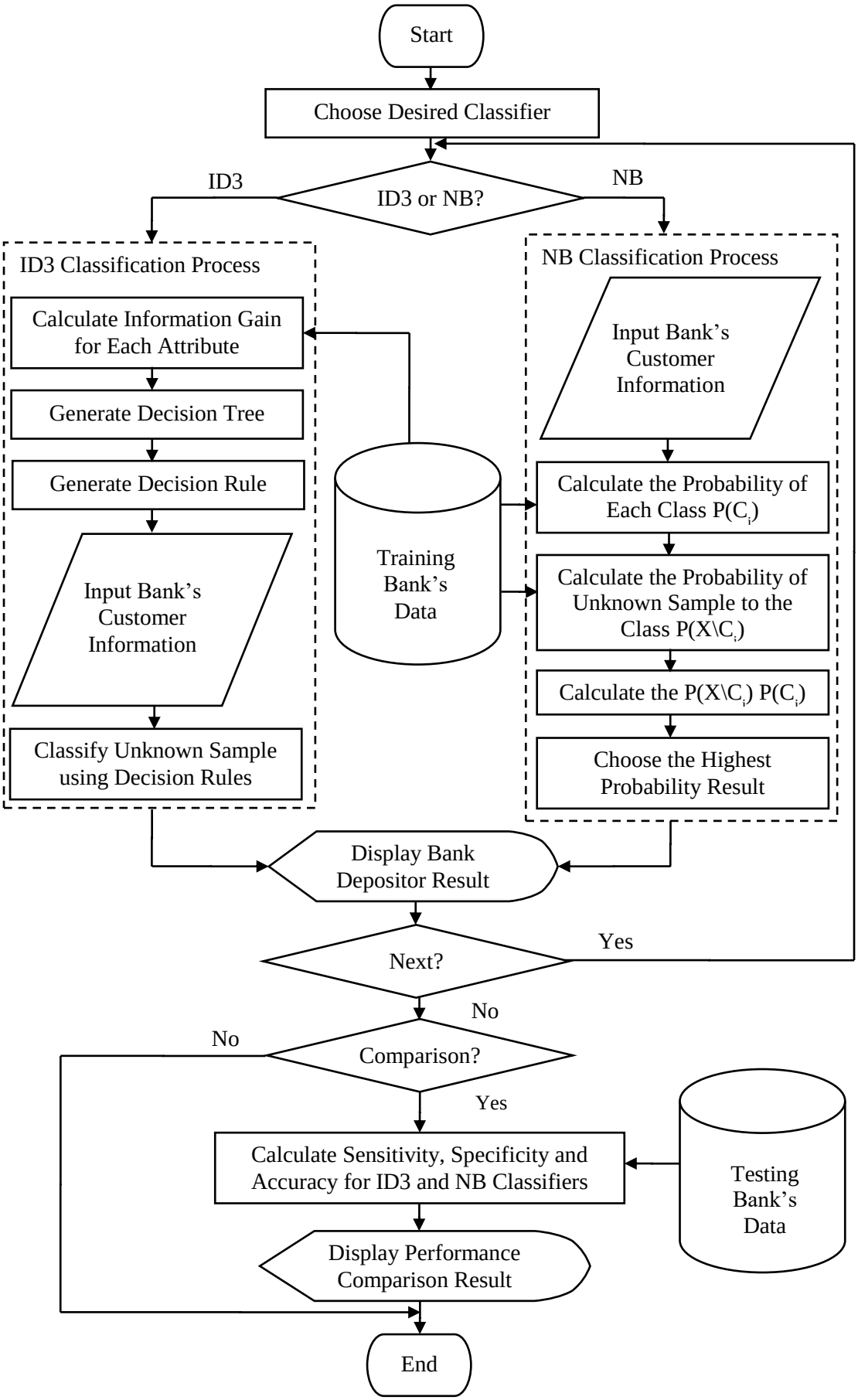

Figure 4.2 Process Flow Diagram of the System

By calculating sensitivity, specificity and accuracy of ID3 and NB classifiers as the performance, this system allows the user to know which classifier is more than another classifier. To measure the performance of each classifier, this system uses the testing data. After measuring the performance of each classifier, this system displays the comparison result to the user.

4.3. Use Case Diagram of the System

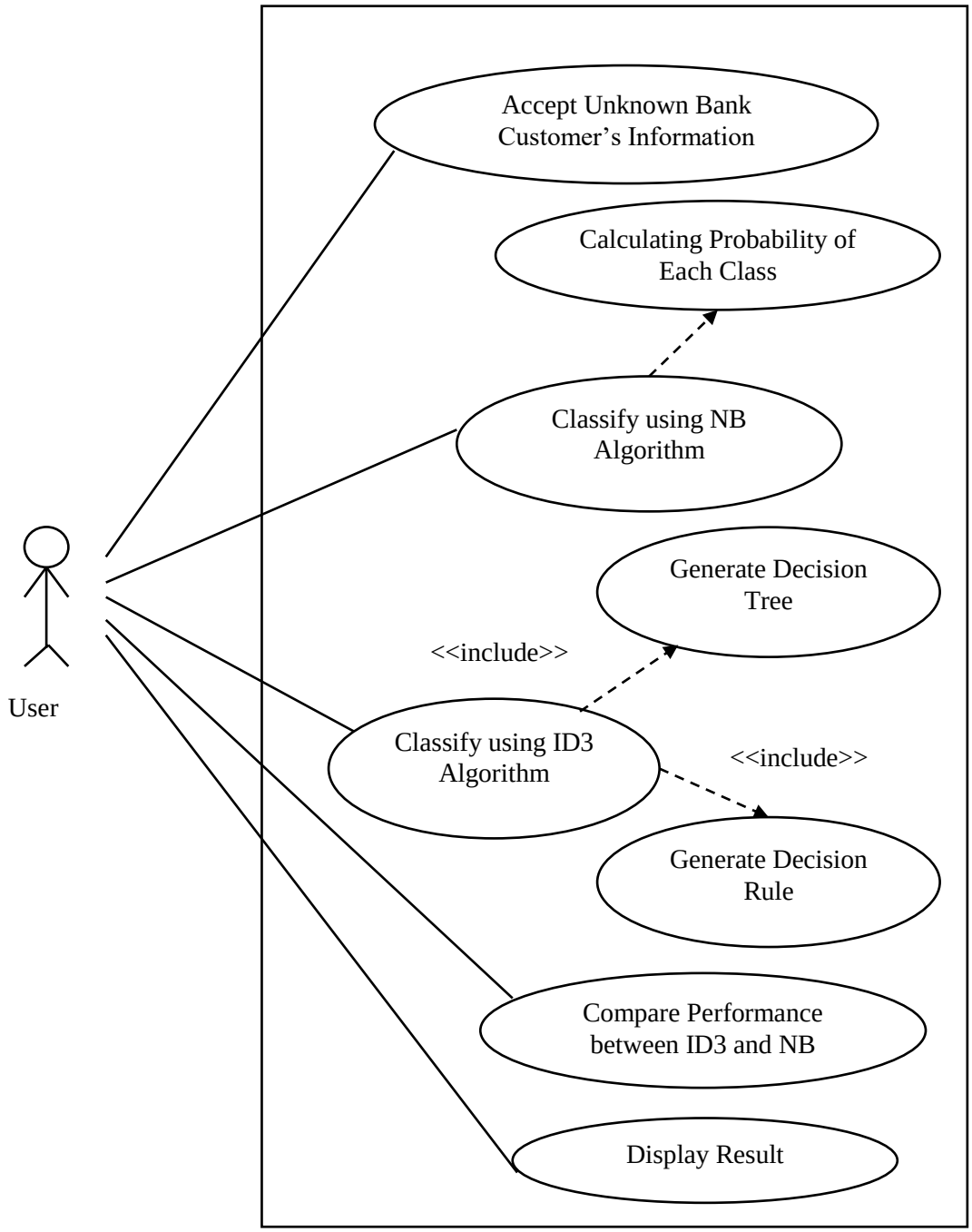

Figure 4.3 Use Case Diagram of the System

Use case diagram of the system is shown in Figure 4.3. In this system, the user can classify the unknown bank's information by using ID3 classifier or NB classifier. If the user chooses the ID3 classifier, it is needed to generate decision tree and rules for bank deposit's classification process. Then, according to the NB classifier, this system calculates the probability of each class to choose the highest probability for unknown sample. Finally, this system compares the performance of ID3 and NB classifier.

4.4. Database and Its Attribute Information

This system extracts the bank customer's dataset from the https://archive.ics.uci.edu/ml/ datasets.php website. The size of this dataset is "4503" KB. Dataset includes the "4516" records, 16 attributes and one class. Bank customer's attribute information are shown in Table 4.1.

Table 4.1. Bank Customer's Attribute and Its Information

ID	**Attribute and Its Information**	
	Attribute	**Information**
1	Age	Age of Bank Customer
2	Job	Type of job (Admin, Unknown, Unemployed, Management, Housemaid, Entrepreneur, Student, Blue-collar, Self-employed, Retired, Technician, Services)
3	Marital	Marital status (Married, Divorced, Single)
4	Education	Unknown, Secondary, Primary, Tertiary
5	Default	Has credit in default? (Yes, No)
6	Balance	Average yearly balance in euros
7	Housing	Has housing loan? (Yes, No)
8	Loan	Has personal loan? (Yes, No)
9	Contact	Contact communication type (Unknown, Telephone, Cellular)

10	Day	Last contact day of the month
11	Month	Last contact month of year (Jan, Feb,…, Dec)
12	Duration	Last contact duration, in seconds
13	Campaign	Number of contacts performed during this campaign and for this client
14	Pdays	Number of days that passed by after the client was last contacted from a previous campaign
15	Previous	Number of contacts performed before this campaign and for this client
16	Poutcome	Outcome of the previous marketing campaign (Unknown, Other, Failure, Success)
17	Class : deposit	Yes No

4.5. Explanation of the System

Proposed bank depositor classification system is explained by using ID3 and NB classifiers. This system is tested by using 16 patient records. Sample bank training data is shown in Table 4.2.

Table 4.2. Sample Bank Training Data

Age	**Job**	**Martial**	**Education**	**Default**	**Balance**	**Housing**	**…**	**Class**
35	management	single	Tertiary	no	1350	yes	…	no
33	management	married	secondary	no	3935	yes	…	yes
61	admin	married	unknown	no	4629	yes	…	yes
34	technician	married	Tertiary	no	1539	yes	…	yes
32	blue-collar	single	secondary	no	228	no	…	no
25	admin	single	Tertiary	no	760	yes	…	yes
32	technician	single	Tertiary	no	13711	yes	…	no
59	technician	married	primary	no	4198	no	…	yes

26	blue-collar	single	primary	no	155	yes	…	no
37	management	married	Tertiary	no	0	no	…	yes
51	blue-collar	married	secondary	no	160	yes	…	no
73	retired	married	primary	no	796	no	…	yes
83	retired	married	primary	no	425	no	…	yes
35	blue-collar	single	secondary	no	0	yes	…	yes
34	management	married	Tertiary	no	1310	no	…	no

To know the bank customer's deposit result, the user first inputs the unknown sample (X) that is unknown bank customer's information. Unknown sample (X) is shown in Table 4.3.

Table 4.3. Unknown Sample (X)

Age: 29	Contact: cellular
Job: Admin	Day: 16
Marital: Single	Month: Apr
Education: Secondary	Duration: 185
Default: No	Campaign: 1
Balance: 1350	Pdays: 330
Housing: Yes	Previous: 1
Loan: No	Poutcome: Failure

According to the Naive Bayesian classifier, this system calculates the probability for each attribute. Based on probability results, this system classifies the unknown sample. The probability results of NB classifier are shown in Table 4.4.

Table 4.4. Probability Results of NB Classifier

Attribute	Probability Results	
	Class (No)	Class (Yes)
Age: 29	0.142857	0

Job: admin	0.142857	0.222222
Marital: single	0.714286	0.222222
Education: secondary	0.428571	0.222222
Default: no	1	1
Balance: 1350	0.142857	0
Housing: yes	0.571429	0.555556
Loan: no	1	1
Contact: cellular	0.714286	0.888889
Day: 16	0.142857	0.333333
Month: apr	0.142857	0.111111
Duration: 185	0.142857	0
Campaign: 1	0.857143	0.777778
Pdays: 330	0.142857	0
Previous: 1	0.285714	0.333333
Poutcome: failure	0.714286	0.444444

After calculating each attribute probabilities, this system obtains the "Class = No" probability that is "0.000000011611" and the "Class=Yes" probability that is "0". So, this system produces that the "**Bank Customer's Deposit Result is No**" for user inputted unknown sample.

According to the ID3 classifier, this system calculates information gain for each iteration. First iteration gain results are shown in Table 4.5.

Table 4.5. First Iteration Gain Results

ID	**Attribute and Its Information Gain Results**	
	Attribute (A)	**Information Gain Results**
1	Age	0.014
2	Job	0.007

3	Marital	0.182
4	Education	0.023
5	Default	0
6	Balance	0.989
7	Loan	0.001
8	Contact	0
9	Day	0.036
10	Month	0.661
11	Housing	0.239
12	Duration	0.989
13	Campaign	0.055
14	Pdays	0.989
15	Previous	0.462
16	Poutcome	0.057

In the first iteration, the "Balance" attribute is the root node because it has highest information gain. The decision tree from the first iteration is shown in Figure 4.4.

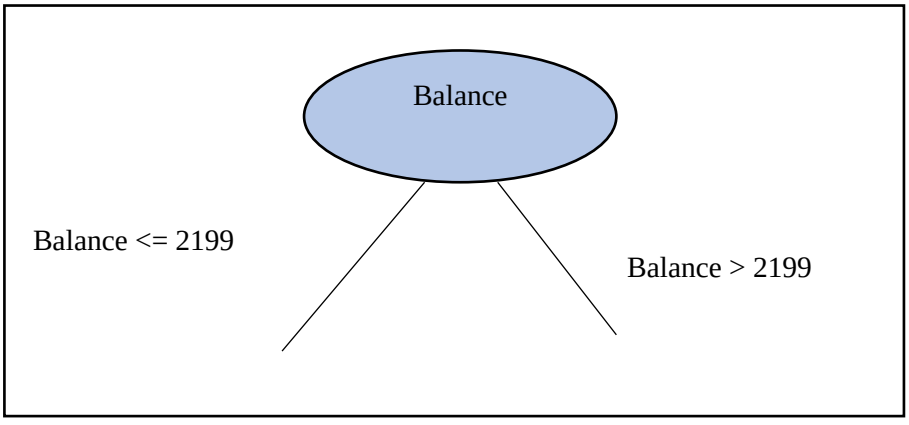

Figure 4.4 Decision Tree from the First Iteration

Gain result from the second iteration for Balance >2199 is shown in Table 4.6.

Table 4.6. Information Gain Result from Second Iteration for Balance >2199

ID	Attribute and Its Information Gain Results	
	Attribute (A)	**Information Gain Results**
1	Age	0.811
2	Job	0.311
3	Marital	0.811
4	Education	0.811
5	Default	0
6	Housing	0.122
7	Loan	0
8	Contact	0
9	Day	0.311
10	Month	0.311
11	Duration	0.811
12	Campaign	0
13	Pdays	0.811
14	Previous	0.811
15	Poutcome	0.122

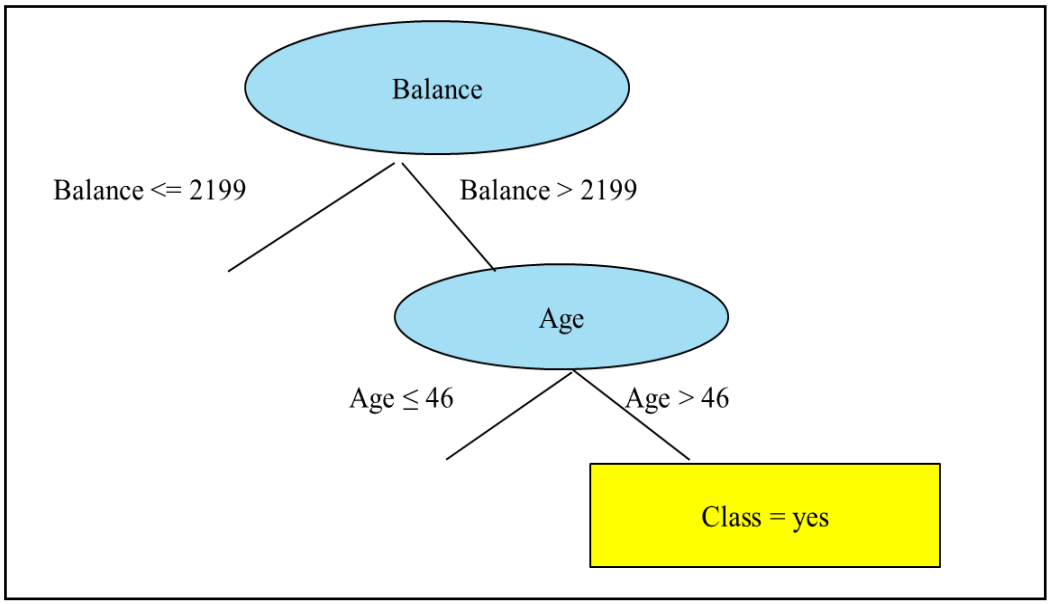

Figure 4.5 Decision Tree from the Second Iteration for Balance >2199

In the second iteration for Balance >2199, the “Age” attribute is the root node because it has highest information gain. The decision tree from the second iteration for Balance >2199 is shown in Figure 4.5. Gain result from the second iteration for Balance ≤ 2199 are shown in Table 4.7.

Table 4.7. Information Gain Result from Second Iteration for Balance ≤ 2199

ID	Attribute and Its Information Gain Results	
	Attribute (A)	**Information Gain Results**
1	Job	0.333
2	Marital	0.082
3	Education	0.096
4	Default	0
5	Housing	0
6	Loan	0
7	Contact	0.027
8	Day	0.096
9	Month	0.5
10	Duration	0.770
11	Campaign	0.090
12	Pdays	0.699
13	Previous	0.104
14	Poutcome	0.699

In the second iteration for Balance ≤ 2199, the “Duration” attribute is the root node because it has highest information gain. The decision tree from the second iteration for Balance ≤ 2199 is shown in Figure 4.6. Gain result from the third iteration for Age ≤ 46 are shown in Table 4.8.

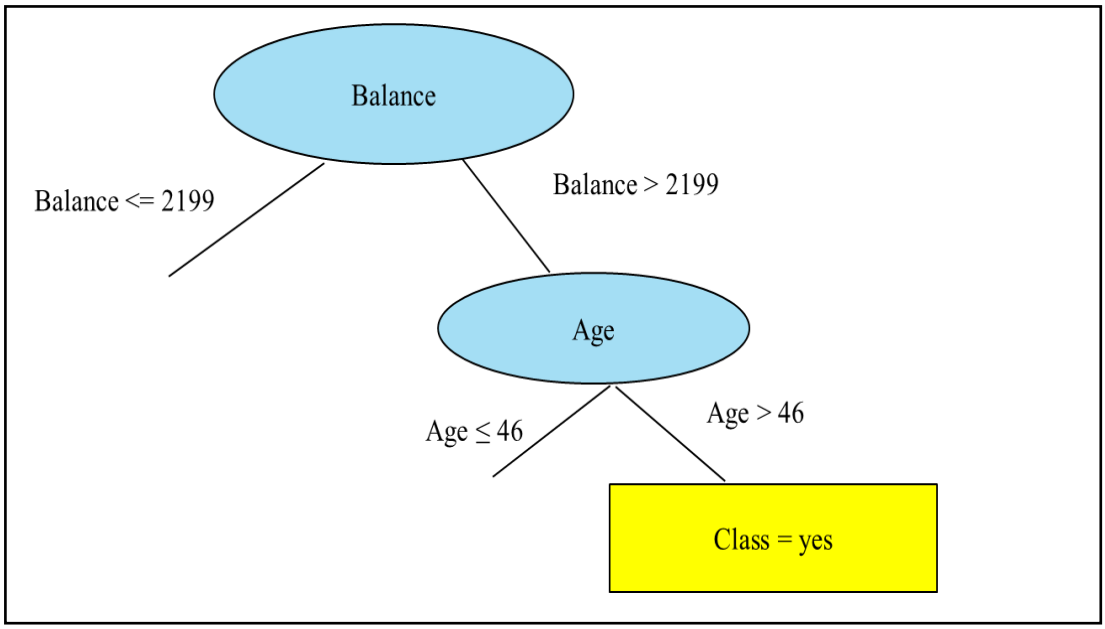

Figure 4.6 Decision Tree from the Second Iteration for Balance >2199

Table 4.8. Information Gain Result from Third Iteration for Age ≤ 46

ID	Attribute and Its Information Gain Results	
	Attribute (A)	**Information Gain Results**
1	Job	0.333
2	Marital	0.699
3	Education	0.096
4	Default	0
5	Housing	0
6	Loan	0
7	Contact	0.027
8	Day	0.096
9	Month	0.5
10	Pdays	0.5
11	Previous	0.104
12	Poutcome	0.333

In the third iteration for Age ≤ 46, the "Martial" attribute is the root node because it has highest information gain. The decision tree from the third iteration for Age ≤ 46 is shown in Figure 4.7. Gain result from the third iteration for Duration ≤ 410 are shown in Table 4.9.

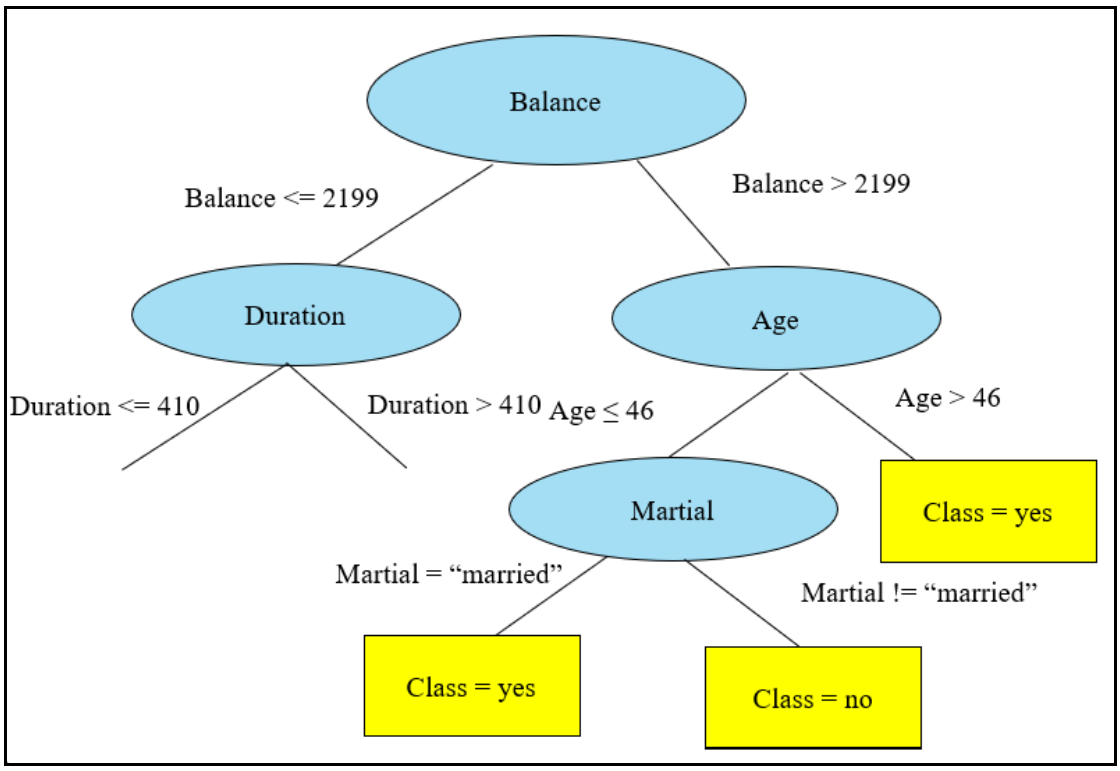

Figure 4.7 Decision Tree from the Third Iteration for Age ≤ 46

Table 4.9. Information Gain Result from Third Iteration for Duration ≤ 410

ID	Attribute and Its Information Gain Results	
	Attribute (A)	**Information Gain Results**
1	Job	0.333
2	Default	0
3	Housing	0
4	Loan	0
5	Contact	0.027
6	Day	0.096
7	Month	0.5

8	Campaign	0.867
9	Pdays	0.5
10	Previous	0.104
11	Poutcome	0.333

In the third iteration for Duration ≤ 410, the "Campaign" attribute is the root node because it has highest information gain. The decision tree from the third iteration for Duration ≤ 410 is shown in Figure 4.8. Gain result from the third iteration for Duration > 410 are shown in Table 4.10.

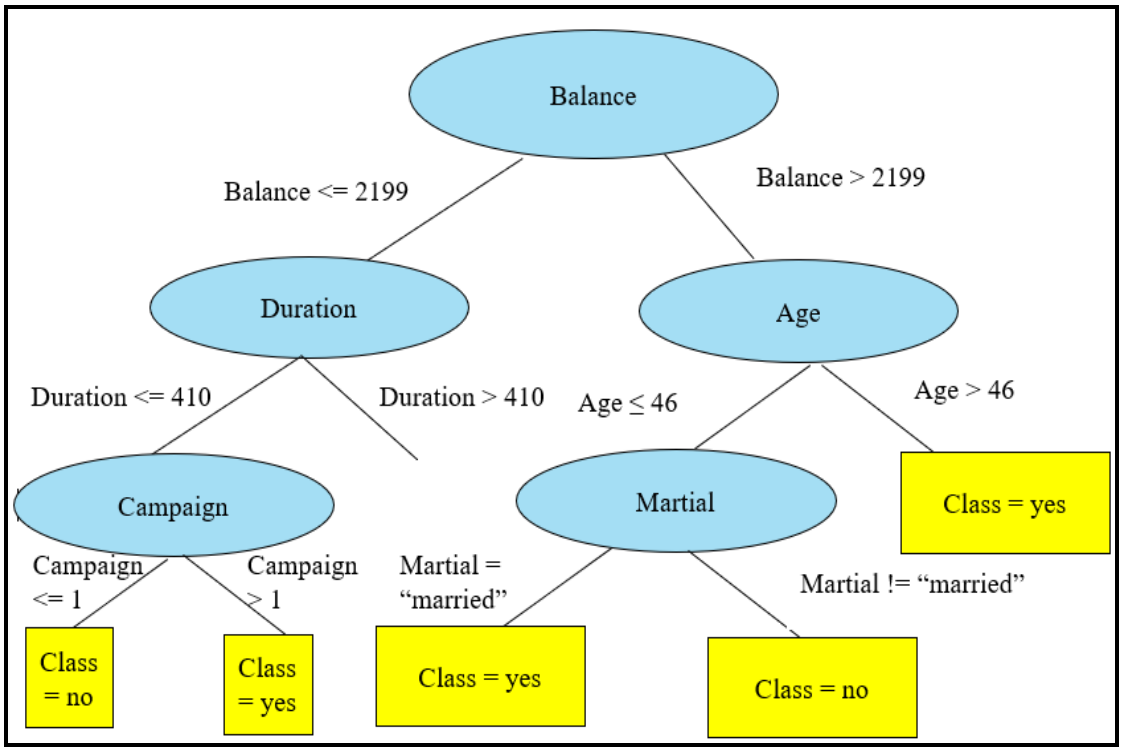

Figure 4.8 Decision Tree from the Third Iteration for Duration ≤ 410

Table 4.10. Information Gain Result from Fourth Iteration for Duration > 410

ID	Attribute and Its Information Gain Results	
	Attribute (A)	**Information Gain Results**
1	Job	0.51
2	Default	0

3	Housing	0
4	Loan	0
5	Contact	0.027
6	Day	0.096
7	Month	0.333
8	Pdays	0.231
9	Previous	0.104
10	Poutcome	0.333

In the fourth iteration for Duration > 410, the "Job" attribute is the root node because it has highest information gain. The decision tree from the fourth iteration for Duration > 410 is shown in Figure 4.9.

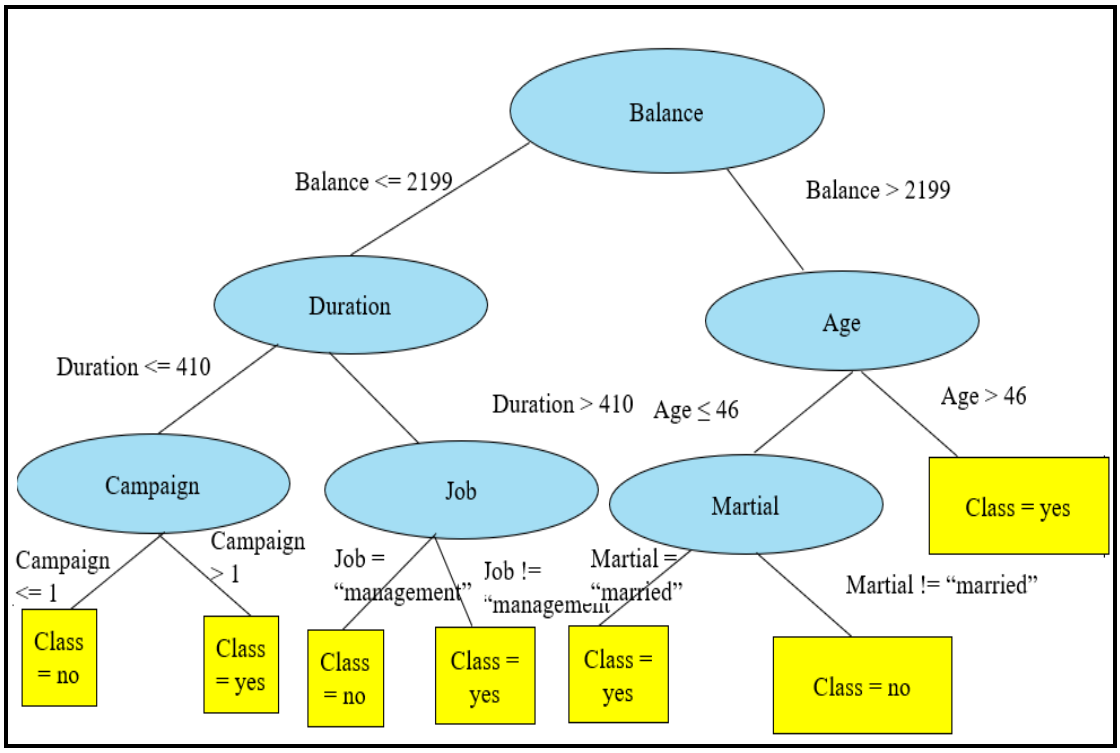

Figure 4.9 Decision Tree from the Fourth Iteration for Duration > 410

By using final decision tree, this system produces the decision rules to classify the unknown sample (user inputted bank customer's information). The decision rules are as follows:

- Rule 1 is {IF "Balance ≤ 2199" AND "Duration ≤ 410" AND "Campaign ≤ 1" THEN Class = No}
- Rule 2 is {IF "Balance ≤ 2199" AND "Duration ≤ 410" AND "Campaign > 1" THEN Class = Yes}
- Rule 3 is {IF "Balance ≤ 2199" AND "Duration > 410" AND "Job = Management" THEN Class = No}
- Rule 4 is {IF "Balance ≤ 2199" AND "Duration > 410" AND "Job != Management" THEN Class = Yes}
- Rule 5 is {IF "Balance > 2199" AND "Age ≤ 46" AND "Martial = married" THEN Class = Yes}
- Rule 6 is {IF "Balance > 2199" AND "Age ≤ 46" AND "Martial != married" THEN Class = No}
- Rule 7 is {IF "Balance > 2199" AND "Age > 46" THEN Class = Yes}

According to the **Rule 1**, this system produces that the "**Bank Customer's Deposit Result is No**" for user inputted unknown sample.

4.6. Implementation of the System

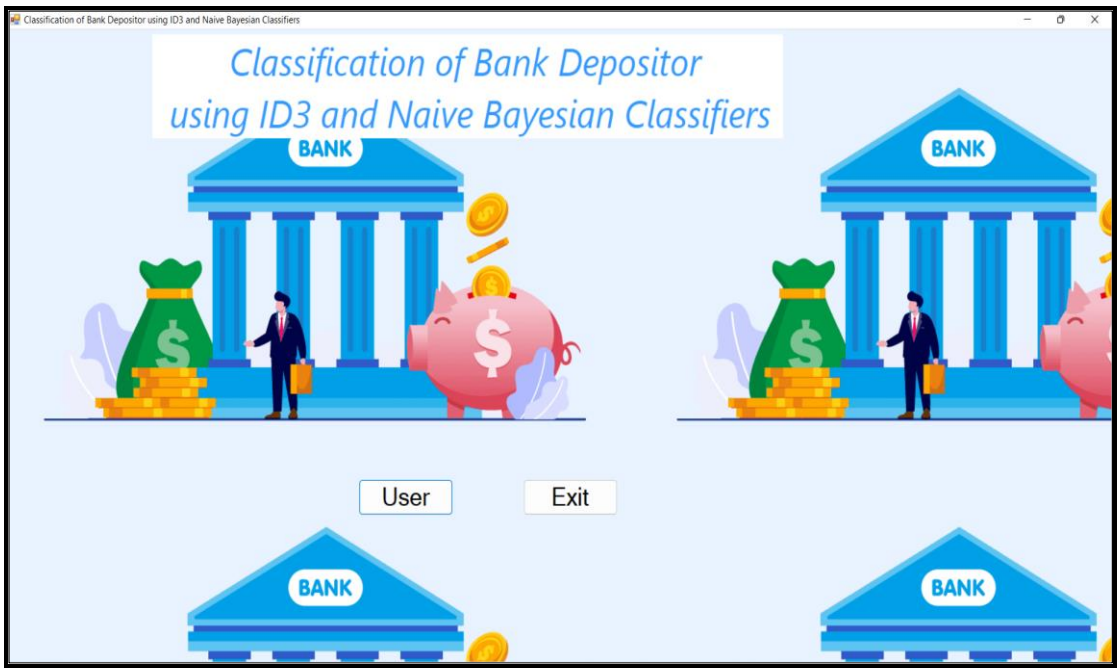

Figure 4.10 Welcome Form of the System

For the financial domain, this system is proposed as the bank depositor classification system by using ID3 and Naive Bayesian classifiers. This system is implemented by using

Microsoft visual studio 2012, C# programming language. This system also uses the Microsoft access database for bank training and testing data. Welcome page of the system is shown in Figure 4.10. In the welcome page, “User” and “Exit” buttons are included. By using the “User” button, the bank manager (user) can use the proposed bank depositor classification system. Otherwise, the user can leave from the system by using “Exit” button.

4.6.1. Home Menu of the System

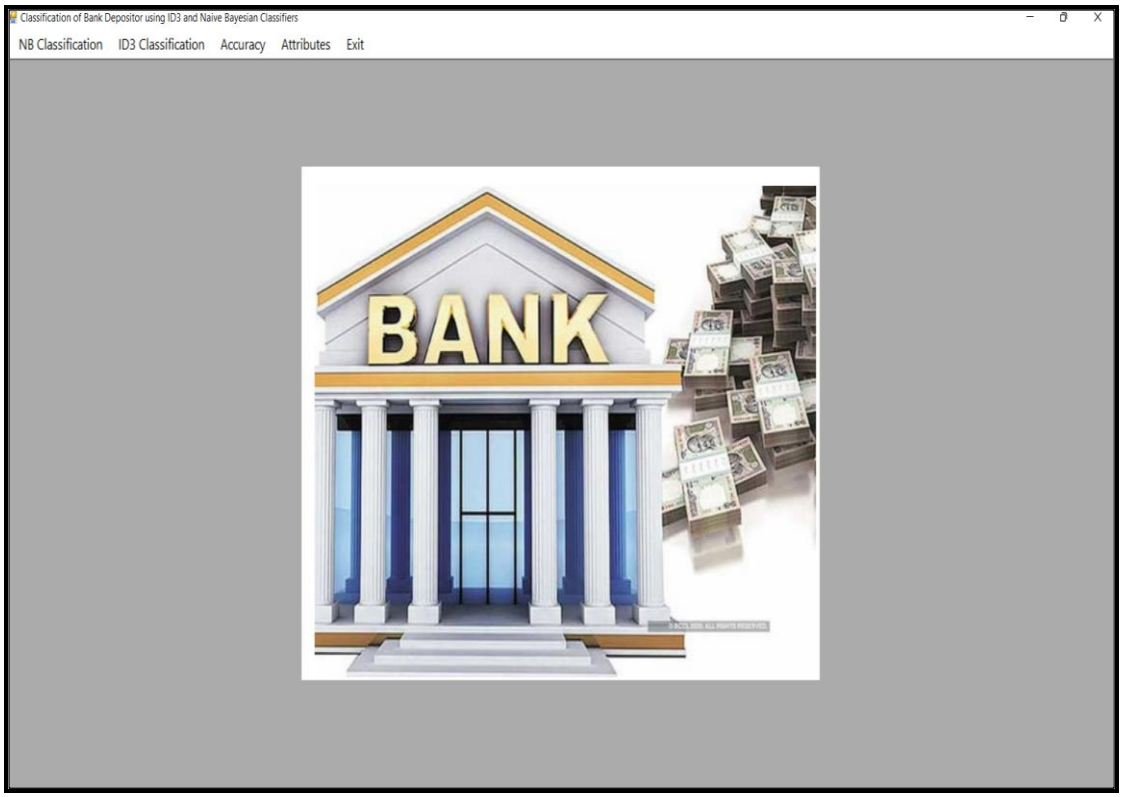

Figure 4.11 Main Menu of the System

Home Menu of the system is shown in Figure 4.11. In this home page, there are five menus. These are “NB Classification”, “ID3 Classification”, “Accuracy”, “Attributes” and “Exit” menus. If the user wants to classify the unknown sample according to Naive Bayesian classifier, the user must use the “NB Classification” menu. Otherwise, if the user wants to classify these unknown sample by using ID3 decision tree classifier, the user must use the “ID3 Classification” menu. To point out the accuracy of each classifier and to compare the performance of these two classifiers, the user must use “Accuracy” menu. The user can view the bank customer’s attribute information via “Attributes” menu. To release from the system, the user can use “Exit” menu.

4.6.2. Naive Bayesian Classification Process

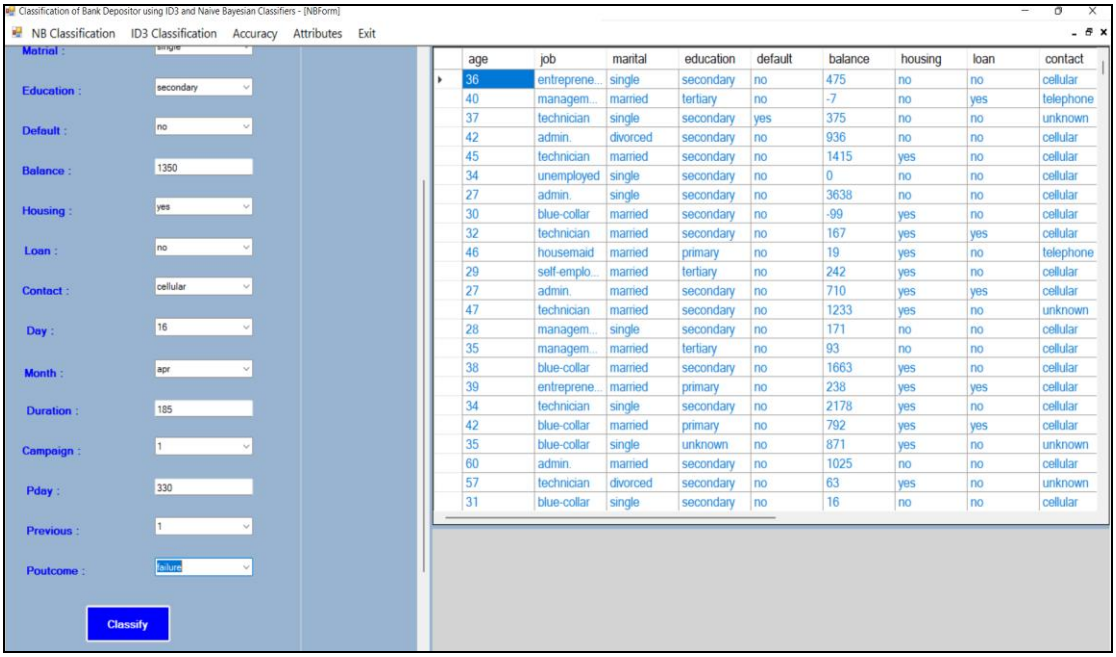

Figure 4.12 Unknown Sample for NB Classier

Unknown sample for NB classifier is shown in Figure 4.12. For Naive Bayesian classification, the user must first input the bank customer's attribute information that includes "Age", "Job", "Martial", "Education" and so on. After inputting the unknown sample, the user must use "Classify" button to obtain the class (Yes or No) of the unknown sample. In this form, this system shows the training bank customer' information.

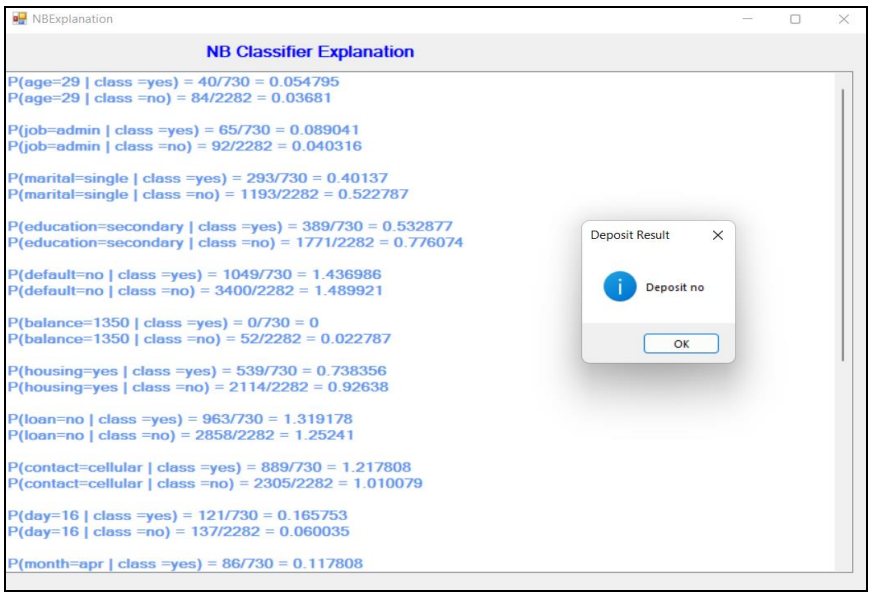

Figure 4.13 Probability Results of NB Classier

Probability results of NB classifier are shown in Figure 4.13. After using "Classify" button, the system produces the deposit result and detail probability results about NB classifier. Based on the class (Yes, No) probability results, this system chooses the highest probability class result for the unknown bank customer's information. Highest probability class result of NB classifier is shown in Figure 4.14.

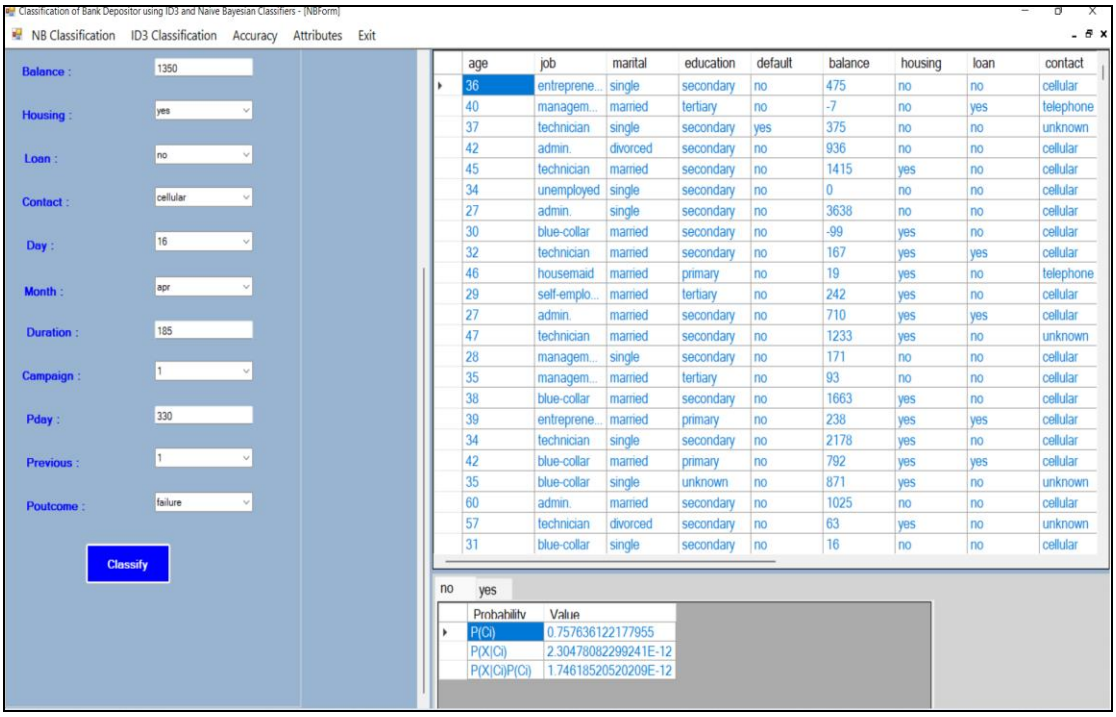

Figure 4.14 Highest Probability Class Result of NB Classier

4.6.3. ID3 Classification Process

For ID3 classification, the user must use "ID3 classification" menu. Then, to generate the decision tree, this system first calculates the information gain about each bank customer's attribute. For each iteration, this system chooses the attribute as the root node that has highest probability result. After finishing each iteration, this system produces the decision tree that is needed to generate the decision rule generation. The decision tree result is shown in Figure 4.15.

Then, this system generates the decision rules from the decision tree. By using the generated decision rules, this system classifies the user inputted bank customer's information to produce the customer who is bank depositor or not. The decision rules result is shown in Figure 4.16.

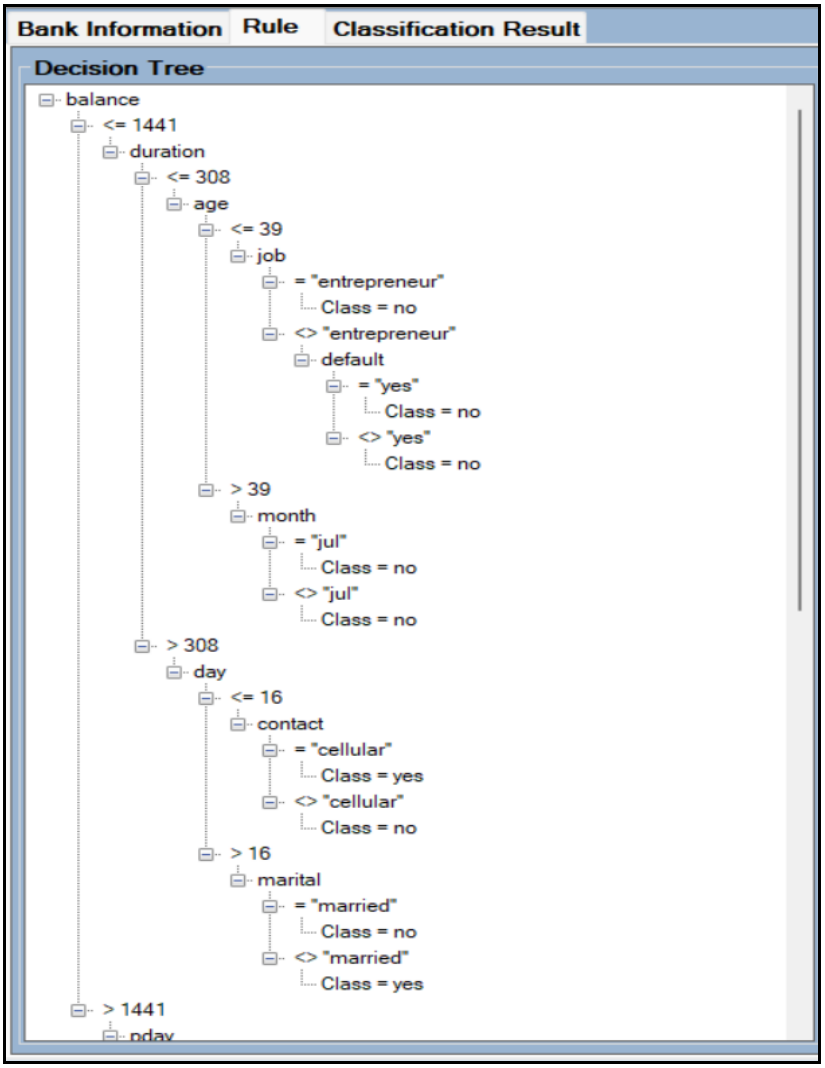

Figure 4.15 Decision Tree Result

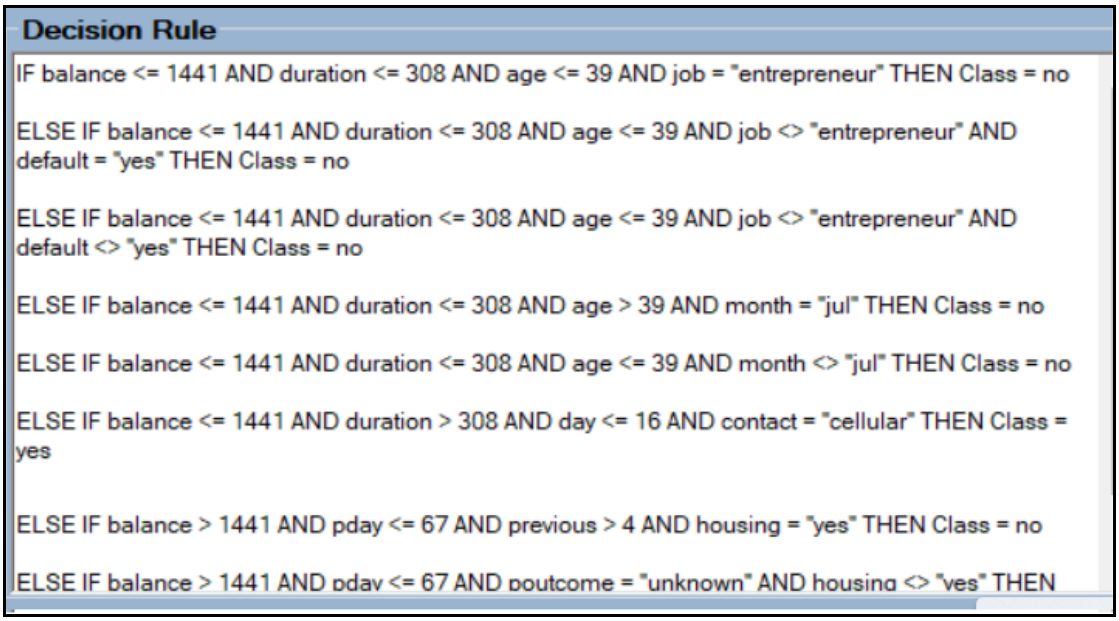

Figure 4.16 Decision Rule Result

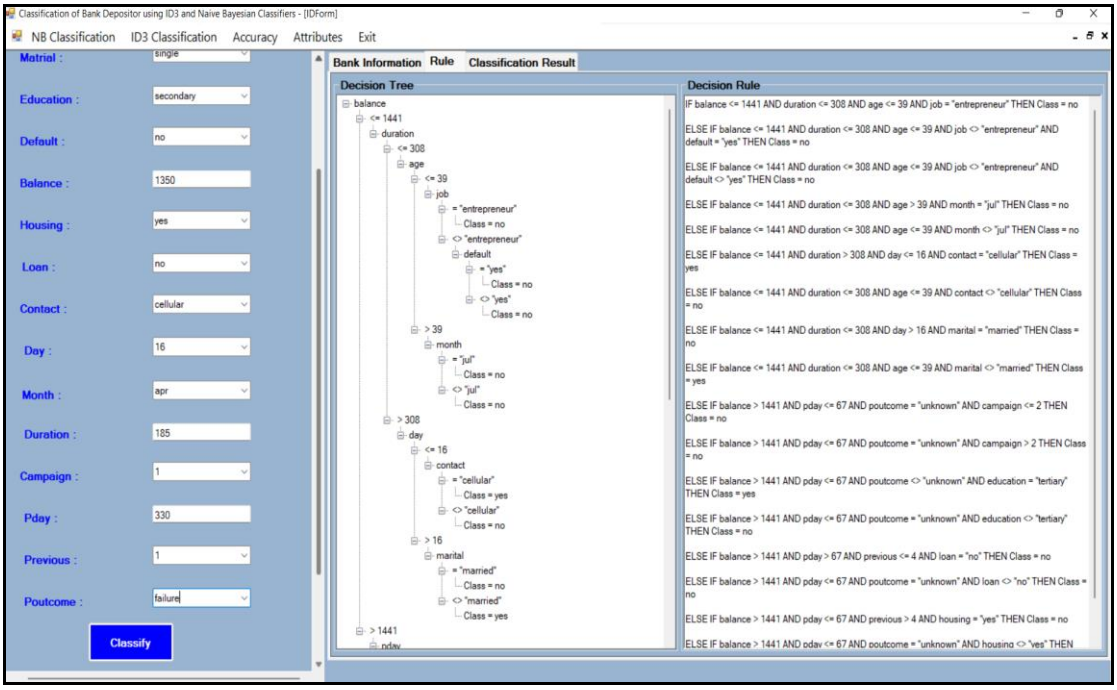

Figure 4.17 Unknown Sample for ID3 Classifier

Unknown sample for ID3 classifier is shown in Figure 4.17. In the ID3 classification process, the user must input the bank customer's attribute information for classification. After inputting this information, the user must use the "Classify" button. And then, this system produces the classification result that points out the customer who is bank depositor or not. The classification result of ID3 classifier is shown in Figure 4.18.

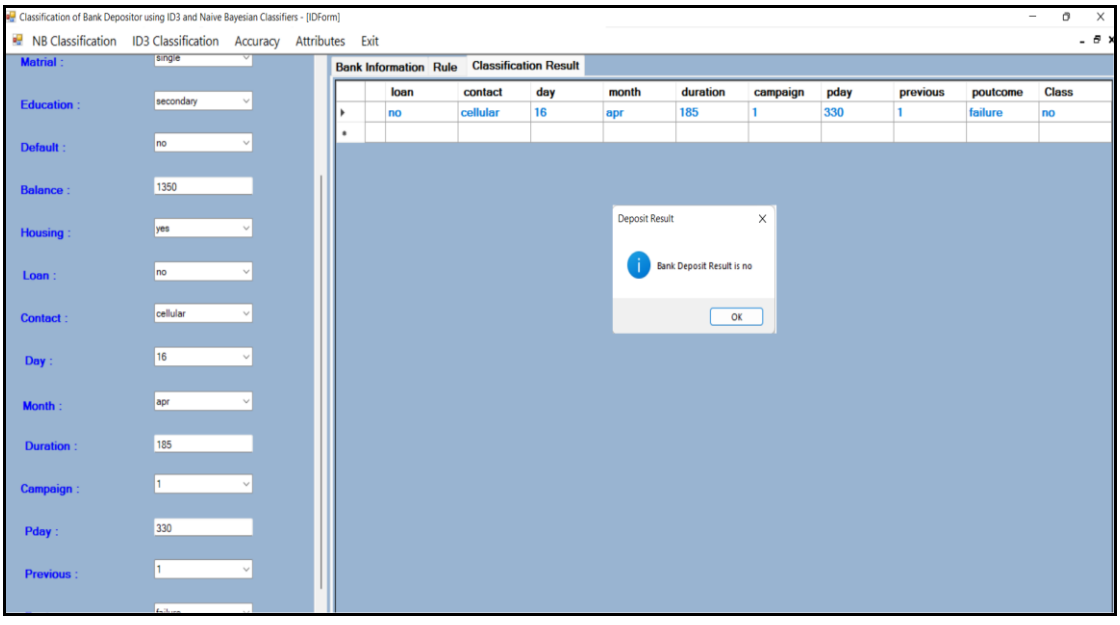

Figure 4.18 Classification Result of ID3 Classifier

4.6.4. Accuracy of the System

To measure the accuracy of each classifier, this system uses the testing bank customer's dataset and holdout method. The user must use the "Accuracy" menu to calculate the sensitivity, specificity and accuracy of each classifier. The testing bank customer's dataset is shown in Figure 4.19.

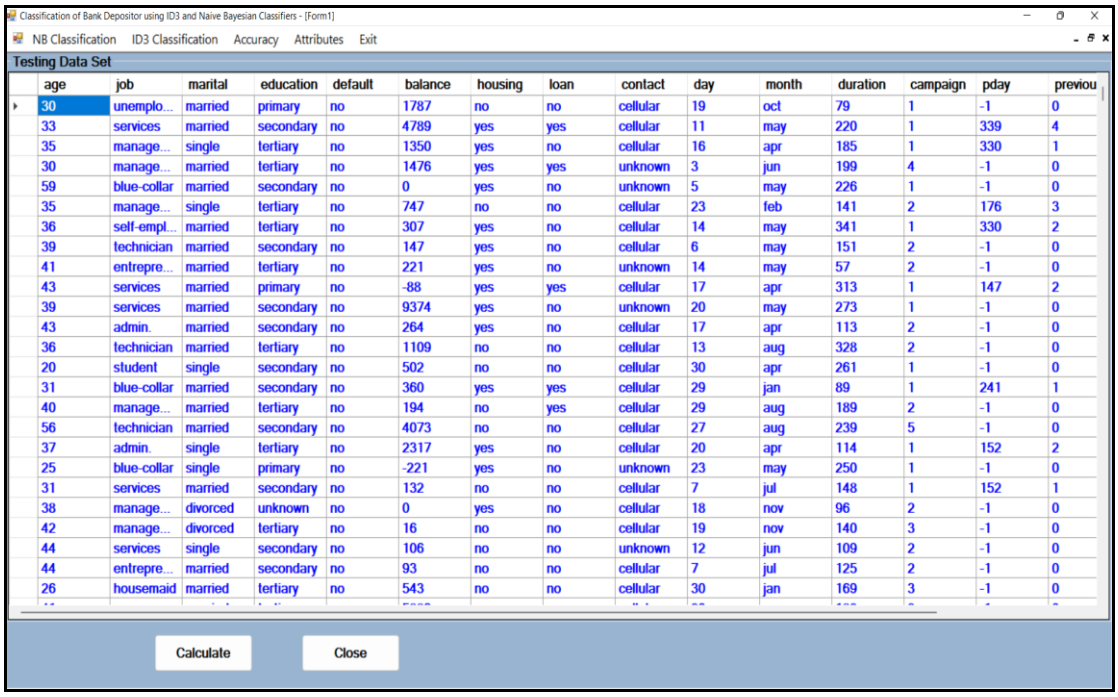

Figure 4.19 Testing Bank Customer's Dataset

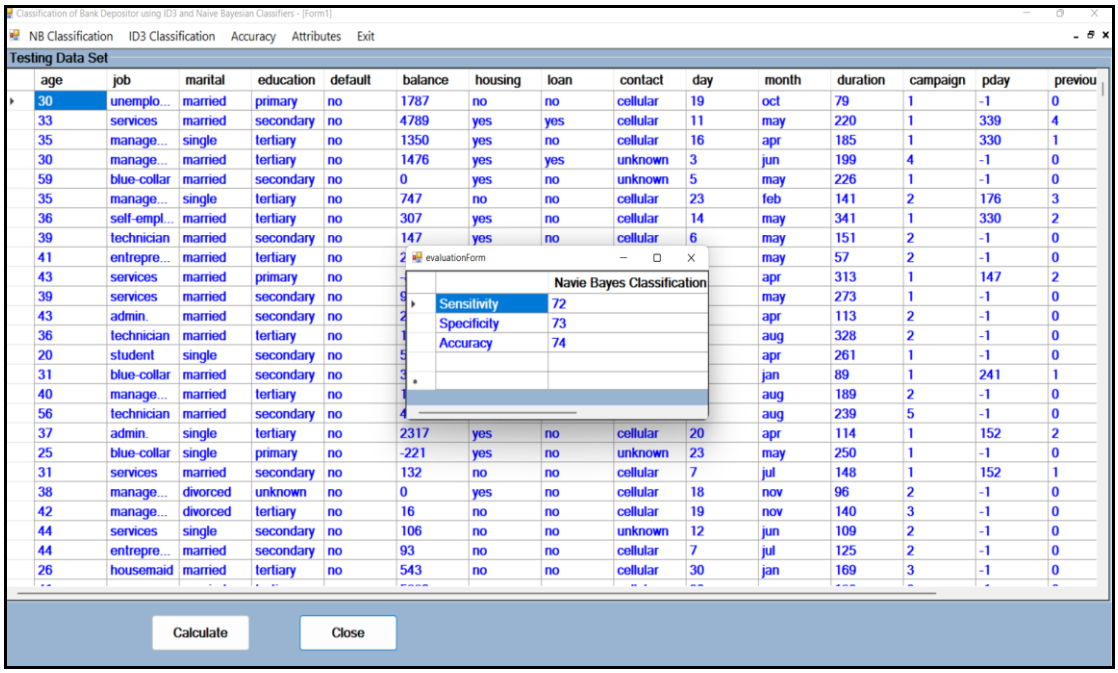

Figure 4.20 NB Accuracy Result

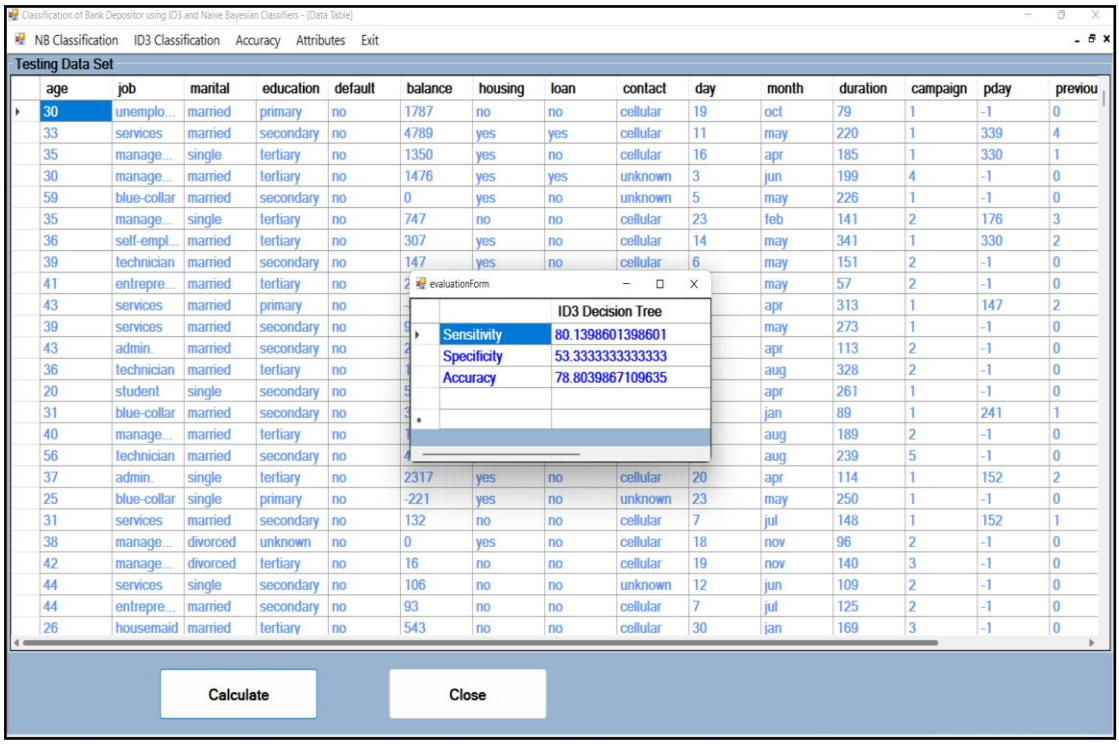

Figure 4.21 ID3 Accuracy Result

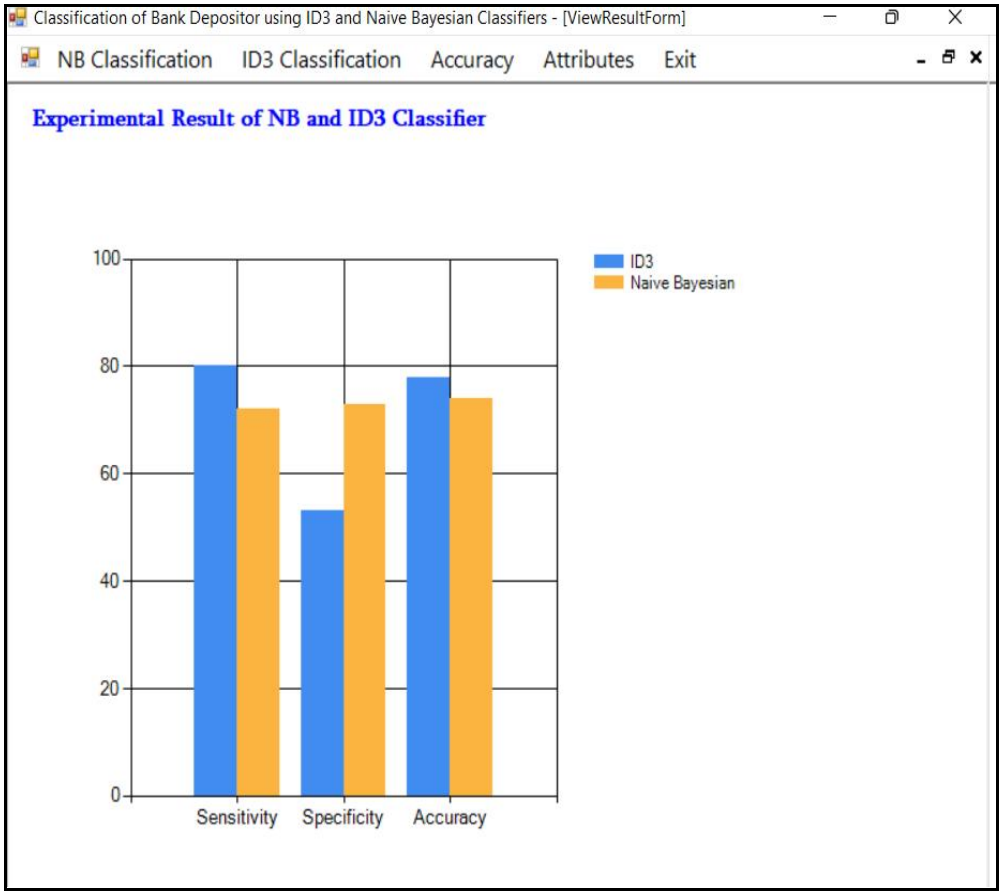

Figure 4.22 Two Classifier Results

After choosing “Accuracy” menu, the user continues to choose the desired sub-menu among three sub-menus. These menus are “NB Accuracy”, “ID3 Accuracy” and “View Result” sub-menus. If the user wants to calculate the performance of NB classifier, the user must choose “NB Accuracy” sub-menu. If the user wants to calculate the performance of ID3 classifier, the user must choose “ID3 Accuracy”. Then, if the user wants to compare these two classifiers, the user must use the “View Result” sub-menu. NB accuracy result is shown in Figure 4.20. ID3 accuracy result is shown in Figure 4.21 and two classifier results are shown in Figure 4.22.

4.6.5. Bank Customer’s Attribute Information

This system allows the user to understand the detail attribute information. To look this information, the user must use “Attributes” menu. Bank customer’s attribute information is shown in Figure 4.23.

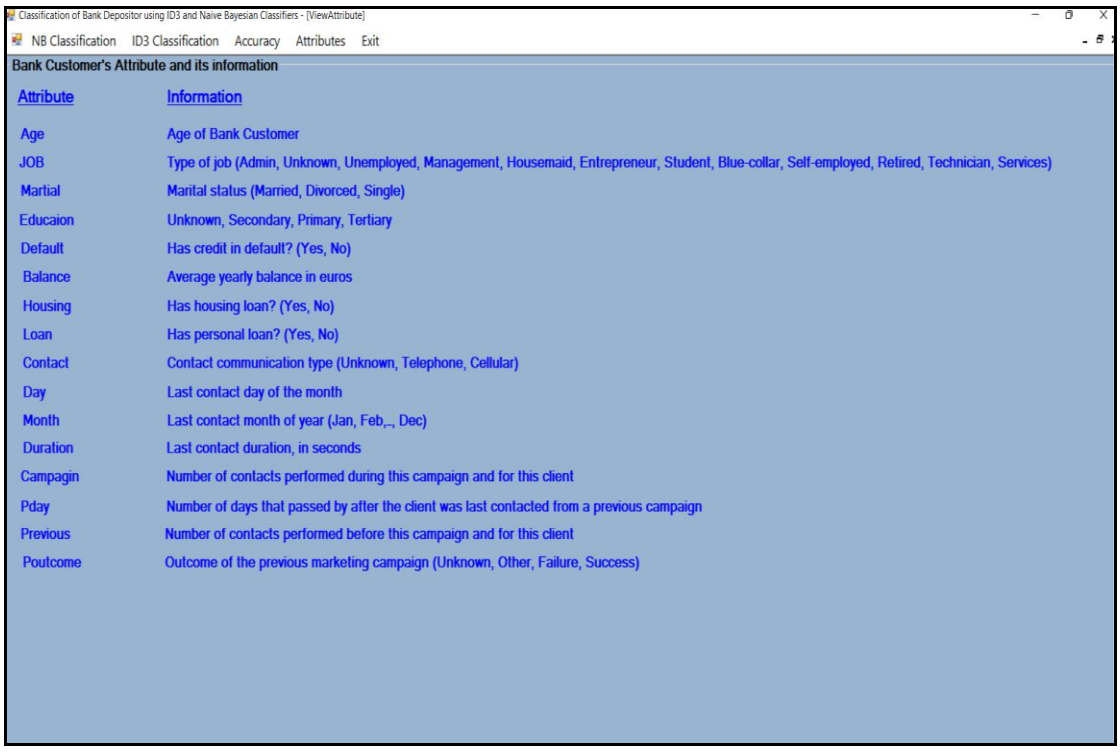

Figure 4.23 Bank Customer’s Attribute Information

4.7. Experimental Results of the System

To compare the performance of ID3 and NB classifiers, this system calculates the sensitivity, specificity and accuracy of these two classifiers. For experimental result, this system uses “4516” bank data records. According to holdout method, this

system splits these bank data records into “3008” training and “1508” testing records. Table 4.11 shows the sensitivity, specificity and accuracy results of ID3 and NB classifiers.

Table 4.11. Sensitivity, Specificity and Accuracy Results of ID3 and NB Classifier

Performance Measurement	ID3 Classifier	NB Classifier
Sensitivity	80 %	74 %
Specificity	53 %	73 %
Accuracy	79 %	76 %

Sensitivity results of ID3 and NB classifier are shown in Figure 4.24. According to the sensitivity results, the ID3 classifier has truer positive rate than NB classifier.

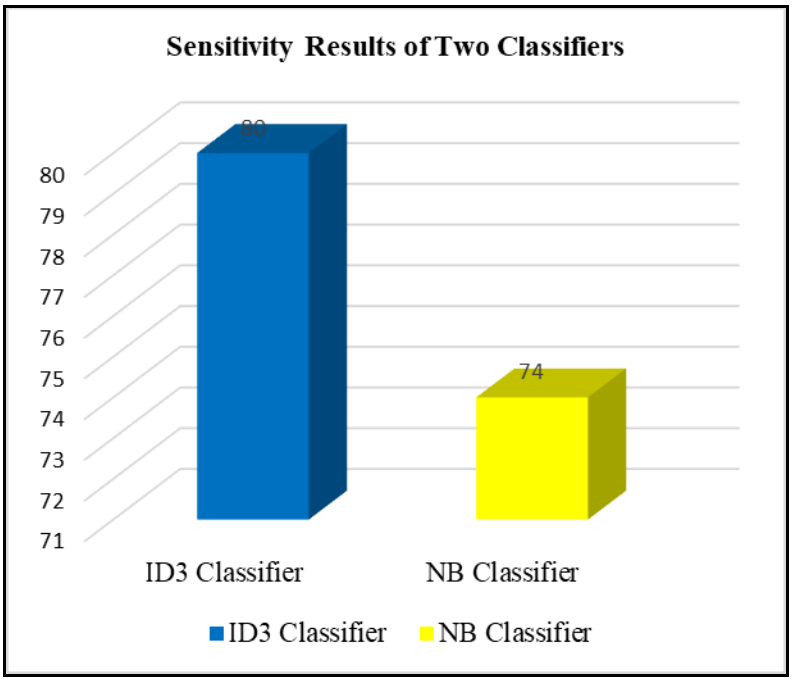

Figure 4.24 Sensitivity Result of ID3 and NB Classifier

According to the specificity results, the NB classifier has true negative rate than ID3 classifier. Specificity results of ID3 and NB classifier are shown in Figure 4.25.

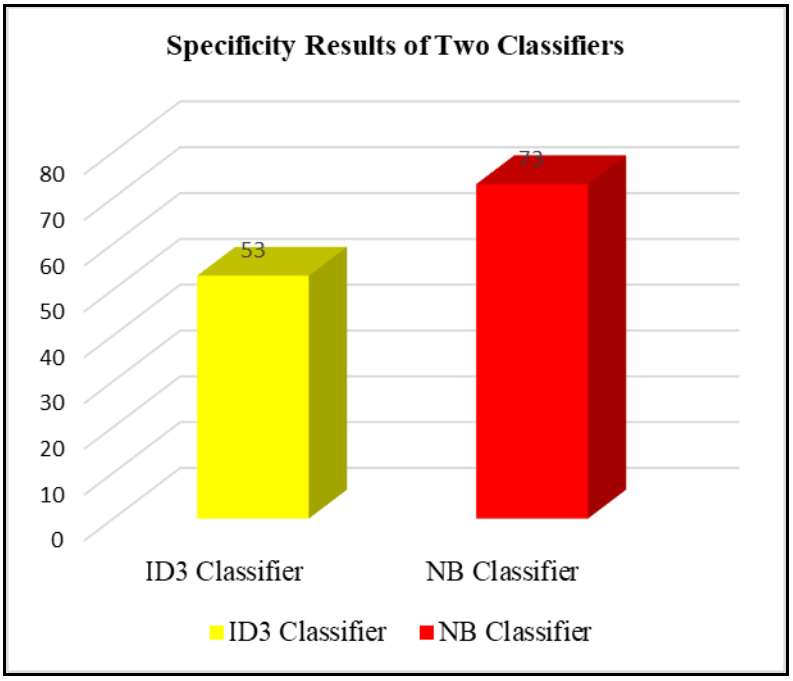

Figure 4.25 Specificity Result of ID3 and NB Classifier

According to the accuracy results, the ID3 classifier is more precise than NB classifier. Accuracy results of ID3 and NB classifier are shown in Figure 4.26.

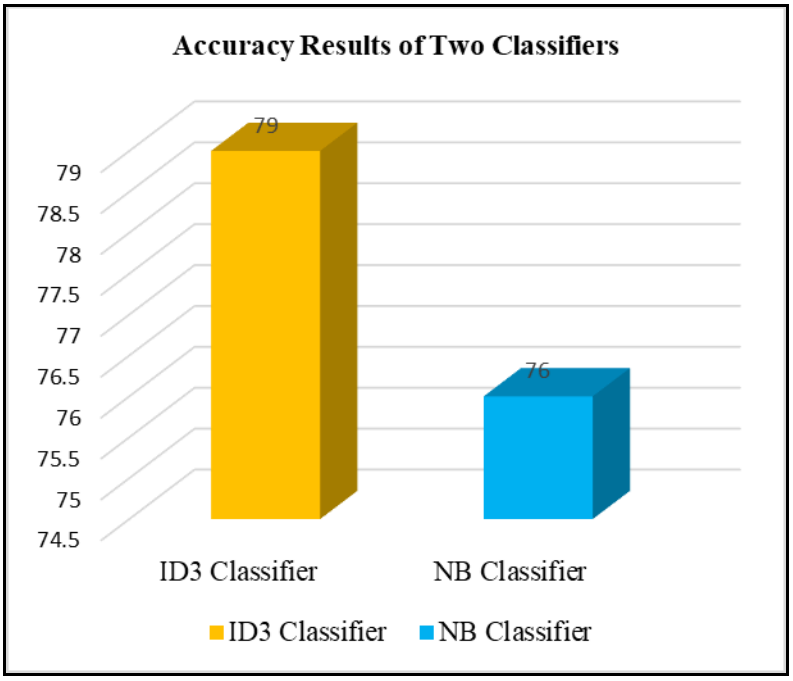

Figure 4.26 Accuracy Result of ID3 and NB Classifier

CHAPTER 5
CONCLUSION AND FURTHER EXTENSION

5.1 Conclusion

In conclusion, banking and marketing knowledge base are one of the most challenging issues that appears due to the vast growth of data records amount and transactions. Banks are as institutions which channel people' savings into productive loans and investments. Thus, banking mainly refers to deposits and loans. Deposits are broken down into two main categories that are demand deposits and time deposits.

Any deposit the customer makes that he/she can withdraw without notice is a demand deposit. Within this category, there are three main types of demand deposits: checking accounts, saving accounts and money market accounts. Then, it is considered a time deposit whenever a bank deposit comes with a fixed rate and term. With time deposits, the customer isn't allowed to withdraw money for a specific period of time. Accordingly, various data mining approaches have been presented to enable better decision making. So, this system focuses on decision tree and naive Bayesian theory as their abilities for classification of new objects (new bank customer's information).

For bank depositor classification, this system used the ID3 decision tree and Naive Bayesian (NB) classifiers. This system resulted in each classifier to predict whether a customer will subscribe to a long-term deposit or not. This system is also implemented to show which classifier is more than another. To compare the performance of ID3 and NB, this system calculates the sensitivity, specificity and accuracy of each classifier. According to the performance analysis results that are obtained by testing 4516 bank customer's data records, ID3 classifier is more precise than NB classifier. But NB classifier has more specificity results than ID3 classifier. Due to the ID3 classifier is based on the information gain measure and rule-based methods, this classifier is more precise than NB classifier. NB classifier can't solve the zero-probability situation. So, NB classifier would be sometimes classified the bank customer's data at a wrong situation.

Finally, the system is helpful for bank manager during bank depositor classification. This system can help the bank for identifying customers who will potentially open a time deposit so that it can be used to assist the performance and

operations of the bank. This system also showed the effectiveness of decision tree (ID3) and Naive Bayesian classifiers. So, the bank manager can know which data mining classifier is more useful and precise than another classifier.

5.2 Further Extension

The proposed bank depositor classification system can be classified only bank customer's data from the UCI (university of California) website. This system can be extended to implement for direct marketing through email or instant messaging. Then, this system only used two classifiers for bank depositor classification. In this situation, this system can be extended by using deep learning such as artificial neural network method.

REFERENCES

[1] A. Rajkumar, G. S. Reena, “Diagnosis of Heart Disease Using Data mining Algorithm”, Global Journal of Computer Science and Technology, September 2010.

[2] B. Liu, “Web Data Mining”, ACM Computing Classification, 1998.

[3] F. Voznika, L. Viana, “Data Mining Classification”, European Conference, PKDD, 2001.

[4] G. Subbalakshmi, K. Ramesh, M. C. Rao, “Decision Support in Heart Disease Prediction System using Naïve Bayes”, Indian Journal of Computer Science and Engineering (IJCSE), May 2011.

[5] H. I. Witten, E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques”, Second Edition, Morgan Kaufmann Publishers, San Francisco, 2005.

[6] J. A. M. Berry, L. S. Gordon, “Data Mining Techniques for Marketing, Sales, and Customer Relationship Management”, Second Edition, Wiley Publishing, Inc., 2004.

[7] J. Han, M. Kamber, “Data Mining: Concepts and Techniques”, Second Edition, Morgan Kaufmann Publishers, San Francisco, 2006.

[8] M. Kantardzic, “Data Mining: Concepts, Models, Methods, and Algorithms”, John Wiley & Sons, New Jersey, 2003.

[9] M. Shouman, T. Turner, R. Stocker, “Using Decision Tree for Diagnosing Heart Disease Patients”, Australasian Data Mining Conference, 2011.

[10] Y. Singh, A. S. Chauhan, “Neural Networks in Data Mining”, Journal of Theoretical and Applied Information Technology, 2009.

[11] https://en.wikipedia.org/wiki/Data_mining

[12] http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm

[13] S. K. Agrawal, "Metrics to Evaluate Your Classification Model to Take the Right Decisions", Data Science Blogathon, July, 2021.

[14] W. Zhu, N. Zeng and N. Wang, "Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS Implementations", Health Care and Life Sciences, 2010.

[15] M. H. Effendy, D. Anggraeni, “Classification of Bank Deposit Using Naive Bayes Classifier (NBC) and K-Nearest Neighbor (KNN)”, Proceedings of the International Conference on Mathematics, Geometry, Statistics and Computing, 2021.

[16] F. Safarkhani and S. Moro, “Improving the Accuracy of Predicting Bank Depositor's Behavior Using a Decision Tree”, Applied Sciences, 2021.

[17] S. Abbas, "Deposit Subscribe Prediction using Data Mining Techniques based Real Marketing Dataset", International Journal of Computer Applications, vol. 110, no. 3, pp. 1-7, 2015.

AUTHOR'S PUBLICATION

Moe San Phyu, Dr. Zaw Tun, "Classification of Bank Depositor using ID3 and Naïve Bayesian Classifiers", the Proceedings of the 10th Conference on Parallel and Soft Computing (PSC 2022), Yangon, Myanmar, 2022.