

การประยุกต์โครงข่ายประสาทเทียมเพื่อการทำนายระยะเวลาการเป็นลูกค้ำ กรณีศึกษาธุรกิจ
โทรคมนาคม



นาย อี้หวัง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์และสารสนเทศ ภาควิชาคณิตศาสตร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2552

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Application of Neural Network For Customer Lifetime Value Prediction: A Case Study In
A Telecommunication Business



Mr. Yi Wang

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science and Information

Department of Mathematics

Faculty of Science

Chulalongkorn University

Academic Year 2009

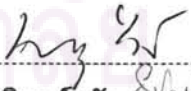
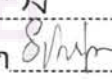

Copyright of Chulalongkorn University

นาย Yi Wang: (การประยุกต์โครงข่ายประสาทเทียมเพื่อการทำนายระยะเวลาการเป็นลูกค้า
กรณีศึกษารูทกิจโทรคมนาคม) อ.ที่ปรึกษาวิทยานิพนธ์หลัก :

อ. ดร. สิริพันธุ์ สงวนสินธุกุล, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ศาสตราจารย์ ดร. ชิดชนก เหลือ
สินทรัพย์ จำนวนหน้า 98 หน้า.

วิทยานิพนธ์ฉบับนี้ ศึกษาวิธีการปฏิบัติต่อความสัมพันธ์ที่มีกับลูกค้าในอุตสาหกรรม
โทรคมนาคม ดังนั้น การวัดและการจัดการต่อมูลค่าระยะยาวของลูกค้าอย่างเหมาะสมเพื่อใช้ในการ
การ พิจารณากำไรที่จะได้จากลูกค้าในระยะยาวจึงเป็นสิ่งสำคัญยิ่ง เนื่องจากลูกค้าโดยส่วน
ใหญ่มักจะมองหาสินค้าและบริการที่มีคุณภาพที่ดีแต่ในราคาที่ย่อมเยากว่าเสมอ ซึ่งมูลค่า
ระยะยาวของลูกค้านอกจากพิจารณาถึงการยกเลิกสินค้าและบริการจากลูกค้ารายเดิมแล้วยังรวม
ถึงการ พิจารณา การซื้อต่อเนื่องและการซื้อต่อยอดของลูกค้าเพื่อจูงใจลูกค้าอีกด้วย เพราะ
ฉะนั้น การได้มาซึ่ง ความไว้วางใจของลูกค้านอกจากการซื้อ จึงมีความสำคัญเป็นอย่างยิ่งใน
ปัจจุบัน ด้วยเหตุนี้ วิทยานิพนธ์ฉบับนี้จึงนำเสนอวิธีการที่ใช้ในการวิเคราะห์และทำนายมูลค่า
ระยะยาวของลูกค้าโดยใช้โครงข่ายประสาทเทียมซึ่งในงานวิจัยนี้ จะใช้โครงข่ายเพอร์เซ็ปตรอน
หลายชั้นผนวกกับอัลกอริทึมของ Levenberg-Marquardt ในการทำนายมูลค่าระยะยาวของ
ลูกค้า นอกจากนี้สมรรถนะของโครงข่ายประสาทเทียมจะถูกเปรียบเทียบกับต้นไม้การตัดสินใจซึ่ง
ใช้ อัลกอริทึม C4.5 ความถูกต้องของค่าการทำนายของโครงข่ายเพอร์เซ็ปตรอน อยู่ที่ 96.5
เปอร์เซ็นต์ ผลลัพธ์จากการทดลองแสดงให้เห็นว่าตัวแบบโครงข่ายประสาทเทียมมีประสิทธิภาพ
ดีกว่าต้นไม้ตัดสินใจ

ภาควิชาคณิตศาสตร์
สาขาวิชาวิทยาการคอมพิวเตอร์และสารสนเทศ
ปีการศึกษา 2552

ลายมือชื่อนิสิต 
ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก 
ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม 

4973611423 : MAJOR COMPUTER SCIENCE AND INFORMATION

KEYWORDS : Expert system / Data Mining / Neural Network / Multi-Layer Perceptron /

Customer Lifetime Value / Customer Loss Prediction

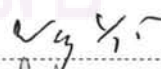
YI WANG: Application of neural network for customer lifetime value prediction: A case study in a telecommunication business. THESIS ADVISOR: SIRIPUN SANGUANSINTUKUL, Ph.D, THESIS COADVISOR : CHIDCHANOK LUPSINSAP, Ph.D, 98 pp.

This thesis studies the way to treat customer relationship in the telecommunications industry. Therefore, how a person measure and manage customer lifetime value (CLV) for determining the likely future profit from the customer is very important because the customer is always looking for better and cheaper products and services. The CLV not only combines with the churn management but also considers the cross-selling and up-selling to allure customer. Earning, not just buying, customers' loyalty is now mandatory. The method to analyze and predict customer lifetime value (CLV) using Artificial neural network (ANN) is proposed here. In this study, multi-layer perceptron (MLP) network with Levenberg-Marquardt algorithm is used to predict the CLV. Additionally, the performance of neural network is compared with decision tree using C4.5 algorithm. The accuracy of the prediction value of the neural network is 96.5%. The experiment result illustrated that the neural network model has a higher performance than the decision tree.

ศูนย์วิทยทรัพยากร

จุฬาลงกรณ์มหาวิทยาลัย

Department : Mathematics

Student's Signature : 

Field of Study : Computer Science and information

Advisor's Signature : 

Academic Year : 2009

Co-advisor's Signature : 

Acknowledgments

I would like to thank my advisor Dr. Siripun Sanguansintukul for all her inputs and wish to have updated key steps and for discussing and helping, resolve her concerns with the thesis. I also would like to thank Dr. Chidchannok Lursinsap for his support and methodical approaches to my study and everything. In addition, this research project would not be possible without the support of many people. As the author, I wish to express my sincere gratitude to Mr. Twatchai Sutitosatham, Mr. Awacharin Nachin (International Research Corporation Public) for they provide valuable devices without their helps and assistance this study would not have been successful.

I am also grateful to the thesis committee, The Chairman, Professor Suphakant Phimoltares, and Dr. Panjai Tantatsanwong, for their constructive criticisms and invaluable advises.

I also express the deeply thanks to my family. My father, my mother and my elder sister for their encourage during beginning of this study and when the time I ran into the steep troubles.

Finally, I would like to greatly appreciate to my wife infinity love and her support help me until now. And it would be worth to say that the thesis work for birthing baby, And the little one will be born on this coming Songkran Festival Day (2010).

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Contents

Abstract in Thai.....	v
Abstract in English.....	vi
Acknowledgements.....	vii
Contents.....	viii
List of Figures.....	x
List of Tables.....	xi
CHAPTERS	
1 INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Study Description.....	1
1.3 The Objectives of Research.....	2
1.4 The Scope of Study.....	3
1.5 Literatures Review.....	3
2 BACKGROUND, METHODOLOGY AND MODEL DESIGN.....	5
2.1 Industry Background and Data Selection.....	5
2.1.1 Industry Background.....	4
2.2 Modeling CLV.....	7
2.3 Methodologies.....	9
2.3.1 MLP Neural Networks.....	9
2.3.2 Introduction to Neural Networks.....	9
2.3.3 MLP Neural Network Architecture.....	12
2.3.4 Learning Algorithm for MLP Neural Network.....	14
2.3.5 Levenberg-Marquardt Algorithm.....	14
2.4 Dimension Reducing Methods.....	16
2.4.1 Principal Components Analysis.....	16
2.5 Decision Tree.....	18
2.5.1 Decision Tree with C4.5 Algorithm.....	19
2.6 Prediction Process Flow Design.....	20
3 EXPERIMENTAL APPLICATION.....	22
3.1 Experimental Data.....	22
3.2 Data Preparation Step.....	24

3.3	Input Refining.....	25
3.3.1	Data Normalization.....	25
3.3.2	Input Reduction.....	27
3.3.3	Data Partition.....	29
3.4	Tuning Decision tree Models.....	29
3.4.1	Maximal Tree.....	29
3.4.2	The Right-sized tree.....	30
3.5	Tuning Neural Networks Models.....	32
3.5.1	Preliminary Training.....	32
3.5.2	Early Stop.....	33
3.5.3	Number of Hidden Units.....	34
3.6	Cross Validation.....	35
4	EXPERIMENT RESULT	37
4.1	Experimental processing results.....	37
4.2	Accuracy Evaluation.....	40
4.2.1	Results of MLP Neural Network Models.....	41
4.2.2	Result of Decision tree Models with C4.5 Algorithm.....	43
4.3	Comparison.....	45
5	CONCLUSIONS AND FUTURE WORKS.....	47
5.1	Conclusions.....	49
5.2	Discussions.....	50
5.3	Future works	49
	REFERENCES.....	50
	APPENDICES.....	51
	Appendix A Data Fields For Prediction.....	53
	Appendix B Coding.....	69
	Appendix C Quality Measure Methods Study.....	79
	Appendix D Decision Tree Prediction Reference.....	80
	Appendix E Data Fields after PCA.....	87
	VITAE.....	89

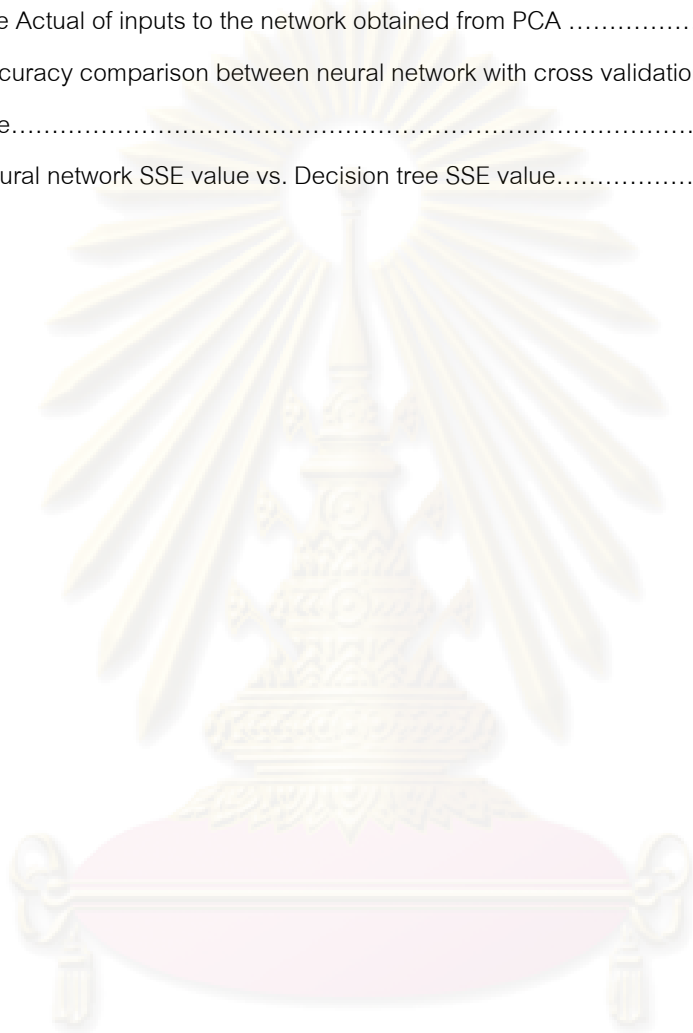
List of Figures

Figure		Page
2.1	The Processes of Telecommunication network.....	6
2.2	A neural network MLP architecture.....	13
2.3	Surface and contour plot of sample neural network.....	15
2.4	Example of Levenberg-Marquardt Training Algorithm.....	16
2.5	Geometric Properties of the first principal component.....	18
2.6	Flow Chart of Prediction.....	21
3.1	Interface of sample input source data.....	22
3.2	Source data selection flow.....	24
3.3	Principal component analysis result.....	27
3.3	Neural Network Architecture for CLV prediction.....	36
4.1	Working Environment Flow Design.....	41
4.2	Actual Values vs. prediction values. (MLP)	42
4.3	Actual values vs. prediction Values after Cross Validation. (MLP).....	42
4.4	Actual Values vs. prediction values (Decision tree)	44
4.5	Actual values vs. prediction Values after Cross Validation (Decision Tree)	44
5.1	Detailed design layouts.....	47

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

List of Tables

Table		Page
3.1	Name Rule of analytic Table.....	24
3.2	The Actual of inputs to the network obtained from PCA	28
4.1	Accuracy comparison between neural network with cross validation and decision tree.....	45
5.1	Neural network SSE value vs. Decision tree SSE value.....	48



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER 1

INTRODUCTION

1.1 Motivation

Customer loss is a serious problem in every industry nowadays. And many researchers have been worked on it. In banking field, insurance field, securities exchanges field, etc, there are many existing literatures. Yet, still just few companies in telecom industry have effective processes and programs in place to evaluate why customers leave. Telecom is a huge and multidisciplinary industry and how to increase customers' loyalty and calculate the customer value at the time is the key point. Any telecom enterprise is motivated to become a tycoon in their field, but stymied in their effort to do so by their technical knowledge. This research has focused on the telecommunication industrial field, because there are just a few of literatures in the industrial world. Second, the research study remedies this limitation by neural network prediction, and it had combined and used the data from real commercial telecom world. The strategic and operational decisions to retain a customer lifetime value (refer to CLV) keep increasing. On another hand, SAS is software with good competitive ability. To achieve the prediction automatically and programming study by using SAS software (SAS Enterprise Miner 5.3) is the first time try coupled with commercial licensed software with academic study.

1.2 Study Description

In research of customer value, customer lifetime value, it is accepted by more and more scholars and enterprise as standard that identifies the customer's value to the company. CLV can be defined as the total present value during the time since the customer started and keeps buying relationship with the company. CLV can be divided into two parts to the present customer. First part, profit the customer has made till now, the second is future profit, namely the total present profit plus with the customer is likely to make in the future. The normal CLV is only the future profit, so the forecasting CLV refers to the profit.

A customer's CLV is typically assessed using customer behavior data from telecoms company database to predict customer behavior and profitability. The traditional direct marketing literature by Berger et al [1] has mentioned a simple model using amounting total data predict CLV. The CLV is the income contributing to profit of the company and it takes off the cost while the company is gaining the income. And what is more important, CLV predicts well the longtime customers' potential selling up in the future to the company, so it can measure the customer's total value in the future objectively and completely. The company can know clearly if a customer is profitable or not by CLV prediction. This research study continues to use the definition. Namely, the value of a customer writes as total profit present with specific time in the future. In the telecom business, the profit of the customer can be embodied by the information of their use. It has exploited the data out from the Data warehouse. All the data come from the Data Warehouse that has been done already by using Data Warehouse Solution for Telecommunications. It has covered of all 4 categories data. The fields are data telecommunications making process were mentioned above. Respectively, they are Revenue, Outbound (namely call out), International Calling, Call Duration for the input variables,

1.3 The Objectives of Research

In this study, the author does advocate to use the model of Neural Network predicting the CLV in telecom industry. The reason why used this model because the data is huge and complexity and tasks make by hand impractical. But if neural networks implemented functions can be inferred for the case from data observations. Second, to understand the performance of neural network, thereby archive the goal more better prediction exactness. Third, the research has also focused on comparison purpose, and building an utilizable decision tree model compare with neural network prediction. Therefore, in this study, another model decision trees performance and exactness will compared with the neural network. It has applied to two different data mining techniques for the prediction. Mainly, the research has employed neural network Multilayer perceptrons with Leveberg-

Marquart algorithm. And the following, it has compared the performance of Decision tree with C4.5 Algorithm.

1.4 Scope of study

Recently, many expert systems and CRM (Customer relationship management) systems and more methodologies have been exploited in industry world, and in commercial world.

The purpose of the CRM has its roots in relationship marketing which is based on the design work by Berry [2]. The IMP Group Relationship marketing improves long run profitability by shifting from transaction based marketing, with its emphasis on winning new customers, to customer retention through effective management of customer relation. In brief, CRM is a “customer centric” business strategy with the goal of maximizing profitability, revenue, and customer satisfaction. Customer Lifetime Value (CLV) can be obtained by calculating total present profits form a customer during the course of customer lifetime. The Estimate lifetime, the number of months a customer may be expected to remain on the network, estimated from the current month. It can be calculated generically for a small segment of customers, based on the typical churn rate for the customers; or more specifically, for each customer, based on their propensity to churn score when this is available. Although, many articles are currently available, CLV in the telecommunication field still has not been properly solved yet, it has been written in Literatures review.

1.5 Literatures Review

There are several existing and popularly used methods in CLV prediction such as present two models forecasting CLV which correspond the two groups customers buying profile, without propose of the mathematical modeling by Kamakura et al. 2005 [3] and Malthouse and Blattberg [4] have quoteworthy research on modeling CLV is one of the MSI research priorities, and Bas Donkers et al. But those kind methods are prone to statistic and probabilistic forecasting as discussed above cannot train the data time after time and also cannot contribute to telecommunication. Actually Gary Cokins [5] SAS white paper mentioned that to treat customers as investments in telecommunication Industry. However, it is simple to say

in commercial way. The author added the argument on to say that one should focus on profitability instead of longevity only, on detail of telecommunication likely to churn if customers have more mobile numbers but not really use. This study discusses the feature our neural network it can learned by itself and the predictive CLV which needs more special knowledge. Yet the neural network has the characteristics of training the known data again and circulating until get the right value. It has good learning rate to known data get best performance, Multi-Layer Perceptron (MLP) is drawn upon typical architecture network model. This model needs adjust value to every input and output pairs, on another hand MLP compare with Radial Basis Function (RBF), normally RBF by using with Gaussian Algorithm used to predictive classification problem, but in this study, we use MLP with Levenberg-Marquardt algorithm, which is preferred for "small" networks (less than one hundred input parameters), when the data increasing in the future it is more efficiency method and feasible method for commercial used.

This study is organized as follows. In Chapter 2, the background knowledge, related works and methodologies are described. The definition and analysis of CLV is discussed. The Algorithm of Decision Tree theory and Multi-Layer Perceptron (MLP) network theory are elaborated. In Chapter 3, Data selection and preparation and implementation for our project have been written. In term of our Project, it was done in a mobile telecom company. It has been explored the feasibility of the proposed Decision Tree and MLP Neural Network method applied to CLV prediction and the implementation results are given at the end of this section. Experiment results are brought out in Chapter 4 and conclusions.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER 2

BACKGROUND, METHODOLOGIES AND MODEL DESIGN

2.1 Industry Background

This chapter provides a summary of essential theoretical backgrounds of telecom and prediction technologies that are required in this research. It contains three main sections: Telco company growth and customer campaign, partial business rules, data mining prediction of ways to measure and manage Customer Lifetime Value (CLV) for determining the likely future profit from the customer and related works. Customer acquisition and retention is a concern for all industries, but it is particularly acute in the strongly competitive and now broadly liberalized telecom industry. For the marketing departments of new companies, the major short to medium-term issue is likely to be attracting new customers. However, for the incumbent operators and the more mature market entrants, retaining profitable customers is the number one business plan as oppose to just leave them. The information delivery systems of many telecom companies are reaching a maturity level that allows them to make the step from simple query and reporting of past cancellations towards the semi-automatic creation of predictive models. Companies can assure a more constant flow of revenue and higher profit margins through targeted activities such as the fine tuning of services and promotional messages or gift sets etc.,.

Churn is a process of a customer terminating a specific service which is provided by the telecom company. Previously, we even cannot get the meaning of 'churn' in dictionaries [6], but it is now an ordinary word. It is the common denominator in the world's liberalized telecom industry. It now costs European and US telecom close to 4 billion US dollars each year, and the global cost of customer defection may well approach a staggering 10 billion US dollars. [7] Annual churn rates of 25 to 30 percent are the norm, and carriers at the upper end of this spectrum will get no return on investment on new subscribers. Why? Because it typically takes three years to pay back the cost (approximately 400 US dollars in the United States and 700 US dollars in Europe) of replacing each lost

customer with a new one (customer acquisition). European and Asian markets, in particular, the number of new market entrants is adding to the churn phenomenon. In Europe, 30 new telecoms entered the market in 1998, seeking the 15 percent market share that analysts say is required for them to survive. The growth in the number of subscribers has eased this situation in the past, but as market growth slows and average revenue per user declines, we are likely to see an increase in predatory activity.

The research study introduces churn actions should be taken in place in conditions, some of the terms and concepts involved in the process of accurately predicting which customers are likely to deliver high value and at the same time are likely to exhibit a high propensity to change suppliers.

Firstly, to understand a general telecom industry operation process is necessary, so we have referenced and might be considering in Figure 2.1.

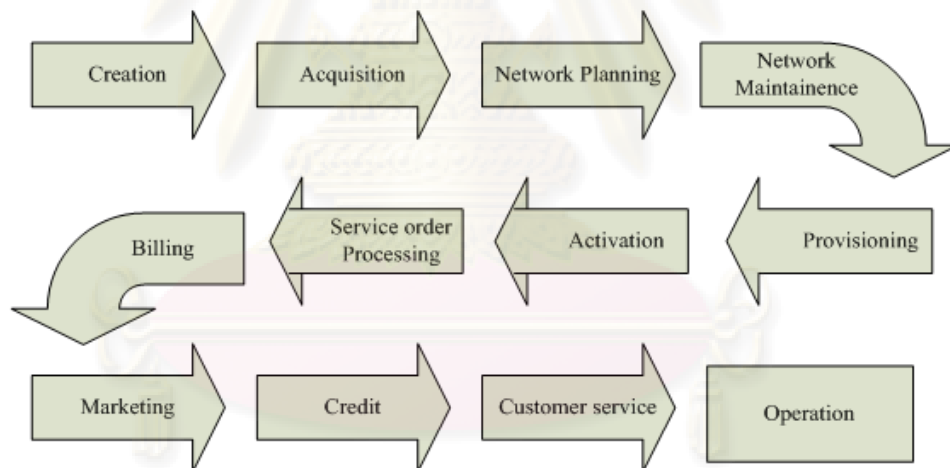


Figure 2.1: The Processes of Telecommunication network [8]

The figure is said to be more visual, it carries out the whole process of Telecom Company. Since from start, a telecom network started from network creation, acquired a portion of customers. Next, start a basic network planning (starting, growth, maintenance or etc), network maintenance, provisioning (network prepared), and do the activations of network (may be take consideration in more advertising). Keep the customers buying actions and retain the network growth. Afterwards, the billing system output reports and the revenue feedback to the marketing and gains the credit. (Actually, this research study is

something between billing pane and marketing pane.) Next, it improves credit, customer service and turns to the good cycle operation.

2.2 Modeling CLV

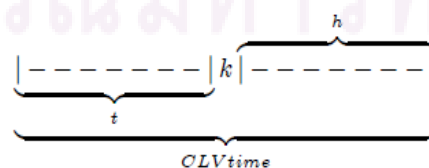
First, the CLV model is formulated to help the finance department in the telecommunication company. The CLV is not easy to implement especially a company which has not get implemented the Business intelligence. Collecting data is another important issue for the CLV. Aim if may takes such a long time starting from the time customers begin opening or using the services until they stop using the services. Different formulas have been proposed to calculate the CLV such as methods done by Berger et al and Bas Donkers et al. The formula for CLV used in this paper derived from Nicolas Glady et al. The concept of the cash flow is integrated in the formula. Cash Flow of a customer i , on t period, product j usage, defined as $CF_{i,j,t} = \pi_j x_{i,j,t}$. Here, $x_{i,j,t}$ is the product j usage during period t of customer i and π_j is the marginal profit by unit of product j usage. The value of $x_{i,j,t+1}$ is computed from the previous $x_{i,j,t}$ as follows:

$$x_{i,j,t+1} = \alpha_{i,j,t} x_{i,j,t} \quad (1)$$

$$\alpha_{i,j,t} = \begin{cases} \text{Growth rate, if } \alpha \gg 1; \\ \text{Retention rate, if } \alpha = 1; \\ \text{Churning rate, if } \alpha \ll 1; \end{cases} \quad (2)$$

With $x_{i,j,t}$ being the product j usage during period t , of customer i , define $\alpha_{i,j,t}$ as the slope of the product usage, hereby, $x_{i,j,t+1} = \alpha_{i,j,t} + x_{i,j,t}$, When much greater than 1 interpreted as a growth rate, as a retention rate when approximately equal 1, and a churning rate for it should be much less than 1.

Parameters given in the formula can be explained with the following chart:



By using (3), the CLV for customer i at $t+h$ for the j products which is the formula present value and with future earning should be represent can be defined as:

$$CLV_{i,j,t,Total} = \sum_{k=1}^h \frac{1 + \prod_{v=0}^{k-1} \alpha_{i,j,t+v}}{(1+r)^k} \pi_j x_{i,j,t} \quad (3)$$

Where,

h = horizon h from the period k

π_j = Marginal profit by unit of product j usage¹

$\alpha_{i,j,t+v}$ = slope of product usage

v = index accounting for the time

i = customer i

k = time period ($k=0, 1, 2 \dots$) ($k=0$, today)

r = discount rate²

Thus, we use CLV_{Total} (according to work of Nicolas Gladys et al) a single product view per customer, which is means *Action* part and *Without Action* part.³ Compact it, we can have,

$$CLV_{i,j,t,Total} = \frac{1}{(1+r)^h} (1 + (\alpha_{i,j,t})^h) \pi_j x_{i,j,t} \quad (4)$$

Suppose we set discount rate = 0.72% of product j and horizon h = 3 months, and we set $\alpha_{i,j,t} = 1$ as a normal user and then we have the $CLV_{i,j,t,Total} \approx 1.96$ times the Cash Flow ($CF_{i,j,t}$) in this sum of amount 6 months which are horizon 3 months to predict and 3 months data we have. Months of future of horizon h namely 0.96 time Cash Flow.

¹ It may depend on the type of customer namely i . Customer may have preferential conditions according to their status. For simplicity, we just consider an average product margin.

Marginal profit = Amount Total Revenue - Variable Costing.

In telecommunications, lease satellite, add new devices etc, can be the Variable Costing.

² Same above, an assumed constant which we can consider the discount as if bankroll obtained.

³ The data which form data warehouse due to a single product view per customer.

2.3 Methodologies

2.3.1 MLP Neural Networks

2.3.2 Introduction to Neural Networks

Neural network is not a new technology; it has been around the world since 1943. McCulloch and Pitts gave birth to the field of artificial neural networks. What is the neuron? What is a neural network? First, the complexity and diversity in nervous systems is dependent on the interconnections between neurons in our body. Neurons exist in a number of different shapes and sizes and can be classified by their morphology and function. The anatomist Camillo Golgi grouped neurons into two types; type I with long axons used to move signals over long distances and type II without axons. Type I cells can be further divided by where the cell body or soma is located. The basic morphology of type I neurons, represented by spinal motor neurons, consists of a cell body called the soma and a long thin axon which is covered by the myelin sheath. Around the cell body is a branching dendritic tree that receives signals from other neurons. The end of the axon has branching terminals (axon terminal) that release transmitter substances into a gap called the synaptic cleft between the terminals and the dendrites of the next neuron. According to this certain theory things, when we are talking about a neural network, we should more properly say “artificial neural network” (ANN), because that is what we meant most of the time. Biological neural networks are much more complicated than the mathematical models that we use for ANNs. But it is customary to be lazy and drop the “A” or the “artificial”.

There is no universally accepted definition of an NN. But perhaps most people in the field would agree that an NN is a network of many simple processors (“units”), each possibly having a small amount of local memory. The units are connected by communication channels (“connections”) which usually carry numeric (as opposed to symbolic) data, encoded by any of various means. The units operate only on their local data and on the inputs they receive via the connections. The restriction to local operation is often relaxed during training.

Some NNs are models of biological neural networks and some are not, but historically, much of the inspiration for the field of NNs came from the desire to produce artificial systems capable of sophisticated, perhaps “intelligent”, computations similar to those that the human brain routinely performs, and thereby possibly to enhance our understanding of the human brain.

A neural network is first and foremost a graph, with patterns represented in terms of numerical values attached to the nodes of the graph, and transformations between patterns achieved by simple message-passing algorithms. Many neural network architectures, however, are also statistical processors, characterized by making particular probabilistic assumptions about data [9]. This conjunction of graphical algorithms and probability theory is not unique to neural networks, but characterizes a wider family of probabilistic systems in the form of chains, trees, and networks that are currently studied throughout AI.

Neural networks have found a wide range of applications, the majority of which are associated with problems in pattern recognition and control theory. In this context, neural networks can best be viewed as a class of algorithms for statistical modeling and prediction. Based on a source of training data, the aim is to produce a statistical model of the process from which the data are generated, so as to allow the best predictions to be made for new data.

A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for usage [10]. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.
2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

Most neural networks involve combination [11], activation, error, and objective functions.

Combination functions: Each non-input unit in a neural network combines values that are fed into it via synaptic connections from other units, producing a single value

called the “net input”. For the function that combines values, it is called the “combination function”. The combination function is a vector-to-scalar function. Most Neural networks use either a linear combination function (as in MLPs) or a Euclidean distance combination function (as in RBF networks).

Activation functions: Most units in neural networks transform their net input by using a scalar-to-scalar function called an “activation function”, yielding a value called the unit’s “activation”. Except possibly for output units, the activation value is fed via synaptic connections to one or more other units. The activation function is sometimes called a “transfer”, and activation functions with a bounded range are often called “squashing” functions, such as the commonly used tanh (hyperbolic tangent) and logistic ($1/1+\exp(-x)$) functions. If a unit does not transform its net input, it is said to have an “identity” or “linear” activation functions.

Error functions: Most methods for training supervised networks require a measure of the discrepancy between the networks output value and the target value. The difference between the target and output values is called the “residual” or “error”. This is NOT the “error function”. The residual can be either positive or negative, and negative residuals with large absolute values are typically considered just as bad as large positive residuals. Error functions, on the other hand, are defined so that the bigger is the worse.

Objective functions: The objective function is what you directly try to minimize during training. Neural network training is often performed by trying to minimize the total error or the average error for the training set. However, minimizing training error can lead to over-fitting and poor generalization if the number of training cases is small relative to the complexity of the network. A common approach to improving generalization error is regularization function. If no regularization function is used, the objective function is equal to the total or average error function.

Neural networks offer a computational approach that is quite different from conventional digital computation. Digital computers operate sequentially and can do arithmetic computation extremely fast. Biological neurons in the human brain are extremely slow devices and are capable of performing a tremendous amount of computation tasks necessary to do everyday complex tasks,

commonsensical reasoning, and dealing with fuzzy situations. The underlining reason is that, unlike a conventional computer, the brain contains a huge number of neurons, information processing elements of the biological nervous system, acting in parallel. Neural networks are thus a parallel, distributed information processing structure consisting of processing elements interconnected via unidirectional signal channels called connection weights.

2.3.3 MLP Neural Network Architecture

In the architecture of neural networks, typically, the network consists of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. These neural networks are commonly referred to as multilayer perceptrons (MLPs). Many applications based on MLP neural network have successfully solved the problems in related fields.

A multilayer perceptron has three distinctive characteristics:

1. The model of each neuron in the network includes a nonlinear activation function. The important point to emphasize here is that the nonlinearity is smooth (i.e., differentiable everywhere), as opposed to the hard-limiting used in Rosenblatt's perceptron [12]. A commonly used form of nonlinearity that satisfies this requirement is a sigmoid nonlinearity defined by the logistic function:

$$y = \frac{1}{1 + \exp(-v_j)}, \quad (5)$$

where v_j is the induced local field (i.e., the weighted sum of all synaptic inputs plus the bias) of neuron j , and y_j is the output of the neuron. The presence of nonlinearities is important because otherwise the input-output relation of the network could be reduced to that of a single-layer perceptron. Moreover, the used

of the logistic function is biologically motivated, since it attempts to account for the refractory phase of real neurons.

2. The network contains one or more layers of hidden neurons that are not part of the input or output of the network. These hidden neurons enable the network to learn complex tasks by extracting progressively more meaningful features from the input patterns (vectors).

3. The network exhibits high degrees of connectivity, determined by the synapses of the network. A change in the connectivity of the network requires a change in the population of synaptic connections or their weights.

Figure 2.2 displays the network diagram that represents the corresponding statistical model of a MLP multiple layer perceptron architecture consisting of one input layer with three inputs, neurons or units (X_1, X_2, X_3) with three input weights (W_1, W_2, W_3) going into a single hidden layer with one hidden unit and an activation function that is connected to a single output unit.

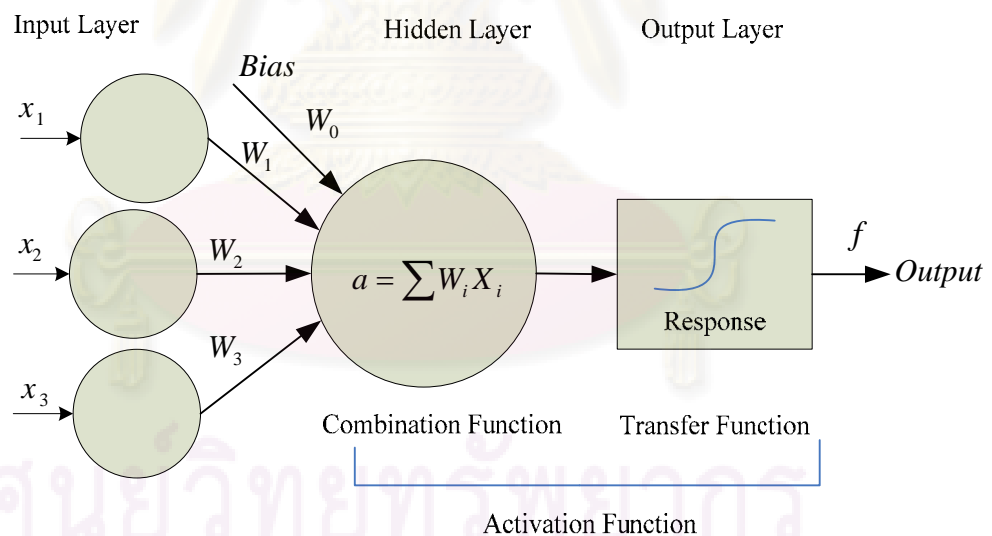


Figure 2.2: A neural network MLP architecture. [11]

Each layer is made up of "units". A unit is synonymous with the term "neuron" in the literature. It is the smallest computational entity in the network. The set of inputs of the neuron generally includes a specific input, a termed bias, the value of which is constant, equal to 1. The expression of output of the neuron is

$$y = f[w_0 + \sum_{j=1}^{n-1} w_j x_j], \quad (6)$$

MLPs are general-purpose, flexible, nonlinear models that, given enough hidden neurons and enough data, can approximate virtually any function to any desired degree of accuracy. In other words, MLPs are universal approximators. MLPs can be used when you have little knowledge about the form of the relationship between the independent and dependent variables.

Another thing which we need to know with regards to the MLP neural network that is Network Convergence:

1. Convergence is declared when the specified error function stops improving.
2. Convergence is declared if the magnitude of the parameters stops changing substantially.
3. Convergence is declared if the gradient has no slope, implying that a minimum has been reached.

2.3.4 Learning Algorithms for MLP Neural network

Estimating the unknown parameters in neural network called learning, considering the size of the inputs data in this research, Levenberg-Marquardt Algorithm suitable for medium sized data is chosen to optimize the learning process. The Jacobian matrix contains the second derivatives of the error function with respect to the weight estimates. Levenberg-Marquardt Algorithm provides a numerical solution to the problem of minimizing a function, generally nonlinear, over a space of parameters of the function. These minimization problems arise especially in least squares curve fitting and nonlinear programming.

2.3.5 Levenberg-Marquardt Algorithm

The error function of linear model is always a parabola. For nonlinear models like neural networks, however, it is a complex landscape consisting of deep valleys and steep cliffs. Numerical optimization methods use (local) features of this error surface to search for the error minimum in a principled way.

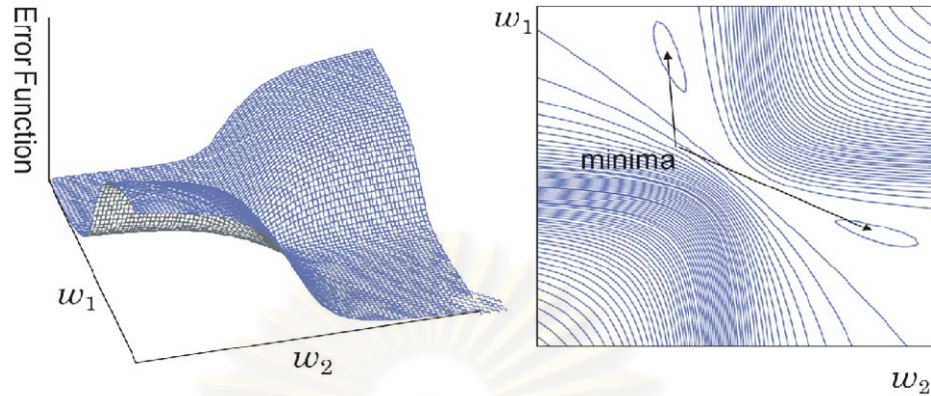


Figure 2.3 (a) Surface plot

Figure 2.3 (b) Contour plot

Figure 2.3 Surface and contour plot of sample neural network

Figure 2.3 (a) shows a sample of the surface plot of the logistics activation function and two inputs node (w_1 and w_2). Figure 2.3 (b) displays the contour plot of the local minima direction.

Afterward is the equation part. If the error function uses deviance (maximum likelihood), the task of minimizing the deviance can be cast as a least squares problem using residuals [13]. This means that the update rule (or learning algorithm) can use a Jacobian (first derivative) matrix, \mathbf{J} , to calculate the update. Calculation of the Jacobian matrix is computationally less intensive than calculating the Hessian. The resulting LM Algorithm update rule is shown below.

$$\delta^{(t)} = -(\mathbf{J}^{(t)'}\mathbf{J}^{(t)} + \lambda^{(t)}\mathbf{I})^{-1}\mathbf{J}^{(t)'}\mathbf{r}^{(t)} \quad (7)$$

Here $\mathbf{J} = \partial\mathbf{r} / \partial\mathbf{w}'$ is a Jacobian (first derivative) matrix of partial residuals with respect to the weights. Given Jacobian matrix, \mathbf{J} , the gradient in the Newton step can be approximated by the equation $\mathbf{J}^{(t)'}\mathbf{r}^{(t)}$, and its Hessian is approximated by $\mathbf{J}^{(t)'}\mathbf{J}^{(t)} + \lambda^{(t)}\mathbf{I}$. Notice that the matrix $\mathbf{J}^{(t)'}\mathbf{J}^{(t)}$ is first augmented by adding a multiple of the identity matrix $\lambda^{(t)}\mathbf{I}$ before inverting the result. This keeps the diagonal of $\mathbf{J}^{(t)'}\mathbf{J}^{(t)}$ positive, creating a hyper-elliptical neighborhood where a quadratic approximation is adequate. The Levenberg-Marquardt method is the default for problems with less than 100 weights.

Figure 2.4 illustrates the possible steps LM algorithm steps. The algorithm could take on just few steps.

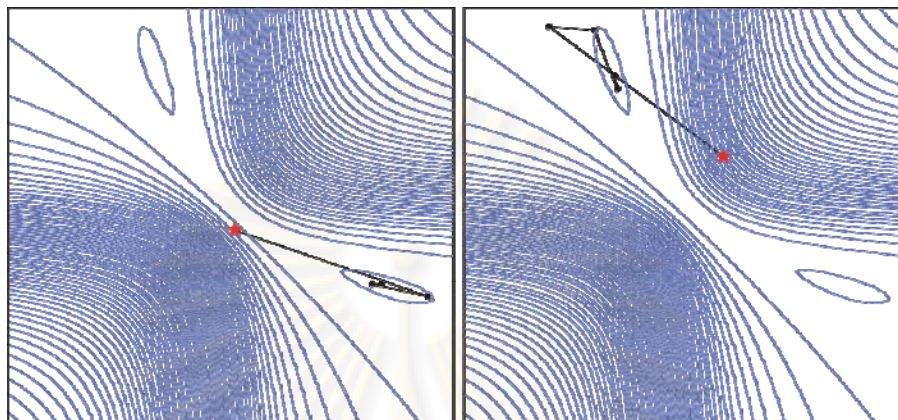


Figure 2.4 Example of Levenberg-Marquardt Training Algorithm.

2.4 Dimension Reducing Methods

In predictive modeling, there are two reasons for eliminating variables from the analysis: redundancy and irrelevancy. In other words, most of the modeling selection routines are designed to minimize input redundancy and maximize input relevancy. At times, the statistical model can potentially consist of an enormous number of input variables in the model to predict the target variable. Therefore, irrelevancy in some of the input variables might not provide a sufficient amount of information in describing or predicting the target variable. Redundancy in the input variables suggests that a particular input variable does not provide any added information in explaining the variability in the target variable that has not already been explained by some other input variables which are already in the model.

2.4.1 Principal Components Analysis

The purpose of principal components analysis is both data reduction and interpretation of a linear combination of the input variables in the data that best explains the covariance or correlation structure. The analysis is designed to reduce the dimensionality of the data while at the same time preserving the structure of the data. The advantage is that a smaller number of linear independent variables, or

principal components, without losing too much variability in the original data source, where each principal component is a linear combination of the input variables in the model [14].

Principal components analysis is based on constructing an independent linear combination of input variables in which the coefficients (eigenvectors) capture the maximum amount of variability in the data. Typically, the analysis creates as many principal components as there are input variables in the data set in order to explain all the variability in the data where each principal component is uncorrelated to each other. This solves one of the two problems in the statistical model. First, the reduction in the number of input variables solves the dimensionality problem. Second, this will solve co-linearity among the input variables since the components are uncorrelated to each other. The goal of the analysis is first finding the best linear combination of input variables with the largest variance in the data, called the first principle component. The basic idea is to determine the smallest number of the principal components to account for the component consists of a line that is perpendicular to each data point by minimizing the total squared distance from each point that is perpendicular to the line. This is analogous to linear regression modeling, which determines a line that minimizes the sum-of-squares vertical distance from the data points that is always perpendicular to the axis of the target variable. Principal components analysis is designed so that the first principal component is perpendicular, orthogonal, and uncorrelated to the second principal component, with the second principal component following the first principal component in explaining the most variability in the data. The number of principal components to select is an arbitrary decision. This can be achieved by observing the magnitude of the eigenvectors within each principal component.

The principal components are comprised of both the eigenvalues and eigenvectors that are computed from the variance-covariance matrix. The eigenvalues are the diagonal entries in the variance-covariance matrix. The principal components are the linear combination of the input variables with coefficients equal to the eigenvectors of the corrected variance-covariance matrix.

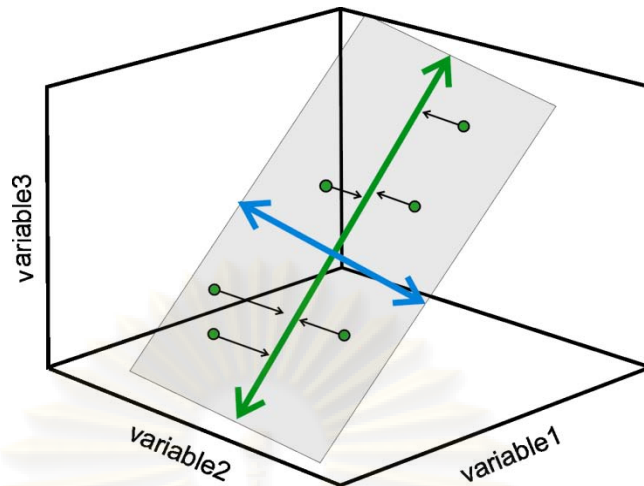


Figure 2.5: Geometric Properties of the first principal component.

Principal components analysis is basically designed to determine the minimum distance between all the data points in the model. This is illustrated in the figure 2.5. In the figure 2.5, the line accounts for the largest majority of variability in the data where the principal components line is not parallel to any of the three variable axes. The inclined plane represents the distribution of the data points in a three-dimensional plane and the display the two-dimensional scatter plots, where there is a positive correlation between all pairs of variables in the data.

2.5 Decision Tree

Decision tree modeling method is based on non-parametric statistics. And it is easier to understand. Decision trees are statistical models designed for supervised prediction problems. Supervised prediction is a generic term that encompasses many similar tasks such as predictive modeling, pattern recognition, multiple regression, discriminant analysis, multivariate function estimation, and supervised machine learning. In supervised prediction, a set of input variables (predictors) is used to predict the value of a target variable (outcome). The mapping of the inputs to the target is a predictive model. The data used to estimate a predictive model is a set of cases (observations, examples) consisting of values of the inputs and target. The fitted model is typically applied to new cases where the target is unknown.

Handwriting recognition is a classic application of supervised prediction. The example data set is a subset of the pen-based recognition of handwritten digits data [15], available from the UCI repository (Blake et al 1998). The cases are digits written on a tablet of sensitive-pressured. The input variables measure the position of the pen. They are scaled to be between 0 and 100. Two of the original 16 inputs are shown (X_1 and X_{10}). The target is the true written digit (0-9). This subset contains the 1064 cases corresponding to the three digits 1, 7, and 9. Each case represents a point in the input space. (The data were jittered for display because many of the points overlap.)

A decision tree is thus called because the predictive model can be represented in a tree-like structure. A decision tree is read from the top down starting at the root node. Each internal node represents a split based on the values of one of the inputs. The inputs can appear in any number of splits throughout the tree. Cases move down the branch that contains its input value. In a binary tree with interval inputs, each internal node is a simple inequality. A case moves left if the inequality is true and right otherwise. The terminal nodes of the tree are called leaves. The leaves represent the predicted target. All cases upon reaching a particular leaf are given the same predicted value. When the target is categorical, the model is called a classification tree. The leaves give the predicted class as well as the probability of class membership.

2.5.1 Decision Tree with C4.5 Algorithm

Decision Trees are supervised learning algorithms for data mining that use class-labeled training tuples to classify data. The algorithm and concept of decision trees was developed by J. Ross Quinlan [16], [17]. The major decision tree algorithm that we use in this experiment is C4.5.

Among classification algorithms, decision tree based upon a C4.5 algorithm deserves a special mention for several reasons. It represents the result of research in machine learning that traces back to the ID3 system [18]. The result of T.S. Lim [19] shows that the C4.5 tree-induction algorithm provides good

classification accuracy and is the fastest among the compared main-memory algorithms for machine learning and data mining.

At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. For help to avoid overfitting the formula as following:

$$\text{Split Information } (S, A) = - \sum \frac{|S_i|}{S} \log_2 \frac{|S_i|}{S} \quad (8)$$

$$\text{Gain Ratio } (S, A) = \frac{\text{Gain } (S, A)}{\text{Split Information } (S, A)} \quad (9)$$

Where, Split information denoted the entropy with regard to the value of distribution of attribute A , and Gain Ratio is the information gain with regard to attribute A .

The implementation steps are as follow:

1. Calculate the gain ratio for current training dataset.
2. Choose the gain ratio maximal value of attribute A_i as root node.
3. Separate the subset According to the value of A_i from training dataset.
4. In every subset, Calculate gain ratio of the remaining attributes, choose the maximal value as the test attribute of node.
5. Recursive running above steps until all of the subset achieve to be a node (no more split) or all attributes have been run out.

2.6 Prediction Process Flow Design

Usually, the data in every special field are complicated with much noisy and it is almost impossible to get a successful predictive modeling performance with initially setting parameters. The prediction represents your “best guess” for the target given a set of input measurements. And the predictions are based on associations “learned” from the training data by prediction model.

In this research study, the author proposes a compared Decision Tree Modeling and MLP neural network modeling based on the idea of choosing data preparation steps and training algorithms, selecting the comparable solution lead in to go on the good prediction process result. In data preparation step, we will use PCA (Principal Components Analysis) to reduce the dimension of the inputs data. And then, Decision tree model we should be taken into consideration of C4.5 Algorithm, For neural network model, the algorithm Levenberg-Marquardt Algorithm will be implemented concerned by the real training parameter counts.

As follows, a workflow is designed for the whole forecasting process. Firstly, the input data loaded from the production data source in everyday will be transformed in 4 categories, 176 variables imported to the data mart for every hour to match the related customer actions, whatever made a call, or paid a bill. Then, this dataset will be tagged as the first source dataset. The second source dataset which used for forecasting is just from the transaction of averaging every of the six transposed columns. Towards physical cleansing mostly aim at the missing data. (Imputation step will be mentioned later in chapter 3), after the step, the technique of PCA and the standard variable selection technique based on Step-wise regression will be used separately. Two different MLP neural networks models will be trained individually with the two kinds of prepared data from the above steps. Finally, an accuracy evaluation would be made to help to choose the best model with the highest performance. The forecasting process flow design is given in the following Figure 2.7.

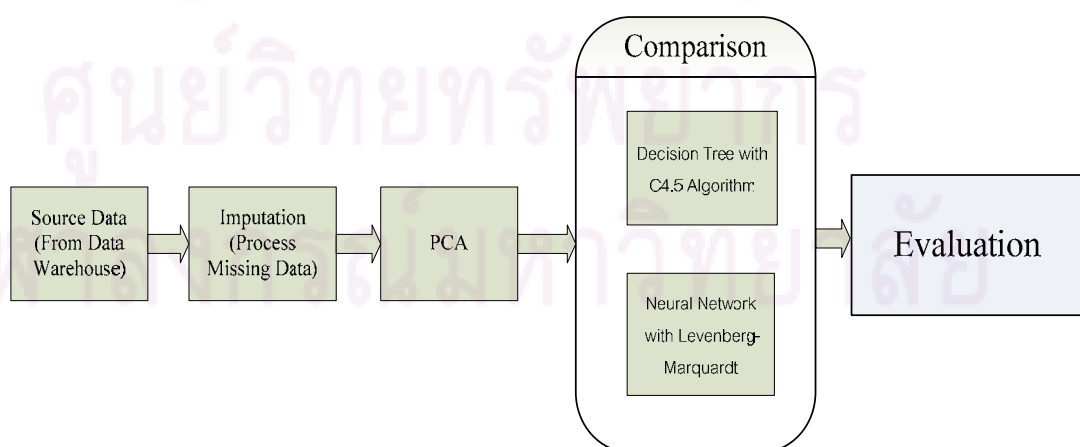


Figure 2.6: Flow Chart of Prediction.

CHAPTER 3

EXPERIMENTAL APPLICATION

3.1 Experimental Data

In telecom industry, the workflow can be controlled and monitored in real-time by the production system. Most data parameters either in their ERP system or RDBMS can be checked or loaded to prepare for prediction. Meanwhile, IT analysts and IT operators can easily observe the steps of each parameter during the processing. Therefore, database supports storage of huge numbers of records of source data for the telecom research. Intuitively, the visual sample data source interface (Figure 3.1).

	TOT_PRO_CNT	TOT_ACTV_PRO_CNT	TOT_PRO_DRO_PPED_CNT	SUBS_T ENURE	TOT_REV_OC_VOICE_AMT	TOT_REV_DATA_AMT	TOT_RE V_SMS_AMT	TOT_REV_LOC_AMT	TOT_REV_AMT	AVG_REV_AMT	TOT_REV_VO ICE_AMT	AVG_REV_VOICE_AM T
2	0.6666666666	4	-1	8.5	-0.786315995	-1	-1	-0.417892805	-0.419550808	-0.419550808	-0.786315995	-0.786315995
3	3.3333333333	10	0	9.5	0.0146206894	0	0	0.5778789494	0.5762329891	0.5762329891	0.0146206894	0.0146206894
4	3.3333333333	5	1.6666666666	8	0	0	0	0.5625131936	0.5609109994	0.5609109994	0	0
5	3.3333333333	5	1.6666666666	8	0	0	0	0.5625131936	0.5609109994	0.5609109994	0	0
6	6.6666666666	5	5	8	0	0	0	0.5625131936	0.5609109994	0.5609109994	0	0
7	3.3333333333	5	1.6666666666	8	0	0	0	0.5625131936	0.5609109994	0.5609109994	0	0
8	1.6666666666	5	0	7.5	0	0	0	0.5625131936	0.5609109994	0.5609109994	0	0
9	5	5	3.3333333333	7.5	0	0	0	0.5625131936	0.5609109994	0.5609109994	0	0
10	1.6666666666	5	0	7.5	0	0	0	0.5625131936	0.5609109994	0.5609109994	0	0
11	1.6666666666	5	0	7.5	0	0	0	0.5625131936	0.5609109994	0.5609109994	0	0
12	3.3333333333	5	1.6666666666	7.5	0	0	0	0.3758440046	0.3747734962	0.3747734962	0	0
13	3.3333333333	5	1.6666666666	7.5	0	0	0	0.5625131936	0.5609109994	0.5609109994	0	0
14	3.3333333333	5	1.6666666666	6	0	0	0	0.3758440046	0.3747734962	0.3747734962	0	0
15	3.3333333333	5	1.6666666666	6	0.1331501583	0	0	0.6922921285	0.6903202876	0.6903202876	0.1331501583	0.1331501583
16	3.3333333333	5	1.6666666666	6	0.1219558887	0	0	0.6814886133	0.6795455496	0.6795455496	0.1219558887	0.1219558887
17	3.3333333333	5	1.6666666666	6	0.3343160495	0	0	0.8884701145	0.8839470145	0.8839470145	0.3343160495	0.3343160495
18	3.3333333333	5	1.6666666666	6	0.4790563328	0	0	1.0261856782	1.0232628155	1.0232628155	0.4790563328	0.4790563328
19	3.3333333333	5	1.6666666666	6	0	0	0	0.3758440046	0.3747734962	0.3747734962	0	0
20	3.3333333333	5	1.6666666666	6	0	0	0	0.3758440046	0.3747734962	0.3747734962	0	0
21	3.3333333333	5	1.6666666666	6	0	0	0	0.3758440046	0.3747734962	0.3747734962	0	0
22	3.3333333333	5	1.6666666666	6	0.3936878596	0	0	0.9437818801	0.9410937264	0.9410937264	0.3936878596	0.3936878596
23	3.3333333333	5	1.6666666666	6	0	0	0	0.6890473419	0.6870847431	0.6870847431	0	0
24	3.3333333333	5	1.6666666666	6	0.1839948551	0	0	0.7413710914	0.73925946	0.73925946	0.1839948551	0.1839948551
25	3.3333333333	5	1.6666666666	6	0	0	0	0.563786007	0.5621602444	0.5621602444	0	0
26	3.3333333333	5	1.6666666666	6	0.0009993724	0	0	0.5647306733	0.563122163	0.563122163	0.0009993724	0.0009993724
27	3.3333333333	5	1.6666666666	6	0	0	0	0.3758440046	0.3747734962	0.3747734962	0	0
28	3.3333333333	5	1.6666666666	6	0	0	0	0.3758440046	0.3747734962	0.3747734962	0	0
29	3.3333333333	5	1.6666666666	6	0	0	0	0.3758440046	0.3747734962	0.3747734962	0	0
30	3.3333333333	5	1.6666666666	6	0.0585800993	0	0	0.7957056064	0.7934392151	0.7934392151	0.0585800993	0.0585800993
31	3.3333333333	5	1.6666666666	6	0	0	0	0.3758440046	0.3747734962	0.3747734962	0	0

Figure 3.1: Interface of sample input source data.

The existing methods aforementioned of CLV prediction proposed by Kamakura et al, and Malthouse and Blattberg are prone to statistic and probabilistic forecasting. As we discussed in chapter 1 and 2, they cannot train the data time after time and also cannot contribute to the industry. The previous studies focused only on the length of time that the customers stay with the company.

Although this concern is essential but, in the real situation, the profitability should be another considered factor involved in the decision making. Predicting the CLV of a customer requires several related factors from the customer. The prediction problem is transformed to a problem of creating a mapping function having all related features as its variables. The value computed from this function will denote the CLV. Since this predicting function involves many variables, a simple nonlinear regression technique may be inappropriate due to the difficulty in selecting a proper polynomial in high dimensional space.

It need to be treated the telecom customers as the investment. The profit of the customer can be embodied by the information of their usage. All of the data come from the Data Warehouse that has been done already by using Data Warehouse Solution for Telecommunications, For the Data warehouse and real-time working mechanism, we have 176 parameters covering all 4 categories of data of telecommunications making process mentioned above, (Revenue, Outbound (namely call out) ,International Calling, Call Duration) for the input variables. All data variables from data warehouse are users billing data and owner revenue data in 2006 till 2007. The details of the relation between the source data system and the research final prediction result as follow.

In figure 3.2, it need to be processed the steps of extract, transform and load data, and the storage data would be normalized, or it might be has campaign data from customer segmentation campaign, loaded to data mart and final prediction.

For more information with regard to input variables, the author has collected them in Appendix A.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

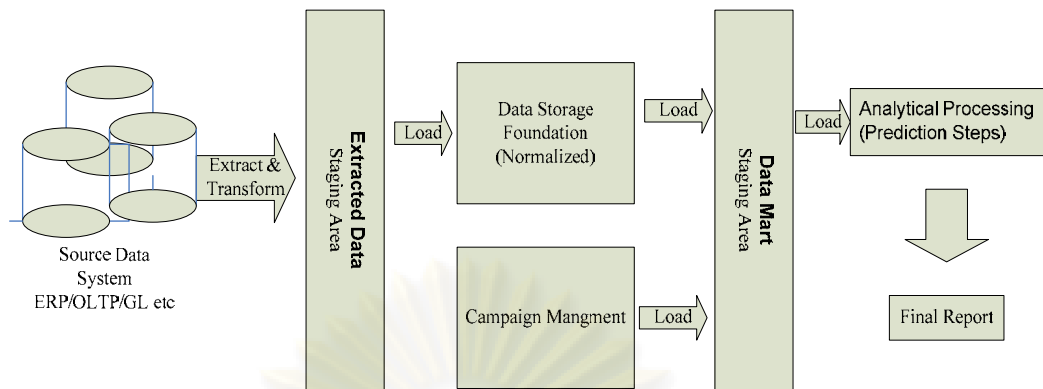


Figure 3.2: Source data selection flow [20]

3.2 Data Preparation Steps

Before performing the following variable selection routines in order to develop a predictive model, it is advisable to perform the various preprocessing routines such as filtering the extreme values from the training data set, transforming all the variables that are not normally distributed, and imputing missing values and replacing incorrect non-missing value. The data of telecom billing system are seldom missed data, even it has, it just like TOT_OB_CALL_INTL_ROAM_CNT, it is sum of 6 months outbound calls using roaming internationally. If customer had not used the internationally roaming, it will be blank value in the system, so we just put them as zero.

PREFIX	EXPLANATION
TOT	Total Value
AVG	Average Value
PROPN	Proportion
PCT	Percent
SECOND AFFIX	
REV	Revenue
OB	Outbound (namely call out)
INTL	International
DUR	Duration

THIRD AFFIX AND SUFFIX	
WKEND	Weekend
WK	Week
LOC	Local
PRD	Period
CNT	Counter of number
AMT	Amount
BASE 1	Base 1 month
BASE 2	Base 2 months
SMS	SMS Service
VOICE	Voice Service
DATA	Data Service
NON-VOIVE	Non-voice Service

Table 3.1 Name Rule of analytic Table.

Data Preparation is a complex step to neural network model. To build a successful predictive model you must “unambiguously”. Define an analytic objective. The predictive model serves as a means of fulfilling the analytic objective. That is, limiting the number of input variables in the model in order to reduce the effect in the “curse of dimensionality” in avoiding undesirable bad local minimums and accelerate convergence to the minimization process. And yet, retaining as much of the relevant information as possible with respect to the input variables included in the model that best explains the output responses. Also, exclude the outliers or extreme values from the analysis and perform transformations to achieve “formality trainable” in the input and target variables.

3.3 Input Refining

3.3.1 Data Normalization

The dataset ought to standardize the input variables in the model to assure convergence in the optimization process and if the model is unsatisfactory then a transformation might be advised. If one input has a range of 0 to 1, while

another input has a range of 0 to 1,000,000, then the contribution of the first input to the distance will be overwhelmed by the second input, it lead into no movements result on reference axis. So it is essential to rescale the inputs so that their variability reflects their importance, or at least is not in inverse relation to their importance. For lack of better prior information, it is common to standardize each input to the same range or the same standard deviation.

Standardizing either input or target variables tends to make the training process better behaved by improving the numerical condition of the optimization problem and ensuring the various default values involved in initialization and termination are appropriate.

Min-max normalization performs a linear transformation on the original data [21]. Suppose that \min_A and \max_A are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, v , of A to v' in the range $[\text{new_min}_A, \text{new_max}_A]$ by computing

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A. \quad (10)$$

If the new data range $[0, 1]$, the above can brief to:

$$v' = \frac{v - \min_A}{\max_A - \min_A}. \quad (11)$$

Suppose that the maximum and minimum values for the attribute a customer monthly charged payment (TOT_REV_CHARGED_MONTHLY_AMT) are 2,578 Baht and 38.6, respectively. We would like to map payment to the range $[0, 1]$.

By min-max normalization, a value of 497 Baht for income is transformed to:

$$\frac{497 - 38.6}{2578 - 38.6} (1 - 0) + 0 = 0.181.$$

Min-max normalization preserves the relationships among the original data values. It will encounter an “out-of-bounds” error if a future input case for normalization falls outside of the original data range for A.

In this research, two datasets were firstly normalized by min-max normalization into the range of $[0, 1]$ that will benefit for the process weight balance and the Decision Tree C4.5 Algorithm and Levenberg-Marquardt training algorithm.

3.3.2 Input Reduction

In this study, PCA (Principal Components Analysis) will be adopted to optimize the source datasets. Because of the large number of the first dataset columns and many variables in the same parameter, Principal Components Analysis is based on constructing an independent linear combination of input variables in which the eigenvectors capture the maximum amount of variability in the original dataset.

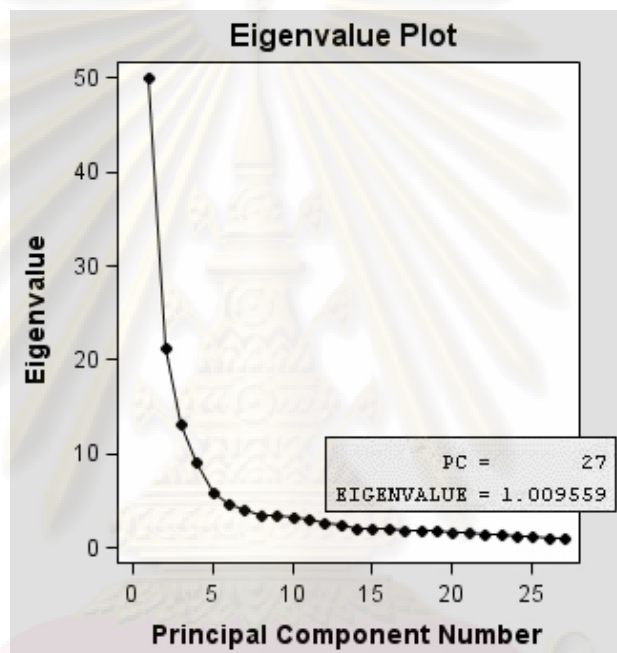


Figure 3.3: Principal component analysis result.

Originally, they are 176 input attributes, after we took out personal information 166 attributes remaining; they used the rule to naming the table columns shown on Table I. To make a column, pick the name from table prefix, second affix and third affix and suffix. For example TOT_REV_SMS_AMT namely Total revenue SMS amount charge. The table columns combined the prefix affix and suffix. Therefore, the PCA (Principal Component Analysis) technique is used to prune the input attributes which used eigenvalues greater than 1 decision. This is often called the Kaiser Guttman criterion. At result, the Eigenvalue of 27th PC. Therefore, 27 inputs attributes are obtained by Correlation Matrix PCA columns and original columns and shown on Table 3.2. The PCA Eigenvalue Chart is shown in Figure 3.3.

1	Total outbound call duration (locally)
2	Total international call duration
3	Total outbound data service base on last month
4	Duration per call of Average outbound local call
5	Total Number of Full pay
6	Proportional call on week end calculate weekend call duration divided by total call duration last 6 months
7	Total revenue SMS amount
8	Percentage of outbound call number per month
9	Proportional outbound local calls duration of per call
10	Proportional duration outbound local on weekend calculate by duration present month divided by average duration last 6 months
11	Total numbers used promotion
12	Total number partial pay
13	Percent in the number changed outbound data services
14	Total numbers of days overdue day of payment
15	Percent change outbound call duration.
16	Proportional voice service revenue amount calculated by present month voice call amount divided by last 6 months revenue voice call amount
17	Total times of overdue 30 days no obtained payment
18	Total number (times) dropped promotion plan (maybe changed to another)
19	Total Number of Payment last 6 months
20	Total outbound international calls on weekend base on last 2 months
21	Percent change in the number of outbound call
22	Percent change in the number of SMS services
23	Average of duration outbound call on weekend last 6 months
24	Percent change in the number of outbound call
25	Total revenue charge of present month
26	Proportional Partial Pay calculated by present month partial payment divided by Total number of partial payments made over last 6 months
27	Total number of Miss payment overdue date

Table 3.2 The actual of inputs to the network obtained from PCA.

3.3.3 Data Partition

Data partition, we have partition the dataset into training, testing datasets. The training data is used for preliminary model fitting. The testing dataset is an additional holdout dataset that you can use for model assessment. The partition part was a random sampling step.

The testing data has only to give a final honest estimate of generalization. Consequently, cases in the test set must be treated in the same way that new data would be treated. In addition, with moderate or large dataset, the computer intensive methods must be added, such as cross validation, we will be bringing them out later. Again, the reason for creating the test data set is that at times the validation data set might generate inaccurate results. Therefore, a test data set might be created in providing an unbiased assessment of the accuracy of the statistical results. The purpose of the validation and test data sets is to fit the model to new data in order to assess the generalization performance of the model.

In summaries, training set is a set of examples used for learning, which is to fit the parameters of the classifier. Validation set is a set of examples used to tune the parameters of a classifier, for example to choose the number of hidden units in a neural network. Test set is a set of examples used only to assess the performance of a fully-specified classifier.

In this Decision Tree model and MLP modeling, the prepared datasets were partitioned into 2 parts, 60% for training set, and 40% for testing set. There are totally 12005 observations for each Preparation datasets, suitable to adopt Levenberg-Marquardt algorithm to optimize the learning process.

3.4 Tuning Decision Tree Models

3.4.1 Maximal Tree

A large decision tree can be grown until every node is as pure as possible. If at least two observations have the same values on the input variables, but different target values, it is not possible to achieve perfect purity. The tree with the greatest possible purity on the training data is the Maximal Classification tree.

The maximal tree is the result of overfitting; it adapts to both the systematic variation of the target (signal) and the random variation (noisy data). It usually does not generalize well on new (noisy data).

Tree complexity is a function of the number of leaves, the number of splits, and the depth of the tree. Determining complexity is crucial with flexible models like decision trees. A well-fit tree has low bias (adapts to the signal) and low variance (does not adapt to the noise). The determination of model complexity usually involves a tradeoff between bias and variance. An underfit tree that is not sufficiently complex has high bias and low variance. In contrast, an overfit tree has low bias and high variance.

3.4.2 The Right-sized tree

Pruning refers to the various methods for selecting tree complexity. Top-down pruning is analogous to forward variable selection in regression. Bottom-up pruning is analogous to backward variable selection.

1. Top-down stopping rules (Pre-pruning)

Forward stopping rules:

- 1) Limit the depth of the tree.
- 2) Limit the amount of fragmentation. For example, do not split a node if the number of cases drops below some threshold
- 3) Statistical significance.

If a chi-squared or F test is used as the splitting criterion, then the p-value is a natural stopping rule. Stop growing if no splits are statistically significant. One problem with this method is that the effects of selection invalidate the distribution theory of the tests. The p-values are typically too small. The Bonferroni adjustments, proposed by Kass (1980), have two uses: (1) to equalize the split selection among inputs with different numbers of potential splits, and (2) to

correct the p-values for the effects of selection. The Bonferroni adjustments are approximations (with trees they are not necessarily conservative approximations). The statistical distributions of the tests are intractable.

Even if you could determine the correct p -value, there is no rational method for deciding what value is significant enough. In scientific applications of statistical inference, the customary superstition is to use significance levels of .05 (logworth = 1.3) or .01 (logworth = 2). In the case, we use the value 0.01 for our experiment.

The Decision Tree node has an additional adjustment that depends on the depth of the split. In a binary tree, the multiplier is 2^d where d is the depth of the split (parent node).

Therefore, we need to set the Split Size against amount of fragmentation. It specifies the smallest number of training observations that a node must have to be considered for splitting. In here, we use customary case 120 for our experiment.

2. Bottom-up selection criteria (post-pruning)

- 1) Grow maximal tree.
- 2) Prune to create optimal sequence of subtrees

In bottom-up (post) pruning, a large tree is grown and then branches are lopped off in a backward fashion using some model selection criterion. The bottom-up strategy of intentionally creating more nodes than will be used is also called retrospective pruning and originated with cost-complexity pruning [22]. For any subtree, T in a tree grown from 1 to n leaves, define its complexity or size (number of leaves) as $|T|$, $R(T)$ as the training set misclassification cost, and α , the complexity parameter. The cost complexity measure $R_\alpha(T)$ is a linear combination of the misclassification cost of the tree and its complexity.

$$R_\alpha(T) = R(T) + \alpha |T|. \quad (12)$$

For a given α , find a nested subtree that minimizes $R(T)$ on each of the training sets. For this topic the solution is the subtree model selection criterion selected the Average Square Error option for interval target.

Therefore, finally, the decision tree architecture employed as follow. Maximum Depth is 10, and Splitting Criterion is entropy cause this for the algorithm C4.5. Split size is 120. Significance Level is 0.01. Another important property is Leaf Size, which is used to specify the minimum number of training observations that are allowed in a leaf node. It was set as 80 in the experiment. Because we were not use multi-way split trees. It can get quite large and bushy. So, splitting is constrained to Leaf Size of at least 80.

3.5 Tuning Neural Networks Models

3.5.1 Preliminary Training

In neural network modeling, there is not well known standard method for computing the initial weight estimates. Therefore, preliminary training is performed before network training, which is designed to determine the most appropriate starting values to be used as the initial weight estimates for the subsequent network training run that is critical to the iterative convergence procedure. Furthermore, preliminary training is used to accelerate convergence in the iteration process, with the idea of avoiding bad local minimums in the error function.

Preliminary training is a compromise between an exhaustive search of the parameter space and doing nothing at all. It can substantially improve model fit. Regardless of the training technique chosen, preliminary training should almost always be used to help avoid the worst local minima.

The preliminary training algorithm is simple:

1. Start at a small number (5 by default) of randomly chosen locations and take a few training steps (10 by default) from each, recording each starting location's final error value.

2. Use the parameter set corresponding to the smallest error value as the starting value for training.

This raises the question of whether it is better to start at many locations and take a few steps, or to start at a few locations and take many steps. The empirically determined answer seems to be that you are better to begin at fewer places and take more steps from each starting point. For instance, if you only had 100 steps in total to “spend,” you would be better to start in 4 different locations and taking 25 steps from each location rather than starting at 25 different locations.

3.5.2 Early Stop

Early stopping is designed to improve generalization in controlling network training by terminating network training once the validation error begins to increase in order to prevent over-fitting to the network model. Early stopping requires an enormous number of hidden layer units in order to avoid bad local minimums.

It penalizes large weights or bumps to the iterative process in the interpolation of the curve. The basic idea in early stopping is to stop the neural network iterative process when the validation error reaches a desirable minimum and avoiding a global minimum usually due to over-fitting in the neural network model.

Early stopping proceeds as follows:

Divide the available data into two separate training and validation sets.

1. Use a large number of Hidden Units
2. Use small random initial values.
3. Use a slow learning rate.
4. Compute the validation error periodically during training.

Stop training when the validation error “starts to go up”.

Early stopping is closely related to ridge regression. If the learning rate is sufficiently small, the sequence of weight vectors on the iteration will approximate the path of continuous steepest descent down the error function. Early

stopping chooses a point along this path that optimizes an estimate of the generalization error computed from the validation set. Ridge regression also defines a path of weight vectors by varying the ridge value. The ridge value is often chosen by optimizing an estimate of the generalization error computed by cross-validation, generalized cross-validation, or boot-strapping.

Considering that the pruning technology of early stopping and preliminary training cannot be used in the same sub-model, we utilize early stopping in Levenberg-marquardt sub-model to decrease the time consuming and avoid local minimum.

Early stopping has several advantages first, fast. Secondly, it can be applied successfully to networks in which the number of weights far exceeds the sample size. Third, it requires only one major decision by the user, what proportion of validation cases to use.

3.5.3 Number of Hidden Units

In MLPs with any of a wide variety of continuous nonlinear hidden-layer activation functions, on hidden layer with an arbitrarily large number of units suffices for the “universal approximation” property.

Therefore, usually a single hidden layer unit is applied to a single input variable in the neural network model. But by adding more hidden units to the model, it will increase the complexity to the network design and can approximate any relatively smooth nonlinear function to any degree of accuracy. One of the most important decisions in network designing is the number of units in the hidden layer. Selecting the correct number of hidden units is an important aspect in producing good generalization performance, which is the main goal in network training. The best number of hidden units to apply to the neural network design depends on the number of input and output variables to the network model, the number of observations in the training data and the noise level in the underlying distribution of the training data, the number of training cases, the amount of noise in the targets, the complexity of the function or classification to be learned, the architecture, the

type of hidden unit activation function, the training algorithm and regularization. In most situations, there is no way to determine the best number of hidden units without training several networks and estimating the generalization error of each. Therefore, it is recommended to fit the network model numerous times with a different number of hidden units and analyzing the various modeling assessment statistics and stop the iterative process when the goodness-of-fit statistics begins to increase.

Generally, the number of hidden units is another important factor. There is no standard method in selecting the appropriate number of hidden units. Selecting too many hidden units will lead to over-fitting [23]; otherwise, too little hidden units will lead to under-fitting. In order to select the proper number of hidden units for each sub neural networks model, there are several researches about the number of hidden units such as Elisseeff's, A. et al. [24] we decided to try some approximate number of hidden units and measured the performance. Considered about the observations and variables number of training data, in the experiment phase, we set the number of hidden units according with method has been mentioned in Elisseeff's paper adopted. The equation to estimate the number of hidden units (h) is given as:

$$h \geq \frac{n-m}{m(k+2)}, \quad (13)$$

The number of inputs defined as k , and h is the number of hidden units. In order to lightly over fit the data, there must be fewer than m cases for each parameter. Normally set to 10. The test sample size interprets as n . As result, 4800 test data test sample as mentioned above, and we can have h properly at 16 hidden units.

3.6 Cross Validation

The Cross Validation is also known as Honest Evaluation. It makes as several different divisions of the observed data into training set and test set. This

is called cross validation [25]. It is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called testing set). To reduce variability in the overall assessment of generalizability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

The whole dataset will be split to several folds such as 5-fold or 10-fold, (in this research study, it used 10-fold), and separate 1-fold as the training set records and test dataset records, 10 training data sets correspond to 1 test data sets. 10 fold may run on a computing cluster. Our idea is to exchange the high error records (such as top 20 high error records) in test datasets with lowest error records (such as 20 lowest error records) which in another network test set. After the exchanging, retrain them until all networks reach lower error.

Finally, the employed networks are three layers, fully connected, feed-forward networks. The Architecture is 27-16-1, this mean that they are 27 nodes for the input layer, the inputs are refer to x_1 till x_{27} , 16 nodes for the hidden layer, refer to h1 till h16 and one output node for the output layer.

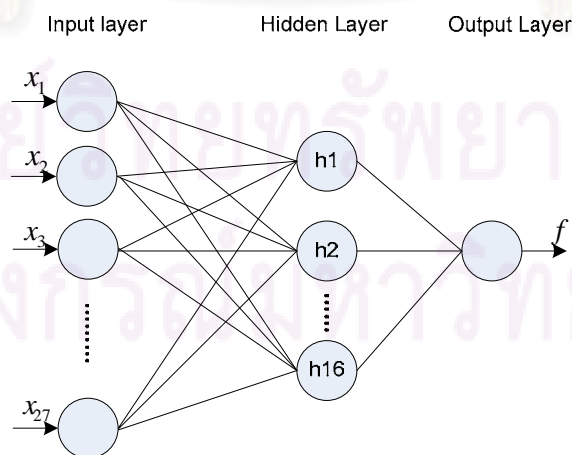


Figure 3.4 Neural Network Architecture for CLV prediction

CHAPTER 4

EXPERIMENT RESULT

4.1 Experiment Processing Results

In predictive modeling, obtain an exactness estimate of the MSE or Mean Squared Error and SSE or Sum Squared Error. The MSE is the most commonly used statistic for measuring the accuracy of the model. It is the squared difference between the target values and the predicted values, averaged over the number of observations that the model is fitting. A good model is the one that has the lowest SSE from the testing dataset. (MSE value is referenced value). This is because the SSE estimates from the training data set that is used to fit the model will almost certainly be overly optimistic since the same data is used to fit the model. To understand that just fitting the model to the training data set does not mean that the model is necessarily correct and that the model will fit well to new data.

As here, both MSE and SSE were utilized. SSE for main error estimate method and MSE for secondary reference. Besides the usual estimators and test statistics produced for a regression, a fit analysis can produce many diagnostic statistics. Collinearity diagnostics measure the strength of the linear relationship among explanatory variables and how this affects the stability of the estimates. Influence diagnostics measure how each individual observation contributes to determining the parameter estimates and the fitted values.

In statistics, the mean squared error or MSE of an estimator is one of many ways to quantify the amount by which an estimator differs from the true value of the quantity being estimated. The MSE can be written as the sum of the variance and the squared bias of the estimator:

$$MSE\left(\hat{\theta}\right) = Var\left(\hat{\theta}\right) + \left(Bias\left(\hat{\theta}, \theta\right)\right)^2. \quad (14)$$

In matrix algebra notation, a linear model is written as

$$y = X\beta + e. \quad (15)$$

Where y is the $n \times 1$ vector of responses of n rows, X is the $n \times p$ design matrix (rows are observations and columns are explanatory variables),

β is the $p \times 1$ vector of p parameter estimates and, e is a $(n \times 1)$ vector of residuals of n rows.

Each effect in the model generates one or more columns in a design matrix X . The first column of X is usually a vector of 1's used to estimate the intercept term. In general, no-intercept models should be fit only when theoretical justification exists. The classical theory of linear models is based on some strict assumptions. Ideally, the response is measured with all the explanatory variables controlled in an experimentally determined environment. If the explanatory variables do not have experimentally fixed values but are stochastic, the conditional distribution of y given X must be normal in the appropriate form.

Less restrictive assumptions are as follows:

- The form of the model is corrected (all important X variables have been included).
- Explanatory variables are measured without error.
- The expected value of the errors is 0.
- The variance of the errors (and thus the response variable) is constant across observations (denoted by σ^2).
- The errors are uncorrelated across observations.

If all the necessary assumptions are met, the least-squares estimates of σ^2 are the best linear unbiased estimates (BLUE); in other words, the estimates have minimum variance among the class of estimators that are unbiased and are linear functions of the responses. In addition, when the error term is assumed to be normally distributed, sampling distributions for the computed statistics can be derived. These sampling distributions form the basis for hypothesis tests on the parameters.

The method used to estimate the parameters is to minimize the sum of squares of the differences between the actual response values and the

values predicted by the model. An estimator b for β is generated by solving the resulting normal equations

$$a(X'X)b = X'y, \quad (16)$$

yielding,

$$b = (X'X)^{-1}X'y, \quad (17)$$

Let H be the projection matrix for the space spanned by X , sometimes called the hat matrix,

$$H = (X'X)^{-1}X', \quad (18)$$

Then the predicted mean vector of the n observation responses is

$$H = (X'X)^{-1}X',$$

$$SSE = (y - \hat{y})^2 = \sum_{i=1}^n (y_i - x_i b)^2, \quad (19)$$

The Sum of Squares for Error is

$$\hat{y} = Xb = Hy, \quad (20)$$

where x_i is the i^{th} row of the X matrix.

Assume that X is of full rank. The variance $\frac{3}{4}$ of the error is estimated by the mean square error

$$S^2 = MSE = [SSE / (n - p)], \quad (21)$$

$$SSE = MSE(n - p) \quad (22)$$

According to the PCA experiment results in Table 3.2, neural network proceed to use it. The data set are generated by PCA so called the data preparation step. It generated figure 2.6 by using SAS enterprise miner 5.3 in Figure 4.1

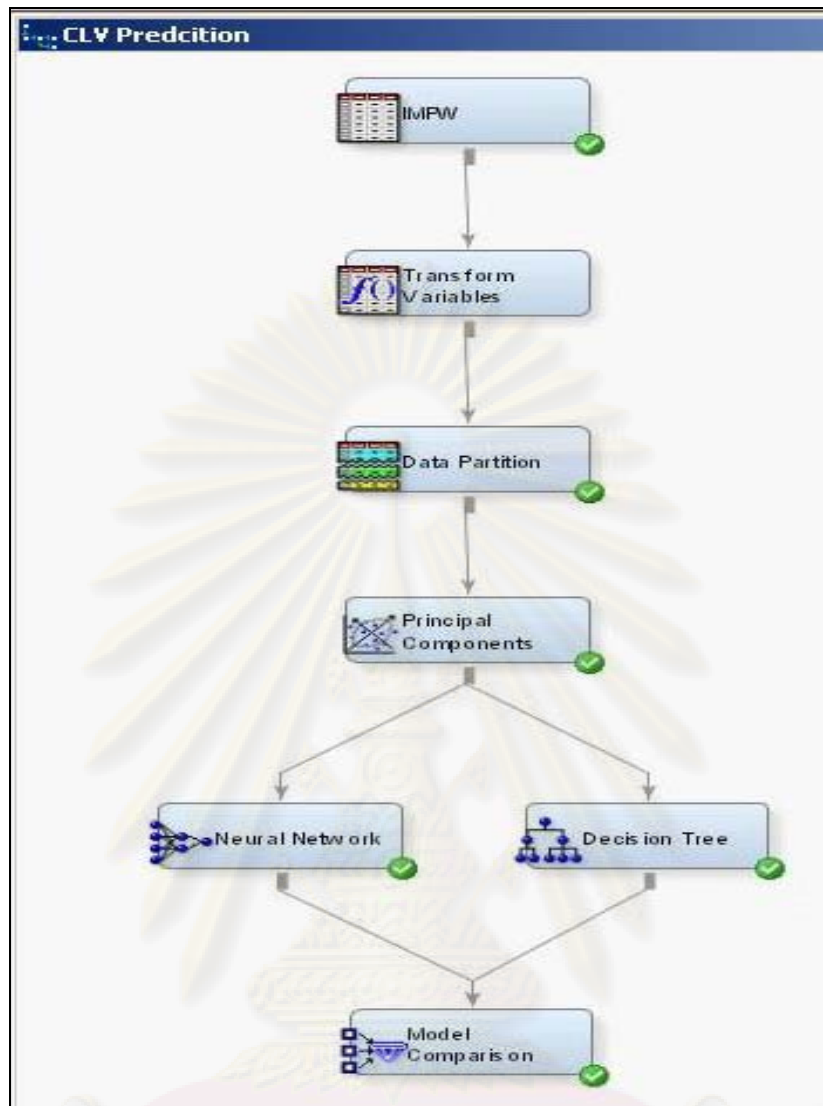


Figure 4.1 Working Environment Flow Design

4.2 Accuracy Evaluation

Therefore, in this experiment, customer names and mobile number have been taken out for the confidential reason. (Only the past observation data of customers are useful) The data consist of 12005 records of which 60 percent is used for training and then 40 percent is used for testing.

In this study, it has used total revenue (net profit) of the customer instead of cash flow in our modeling have mentioned beforetime. We create a new column into the dataset called TOT_CLV_CHR which is stand for Customer Churning (referenced by Table 3.1) as our prediction. To calculate the profit for each customer we can get from finance report forms monthly (yearly). And the can

be acquired from SAS Solution Customer Retention. All experiments were performed on Pentium IV 2.5GHz with 2GB memory, running Windows XP. The code was written in SAS Code [15]. In the jargon, a correct prediction is called true, while an incorrect prediction is called false. For example, if customers have been decided to drop the services (churn) from the company, and the prediction properly detects it, it is said to be a true-positive. Likewise, if customers no willing to drop the services (not churn) from the company, and the prediction indicates that thing is not present, it is said to be a true-negative. A false-positive occurs when the customers have not been decided to churn the services, but the prediction erroneously indicates that they do. These results are needless worry and the worries and need to be planning to earn them back, more expenses additionally. An even worse scenario occurs with the false-negative, where churn is present, but the prediction indicates the customers are not churn. As we all know, ignore the cases can be even worse because maybe network itself problem might be lead into more and more customers churn actions.

4.2.1 Results of MLP Neural Network

The Mean of Squared error (MSE) is less than 0.00001 in 2311 seconds. It can be seen that the network is converged after 998 iterations. The Sum of Squared Error (SSE) of train set was not greater than 0.0392 and the testing SSE value is less than 0.0989, for whole steps. In order to test the performance of the network, the actual net profit values are plotted against the prediction net profit value as in Figure 4.2. The points below the diagonal line imply that the predicted value is more than the actual value. In this case, it is the exaggeration profit value. All data after rescaling and normalizing our data have negative values because of the payment delay.

In Figure 4.2, from 4800 testing records, the true positive values are 4194 records, true negative values are 606 records, false positive values are 307 records and false negative values are 299 records.

$$\text{The accuracy is } \text{acc} = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|} \approx 88.9\% .$$

The true positive rate is called *Sensitivity*, also known as

$$TPR = \text{Sensitivity} = \frac{|TP|}{|TP| + |FN|} \approx 93\% .$$

The false negative rate denoted as

$$FNR = (1 - \text{Specificity}) = \frac{|FP|}{|TN| + |FP|} \approx 33\% .$$

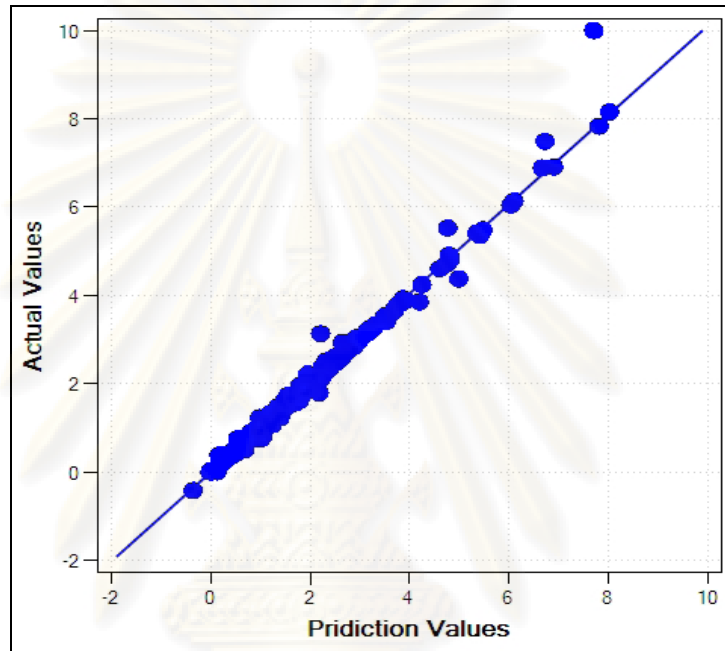


Figure 4.2 Actual Values vs. prediction values. (MLP)

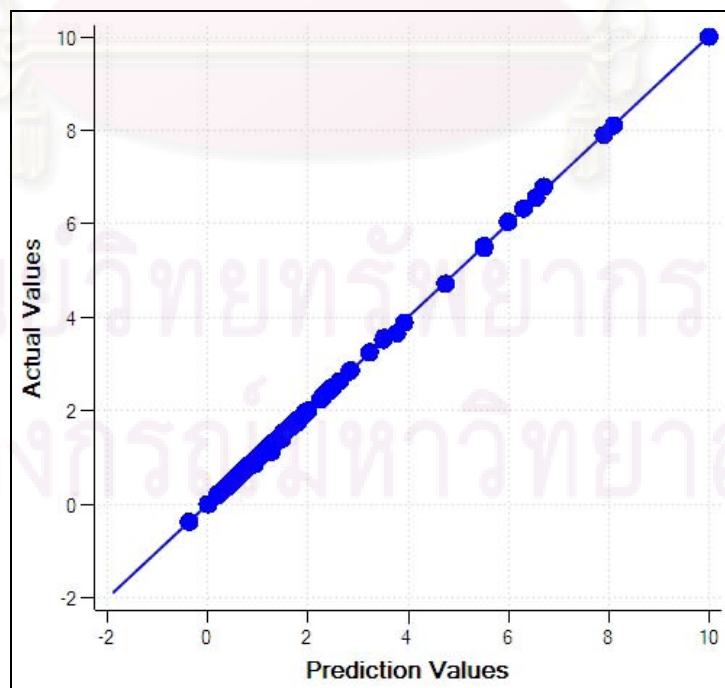


Figure 4.3 Actual values vs. prediction Values after Cross Validation (MLP).

The result has ready to use, but it can be get even more exactness by using Cross Validation, as mentioned above, we have fold 10 networks and get their results average them, the whole dataset 12005 records, they have been divided to 1200 records 1-fold, namely training set 10800 records and test dataset 1205 records, 10 training data sets correspond to 1 test data sets. There are 10 networks for all data sets. Our idea is to exchange the high error records, such as top 20 high error records in test datasets with lowest error records, such as 20 lowest error records which in another network test set. After exchanging, retrain them until all networks reach lower error. The Cross validation training has been used 1732 seconds, and then the training SSE value is 0.0391 which is almost the same as before. But testing SSE value is 0.0798. Figure 4.3 shows the accuracy after the cross validation.

In Figure 4.3, among 4800 testing records, there are 4693 true positive records, 117 true negative records, 34 false positive records, and 83 false negative records.

$$\text{The accuracy is } \text{acc} = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|} \approx 96.5\% .$$

The true positive rate is,

$$\text{TPR} = \text{Sensitivity} = \frac{|TP|}{|TP| + |FN|} \approx 98\% .$$

The false negative rate is,

$$\text{FPR} = (1 - \text{Specificity}) = \frac{|FP|}{|TN| + |FP|} \approx 22\% .$$

4.2.2 Results of Decision Tree with C4.5 Algorithm

The Mean of Squared error (MSE) is 0.0137 in 7.73 seconds. The Sum of Squared Error (SSE) of 3.173 and the testing SSE value is less than 5.189, for whole steps.

In Figure 4.4, among 4800 testing records, there are 3912 true positive records, 888 true negative records, 669 false positive records, and 219 false negative records.

$$\text{The accuracy is } \text{acc} = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|} \approx 84.3\% .$$

The true positive rate is,

$$TPR = \text{Sensitivity} = \frac{|TP|}{|TP| + |FN|} \approx 86.7\% .$$

The false negative rate is,

$$FPR = (1 - \text{Specificity}) = \frac{|FP|}{|TN| + |FP|} \approx 42.97\% .$$

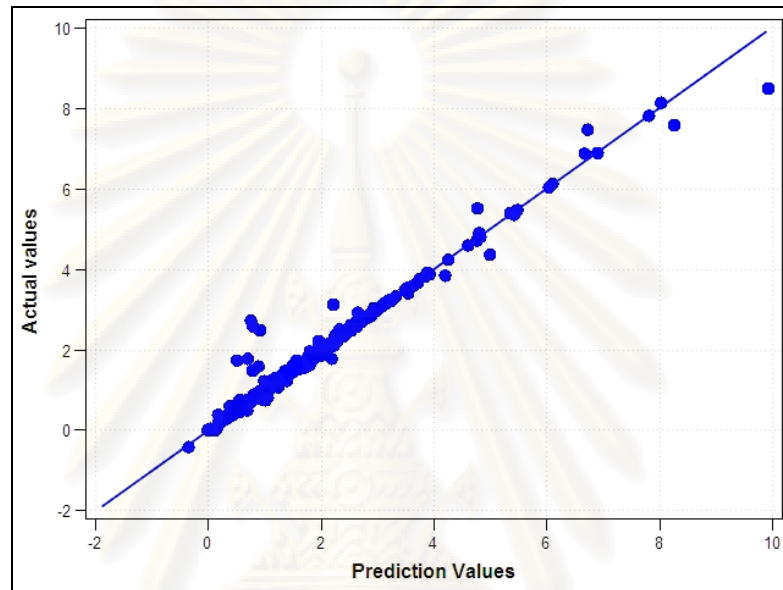


Figure 4.4 Actual Values vs. prediction values (Decision tree)

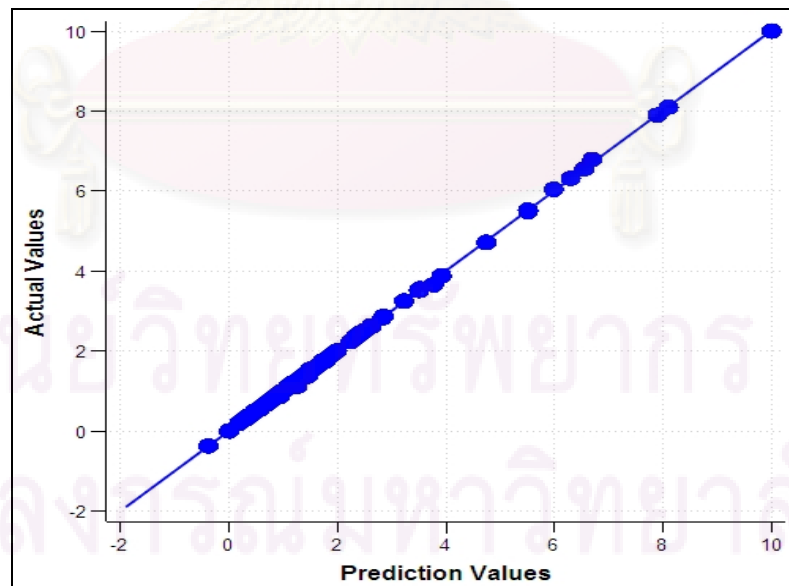


Figure 4.5 Actual values vs. prediction Values after Cross Validation (Decision Tree)

After applied the cross validation, we have handled the Mean of Squared error (MSE) is 0.0014 in 10.63 seconds. The Sum of Squared Error (SSE) training set of 7.173 and the testing SSE value is less than 11.189, for whole steps. It is easy to see that the SSE value still higher the SSE value of neural network experiment. In Figure 4.5, there are 4386 true positive records, 414 true negative records, 258 false positive records, and 153 false negative records.

$$\text{The accuracy is } \text{acc} = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|} \approx 92.1\% .$$

The true positive rate is,

$$\text{TPR} = \text{Sensitivity} = \frac{|TP|}{|TP| + |FN|} \approx 96\% .$$

The false negative rate is,

$$\text{FPR} = (1 - \text{Specificity}) = \frac{|FP|}{|TN| + |FP|} \approx 38\% .$$

4.3 Comparison

Without cross validation		Cross validation	
Neural Network	Decision Tree	Neural Network	Decision Tree
Accuracy	Accuracy	Accuracy	Accuracy
88.90%	84.30%	96.50%	92.10%

Table 4.2 Accuracy comparison between neural network with cross validation and decision tree

In Table 4.2 is the result accuracy value of the neural network compare with the decision tree. The result is easy to spread out we can get more exactness from neural network modeling, but the time expenses comparison decision tree is better, whatever with or without cross validation. As it seen in the experiments, complexity optimization for the neural network is an integral part. Decision tree modeling method is based on non-parametric statistics. And it is easier to understand. So, they can be compared? The experiment result proves the accuracy is best. It will be using scheduler that if set for offline model application, offline model means run it before read specify report. The prediction CLV values are

using in reality that need to bring out them in the report. In sum of that, if it has employed neural network model, scheduled the prediction before run the report approximately an hour.

Neural network model has been used approximately 40 minutes (38.5 minutes) without cross validation and with cross validation around 30 minutes (28.87 minutes). The new data are coming every day in actually, need to be adding in to the neural network train them first, and more description will briefly in chapter 5.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER 5
CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions

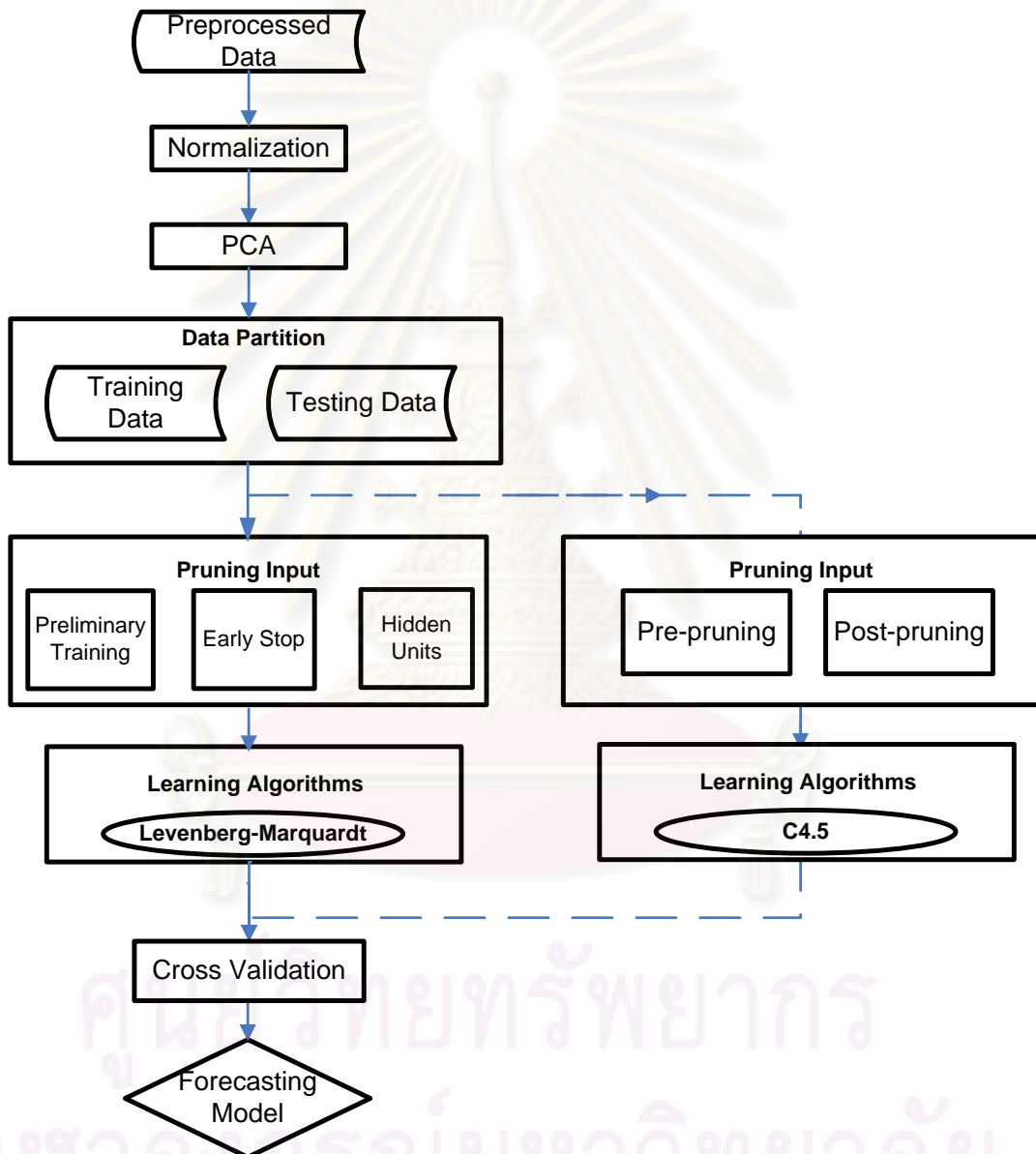


Figure 5.1 Detailed design layouts.

It would be clear in Figure 5.1. The preprocessed data would be normalizing first by max-min transformation. After that the data should be get through with dimensions reduction so called PCA. And then the data unit would be separated

into two groups, training data, and testing data. By some pruning methodologies, such as preliminary training, early stop, it also has used decision tree pruning methodologies pre-pruning and post-pruning, it would be cleared divided by broken line because of neural network is primary method as mentioned. After that, the learning algorithms (c4.5 and Levenberg-marquardt) will be used to train the models grouped with different preprocessed data sets, through setting several numbers of hidden units in the experiment. A best model will be chosen from the compared experiment result. The value of MSE and SSE shows that, neural network is truly better than decision tree, the training set SSE with cross validation is 0.0391 and the test SSE is 0.0798. For the effective and useful better than value decision tree. It would be worth to say that during the data increasing the neural model is better than decision tree.

It is feasible and rather efficient to apply an artificial neural network to predict the CLV in telecommunication business. The research study focused on the usability in commercial world instead of only theoretical aspect. At here, it also has shortcomings. Firstly, the author only take in consideration about contractual product type namely postpaid customers, but for prepaid customers are difficult to predict and calculate CLV, as well as their cross-selling or up-selling and associated effect values. Secondly, defining $\alpha_{i,j,t}$ from churning model, retention model, and growth model should be further investigated.

Prediction without cross validation (SSE value)		Prediction with cross validation (SSE value)	
Neural Network	Decision Tree	Neural Network	Decision Tree
Training =0.0392	Training =7.173	Training =0.0391	Training =3.163
Testing =0.0989	Testing =11.189	Testing =0.0798	Testing =5.089

Table 5.1 Neural network SSE value vs. Decision tree SSE value

As mentioned above, the experiment proves that the neural network model is better for the prediction. The comparison with decision tree help to verify neural network is the most suitable modeling. It is focused on the detailed target in working environment of telecommunication industry.

5.2 Discussions

In the forecasting and prediction area of business customer buying action, the following issues should be highlighted when it designed different models for individual environment:

- 1) The source data selection to support different target output prediction, However a good predictive model which is widely used and come must be able to adjust its complexity to compensate for noisy training data.
- 2) More refining to get more fit data for the Data preparation to be training
- 3) Based on the customized learning algorithm instead of standard algorithms in order to get more effective spacing and flexible structure.

In the design of the prediction models, some other methods can be used to try to resolve the situations. For example, a combination of neural network and regression could also be an ideal solution for this issue.

5.3 Future works

Firstly, in existing data, more actions can be applied for customer segmentations. Classify customers with their profession, gender, age level, nationality etc, and retrain the network. Because of different category people, or the races have their owned logical action. Secondly, our designed modeling more data need to be added during the time past. However, the production process data of customer's actions are time series. How to reduce the shortcomings appeared for training data during the more data come in. The related data are increasing tons after a period of time. How to design a suitable model in time series? Suitable time, time also another reason will affect the result, such the time for financial crisis, will lower down the buying. Hence, it would be better predict the results of a period time. Finally, how to move the experiment model to real time prediction function will be a task in the future.

References

- [1] Berger, Paul D., & Nasr, Nada I. "Customer lifetime value: Marketing models and applications" Journal of Interactive Marketing, 1998, pp. 17-30.
- [2] Berry, L. L.. Relationship Marketing L. Berry & L. Shostack, Eds. "Emerging Perspectives on Services Marketing". Chicago, IL: American Marketing Association, 1983
- [3] Kamakura, W. A., Mela, C., Ansari, A., Fader, P., Iyengar, R., Naik, P. "Choice models and customer relationship management," Marketing Letters, 2005, pp. 192-279.
- [4] Malthouse, E. C., & Blattberg, R. C., "Can we predict customer lifetime value", Journal of Interactive Marketing, 19(1), 2005, pp. 2-16.
- [5] Gary Cokins, "Intelligence Value Chain White Paper", SAS White Paper, 2004.
- [6] Berry and Linoff, Michael J.A. and Gordon S.. "Mastering Data Mining: The Art and Science of Customer Relationship Management," John Wiley & Sons, 2000
- [7] The big broadband gamble, Telecommunications Online, February 1999.
- [8] "The customers telecommunications process", sascom Magazine, August 2007
- [9] J.I. Michael, Neural Networks, A. Tucker, (Ed.), CRC handbook of Computer Science, CRC Press, Boca Raton, FL, 1996.
- [10] H.T. Martin, Neural Network Design, PWS Publishing Company, 1996.
- [11] Haykin Simon, Neural Networks: A Comprehensive Foundation, Pearson Education, Inc, (2001).
- [12] Diggavi, S.N.; Shynk, J.J.; Bershada, N.J. "Convergence models for Rosenblatt's perceptron learning algorithm" IEEE TRANSACTIONS ON SIGNAL PROCESSING, Volume 43, Issue 7, Jul 1995 pp. 1696 - 1702.
- [13] McCullagh, Peter; Nelder, John Generalized Linear Models, Second Edition, Boca Raton: Chapman and Hall/CRC. (1989).

- [14] S. Jonathon, A Tutorial on Principal Component Analysis, Version 2, 2005.
- [15] Newman, D., Hettich, S., Blake, C., & Merz, C. UCI repository of machine learning databases, 1998.
- [16] J. Ross Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, USA, 1993.
- [17] Salvatore Ruggieri, "Efficient C4.5", IEEE Transactions on knowledge and data engineering, Vol. 14, No. 2, march/april 2002.
- [18] T.S. Lim, W.Y. Loh, and Y.S. Shih, "A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty - Tree Old and New Classification Algorithms," Machine Learning, vol. 40, no. 3, 2000, pp. 203 - 228.
- [19] J.R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, no. 1, 1996, pp. 81-106.
- [20] Diego Calvanesea, Luigi Dragoneb , Daniele Nardib, Riccardo Rosatib, and Stefano M. Trisolinic, "Enterprise modeling and Data Warehousing in Telecom Italia", Information systems, 2006, pp 1-32
- [21] Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques second edition, Morgan Kaufmann, San Francisco, 2006.
- [22] L. Breimann, J.H Friedman, R.A.Olshen, and C.J.Stone, Classification and regression tree. The wadsworth statistics/probability series, Belmont, California: Wadsworth, 1984
- [23] Warren S. Sarle, Stopped Training and Other Remedies for Over-fitting: Proceedings of the 27th Symposium on the Interface, 1995.
- [24] Elisseeff, A., and Paugam-Moisy, H., "Size of multilayer networks for exact learning: analytic approach", Advances in Neural Information Processing Systems, Cambridge, MA: The MIT Press, 1997, pp.162-168.
- [25] Kohavi, Ron, "A study of cross-validation and bootstrap for accuracy estimation and model selection," Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1995, pp. 1137-1143.
- [26] M. Randall, Data Mining Using SAS Enterprise Miner, John Wiley & Sons, Inc 2007.



APPENDICES

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



ต้นฉบับไม่มีหน้า 53

NO PAGE 53 IN ORIGINAL

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Appendix A
DATA FIELDS FOR PREDICTION

Data Fields

Description

Gender_CD(Gender code)

Customer gender

Educaton_CD

Education level code of a customer (Standard Occupation Code)

OCCUPATION_CAT_CD

Customer standard occupation

CHURN_FLG

Variable takes value 1 if the subscription is closed and 0 otherwise. This is treated as a target variable for building an analytical model for propensity to churn.

SUBS_TENURE

Subscription tenure. Subscription or line tenure. Prepay tenure in number of months from activation.

OB_CALL_INTL_DUR_BASE_1

Duration of outbound international calls for base 1 month. Sum of the duration of outbound international calls for base 1 month.

OB_CALL_INTL_DUR_BASE_2

Duration of outbound international calls for base 2 month. Sum of the duration of outbound international calls for base 2 month.

OB_CALL_INTL_DUR_BASE_3

Duration of outbound international calls for base 3 month. Sum of the duration of outbound international calls for base 3 month.

OB_CALL_INTL_DUR_BASE_4

Duration of outbound international calls for base 4 month. Sum of the duration of outbound international calls for base 4 month.

OB_CALL_INTL_DUR_BASE_5

Duration of outbound international calls for base 5 month. Sum of the duration of outbound international calls for base 5 month.

OB_CALL_INTL_DUR_BASE_6

Duration of outbound international calls for base 6 month. Sum of the duration of outbound international calls for base 6 month.

OB_CALL_NAT_ROAM_CNT_BASE_1

Number of outbound calls while roaming nationally for base 1 month. Count of outbound calls while roaming nationally for base 1 month.

OB_CALL_NAT_ROAM_CNT_BASE_2

Number of outbound calls while roaming nationally for base 2 month. Count of outbound calls while roaming nationally for base 2 month.

OB_CALL_NAT_ROAM_CNT_BASE_3

Number of outbound calls while roaming nationally for base 3 month. Count of outbound calls while roaming nationally for base 3 month.

OB_CALL_NAT_ROAM_CNT_BASE_4

Number of outbound calls while roaming nationally for base 4 month. Count of outbound calls while roaming nationally for base 4 month.

OB_CALL_NAT_ROAM_CNT_BASE_5

Number of outbound calls while roaming nationally for base 5 month. Count of outbound calls while roaming nationally for base 5 month.

OB_CALL_NAT_ROAM_CNT_BASE_6

Number of outbound calls while roaming nationally for base 6 month. Count of outbound calls while roaming nationally for base 6 month.

OB_CALL_LOC_DUR_BASE_1

Duration of outbound local calls for base 1 month. Sum of the duration of outbound local calls for base 1 month.

OB_CALL_LOC_DUR_BASE_2

Duration of outbound local calls for base 2 month. Sum of the duration of outbound local calls for base 2 month.

OB_CALL_LOC_DUR_BASE_3

Duration of outbound local calls for base 3 month. Sum of the duration of outbound local calls for base 3 month.

OB_CALL_LOC_DUR_BASE_4

Duration of outbound local calls for base 4 month. Sum of the duration of outbound local calls for base 4 month.

OB_CALL_LOC_DUR_BASE_5

Duration of outbound local calls for base 5 month. Sum of the duration of outbound local calls for base 5 month.

OB_CALL_LOC_DUR_BASE_6

Duration of outbound local calls for base 6 month. Sum of the duration of outbound local calls for base 6 month.

OB_CALL_NAT_DUR_BASE_1

Duration of outbound national calls for base 1 month. Sum of the duration of outbound national calls for base 1 month.

OB_CALL_NAT_DUR_BASE_2

Duration of outbound national calls for base 2 month. Sum of the duration of outbound national calls for base 2 month.

OB_CALL_NAT_DUR_BASE_3

Duration of outbound national calls for base 3 month. Sum of the duration of outbound national calls for base 3 month.

OB_CALL_NAT_DUR_BASE_4

Duration of outbound national calls for base 4 month. Sum of the duration of outbound national calls for base 4 month.

OB_CALL_NAT_DUR_BASE_5

Duration of outbound national calls for base 5 month. Sum of the duration of outbound national calls for base 5 month.

OB_CALL_NAT_DUR_BASE_6

Duration of outbound national calls for base 6 month. Sum of the duration of outbound national calls for base 6 month.

TOT_CALL_INTL_ROAM_DUR

Total of monthly duration calls while roaming internationally over last 6 months. Sum of the duration of monthly outbound calls while roaming nationally Total of monthly duration of outbound calls while roaming internationally over last 6 months. Sum of the duration of monthly outbound calls while roaming internationally over last 6 months.

TOT_OB_CALL_INIL_DUR

Total of monthly duration of outbound international calls over last 6 months. Sum of the monthly duration of outbound international calls over last 6 months.

TOT_OB_CALL_LOC_DUR

Total of monthly duration of outbound local calls over last 6 months. Sum of the monthly duration of outbound local calls over last 6 months.

TOT_OB_CALL_NAT_DUR

Total of monthly duration of outbound national calls over last 6 months. Sum of the monthly duration of outbound national calls over last 6 months.

IB_CALL_PRD8_DUR_BASE_1

Duration of inbound calls made in time period 8 for base 1 month. Sum of the duration of inbound calls in time period 8 for base 1 month.

IB_CALL_PRD8_DUR_BASE_2

Duration of inbound calls made in time period 8 for base 2 month. Sum of the duration of inbound calls in time period 8 for base 2 month.

IB_CALL_PRD8_DUR_BASE_3

Duration of inbound calls made in time period 8 for base 3 month. Sum of the duration of inbound calls in time period 8 for base 3 month.

IB_CALL_PRD8_DUR_BASE_4

Duration of inbound calls made in time period 8 for base 4 month. Sum of the duration of inbound calls in time period 8 for base 4 month.

IB_CALL_PRD8_DUR_BASE_5

Duration of inbound calls made in time period 8 for base 5 month. Sum of the duration of inbound calls in time period 8 for base 5 month.

IB_CALL_PRD8_DUR_BASE_6

Duration of inbound calls made in time period 8 for base 6 month. Sum of the duration of inbound calls in time period 8 for base 6 month.

OB_CALL_DUR_BASE_1

Duration of outbound calls for base 1 month. Sum of the duration of outbound calls for base 1 month.

OB_CALL_DUR_BASE_2

Duration of outbound calls for base 2 month. Sum of the duration of outbound calls for base 2 month.

OB_CALL_DUR_BASE_3

Duration of outbound calls for base 3 month. Sum of the duration of outbound calls for base 3 month.

OB_CALL_DUR_BASE_4

Duration of outbound calls for base 4 month. Sum of the duration of outbound calls for base 4 month.

OB_CALL_DUR_BASE_5

Duration of outbound calls for base 5 month. Sum of the duration of outbound calls for base 5 month.

OB_CALL_DUR_BASE_6

Duration of outbound calls for base 6 month. Sum of the duration of outbound calls for base 6 month.

TOT_IB_CALL_DUR

Total of monthly duration of inbound calls over last 6 months. Sum of the duration of inbound calls over last 6 months.

AVG_OB_CALL_DUR

Average duration of outbound voice calls over last 6 months. This variable is calculated as the sum of outbound call duration over last 6 months divided by the total number of outbound voice calls made over 6 months.

TOT_IB_CALL_CNT

Total of monthly number of inbound calls over last 6 months. Sum of the inbound calls over last 6 months.

OB_CALL_CNT_BASE_1

Number of outbound calls for base 1 month. Count of outbound calls for base 1 month.

TOT_OB_CALL_NAT_ROAM_CNT

Total of monthly number of outbound calls while roaming nationally over last 6 months. Sum of the monthly outbound calls while roaming nationally over last 6 months.

TOT_OB_CALL_INTL_CNT

Total of monthly number of outbound international calls over last 6 months. Sum of the monthly outbound international calls over last 6 months.

TOT_OB_CALL_LOC_CNT

Total of monthly number of outbound local calls over last 6 months. Sum of the monthly outbound local calls over last 6 months.

TOT_OB_CALL_NAT_CNT

Total of monthly number of outbound national calls over last 6 months. Sum of the monthly outbound national calls over last 6 months.

TOT_OB_CALL_INTL_ROAM_CNT

Total of monthly number of outbound calls while roaming internationally over last 6 months. Sum of the monthly outbound calls while roaming internationally over last 6 months.

TOT_EARNED_POINTS_CNT

Total number of loyalty points earned in last 6 months.

TOT_DACTV_SUBS_LAST_MTH_CNT

Number of deactivated subscriptions in last month. The grain of this variable is customer. The deactivation can be both voluntary and involuntary.

TOT_DAY_LAST_COMPLAINT_CNT

Number of days from the last complaint. Derived as the difference in days between the last complaint date and the present date.

TOT_DAY_LAST_OB_BARRED_CNT

Number of days from the last suspension due to non-payment. Calculated as the difference between the last suspension due to non-payment date and the present date.

TOT_DAY_LAST_SUSPENDED_CNT

Number of days from the last outbound barred due to non-payment. Calculated as the difference between the last outbound barred due to non-payment date and the present date.

TOT_EMAIL_QUERY_CNT

Total number of email complaints made over last 6 months.

CUST_SEG_CD

Segment of the customer. Calculated based on the number of SIM cards. Segments like SOHO(1-3), SME(4-49), CORPO(50+).

MTH_TO_SUBS_END_CNT

Time to subscription end. This holds for subscriptions with fixed time subscriptions. Can have negative values after the binding period.

TOT_SRV_DROPPED_CNT

Total number of services cancelled in last 6 months.

TOT_SRV_ADDED_CNT

Total number of services added in last 6 months.

TOT_OUTSTAND_30_60_DAY_AMT

Total outstanding amount for delinquency period between 30 to 60 days.

TOT_OUTSTANT_30_DAY_AMT

Total outstanding amount for delinquency period less than or equal to 30 days.

TOT_OUTSTAND_60_90_DAY_AMT

Total outstanding amount for delinquency period between 60 to 90 days.

TOT_REV_FIX_AMT

Total of monthly revenue amount due to voice calls to fixed line phone over last 6 months.

TOT_REV_GPRS_AMT

Total of monthly revenue amount due to GPRS over last 6 months.

TOT_REV_INET_AMT

Total of monthly revenue amount due to internet access over last 6 months.

TOT_COMPLAINT_1_MTH_CNT

Total number of complaints made over last month.

NET_SUBS_CHNG_6_MTH_CNT

Net change in number of subscriptions in last 6 months. The grain of this variable is customer. Customer can activate and deactivate subscriptions in the same period. Deactivation considered both voluntary and involuntary. It is calculated as difference between the number of active subscriptions and deactivated subscriptions.

TOT_MTH_LAST_SUSPENDED_CNT

Total number of months since last suspended due to non-payment.

LAST_PRICE_PLAN_CHNG_DAY_CNT

Number of days from last price plans change. It can be derived by comparing latest price plan change date and the present date.

MTH_SINCE_DATA_ACTVN

Months since activation of data service. Derived as the difference in months between the date of activation of data service and the present date.

MTH_SINCE_VM_ACTVN

Months since activation of voice mail service. Derived as the difference in months between the date of activation of voicemail service and the present date.

BARRING_REASON_CD

Reason for last outbound barred subscription. Must take into account the barring code

TOT_OB_CALL_CNT

Total of monthly number of outbound calls over last 6 months. Sum of the outbound calls over last 6 months.

TOT_ACTV_SRV_CNT

Total number of active services till date.

REV_AMT_BASE_1

Total revenue amount for base 1 month.

REV_AMT_BASE_2

Total revenue amount for base 2 month.

REV_AMT_BASE_3

Total revenue amount for base 3 month.

REV_AMT_BASE_4

Total revenue amount for base 4 month.

REV_AMT_BASE_5

Total revenue amount for base 5 month.

REV_AMT_BASE_6

Total revenue amount for base 6 month.

CURR_PRICE_PLAN_CN

Current price plan code.

CURR_PAY_METHOD_CN

Most recent payment method code.

POSTAL_CD

Post code. The grain of this variable is customer.

PREV_PRICE_PLAN_CD

Previous price plan code.

CUST_AGE

Age of a customer in years. The grain of this variable is customer. Derived from the customer's birth date.

AVG_DISTINCT_CELL_CNT

Average number of distinct cells visited in last 6 months.

MOST_FREQ_COMPLAINT_CD

Most frequent complaint code in last 6 months. Mode of the complaint code over last 6 months.

CUST_SPEC_NO_CNT

Percent change in the number of outbound international voice calls in the latest two months (1-2) with respect to the first four months (3-6). Comparison of last two months with the first four months. This variable is calculated as $(\text{average number of outbound international voice calls in months 3-6} - \text{average number of outbound international voice calls in months 1-2}) / (\text{average number of outbound international voice calls in months 3-6}) * 100$.

PCT_CHNG_OB_INTL_CNT

Percent change in the number of outbound international voice calls in the latest two months (1-2) with respect to the first four months(3-6). Comparison of last two months with the first four months. This variable is calculated as $(\text{average number of outbound international voice calls in months 3-6} - \text{average number of outbound international voice calls in months 1-2}) / (\text{average number of outbound international voice calls in months 3-6}) * 100$.

PCT_CHNG_IB_SMS_CNT

Percent change in the total number of SMS received (inbound) in the latest two months (1-2) with respect to the total SMS received (inbound) in first four months(3-6). Comparison of last two months with the first four months. This variable is calculated as

(average of number of SMS sent in months 3-6 - average of number of SMS received in months 1-2)/(average of number of SMS received in months 3-6)*100.

PCT_CHNG_OB_LOC_CNT

Percent change in the number of outbound local voice calls in the latest two months (1-2) with respect to the first four months (3-6). Comparison of last two months with the first four months. This variable is calculated as (average number of outbound local voice calls in months 3-6 - average number of outbound local voice calls in months 1-2)/(average number of outbound local voice calls in months 3-6)*100.

PCT_CHNG_OB_NAT_CNT

Percent change in the number of outbound national voice calls in the latest two months (1-2) with respect to the first four months (3-6). Comparison of last two months with the first four months. This variable is calculated as (average number of outbound national voice calls in months 3-6 - average number of outbound national voice calls in months 1-2)/(average number of outbound national voice calls in months 3-6)*100.

PCT_CHNG_OB_LOC_SMS_CNT

Percent change in the number of local SMS sent in the latest two months (1-2) with respect to the local SMS sent in first four months(3-6). Comparison of last two months with the first four months. This variable is calculated as (average number of local SMS sent in months 3-6 - average number of local SMS sent in months 1-2)/(average number of local SMS sent in months 3-6)*100.

PCT_CHNG_OB_WAP_CNT

Percent change in the total number of outbound WAP calls made in the latest two months (1-2) with respect to the total outbound WAP calls made in first four months(3-6). Comparison of last two months with the first four months. This variable is calculated as (average of number of outbound WAP calls in months 3-6 - average of number of outbound WAP calls in months 1-2)/(average of number of outbound WAP calls in months 3-6)*100.

PROP_N_IB_LOW_DUR

Proportion of total of duration of voice calls received in low rate hours over last 6 months. Calculated as total duration of inbound voice calls received in low rate hours over last 6 months divided total duration of inbound voice calls received over last 6 months.

RATIO_SPEC_SMS_SMS_REV_AMT

Ratio of revenue due to special SMS services to the normal SMS services over last 6 months. Ratio of average revenue due to special SMS last 6 months to average revenue due to normal SMS over last 6 months.

PCT_CHNG_SUPENDED_CNT

Percent change in the number of times suspended in the latest months (1-2) with respect to the number of times suspended in the first four months (3-6). Comparison of last two months with the first four months. This variable is calculated as $(\text{average number of times suspended in months 3-6} - \text{average number of times suspended in months 1-2}) / (\text{average number of times suspended in months 3-6}) * 100$.

PCT_CHNG_BILL_AMT

Percent change in the bill amount for the latest months (1-2) with respect to the bill amount for the first four months (3-6). Comparison of last two months with the first four months. This variable is calculated as $(\text{average bill amount for months 3-6} - \text{average bill amount for months 1-2}) / (\text{average bill amount for months 3-6}) * 100$.

CUST_SUBS_ID

Subscription identifier.

TOT_REV_AMT

Total of revenue amount over last 6 months.

TOT_PROF_AMT

Total profitability amount over last 6 months.

CUST_ID

Business key for the customer

NAME

Customer Name



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Appendix B

SAS Program Neural Network Cross validation Coding

```

/* 10-fold cross validation SAS program */
ODS LISTING CLOSE;
LIBNAME CLVSC "c:\dmnn\source";
LIBNAME CLVTMP "c:\dmnn\clvtmp";
LIBNAME CLVIN "c:\dmnn";
DATA clvin.time;
current_start=time();
current_end=time();
time=current_start;
run;

/* principal component analysis */
PROC PRINCOMP DATA=CLVSC.CLV_INPUT_NORMALIZED N=27 OUT=CLVIN.PRIN;
RUN;
PROC DATASETS LIBRARY = CLVTMP KILL;
RUN;
PROC DATASETS LIBRARY = WORK KILL;
RUN;

/*the marco of data partition */
%LET SEED = 12345;
DATA TRN TST ;
DROP _C00;
;
SET CLVIN.PRIN;
IF (12004 +1-_N_)*RANUNI(12345) <= (7202 - _C000001) THEN DO;
_C000001 + 1;
OUTPUT TRN;
END;
ELSE DO;
_C000002 + 1;
OUTPUT TST;

```

```

END;
RUN;
PROC DMDB DATA=TRN OUT=OUTSAS DMDBCAT=CBH;
  VAR prin1 prin2 prin3 prin4 prin5 prin6 prin7 prin8 prin9 prin10
  prin11 prin12 prin13 prin14 prin15 prin16 prin17 prin18 prin19 prin20 prin21
  prin22 prin23 prin24 prin25 prin26 prin27 tot_rev_amt;
RUN;
/* first time train the big network */

PROC NEURAL DATA=TRN DMDBCAT=CBH TESTDATA=TST;
INPUT
  prin1 prin2 prin3 prin4 prin5 prin6 prin7 prin8 prin9 prin10 prin11
  prin12 prin13 prin14 prin15 prin16 prin17 prin18 prin19 prin20 prin21 prin22
  prin23 prin24 prin25 prin26 prin27 / level=int;
target tot_rev_amt / level=int;
initial randout outest=rioweight00;
nloptions absgconv=1e-14;
prelim 5 randout outest=w00;
nloptions absgconv=1e-14;
archi mlp hidden=16;
train outest=oe estiter=1 tech=levmar;
score data=trn nodmdb out=trndata outfit=trnfit role=train;
score data=tst nodmdb out=tstdata outfit=tsffit role=test;
RUN;

DATA TRN;
set trn;
therandom = ranuni(86);
RUN;
DATA TST;
SET TST;
THERANDOM = RANUNI(86);
RUN;
/*THEN, DIVIDE THE DATASET INTO 10 GROUPS BASED ON THE RANDOM NUMBER*/
PROC RANK DATA=TRN OUT = TRNRANKED GROUPS = 10;

```

```

        VAR therandom;

RUN;

PROC RANK DATA=TST OUT = TSTRANKED GROUPS = 10;
        VAR therandom;

RUN;

%MACRO RUNIT;
%LET W=1;
/* DO LOOP 10-FOLD TRAIN */
%do x = 0 %to 9;
/* Delete exist datasets*/
data test&x.;
    set tstranked;
    where therandom = &x;
run;
data train&x.;
    set trnranked;
    where therandom = &x;
run;
%let w=%eval(&x+1);
%if w=10 %then w=0;
/*train the network in loop*/
PROC NEURAL data=train&x. dmdbcat=cbh graph testdata=test&x.;
INPUT
prin1 prin2 prin3 prin4 prin5 prin6 prin7 prin8 prin9 prin10 prin11
prin12 prin13 prin14 prin15 prin16 prin17 prin18 prin19 prin20 prin21 prin22
prin23 prin24 prin25 prin26 prin27 / level=int;
target tot_rev_amt / level=int;
initial inest=rioweight00 outest=rioweight&x.;
nloptions absgconv=1e-14;


```

```
score data=train&x. nodmdb out=traindata&x. outfit=trainfit&x. role=train;
score data=test&x. nodmdb out=testdata&x. outfit=testfit&x. role=test;
run;
```

```
PROC APPEND BASE =CLVIN.TRAINDATA DATA =TRAINDATA&X.;
RUN;
PROC APPEND BASE =CLVIN.TRAINFIT DATA =TRAINFIT&X.;
RUN;
PROC APPEND BASE =CLVIN.TESTDATA DATA =TESTDATA&X.;
RUN;
PROC APPEND BASE =CLVIN.TESTFIT DATA =TESTFIT&X.;
RUN;
```

```
/* SQUENCE THE DATASET TRAIN AND TEST AND EXCHANGE*/
```

```
DATA TRAINDATA&X. ;
SET TRAINDATA&X ;
DIFFERENCE = P_TOT_REV_AMT-TOT_REV_AMT;
RUN;
DATA TESTDATA&X.;
SET TESTDATA&X.;
DIFFERENCE = P_TOT_REV_AMT-TOT_REV_AMT;
RUN;
PROC SORT DATA=TRAINDATA&X.;
BY DIFFERENCE;
RUN;
```

```
PROC SORT DATA=TESTDATA&X.;
BY DESCENDING DIFFERENCE;
RUN;
```

```
/*MOVE TO TEST HIGH ERROR TO TRAIN LOW ERROR */
```

```
DATA TRNTMP&X.;
SET TRAINDATA&X. (OBS=20);
RUN;
```

```

DATA TSTTMP&X.;
SET TESTDATA&X. (OBS=20);
RUN;

DATA TRAINDATA&X. ;
SET TRAINDATA&X.;
IF _N_>20;
RUN;

DATA TESTDATA&X.;
SET TESTDATA&X.;
IF _N_>20;
RUN;

/*DELETE RECORDS FROM SOURCE TABLE TEST TRAIN*/
PROC APPEND BASE =TRAINDATA&X. DATA = TSTTMP&X.;
RUN;
PROC APPEND BASE =TESTDATA&X. DATA = TRNTMP&X.;
RUN;

/*RETRAIN ALL NETWORKS*/
PROC NEURAL data=traindata&x. dmdbcat=cbh graph testdata=testdata&x.;
input
prin1 prin2 prin3 prin4 prin5 prin6 prin7 prin8 prin9 prin10 prin11
prin12 prin13 prin14 prin15 prin16 prin17 prin18 prin19 prin20 prin21 prin22
prin23 prin24 prin25 prin26 prin27 / level=int;
target tot_rev_amt / level=int;
initial inest=rioweight&x. outest=roweigh&w.;
nloptions absgconv=1e-14;
prelim 5 randout outest=w&w.;
nloptions absgconv=1e-14;
archi mlp hidden=16;
train outest=oe estiter=1 tech=levmar;
score data=traindata&x. nodmdb out=trainscore&x. outfit=trainfits&x. role=train;
score data=testdata&x. nodmdb out=testscore&x. outfit=testfits&x. role=test;
run;

```



```

PROC APPEND base =clvtmp.trainscore data =trainscore&x.;
run;
PROC APPEND base =clvtmp.trainfits data =trainfits&x.;
run;
PROC APPEND base =clvtmp.testscore data =testscore&x.;
run;
PROC APPEND base =clvtmp.testfits data =testfits&x.;
run;

ods listing;
proc means data=clvin.trainfits;
var _sse_;
output out= clvin.sse;
run;
proc means data=clvin.testfits;
var _tsse_;
output out= clvin.tsse;
run;
ods listing close;
/*
%LET X=%EVAL(&W + 1);
%LET W=%EVAL(&X+1);
*/
/*SWAP THE DATASETS FROM TRAIN TO RETRAIN*/
proc datasets library = work nodetails nolist;
delete train&x. test&x.
run;

proc datasets library = work nodetails nolist;
change traindata&x.=train&x.
run;

proc datasets library = work nodetails nolist;
change testdata&x.=test&x.
run;

```

```

dm 'clear log';

%end;

%MEND RUNIT;

%RUNIT;

/*CALCUATE THE TIME DURATION*/

%MACRO LOWERERR;

data clvin.tsse;
set clvin.tsse;
set clvin.tsse(where = (_stat_ = 'mean'));
tsse=_tsse_;
run;
%do %until (tsse lt 0.03);
%runit;
ods listing;
proc means data=clvin.trainfits;
var _sse_;
output out= clvin.sse;
run;
proc means data=clvin.testfits;
var _tsse_;
output out= clvin.tsse;
run;
ods listing close;

%end;

dm 'clear log';

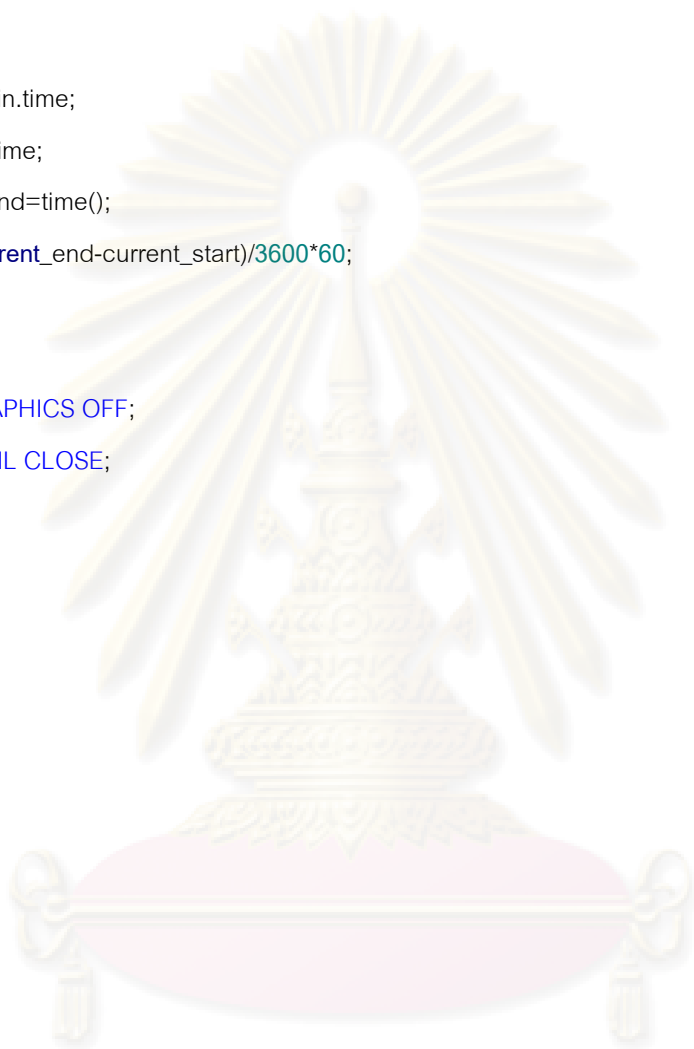
%MEND LOWERERR;

%LOWERERR;

ODS HTML;
ODS GRAPHICS ON;
PROC MEANS data=clvin.trainfits;
var _sse_;
output out= clvin.sse;

```

```
RUN;  
PROC MEANS data=clvin.testfits;  
var _tsse_;  
output out= clvin.tsse;  
RUN;  
  
DATA clvin.time;  
set clvin.time;  
current_end=time();  
time=(current_end-current_start)/3600*60;  
put time;  
RUN;  
ODS GRAPHICS OFF;  
ODS HTML CLOSE;
```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

SAS Program Neural Network Cross validation Coding

```
PROC ARBOR data=EMWS.Part_TRAIN
```

```
  Criterion=ENTROPY
```

```
  alpha=0.2
```

```
  Leafsize=80
```

```
  Mincatsize = 10
```

```
  Maxbranch=2
```

```
  Maxdepth=10
```

```
  Padjust=
```

```
  MAXRULES=5
```

```
  MAXSURRS=0
```

```
  Missing=USEINSEARCH
```

```
  Exhaustive=5000
```

```
  ;
```

```
  input %INTINPUTS
```

```
    / level = interval;
```

```
  input %NOMINPUTS
```

```
    / level=nominal;
```

```
  target TOT_REV_AMT / level=INTERVAL;
```

```
  ;
```

```
  Performance DISK
```

```
  NodeSize=20000
```

```
  ;
```

```
  Assess
```

```
  measure=ASE
```

```
  CVNIter = 20
```

```
  CVRepeat = 20
```

```
  CVSeed = 12345
```

```
  ;
```

```
  SUBTREE BEST
```

```
;  
MAKEMACRO NLEAVES=nleaves;  
save  
MODEL=EMWS.Tree_EMTREE  
SEQUENCE=EMWS.Tree_OUTSEQ  
IMPORTANCE=EMWS.Tree_OUTIMPORT  
NODESTAT=EMWS.Tree_OUTNODES  
SUMMARY=EMWS.Tree_OUTSUMMARY  
;  
RUN;  
QUIT;
```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Appendix C Quality Measure Methods Study

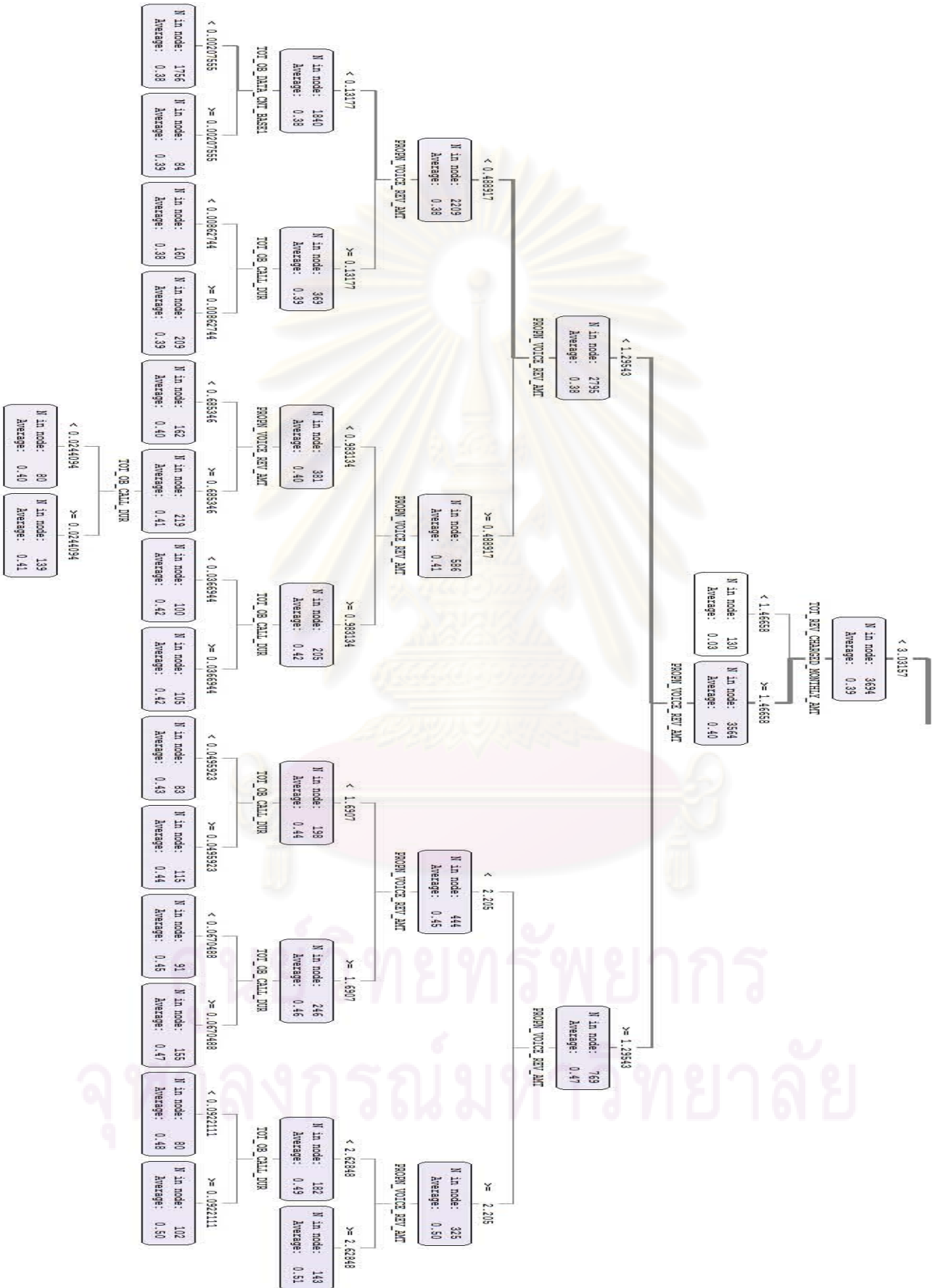
Table Sample. Confusion matrix of record pair classification

Actual	Classification	
	Match (\tilde{M})	Non-match (\tilde{U})
Match (M)	True matches also called True positives (TP)	False non-matches also called False negatives (FN)
Non-match (U)	False matches also called False positives (FP)	True non-matches also called True negatives (TN)

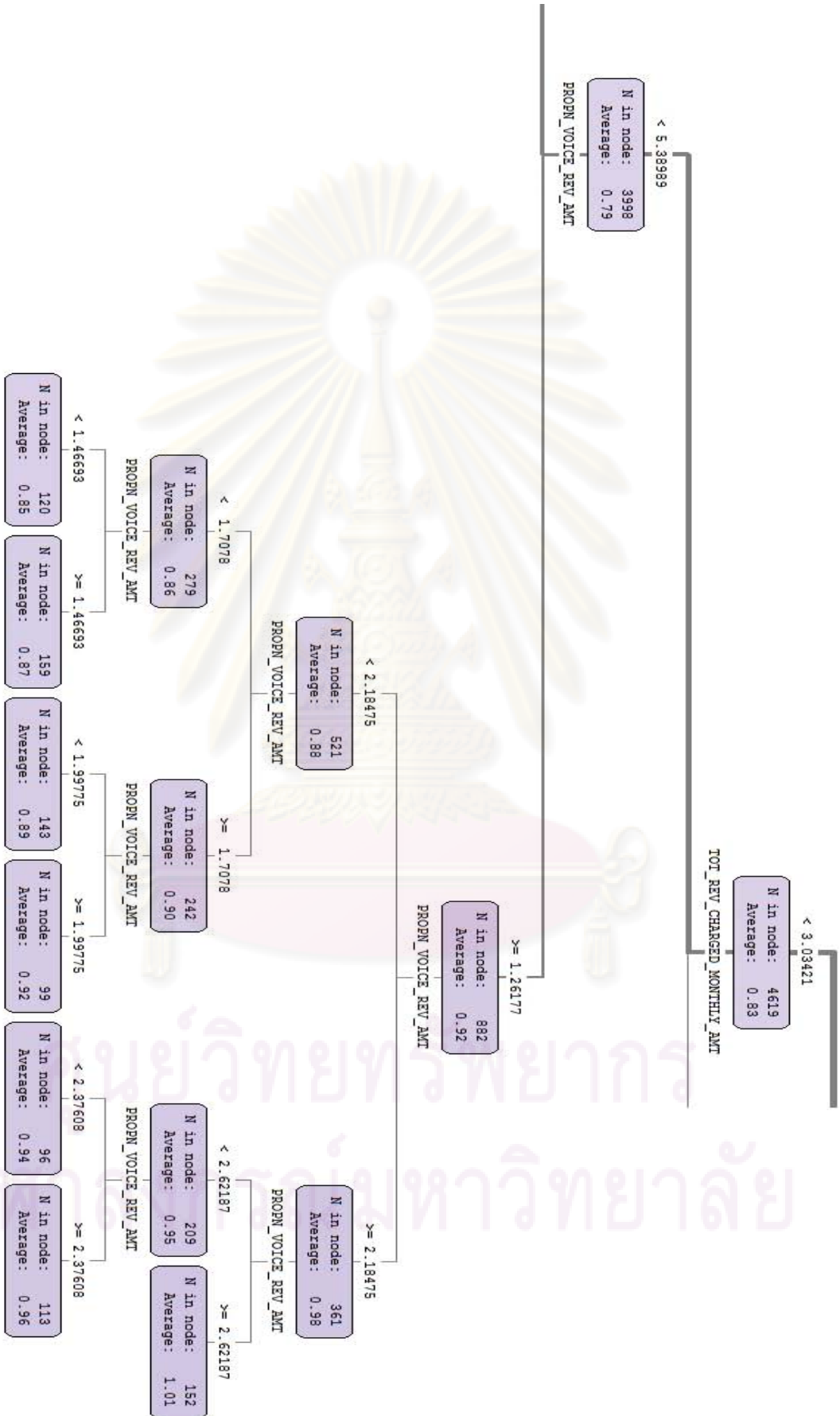
1. **Accuracy** is measured as $acc = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|}$.
2. **Recall** is measured as $rec = \frac{|TP|}{|TP| + |FN|}$ (*true positive rate*). Also known as *sensitivity*.
3. **Specificity** (which is the *true negative rate*) is calculated as $spec = \frac{|TN|}{|TN| + |FP|}$.
4. **ROC curve** (Receiver operating characteristic curve) is plotted as the true positive rate (which is the recall) on the vertical axis against the false positive rate on the horizontal axis for a varying threshold.
5. **False positive rate** is measured as $fpr = \frac{|FP|}{|TN| + |FP|}$. Note that $fpr = (1 - spec)$. As this measure includes the number of TN, it suffers from the same problem as accuracy and specificity.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

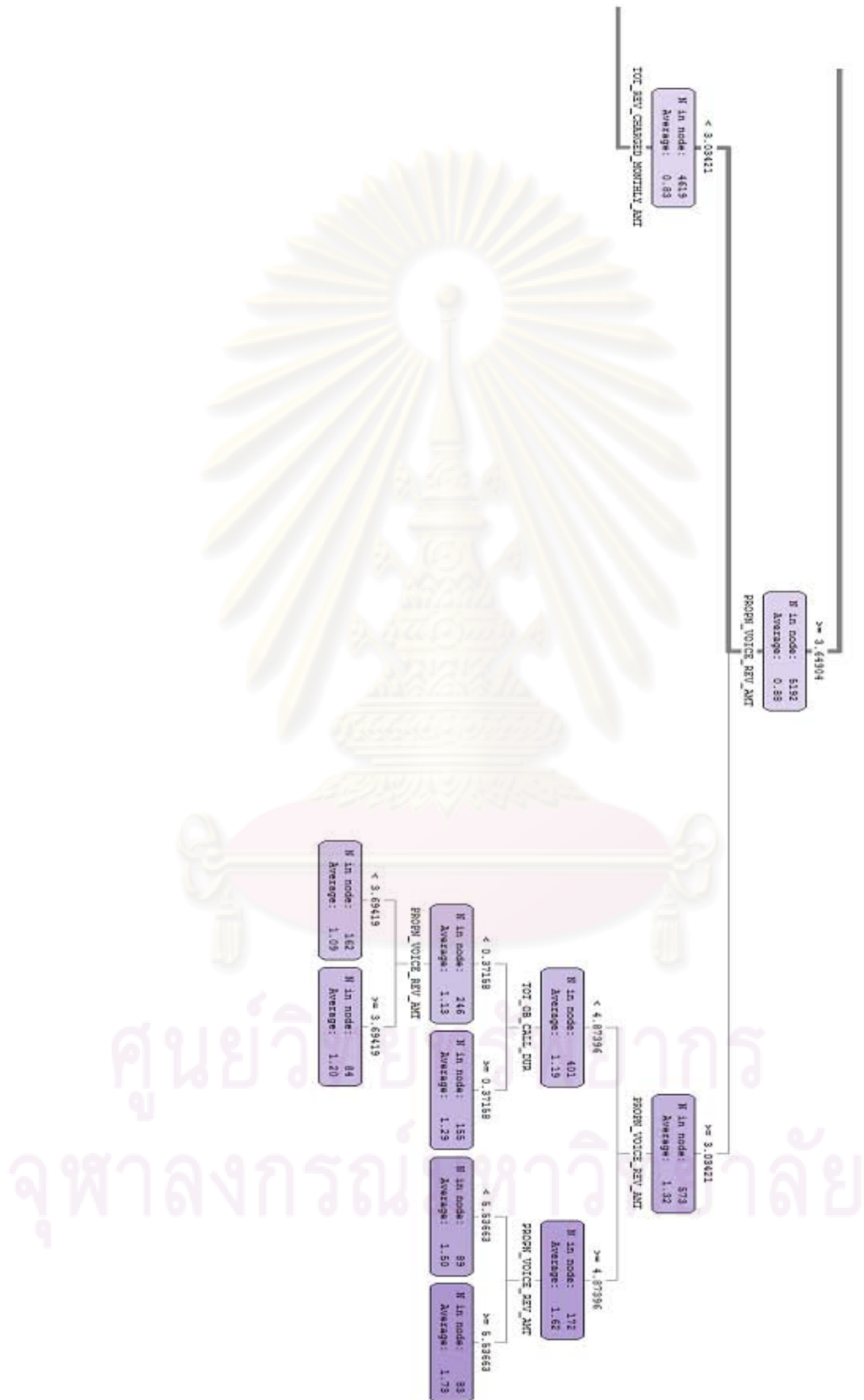
Branch Tree (A)



Branch Tree (D)



Branch Tree (E)



Branch Tree (F)



ศูนย์วิทยทรัพยากร
มหาวิทยาลัย



Appendix E Data fields after PCA

1	Total outbound call duration (locally)	TOT_OB_CALL_DUR
2	Total international call duration	TOT_OB_CALL_INTL_DUR
3	Total outbound data service base on last month	TOT_OB_DATA_CNT_BASE1
4	Duration per call of Average outbound local call	AVG_OB_CALL_DURPERCALL
5	Total Number of Full pay	TOT_FULLPAY_CNT
6	Proportional call on week end calculate weekend call duration divided by total call duration last 6 months	PROP_OB_WKEND_DUR
7	Total revenue SMS amount	TOT_REV_SMS_AMT
8	Percentage of outbound call number per month	PCT_CHNG_OB_LOC_CNT
9	Proportional outbound local calls duration of per call	PROPN_OB_LOC_DUR
10	Proportional duration outbound local on weekend calculate by duration present month divided by average duration last 6 months	PROP_OB_WKEND_DUR
11	Total numbers used promotion	TOT_PRO_CNT
12	Total number partial pay	TOT_PARTIALPAY_CNT
13	Percent in the number changed outbound data services	PCT_CHNG_OB_DATA_CNT
14	Total numbers of days overdue day of payment	TOT_DELIQUENT_DAYS_CNT
15	Percent change outbound call duration.	PCT_CHNG_OB_CALL_DUR
16	Proportional voice service revenue amount calculated by present month voice call amount divided by last 6 months revenue voice call amount	PROPN_VOICE_REV_AMT
17	Total times of overdue 30 days no obtained payment	TOT_30_DAY_DELIQUENT_CNT
18	Total number (times) dropped promotion plan (maybe changed to another)	TOT_PRO_DROPPED_CNT
19	Total Number of Payment last 6 months	TOT_PAY_CNT
20	Total outbound international calls on weekend base on last 2 months	TOT_OB_INTL_CALL_WKEND_CNT_BASE2
21	Percent change in the number of outbound call	PCT_CHNG_OB_CALL_CNT
22	Percent change in the number of SMS services	PCT_CHNG_OB_SMS_CNT
23	Average of duration outbound call on weekend last 6 months	AVG_OB_CALL_WK_DURPERCALL
24	Percent change in the number of outbound call	PCT_CHNG_OB_CALL_CNT
25	Total revenue charge of present month	TOT_REV_CHARGED_MONTHLY_AMT

26	Proportional Partial Pay calculated by present month partial payment divided by Total number of partial payments made over last 6 months	PROP_PARTIALPAY
27	Total number of Miss payment overdue date	TOT_MISSPAY_CNT



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

VITAE

Yi Wang was born in August 30th, 1980, in Tianjin, China. He obtained his Bachelor's degree in Computer Information Management from the Faculty of Information Science and Technology, Tianjin Electromechanical Professional Technology College in 2003.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย