Expert System in Credit Analysis Using Machine Learning(ID3 and FP-growth algorithms)

By

TAHANI M. A. KASIH

Supervisor:

Dr. Abdelhamead M. Abdelkafy

This Thesis was submitted in Partial Fulfillment of the Requirements for Master's Degree of Science in Computer.

University of Benghazi
Faculty of Information Technology

December 2017

This Thesis (Expert System in Credit Analysis Using Machine Learning(ID3 and FP-growth algorithms)**) was Successfully**

Defended and Approved on

Examination CommitteeSignature

Dr………………………………………………..…,(Supervisor) , Chairman ……..…………………….....

Title (for example , Professor of Electrical Engineering)

Dr……………………………………………..,Member …………………………….………………

Title

Dr……………………………………………..…,Member………………………………..…………..

Title

Dr……………………………………………..…,Member………………………………..…………

Title

Dedication

To my father and the soul of my mother.

Acknowledgements

I begin with praising Allah for all of his blessings including fulfilling this work. I thank my father Muftah Alkaseh and my mother for their support and care. I also extend my thanks to my supervivor Dr. Abdulhameed Abdulkafi. I thank all of the staff members of the faculty of economy in Benghazi University Dr. Ali Zagoub, Dr. Abdulsalam Alkezza, and Dr. Ibraheem Blkhair. My thanks to financial analyst Mr. Husni Alomami. My thanks are also extended to Mr. Mohannad Ferjany for his proof editing. I would also like to extend my thanks to engineer Amina Alshakhi and engineer Rehab Alaqeeli for their support and encouragement. My thanks to my sisters and brother who shared me this jourrey.

TAHANI M. A. KASIH

List of Contents

List of Tables

List of Figures

List of Abbreviations OR SYMBOLS

Abbreviation	Meaning
AI	Artificial Intelligence
A. P. I. C.	Account Payable for Increasing Capital
ACC. Payable	Accrued Payable
Accum. Dep.	Accumulated Depreciation
Adv. Pay. Supp.	Advanced Pay Suppliers
ATO	Asset Turnover Rate
Balance sheet sta. Date	Balance Sheet Statement's Date
Build. and Equip.	Building And Equipment
Cash Coll.	Cash Collecting
CHK Under Coll.	Checks Under Collecting
COGS	Cost Of Goods
COGS/Sales	Cost Of Goods To Net Sales
Comm. Rec. No.	Commercial Record No.
Const. in Progress	Constructing In Progress
CPLTD	Current Payable Long Term Debt
Current	Current Ratio
Current Inc. Taxes	Current Income Taxes
Dep. Name	Department's Name
Dividends Prov.	Dividends Provision
Due From Sister Co.	Due From Sister Companies
Due To Related Co	Due To Related Companies
Due To Related Co.	Due To Related Companies
E. Sales Taxes	Eligibility For Sales Taxes
ES	Expert System
FG	Full Product
Fin. Stas. Date	Financial Statements Date
FL	Financial Leverage
FP Growth	Frequent Pattern Growth
Ga. on Sa. of Inv.	Gains On Sales Investments
GPM	Gross Profit Margin
ID3	Iterative Dichotomiser 3
IG	Information Gains
Income sta. Date	Income Statement Date
Int. Co. Pay Parent	Interest Co. pay parent
KE	Knowledge Engineering
LT E. Sales Taxes	Long-Term Eligibility For Sales Taxes
LTD to Affil. Co.	Long-Term Debt To Affiliate Companies
ML	Machine Learning
New CPLTD	New Current Payable Long Term Debt
New Int. Exp.	New Interest Expense
New LTD	New Long Term Debt

NLP	Natural Language Processing
Non Trade Recei.	Non Trade Receivable
NOP	Net Operating Profit
NOP/Sales	Net Operating Profit (Operating Margin)/Sales
NPAT	Net Profit After Taxes
NPAUI	Net Profit After Unusual Items
NPBT	Net Profit Before Taxes
Other Exp. or Inc.	Other Expenses Or Income
Prepaid Exp.	Prepaid Expenses
Prov. Doubt Recei.	Provisions Doubt Receivables
Quick	Quick Ratio
RM	Raw Material
RM Purchase Prov.	Raw Material Purchase Provision
ROA	Return On Assets
ROE	Return Of Equity
ROS	Return Of Sales
SGA	Selling, General, And Administrative Expenses
SGA/Sales	Selling, General, And Administrative Expenses To Sales
Sun. C. A.	Sundry Current Assets
Sundry Current Liabs.	Sundry Current Liabilities
Sundry Exp.	Sundry Expenses
Sundry NCL	Sundry Non-Current Liabilities
SVM	Support Vector Machines
Total Liabs. and Worth	Total Liabilities And Worth
WI/Sales	Working Investment To Sales
WIP	Working Investment Production

List of Appendices

Expert System in Credit Analysis Using Machine Learning(ID3 and FP-growth algorithms)

By

TAHANI M. A. KASIH

Supervisor

Dr. Abdulhamed M. Abdulkafi

Abstract

The major part of the expert system(ES) is the knowledge. The process of acquiring knowledge and learning compose a very difficult stage. Banks are commonly permitted to develop a model for determining their credit risk, the need of the methodology for developing and keeping this model is an important field. This poses questions on banks on which methods are appropriate, how to be consistent, how to be complete and how to provide transparency in their analysis and diagnosing. This thesis develops an expert system that can learn by a dataset from the Centeral Wahada Bank. By taking the data set and inducing rules. Consequently, the expert system can be constructed according to the induced rules. Thus, banks can be able to develop their models by the expert sysem which has capability of learning by using Iterative Dichotomiser 3 (ID3) algorithm and Frequent Pattern Growth (FP-Growth) algorithm as ML techniques.

Chapter 1. Artificial Intelligence

1.1.Introduction

Throughout the past ten centuries, humans have been trying to comprehend how they think. That is, how can just a bunch of matter observe, understand, predict, and manipulate a much bigger and more complex changing world. Artificial intelligence(AI) goes further than that, it does not only strive for understanding but also for building intelligent entities(Russell & Norvig, 2010).

AI is one of the most recent of science. Its name was formally iconed in 1956(Carbonell, Michalski, & Mitchell, 1983), (Smith, McGuire, Huang, & Yang, 2006). Computer requires *knowledge representation* to keep what it knows or hears; *machine learning(ML)* to be suitable for new situations and to expose and extrapolate or induce patterns; *automated reasoning* to use the stored information to respond to questions and to draw new conclusions; *natural language processing(NLP)* to enable it for connecting successfully in English, *computer vision* for recognizing objects, and *robotics* for manipulating objects. These six disciplines compose most of Al(Russell & Norvig, 2010).

ESs is considered one of the most successful applications(Villena Román, Collada Pérez, Lana Serrano, & González Cristóbal, 2011). Early 1950s, experiments in writing programs were conducted by several researchers to simulate human thought processes. AI community’s works demonstrated the concept of intelligent computer programs that is considered to be the start of the topic Artificial Intelligence. AI simulates human perception, learning and reasoning for solving complex problems(Krishnamoorthy & Rajeev, 1996). Due to the need for powerful knowledge base, ML is used as a technique for building and also updating knowledge base. Therefore, this study discusses the decision tree learning for financial categorization in credit analysis which provides a base model trained with labeled classes, with a rule-based Expert System(ES). The main advantage is that the system describes an implementation based on decision tree.

Chapter One: This chapter introduces discusses problem of the study, motivations and challenges, credit analysis, A historical background and literature review are also presented.

1.2. A Historical Background

Several attempts to simulate human intelligence before the invention of computer have been carried out. Since computer invention, computer scientists tried to make an intelligent machine(Partridge, 1998). In this study, a brief hository of AI is presented.

- In 1940s, AI was carried out by Warren McCulloch and Walter Pins. They submited the first research to be called a neural network with ability of learning. They also proposed an artificial neurons model. In 1943, Alan Turing proposed the Turing Test. In 1949, information theory invented by Claude Shannon described digital signals. These ideas converged and were spread widely in the early 50s(" ID3_algorithm," 2016; Russell & Norvig, 2010)
- In 1950s, Simon et al carried out experiments in creating intelligent programs for imitating processes of human thought. In the middle fifties, Marvin Minsky et al organized the Conference of Dartmouth. The suggestion of this conference contains the emphasis that *"every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it"*. At the conference, AI is considered to be the name for this field by Newell, Simon and McCarthy who obviously convinced the attendees for accepting(Wilks & Titanium,2016)(Luger, 2005). Marvin Minsky and Dean Edmonds, built the first neural network computer(Russell & Norvig, 2010).

- In 1960s, the Dartmouth conference was the starting of discovery era. Rosenblatt introduced neural network, showing that his learning algorithm could adjust the connection strengths of a perceptron to match any input data. At the beginning at the 1960s, success of Newell and Simon was developed with the General Problem Solver(GPS) that is considered to be the first program to incorporate the "thinking humanly" approach. Knowledge had to obtained by interviewing experts, who obtain it from textbooks. The domain knowledge importance was also obvious in understanding natural language(Russell & Norvig, 2010).
- In the 1970s, according to (Wilks & Titanium,2016), the ability of AI programs was restricted to some challenges such as:
    1. Computer power was limited
    2. Several problems can possibly be consumed in exponential time (in the size of the inputs). Optimal solutions of these problems demand unimaginable amounts of computer time.
    3. There are many important applications of AI such as vision or natural language which demands an enormous mass of information in the world. These programs need a large database and no one can build a large database. There was no one of AI researchers who knew how to create a program that can learn.
- In the 1980s, an intelligent application in AI called “ESs” was success. Firms all over the world focused on the knowledge in researches. The earliest researcher of ES is Edward Feigenbaum. In 1965, Dendral was developed to identify compounds from spectrometer. MYCIN was developed in 1972 for diagnosing infectious blood diseases. In 1980, XCON was a massive success. It was built at Carnegie Mellon University(CMU) for the Corporation of Digital Equipment. The expert knowledge is the ESs power that is contained in the ES. Despite the success of ESs in this period, there were some problems in ESs as they were too expensive for maintenance. They were complicated for

updating, along with disability of ESs for learning((Luger, 2005)). The problem of knowledge is encountered by creating a large database which would consist of all the facts that a person knows. The Bayesian network coined by Judea Pearl in 1988 was used on uncertain reasoning and ESs. Use of this approach allows the ability to learn from experience(Russell & Norvig, 2010). In the late 80s, appearance of a new approach to AI depended on robotics which was advocated by researchers. They think that, for providing real intelligence, a machine has to be a body, it could perceive, move, survive and deal with the world(Luger, 2005).

- During the 1990s, intelligent agents were accepted as a new paradigm. An intelligent agent is a program that realizes its environment and takes actions that increase its success chances(Luger, 2005). AI research is defined by the paradigm of the intelligent agent as *"the study of intelligent agents"*. One of the most important environments for intelligent agents is the Internet. Technologies underlie many Internet tools, such as search engines. The approaches were used during this period for the problem depending on expert-labeled examples along with learning algorithms(Russell & Norvig, 2010).
- During the 2000s, the focus was on solving the problem of knowledge acquisition that is considered to be the knowledge bottleneck for constructing the ES by many researchers such as Feigenbaum et al, which means the method used to extract the knowledge that is needed by a system. ML methods are utilized to solve that problem as long as enough data are avialable. Thus, the learning algorithms are treated as the major theme of Al researches proposed for various problems(Russell & Norvig, 2010).

1.3. Research Problem

The main process for credit management in the banks is to take credit decisions which is known as a credit rating. The credit rating is concerned with the collection, analysis, and classification of credit elements for assessing credit decisions. The value of credit in banks is the main factor of competition, survival and profitability. The credit rating for classifying the bank's customers is conducted individually by the decision-maker. The success of the judgmental process is based on experience and intuition of credit analyst as the process of assessing the credit to decrease the current and expected risks of customers being bad credit. Judgemental techniques are related to subjectivity, inconsistency and individual preferences(Abdou & Pointon, 2011). This thesis is concerned with helping bank decision makers in the process of credit approval.

1.4.Research Statement

The research objective of this study is to provide a method that allows us to understand financial credit analysis processes by applying Machine Learning techniques. The goal of having deeper knowledge about processes is to have a solid base to develop a system that efficiently supports and controls decision making processes for credit approval

Based on the research objective, the main question of the research is derived from:

- Is decision tree a good machine learning technquie for modelling financial credit analysis processes for firms assessment?

1.5.Aims Of The Study

- The main goal is to build an ES with knowledge base that would update knowledge by using ML techniques(ID3 algorithm and FP-Growth algorithm)
- Due to the fact that knowledge acquisition is time and effort consuming which would complicate the process of extracting the production rules by human experts, ML techniques are used in this study to build and update knowledge base.

1.6.Motivations And Challenges

According to this literature review, we conclude that using ML techinques for ESs makes the expert system more flexible to build and learn from new data.

Some features of decision tree as ML technique that are represented as follows: ease of understanding output of decision tree for people without analytical background where that output does not need any statistical knowledge for interpreting. Another benfit of decision tree is data exploration. In addition, influence of both outliers and missing values for decision tree is reasonably effective. Finally, the ability of decision tree for handling both nominal and numerical variables(Python), 2015).

ES and any other rule extraction techniques from different data along with ML techniques are welcomed to the credit scoring and banking industry because of the explicit conditions to accept / reject applicants, and it is easily understood by the business compared with different techniques(Sadatrasoul, Gholamian, Siami, & Hajimohammadi, 2013).

1.7.Literature Review

According to (Kaimal, Metkar, & Rakesh, 2014), Kaimal et al consider that the database is a part of rules framed by domain experts for the ES. They present an efficient algorithm that generates all significant rules based on the real data. They also consider the Learning Engine as an extension of ESs where new rules have to be learned by the shell for any domain. In their work, they use Association Based Learning methods for generating new rules from data. The new rules generated have to be validated by the expert and then can be updated to the knowledge base.

According to (Villena Román et al., 2011), in 2011, Villena-Román et al introduce a study which discusses using a ML in the ES “Hybrid Approach Combining ML and a Rule-Based ES for Text Categorization”. They also describe an implementation based on k-Nearest Neighbor and a simple rule language to express lists of positive, negative and relevant(multiword) terms appearing in the input text. It provides a base model trained with a labeled corpus, with a rule-based ES, which is used to improve the results provided by the previous classifier.

According to (Huang & Jensen, 1997), Huang and Jensen use a ML approach to automate building knowledge base for image analysis ESs. The method uses a learning algorithm to generate production rules from training data. The knowledge base built by the learning algorithm is used by an ES to perform a wetland classification using GIS(Graphic Image System) data. To evaluate the performance of the resultant knowledge base, the classification result is compared to classifications with two conventional methods.

According to (Rodrigues, 2015), Rodrigues R. introduces an ES for disbursement of home loans, which uses a goal driven approach along with knowledge Engineering(KE) in order to make a uniform, well informed decision. The system was verified and validated for its correctness and efficiency by running a series of tests and comparing the results to the decisions the human experts have taken in the past.

According to (Nosratabadi, Nadali, & Pourdarab, 2012), Nosratabadi H. et al design a Fuzzy ES based on the selected rules from Association Rules to specify the Credit Degree of banks' customers. This kind of research is carried out by classifying the bank's customers via Association Rules with the use of the APRIORI algorithm and considering the Experts' opinions to filter the obtained rules and define the Membership functions for the considered criteria,

According to (Baesens, Setiono, Mues, & Vanthienen, 2003), Baesens et al build credit-risk evaluation ESs using neural network rules extraction. Clarifying the neural network decisions by explanatory rules that capture the learned knowledge embedded in the networks can help the human experts in explaining why a particular decision is made. Hence, extracting rules from trained neural networks and representing these rules as a decision table may offer a viable and valuable alternative for building credit-risk evaluation ESs.

According to (Fekri-Ershad, Tajalizadeh, & Jafari, 2013), in this paper, a full automatic system is designed to use expert systems. This ES is included two main steps. The frist one, the human expert's knowledge is represented as decision trees. The second step is included an expert system which is assessed using extracted rules of these decision trees.

Based on the above literature, one may note that reseaches use different ML techniques for constructing knowledge base. Some researches use neural network with a decision table, Association Rules, k-Nearest Neighbor, decision trees, whereas others use convential methods such as the goal driven approach. This study uses decision trees are built by ID3 algorithm and FP-growth algorithm.

1.8.Credit Analysis

Credit analysis is a process adopted by some organizations such as banks and credit card firms for determining whether the credit applicant is to be granted a credit depending on known criteria that are previously defined(Hoseini, 2013). Credit analysis uses commonly financial statements, reflecting the several forms of debts that constitute the most part for the operation of a modern economy. Traders who exchange goods for promises of payment whose reliability needs assessement. Commercial banks that lend merchant to finance their inventories likewise need to count the probability of being repaid in full and on time. The banks have to be able in turn to demonstrate their creditworthiness to other financial institutions that are provided through the purchase of their certificates of deposits and bonds(Fridson & Alvarez, 2002). In these situations, financial statement analysis can considerably affect the decision of extending credit or not(Fridson & Alvarez, 2002; Khanbabaei & Alborzi, 2013).

> *"financial statements analysis is the application of analytical tools and techniques to general-purpose financial statements and related data to derive estimates and inferences useful in business analysis."* (Fridson & Alvarez, 2002)

The definition of the credit analysis in this study is limited only to Short-Trem Debt in the credit portfolio for private sector.

According to (Tugas, 2012), Financial ratios can be classified according to the information they provide:

1.8.1. Liquidity Ratios

Liquidity Ratios are important to determine the capability of the customer to pay their short term liabilities by assets that could be fast turned into cash (current assets). If capability of a firm for paying a short-term debt cannot guarnteed, it will not be able to preserve a paying ability of long-term debt. In this study, two Liquidity Ratios are used, namely: Current Ratio and Quick ratio(Tugas, 2012).

Table 1.1: Liquidity Ratios used in this study

Ratio’s name	**Calculation**	**Description**
Current Ratio	$\frac{(\text{Current Assets})}{\text{Current Liabilities}}$	This ratio is utilized to determine whether a company has an enough liquidity to pay its liabilities. A current ratio of 1:1 is deemed to be the minimum acceptable liquidity, while a ratio closer to 2:1 is a perfect acceptable liquidity(Bragg, 2002).
Quick Ratio	$\frac{\text{Current Assets} - \text{Inventory}}{\text{Current Liabilities}}$	Because of inventory presence in the current ratio, the current ratio may be a bad measure of the company's liquidity. The quick ratio is a new alternative, which excepts inventory from the current assets in current ratio. Thus, one can be able to get a

		better understanding of the company's ability in the short term to make cash from more liquid assets, like receivable accounts and securities(Bragg, 2002).

1.8.2. Activity Ratios

Activity Ratios measure the quality of a business receivables and how efficiently it uses and controls its assets. Some activity ratios shown in the table below are used in the study(Tugas, 2012).

Table 1.2: Turnover Ratios utilized in this study

Ratio's name	**Formula**	**Description**
Asset Turnover Rate(ATO)	$\frac{\text{Sales}}{\text{Total Assets}}$	A main question that is presented by the ATO is how many dollars of sales the firm produces for each dollar employed from assets. This ratio helps in the analysis process of the company's efficiency to deal with customers(Vance, 2002).
Gross Profit Margin (GPM)	$\frac{\text{Gross Profit}}{\text{Sales}}$ Where that $Gross\ profit = sales - \text{COGS}$ COGS stands for Cost Of Goods sold	According to (Fridson & Alvarez, 2002), gross margin is a main measure of performance. Interpreting changes in sales and cost of sales is helpful in determining major views of gross profit. Changes in gross profit are usually derived from one or a combination of these changes as

		follows: - Increase (decrease) in sales volume. - Increase (decrease) in unit selling price. - Increase(decrease) in cost per unit.
Cost of goods to Net Sales	$\frac{\text{COGS}}{\text{Net Sales}}$	sometimes in some cases, sales can not to cover sales expenses where that the margin gained from any sale can not cover the cost. This ratio is helpful to identify if the sales system has to result in a more inexpensive approach by a management(Bragg, 2002).
Selling, general, and administrative expenses (SG&A)	$\frac{\text{SG\&A}}{\text{Sales}}$	Generally speaking, SG&A can be fixed in the period of time, because these expenses include salaries and rent. There is a tendency for these expenses to increase, especially in prosperous times. When analyzing these expenses, our analysis should direct attention to both the trend in these expenses and the percentage of revenues they consume (Fridson & Alvarez, 2002).
Operating margin or Net Operating profit(NOP)	$\frac{\text{Operating income}}{\text{Sales}}$ Where that Operating income	Operating margin illustrates how well management has carried out its business by buying and selling

	$\text{Sales} - (\text{COGS} + \text{SG\&A})$	wisely, and controlling selling and administrative expenses (Fridson & Alvarez, 2002)
Working Capital	$\text{Working capital} = \text{Current assets} - \text{Current liabilit}$	it is used to evaluate the ability of a company to meet its short term obligations by using the working capital which provides warnings about approaching illiquidity. The need of creditors for knowing a financial amount of asset value that would be obtainable for paying off its claims (Fridson & Alvarez, 2002).

1.8.3. Profitability Ratios

According to (Tugas, 2012), Profitability Ratios are designed to assess the ability of a firm for generating earnings. The importance of profits for creditors are being a main source of funds to guarantee debt coverage. The following ratios are used in this study:

Table 1.3: Profitability Ratios employed in this study

Ratio's name	**Formula**	**Description**
Return of Equity (ROE)	$\frac{\text{Net Income}}{\text{Owners' Equity}} \times 100$	This Ratio is used by investors to identify the amount of return received from the investment of their capital in a company(Bragg, 2002).
Return of Sales (ROS) or Net profit margin	$\frac{Net\ Income}{Sales} \times 100$	Net margin measures a company's profitability relative to sales. Net margin is one of

		Profit margin measurments. It checks the relation between revenue and expenses. When expenses are less than sales, net profit margin will be high(Vance, 2002).
Return on assets (ROA)	$\frac{\text{Net Income}}{\text{Total Assets}} \times 100$	A company is deemed efficient by investors when the company is able to create an appropriate return while it uses the less amount of assets to do so. This also keeps investors to put more cash into the company and allows the company to turn its excess cash for investing in new investments. As a result, the return on assets measure is deemed a critical one for identifying the overall level of a company operating efficiency(Bragg, 2002).

1.8.4. Long Term Solvency Ratio

Long Term Solvency Ratio is also called as Capital Structure Ratio which is used to measure the ability of a company to gather its obligations; and identify how much assets are financed with debts for the company. They detect the cushion of equity that is used for absorbing any losses that may occur(Tugas, 2012).

Table 1.4: Long Term Solvency Ratio used in this study

Ratio's name	Formula	Description
Financial Leverage (FL)	$\frac{\text{Total Liabilities}}{\text{Total Assets}}$	The financial leverage indicator can be used to know whether a large debt proportion is in relation to equity which is to be used for funding a company's operations(Bragg, 2002).

1.9.Structure Of The Thesis

This section provides a brief of the thesis structure follow as

Chapter Two: this chapter presents introduction to ESs, components of the expert system and major roles of individuals who interact with the system, it also discusses The Need for Expert Systems, advantages and disadvantages of ESs, and Knowledge engineering process from interviewing with domain experts to definition of input and output data which will be used in this study to implement knowledge base.

Chapter Three: this chapter presents introduction to ML and illustrates its importance, and the classification of ML, terminology of Learning Task, goals of ML and machine learning techniques, and proposal solution.

Chapter Four: This chapter provides case study and an evaluation of the expert system with credit data and investigates the results.

Chapter Five: This chapter provides a conclusion of the expert system.

1.10. Summary

In chapter One, The major points are presented, firstly, the research problem is represented to find out a method to develop a model for making credit approval decision. The secondly one, it is represented to study the domain problem(credit analysis). The last one, it is the aims of the study which represented to develop the ES can learn by a dataset from the Centeral Wahada Bank. Consequently, banks can be able to develop their models by the expert sysem.

Chapter 2. Expert System

2.1.Introduction

In the early 1970s, the goals of AI researchers has been to develop computer programs to be intelligent programs, the researchers tried to make programs that can think and solve problems as human experts(Aronson, Liang, & Turban, 2005; Watson, 2008). In the late 1970s, successful applications of AI in ESs were transfered from research stage to the commericial environment. In this period, special AI languages were developed with ESs such as LISP and PROLOG. On the other hand, there were difficulty to develop ESs that required expensive hardware and complex AI languages. In 1980s, however, Personal Computer (PC) was introduced, making the development process of ESs by ES tools easy(Jones, 2008). Today with 50 years of AI researches, intelligent machines have become a truth, which can simulate humans(Sadatrasoul, Gholamian, Siami, & Hajimohammadi, 2013)

ESs are treated as a branch of AI. Since ESs are successfull applications of AI, they have been applied in several disciplines (about 70% of AI applications are ESs and the rest of AI applications use small percentages of other applications compared to ESs) as shown in **Figure 2.1**(Vizureanu, 2010)

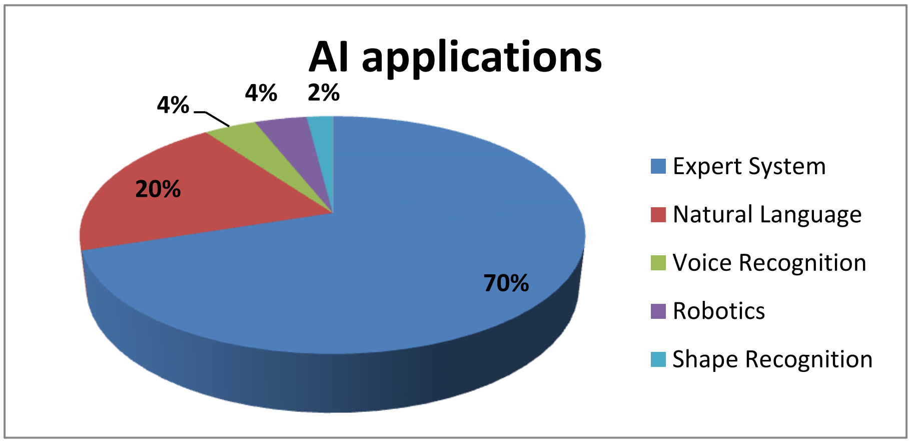


Figure 2: AI applications with percentages (Source : Vizureanu,2010)

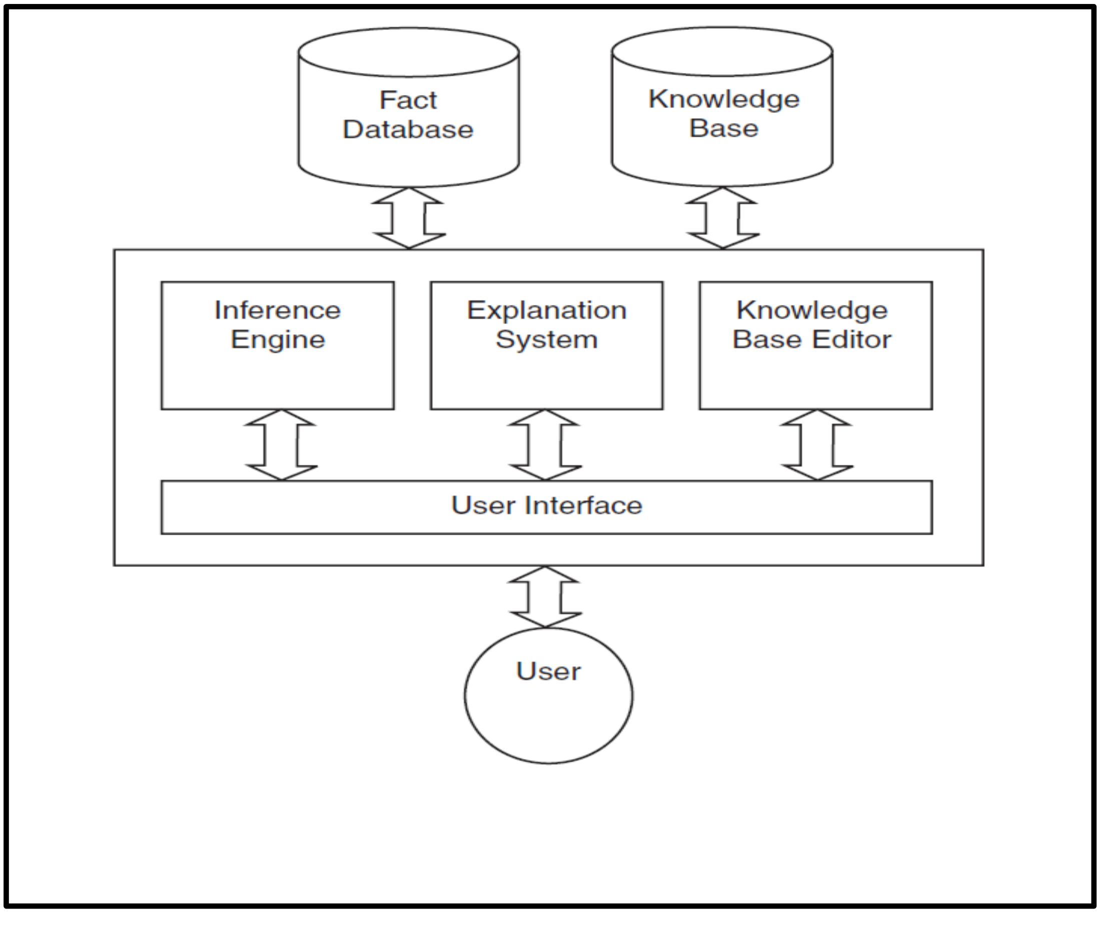
2.2. What Is Expert System

ES is an intelligent program, that is provided to make available some expertise of an expertise to non-expert(Siler & Buckley, 2005). ESs are similar to robots; they have a bit mechanical and simple minded, they also use domain-specific knowledge from human experts for handling a set of situations. They have presented their intelligence practically in dealing with frequent occurring suitations (Lucas & Van Der Gaag). ESs can be defined as:

> *"an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions."*(Krishnamoorthy & Rajeev, 1996)

2.3. Architecture Of An Expert System

According to (Coppin, 2004), The components of ES are the following items that are shown in **Figure 2.2**



2.3.1. Knowledge Base:

Figure 2: Architecture of an Expert System (Source : Coppin,2004)

Knowledge base contains domain knowledge for specific problems to draw conclusions, such knowledge is obtained from human experts(Coppin, 2004), (Negnevitsky, 2005).

2.3.2. Inference Engine

Inference engine is a part of an ES that uses the knowledge contained into knowledge base and facts that are contianed in database to draw conclusions(Coppin, 2004). It links the knowledge into knowledge base with facts provided by database(Negnevitsky, 2005).

2.3.3. Explanation System

It provides information to the user on how to arrive at inferred conclusions by using inference engine. This can be often necessary, especially if advice given is of a critical nature, for example, the case with a medical diagnosis systems. If conclusions of the system are derived by using faulty reasoning, then the user can be notified by examining the data provided by the explanation system(Coppin, 2004). Therefore, the ES has to be able to justify its conclusions(Negnevitsky, 2005).

2.3.4. Fact Database

It contains a specific case that will be used in a particular case to draw the conclusion data(Coppin, 2004).

2.3.5. User Interfaces

User interfaces of an ES provide access to the inference engine, the explanation system, and the knowledge base editor(Lucas & Van Der Gaag). It is considered as the means of communication between an user and the ES for seeking a solution for the problem(Negnevitsky, 2005).

2.3.6. Knowledge Base Editor

According to (Coppin, 2004), Knowledge Base Editor is used to update and provide knowledge. It is usually only available to knowledge engineers and human experts for updating the existing knowledge within the system, but it is not available to end users.

2.4.The Main Players Of The Expert System Development Team

According to (Negnevitsky, 2005), in general, there are five members of the development team of the ES, the domain expert, the knowledge engineer, the programmer, the project manager and the end-user. The success of the ES development team is based on how well the members work together. The main fundamental relationships in the development team are illustrated in **Figure 2.3.**

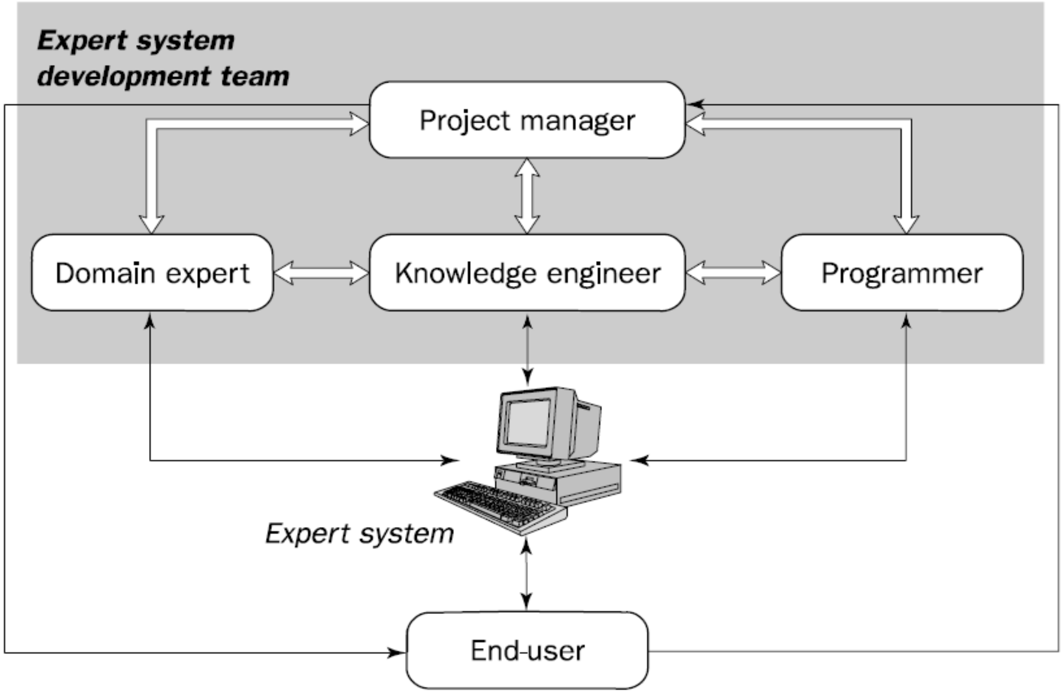


Figure 2: The main players of the expert system development team (Source : Negnevitsky,2005)

2.4.1. The Domain Expert

The domain expert is a person familiar with the skill and able to solve problems in a particular area or field. This person has more experience in a particular field. This experience has to be captured in an ES. Therefore, the knowledge engineer has to be able work with the domain expert or obtain his or her knowledge. The domain expert must be willing to collaborate in the development process of ES. The domain expert has a significant role in the ES development team.

2.4.2. The Knowledge Engineer

A knowledge engineer is a person who is able to design, build and test the ES. This person is responsible of choosing the suitable task for the ES. He or she interviews a domain expert to detect how to solve a specific problem. During interaction with the expert, the knowledge engineer determines what reasoning method is used by the expert for handling facts and rules. He is also responsible for choosing how to represent these fact and rules in the ES. Thus, the knowledge engineer uses programming languages to encode knowledge (and occasionally, it is encoded by him). Finally, the knowledge engineer is responsible for testing, reviewing and maintaining the ES.

2.4.3. The Programmer

A programmer is someone who is responsible of the programming by coding the domain knowledge in expressions which can be understood by the computer. If the shell can not be used, the programmer has to develop representation structures of knowledge base and databases, inference engine along with user Interface must be developed. The programmer can be involved in the testing process of ES.

2.4.4. The Project Manager

The project manager is the leader of the development team of an ES. He is responsible for keeping the project on track. He or she guarantees the completion of all the achievements and milestones. He must also be in touch with the domain expert, the knowledge engineer, the programmer and the end user.

2.4.5. The End-User

The end-user is known only as the user. When the ES is developed, he or she uses it. The final acceptance of the system depends on the user's satisfaction. The user not only has to have confidence in the performance of the ES, but also he must be comfortable using it. Therefore, the design stage of the user interface of the ES is spirited to the success of the project. Contribution of the end-user can be considered crucial.

2.5.Basic Properties Of The Expert System

The characteristics that must be presented in ESs as follows:

- Expertise: the ES is designed to accomplish at a human expert level in a narrow and specific field. Consequently, it is considered as the major characteristic of ESs for its high-quality performance(Negnevitsky, 2005).
- Ability to learn, actual intelligence requires the capability for learning which is represented in acquiring process of knowledge or skill(Siler & Buckley, 2005)
- Ability to symbolic reason from expertise, this feature allows for coding knowledge base in natural language (Cohn & Harris, 1992).
- Ability to explain conclusions, the ES must be able to explain its result that was made to the user(Cohn & Harris, 1992).

2.6. What Problems Can Be Solved By Expert Systems

According to (Aronson et al., 2005; Cohn & Harris, 1992), (Gupta & Singhal, 2013), ESs that can be categorized according to the problems they solve are as follows:

- Interpretation

- Prediction
- Diagnosis
- Monitoring
- Repair
- Design
- Control

2.7.Needs For Expert Systems

According to (Gupta & Singhal, 2013), ESs are concerned with difficulties related with conventional decision-making processes of human, these difficulties are presented below:

- Expertis of humans is not abundant.
- Humans' decisions are not consistent in day to day.
- Humans are sometimes unable to remember necessary details of a problem.
- Humans may be weary from material workload.
- Huamans possess rationed working memory. They are also slow for calling informantion from memeory.
- Humans can not understand a huge amount of data fast and they also can not save them in the memory.
- Humans can be subjective in their decisions.
- Humans can lie, hide or die.

2.8.Advantages Of Expert Systems

ESs play an important role in many disciplines such as finance, engineering, medicine etc, by shrinking the tasks time from days to hours, minutes, or seconds, and supporting decision makers(Aronson et al., 2005), Some advantages of ESs are as follows:

1. ESs can work more quickly comparing to humans(Aronson et al., 2005),(Gonciarz, 2014).
2. They can support frontline decision makers who have to take fast decisions when interacting with clients(Aronson et al., 2005).
3. ESs can increase the quality of ESs by offering consistent advice(Aronson et al., 2005), and decrease human errors chances (Ong, Xu, & Nee, 2008). They are concerned with details and do not discard related information and possible solution, thereby providing the same advise to frequent problems consistently(Aronson et al., 2005) while experts usually overlook some details(Ong et al., 2008).
4. ESs capture scarce expertise(Aronson et al., 2005), (Ong et al., 2008), when, in some situations, human experts are not available (Cohn & Harris, 1992). Thus, experience always exists(Gonciarz, 2014).
5. They assist in training inexperienced employees or novice members, for example, to operate complex equipments(Aronson et al., 2005), Therefore, they reduce the cost of training(Ong et al., 2008), (Gonciarz, 2014).
6. ESs can work in dangerous environments allowing humans to avoid such environments as a malfunctioned nuclear power plant(Cohn & Harris, 1992), (Gonciarz, 2014).
7. ESs can operate with incomplete, imprecise, uncertain data, information, or knowledge. Thus, they are still able to generate answers(Aronson et al., 2005).
8. ESs are able to explain complicated problems deemed to be beyond the ability of humans where they may require knowledge that exceeds individual ability(Ong et al., 2008). The expert can also be tired, unwilling or unable to explian such data at a given time(Aronson et al., 2005).

2.9.Disadvantages Of Expert Systems

1. ESs lack human common sense for making some decisions(Aronson et al., 2005), (Gonciarz, 2014; Ong et al., 2008). It means that they have only one logical reasoning and they connot use creative reasoning approach(Ong et al., 2008),(Long, Lawrey, & Ellis, 2016).
2. If there is an error in knowledge or limited knowledge or an error in reasoning process, the ES makes results that contain the error(Long et al., 2016). Therefore, the ES can make mistakes(Gonciarz, 2014).
3. They are expensive to produce, they also consume a great deal of time to construct knowledge base and they need the efforts of the expert and the knowledge engineer(Long et al., 2016).

2.10. Knowledge Engineering

Knowledge engineering (KE) is the acquisition process of knowledge about the domain problem for creating an ES. This knowledge is usually acquired from the human expert. This knowledge is also represented typically in the formula of knowledge heuristic (rules of thumb), which represent experience gained by the expert over a period of time(Anjaneyulu, 1998). Experts usually have unstructural inexplicit knowledge. The expert cooperates with the knowledge engineer for coding explicit rules. The knowledge engineering principal role is to assist domain experts to express what they know and document their knowledge in a usable form(Aronson et al., 2005). Knowledge acquisition is the bottleneck in the building process of ESs(Anjaneyulu, 1998), (Aronson et al., 2005; Miller, 1993).

According to (Aronson et al., 2005), the knowledge engineering process illustrated in **Figure 2.4**, consists of five main tasks are as follows

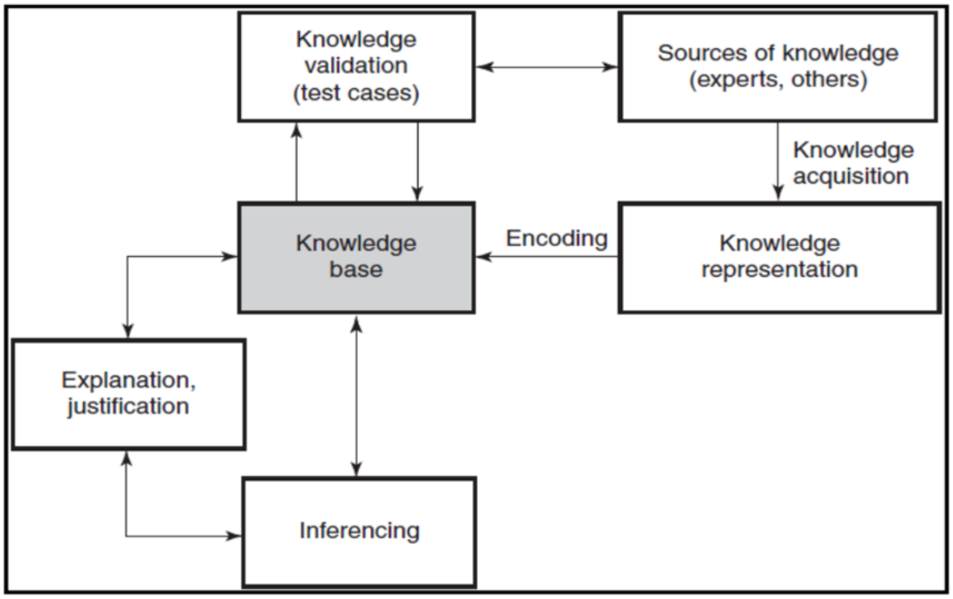

Knowledge validation (test cases)
Sources of knowledge (experts, others)
Knowledge acquisition
Knowledge base
Encoding
Knowledge representation
Explanation, justification
Inferencing

Figure 2: Process of Knowledge Engineering(Source: Turban, Aronson, and Liang,2005)

2.10.1. Knowledge Acquisition

The knowledge engineer can acquire knowledge from domain experts, books, videos, documents, and computer files. This knowledge can be about domain problems or procedures for solving problems(Aronson et al., 2005). There are three methods as follows:

2.10.1.1. Manual Methods

A manual method is mainly about an interview. The knowledge is extracted from the expert or some other sources by the knowledge engineer. The knowledge is then coded in the knowledge base following the process of inference. Note that manual methods are slow and expensive(Aronson et al., 2005).

2.10.1.2. Semiautomatic Methods

According to (Aronson et al., 2005), acquisition of knowledge can usually depend on computer based tools. Such tools allow knowledge engineers and experts to determine knowledge during an interactive process. Semiautomatic methods can be classified into two categories:

- Methods that support experts allowing them to build knowledge bases without help or with little help from knowledge engineers.
- The second methods are those which assist knowledge engineers by allowing them to carry out the necessary tasks in effective manner.

2.10.1.3. **Automatic Methods**

According to (Aronson et al., 2005), minimizing the role of each of the experts and knowledge engineers or even eliminating them. For example, induction method, which generates rules from a group of known examples can be used to construct a knowledge base. Presence of experts and knowledge engineers is almost non-existent. This illustrates the advantage of the automated method compared to other methods; it needs little contribution from experts and knowledge engineers. The main two adventages for using automated knowledge acquisition are as follows:

- It is cost efficient; there is no need to the knowledge engineers.
- The manual and semiautomatic methods are slow and expensive compared to automated methods.

2.10.2. Knowledge Representation

In AI, several problems are represented by search spaces. The search space is a representation of a collection of possible choices in a specific problem. Thus, one choice or more is a solution for this problem. Because the search space comprises a group of states, linked by paths that represent actions, it is also called state space, where the goal is to start with the world in one state and the world ends in the most desirable goals(Partridge, 1998). According to(Lucas & Van Der Gaag), There are three formalisms of knowledge-representation that still recieve much interest, namely: logic, production rules, semantic nets and frames. The semantic is used in this study.

2.10.2.1. **Semantic Nets**

A semantic net is described as a labeled directed graph. It comprises vertices and labeled edges between vertices, required to be acyclic. Every vertex in the graph depict concept, the edges of the graph, on the other hand, describe binary relationships between concepts(Lucas & Van Der Gaag). A semantic tree is considered as a sort of semantic net(Coppin, 2004).

2.10.2.1.1. **Semantic Trees**

A semantic tree is a sort of semantic net(Coppin, 2004). Trees comprise a known data structure. They are also non-linear data structure that store data in a hierarchical manner(McMillan, 2005). The main difference between a semantic net and a semantic tree is that, visually, a semantic net can have cycles, where a semantic tree can not. A cycle means that a path can visit the same vertices more than once in a semantic net(Coppin, 2004). The semantic tree has some characteritics as follows:

1. Each node has one complete predecessor (parent as represented in the node $Activity\,score$ in **Figure 2.5.**), unlike the semantic net where each node may have one predeccessor or more as illustrated in **Figure 2.6-a**. As shown in **Figure 2.5**, the node $Activity\,score$ represents the root node and it is linked by one edge to the node ATO where the node ATO comes below it in the tree. The nodes namely: No, and Yes, are linked immediately by using one edge to the node ATO(Coppin, 2004). On other words, the root node is the top node of the tree and it does not have a predecessor. The node can be connected to the other nodes in the lower level. The top node is referred to as the parent and the nodes in the lower level are known as the parent's children(McMillan, 2005).
2. Relationships between nodes in the semantic tree can be expressed as $succ(ATO) = Yes$ $pred\ (ATO) = \ Activity\,score$(Coppin, 2004), where these edges represent the relationships among nodes(McMillan, 2005). A semantic tree is known as a directed graph, which means that the nature of relationships is asymmetric. As for indirected graphs, there

is no difference between edge from $Activity\,score$ to ATO or edge from ATO to $Activity\ score$(Coppin, 2004).

3. The search process in the semantic tree starts with the root node. Since the root node is considered as the starting point of the problem. Some nodes are known as leaf nodes which do not have children (successors). In general, leaf nodes are referred to as goals nodes. Therefore, the search process is successful when reaching the search to a leaf node(Coppin, 2004).
4. The tree can be divided into levels. At level 0, there is only the root node. At the level 1, there are children of the root node, also the children of these nodes are at the level 2. The node at any level can be deemed as the root node of the subtree. Thus, it comprises children for that root node, and so on(McMillan, 2005).
5. A path is a route which represents the series of edges from one node to another in the semantic tree, it may sometimes consist of only one node tree, the length of the path is then equal to 0. The path's length is equal to 1, which means that the path consists of two nodes. When a path is leading from the root node to a goal node, it is referred to as a *complete path*. While a path which leads from the root vertex to a leaf vertex that is not goal node is referred to as a *partial path*(Coppin, 2004).

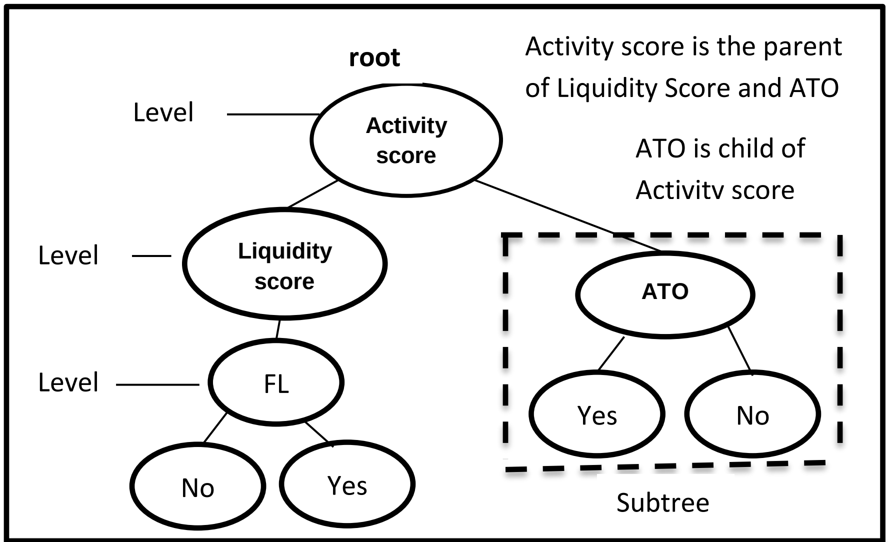

root
Activity score is the parent of Liquidity Score and ATO
Level
Activity score
ATO is child of Activitv score
Level
Liquidity score
ATO
Level
FL
Yes
No
No
Yes
Subtree

Figure 2: : A semantic tree

6. In the semantic tree, an edge which links two nodes is also referred to as a *branch*(Coppin, 2004). The path which consists of nodes deemed as goals and edges deemed as decisions(Aronson et al., 2005).

2.10.3. Inferencing.

As shown in **Figure 2.4**, the inferencing task is attended for designing a software which enables a computer to make reasoning relying on the saved knowledge and the particular problem(Aronson et al., 2005). Search process in a semantic net includes traversing systematically through the net or not systematically in some cases(Coppin, 2004). In general, identifying which nodes can be reached from a particular node is a common part on the semantic net. In semantic nets, the possible paths can be represented as a search tree. A path can be represented at every node in the tree. Note that there is no cyclical paths. A cyclic path does not lead to any solution through the net. It means that some branches in the search tree lead to leaf nodes, which are not considered as goal nodes in a semantic net as shown in **Figure 2.6-b.** (Coppin, 2004),(Russell & Norvig, 2010).

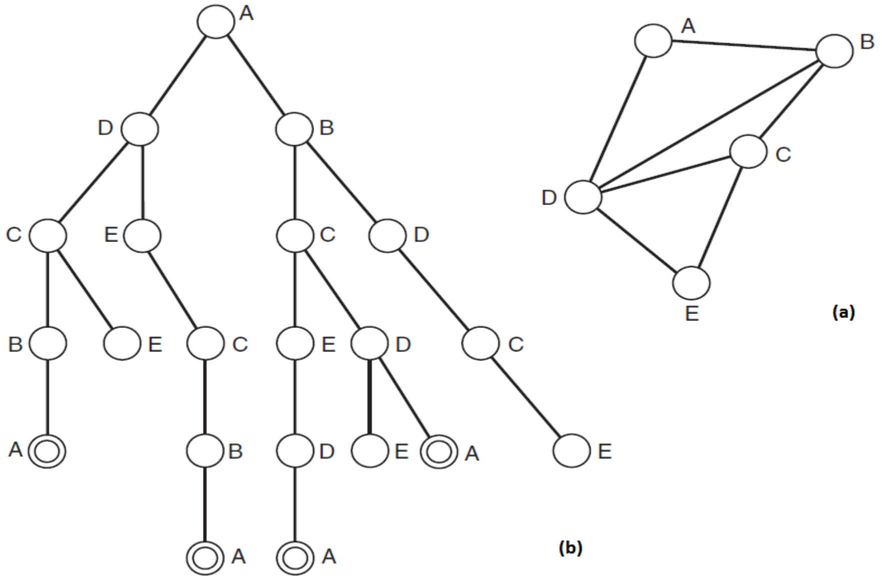


Figure 2: Figure 2.6-b presents A search tree representation for the semantic net in Figure (2.6-a.) (Source:Coppin,2004)

2.10.3.1. **Inference Trees**

An inference tree is also known as a goal tree or logical tree. It shows a schematic perspective of the inference process. It analogous to a decision tree. Every rule consists of an antecedent and a consequence. In inference trees or decision trees, the antecedents and the consequences represent nodes and the branches link the antecedents, AND and OR operators are to mirror rules structures. The major feature of inference trees is that they provide a perfect insight of the rules structure(Aronson et al., 2005).

According to (Aronson et al., 2005),when reasoning in the inference tree, the inference process moves forward its branches which is called tree traversal. Generally, Inference trees are a compost of AND nodes and OR nodes; where they are referred to as AND/OR trees. Inference trees essentially include a collection of goals. Every goal can comprise subgoals (children) and a supergoal (parent). It can be traversed through an inference tree by two ways as follows:

- To traverse an AND node, meaning that it has to traverse all the nodes below it. In other words, a goal is fired only when all its actual subgoals are fired.
- To traverse an OR node, it must traverse only one of the nodes below it. It means that a goal is fired when any of its current goals is fired.

The inference tree is built upside-down from the root at the top, and the branches head for the bottom. When adequate subgoals are fired for preforming the main goal, then the tree is satisfied(Aronson et al., 2005).

2.10.4. Explanation And Justification.

This stage is concrened with desiging and developing the explanation subsystem. In decision trees, the main feature of decision trees explanation process is the ablity to provide guidelines for answering the why and how questions in the explanation stage. The computer pursues the logic in the decision tree(Aronson et al., 2005).

2.10.5. Knowledge Validation.

The knowledge validation task is concerned with validating and verifying the knowledge by testing cases till its quality is reasonable. Tested results are presented to human experts for verifying the accuracy of the system(Aronson et al., 2005).

2.11. Summary

The chapter two introduces the archicture of the ES is used in this study. It acquires knowledge by the Automatic Method. The Semantic tree is utilized as the Knowledge representation method. The major feartures of the ES are that

- it provides a perfect insight of the rule structure.
- It provides guidelines for answering why and how questions in the explanation stage.

Chapter 3. Machine Learning

3.1.Introduction

Learning is one of the most important features for ESs(Siler & Buckley, 2005). Machine Learning (ML) can be helpful by producing rules, instead of using handcrafted rules, which are implemented well in ESs(Ong, Xu, & Nee, 2008). Automated learning or ML is used to program a machine (computer) for learning from input. Learning is the conversion process of experience into knowledge. The input of the learning algorithm is called training data that represent the experience. The output of learning algorithms is knowledge(Shalev-Shwartz & Ben-David, 2014). The classification of ML, its importance, terminology of Learning Task, goals of ML and brief account on ML techniques are all dicussed in this chapter.

3.2.Machine Learning

When solving a problem on a computer, we use a series of instructions that have to be executed for converting input to output. In contrast, ML programs can automatically learn from data using ML algorithms(Domingos, 2012). To overcome knowledge acquisition bottleneck problem by using ML as a technique responding to awkwardness of coding knowledge which became increasing with time(Carbonell, Michalski, & Mitchell, 1983). Mark defines ML as

> *"Machine learning is the study of algorithms that automatically improve their performance with experience."*(M. A. Hall, 1999)

To illustrate the role of ML to overcome knowledge acquisition problem, **Figure 3.1.** shows two methods for acquiring the knowledge for building a knowledge base.

Figure 3.1-a. shows that the traditional method of acquiring knowledge in a computer-usable format to construct a knowledge base involves human domain experts and knowledge engineers. The domain expert explicitly expresses his or her knowledge about a subject in a language that can be understood by the knowledge engineer. The knowledge engineer translates the domain knowledge into a computer-usable format and stores it in the knowledge base(Huang & Jensen,1997).

While **Figure 3.1-b.** illustrates how to depend on the learning program. The system feeds the learning algorithm good training data and extracts knowledge (knowledge base). This knowledge is often stored in a computer-usable format that's readily useable by a machine(Harrington, 2012).

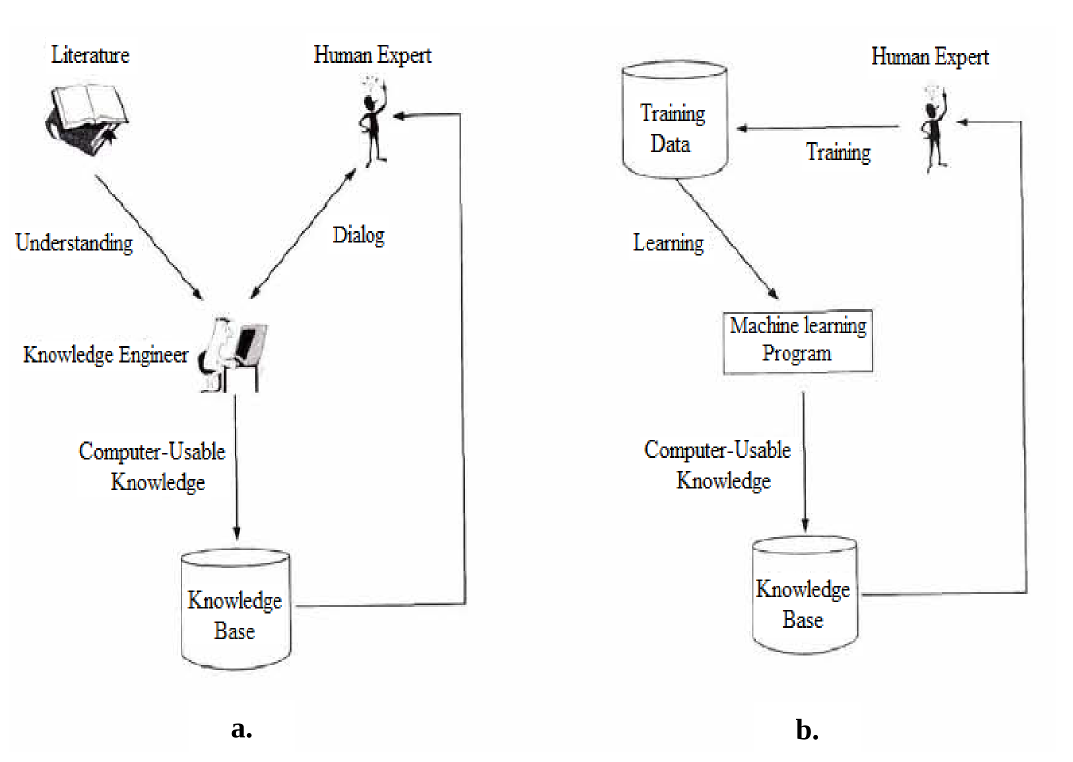


Figure 3.1a / 3.1b

According to (Huang & Jensen,1997), Figure (3.1-a.) The difference between the traditional method of knowledge acquisition used to construct a knowledge base. (3.1-b.) a ML

The purpose of this study is to use ML techinque in the ES for obtaining knowledge correctly, producing correct outputs, discovering new knowledge and adapting with these changes in the environment(Smith, McGuire, Huang, & Yang, 2006)

3.3.Importance Of Machine Learning

According to (Nilsson, 1996), there are many tasks which make ML important as follows:

1. Some tasks can only be defined well by examples. We cannot be able to extract cohesive concept description correctly because it is hard to find out a concise relationship between inputs and outputs. Machines are able to extract these descriptions to produce correct outputs from a large number of examples by inferring their input/ output function that describes the relationship between them.
2. Some tasks contain a large amount of knowledge, consequently, these tasks are too large for explict encoding by human. This knowledge may be more captured by machines that can acquire knowledge gradually than human.
3. Due to continuous changes in the environment over time. Machines can adjust with such changes. As a result, it decreases the need of constant redesign.
4. In tasks, there is new knowledge discovered by human. Due to continuous development in this knowledge produced in the world, thereby continuing redesign of intelligent sysems for adaping to new knowledge is not imparctical. ML methods may be able to track new knowledge.

3.4.Goals Of Machine Learning

According to (Carbonell et al., 1983), the main goals of ML are summarized as follows:

1. ML is concerned with developing and analyzing learning systems that learn from predetermined data set, it is also concerned with engineering approach of knowledge.
2. ML enables a computer to mimic the learning processes of human.
3. ML is intented to study and to provide theoretical analysis of the scope of learning techniques to discover possible learning methods.

3.5.Terminology Of Learning Task

This section presents summary of Learning Task terminology used in ML as follows:

1. ML techniques deal with a single data table called database, dataset, or set of instances. The term of database in ML indicates a set of instances or training examples which are stored in a single file(Frawley, Piatetsky-Shapiro, & Matheus, 1992), a single relation, or a flat file(M. Hall, Witten, & Frank, 2011).
2. A set of instances is the input data to ML algorithm. The instances or training examples are objects that have been classified or associated or clustered according to the learning algorithms used(M. Hall et al., 2011). A record or tuple in a data table represents instances, also known as a feature vector, or an example (training example)[(Frawley et al., 1992), (Fürnkranz, Gamberger, & Lavrač, 2012). The term instances is more commonly used to express input(M. Hall et al., 2011).
3. Each dataset is designated as a matrix of instances versus attributes(M. Hall et al., 2011). Attributes are also known as features in the ML(Frawley et al., 1992). The values of a set of predetermined features characterize instances. Therefore, these instances are defined or analyzed by features(M. Hall et al., 2011).
4. Each instance belongs to the concept(class)(Fürnkranz et al., 2012).
5. Learning algorithm takes a set of instances as input and returns the statement which represents the outcomes of learning as

output(Fürnkranz et al., 2012). Input of the learning algorithm includes the structure of instances and features and concepts, where they are used to learn a concept description(M. Hall et al., 2011).

6. The aim of Learning algorithm is to generate a model for the complete dataset or to detect patterns that attain or catch some part of data set(Fürnkranz et al., 2012).

3.6.Classification Of Machine Learning

To classify different ML algorithms(Huang & Jensen,1997). There are two major methods(Fürnkranz et al., 2012).

3.6.1. Supervised Learning

Supervised learning is known as classification or inductive learning. It is equivalent to human learning from past experiences to acquire knowledge for improving capability of humans to achieve tasks in the real world. While machines do not possess experiences, the machine can learn from historical data that are considered as past experiences(Liu, 2011). The historical data represents a set of training examples with labeled classes(class attributes). The ML algorithms are based on this training data set to generate a model for classifying unseen examples. This is also known as learning from training examples(Marsland, 2015). The main supervised learning algorithms are namely: nearest neighbor, Naive Bayes classifier, support vector machines(SVMs) and Classification trees(Matthiesen, 2010). This study uses decision trees.

3.6.2. Unsupervised Learning

Unsupervised learning or clustering is the main subject in ML known as "class discovery," which is used when there is no class to be predicted(Matthiesen, 2010). Unsupervised learning is a methed to discover these structrues when a set of training examples are without class attributes(Liu, 2011). When labeled classes are not provided, the learning algorithm attempts to find out similarities between input data where such data that have something in common are classified or grouped with each other. The most known one of unsupervised learning is association rules(Marsland, 2015)

3.7.Common Matters For Supervised Learning Algorithms

According to(Kotsiantis, Zaharakis, & Pintelas, 2006), the first phase is to collect data. If a human expert is available, then s/he can propose which attributes are the most informative. If not, it means that measuring everything available in the aspiration that the right informative and relevant attributes can be isolated. However, the data set that has been collected through this way is not suitable for direct induction. It contains a dataset with noise and missing values, thus, it requires significant pre-processing. Impossible values must be checked by the data processing program, actually, at the entered point. Consequently, it can be re-entered. Data cleaning of variables is a filtering approach to values. Relying on the circumstances, several researches used many methods to process missing data. But in this study, the human expert is avaliable, he proposes which attributes are the most informative and also handle missing ones.

3.8.Machine Learning Techniques

There is no a single classification approach that can be the best method for all classification problems(Matthiesen, 2010). Here is a brief for some ML techinques that are related to classification problems as follows.

3.8.1. Nearest Neighbor

It is one of the simplest classifiers that saves the whole training examples and achieves classification only when the tested features match one of the examples completely. An explicit weakness of this technique is the absence of a testing set of unlabeled examples. The reason that these tested records do not completely match any of unseen examples(Alpaydin, 2010). However, researchers have illustrated that the accuracy of classification of Nearest Neighbor may be very strong. The drawback of nearest nighbor is that the classification model is not built. Therefore, each test instance is compared to each training example at a time.

Thus, this algorithms is slow at the classification time when the training data set and testing data are huge(Liu, 2011).

3.8.2. Support Vector Machines(SVMs)

Support Vector Machines (SVMs) are the newest ML technology under supervised learning. SVMs revolves around the idea of a "margin" where both sides a hyperplane that separates the two datasets with labeled classes (Huang, Chen, & Wang, 2007). It selects a small number of critical instances as the boundary which are called support vectors from each class, it constructs a linear discriminant function that splits each class as widely as possible. Hence, SVM is an algorithm that finds a special type for linear model, the maximum margin hyperplane. The maximum margin hyperplane awards the maximum partitions between the decision classes. The training examples that are closest to the far maximum margin hyperplane are support vectors(Aktan, 2011). SVM may not be able to find out any splitting hyperplane at all because the data includes misclassified instance(Kotsiantis et al., 2006).

3.8.3. Naïve Bayes Classifier

Naïve Bayes classifier is that classifier which can be studied surely from the probabilistic perspective of view. The classification task can be considered as estimating the class posterior probabilities(Liu, 2011). Naïve Bayes has a strong presumption that all the variables are independent for the classification in network. It is also easy to build. The process of classification is very efficient, since it is assumed that all the features independent of each other(Marsland, 2015).

3.8.4. Classification Trees

Decision tree is the implementation of divide and conquer strategy to a set of independent instances to learn the problem. Decision tree is composed of root, internal decision nodes and terminal leaves. Each node in decision nodes represents a test of a particular attribute in the instance set to be classified(Aktan, 2011). Decision tree learning looks for relationships between attributes(Huang & Jensen,1997). The consequence of test node represents edges or branches so each edge denotes the test value that the node can use. The process of building decision trees starts at a root and is iterated recursively until a leaf node is catched or accessed. Therefore, examples are classified according to class assigned to the leaf(Aktan, 2011).

Because of its simplicity, rapidity of classifying unlabeled instances, and intuitive graphical representation, decision trees is one of the most widely utilized and popular classification techniques. They can be readily verified and understood by the domain experts(Matthiesen, 2010). The divide-and-conquer approach to decision tree induction is also called Top-Down Induction of Decision Trees (TDIDT) that was sophisticated and improved by J. Ross Quinlan. This approach has been depicted using an information theory(M. Hall et al., 2011). The ID3 shorts for “Iterative Dichotomiser 3” is a decision tree algorithm(Harrington, 2012; Shalev-Shwartz & Ben-David, 2014).

Association rules are similar to classification rules. The jargon comes from market basket analysis. It is to find out associations among items that can be purchased together(M. Hall et al., 2011). Frequent Pattern growth (FP-growth) is concerned to explore frequent itemsets among sets of things that generally happen with each other. The FP-growth algorithm scans the database only twice; the first scan for constructing the FP-tree, and the second for extracting frequent itemsets from the FP-tree(Harrington, 2012). FP-growth which implements a divide-and-conquer strategy compresses the database which consists of frequent items into a frequent pattern tree, thus, the frequent pattern tree maintains the association information of itemset. This is called frequent pattern growth(Han, Pei, & Kamber, 2011).

3.9.Proposed Solution

Step 1: Studying the domain problem and interviewing a domain expert to understand procedural knowledge, then, selecting the observation set: Select the firms with credit approval and firms with credit reject.

Step 2: Identification of candidate financial ratios: The most common dimensions considered in financial performance evaluation are liquidity, activity, profitability and solvency.

Step 3: Selecting final financial ratios: indicators are based on the expert opinion. The resulting set of indicators contains the most relevant decision making dimensions. Thus, we can form the data model which will be used for building knowledge base using ML techniques.

Step 4: Selecting a software tool for building ES in credit analysis using the ML.

Step 5: Validation and testing the ES: After knowledge base is constructed, it needs to be evaluated for performance accuracy by using test data and the human expert.

These steps are carried out iteratively until an accurate ES is established.

The following **Figure 3.9** illustrates how the ES can acquire knowledge and learn from new data.

User Interface: it deals with a user and interacts between the user and the system.

Knowledge Base: is constructed by the decision tree algorithm using the training data as input to the algorithm and rules as output of the algorithm

Inference Engine The main objective of the inference engine is inferring the data from the knowledge base by using different rules, which is built by the ML techinquies such as *decision tree algorithm*, and interacts with working memeory, here we have two options as follows:

- There is a solution(decision) for a specific problem and the system makes a decision to agree or disgree, which means that knowledge base contains a rule which classifies the specific problem.
- There is no solution for a specific problem, *the clustring algorithm* in KE searches for a solution for new data (unlabeled data) from training data. Thus new data is the labeled data, which will be inserted to the training data.

3.10. Summary

The chapter Three introduces the learning process. It also clarifies the learning terminologies that are used in the ML where describing attributes, instances and table, etc. the classificaction of ML are namely: supervised learning and unsupervised learning such as ID3 algorithm and FP-growth algorithm respectively. It also presents ML techniques generally.

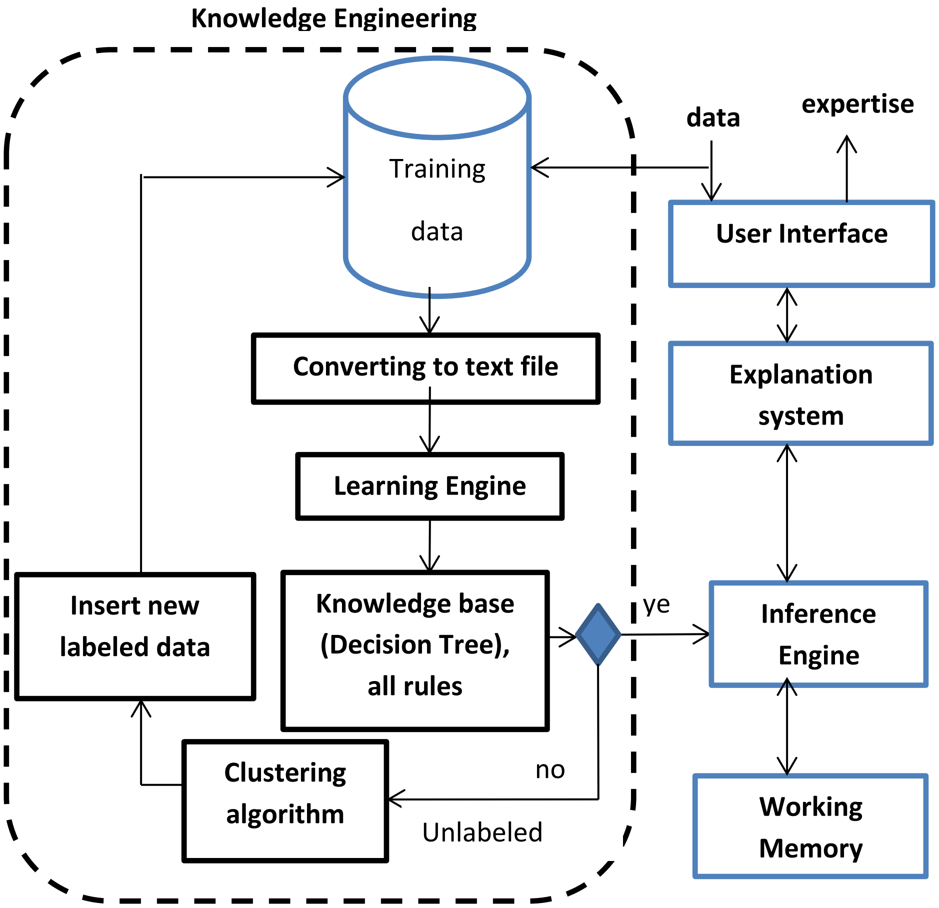


Figure 3.2 : Proposed model for building knowledge and relearning the model from new data

Chapter 4. Case Study

4.1.Introduction

This section presents architecture of the ES discussed in the chapter 2. The ES in credit analysis consists of modules for balance sheet and income and the module used to calculate financial ratios. It also contains the class for ID3 algorithm and the class for FP–growth algorithm that extract decision rules from the database as training data, such decision rules represent the knowledge base of ES in credit analysis. Two classes are used to implement decision tree and FP-tree. By using the directed graph (Digraph) to implement the decision tree as the knowledge. It is useful to explain how and why decision making is created.

The most important part for any ES is the knowledge base. As mentioned in chapter 2, knowledge engineering is knowledge acquision process to build the ES. It starts with acquiring knowledge from domain experts and books, etc. The next step is to represent this knowledge by one of the knowledge representation techniques. Knowledge validation is then concerned to test the knowledge by the domain expert. This study uses automatic methods for knowledge acquision. ID3 algorithm is used to automate the knowledge base and FP-growth algorithm is used to classify new cases. The frist section deals with the implementation of knowledge base as the graph. The second section shows the execution of the ES for entering finanacial data, requesting the credit facility, explanation of the results or the decision. The last one is the knowledge Editor which is concerned with constructing and updating the knowledge base and testing the ES.

4.2.Data Collection and analysis

Credit rating is an important process in Banks. The goal of this study is to sumbit a suitable advice or consulte to credit analyst to avoid or prevent credit risks in credit analysis of Firms by using finanacial ratios and depending on the domain expert. The real input data used for this system is a list of firms collected from the credit department in Central Wahda Bank. Such data are related to the period from “$1998 - 2015$”.

In order to desigin and construct the ES in credit analysis using ML technigues, depending on the empirical work, the study suggests a proposed model that consists of stages used to develop the ES. These stages are:

Stage 1: Studying the domain problem which is represented as “$credit\ approval$” in this study and interveiwing the credit analyst as a domain expert to comprehend procedural knowledge for realizing how to evluate firms. It is also based on textbooks, documents, vedios and computer files. At the beginning, the process of selecting and collecting a sample of observation set with final decisions by a researcher if possible was carried out since it is considered a good approach for the best understanding of the whole domain problem.

Stage 2: Identification of candidate financial ratios: at the beginning, the study uses four dimensions, nemaly liquidity, activity, profitability and solvency, where the credit analyst suggests 28 financial ratios that are divided into 3 profitability ratios and 12 activity ratios, 2 liquidity ratios,and one solvency ratio.

Stage 3: Selecting final financial ratios are based on the credit analyst’s opinion. The resulting set of financial indicators used in this study are presented in chapter 1, these ratios with an interval (a range of information) are presented below in **Table 4.1.** Thus, some financial ratios are ignored or discarded by the domain expert because all the values for these financial ratios are inordinated values. This study is based on credit analysis in the evaluation process of companies depending on finanicial ratios and their scores according to the domian expert “credit analyst”. The evaluation process is divided into two phases. The first phase is concerned with evaluating each financial ratio based on the average or a range of information for the values of each ratio. The $average$ is the statistical scale used in determining acceptable/unacceptable for some financial ratios and other ratios are based on the information range as shown in **Table 4.1.**

Table 4.1: The process of evaluation each financial ratios by the empirical work

Name of The ratio	Range	Average	Domain
ROE	$(9-40)$	12.23	$f(x)=\begin{cases} if\ 9 \leq x < 12.23 & 0 \\ if\ 12.23 \leq x \leq 40 & 1 \\ Otherwise; & 0 \end{cases}$
ROS	$(1-20)$	9.847	$f(x)=\begin{cases} if\ 1 \leq x < 9.85 & 0 \\ if\ 9.85 \leq x \leq 20 & 1 \\ Otherwise; & 0 \end{cases}$
ROA	$(1-10)$	3.663	$f(x)=\begin{cases} if\ 1 \leq x < 3.66 & 0 \\ if\ 3.66 \leq x \leq 10 & 1 \\ Otherwise; & 0 \end{cases}$
FL	$(1-5)$	1.6	$f(x)=\begin{cases} if\ 1 \leq x \leq 1.66 & 1 \\ if\ 1.66 < x \leq 10 & 0 \\ Otherwise; & 0 \end{cases}$
ATO	$(0-1.66)$	0.718	$f(x)=\begin{cases} if\ 0 \leq x < O.72 & 0 \\ if\ 0.72 \leq x \leq 1.6 & 1 \\ Otherwise; & 0 \end{cases}$
$COG/Sales$	$(10-70)$		$f(x)=\begin{cases} if\ 10 \leq x \leq 70 & 1 \\ Otherwise; & 0 \end{cases}$
GPM	$(30-90)$		$f(x)=\begin{cases} if\ 30 \leq x \leq 90 & 1 \\ Otherwise; & 0 \end{cases}$
$SGA/Sales$	$(10-45)$		$f(x)=\begin{cases} if\ 10 \leq x \leq 40 & 1 \\ Otherwise; & 0 \end{cases}$
$NOP/Sales$	$(10-40)$		$f(x)=\begin{cases} if\ 10 \leq x \leq 40 & 1 \\ Otherwise; & 0 \end{cases}$
$WI/Sales$	$(30-50)$		$f(x)=\begin{cases} if\ 30 \leq x \leq 50 & 1 \\ Otherwise; & 0 \end{cases}$
$Current\ Ratio$	$(1.1-2.9)$		$f(x)=\begin{cases} if\ 1.1 \leq x \leq 2.9 & 1 \\ Otherwise; & 0 \end{cases}$
$Quick\ Ratio$	$(0.9-1.1)$		$f(x)=\begin{cases} if\ 0.9 \leq x \leq 1.1 & 1 \\ Otherwise; & 0 \end{cases}$

Table 4.1. illustrates the process of evaluation each financial ratio as a mathematical expression (equation) $y=f(x)$ to determine their evaluation or scores, where each $range$ contains values of the financial ratios represented as x in the equation $y=f(x)$. For example, the range of ROE values is $(9-40)$.

Thus, by using the equations, an assessement process of each ratio is calculated where its score is equal to 0 or 1, which represents $domain\,(0,1)$. Notic that 0 represents bad and, 1 represents good.

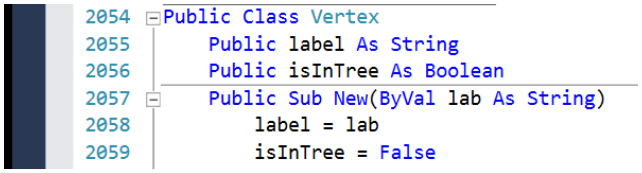
The final evaluation of companies is estabished on the evaluations of some financial ratios and scores of some dimensions according to the domain expert. Thus the data model was created and collected based on the domain expert to be used in the building process of knowledge base by ML techniques.

Stage 4: Selecting a software tool for building ES: The ES has been applied by using "$visual\ basic\ community\ 2013$" programming language. The system feeds the balance sheet and income statements of firms with all necessary data. It then calculates the financial ratios to their assessement and takes final decision by the knowledge which was built by ML techniques as a shell with our ES.

4.3. Implementation Of The Knowledge Base Of The ES

To implement the knowledge base as a tree by using the directed graph, as mentioned in chapter 3. The graph contians a collection of nodes and a collections of edges(McMillan, 2005). One thing is to represent vertices by a vertex class which contains two elements. The first element is to recognise the node in that vertex class to be labeled "$label$". The second element of the vertex class is $isInTree$ used to retain a visited path of the node in the graph as shown in **Figure 4.1.**

The method needs in the vertex class is the constractor to construct $label$ and $isInTree$. The data structure used to keep a list of nodes is an array. The index of the array is used as a reference to refer for their position in the graph class. In addition, credit approval is represented by 1 and credit disapproval is represented by 0 instead of yes and no. The next method is to represent the edges of the gragh referred to as an adjecency matrix. The adjecency matrix is two-dimensional array denoting whether an edge is being between two nodes, where if there is no edge between two nodes, an $inconnect$ is put in that position, otherwise, a weight is set to express the edge between the two nodes as shown in **Figure 4.2.** Note that *inco* is shortcut for *inconnect*.



```
Public Class Vertex
    Public label As String
    Public isInTree As Boolean
    Public Sub New(ByVal lab As String)
        label = lab
        isInTree = False
```

Figure 4.1: Class Vertex and Constructor Method.

List of Nodes

0	FL
1	ATO
2	Active Score
3	Liquidity score
4	yes
5	no

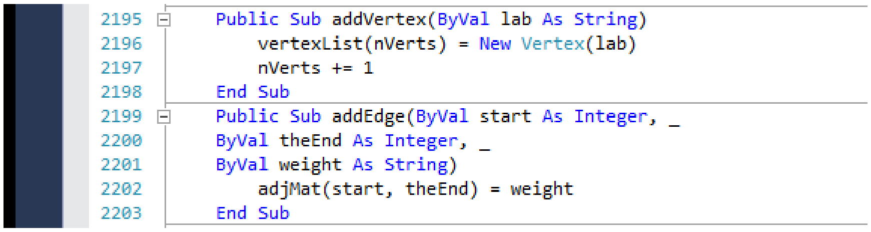
Adjacency Matrix

	0	1	2	3	4	5
0	*inco*	1	*inco*	*inco*	*inco*	0
1	*inco*	*inco*	*inco*	*inco*	1	0
2	*inco*	*inco*	*inco*	4	*inco*	*inco*
3	1	*inco*	*inco*	*inco*	*inco*	*inco*
4	*inco*	*inco*	*inco*	*inco*	*inco*	*inco*
5	*inco*	*inco*	*inco*	*inco*	*inco*	*inco*

Figure 4.2: A list of nodes and adjacency matrix

Thus, the process for inserting a vertex and an edge can be performed in the graph by using this codes.

The *addVertex* procedure requires a string value as argument for *lab*. After that, a new vertex is added to the vertex array. The *addEdge* method recieves three values, two integer values that denote positions of vertices and one string value for the weight attribute as persented in **Figure 4.3.**



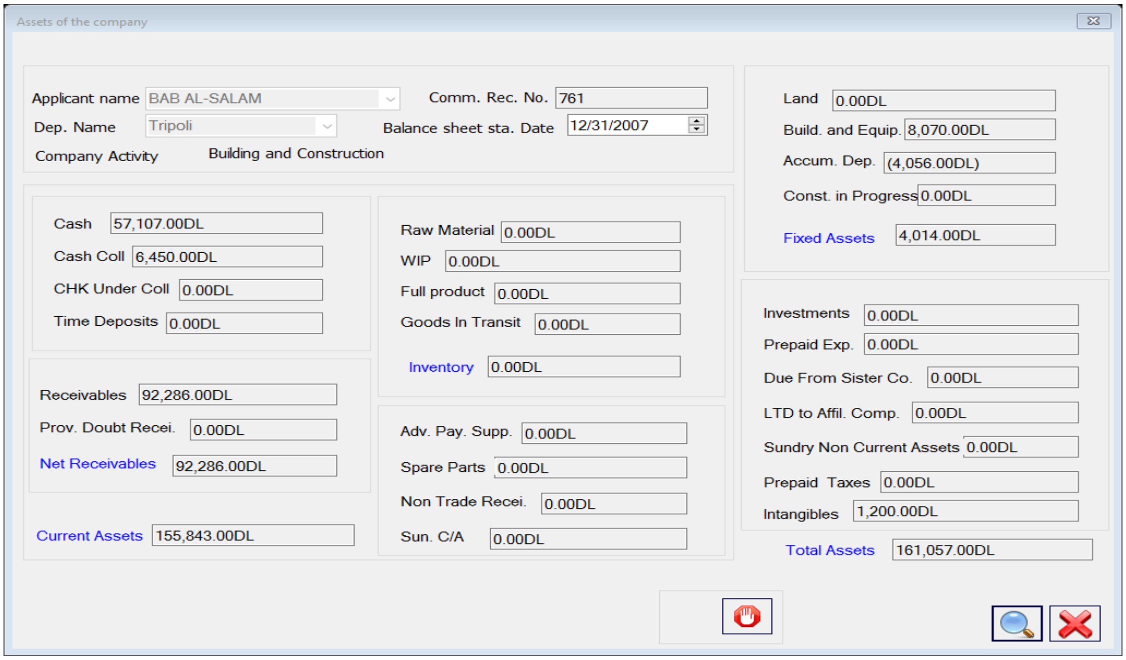
```
    Public Sub addVertex(ByVal lab As String)
        vertexList(nVerts) = New Vertex(lab)
        nVerts += 1
    End Sub
    Public Sub addEdge(ByVal start As Integer, _
    ByVal theEnd As Integer, _
    ByVal weight As String)
        adjMat(start, theEnd) = weight
    End Sub
```

Figure 4.4: Add the vertex to a list of vertices and the edge to a matrix.

4.4.Design Of User Interfaces

4.4.1. Design Of User Interface Of Financial Input Data

In this application, there are three user interfaces for entering the financial data as input present in **Figures 4.4**, **4.5** and **4.6**. **Figure 4.4.** presents user interfaces of entering the balance sheet data espiacially in assets for loan applicant (e.g. in this study is a firm). The balance sheet is divided into two parts, namely: assets and liabilities in **Figure 4.4**. and **Figure 4.5**. respectively. A company as the loan applicant must be inserted into one of the four databases before entering the financial data.



Assets of the company

Applicant name BAB AL-SALAM
Comm. Rec. No. 761
Dep. Name Tripoli
Balance sheet sta. Date 12/31/2007
Company Activity Building and Construction

Cash 57,107.00DL
Cash Coll 6,450.00DL
CHK Under Coll 0.00DL
Time Deposits 0.00DL

Receivables 92,286.00DL
Prov. Doubt Recei. 0.00DL
Net Receivables 92,286.00DL

Current Assets 155,843.00DL

Raw Material 0.00DL
WIP 0.00DL
Full product 0.00DL
Goods In Transit 0.00DL
Inventory 0.00DL

Adv. Pay. Supp. 0.00DL
Spare Parts 0.00DL
Non Trade Recei. 0.00DL
Sun. C/A 0.00DL

Land 0.00DL
Build. and Equip. 8,070.00DL
Accum. Dep. (4,056.00DL)
Const. in Progress 0.00DL
Fixed Assets 4,014.00DL

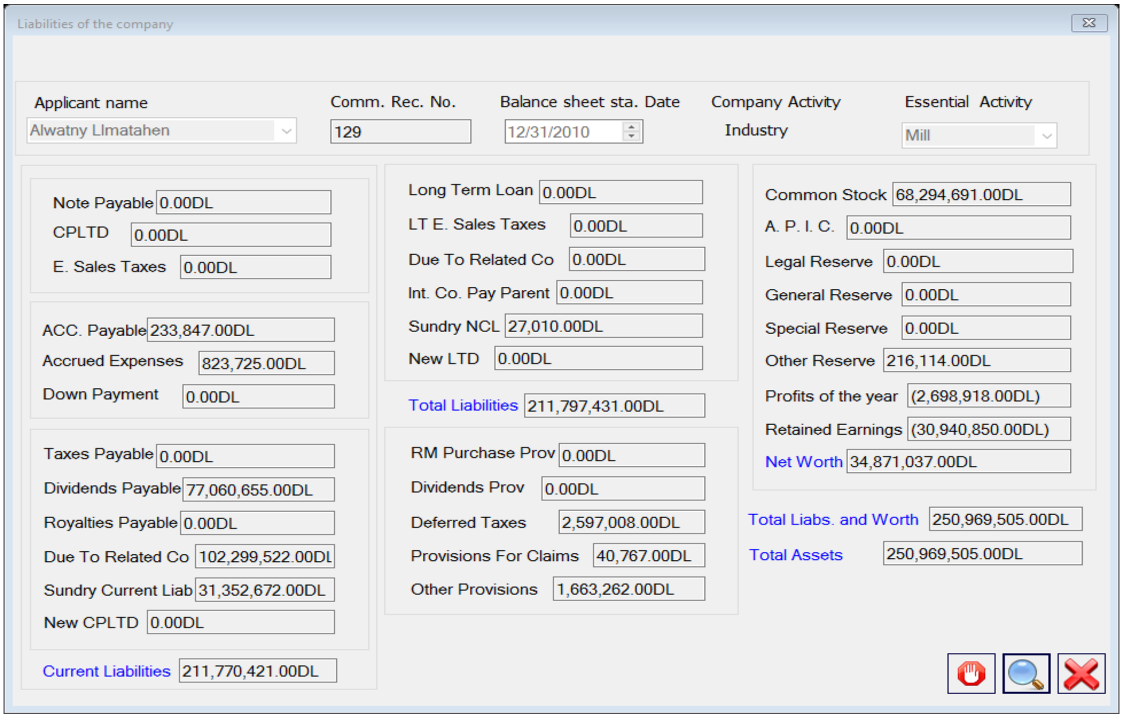
Investments 0.00DL
Prepaid Exp. 0.00DL
Due From Sister Co. 0.00DL
LTD to Affil. Comp. 0.00DL
Sundry Non Current Assets 0.00DL
Prepaid Taxes 0.00DL
Intangibles 1,200.00DL

Total Assets 161,057.00DL

Figure 4.4: The user interface of financial assets input of a loan applicant

As shown in **Figure 4.4,** when essential data have been entered in textboxes, the system calclutes the information needed to calculate the financial ratios, e.g. inventory, net receivable, current Assets, fixed assets and total assets. To confirm that such finacial Assets input is correct, the system compares the value of total assets with a value of formal total assets. The value of formal total assets is entered by a user. If the value of total assets is equal to the value of formal total assets, then the entered financial assets data are corret. Otherwise, the entered data are not.

The second user interface being the second input part of the balance sheet is the liabilities. It is shown in **Figure 4.5.** which presents essential data of balance sheet statement which have to be entered for calculating important information such as current liabilities, total liabilities, worth, and total liabilities and worth. The system works to confirm that such input data are correct, it compares between total liabilities and worth value and total assets value. If these values are equal, then all input data are correct, otherwise, there is an incorrect input data.


Liabilities of the company

Applicant name: Alwatny Llmatahen
Comm. Rec. No.: 129
Balance sheet sta. Date: 12/31/2010
Company Activity: Industry
Essential Activity: Mill

Note Payable: 0.00DL
CPLTD: 0.00DL
E. Sales Taxes: 0.00DL

ACC. Payable: 233,847.00DL
Accrued Expenses: 823,725.00DL
Down Payment: 0.00DL

Taxes Payable: 0.00DL
Dividends Payable: 77,060,655.00DL
Royalties Payable: 0.00DL
Due To Related Co: 102,299,522.00DL
Sundry Current Liab: 31,352,672.00DL
New CPLTD: 0.00DL

Current Liabilities: 211,770,421.00DL

Long Term Loan: 0.00DL
LT E. Sales Taxes: 0.00DL
Due To Related Co: 0.00DL
Int. Co. Pay Parent: 0.00DL
Sundry NCL: 27,010.00DL
New LTD: 0.00DL

Total Liabilities: 211,797,431.00DL

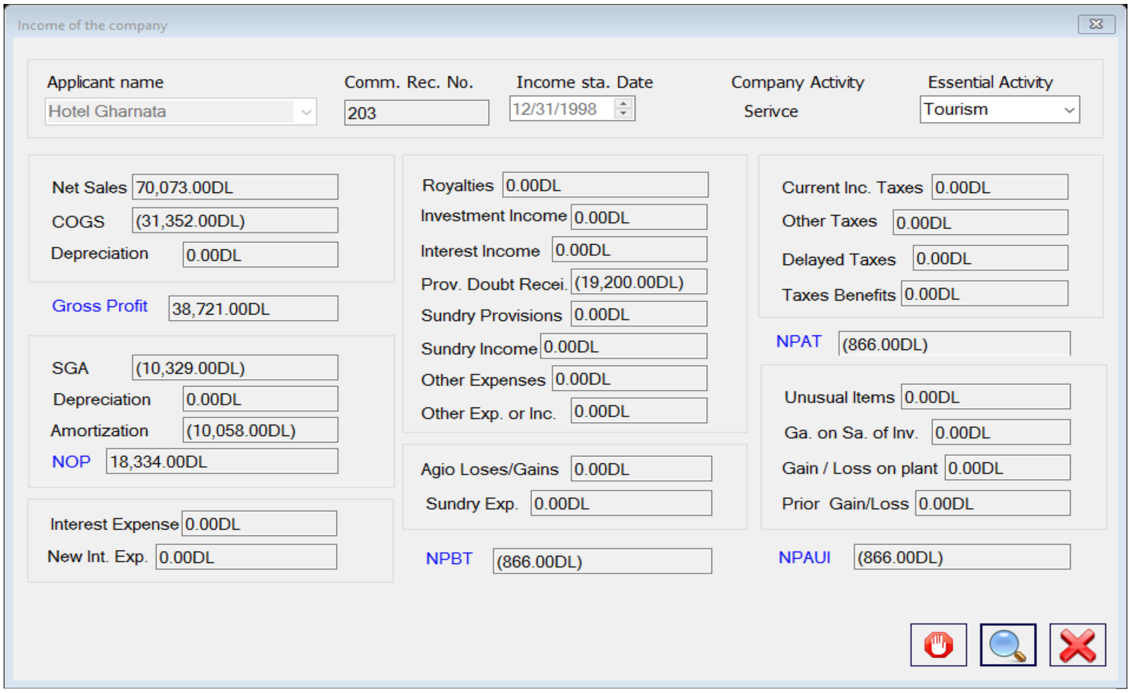
RM Purchase Prov: 0.00DL
Dividends Prov: 0.00DL
Deferred Taxes: 2,597,008.00DL
Provisions For Claims: 40,767.00DL
Other Provisions: 1,663,262.00DL

Common Stock: 68,294,691.00DL
A. P. I. C.: 0.00DL
Legal Reserve: 0.00DL
General Reserve: 0.00DL
Special Reserve: 0.00DL
Other Reserve: 216,114.00DL
Profits of the year: (2,698,918.00DL)
Retained Earnings: (30,940,850.00DL)
Net Worth: 34,871,037.00DL

Total Liabs. and Worth: 250,969,505.00DL
Total Assets: 250,969,505.00DL

Figure 4.5: The user interface of financial liabilities input of a loan applicant

The third user interface is for financial data input of the income statement. From the user interface in **Figure 4.6,** the user can fill all textboxes with input data. The system uses the input data to calculate the major information such as gross profit, NOP, NPBT, NPAT and NPAUI. Consequently, when making a decision for the credit approaval, each company must exist in one of the databases with its two input parts of balance sheet statement and income statement.



Income of the company

Applicant name: Hotel Gharnata
Comm. Rec. No.: 203
Income sta. Date: 12/31/1998
Company Activity: Serivce
Essential Activity: Tourism

Net Sales: 70,073.00DL
COGS: (31,352.00DL)
Depreciation: 0.00DL
Gross Profit: 38,721.00DL
SGA: (10,329.00DL)
Depreciation: 0.00DL
Amortization: (10,058.00DL)
NOP: 18,334.00DL
Interest Expense: 0.00DL
New Int. Exp.: 0.00DL

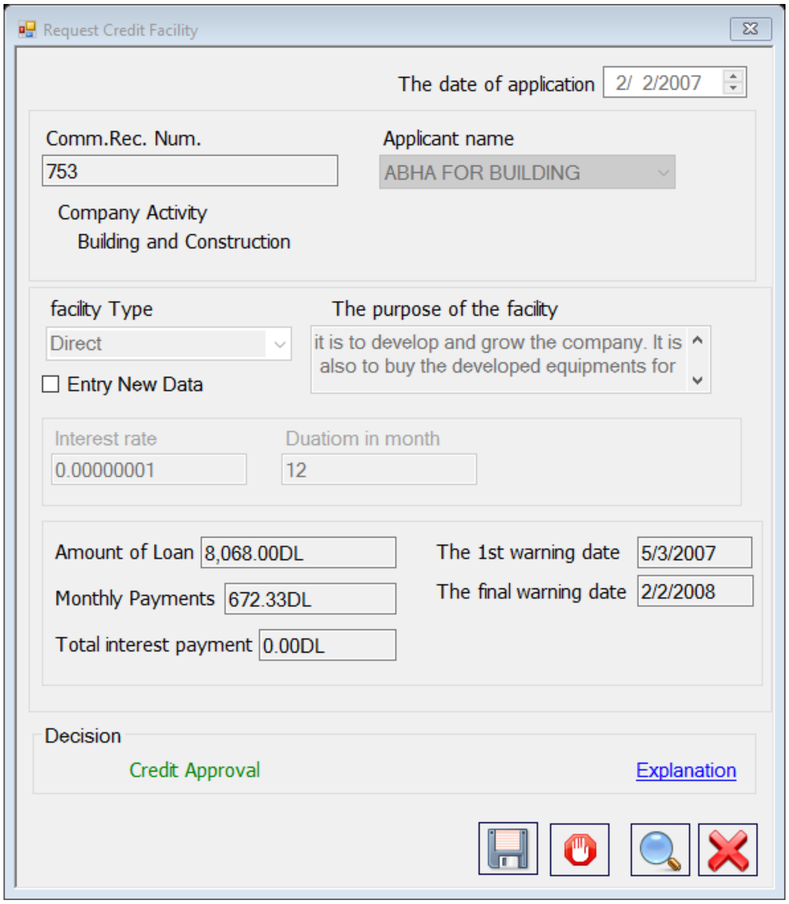
Royalties: 0.00DL
Investment Income: 0.00DL
Interest Income: 0.00DL
Prov. Doubt Recei.: (19,200.00DL)
Sundry Provisions: 0.00DL
Sundry Income: 0.00DL
Other Expenses: 0.00DL
Other Exp. or Inc.: 0.00DL
Agio Loses/Gains: 0.00DL
Sundry Exp.: 0.00DL
NPBT: (866.00DL)

Current Inc. Taxes: 0.00DL
Other Taxes: 0.00DL
Delayed Taxes: 0.00DL
Taxes Benefits: 0.00DL
NPAT: (866.00DL)
Unusual Items: 0.00DL
Ga. on Sa. of Inv.: 0.00DL
Gain / Loss on plant: 0.00DL
Prior Gain/Loss: 0.00DL
NPAUI: (866.00DL)

Figure 4.6: The user interface of input data for income statement of a loan applicant

4.4.2. Design Of User Interface Of Decision Making

Figure 4.7. shows a user interface used to request a financial facility by a firm applicant, where it has input the commerical record number No. of a company and the date of application to verify that this company exists with all financial data. The system then allows a credit analyst to determine the facility type, to enter the facility purpose, interest rate and duration in months which is based on the desires of both creditors and companies. After that, the system uses financial information to calculate financial ratios such as profibility ratios, liquidity ratios, activity ratios with their scores. Note that such information and their scores do not display in this user interface. The ES searches for a specific rule that matches some scores of the company's financial ratios for applying. When the decision is to grant a credit for the company, the amount of loan, monthly payment, the total interest payment, the first warning date, and the last wraning date are determined. The wraning dates are related to merit the loan. The amount of loan reperesents (WI) which is used to determine the credit value of the company as shown in **Table 1.2. Figure 4.7.** only displays the decision without any explanation.



Request Credit Facility

The date of application 2/ 2/2007

Comm.Rec. Num. 753

Applicant name ABHA FOR BUILDING

Company Activity
Building and Construction

facility Type Direct

□ Entry New Data

The purpose of the facility
it is to develop and grow the company. It is also to buy the developed equipments for

Interest rate 0.00000001

Duatiom in month 12

Amount of Loan 8,068.00DL

Monthly Payments 672.33DL

Total interest payment 0.00DL

The 1st warning date 5/3/2007

The final warning date 2/2/2008

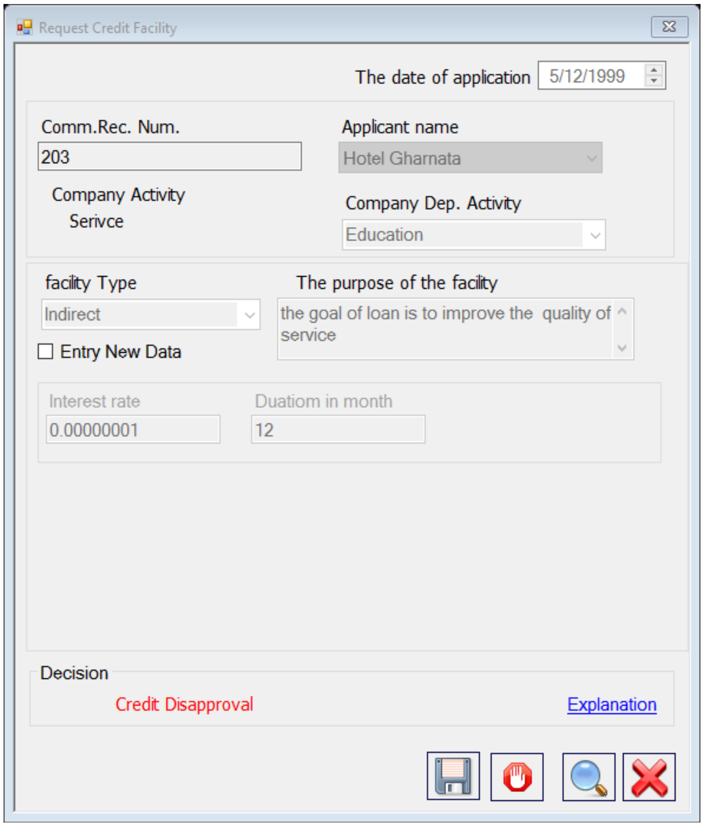
Decision
Credit Approval
Explanation

Figure 4.7: The user interface of financial facility for a company with credit approval

On the other hand, when a decision is denied by the ES, the ES only presents a disapproval credit decision of the company that is identical to the case shown in **Figure 4.8.**

4.4.3. Design Of User Interface Of Explanation

Explanation of decisions or results is an important role of human experts. The ES must be able to justify and explain its decisions. The Why and How questions are inquired by the user to understand why and how a particular conclusion or decision was made. In **Figure 4.9,** the ES clarifies the conclusion reached in **Figure 4.7.**



Request Credit Facility

The date of application 5/12/1999

Comm.Rec. Num. 203

Applicant name Hotel Gharnata

Company Activity Serivce

Company Dep. Activity Education

facilty Type Indirect

☐ Entry New Data

The purpose of the facilty: the goal of loan is to improve the quality of service

Interest rate 0.00000001

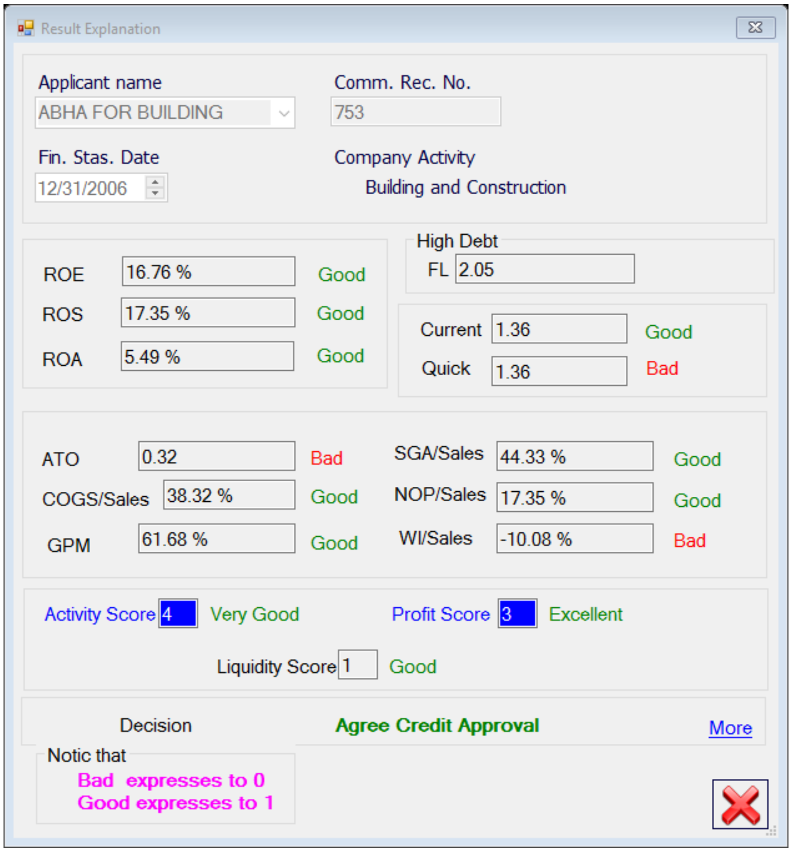
Duatiom in month 12

Decision: Credit Disapproval

Explanation

Figure 4.8: The user interface of financial facility for a company with credit disapproval

Figure 4.9. shows that the ES can explain why a specfied rule was executed or fired. It grants the credit of that company because that company has a good activity score and a good profibility score. The COG/Sales, GPM, SGA/Sales, and NOP/Sales are in the first important of credit analysis for assessing the ability of management to cover the cost from the sales by COG/Sales, the company's COG/Sales of 38.32% is good enough to cover its cost of service. The company's GPM of 61.68% reflects its ability to sell well above its cost of production or service. This SGA/Sales ratio expresses expenses such as salaries and rent from the sales they consume. 44.33% of such expenses compared with the sales they consume is reasonable. This NOP/Sales ratio illustrates the efficiency of the management in carring out its bussiness. The ppro company's NOP/Sales of 17.35% is the inside of the range in **Table 4.1**. Thus, the company has a good operating income.The ROE, ROS, ROA are in the second important part of credit analysis for judging the performance of the company.



Applicant name: ABHA FOR BUILDING

Comm. Rec. No.: 753

Fin. Stas. Date: 12/31/2006

Company Activity: Building and Construction

ROE	16.76 %	Good
ROS	17.35 %	Good
ROA	5.49 %	Good

High Debt

FL 2.05

Current	1.36	Good
Quick	1.36	Bad

ATO	0.32	Bad	SGA/Sales	44.33 %	Good
COGS/Sales	38.32 %	Good	NOP/Sales	17.35 %	Good
GPM	61.68 %	Good	WI/Sales	-10.08 %	Bad

Activity Score 4 Very Good

Profit Score 3 Excellent

Liquidity Score 1 Good

Decision Agree Credit Approval

Notic that

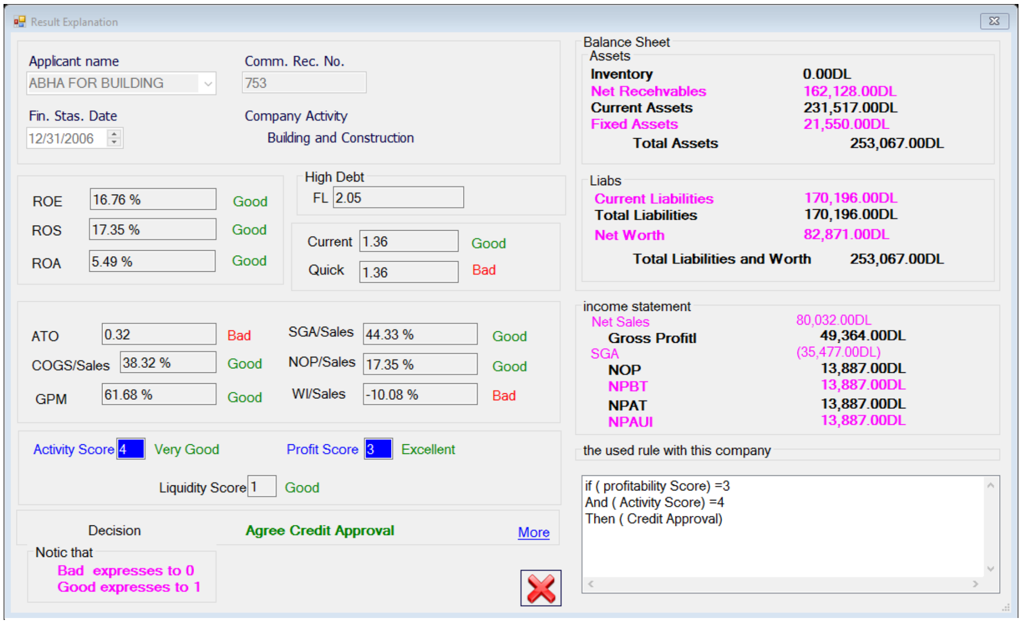
Bad expresses to 0

Good expresses to 1

Figure 4.9: The why and how explanation of ES

The ROE ratio indicates that the company earns $0.168\,Dirhams$ annually for each 1DL. The company’s ROS $of\ 17.35\%$ is well above average. The ROA $of\ 5.49\%$ implies that $1DL$ assets investement generates $0.549\,Dirhams$ of annual earning. All of profitability ratios have good scores. Thus, It can generate earnings from onwers’ equity, sales, and total assets. Consequently, the company has a good profit and its profit score is 3, which represents the sum of ROE, ROS and ROA scores. ROE, ROS, and ROA for profibility ratios are acceptable by the ES. The company’s four activity ratios are good. It means that each of them has one score, therefore, the sum of its activity score is 4. The profit score and the activity are summed. The company has (7/12)scores. As a result, the bank grants the credit to this company.

Figure 4.10. illustrates a sequence of rules applied to arrive at the decision to realize how and why the decision was reached by the ES along with more information that are used to calculate financial ratios. The sequence of rules contains two condition parts, the first condition consists of (profibility score=3), which is based on scores of ROE, ROS, and ROA ratios. The second condition is (activity score =4), which depends COG/Sales, GPM, SGA/Sales, and NOP/Sales ratios, and the action part is (credit approval). **Figure 4.11.** presents a part of the knowledge base that contains the rule used in **Figure 4.10,** where credit approval expresses to 1 and credit disapproval expresses to 0



Applicant name: ABHA FOR BUILDING
Comm. Rec. No.: 753
Fin. Stas. Date: 12/31/2006
Company Activity: Building and Construction

ROE	16.76 %	Good
ROS	17.35 %	Good
ROA	5.49 %	Good

High Debt
FL 2.05

Current	1.36	Good
Quick	1.36	Bad

ATO	0.32	Bad	SGA/Sales	44.33 %	Good
COGS/Sales	38.32 %	Good	NOP/Sales	17.35 %	Good
GPM	61.68 %	Good	WI/Sales	-10.08 %	Bad

Activity Score 4 Very Good
Profit Score 3 Excellent
Liquidity Score 1 Good

Decision: Agree Credit Approval

Notic that
Bad expresses to 0
Good expresses to 1

Balance Sheet

Assets

Inventory	0.00DL
Net Recehvables	162,128.00DL
Current Assets	231,517.00DL
Fixed Assets	21,550.00DL
Total Assets	253,067.00DL

Liabs

Current Liabilities	170,196.00DL
Total Liabilities	170,196.00DL
Net Worth	82,871.00DL
Total Liabilities and Worth	253,067.00DL

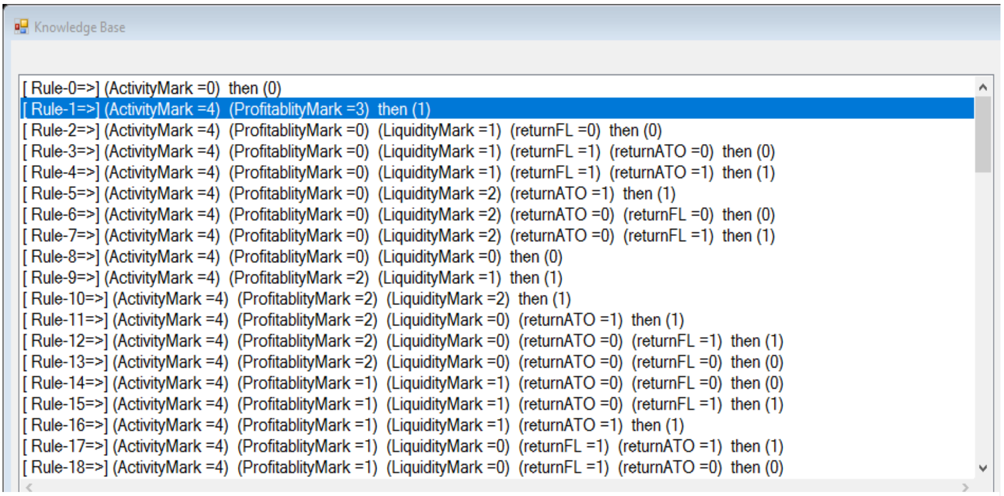
income statement

Net Sales	80,032.00DL
Gross Profitl	49,364.00DL
SGA	(35,477.00DL)
NOP	13,887.00DL
NPBT	13,887.00DL
NPAT	13,887.00DL
NPAUI	13,887.00DL

the used rule with this company

```
if ( profitability Score) =3
And ( Activity Score) =4
Then ( Credit Approval)
```

Figure 4.10: The why and how explanation of ES with series of rules

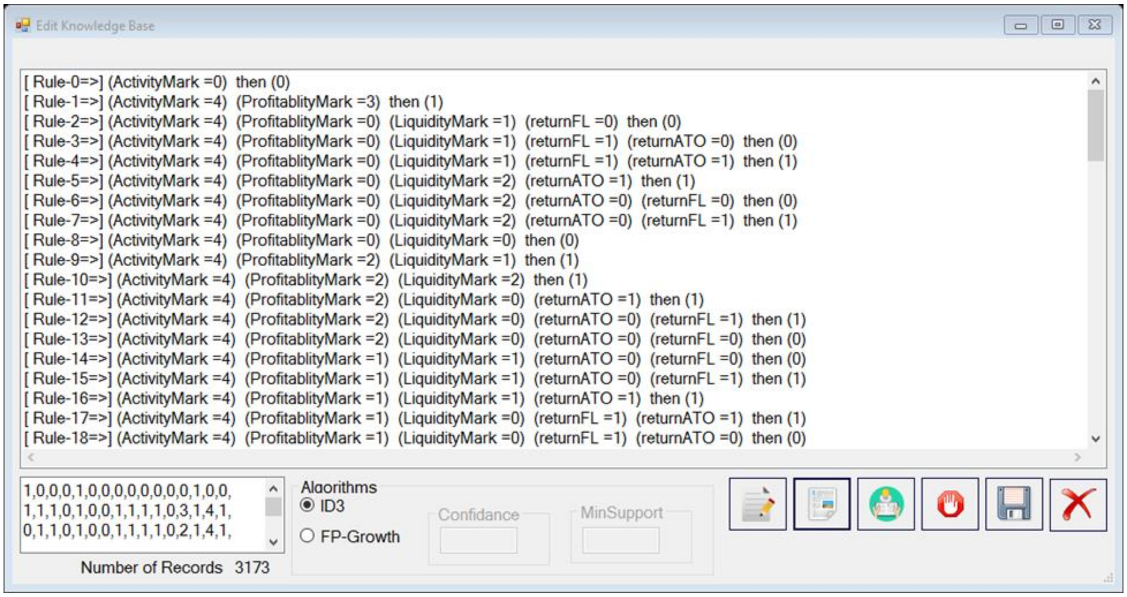


```
[ Rule-0=>] (ActivityMark =0)  then (0)
[ Rule-1=>] (ActivityMark =4)  (ProfitablityMark =3)  then (1)
[ Rule-2=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =1)  (returnFL =0)  then (0)
[ Rule-3=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =1)  (returnFL =1)  (returnATO =0)  then (0)
[ Rule-4=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =1)  (returnFL =1)  (returnATO =1)  then (1)
[ Rule-5=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =2)  (returnATO =1)  then (1)
[ Rule-6=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =2)  (returnATO =0)  (returnFL =0)  then (0)
[ Rule-7=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =2)  (returnATO =0)  (returnFL =1)  then (1)
[ Rule-8=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =0)  then (0)
[ Rule-9=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =1)  then (1)
[ Rule-10=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =2)  then (1)
[ Rule-11=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =0)  (returnATO =1)  then (1)
[ Rule-12=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =0)  (returnATO =0)  (returnFL =1)  then (1)
[ Rule-13=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =0)  (returnATO =0)  (returnFL =0)  then (0)
[ Rule-14=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =1)  (returnATO =0)  (returnFL =0)  then (0)
[ Rule-15=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =1)  (returnATO =0)  (returnFL =1)  then (1)
[ Rule-16=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =1)  (returnATO =1)  then (1)
[ Rule-17=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =0)  (returnFL =1)  (returnATO =1)  then (1)
[ Rule-18=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =0)  (returnFL =1)  (returnATO =0)  then (0)
```

Figure 4.11: Knowledge base of ES presents a specific rule used in Figure 4.10.

4.4.4. Design Of User Interface Of Knowledge Editor

This user interface helps human experts to build knowledge base by the ML algorithms as a model to take decisions through this user interface, the domain expert can update the knowledge of the ES. The ES allows a domain expert to insert and view data from database to text files to be used to construct knowledge base. There are two options to update the knowledge base of the ES as shown in **Figure 4.12**. After building the knowledge base by ID3 algorithm, if the knowledge base is accepted by the domain expert, s/he can save the knowledge in the flat file that is illustrated in **Figures 4.12**. and **Figure 4.13.**



```
[ Rule-0=>] (ActivityMark =0)  then (0)
[ Rule-1=>] (ActivityMark =4)  (ProfitablityMark =3)  then (1)
[ Rule-2=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =1)  (returnFL =0)  then (0)
[ Rule-3=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =1)  (returnFL =1)  (returnATO =0)  then (0)
[ Rule-4=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =1)  (returnFL =1)  (returnATO =1)  then (1)
[ Rule-5=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =2)  (returnATO =1)  then (1)
[ Rule-6=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =2)  (returnATO =0)  (returnFL =0)  then (0)
[ Rule-7=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =2)  (returnATO =0)  (returnFL =1)  then (1)
[ Rule-8=>] (ActivityMark =4)  (ProfitablityMark =0)  (LiquidityMark =0)  then (0)
[ Rule-9=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =1)  then (1)
[ Rule-10=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =2)  then (1)
[ Rule-11=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =0)  (returnATO =1)  then (1)
[ Rule-12=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =0)  (returnATO =0)  (returnFL =1)  then (1)
[ Rule-13=>] (ActivityMark =4)  (ProfitablityMark =2)  (LiquidityMark =0)  (returnATO =0)  (returnFL =0)  then (0)
[ Rule-14=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =1)  (returnATO =0)  (returnFL =0)  then (0)
[ Rule-15=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =1)  (returnATO =0)  (returnFL =1)  then (1)
[ Rule-16=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =1)  (returnATO =1)  then (1)
[ Rule-17=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =0)  (returnFL =1)  (returnATO =1)  then (1)
[ Rule-18=>] (ActivityMark =4)  (ProfitablityMark =1)  (LiquidityMark =0)  (returnFL =1)  (returnATO =0)  then (0)
```

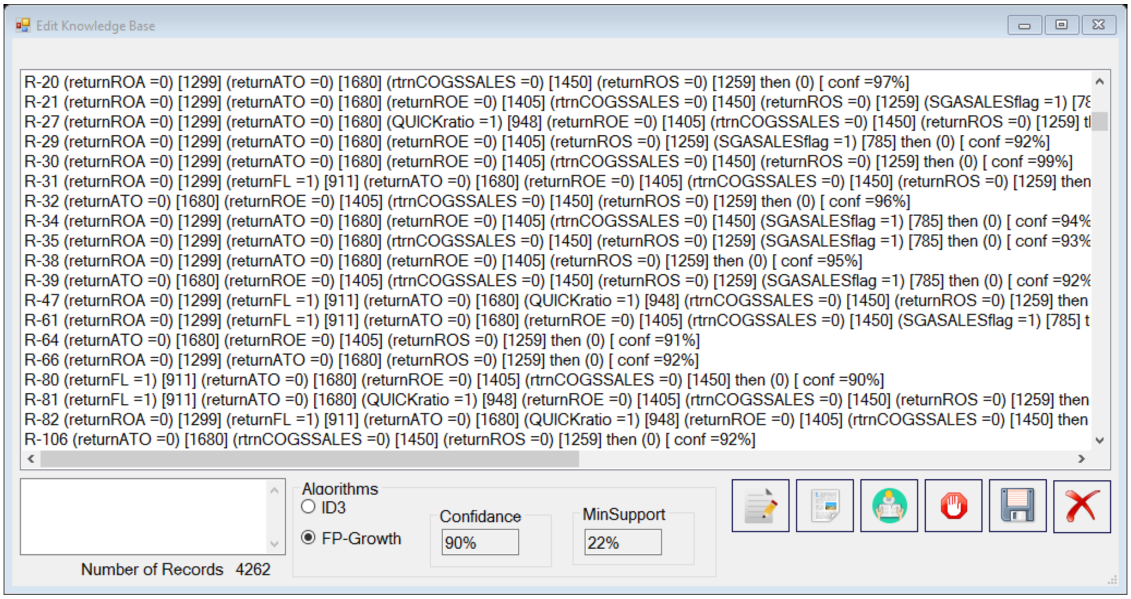
```
1,0,0,0,1,0,0,0,0,0,0,0,1,0,0,
1,1,1,0,1,0,0,1,1,1,1,0,3,1,4,1,
0,1,1,0,1,0,0,1,1,1,1,0,2,1,4,1,
```

Number of Records 3173

Algorithms
● ID3
○ FP-Growth

Figure 4.12: Editing knowledge base by ID3 algorithm

The domain expert will not be able to updtae the knowledge base while the total of instances is less than 3000 examples to avoid the overfitting, that caused by limited training data set. In this study, there are a lot of instances, The forest trees is used to overcome the overfitting.



```
R-20 (returnROA =0) [1299] (returnATO =0) [1680] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ conf =97%]
R-21 (returnROA =0) [1299] (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] (SGASALESflag =1) [78
R-27 (returnROA =0) [1299] (returnATO =0) [1680] (QUICKratio =1) [948] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] t
R-29 (returnROA =0) [1299] (returnATO =0) [1680] (returnROE =0) [1405] (returnROS =0) [1259] (SGASALESflag =1) [785] then (0) [ conf =92%]
R-30 (returnROA =0) [1299] (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ conf =99%]
R-31 (returnROA =0) [1299] (returnFL =1) [911] (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then
R-32 (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ conf =96%]
R-34 (returnROA =0) [1299] (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (SGASALESflag =1) [785] then (0) [ conf =94%
R-35 (returnROA =0) [1299] (returnATO =0) [1680] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] (SGASALESflag =1) [785] then (0) [ conf =93%
R-38 (returnROA =0) [1299] (returnATO =0) [1680] (returnROE =0) [1405] (returnROS =0) [1259] then (0) [ conf =95%]
R-39 (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] (SGASALESflag =1) [785] then (0) [ conf =92%
R-47 (returnROA =0) [1299] (returnFL =1) [911] (returnATO =0) [1680] (QUICKratio =1) [948] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then
R-61 (returnROA =0) [1299] (returnFL =1) [911] (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (SGASALESflag =1) [785] t
R-64 (returnATO =0) [1680] (returnROE =0) [1405] (returnROS =0) [1259] then (0) [ conf =91%]
R-66 (returnROA =0) [1299] (returnATO =0) [1680] (returnROS =0) [1259] then (0) [ conf =92%]
R-80 (returnFL =1) [911] (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] then (0) [ conf =90%]
R-81 (returnFL =1) [911] (returnATO =0) [1680] (QUICKratio =1) [948] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then
R-82 (returnROA =0) [1299] (returnFL =1) [911] (returnATO =0) [1680] (QUICKratio =1) [948] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] then
R-106 (returnATO =0) [1680] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ conf =92%]
```

Algorithms
○ ID3
● FP-Growth

Confidance: 90%

MinSupport: 22%

Number of Records 4262

Figure 4.13: Editing knowledge base by FP-Growth algorithm

4.5.Knowledge Validation

As mentioned in chapter 2, knowledge validation task is concerned with validing and verifying the knowledge by testing cases until its quality is reasonable. Tested results are presented to human expert for verifying the accuracy of the system(Aronson, Liang, & Turban, 2005). In this section, the test phase contains 32 testing cases. Thus, the rate of correct classifications in testing cases is 91%, the rate of incorrect classifications in testing cases is 9%. The human expert is not available in this stage in KE.

4.6. Summary

The chapter Four presents the process of data collection and analysis and implements the knowledge base of the ES in this study. The results of the study is presented as the user interfaces of financial input data, decision making, results explanation. The rate of correct classification is 91% that reflects the accuracy of the ES.

Chapter 5. Conclusion

5.1.Introduction

The major objective of this work is to develop the ES in credit analysis using a novel algorithm called ID3 algorithm and FP-growth algorithm as ML techniques. The ID3 algorithm and FP-growth algorithm were designed to overcome the problem of knowledge acquisition bottleneck in this ES, where ES is able to make decisions. This chapter displays the important steps throughout this study as a summary of the study. It also suggests future recommendations.

5.2.Conclusion of the study

The major contribution of this study is to propose an ES for evaluating and supporting decisions for credit analysis of firms based on financial data. This thesis introduces an ES which adapts to its environment by using ML techniques for learning from the data set. This ES has the following advantages: First, the representation method of semantic tree that used to represent the knowledge is straightforwardly understood, secondly, knowledge can be altered without changing source code; as a result, there no needs to recode the ES. In other words, It does not require the domain experts to explicitly express their knowledge and does not require knowledge engineers to code the knowledge.

This enables banks to obtain the knowledge to support decision making process in a high quality manner as expert knowledge is combined. This process can be standardized and described in a structured manner. The ES with ML algorithms (ID3 and FP-growthing algorithms) can be applyed to provide a consultation to support in the process of decision making.

Hence, the ES and ML algorithms (ID3 and FP-growth algorithms) can be considered a valuable and effective tool to support banks in their challenge of optimally performing the process of developing the models according to changing environments to support the decision taking of credit approval.

List of References

Abdou, H. A., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: a review of the literature. *Intelligent Systems in Accounting, Finance and Management, 18*(2-3), 59-88.

Anjaneyulu, K. S. (1998). Expert systems: An introduction. *Resonance, 3*(3), 46-58.

Aronson, J. E., Liang, T.-P., & Turban, E. (2005). *Decision support systems and intelligent systems*: Pearson Prentice-Hall.

Aktan, S. (2011). Application of machine learning algorithms for business failure prediction. *Investment Management and Financial Innovations, 8*(2), 52-65.

Alpaydin, E. (2010). Introduction to Machine Learning. [Sl]. In: The MIT Press.

Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management science, 49*(3), 312-329.

Bragg, S. M. (2002). *Business Ratios and Formulas: A Comprehensive Guide*: Wiley.

C. J. (1992). Knowledge discovery in databases: An overview. *AI Magazine, 13*(3), 57.

Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). Machine learning: A historical and methodological analysis. *AI Magazine, 4*(3), 69.

Cohn, L. F., & Harris, R. A. (1992). *Knowledge based expert systems in transportation* (Vol. 183): Transportation Research Board.

Coppin, B. (2004). *Artificial Intelligence Illuminated*: Jones & Bartlett Learning.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*(10), 78-87.

Elmasri, R., & Navathe, S. B. (2011). Fundamentals of Database Systems, 2011. In: Addison-Wesley, ISBN-10.

Fekri-Ershad, S., Tajalizadeh, H., & Jafari, S. (2013). Design and Development of an Expert System to Help Head of University Departments. *arXiv preprint arXiv:1308.0356*.

Frawley, W. J., Piatetsky-Shapiro, G., & Matheus,

Fridson, M. S., & Alvarez, F. (2002). *Financial Statement Analysis: A Practitioner's Guide*: Wiley.

Fürnkranz, J., Gamberger, D., & Lavrač, N. (2012). *Foundations of rule learning*: Springer Science & Business Media.

Gonciarz, T. (2014). An expert system for supporting the design and selection of mechanical equipment for recreational crafts. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation, 8*.

Gupta, S., & Singhal, R. (2013). Fundamentals and characteristics of an expert system. *International Journal on Recent and Innovation Trends in Computing and Communication, 1*(3), 110-113.

Hall, M., Witten, I., & Frank, E. (2011). Data mining: Practical machine learning tools and techniques. *Kaufmann, Burlington*.

Hall, M. A. (1999). Correlation-based feature selection for machine learning.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*: Elsevier.

Harrington, P. (2012). *Machine learning in action* (Vol. 5): Manning Greenwich, CT.

Hoseini, S. (2013). Creation and Application of Expert System Framework in Granting the Credit Facilities. *International Journal of Academic Research in Business and Social Sciences, 3*(9), 358.

Huang, C.-L., Chen, M.-C., & Wang, C.-J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert systems with applications, 33*(4), 847-856.

Huang, X., & Jensen, J. R. (1997). A machine-learning approach to automated knowledge-base building for remote sensing image analysis with GIS data. *Photogrammetric engineering and remote sensing, 63*(10), 1185-1193.

ID3_algorithm. (2016). Retrieved from http://en.wikipedia.org/wiki/ID3_algorithm.

Jones, M. T. (2008). *Artificial intelligence: a systems approach*: Laxmi Publications, Ltd.

Kaimal, L. B., Metkar, A. R., & Rakesh, G. (2014). Self learning real time expert system. *International Journal on Soft Computing, Artificial Intelligence and Applications, 3*(2), 13-25.

Khanbabaei, M., & Alborzi, M. (2013). The use of genetic algorithm, clustering and feature selection techniques in construction of decision tree models for credit scoring. *International Journal of Managing Information Technology, 5*(4), 13.

Krishnamoorthy, C. S., & Rajeev, S. (1996). *Artificial Intelligence and Expert Systems for Engineers*: Taylor & Francis.

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review, 26*(3), 159-190.

Long, P., Lawrey, S., & Ellis, V. (2016). *Cambridge International AS and A Level IT Coursebook with CD-ROM*: Cambridge University Press.

Liu, B. (2011). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*: Springer Berlin Heidelberg.

Lucas, P., & Van Der Gaag, L. *Principles of expert systems*.

Luger, G. F. (2005). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*: Addison-Wesley.

Marsland, S. (2015). *Machine learning: an algorithmic perspective*: CRC press.

Matthiesen, R. (2010). *Bioinformatics methods in clinical research*: Springer.

Maimon, O., & Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook*: Springer US.

McMillan, M. (2005). *Data structures and algorithms using Visual Basic. NET*: Cambridge University Press.

Miller, B. M. (1993). Object-oriented expert systems and their applications to sedimentary basin analysis.

Negnevitsky, M. (2005). *Artificial Intelligence: A Guide to Intelligent Systems*: Addison-Wesley.

Nilsson, N. J. (1996). Introduction to machine learning. An early draft of a proposed textbook.

Nosratabadi, H. E., Nadali, A., & Pourdarab, S. (2012). Credit assessment of bank customers by a fuzzy expert system based on rules extracted from association rules. *International Journal of Machine Learning and Computing, 2*(5), 662.

Ong, S. K., Xu, Q., & Nee, A. Y. (2008). *Design reuse in product development modeling, analysis and optimization*: World Scientific.

Partridge, D. (1998). *Artificial Intelligence and Software Engineering: Understanding the Promise of the Future*: Glenlake Publishing Company, Limited.

Python), A. (2015). A Complete Tutorial on Tree Based Modeling from Scratch (in R & Python). Retrieved from http://www.analyticsvidhya.com/blog/2015/01/decision-tree-simplified

Rodrigues, R. (2015). AN EXPERT SYSTEM FOR DISBURSEMENT OF HOME LOANS. *International Journal of Technical Research and Applications, 3*(5), 142-148.

Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*: Prentice Hall.

Sadatrasoul, S., Gholamian, M., Siami, M., & Hajimohammadi, Z. (2013). Credit scoring in banks and financial institutions via data mining techniques: A literature review. *Journal of AI and Data Mining, 1*(2), 119-129.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*: Cambridge University Press.

Siler, W., & Buckley, J. J. (2005). *Fuzzy Expert Systems and Fuzzy Reasoning*: Wiley.

Smith, C., McGuire, B., Huang, T., & Yang, G. (2006). The history of artificial intelligence. *University of Washington*, 27.

Tugas, F. C. (2012). A Comparative Analysis of the Financial Ratios of Listed Firms Belonging to the Education Subsector in the Philippines for the Years 2009-2011. *International Journal of Business and Social Science, 3*(21).

Vance, D. E. (2002). *Financial Analysis and Decision Making*: McGraw-Hill Education.

Villena Román, J., Collada Pérez, S., Lana Serrano, S., & González Cristóbal, J. C. (2011). *Hybrid approach combining machine learning and a rule-based expert system for text categorization*.

Vizureanu, P. (2010). *Expert systems*. Rijek, Crotia: InTech.

Watson, M. (2008). *Practical Artificial Intelligence Programming With Java* (3rd ed ed.). United States: Mark Watson.

Appendix A. ID3 Algorithm And FP-Growth Algorithm

A.1. Attribute Selection Measures

The information measure is entropy shorts for Shannon entropy which comes from the information theory's father Claude Shannon(Harrington, 2012; Shalev-Shwartz & Ben-David, 2014). The important question is how to use information theory in decision trees for spliting a dataset(Harrington, 2012). ID3 uses Information gain(IG) for spliting (Shalev-Shwartz & Ben-David, 2014). ID3 algorithm starts at the root node by iteratively calculating both of the entropy and IG of every unused attribute for all the dataset. Thus, it chooses the attribute with smallest entropy value or largest IG value(" ID3_algorithm," 2016). The dataset is split by the selected attribute.

A.1.1. Entropy

The definition of Entropy is the expected value of the information(Harrington, 2012), in order to classify an instance in the dataset T or to identify the class of an instance in the dataset T(Elmasri & Navathe, 2011). As a rule, the logarithms are used in base 2, then the unit of entropy is called bits(Hall, Witten, & Frank, 2011; Han, Pei, & Kamber, 2011).

$$H(T) = -\sum_{x \in X} p(x) log_2 p(x) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (A.1)$$

Where that

- T is the current training dataset for which Entropy is calculated(" ID3_algorithm," 2016).
- X is a set of classes in T (" ID3_algorithm," 2016).
- $p(x)$ is the probability of the number of objects in x to the number of objects in T(" ID3_algorithm," 2016).

A.1.2. Information Gain

Information gain (IG) is the difference measure in entropy before spliting the dataset T based on only proportion of classes and after spliting the dataset T by using a specific attribute A(Han et al., 2011; " ID3_algorithm," 2016). IG illustrates how much information would be gained by selecting the best spliting attribute for partitioning data whenever necessary. In other words, it is equivalent to saying that we need to partition based on the spliting attribute which would do the best classification. It means that the information amount required to terminate classifying a set of instances is minimal $\text{information}(\text{entropy})$(Han et al., 2011).

$$IG(A,T) = H(T) - \sum_{s \in S} p(s)H(s) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (A.2)$$

Where

- $H(T)$ is entropy of the training dataset(" ID3_algorithm," 2016).
- A is the spliting attribute which is selected based on the highest $IG(A,T)$(" ID3_algorithm," 2016)
- S is the subsets generated by splitting the dataset T depended on the attribute A(" ID3_algorithm," 2016).
- $p(s)$ is the probability of the occurrence number of objects in the subset s to the occurrence number of objects in T (" ID3_algorithm," 2016).
- $H(s)$ is the entropy of the subset s(" ID3_algorithm," 2016).

The majar question is: how does ID3 algorithm use IG to construct decision tree. The next section explains how to do that.

A.1.3. Constructing Decision Tree

The decision tree comprises nodes that make up a rooted tree, this means that it is directed tree with the root node that is without incoming branches(edges), unlike all other nodes with incoming branches. Nodes that have outgoing branches are referred to as internal or test nodes. The rest of other nodes with outgoing branches are called leaves that hold labeled classes(Han et al., 2011). A decision tree is a classification model indicated as a recursive partition for the instance space(Maimon & Rokach, 2010).

To illustrate how to employ IG for constructing a decision tree by considering wheather data as shown in **Table 3.1**. Every attribute is discrete-valued. The class label attribute is $Play$. It has two values namely: yes and no. The attribute $Play$ has two classes. As shown in **Table 3.1** which presents nine instances of class yes and five instances of class no. To find the spliting attribute of the wheather dataset, it needs to calculate IGs for each attribute. Hence, the attribute is selected as the root node that can be created with the highest IG at the top level(Hall et al., 2011; Han et al., 2011), and creats all branches of the root node for each possible value of the root node(Hall et al., 2011).

Table A.5.1: Weather dataset(Source: Witten et al, 2011)

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Rainy	Mild	High	FALSE	Yes
Rainy	Cool	Normal	FALSE	Yes
Rainy	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Mild	High	TRUE	No

Now we must calcuclate the entropy desired to classify a set of instances in the wheather dataset T

$$H(T) = -9/14\, log_2(9/14) - 5/14\, log_2(5/14) = 0.940 bits$$

Then, we need to check the distribution of yes and no instances for each nominal value of the attribute as follows

$$H_{Outlook}(T) = 5/14 \times (-2/5\,log_2(2/5) - 3/5\,log_2(3/5)) + 4/14 \times \left(-4/4\,log_2(4/4)\right) + 5/14 \times \left(-3/5\,log_2(3/5) - 2/5\,log_2(2/5)\right) = 0.694\,bits$$

$$H_{Temp}(T) = 4/14 \times (-2/4\,log_2(2/4) - 2/4\,log_2(2/4)) + 6/14 \times \left(-4/6\,log_2(4/6) - 2/6\,log_2(2/6)\right) + 4/14 \times \left(-3/4\,log_2(3/4) - 1/4\,log_2(1/4)\right) = 0.911\,bits$$

$$H_{Humidity}(T) = 7/14 \times (-3/7\,log_2(3/7) - 4/7\,log_2(4/7)) + 7/14 \times \left(-6/7\,log_2(6/7) - 1/7\,log_2(1/7)\right) = 0.789\,bits$$

$$H_{Windy}(T) = 8/14 \times (-6/8\,log_2(6/8) - 2/8\,log_2(2/8)) + 6/14 \times \left(-3/6\,log_2(3/6) - 3/6\,log_2(3/6)\right) = 0.892\,bits$$

Consequently, the IGs for such a partitioning would be

$$IG(Outlook, T) = 0.940 - 0.694 = 0.246\,bits.$$

$$IG(Temp, T) = 0.940 - 0.9111 = 0.029\,bits.$$

$$IG(Humility, T) = 0.940 - 0.789 = 0.152\,bits.$$

$$IG(Windy, T) = 0.940 - 0.892 = 0.048\,bits.$$

Because $outlook$ attribute is the one with the highest IG among the attributes, it is chosen to be the splitting attribute as the root node. It is only selected once; in addition, branches are created for each value of the splitting attribute. The instances are then splitted according to the values of attribute $outlook$. So that we have three subsets which are partitioned according to outlook's values namely: $sunny$, $rainy$, and $overcast$(Han et al., 2011).

After spiltting the original dataset into three subsets by values of the attribute $outlook$, some of these subsets may not be completely pure(Hall et al., 2011). Then the algorithm continues, recursively, the algorithm picks a subset from three subsets and checks its purity. The subset based on $Outlook = sunny$. It is not pure. For the After spiltting the original dataset into three subsets by values of the attribute $outlook$, some of these subsets may not be completely pure(Hall et al., 2011). Then the algorithm continues, recursively, the algorithm picks a subset from three subsets and checks its purity. The subset based on $Outlook = sunny$. It is not pure. For the best classification, the same question is asked: which attribute must be selected to provide the best classification among three attributes $Temperature$, Humidity, and $Windy$. The IGs for each shows to be $IG(temperature) = 0.571\, bits$, $IG(Humidity\,) = 0.971\, bits$, and $IG(Windy) = 0.020\, bits$. Thus, $Humidity$ is the best classification with the highest IG for the subset which is based on $Outlook = sunny$. So it is selected. By the attribute $Humidity$, all instances are classified well. The result two subsets based on $Humidity$ $a$re pure. Notice that the second subset based on $Outlook = overcast$ is pure. It means all instances belong to the same class yes.

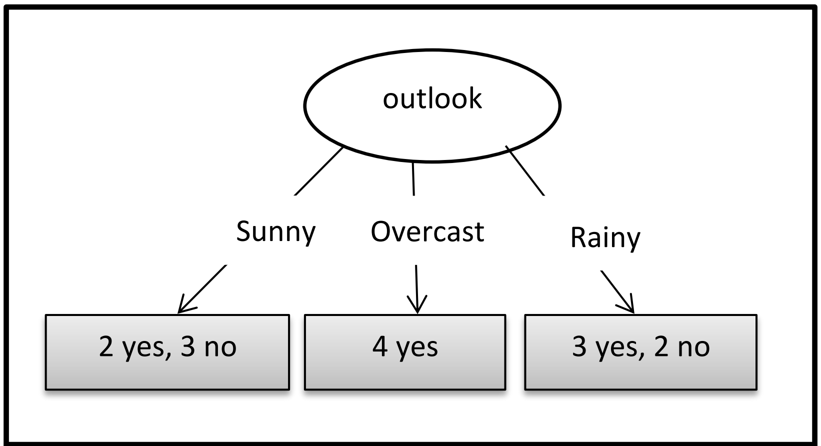


Figure A.1. Tree stumps of the weather data by the attribute outlook. **(Source:** Witten et al, 2011**)**

The last subset based on $Outlook = Rainy$ is not pure. So it needs more classification. Again, the repeated question is, which attribute has to be selected for the best classification among $Temperature$ and $Windy$. The IGs for each shows to be $IG(temperature) = 0.020\, bits$ and $IG(Windy) = 0.971\, bits$. Thus, $Windy$ is the best classification with the highest information for the subset based on $Outlook = Rainy$. Hence, all two subset based on $Windy$ are pure(Hall et al., 2011). As a result, the algorithm returns to the final decision Tree for wheather dataset as shown in **Figure A.2.**

According to (Liu, 2011), When the dataset T turns out to be pure, the $H(T)$ turns out to be smaller. Actually, it can be presented for this binary case (two classes) namely: $\{yes, no\}$,

- when $P(yes) = 0.5$ and $P(no) = 0.5$ the $H(T)$ has the maximum value,(1 bit).
- When all the data in T belong to one class the $H(T)$ has the minimum value (0 bit).

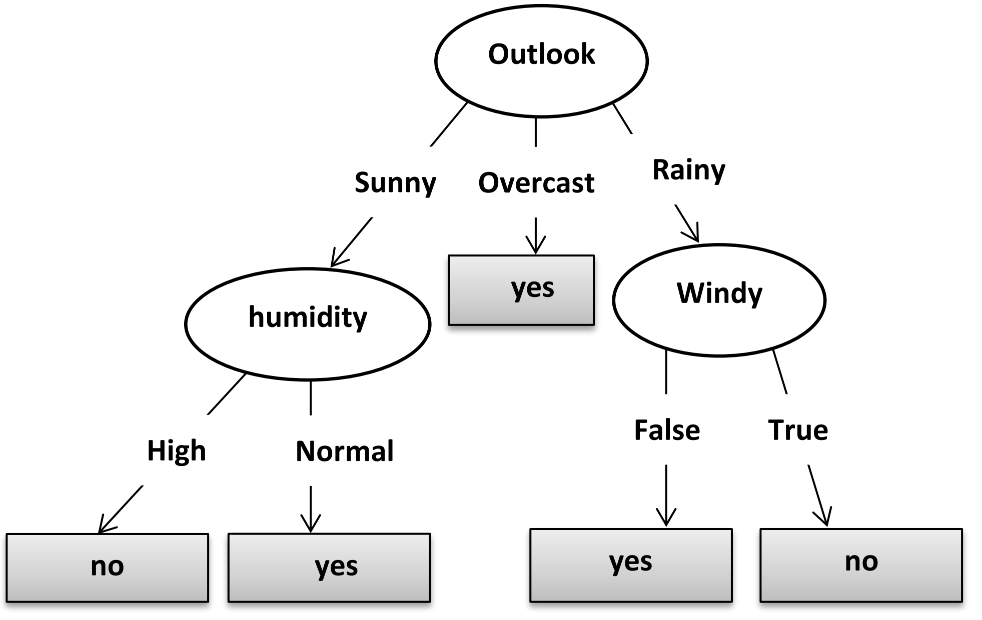


Figure A.2: **Decision tree for the weather data**. **(Source:** Witten et al, 2011**)**

A.1.4. Overfitting

A decision tree algorithm recursively splits the data until there is no impurity or there is no attribute left. This process may generate trees that are very deep and many tree leaves may cover very few training examples. If we use a tree to predict the training data set, the accuracy will be very high. However, when it is used to classify an unseen test set, the accuracy may be very low. The learning is thus not effective. The decision tree does not generalize the data well. This phenomenon is called overfitting. Overfitting is usually caused by noise in the data, wrong class values/labels and/or wrong values of attributes, but it may also be due to the complexity and randomness of the application domain(Liu, 2011).

To overcome overfitting in decision tree learning, the method of “random forests” invented by Breiman (2001) could be used. A random forest comprises a set of decision trees, where each tree is built by carrying out the algorithm such as using the ID3 algorithm. This method considers all the random attributes from a vector of attributes. The ID3 algorithm depends on a random subsample from the samples S which are subsets of training data set, where at each splitting phase, the algorithm selects the splitting attribute based on maximizing IG. It means that when the size of the subset is small or large, the overfiting can be prevented(Shalev-Shwartz & Ben-David, 2014).

A.2. Pattern-Growth Approach For Association Rules

The FP-tree is similar to other trees in computer science, but it has links that are used for associating similar items. The linked items can be considered as a linked list. A search tree is different, where an item can frequent many times in the same tree. The FP-tree is utilized to keep the frequency of occurrence of itemsets(Harrington, 2012). FP- growth was presented by Han et al in 2004(Maimon & Rokach, 2010).

The data used with FP- growth are illustrated in **Table A.2.** To illustrate how to work with the data, the first step for building the FP-tree is to count the frequency of occurrence for each item($support\ count$)(Harrington, 2012). The support is exactly the count for transactions comprising the item(Elmasri &

Navathe, 2011). Any Item with support less than minimum support is then ignored as shown in **Table A.3.** Let's use 2 to be a minimum support, when building the tree, each itemset inserts to an existing path in FP-tree if it exists. It just increases its $support\,count$, where items in the itemsets are sorted along with their support descendingly(Elmasri & Navathe, 2011). If it doesn't exist, the algorithm creates a new path(Harrington, 2012).

The item header table contains items with support count which are equal to or greater than the minimum support. The main fields of header table for the item are item identifier, support count, and node link(Elmasri & Navathe, 2011).

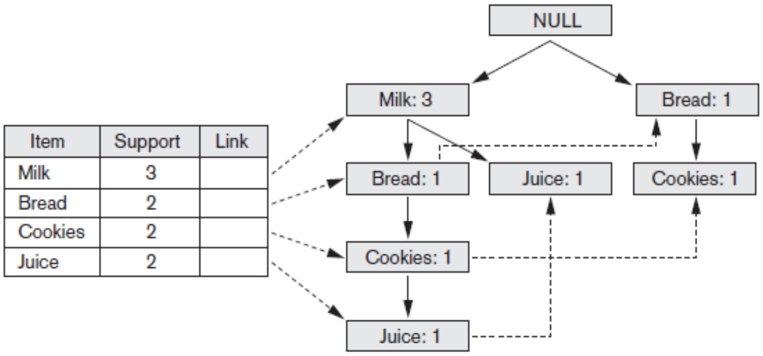
Table 5.2: Sample transactions in market-basket model (Source: Elmasri and Navathe , 2011)

Transactions	Itemsets	Filtered and Sorted Itemsets
T1	milk, bread, cookies, juice	milk, bread, cookies, juice
T2	milk, juice	milk, juice
T3	milk, eggs	milk
T4	bread, cookies, coffee	bread, cookies

Table A.3: Items with their support counts (Source: Elmasri and Navathe , 2011)

Item	Support Count
milk	3
bread	2
cookies	2
juice	2
eggs	1(ignored)
coffee	1(ignored)

With transaction *T1* represented in the sorted itemset $\{milk, bread, cookies, juice\}$ in **Table A.2**. The FP-growth algorithm makes a $null$ root node of the FP-tree and adds the first item in a transaction $T1$, it is $milk$ that has the most support in the itemset as child for the root node and increases its count to 1. The second item is $bread$ in the sorted itemset in $T1$, it is added to FP-tree as a child for the $milk$ node and increases its count to be 1. The thrid item in the itemset is $cookies$, it is added to Fp-tree as a child for the $bread$ node and increases the count of the $cookies$ node to 1. The final item in this itemset is $juice$. It is added as child for the $cookies$ node and then increases its count to 1. As shown in **Figure A.3.** The next transaction $T2$ with the sorted itemset is $\{milk, juice\}$. Initiating, at the root node, notic that the node of $milk$ exists with the $count = 1$. So it increases count of the $milk$ node to be 2 and moves on for the second item in $T2(\ juice)$. Notic also that there is no child for the $milk$ node with a $juice$ item. The algorithm creates a new node for $juice$ and increases its count to 1. For the transaction *T3*, it contians one item $\{milk\}$.



Item	Support	Link
Milk	3	
Bread	2	
Cookies	2	
Juice	2	

Figure A.3. FP-tree and item header table(**Source:**Elmasri and Navathe,2011)

Beginning at the root node, we see that it exists in Fp-tree. Thus it just updates the count of the $milk$ node to be 3. Finally the last transaction $T3$ includes $\{bread, cookies\}$. Beginning at the root node, it is clear that item $bread$ does not exist as a child for the root node, so that it creates a new node for $bread$ item and increases its count to 1 and than adds the $cookies$ item as child for the $bread$ node and increases the count to be 1. The final FP-tree is shown in **Figure A.3**(Elmasri & Navathe, 2011).

For extracting frequent itemsets from the FP-tree, we strat with the item of the minimum support and the last entry into the header table. As shown in Figure A.3 Juice. Note that juice happens in two FP-tree paths (frequence of juice can be followed by node-links). These paths are $\{\,\mathrm{milk, bread, cookies, juice}\,\}: 1$ and $\{\mathrm{milk, juice}\}: 1$. Prefix paths of juice as the conditional pattern base are $\langle\mathrm{milk, bread, cookies}: 1\rangle$ and $\langle\mathrm{milk}: 1\rangle$. Thus, a conditional FP-tree which consists of a single node, that is $\langle\mathrm{milk}: 2\rangle$. bread and $\mathrm{cookies}$ are ignored, because thier support count of 1 are less than minimum support of 2. All frequent patterns of juice are generated, that is $\{\mathrm{milk, juice}\,\}$ with support count of 2. The next one with minimum support is cookies that entered before Juice. We see that there are two prefix paths for cookies namely: $\langle\mathrm{milk, bread}: 1\rangle$ and $\langle\mathrm{bread}: 1\rangle$ as a conditional pattern base. The conditional FP-tree of cookies contians only one node, that is $\langle\mathrm{bread}: 2\rangle$. All frequent patterns of cookies are generated $\{\mathrm{bread, cookies}: 2\}$. The next node is bread, when building the conditional pattern base for bread, note that there is only one node with support count of 1 that is $\langle\mathrm{milk}: 1\rangle$. Since It is also less than minimum support, the conditional FP-tree of bread is empty. Consequently, there is no frequent pattern for bread.

The final frequent item in the header table is milk. This item is the top item of the FP-tree. Being at the top, there is no conditional pattern base for milk and the conditional FP-tree of milk is empty. Therefore, no frequent patterns for milk are generated(Han et al., 2011),(Elmasri & Navathe, 2011).

According to (Han et al., 2011), generating strong association rules form such transaction in database can be done where satisfying the minimum support and minimum confidence by using the following equation

$$confidence\left(A \Rightarrow B\right) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (A.3)$$

Where :

$P(B|A)$ that represents the conditional probability is used to express $confidence\left(A \Rightarrow B\right)$

$support\_count(A \cup B)$ that represets a total of transcations that are resulted from combining members of itemset A with members of itemsets B.

$support\_count(A)$ that represests a total of transactions comprising members of the itemset A

Depending on the confidence equation $(A.3)$, the strong associacion rules can be generated

$milk \Rightarrow juice : 2$	$\boldsymbol{confidence\ = 2/3}$ **=66.6%**
$bread \Rightarrow\ cookies: 2$	$\boldsymbol{Confidence =\ 2/2 = 100}\%$

Let’s assume that the minimum confidence threshold in this example is 75%. Thus, the last one rule is a strong rule. Because its confindince=100% which is greater than the minimum confidance threshold(75%).

Appendix B. ID3 Algorithm And FP-Growth Algorithm In Practice

B.1. The ID3 Algorithm

The ID3Algorithm class is concerned with constructing the decision tree as the knowledge base for the ES. The ID3 algorithm takes a 2-dimensional array that represents financial data(scores) with labeled classes as the training data set and two arraylists to keep the names of attributes for financial data that comes from the database. We also need to counter rows as examples in the array and an arraylist to store the size of the arrays that are used to construct the decision tree. The ID3 algorithm demands to invoke some methods such as $findFreqValuesForAttrbtAndThierIG()$ that is concerned with finding to an occurence of all attributes with their values and their IGs. It returns to an arraylist for frequency of values for all attributes and another arraylist for their IGs. After obtianing that information by using the preivous method, the ID3 algorithm uses such information by $FindAttributeWithMaxIG()$ to get the selected attribute with a maximum IG. By using this method $returnRatioNoWithItsValuesAndFreq(),$ the algorithm can have the values for a specific attribute and the frequency of its values. The values of a particular attribute and their frequency that are utilized by $CutArrayIntoPieces()$ to cut the original array to pieces according to frequency of values for a particular attribute. When constructing a tree, usually backtracks to a parent will be needed to insert another child node to that parent, as $toComebackToParentNode()$ method does. $checkingReply()$ method is employed to check if all rows in the array belong to the same labeled class or not. If all rows belong to the same class then invoking $addExamplesWiththeSameLabeledClassToGraph$ that is used to insert data to the graph, otherwise it recurves the ID3 algorithm again. The ID3 algorithm continues to classify examples recursively until there is no attribute to select or there is no data for classifying.

```
Public Sub ID3(ByRef OrginalArray(,) As String, ByRef AttrbtName
As ArrayList, ByRef AtrbtNam As ArrayList, EXcounter As Integer,
ByRef EXcntr As ArrayList, ByRef indxAtrbt As ArrayList, ByRef
TreeGraph As Graph, ByRef strvex As ArrayList, ByRef collctedRule
As ArrayList, ByRef EdgeParent As ArrayList)
Dim arlst As New ArrayList
Dim frqValArr, ValuesOfAttrbt, attrbuteArr, infoValues,
frqValArr12 As New ArrayList
findFreqValuesForAttrbtAndThierIG(OrginalArray, frqValArr,
ValuesOfAttrbt, attrbuteArr, infoValues)
If AttrbtNam.Count = 0 Then
Return
End If
DetermineNameOfAttributes(str, arlst)
arlst.RemoveAt(arlst.Count - 1)
Dim infoValues11() As Double = infoValues.ToArray(GetType(System.Double))
Dim AtrtNme() As String = arlst.ToArray(GetType(System.String))
Dim strlst11 As New ArrayList
Dim r As Integer = 0
Dim indx As Integer = 0
Dim max As Double = 0
Dim ratioName As String = ""
FindAttributeWithMaxIG(infoValues11, indx, max, ratioName, arlst)
indxAtrbt.Add(indx)
While AtrbtNam.Contains(ratioName) AndAlso r < EXcounter
returnRatioNoWithItsValuesAndFreq(indx, attrbuteArr,
ValuesOfAttrbt, strlst11, frqValArr, frqValArr12)
For i As Integer = 0 To strlst11.Count - 1
Dim APartOfArray(frqValArr12(i) - 1, arlst.Count) As String
CutArrayIntoPieces(OrginalArray, strlst11(i), APartOfArray, indx)
toComebackToParentNode(EdgeParent, strlst11, i)
If checkingReply(ss1, reply, AtrtNme) = True Then
Dim CollectionRules As String = ""
addExamplesWiththeSameLabeledClassToGraph(TreeAGraph, EdgeParent,
indxAtrbt, CollectionRules, strlst11, i, indx, strvex)
collctedRule.Add(CollectionRules)
EXcntr.Add(ss1.Length)
Else
ID3(APartOfArray, AttrbtName, AtrbtNam, EXcounter, EXcntr,
indxAtrbt, TreeGraph, strvex, collctedRule, EdgeParent)
End If
Next
```

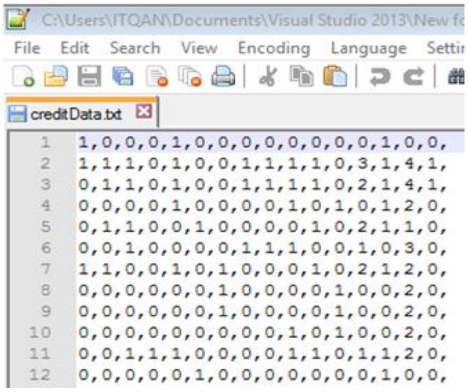
```
AtrbtNam.Remove(x)
indxAtrbt.Remove(indxAtrbt(indxAtrbt.Count - 1))
EdgeParent.RemoveAt(EdgeParent.Count - 1)
For item As Integer = 0 To EXcntr.Count - 1
r = r + EXcntr(item)
Next
End While
End Sub
```

B.2. FP-Growth Algorithm

As menationed in appendix A., The FP-growth algorithm scans the database only twice. The first scan for constructing the FP-tree, and the second scan for extracting frequent itemsets from the FP-tree. In this study, the first scan for constructing the FP-tree is used, we need the *Fp_tree_algorithm* method that picks up an array that contains all transactions and builds a FP-tree from transactions as paths and calculates thier frequency and inserts into the graph. The second step consists of two phases, one phase demands a *ConditionalPatternBases*() method which is employed to extract a conditional pattern base. The *ConditionalPatternBases*() method receives itemsets from the FP-tree and extracts itemsets for a particular item which represents only the decision.

Before applying the FP-Growth algorithm, a *CreateItemssetsArray* that is used to transform data in the text file “creditData.txt” to the array of itemsets where all itemsets are sorted according to the sorted list and saves all itemsets into “itemsets.txt” as shown in **Figure B.1.** and **Figure B.2.** resepectively. **Table B.1.** presents items with their supports. It also represents the sorted list for ordering items in itemsets before added to the FP-tree by the *InsertPathToTheGraph*() method.

The *conditionalFrequentPatterns*() method is used to obtain itemests that have support counts equal to or greater than *minSup*. The function of a *GenerateFrequentPattern* method is to generate association rules needed by the ES to classify new cases. It takes itemsets that are generated by the method of *conditionalFrequentPatterns*. Consequently, the *GenerateFrequentPattern* method uses a $Class\,TSet$ to generate all subsets and thus, association rules According to the minimum confidence threshold as shown in **Figure B.3.**


C:\Users\ITQAN\Documents\Visual Studio 2013\New fo
File Edit Search View Encoding Language Settir
creditData.txt

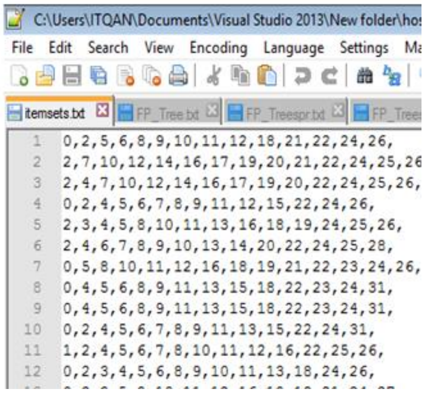
```
1,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,
1,1,1,0,1,0,0,1,1,1,1,0,3,1,4,1,
0,1,1,0,1,0,0,1,1,1,1,0,2,1,4,1,
0,0,0,0,1,0,0,0,0,1,0,1,0,1,2,0,
0,1,1,0,0,1,0,0,0,0,1,0,2,1,1,0,
0,0,1,0,0,0,0,1,1,1,0,0,1,0,3,0,
1,1,0,0,1,0,1,0,0,0,1,0,2,1,2,0,
0,0,0,0,0,0,1,0,0,0,0,1,0,0,2,0,
0,0,0,0,0,0,1,0,0,0,0,1,0,0,2,0,
0,0,0,0,0,0,0,0,0,1,0,1,0,0,2,0,
0,0,1,1,1,0,0,0,0,1,1,0,1,1,2,0,
0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,
```

Figure 5.1: Sample of Data from "creditData.txt"

Table 5.1. Items with their support counts for the study

Item	**Ratio with its value**	**Support Count**
0	ROA=0	2251
1	FL=1	2241
2	ATO=0	2228
3	QUICKratio=1	2221
4	ROE=0	2193
5	COGSSALES=0	2190
6	ROS=0	2172
7	SGASALES=1	2170
8	Decision=0	2167
9	NOPSALES=0	2157
10	WIsales=0	2141
11	GPM=0	2137
12	CRNTratio=1	2135
13	CRNTratio=0	2127
14	GPM=1	2124
15	WIsales=1	2121
16	NOPSALES=1	2105

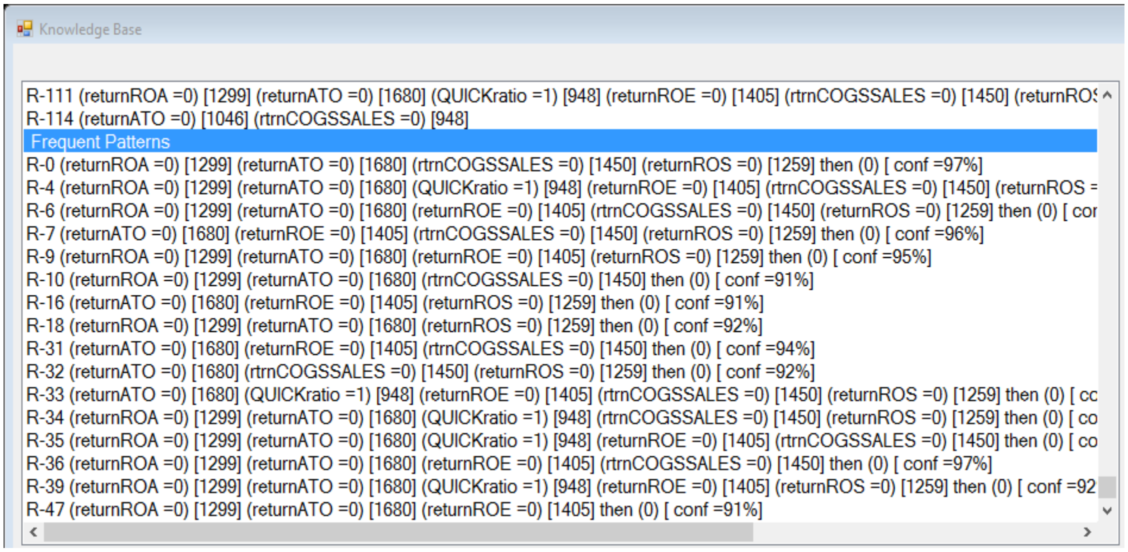
17	Decision=1	2095


C:\Users\ITQAN\Documents\Visual Studio 2013\New folder\ho
File Edit Search View Encoding Language Settings Ma
itemsets.txt
FP_Tree.txt
FP_Treespr.txt
FP_Tree

```
0,2,5,6,8,9,10,11,12,18,21,22,24,26,
2,7,10,12,14,16,17,19,20,21,22,24,25,26
2,4,7,10,12,14,16,17,19,20,22,24,25,26,
0,2,4,5,6,7,8,9,11,12,15,22,24,26,
2,3,4,5,8,10,11,13,16,18,19,24,25,26,
2,4,6,7,8,9,10,13,14,20,22,24,25,28,
0,5,8,10,11,12,16,18,19,21,22,23,24,26,
0,4,5,6,8,9,11,13,15,18,22,23,24,31,
0,4,5,6,8,9,11,13,15,18,22,23,24,31,
0,2,4,5,6,7,8,9,11,13,15,22,24,31,
1,2,4,5,6,7,8,10,11,12,16,22,25,26,
0,2,3,4,5,6,8,9,10,11,13,18,24,26,
```

Figure B.2: Sample transactions in “itemsets.txt

When (in this study, minimum support=22%), we have got frequent patterns. The minimum confidence threshold is 90%. Consequently, such rules (the only ones) generated which are robust.



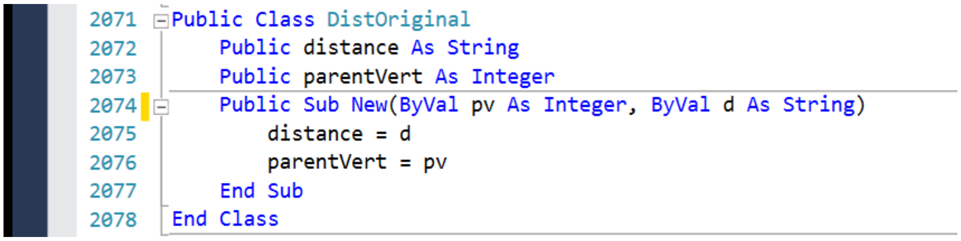
```
R-111 (returnROA =0) [1299] (returnATO =0) [1680] (QUICKratio =1) [948] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS
R-114 (returnATO =0) [1046] (rtrnCOGSSALES =0) [948]
Frequent Patterns
R-0 (returnROA =0) [1299] (returnATO =0) [1680] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ conf =97%]
R-4 (returnROA =0) [1299] (returnATO =0) [1680] (QUICKratio =1) [948] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =
R-6 (returnROA =0) [1299] (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ cor
R-7 (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ conf =96%]
R-9 (returnROA =0) [1299] (returnATO =0) [1680] (returnROE =0) [1405] (returnROS =0) [1259] then (0) [ conf =95%]
R-10 (returnROA =0) [1299] (returnATO =0) [1680] (rtrnCOGSSALES =0) [1450] then (0) [ conf =91%]
R-16 (returnATO =0) [1680] (returnROE =0) [1405] (returnROS =0) [1259] then (0) [ conf =91%]
R-18 (returnROA =0) [1299] (returnATO =0) [1680] (returnROS =0) [1259] then (0) [ conf =92%]
R-31 (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] then (0) [ conf =94%]
R-32 (returnATO =0) [1680] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ conf =92%]
R-33 (returnATO =0) [1680] (QUICKratio =1) [948] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ cc
R-34 (returnROA =0) [1299] (returnATO =0) [1680] (QUICKratio =1) [948] (rtrnCOGSSALES =0) [1450] (returnROS =0) [1259] then (0) [ co
R-35 (returnROA =0) [1299] (returnATO =0) [1680] (QUICKratio =1) [948] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] then (0) [ co
R-36 (returnROA =0) [1299] (returnATO =0) [1680] (returnROE =0) [1405] (rtrnCOGSSALES =0) [1450] then (0) [ conf =97%]
R-39 (returnROA =0) [1299] (returnATO =0) [1680] (QUICKratio =1) [948] (returnROE =0) [1405] (returnROS =0) [1259] then (0) [ conf =92
R-47 (returnROA =0) [1299] (returnATO =0) [1680] (returnROE =0) [1405] then (0) [ conf =91%]
```

Figure 5.3:Frequent Pattern with The minimum confidence threshold is 90%.

Appendix C. Inference Engine In Practice

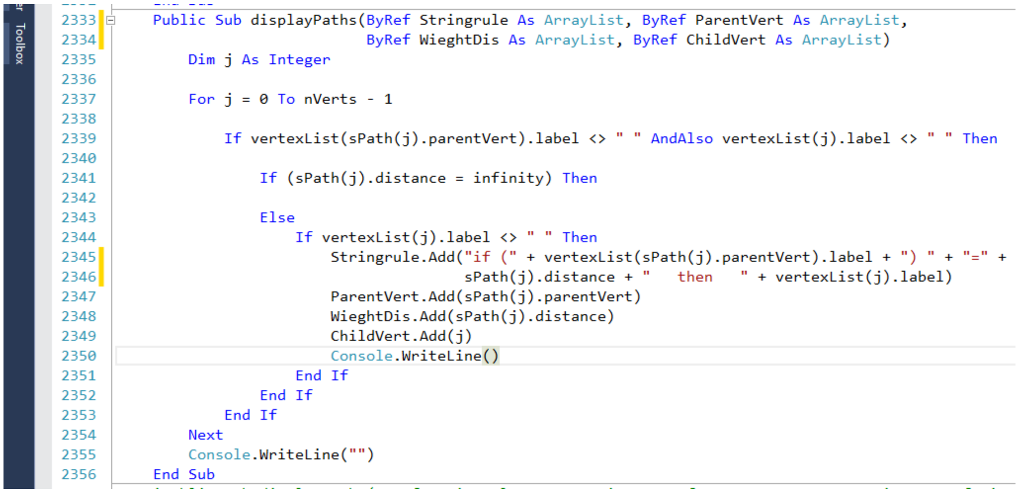
C.1. Inference Engine

A finding process of the path from a vertex to another is one important task in the graph. The path can be considered as a rule from knowledge base. Dijkstra's algorithm is used but it does not find the short path. It is to find all paths. *DistOriginal class* is used to keep the parent node with its distance that represents the weight. For finding the path we can use *path*() method that takes one integer value to represent the position of the vertex and returns to three arraylists. The *path*() method evokes the *getAdjNode* method which is to find and return an adjacent node as integer value. The *displayPath*() method is used to print the path.



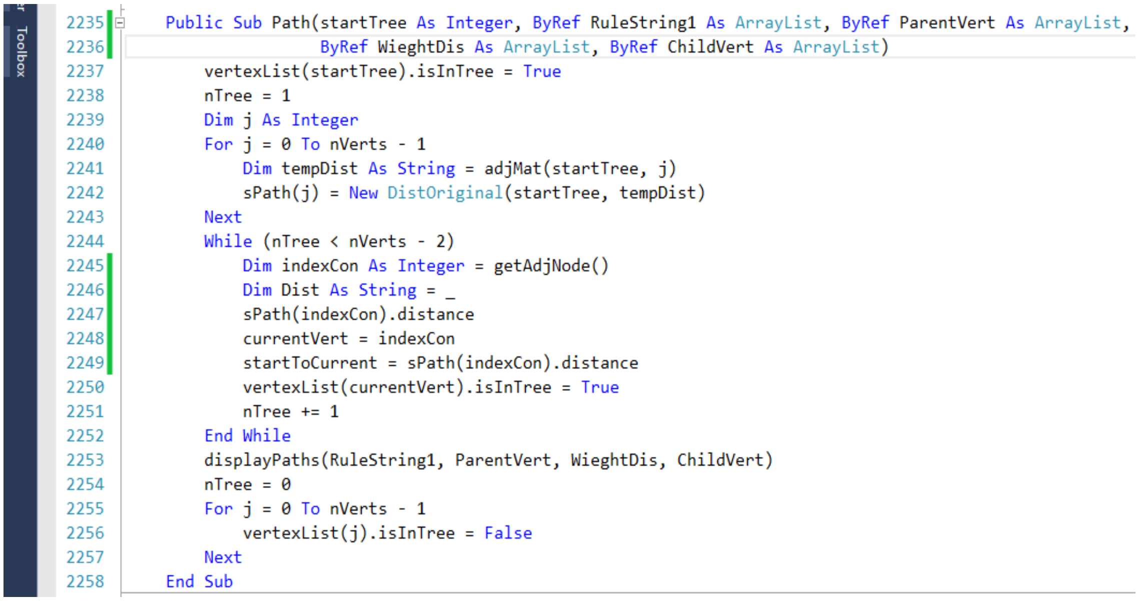
```
Public Class DistOriginal
    Public distance As String
    Public parentVert As Integer
    Public Sub New(ByVal pv As Integer, ByVal d As String)
        distance = d
        parentVert = pv
    End Sub
End Class
```

Figure 5.1:A Keeping Method of parents



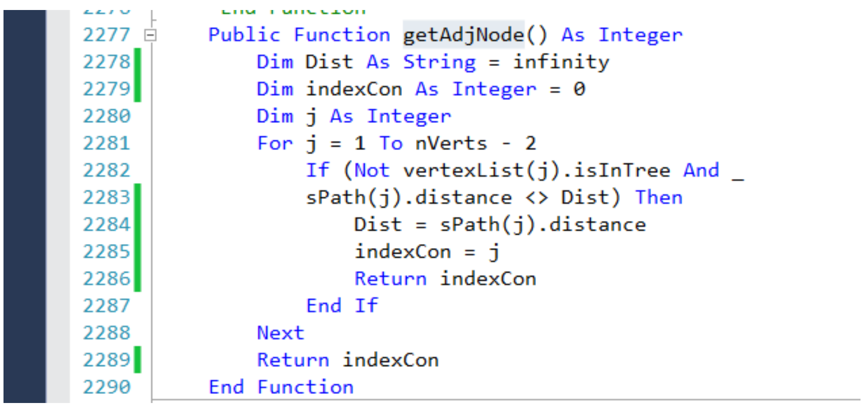
```
    Public Sub displayPaths(ByRef Stringrule As ArrayList, ByRef ParentVert As ArrayList,
                            ByRef WieghtDis As ArrayList, ByRef ChildVert As ArrayList)
        Dim j As Integer

        For j = 0 To nVerts - 1

            If vertexList(sPath(j).parentVert).label <> " " AndAlso vertexList(j).label <> " " Then

                If (sPath(j).distance = infinity) Then

                Else
                    If vertexList(j).label <> " " Then
                        Stringrule.Add("if (" + vertexList(sPath(j).parentVert).label + ") " + "=" +
                                       sPath(j).distance + "   then    " + vertexList(j).label)
                        ParentVert.Add(sPath(j).parentVert)
                        WieghtDis.Add(sPath(j).distance)
                        ChildVert.Add(j)
                        Console.WriteLine()
                    End If
                End If
            End If
        Next
        Console.WriteLine("")
    End Sub
```

Figure .C.2 : The displayPaths Method



```
    Public Sub Path(startTree As Integer, ByRef RuleString1 As ArrayList, ByRef ParentVert As ArrayList,
                    ByRef WieghtDis As ArrayList, ByRef ChildVert As ArrayList)
        vertexList(startTree).isInTree = True
        nTree = 1
        Dim j As Integer
        For j = 0 To nVerts - 1
            Dim tempDist As String = adjMat(startTree, j)
            sPath(j) = New DistOriginal(startTree, tempDist)
        Next
        While (nTree < nVerts - 2)
            Dim indexCon As Integer = getAdjNode()
            Dim Dist As String = _
            sPath(indexCon).distance
            currentVert = indexCon
            startToCurrent = sPath(indexCon).distance
            vertexList(currentVert).isInTree = True
            nTree += 1
        End While
        displayPaths(RuleString1, ParentVert, WieghtDis, ChildVert)
        nTree = 0
        For j = 0 To nVerts - 1
            vertexList(j).isInTree = False
        Next
    End Sub
```

Figure 5.2.C.3 :A finding method of paths



```
    End Function
    Public Function getAdjNode() As Integer
        Dim Dist As String = infinity
        Dim indexCon As Integer = 0
        Dim j As Integer
        For j = 1 To nVerts - 2
            If (Not vertexList(j).isInTree And _
            sPath(j).distance <> Dist) Then
                Dist = sPath(j).distance
                indexCon = j
                Return indexCon
            End If
        Next
        Return indexCon
    End Function
```

Figure.C.4: The method of getAdjNode

نظام خبير في التحليل الائتماني باستخدام تعلم الالة
(خوارزمية أي دي 3 و خوارزمية التطور- اف بي)

إعداد

تهاني مفتاح عبدالسلام حمد

المشرف

د. عبدالحميد محمد عبدالكافي

ملخص

الجزء الاهم في نظم الخبرة هو المعرفة. عملية اكتساب المعرفة و التعلم تكون مرحلة صعبة جدا. البنوك عادة يكون مسموح لها بتطوير نموذج لاجل تحديد مخاطرها الائتمانية، و الحاجة لمنهجية لاجل تطوير وحفظ هذا النموذج يكون موضوع مهم. هذا يضع اسئلة للبنوك أي طرق تكون مناسبة ومتماسكة ومتكاملة وتقدم بشفافية تحليلها وتشخيصها. هذه الرسالة تطور نظام خبير يمكنه التعلم من فئة البيانات تم الحصول عليها من مصرف الوحدة المركزي. ومن تلك الفئة يتم استدلال القواعد. وبتالي، النظام الخبير يمكن بنائه وفقا للقواعد التي تم استدلالها. وهكذا، البنوك يمكنها تطوير نماذجها بوسطة النظام الخبير الذي تكون لديه المقدرة للتعلم بوسطة خوارزمية الانقسام الذاتي 3 (اي دي 3) و تطور الانماط المتكررة(تطور اف بي).

نظام خبير في التحليل الائتماني باستخدام تعلم الالة

(خوارزمية أي دي 3 و خوارزمية التطور- اف بي)

قدمت من قبل:

تهاني مفتاح عبدالسلام حمد

تحت إشراف:

د. عبدالحميد محمد عبد الكافي

قدمت هذه الرسالة استكمالا لمتطلبات الحصول على درجة الماجستير في علوم الحاسوب

جامعة بنغازي
كلية تقنية المعلومات

ديسمبر2017