A MODULAR APPROACH FOR SMEs CREDIT RISK ANALYSIS

by

Gülnur Derelioğlu

B.S., Mathematical Engineering, Yıldız Technical University, 2005

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2009

A MODULAR APPROACH FOR SMEs CREDIT RISK ANALYSIS

APPROVED BY:

Prof. Fikret Gürgen ……………….
(Thesis Supervisor)

Prof. Lale Akarun ……………….

Assoc. Prof. Nesrin Okay ……………….

DATE OF APPROVAL: 20.01.2009

ACKNOWLEDGEMENTS

I would like to thank those people who contributed to this research in different phases.

First of all, I would like to thank my supervisor Prof. Dr. Fikret Gürgen, who proposed me this thesis, for his knowledge, experience, support and patience to answer all my questios. With his encourage and guidance, I was able to finish this thesis succesfully.

Secondly, I would like to thank Prof. Dr. Lale Akarun and Assoc. Dr. Nesrin Okay who agreed to participate in my thesis examining committee.

I am very grateful to my family; my parents and my sister, who were always ready to give me unconditioned love and support. Also I would like to thank my cousin Mesut Ünal for his patience and support whenever I need. Without their patience, encourage and understanding, I would never have been able to achieve so much.

Finally, I would like to thank Yapı ve Kredi Bankası A.Ş. for providing me a small subset of the original SME data. I am also grateful to all my colleagues at Yapı ve Kredi Bankası A.Ş. for their great motivation and patience they provided me in each steps of this thesis. And, special thanks to everyone else who contributed to this research only with their motivations.

ABSTRACT

A MODULAR APPROACH FOR SMEs CREDIT RISK ANALYSIS

Credit risk analysis is a challenging problem in financial analysis domain. It aims to estimate the risk occurred when a customer is granted. The risk estimation depends on both customer behavior and economical condition. The challenge is how the credit expert will determine which information should be collected from applicants, under which condition a customer will be classified as good and how much risk will be taken if the credit is granted to the customer. Consequently, credit experts need intelligent customer-specific risk analysis modules to support them when they make these decisions.

In this thesis, we present a cascaded multilayer perceptron (MLP) rule extractor and a logistic regression (LR) model a for real-life Small and Medium Enterprises (SMEs). In the preprocessing phase, the features of Turkish SME database are selected by decision tree (DT), recursive feature extraction (RFE), factor analysis (FA) and principal component analysis (PCA) methods. The best feature set is obtained by RFE. In the first module, the classifier is selected among MLP, k-nearest neighbor (KNN) and support vector machine (SVM). The optimal classifier is obtained as MLP and the following modules are built on MLP. For classification purpose, MLP is followed by neural rule extractor (NRE) in the second module. NRE reveals how the decision is made for customers as being “good”. For the probability of default estimation (PD), we propose a cascaded MLP which is followed by a LR model in the third module. MLP-LR model is followed by clustering method in the last module for scorecard development purpose. In experiments, confidential Turkish SME database is used. The cascaded MLP-LR model provides high accuracy rate and outperforms commonly used classical LR.

ÖZET

KOBİ KREDİ RİSK ANALİZİNDE MODÜLER YAKLAŞIM

Kredi risk analizi, finansal alanda ilgi duyulan problemlerden biridir ve müşteriye kredi verildiğinde oluşacak riski tahmin etmeyi hedefler. Risk tahmini hem müşteri davranışına, hem de ekonomik duruma bağlıdır. Buradaki zorluk, kredi uzmanlarının müşterilerden hangi verileri toplaması gerektiği, hangi koşullarda müşterilerin iyi olarak sınıflandırıldığı ve müşteriye kredi verildiğinde ne kadar risk alındığının tahmin edilmesidir. Bu nedenle, kredi uzmanları ilgili kararları verirken müşteri tipine özel risk analiz modüllerine ihtiyaç duyarlar.

Bu tezde, gerçek Küçük ve Orta Boylu İşletmeler (KOBİ) için, kademeli çok tabakalı yapay sinir ağı-sinirsel kural çıkarıcı ve lojistik regresyon modeli sunuyoruz. Önhazırlık aşamasında, KOBİ veritabanının öznitelikleri; karar ağacı, özyinelemeli öznitelik çıkarıcı, faktör analizi ve temel bileşen analizi ile seçiliyor. En iyi öznitelik kümesi özyinelemeli öznitelik çıkarıcı ile elde ediliyor. İlk modülde, sınıflama metodu çok tabakalı yapay sinir ağı, k-yakin komşu ve destek vektör makinesi arasından seçilmiştir. Optimal sınıflayıcı olarak çok tabakalı yapay sinir ağı elde edilmiş ve takip eden modüller bunun üzerine kurulmuştur. İkinci modülde, sınıflandırma amacıyla çok tabakalı yapay sinir ağını sinirsel kural çıkarıcı takip etmektedir. Sinirsel Kural Çıkarıcı, müşteriler için “iyi” kararının nasıl verildiğini ortaya çıkarır. Temerrüt olasılığının tahmin edilmesi için, üçüncü modülde, lojistik regresyon tarafından takip edilen kademeli çok tabakalı yapay sinir ağı modelini öneriyoruz. Son modülde, skor kartı elde etmek için, çok tabakalı yapay sinir ağı-lojistik regresyon modeli kümeleme metodu tarafından takip edilmiştir. Deneylerde, özel Türk KOBİ veritabanı kullanılmıştır. Kademeli çok katmanlı yapay sinir ağı-lojistik regresyon modeli yüksek doğruluk oranı sağlamaktadır ve genel olarak kullanılan klasik lojistik regresyondan daha üstündür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

w	Eigenvector
∑	Covariance
λ	Eigenvalue
Є	Error
μ	Mean
Acc	Accuracy
ANN	Artificial Neural Network
BIS	Bank for International Settlements
BDDK	Banking Regulation and Supervision Agency
DR_cal	Calibrated Default Ratio
PD_cal	Calibrated PD Values
CRED	Continuous/Discrete Rule Extractor via Decision Tree Induction
DT	Decision Tree
EU	European Union
FA	Factor Analysis
FN	False negative
FP	False positive
GA	Genetic Algorithm
kNN	K-Nearest Neighbor
LR	Logistic Regression
Mcc	Mathew’s Correlation Coefficients
MLP	Multilayer Perceptron
MDA	Multivariate Discriminant Analysis
NN	Neural Network
NRE	Neural Rule Extractor
DA_port	Portfolio Default Average
PCA	Principal Component Analysis
PNN	Probabilistic Neural Network

PD	Probability of Default
Q1	Question 1
Q2	Question 2
Q3	Question 3
Q4	Question 4
RBF	Radial Basis Function
RFE	Recursive Feature Elimination
SVM-RFE	Recursive Feature Elimination with Support Vector Machine
DA_sample	Sample Dataset Default Average
SBA	Small Business Administration
SME	Small and Medium Enterprises
S&P	Standard and Poor
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
DR_unc	Uncalibrated Default Ratio
US	United States

1. INTRODUCTION

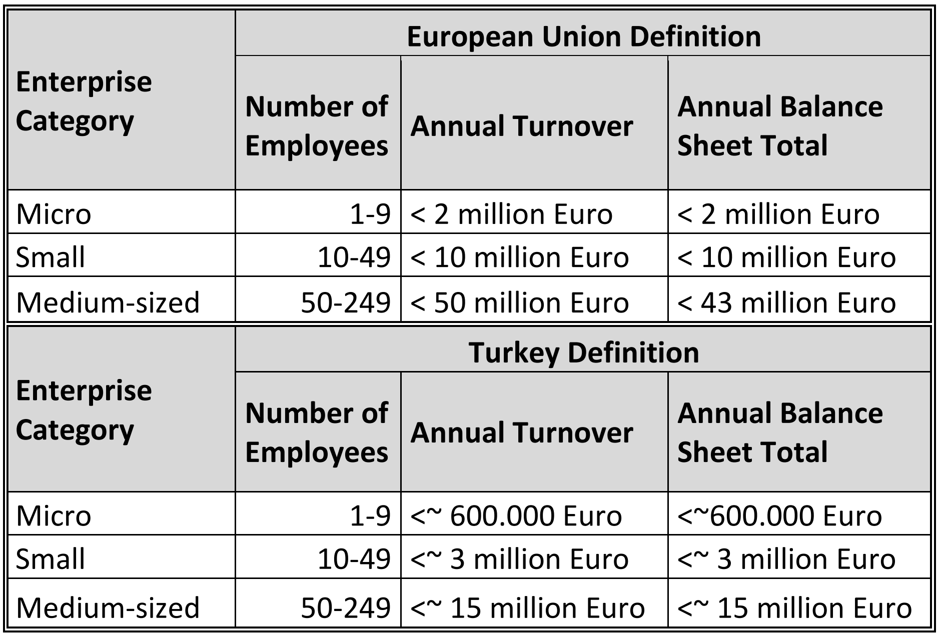
Credit Risk Analysis is an important and challenging data mining problem in financial analysis domain which is commonly used by many financial organizations such as banks etc. It has been taking much more importance since *Basel 2 Recommendations* were released [1] and economical fluctuations has become more often. Basel 2 Recommendations which is commonly known as Basel 2, create an international standard that banking regulators can use when creating regulation about how much *regulatory capital* banks need to put aside to guard against the types of financial and operational risk bank face. As a result, Credit Risk Analysis starts to be a regulatory requirement for the banks not only by Basel 2, but also by high credit growth rate all over the world. In Turkey, *Banking Regulation and Supervision Agency (BBDK)* has been working on determining the *Turkish Banking Regulations* according to BASEL 2 Recommendations and would be compulsory for all Turkish Banks in 2010 [2].

Credit Risk is a general term which implies to future losses. In credit risk analysis, the aim is to decrease future losses by estimating the potential risk and eliminating the new credit proposal if the risk is higher than tolerance value.

Risk patterns of credit risk analysis are generally handled in three groups: *Binary Risk Prediction, Net Risk Value Prediction and Segmenting Customer into Risk Related Groups* [3]. Binary risk predication labels a new customer as either zero for good customer if not risky or one for bad customer if risky. This is also known as *Credit Risk Classification*. Net risk value prediction aims to calculate the probability of a new customer to default. Net risk value is called as *Probability of Default* in Basel 2 Recommendations. Segmenting customer into risk related groups is also known as *Scorecard Development* where the predicted risk is segmented into number of clusters and customers who carry risk in the same range are grouped into the same risk group.

Each country, definition of SME may change. In USA, SMEs are defined by a government office called Small Business Administration (SBA). SBA generated size standards for SME definition such as “500 employees for most manufacturing and mining industries" or "$6.5 million of annual receipts for most retail and service industries" [4]. In European Union (EU), SMEs are defined as enterprises with less than 50 million euro turnover, maximum 250 employees, annual balance sheet total is less than 43 million euro and also no more than 25 per cent of shares are not in ownership of another enterprise [5]. In Turkey, enterprises are accepted as SME if number of employees is less than 250, annual turnover is less than 15 million euro or annual balance sheet total is less than 15 million euro. Also, there is an internal segmentation for SME both in Turkey and EU: *micro enterprises*, *small enterprises* and *medium enterprises* according to their annual turnover and number of employees. Details of segmentation for both Turkey and EU are given in the Table 1.1:

Table 1.1. SME definition for European Union and Turkey



Enterprise Category	**European Union Definition**		
	Number of Employees	**Annual Turnover**	**Annual Balance Sheet Total**
Micro	1-9	< 2 million Euro	< 2 million Euro
Small	10-49	< 10 million Euro	< 10 million Euro
Medium-sized	50-249	< 50 million Euro	< 43 million Euro
Enterprise Category	**Turkey Definition**		
	Number of Employees	**Annual Turnover**	**Annual Balance Sheet Total**
Micro	1-9	<~ 600.000 Euro	<~600.000 Euro
Small	10-49	<~ 3 million Euro	<~ 3 million Euro
Medium-sized	50-249	<~ 15 million Euro	<~ 15 million Euro

On the other hand, Basel 2 standardizes the definition of Small and Medium Enterprises (SMEs) as enterprises according to a single criterion. Basel 2 defines SME as enterprises with annual net sales turnover amount less than 50 million euro [1, 6]. Also, Basel 2 classifies SMEs into *retail SMEs* and *corporate SMEs* according to their credit amount. If credit amount is less than one million euro, SME is accepted as retail SME, otherwise it is accepted as corporate SME. In Turkey, 95 per cent of real enterprises are accepted as SME that reveals the importance of SMEs in the national economy [7]. Not only in Turkey, but also in many developing countries in the world, especially in the recent years, SME Credits have been gaining much more importance according to their high growth in financial world [8]. Furthermore, Basel 2 recommends deeply change for risk analysis of SME credits, starting from the definition of SME to necessary information for risk analysis [1, 6, 7]. On the other hand, in contrast with its increasing growth rate in the world-wide financial sector, there is not enough research for SME credit risk analysis.

Previous studies in this area generally cover risk analysis for corporates, individuals and credit card customers. In contrast with them, in this research, we propose a four level modular approach developed for only SMEs. The proposed method handles all three steps of credit risk analysis according to Basel 2 Regulations: classification, estimation of probability of default and scorecard development. In addition to these, a rule-base extraction level is utilized to understand how the decision is made for customers as being “good”. Our proposed four-level method gives the answers of following four questions:

Q1. Which class will the customer be assigned into?
Q2. How is this decision made?
Q3. What is the probability of a customer to default?
Q4. Which ratings customer would be assigned into?

In the experiments, we use real-life Turkish SME database which is provided by Yapı ve Kredi Bankası A.Ş.

1.1. Motivation

Bank for International Settlements (BIS) was formed in 1930 by governments and private individuals as a part of Young Plan which was a program for settlement of German reparations debts after World War I [1]. Firstly the BIS aimed to collect, to manage and to distribute the annuities payable as reparations. However in the following years, it promoted to central bank cooperation as reparations issue ended. Now, it is the oldest international financial organization in the world and owned by its member central banks. It acts as a bank for central banks, in addition to this, it fosters international monetary and financial cooperation.

Basel Committee on Banking Supervision is an institution provided by the BIS. It was created in 1974 and members are countries central banks: Belgium, Canada, France, Germany, Italy, Japan, Luxembourg, the Netherlands, Spain, Sweden, Switzerland, the United Kingdom and the United States. The Committee provides regular cooperation on banking supervisory matters and famous for Basel 2 Capital Accord which is studied in this work. Basel Committee first agreed on “The International Convergence of Capital Measurements and Capital Standards” and released in 1988 for member countries. This Capital Accord is commonly known as Basel 1 and concentrated on minimal capital requirements on the only credit risk basis. In Basel 1, the aim was providing adequate capital to protect from only credit risks. This minimum capital was proposed for internationally active banks. As a result of the financial fluctuations and banking crises during 1990’s, in 2004, “A Revised Framework on International Convergence of Capital Measurement and Capital Standards” was released and revised later. This is commonly known as Basel 2. Basel 2 is a widely expanded version of the previous accord, covering credit, market and operational risks that banks may face. Also, in Basel 2, minimum capital requirement is mandatory which uses wide range variables for its calculation; from country risk grade to market information, financial fluctuations to credit types. Minimum capital is also known as *regulatory capital*.

Basel 2 defines minimum capital amount as eight per cent of total credit risk-adjusted assets. Regulatory capital becomes very important for financial institutions in the recent years according to the economical fluctuations and high growth rate of credits. Regulatory capital is determined for each financial company under different conditions: country risk grade, customer risk grade, credit amount, credit risk grade and financial ratios. Country risk rating and credit risk grade calculation are determined in detail by Basel 2. Customer risk rating corresponds to probability of default which we aim to obtain in this work Financial ratios are determined according to the international and internal changes. By these criterions, for each credit application, amount of regulatory capital is calculated and put aside as a guard if the credit applicant would not pay the loan back.

In banking domain, each credit which returns back to bank is a source for other credits and other banking products. As seen, this is a critical and continuous circuit. If only a few number of applicants default, there would not be significant hitch on the circuit.

However in the economical crisis terms, a large numbers of applicants fail to pay their debts which badly damage that cycle and may destroy some of these banks. Regulatory capital becomes very important during those terms. Banks use this source as a guarantee for circle continuity. As a result, regulatory capital is vital for credit companies where especially in economically critical years, many credit companies have faced to bankruptcy without regulatory capital.

SMEs form of 95 per cent of real market in Turkey. This reveals how important SMEs and SMEs credits in real sector. In Turkey, generally SMEs have less equity and mostly supply remaining financial needs from bank credit. Thus, Basel 2 regulations would directly reflect and affect SMEs in the real market.

In Turkey, SME Credits are handled separately from individuals or corporate credits as many countries in the world. SME Credits are specialized according to the SMEs general needs and sector-special needs. General needs can be defined as common need of the SMEs such as Cash Support Credits. Cash Support Credit is proposed to meet capital need of SME. Sector-special needs change for each sector such as Greenhouse Construction Credit or New Session Preparation Credit. Greenhouse Construction Credit is proposed for SMEs which are working on agriculture and greenhouse. New Session Preparation Credit is proposed for SMEs in the tourism sector. Each types of credit have different limits under different conditions. These limits range according to types of credit and types of SME. As an example, Greenhouse Construction Credit is limited by the greenhouse total-cost. However Greenhouse credit is unlimited, it covers only cost of seed, manure and pest control. On the other hand unlimited does not corresponds to that each applicant can get how much they apply for. The unlimited means applicant can get how much they need and afford to pay back. These limits are determined by the bank internal decision criterions according to the each applicant demographic and financial information.

As stated earlier, regulatory capital is non-working capital which means, it can not be used as a source for new credits. If the credit risk analysis does not perform properly and realistic, it would be possible to keep regulatory capital more than necessary which causes a great amount of unpredicted loss for the bank. This unpredicted loss directly reflects on credit costs of the credit applicants. When the huge credit amount is taken into consideration, the SME credit analysis becomes critically important in the real sector.

1.2. Previous Works

Credit Risk Analysis is an appealing topic where a one per cent improvement in accuracy, which seems insignificant, will reduce losses in a large loan portfolio and save billions of euro. Thus, there have been many techniques proposed for all three segments of credit risk analysis. These techniques are generally based on non-linear classifier in order to handle real-life datasets. Neural Network (NN) is one of the most popular method that succesfully acquire the knowledge in the given data and used for both classification and probability of default (PD) estimation. Pang, Wand and Bai [9] obtained 97.67 per cent accuracy with their Neural Network based credit risk classification model in 2002. The dataset they used is real life dataset obtained from China. Yeh and Lien [10] compared many data mining techniques applied on Taiwan real-life data for credit risk analysis. They obtained that neural networks are powerful on estimating PD than other techniques such as logistic regression.

Discriminant Analysis is also widely used in this area. Support Vector Machines (SVM) have expanded application domain and become also popular in credit risk analysis domain. Yang and Duan [11] applied SVM on real life data obtained from Shijiazhuang City Commercial Bank. They also applied SVM on the same data after reducing the dimensions with Principal Component Analysis (PCA). They obtained classification correct rate over 90 per cent with PCA based SVM. Zhou and Bai [12] tried Genetic Algorithm (GA) based SVM that outperformed standard SVM. In order to handle non-linear data, kernelized methods have been proposed. Wei, Li and Chen [13] applied SVM with mixture of kernel on one of major US commercial bank and they obtained good classification performance.

Logistic regression (LR) is another discriminant analysis method which is widely used in credit analysis domain for both classification and PD estimation [14]. Rahayu, Purnami and Embong compared classification accuracy of logistic regression with kernelized logistic regression on German Credit Dataset [15]. They obtained better results with kernelized logistic regression than logistic regression. Also, Kaya, Gürgen and Okay obtained good results for PD estimation by logistic regression on German Credit Dataset [16].

k-Nearest Neighbor (kNN) is a non-parametric classifier which is also used for credit classification. Gaganis, Pasiouras, Spathis and Zopounidis compared the classification performance of kNN with discriminant analysis and LR where kNN outperformed both them on the FAME dataset obtained from Bureau Van Dijk's Company [17]. Galindo and Tamayo applied kNN on mortgage loan dataset provided by Mexico's security exchange and banking commission: Comision Nacional Bancaria y de Valores (CNBV), although kNN's performance was lower than neural networks, it outperformed standard logistic model [14].

Although many researches have been done on credit risk analysis for large corporates, personal credits and credit cards, there have been only a few works for SME Credit Risk Analysis. From a credit risk point of view, SMEs show different behavior than corporates and individuals. Altman and Sabato remind that according to their analysis on German and French SMEs, SMEs are riskier with lower assets correlation with each other than large corporates [18]. Thus, models developed for corporates would not be suitable for SMEs. Altman and Sabato developed a one year default prediction model based on LR using United States (US) SMEs data from WRDS COMPUSTAT database. They indicated that the proposed method outperformed Multivariate Discriminant Analysis (MDA). Fantazzini and Figini also proposed Random Survival Forest Model which gave slightly better performance than classical logistic model on the real-life dataset obtained from Creditreform [8].

1.3. Proposed Method

In this research, a modular method is proposed as a response to the four research questions mentioned in the previous subsections. The proposed solution has four modules, each module corresponds to one questions. The main flow of proposed method is given in the Figure 1.1.

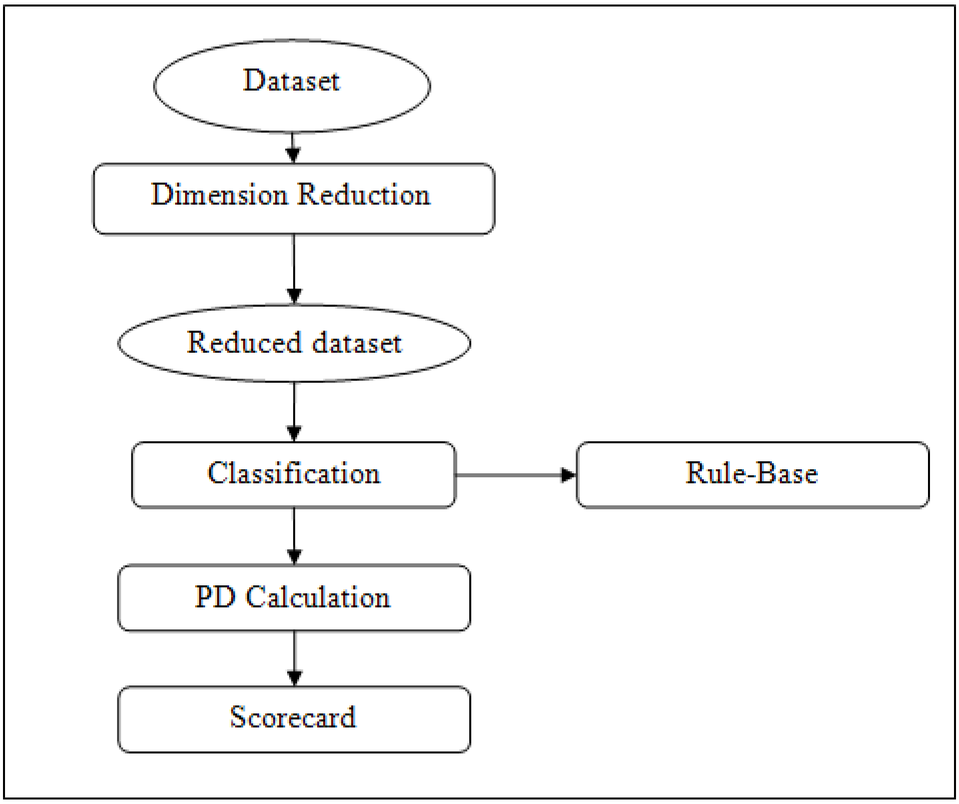


Figure 1.1. The proposed modular approach

As a preprocessing for the proposed method, we use dimension reduction techniques to reduce the input data volume. Although we use small subset of the original SME portfolio which is given in detail in subsection 1.4, when the real portfolio size is taken into consideration, dimension reduction becomes indispensable phase of the proposed method. The detailed flow diagram for the dimension reduction phase is given in the Figure 1.2. In the dimension reduction phase, we apply Decision Tree (DT), Recursive Feature Elimination with Support Vector Machine (SVM-RFE), Factor Analysis (FA) and Principal Component Analysis (PCA) on the original dataset. They produce five different reduced sets of data from original dataset.

In the first module, for performance comparison of dimension reduction algorithms, we use each reduced set of data as input for classification methods. Then, we compare classification performance of each classifier for five reduced input data and original dataset. According to the performance results, we choose the optimal classifier and reduced input we will continue with. This module produces the response for the second research question. The detailed flow chart for the first module is given in the Figure 1.3.

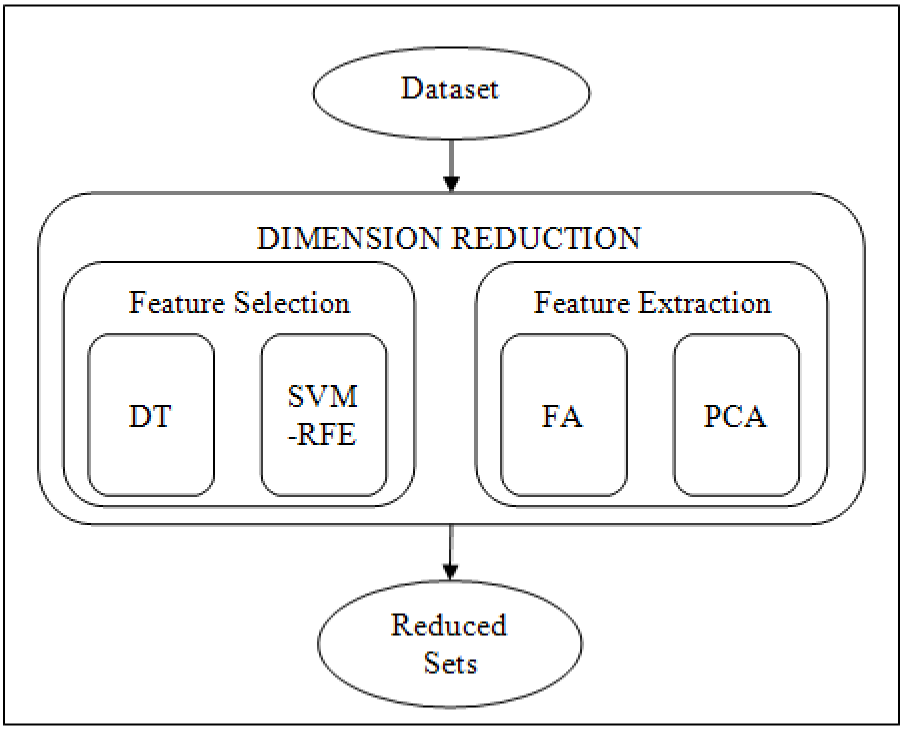


Figure 1.2. Dimension reduction phase

In the second module we apply neural rule extractor to reveal how the MLP reached at the final decision on the optimal input. From the results, we form a rule-base showing under which circumstances a customer is classified as good.

In the third module, for probability of default estimation purpose, we propose a cascaded multilayer perceptron model that is followed by logistic regression. Multilayer perceptron (MLP) is a successful algorithm to acquire underlying information of the input. Thus we apply logistic regression on the results of MLP. The flow diagram for PD calculation phase is given in the Figure 1.4.

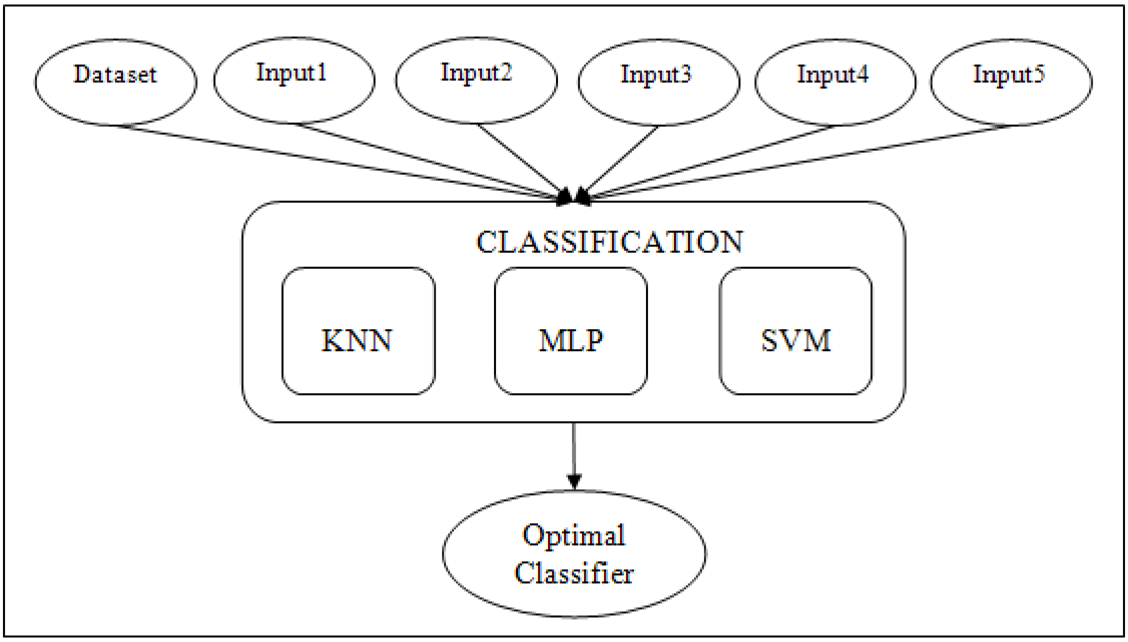


Figure 1.3. The first module: classification

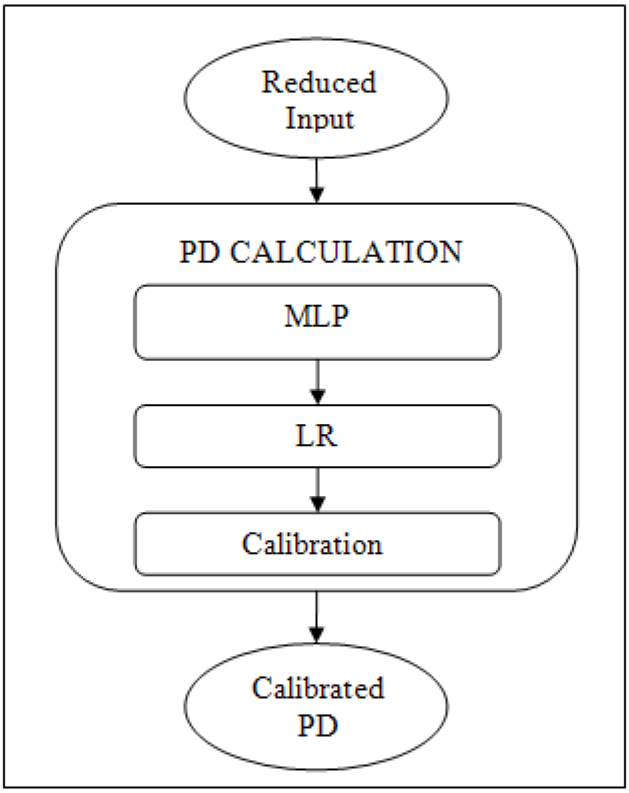


Figure 1.4. The third module: MLP-LR model for PD calculation

As a response for the last research question, we propose k-means clustering algorithm to segment calibrated PD into risk groups for scorecard development phase in the last module. In the real market, mostly 10, 16 and 20 segments are preferred. When number of segments increases, risk calculation becomes more sensitive.

1.4. Dataset

The real-life dataset is provided by Yapı ve Kredi Bankası A.Ş. and consists of Small and Medium Enterprises information that is collected from credit applicants between January 2006 and April 2007. This information covers not only the financial background of the applicants but also demographical and delinquency information of SMEs. Dataset is obtained randomly, without any sampling methodology, only a small subset of the original portfolio is taken. Thus we do not affirm that the dataset reveals behavior of the whole SME portfolio perfectly.

Dataset has 512 samples with 27 features and a class variable which extracts if the applicant was classified as good customer who paid the loan back on time or bad customer who did not pay the loan and defaulted. In the dataset, good and bad customer distribution is not homogeneous as 144 customers (28 per cent) were classified as good customers and 368 customers were classified as bad customers.

The dataset consists of 27 features; six of them are *categorical* and the others are *continuous* variables. These features mainly cover four different types of information: *Demographical, Financial, Risk* and *Delinquency* information. Demographical information includes the customer-based information collected during the application, such as age of the SME. Risk information is collected after customer application, during approval level. It covers the other products risks of customers in the bank. Financial features are collected both during applications and during approval. For example, net annual turnover is collected during application however total amount of existing unsecured exposures is collected during approval. Delinquency information is collected during approval which shows if the customer has been late for any of his products in the bank before. The features of the dataset, their data types and information distribution are shown in Table 1.2.

Table 1.2. Features and distribution of the dataset

Feature Code	Data Type	Feature Information
A1	Categorical	Risk
A2	Continuous	Risk
A3	Continuous	Financial
A4	Continuous	Financial
A5	Continuous	Financial
A6	Categorical	Risk
A7	Categorical	Demographic
A8	Categorical	Demographic
A9	Continuous	Demographic
A10	Continuous	Demographic
A11	Continuous	Demographic
A12	Continuous	Financial
A13	Continuous	Financial
A14	Continuous	Financial
A15	Continuous	Financial
A16	Continuous	Financial
A17	Continuous	Financial
A18	Continuous	Financial
A19	Continuous	Delinquency
A20	Continuous	Delinquency
A21	Categorical	Delinquency
A22	Continuous	Delinquency
A23	Continuous	Delinquency
A24	Continuous	Delinquency
A25	Continuous	Delinquency
A26	Continuous	Delinquency
A27	Categorical	Delinquency

1.5. Outline

In the previous parts, general information about credit risk analysis, the dataset we work on and the method we propose to solve the problems of research domain are given. In the following chapters, we continue to describe the proposed model in detail. In Chapter 2, dimension reduction techniques are described in detail. Although we use a small subset of original dataset, to develop a comprehensible model, we also work on dimension reduction techniques. The third chapter covers all four modules of the proposed model: classification, rule-base development, PD estimation and scorecard development. In the fourth chapter, we give experimental results for all four modules and their comparison. Furthermore, to make the comparison, the necessary performance metrics are explained in detail. Finally, the overall conclusion and discussion of the proposed approach are given in the Chapter 5.

2. DIMENSION REDUCTION

In data mining applications, the time and space complexity of any classifier or regressor directly depends on the input data size [19]. *Dimensionality Reduction* techniques can be applied to the dataset to obtain a reduced representation of the input data which have less number of variables without losing the integrity of the original data [20]. Briefly, dimension reduction decreases the number of input data variables while remaining the information that the original dataset contains.

Dimensionality Reduction techniques are generally applied as a preprocessing step in data mining applications, not as a part of learning algorithms. Thus not only the time and space complexity is reduced but also it is possible to obtain more robust but simpler model [19]. These techniques can be divided into two different groups: *Feature Selection* and *Feature Extraction*. Feature Selection aims to obtain a subset of the original dataset features with minimum loss of information. In Feature Selection, we are trying to find the best *k* dimensions of *d* original dimensions that remains the most information of the original dataset and omit the other *(d – k)* dimensions. There are two approaches for feature selection algorithms: *forward selection* and *backward selection*. Forward selection algorithm starts with no variables and in each step continues with adding the most relevant feature which decreases the error most. Backward selection algorithm starts with all variables and continues with leaving the most irrelevant feature that does not decrease the error [19]. Infogain is a popular example of subset selection algorithms. On the other hand, feature extraction aims to find a new set of features that are the combinations of the original features. There are different techniques to extract the new feature set. Principal Component Analysis and Factor Analysis are well known and widely used Feature Extraction Methods.

In the proposed approach, we try to obtain a strong input with smaller volume without losing accuracy. The dataset we use in this research is real life financial dataset and the information it contains should remain thoroughly in the case of proposed method feasibility. On the other hand, the real life data is collected from SME and some of this information can be non-informative. Thus, reducing the dimension and revealing the underlying information can be quite important in this work. However, as widely accepted, even though a large number of algorithms have been developed, there is no precise algorithm for dimension reduction techniques [21]. As a reason, we try both feature selection and feature extraction algorithms then prefer to use the one which gives the highest accuracy.

2.1. Feature Selection

As stated earlier, Feature Selection algorithms select the most relevant k features of d original features and discard the unnecessary (d – k) ones. The main objectives of feature selection algorithms are [22]:

- Selecting highly informative variables can improve the model accuracy and reveal the underlying process which generates the original data
- Decreasing the number of inputs avoid from over fitting
- Small volume of data provides faster and less-complex model

Feature Selection techniques are also called *Subset Selection* and can be handled in two groups: *Filter Methods* and *Wrapper Methods*. Briefly, in filter methods, the most relevant k dimensions are determined by an evaluation function using certain measure such as distance or entropy etc. That process is independent from actual learning algorithm [10]. However in wrapper methods, the learning algorithm is used to estimate the value of a given subset [24, 25]. When filters select features that maximize the evaluation function, wrappers select features that optimize the performance of actual learning algorithms. Generally, wrapper methods give higher accuracy than filter methods despite wrapper's evaluation criterions directly depend on induction algorithm of the learning methods [25]. Wrappers may cause excessive computational complexity according to retraining the learning method for each subset considered. On the contrary, filters are faster and computationally more efficient methods that make filters more efficient than wrappers on large volume dataset [25].

Many approaches have been proposed for feature selection such as the well known methods of decision tree as filter and recursive feature elimination with support vector machine as wrapper. Decision tree utilizes tree induction algorithm with the entropy as an evaluation measure [26], on the other hand SVM-RFE tries to minimize cost function as performance measure [27].

2.1.1. Decision Tree

Decision tree is a well known hierarchical data structure for supervised learning and used for both classification and regression. Decision tree learning algorithm is greedy and based on divide-and-conquer.

A decision tree has two main components: decision nodes and terminal leaves. Each decision node applies its test function to the given input and produces a discrete value that determines which branch is taken. A decision node creates a discriminant in the d-dimensional input space and dividing it into smaller regions as shown in Figure 2.1. Each leaf has an output label for all income which is a class label for classification problem and a numeric value for regression problem.

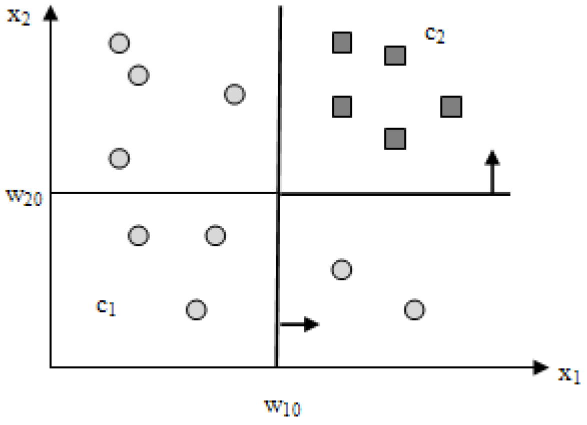


Figure 2.1. Decision node discriminant

Decision tree can be examined in two sub-groups: *Univariate Trees* where each internal node use only one variable as is shown in Figure 2.2 and *Multivariate Trees* where all features can be used in each decision node.

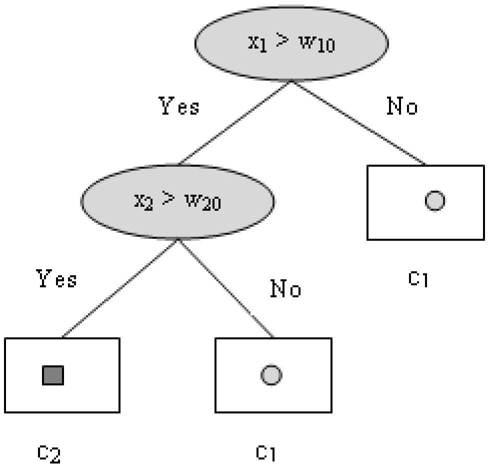


Figure 2.2. Univariate decision tree

In a univariate classification tree, learning starts at the root node with all features and the aim is obtaining the best split. This process continues recursively with the corresponding subset until a leaf node is obtained. The measure of the good split is impurity which is determined as if all instances of the branch are labeled as the same class.

$$\hat{P}(c_i \mid x, m) = p_m^i = \frac{N_m^i}{N_m} \tag{2.1}$$

For node m, N_m is the number of training instances reaching node m and N_m^i of them belong to class c_i. Node m is pure if p_m^i is zero or one.

The measure of impurity is *entropy* [19]. The best split is obtained when entropy is minimized. Entropy formula for node m is given in Equation 2.2.

$$\mathrm{I}_m = -\sum_{i=1}^{k} p_m^i \log_2 p_m^i \tag{2.2}$$

Decision tree is also known as a feature selection algorithm. The final univariate tree consists of the most relevant features and discards irrelevants. In this work, we use C4.5

tree as a feature selection method [28]. C4.5 tree is a univariate classification tree and recursively searches the input data until maximizes the classification performance and extracts the features that create the best splits. We use J48 tree which is the C4.5 decision tree implemented in *Weka* [29].

2.1.2. Recursive Feature Elimination with Support Vector Machine

Recursive Feature Elimination (RFE) is a wrapper method that utilizes the generalization capability embedded in support vector machines (SVM). RFE keeps the independent features containing the original dataset information while eliminating weak and redundant features [30]. However, the subset produced by SVM-RFE is not necessarily the ones that are individually most relevant. Only taken together the features of a produced subset are optimal informative [31].

The working methodology of SVM-RFE is based on backward selection where algorithm starts with whole features and iteratively eliminates the worst one until the predefined size of the final subset is reached. At each iteration, the remaining features must be ranked again [32].

SVM-RFE working principles at each iteration could be examined in three steps:

- Training the classifier (SVM)
- Computing the ranking criterion for all features
- Removing the feature with smallest ranking criterion

There are different ranking criterions proposed for SVM-RFE such as entropy [33] or square of the weight of separating hyperplane (w^2) [32]. In this work, we use Weka SVM-RFE tool [29] with square of weight as ranking criteria where in each iteration the feature which causes minimum variation in the SVM cost function is removed from feature space. We assume that in each step, trained SVM produces weight vector w^* according to the formula below where α_i are Lagrange multipliers which are greater than zero for support vectors:

$$w^* = \sum_{i \in SV} y_i \alpha_i^* x_i \tag{2.3}$$

For the trained SVM with the weight vector w^*, the cost function is $J(w)$:

$$J(w) = \frac{1}{2} \| w \|^2 \tag{2.4}$$

In order to find the variation in cost function of SVM ($\delta J(i)$):

$$\delta J(i) = \frac{1}{2} \frac{\partial^2 J(w)}{\partial w_i^2} (\delta w_i)^2 = \frac{1}{2} (w_i)^2 \tag{2.5}$$

Feature, which causes minimum variation is ranked and removed from feature space. SVM-RFE algorithm is given in Figure 2.3. In SVM-RFE, computational cost is higher while only one feature is removed in each step. When several features are removed at a time, feature subset ranking must replace with feature ranking.

```
Function RFE-SVM(TD, AF, RS)
Initialize
        TD : Training data
        AF : All Fetures in the dataset
        RS : Reduced feature subset
Begin
      While( number of AF > RS)
                Train SVM on TD with the feature space AF
                Rank the features of F in the descending order
                RFS := AF – { feture with the smallest rank in AF}
                AF = RFS
        End
        Return AF
end
```

Figure 2.3. RFE-SVM Algorithm

2.2. Feature Extraction

Feature extraction aims to replace original variables by a smaller set of underlying variables. It uses linear transformation while transforming all variables to a reduced dimension space without loss of information [34]. In recent research, kernel and nonlinear transformation techniques are proposed for feature extraction [35-37]. Principal Component Analysis (PCA) and Factor Analysis (FA) are widely used feature extraction algorithms that both use linear transformation. In this work, we use principal component analysis to reduce the input data dimension and compare the factor analysis results to reveal the underlying factors of original dataset.

2.2.1. Principal Component Analysis

PCA is well-known and widely used supervised feature extraction method which transforms possibly correlated variables into smaller number of uncorrelated variables. PCA tries to maximize variance of features and use covariance matrix of input variables to obtain eigenvector and their corresponding eigenvalues. In PCA, to determine the optimal number of dimensions, *proportion of variance* is used which is preferred to be higher than a predefined threshold value. In this work, we use PCA implemented in *Matlab R2007a* [38], 0.90 as threshold for proportion of variance. Assume that the input data with d dimension, then proportion of variables is calculated according to the formula below where λ_i is the eigenvalue of eigenvector w_i and λ_i are in the decreasing order:

$$\frac{\lambda_1 + \lambda_2 + \ldots + \lambda_k}{\lambda_1 + \lambda_2 + \ldots + \lambda_k + \ldots + \lambda_d} \tag{2.6}$$

The *principal components* are the eigenvectors with the highest *k* eigenvalues which meet the proportion of variance. In order to obtain k dimensional reduced set, the linear projection is applied to principal components on original data.

2.2.2. Factor Analysis

Factor Analysis is an unsupervised *Feature Extraction* method and is trying to explain the dataset in terms of its common underlying factors with minimum loss of information. Factor Analysis is generally used for extracting underlying relationships in the data and assessment of the extracted information which is explained with fewer variables [39].

Factor analysis assumes that observable variables are linear combinations of underlying factors and error terms. Let x be our input data with d dimensions $x_1, x_2, .., x_d$ that have the mean µ and covariance matrix ∑.

$$x_i = v_{i1} f_1 + v_{i2} f_2 + ... + v_{im} f_m + e_i \ , \ \text{m} < \text{d} \tag{2.7}$$

In matrix notation:

$$X = VF + e \tag{2.8}$$

FA has three main assumptions:

1. Error terms e_i are independent where $mean(e_i) = 0$ and $\text{Var}(\text{e}_\text{i}) = \sigma^2$
2. Factors f_j are independent of one another and of the error terms.
3. Factors are standardized where $mean(f_j) = 0$ and $\text{Var}(\text{f}_j) = 1$

In FA, the key measurement is correlation between observable variables. If two variables are highly correlated that indicates these two variables are related by factors. FA calculates the eigenvectors and corresponding eigenvalues from input correlation matrix. Then FA determines the optimal number of factors by two different criterions: proportion of variance or *Kaiser Criterion* [40] where factors with eigenvalue greater than one are chosen. *Factor loading* which is indicated as V in Equation 2.8, represents the correlations of variables with the factors. Factor loading is a linear projection of m eigenvectors and square roots of corresponding eigenvalues. The FA reduced set is obtained by projection of factor loadings V and estimated covariance of V on observable variables. In this work, we use *SAS Enterprise Guide* tool [41] to determine the number of factors underlying the real life financial dataset. SAS Enterprise Guide produces factors according to Kaiser Criterion. Then factor loading and reduced set is calculated by Matlab R2007a.

3. CREDIT RISK ANALYSIS

Credit Risk Analysis can be examined in three different subgroups as Credit Risk Classification, Estimating Probability of Default (PD) and Customer Segmentation. In credit risk classification, a credit applicant is classified either good customer who pays the loan on time or bad customer who does not pay the loan back. Probability of default (PD) indicates the probability of the customer to default. In PD estimation, PD value which is closer to zero corresponds that the customer has low credit risk. On the other hand PD value which is closer to one corresponds that the customer has high credit risk and can be accepted as bad borrower and *Customer Segmentation* where customers with almost same risk are assigned into the same risk segment. In credit risk analysis, there are two key points, the first one is minimizing number of debtors estimated as good borrowers with low risk who, in actual, are bad borrowers with high risk. The second key point is that the average estimated default probability of given portfolio should be close to the given country international ratings.

3.1. Credit Risk Classification

Credit Risk Classification is the most common credit risk analysis method. Credit risk classification techniques aim to estimate if a borrower could pay the loan back or not. If the credit applicant is estimated as good borrower, this corresponds that he/she could pay the loan back on time. On the other hand when the applicant is estimated as bad borrower, this corresponds that he/she could have difficulties to pay the loan back on time, in the worst case he/she could not pay the loan back. As seen, these techniques divide the customer portfolio into two groups: good customers and bad customers.

In literature, Credit Risk Classification techniques are generally statistical techniques such as discriminant analysis or data mining techniques such as kNN, SVM etc. In this work, we only focus on data mining techniques to classify our SME portfolio. We apply kNN, MLP and SVM on the SME input and compare their results. For the proposed approach, we use the one giving the highest classification accuracy.

3.1.1. K-Nearest Neighbor

K-Nearest Neighbor (kNN) is a well-known non-parametric classifier which classifies a new instance according to the majority class of the k closest training data points. K is generally chosen as an odd number to minimize confusion between two neighboring classes. The measure of closeness is in terms of d dimensional input space. There are different measurements such as *Euclidean Distance* or *Mahalanobis Distance*. Euclidean distance is a linear distance between two points which is given in Equation 3.1. Mahalanobis Distance calculates the distance between two data points by the variation in each component of the points which is given in Equation 3.2. [42]:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^{2}} \qquad (3.1)$$

$$d(x,y) = \sqrt{(x-y) \Sigma^{-1} (x-y)} \qquad (3.2)$$

After distances between training data and new instance are calculated, k nearest neighbors are determined. Then, the class probabilities are calculated as a proportion of the number of training instances which belong to class *i* to the total number of training instances. In this work, we use Weka IBK to apply kNN on real life financial dataset. We prefer to consider five closest neighbors thus we use five as k.

3.1.2. Multilayer Perceptron

Multilayer Perceptron (MLP) is a nonparametric neural network structure and used for both classification and regression. Feedforward MLPs are the most widely used Artificial Neural Network (ANN) models. MLP is composed of three layers: an input layer, hidden layers and an output layer. A three-layer MLP is shown in Figure 3.1. In MLP, using one hidden layer is generally preferred in the case of reducing the complexity. Furthermore, large number of hidden units may cause overfitting, thus hidden layer may contain either predefined number of hidden units or optimal number of hidden units can be determined during learning.

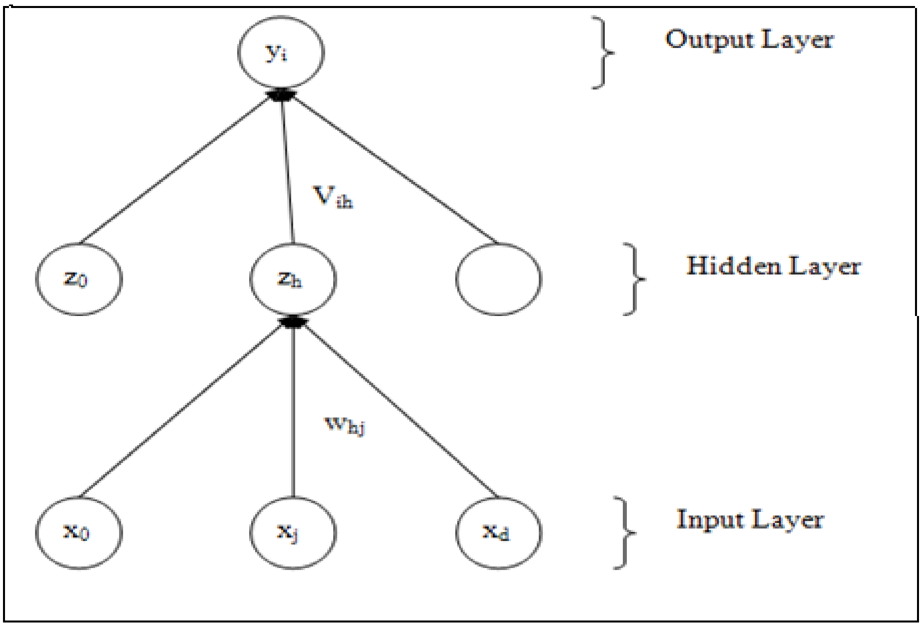


Figure 3.1. Three-layer perceptron

MLP learning process starts at the input layer where no calculation is applied. Briefly, hidden units nonlinearly transform the d dimensional input space to h dimensional space. The output units produce the output values as linear combinations of the h dimensional activation values computed by hidden units [19]. MLP learning algorithm used in this work is shown on Figure 3.2. At initialization step, weights are initialized in the range of [-0.01, 0.01]. Then, in each epoch, weighted sum of input variables are sent as input to hidden units where nonlinear activation function is applied. Hidden units produce h dimensional data as inputs for output unit which calculates weighted sum of inputs to produce output value. In back-propagation algorithm, output value of each layer is used for previous layer weight updates. This process continues until one of the stopping criterions is reached.

```
Initialize all $v_{ih}$ and $w_{hj}$ in the range of (-0.01,0.01)
Repeat
        For all $(x^t, r^t) \in X$
                For h= 1,2,…,H
                        $z_h \leftarrow sigmoid(w_h^T x_t)$
                For i = 1,2,…,d
                        $y_i = v_i^T z$
                For i = 1,2,…,d
                        $\Delta v_i = \eta(r_i^t - y_i^t) z$
                For h = 1,2,…,H
                        $\Delta w_h = \eta(\sum_i (r_i^t - y_i^t) v_{ih}) z_h (1 - z_h) x^t$
                For i = 1,2,…,d
                        $v_i \leftarrow v_i + \Delta v_i$
                For h = 1,2,…,H
                        $w_h \leftarrow w_h + \Delta w_h$
Until convergence
```

Figure 3.2. Pseudo code of MLP learning algorithm

In this work, we use Weka Multilayer Perceptron function, trained by back-propagation algorithm. Back propagation algorithm updates the current values according to the predicted output value of previous layer. We use 0.8 as learning rate and maximum 500 epochs are allowed. Only one hidden layer is preferred with five hidden units. As activation function, *sigmoid* is used which is given in Equation 3.3. Sigmoid produces the output in the [0, 1] range.

$$f(u) = 1/(1 + e^{-u}) \tag{3.3}$$

As output is produced by sigmoid and the problem is binary class problem, we use sigmoid(0) = 0.5 as threshold value for good class [19]. If the produced output is greater than 0.5, credit applicant is accepted as bad customer, otherwise good customer.

Furthermore, we apply Probabilistic Neural Network (PNN) for classification on our input space. Probabilistic Neural Network is a feed-forward neural network structure. It classifies the samples according to the probability density function of each class. The data sample will be assigned into the class with the greatest probability density function value. In this work, we use PNN program implemented in Matlab that utilizes radial basis function in the hidden neurons.

3.1.3. Support Vector Machine

Support Vector Machine (SVM) is a discriminant-based method and used for both classification and regression. In classification, SVM tries to find the optimal separating hyperplane which maximizes the distance between data points from different classes as shown in Figure 3.3.

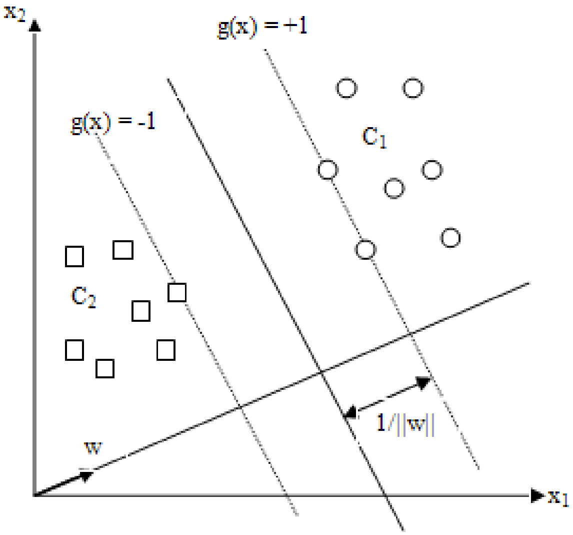


Figure 3.3. Support vector machine

The distance from the hyperplane on each side is called as *margin* and SVM tries to maximize the margin. For two class problem, assume that the sample data $X=(x^t, r^t)$ where *x* is the input, *r* is the target value and $r \in \{-1, +1\}$. The separating hyperplane can be expressed as function of *x* [19, 43]:

$$g(x) = w^t x^t + w_0 \tag{3.4}$$

where $g(x) \geq 1$ for $r^t = +1$ and $g(x) \leq -1$ for $r^t = -1$. The distance of x^t to the hyperplane that we want to maximize should be greater than ρ for $\forall t$. The formula is given in Equation 3.5. Thus, in order to obtain maximum margin, we should minimize $\| w \|$ [19, 43].

$$\frac{r^t (w^t x^t + w_0)}{\| w \|} \geq \rho \tag{3.5}$$

The cost function in SVM is obtained as stated in Equation 3.6 where the problem is converted into an optimization problem.

$$\frac{1}{\| w \|^2} \tag{3.6}$$

When the minimum *w* is found, the maximum margin can be calculated by the formula given in Equation 3.7.

$$\rho = \frac{1}{\| w \|} \tag{3.7}$$

The data points which lie on the margin are called as *support vectors* which satisfy the formula given in (3.8).

$$r^t (w^t x^t + w_0) = 1 \tag{3.8}$$

SVM can also handle non-linear problem by mapping the input space into non-linear space by non-linear transformation. For transformation, different kernels such as *Polynomial Kernel* or *Radial Basis (RBF) Kernel* is widely-used. In this work, we use SVM as classifier with different kernels and compare the results with each other and also with other classifiers. As SVM tool, we use Weka *SVM (SMO)* tool and its containing kernels.

3.2. Classification Rule-Base Development

Credit risk classification techniques make the final decisions for credit applicants. These techniques work online and do not give any information about how they arrived at the final decision. To reveal under which circumstances an applicant is assigned as good or bad, we propose a rule-extraction algorithm from trained MLP. There are several rule-extraction methods such as DecText etc. DecText only handles categorical data. The dataset we use mostly includes continuous data. As a reason, we prefer to use CRED which does not need to discretize the input variables and can handle continuous variables successfully.

3.2.1. Neural Rule Extraction

Neural Network is a black box data mining model which acquires hidden knowledge in dataset with high accuracy rate. On the other hand, understanding how neural network arrived at its decision is not easy. Because these are represented by the weights on the connections and activation functions of hidden and output nodes. These representations are not easily understandable and not useful in practice. For this reason, several techniques have been proposed for rule extraction from trained neural network. Some of these techniques aim to obtain rule-base from trained neural network such as CRED [44] while others extract decision tree such as DecText [45] and Trepan [45, 46].

CRED (Continuous/Discrete Rule Extractor via Decision Tree Induction) is used in this work. CRED is composed of four steps:

- Step1: Train a neural network with three layers: an input layer, a hidden layer and an output layer.

- Step2: Build a hidden-output tree. The input variables are activation values of hidden units and output variable is target value produced by neural network.

Activation value of *m*-th hidden unit is calculated according to the formula is given in (3.9):

$$\alpha^m = f(\sum_{l=1}^{n}(x_l w_l^m) - \theta^m) \tag{3.9}$$

where *f* is the activation function of hidden units, *n* is the number of hidden units, w_l^m is the connection weight of input node *l* to hidden node *m*, *x* is the input space and θ^m is the bias.

Then, extract the input rules called *intermediate rules* from composed decision tree in the form of (3.10). Simplify each intermediate rule by removing useless literals and eliminate overlapping rules.

IF hiddenm ≤ b1 and hiddeni > b2 then targetclass (3.10)

For each remaining intermediate rules, generate functions which covers boundaries of hidden nodes. For the rule given in (3.10), two functions will be generated. One of these functions is generated as what is the condition that activation values of *m*-th hidden unit greater than or equal to b1.

- Step3: Build a new decision tree for each intermediate function that produces our target class. Each decision tree should correspond to one function. The input variables are obtained from the original input space whose activation values meet the function's conditions and output is their discrete target value produced by neural network. After composed of decision tree for each condition, extract the final rules from each of them.

- Step4: If necessary, simplify and eliminate redundant rules. The final rule set is rule base which describes the relationships between input variables and target.

In this work, in order to develop classification rule-base for offline risk classification, CRED is used for rule extraction from trained three-layer perceptron. Weka is used for training three-layer perceptron, obtaining decision tree and corresponding rules in Step2 and Step3.

3.3. Probability of Default Estimation

Probability of default estimation is part of Basel 2 regulations. PD indicates the probability of a credit applicant to default. In credit risk classification, there are only two decision levels: good or bad. During credit application approval, it would be impossible to reject all applicants who are classified as bad. Financial firms can take risk under a certain level that changes from internal financial regulation of countries to internal regulations of financial firms. Furthermore, good customers who are very close to bad/good customer separating conditions should handle more risky than customers who are far from the separating conditions. On the other hand, bad customers who are closer to separating conditions should be accepted as less risky than other bad customers. For these reasons, in real financial sector, customer classification would not be feasible. Thus, probability of default is applied to estimate how risky a credit applicant is. If the probability of default for each customer is higher than a predefined threshold value (changing according to internal regulations of financial firms), those applicants would not be granted, otherwise, financial firms prefer to take the risk of granting those customers.

In literature, there are different techniques for PD estimation in the range of statistical methods to data mining methods. Logistic regression is widely used for PD estimation and directly produces PD from output. In this work, we use boosted logistic regression to estimate how risky the new customer is [47]. However, the PD value which is produced by logistic regression can not be used directly as real PD value, because it needs calibration according to the actual portfolio default average [48]. The portfolio default average is determined in years while economical changes and market fluctuations should be considered. While we work on this research, we are only provided little unbalanced part of the original sample space, thus, we only assume portfolio default average according to country ratings.

3.3.1. Boosted Logistic Regression

Logistic Regression is a well known statistical method used for estimating the probability of occurrence. Basically, it fits the data to logistic curve and produces the target value in the range of [0, 1], thus the target value can be interpreted as actual probability. Boosted logistic regression, also called as LogitBoost, uses Newton steps for fitting a logistic curve by maximum binomial likelihood. It is also used for classification; the sample is assigned into the class with the maximum probability. The pseudo code for binary class problem is given in the Figure 3.4.

```
Function LogitBoost
Input: $S = \{(x_1,y_1), (x_2,y_2),\ldots,( x_N,y_N)\}$ where $x_i$ Є X and $y_i$ Є Y, $Y = \{0,1\}$
Initialize the weights:
        $w_i = 1/ N, i = 1,2,\ldots,N$
        $F(x) = 0$
        $P(x) = P(y=1|x) = 1/2$
Repeat $t = 1,\ldots,T$
        Weight update :
                $w_i = P(x_i)[1 - P(x_i)]$
                $z_i = [y_i^* - P(x_i)]/ w_i$ ,  $y_i^* = (y_i+1)/2$
        Fit the function $f_t(x)$ by a weighted least squares regression of $z_i$ to $x_i$
        using weight $w_i$
        $F(x) = F(x) + f_t(x)/2$
        $P(x) = e^{F(x)} / (e^{F(x)}+ e^{-F(x)})$
The final output: If $F(x) <= 0.5$ then 0 else 1
```

Figure 3.4. Boosted logistic regression

In this work, we use boosted logistic regression for estimating the default probability which corresponds to logarithm of bad customers to good customers. We apply boosted logistic regression on Weka which is called SimpleLogistic.

3.3.2. PD Calibration

PD Calculation phase produces uncalibrated default probability values in actual. Real PD values are obtained after scaling uncalibrated default probability according to the actual portfolio default average [48]. As stated in the previous section, uncalibrated PD value corresponds to logarithm of the ratio of bad customers to the good customers *(DR_unc)*. The calibration process has two steps, *default ratio calibration* and *PD calculation*. In the first step, the main aim is to calibrate default ratios according to formula given in (3.11) where *DR_cal* is calibrated default ratio, *DA_port* corresponds to portfolio default average and *DA_sample* is the sample space default average which is calculated from DR_unc:

$$DR\_cal = DR\_unc - (DA\_port / DA\_sample) \tag{3.11}$$

The second step aims to produce calibrated PD values according to formula below, where *PD_cal* indicates calibrated PD value:

$$PD\_cal = DR\_cal / (1 + DR\_cal) \tag{3.12}$$

In this work, we process PD calibration on the uncalibrated logistic regression results on Matlab.

3.4. Scorecard Development

Probability of default indicates the risk of customer to default. It ranges from zero to one and when PD is closer to one the risk of customer to default is increasing. There should be a threshold value to discard the worst customers carrying high risk. In international case, many credit scoring firms use scorecards where customers are segmented into predefined number of risk groups. Customers who are in the same segment carry almost same risk for financial firms. In real sector, when number of customer is taken into consideration, scorecard is more feasible then using direct PD values. Thus, we use clustering algorithms to bucket customers into predefined number of segments.

In clustering literature, there are different algorithms which determine optimal number of clusters on its own. On the other hand, in real financial sector, conditions are changing according to economical fluctuation thus it would pretty much better to use predefined number of clusters which are determined by experts' experience. In this work, we use k-Means Clustering algorithm to segment SME portfolio. We use 10, 16 and 20 customer segments. 10 is the most common number of segments and used by many credit ratings companies such as Standard and Poor (S&P) [49]. S&P uses the following system:

AAA: Best credit quality - Extremely reliable with regard to financial obligations.

AA: Very good credit quality - Very reliable.

A: More susceptible to economic conditions - still good credit quality.

BBB: Lowest rating in investment grade.

BB: Caution is necessary - Best sub-investment credit quality.

B: Vulnerable to changes in economic conditions - Currently showing the ability to meet its financial obligations.

CCC: Currently vulnerable to nonpayment - Dependent on favorable economic conditions.

CC: Highly vulnerable to a payment default.

C: Close to or already bankrupt - payment on the obligation currently continued.

D: Payment default on some financial obligation has actually occurred

Figure 3.5. S&P scorecard

3.4.1. K-Means Clustering

Clustering algorithms assume that sample data contains groups and aim to find which group each sample belongs to. In k-means clustering algorithm, number of groups is predefined. K-means clustering algorithm initializes k random cluster centers randomly. Then, at each iteration, assign each instance to the closest group according to the distance between data point and center of cluster. When all instances are assigned to a single group, each cluster centers are updated and set to the mean of all instances it contains. This process continues iteratively until *total reconstruction error* minimizes. Total reconstruction error is sum of squared distance between each data points in the groups and the corresponding group centers.

In this work, in order to develop a scorecard we use k-means clustering algorithm which is implemented on Matlab. We try 10, 16 and 20 groups. The optimal number of segments are determined according to the financial policy that the firm or bank follow and the credit expert’s experience.

4. EXPERIMENTAL RESULTS

In this research, we propose a modular approach that thoroughly covers all needs of credit risk analysis. These needs are summarized in four questions. The first question (Q1) focuses on which class the customer would be assigned into. This question only needs if customer is classified as either good or bad. We answer this by choosing the best performed algorithm after comparing classification performance of three different classifiers. Those classifiers are trained on not only real-life SME input but also its reduced sets. As stated before, the dataset is unbalanced thus we apply those classifiers with 10 fold cross validation. With 10 fold cross validation, at each iteration, input is divided into 10 partitions. Nine of them are used for training and remaining samples are used for validation. It is obvious that with 10 fold cross validation, when training is finished, each data samples is used nine times as training sample and one time as validation sample.

The second question (Q2) tries to reveal the underlying decision criterions of the model that indicates how the model arrived at the final decision. We answer this question according to the rule extraction algorithms after the best classifier we have obtained in the previous step is trained on the optimal input.

The third one (Q3) aims to estimate the probability of customer to default. For PD estimation purpose, we use a MLP cascaded model which is followed by boosted logistic regression. Although we obtain PD values from the proposed model, a calibration phase is needed to scale the PD values according to the default average of the real portfolio.

The last question (Q4) aims to segment the customer into risk groups according to the default risk. This is also called as scorecard. Scorecard is very feasible especially for large loan portfolio.

As stated earlier, we use real-life dataset which is a small subset of real database which were collected directly from SME credit applicants according to legal information they provided. When gaining subset of real database, we first use the advantage of experience of credit analyst. However, the subset we obtained has 27 features; it is still possible to contain some non-informative variables. For this reason, previous to model development phase, we prefer to apply dimension-reduction techniques to reduce our input dimension and discard non-informative variables. We apply different dimension reduction techniques on our dataset and compare their results.

4.1. Performance Metrics

We use different performance metrics in this study. In order to show the classification results, we use confusion matrix. An example confusion matrix is shown on Table 4.1. For our study, the entries in the confusion matrix have the following meaning:

- True positive (TP): The number of correct predictions that a customer is good
- False negative (FN): The number of incorrect predictions that a customer is bad. (The customer who is good in actual is predicted as bad customer.)
- False positive (FP): The number of incorrect predictions that a customer is good. (The customer who is bad in actual is predicted as good customer.)
- True negative (TN): The number of correct predictions that customer is bad.

Table 4.1. Confusion matrix

	Predicted	
Actual	**Good**	**Bad**
Good	TP	FN
Bad	FP	TN

For performance comparison, we prefer to use accuracy (Acc). Accuracy is the proportion of the number of correct predictions to total number of predictions. Acc ranges from zero to one. If Acc is closer to one, the performance of classifier increases.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{4.1}$$

In the first module, which is classification phase, in addition to accuracy, we prefer to use *Mathews Correlation Coefficient (Mcc).* Mcc is used to indicate the quality of classifier for binary class problem especially when two classes are of very different sizes [50]. Mcc ranges between [-1, 1]. When Mcc close to minus one, this corresponds to inverse classification, zero corresponds to average classification performance and one represents perfect classification.

$$Mcc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.2}$$

The third measure we use to reveal classifier performance is *Misclassification rate* (*ms_rate*), which is the proportion of misclassified instance to the total number of classified instances. Ms_rate formula is given below:

$$ms\_rate = \frac{FP + FN}{TP + FN + FP + TP} \tag{4.3}$$

When comparing the results, another performance metric we use is expert’s view. In real market, the aim is increasing the credit amount when decreasing the risk. From this point of view, the ratio of good-classified customers who are in actual bad customers becomes very important. When we accept them as good, we keep their risk. For this reason, we should take the ratio of false positives as comparison metrics. On the other hand, the ratio of bad-classified customers is also very important who are in actual good customers. When we accept them as bad, we lose those customers and the source those would create. Thus, we use *fp_rate* and *fn_rate*. fp_rate is false positives rate which indicates the proportion of bad customers who classified as good to total number of bad customers, the formula is given in (4.4).

$$fp\_rate = \frac{FP}{FP + TN} \tag{4.4}$$

fn_rate is false negatives’ rate which indicates the proportion of number of good customers who classified as bad to total number of good customers, as given in (4.5).

$$fn\_rate = \frac{FN}{FN + TP} \qquad (4.5)$$

The second module in the proposed approach is knowledge extraction. In that step, we aim to reveal the underlying decision criterions of the classifier. Performance measure to understand how applicable the rule set we obtain are defined as *precision*. Precision corresponds to accuracy in classification, which is the proportion of truly classified instances to total number of instances that meets rule’s conditions. When we consider a rule’s performance, we aim to maximize precision to obtain most accurate rule-base.

4.2. Experimental Results for Dimensionality Reduction

Dimensionality Reduction phase is a pre-processing step of the proposed method. Our dataset is composed of 27 attributes; six of them are categorical and others are continues. In data mining techniques, time and space complexity of the model is dependent on the input data size. Thus, we use dimension reduction techniques to reduce our input size without losing accuracy.

There are two different approaches for dimension reduction. Feature selection composes a subset of original dataset by choosing most informative variables. Feature extraction maps the original dataset into lower dimension space while remaining the carrying knowledge as mentioned in detail earlier.

In this work, we use different techniques from both approaches and compared their results on different classifiers. Then we choose the dimension reduction techniques best fitting to our dataset which remains the knowledge mostly.

4.2.1. Experimental Results for Feature Selection

In this work, we try two different well-known feature selection methods: decision tree and recursive feature elimination with support vector machine. Both them aim to select the most informative variables.

4.2.1.1. Decision Tree. We apply C4.5 decision tree called as J48 on Weka with 10 fold cross validation. When reached the best split, the result tree is composed of 15 variables and 28 leaves. These variables are: A2, A6, A7, A8, A9, A10, A11, A12, A13, A14, A16, A19, A23, A25, A26. The reduced dataset is called as *Input1* in the following part of this work.

4.2.1.2. Recursive Feature Elimination with Support Vector Machine. We apply SVM-RFE methods on the original dataset with Weka SVMAttrbuteEvaluator algorithm. It gives us two choices to obtain optimal subset: to select predefined number of features or to select features whose rank are greater than predefined threshold value. We use two predefined number of variables as 13 and seven. As a result, we obtain two subsets. When we define the number of dimension as 13, *Input2* is obtained which contains A5, A6, A12, A14, A15, A16, A18, A19, A20, A21, A22, A23, A24 variables. When we define the number of dimension as seven, *Input3* is obtained with A6, A18, A19, A20, A21, A23, A24 variables.

4.2.2. Experimental Results for Feature Extraction

In feature extraction phase, we try two different algorithms on full dataset: principal component analysis which linearly transform the original dataset into low dimensional feature space and factor analysis which aims to reveal underlying factors of original dataset.

4.2.2.1. Principal Component Analysis. In this study, we apply PCA on original dataset where proportion of variance is defined as 0.90. PCA transforms the original dataset into one dimensional input space which we mention as *Input4* since now.

4.2.2.2. Factor Analysis. Factor analysis aims to reveal underlying information of the given dataset. In this work, we use SAS Enterprise Guide to apply factor analysis on the dataset. We decide the number of factors according to Kaiser criterion and choose factors whose eigen values are greater than one. The eigen values of factors are given in Table 4.2 where only nine factors meet the criterion.

Table 4.2. Eigen values of correlation matrix

Dimension	Eigenvalue
1	5,4990
2	4,4049
3	2,5200
4	2,0029
5	1,5347
6	1,3629
7	1,3031
8	1,1320
9	1,0416
10	0,9400
11	0,9000
12	0,7900
13	0,7226
14	0,6670
15	0,5442
16	0,4388
17	0,3568
18	0,3111
19	0,1696
20	0,1267
21	0,0779
22	0,0641
23	0,0486
24	0,0357
25	0,0019
26	0,0014
27	0,0011

In order to determine factors, we choose highly correlated variables. High correlation between variables indicates that they are strongly related through factors. When we define 0.5 as correlation threshold value, the variables and related factors we obtain are given in Table 4.3. From the results we see that some of the variables are not related to any of the factors. These results indicate that those variables are not related to other variables among any of the factors.

Table 4.3. Variables and related factors

Variables	Factors
A1	F6
A2	
A3	
A4	F1
A5	F1
A6	F1
A7	F6
A8	
A9	F8
A10	F3
A11	F3
A12	F3
A13	F1 + F5
A14	F1
A15	F1
A16	F1
A17	F7
A18	F1 + F5
A19	F9
A20	F2
A21	F2
A22	F2
A23	F2
A24	F2
A25	F2
A26	F3 + F4
A27	F3 + F4

The nine factors and their descriptions are given in Figure 4.1.

Factor1 : Combination of customer existing risk information
Factor2 : Combination of customer delinquency information
Factor3 : Combination of customer historical information
Factor4 : Combination of customer historical delinquency information
Factor5 : Combination of customer credit and corresponding guarantee information
Factor6 : Combination of customer demographic information
Factor7 : Combination of customer financial information
Factor8 : Specific customer demographic information
Factor9 : Maximum delinquency information

Figure 4.1. Factors extracted from the original dataset

From FA, we obtain a reduced input set which is composed of nine features by projection of factor loadings V and their estimated covariance matrix. We calculate factor loadings and estimated covariance matrix by Matlab on observable variables. The nine dimensional input set is obtained by FA and called as *Input5* in the following subsections.

Furthermore, we apply FA on Input2 for only releasing the underlying factors. We obtain only four factors which are given below:

Factor1: Combination of customer existing risk information
Factor2: Combination of customer delinquency information
Factor3: Maximum delinquency information
Factor4: Combination of customer financial information

Figure 4.2. Factors extracted from Input2

From the results we see that factors extracted from Input2 are a subset of the factors extracted from original dataset. This result also indicates that Input2 almost carries the information behind the original dataset.

4.3. Experimental Results for Credit Risk Classification

In credit risk classification phase, we aim to determine the best classifier for our modular approach. Thus, we compare performance of three different classifiers; k-NN, MLP and SVM on six input we have composed in the previous subsection. For comparison, we use all performance measures as stated in subsection 4.1. While choosing the optimal classifier, each performance measures have to be considered carefully. Because in credit risk analysis domain, maximizing the number of credit applicant which corresponds to minimizing the false negatives in confusion matrix is as important as minimizing the total unobserved risk which corresponds to minimizing the false positives. In the following subsections, classification results are given in detail for each classifier on six inputs.

4.3.1. k-Nearest Neighbor

kNN is a nonparametric unsupervised classifier which gives high classification performance either linear or nonlinear data space. In this work, we apply kNN on all our six inputs, one is the original dataset and other five are reduced versions. We use Weka *IBK* algorithm for kNN where we set k = 5. We prefer to use 10 fold cross validation and guarantee that model is validated by unseen samples. Results are given in Table 4.4.

From the results, we see that kNN provides feasible classification performance. According to the results, it is obvious that original dataset includes some non-informative features. Input spaces which are obtained from feature selection methods provide significantly better classification performance. On the other hand, kNN failed to classify input spaces which are extracted by feature extraction methods. Between the two feature selection methods, we see that SVM-RFE produces the most informative results for k-NN where both accuracy and Mathew’s correlation coefficient reach highest score among all inputs. Furthermore, as we mention before, false classification rate is another important measure for us, we aim to reach minimum fp_rate with minimum fn_rate. Input2 minimizes both fp_rate and fn_rate. On the other hand, we see that, feature extraction methods have negative effect on classifier performance for our dataset, because kNN worst performs on Input4 which is reduced by PCA and Input5 which is reduced by factor analysis. Those algorithms use linear transformations while reducing the input space.

Table 4.4. Classification results for kNN

Dataset	Performance Measures				
	fn rate	fp rate	ms rate	Acc	Mcc
Original Dataset	0,6805	0,0898	0,2559	0,7441	0,2860
Input1	0,6528	0,1033	0,2578	0,7422	0,2907
Input2	0,4444	0,0951	0,1934	0,8066	0,4961
Input3	0,4792	0,1141	0,2168	0,7832	0,4355
Input4	0,8472	0,1494	0,3457	0,6543	0,0042
Input5	0,8333	0,1739	0,3594	0,6406	-0,0086

From k-NN point of view, it is obvious that, SVM-RFE produces the most informative subset of original dataset which significantly outperform other five inputs, including the original one.

4.3.2. Multilayer Perceptron

MLP is a non-parametric supervised classifier which provides high performance when acquiring hidden knowledge. In this research, we apply three-layer neural network on six different inputs. We use sigmoid as activation function in hidden units which produce output value in the range of [0, 1]. We discretize the output as good or bad. If predicted output is equal to or less than 0.5, the sample is classified as good, otherwise it is classified as bad. We train MLP with 10 fold cross validation. Thus, we guarantee that each instance is used for both training and validation at different iteration. When we compose hidden layer, we prefer to use five hidden units. The results of MLP on six different input spaces are given in Table 4.5.

For performance comparison, we apply probabilistic neural network on our input space. It reaches high classification accuracy for good customers on the other hand it fails to classify bad customers. As a result of bad customer classification failure, we do not use PNN results for performance comparison and do not mention its results.

From the results, we see that MLP provides high classification performance. We see that MLP worst performs on inputs which are reduced by feature extraction methods. All performance measures reached the worst value for these two input sets. In addition to this, we obtain worse results for Input3 which is also reduced by SVM-RFE like Input2 but with fewer dimensions. As a result, from MLP point of view, we can say that MLP produces slightly better results for both original dataset and its subsets, especially on Input1 where both fp_rate and fn_rate are minimum and Mcc is maximum among all.

Table 4.5. MLP results for all inputs

Dataset	**Performance Measures**				
	fn_rate	**fp_rate**	**ms_rate**	**Acc**	**Mcc**
Original Dataset	0,527778	0,13587	0,246094	0,7539	0,3591
Input1	0,451389	0,154891	0,238281	0,7617	0,4008
Input 2	0,548611	0,125	0,244141	0,7559	0,3561
Input 3	0,784722	0,029891	0,242188	0,7578	0,3037
Input 4	1	0	0,28125	0,7188	-
Input 5	1	0	0,28125	0,7188	-

4.3.3. Support Vector Machine

SVM is well known discriminant analysis methods. We train SVM with polynomial kernel and radial basis kernel however RBF kernel failed to classify any of the input we use. Thus, we do not mention RBF kernel results here. SVM with polynomial kernel is trained with 10 fold cross validation and we classify customer as good if predicted output is equal to or less than zero, as bad customer elsewhere. Classification result for SVM with polynomial kernel is given in Table 4.6.

Table 4.6. SVM results for all input sets

Dataset	Performance Measures				
	fn_rate	fp_rate	ms_rate	Acc	Mcc
Original Dataset	0,8403	0,0136	0,2461	0,7539	0,2890
Input 1	0,8403	0,0109	0,2441	0,7559	0,2994
Input 2	0,8403	0,0082	0,2422	0,7578	0,3104
Input 3	0,8403	0,0082	0,2422	0,7578	0,3104
Input 4	1	0	0,2813	0,7188	-
Input 5	0,9931	0	0,2793	0,7207	-

SVM produces high classification performance on dataset and its subsets but fail for extracted datasets. Especially for Input2 and Input3 which are subsets of original dataset obtained by SVM-RFE, classification results show significant increase, both accuracy and Mcc are maximized. On the other hand, when we consider fp and fn rates, we see that SVM provides great success to eliminate bad customer, however fail to eliminate good customer.

4.3.4. Classifier Evaluation

In dimensionality reduction phase of this work, we apply the different classifiers: k-NN, MLP and SVM. While we determine the best techniques to use as main classifier in the proposed method, we should consider each performance measures. Mcc carries as much importance as accuracy because of the unbalanced distribution of good/bad class probabilities. Furthermore, from expert point of view, we mostly consider to minimize fp_rate at the same time with minimum fn_rate. The most risky situation in credit risk analysis is high fp_rate where the company faces to unobserved risks. As a result, we take fp_rate and fn_rate into account when choosing the optimal classifier. Other decision criterion for us while choosing the classifier is applicability. Applicability means that, we aim to choose the optimal classifier which remains the same or better performance for all inputs derived from original datasets. The summary of classifier performance for all input is given in Table 4.7 together where better performance for each are shown in bold.

From the results, it is obvious that none of the classifier can be selected as the best as none of them producing the best performance on all inputs. We see that, on the original dataset and its subset Input1, MLP provides highest classification accuracy with significant difference from other the classifiers. For Input3 and Input4, kNN reaches the highest performance among all classifiers on all datasets where MLP produces closer results. For Input4 and Input5, all three classifiers fail to catch any of good customers. On the other hand, we abstain from using k-NN as main classifier because its classification method is based on neighbor’s behavior which could be colorable especially during economical fluctuations.

Table 4.7. Summary of classifier performance

Classifier	Dataset	Performance Measures			
		fn_rate	**fp_rate**	**Acc**	**Mcc**
k-NN	**Original Dataset**	0,680556	0,089674	0,7441	0,286
	Input1	0,652778	0,103261	0,7422	0,2907
	Input2	0,444444	0,095109	**0,8066**	**0,4961**
	Input3	0,479167	0,11413	**0,7832**	**0,4355**
	Input4	0,847222	0,149457	0,6543	0,0042
	Input5	0,833333	0,173913	0,6406	-0,0086
MLP	**Original Dataset**	0,527778	0,13587	**0,7539**	**0,3591**
	Input1	0,451389	0,154891	**0,7617**	**0,4008**
	Input2	0,548611	0,125	**0,7559**	**0,3561**
	Input3	0,784722	0,029891	0,7578	0,3037
	Input4	1	0	0,7188	-
	Input5	1	0	0,7188	-
SVM	**Original Dataset**	0,840278	0,013587	0,7539	0,289
	Input1	0,840278	0,01087	**0,7559**	**0,2994**
	Input2	0,840278	0,008152	**0,7578**	**0,3104**
	Input3	0,840278	0,008152	**0,7578**	**0,3104**
	Input4	1	0	0,7188	-
	Input5	0,993056	0	0,7207	-

As indicated previous sections, we aim to increase number of debtor with taking minimum risks. As a result, we should take both fp_rate and fn_rate into account at the same time with Mcc. From this point of view, we prefer to use MLP as classifier in our work, because it is obvious that, MLP provides good classification performance thorough six inputs we use. In the following subsections, we accept MLP as main classifier and the other parts of our proposed method is built on MLP. Furthermore, Input2 provides better performance for SVM, k-NN and almost same results with original dataset for MLP. Also it provides minimum false-positive rate which minimizes the unobserved risk we meet. Input2 is a subset of original dataset and obtained by SVM-RFE including 13 features. We use Input2 as sample space for the following parts of our proposed approach. The data plot for the highest rated three dimensions of Input2 is given in the Figure 4.3. In the data plot x axis called Dimension1 is the highest rated dimensions and z axis called Dimension3 is the third highest rated dimension of the Input2. In the figure, stars indicate the good customers and unfilled circles correspond to bad customers.

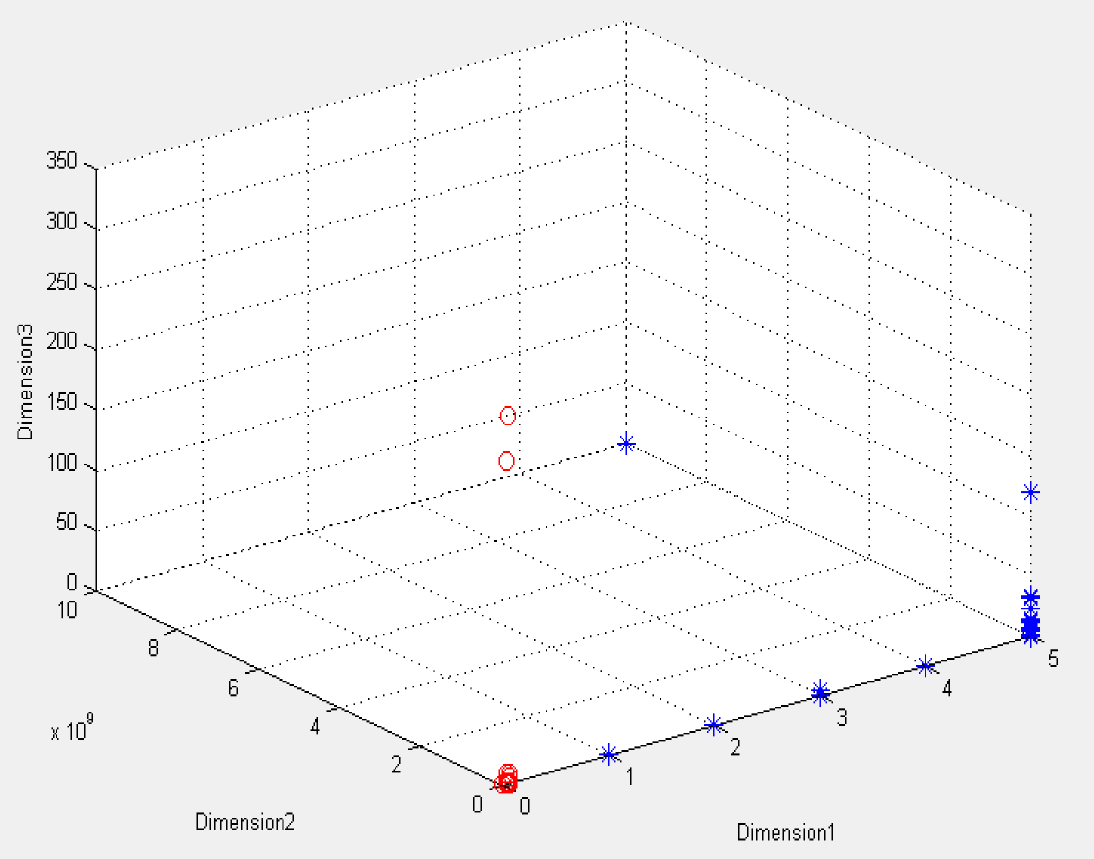


Figure 4.3. Data distribution for the first three highest rated dimensions of Input2

4.4. Experimental Results for Rule-Base Extraction

The first module in our proposed method is applied for classification which determines if customer will be granted or not. The second module reveals how the classifier reached at the final decision. In credit risk analysis, especially in high volume credits, it is always possible to include expert's view at the decision phase according to the economical changes, general fluctuation in the real market or under market specific conditions. Thus, knowledge-extraction becomes very important in real applications to include expert view for best decisions.

In this work, we choose MLP as main classifier and compose our model based on three-layer MLP. Thus, we reveal the decision-making criterions for good customers from trained neural network which determines final classification of credit applicants. For rule-extraction purpose, we apply CRED as neural rule extractor on the trained MLP.

4.4.1. Neural Rule Extraction

Neural rule extraction methods aim to extract the criterions from trained neural network. In literature, three-layer neural networks are used for rule-extraction. Rule-extraction from MLP mechanism consists of four steps. First step is MLP training. Our proposed model composed of three layers MLP with five hidden units inside where hidden units use sigmoid as activation function.

In the second step, we develop a decision tree called hidden-output tree. Input variables of this tree are activation values of hidden units. We have five hidden units in MLP thus, input consists of five dimensions each correspond to hidden unit activation values and the output variable is target value produced by neural network which is one if customer is classified as good, else zero. We run Weka J48 algorithm, which is corresponds to C4.5 decision tree, on Input2. From the result tree, we obtain three rules where hidden_rule1 and hidden_rule3 produce bad class and hidden_rule2 produces good class. Hidden rules are given in Figure 4.4, where nodes are indicated by *hiddeni*, *i* = 1,2,…,5, do not corresponds to real dataset features, only they indicate activation value of *i*-th hidden node.

HIDDEN_RULE1: IF hidden1 <= 0.99986 and hidden4 <= 0.62039 THEN bad
HIDDEN_RULE2: IF hidden1 <= 0.99986 and hidden4 > 0.62039 THEN good
HIDDEN_RULE3: IF hidden1 > 0.99986 THEN bad

Figure 4.4. Results of hidden-output tree

In this step, we also generate functions from the rules according to their target class. As stated in the previous sections, we aim to determine if the credit applicant will pay the loan back or not, as a reason, we choose good as our target class. According to the target class, we continue with hidden_rule2 and generate functions for each boundary. Generated functions for good customers are given in Figure 4.5.

Function1 : What is the condition that the activation values of hidden1 less than or equal to 0.99986
Function2 : What is the condition that activation values for hidden4 greater than 0.62039

Figure 4.5. Functions generated from hidden-rule2

In the third step, we recompose input-hidden decision trees for each functions created in step2. Each input-hidden tree first eliminates the instances which they meet function's conditions and target class. Then those instances are accepted as input and their corresponding discrete output variables as output.

For Function1, the instances whose first hidden node activation value is less than or equal to 0.99986 are accepted with their 13 dimensions as inputs and their predicted class labels are accepted as target value. This sample space is called as *function1_dataset* since now. Then, we apply C4.5 decision tree on function1_dataset, we obtain new eight rules that are given in Figure 4.6.

RULE1 : IF A2 ≤ 4 THEN good
RULE2 : IF A2 > 4 and A12 ≤ 2 THEN bad
RULE3 : IF A2 > 4 and A12 > 2 and A7 ≤ 80000 and A4 ≤ 1223 THEN bad
RULE4 : IF A2 > 4 and A12 > 2 and A7 ≤ 80000 and A4 > 1223 and A9 ≤ 1 and a6 ≤ 0 THEN bad
RULE5 : IF A2 > 4 and A12 > 2 and A7 ≤ 80000 and A4 > 1223 and A9 ≤ 1 and a6 > 0 THEN good
RULE6 : IF A2 > 4 and A12 > 2 and A7 ≤ 80000 and A4 > 1223 and A9 > 1 THEN good
RULE7 : IF A2 > 4 and A12 > 2 and A7 > 80000 and A8 ≤ 28 THEN good
RULE8 : IF A2 > 4 and A12 > 2 and A7 > 80000 and A8 > 28 THEN bad

Figure 4.6. Rule set extracted for Function1

In each rule, *Ai* corresponds the *i*-th dimension of Input2. We discard rule2, rule3, rule4 and rule8 unfortunately they do not produce our target class and continue with rule1, rule5, rule 6 and rule7 in the following steps.

For the second function, the instances whose fourth hidden node activation value is greater than 0.62039 are accepted with their 13 dimensions as inputs and their predicted class labels are accepted as target value. This input space is called as *function2_dataset* in the following part. We apply C4.5 decision tree which is called J48 in Weka. J48 extract five rules, shown in Figure 4.7. Only rule9 and rule11 meet our target class condition. Thus, when rule-base is developed for good customers, only these two rules are considered.

RULE9 : IF A2 ≤ 4 THEN good
RULE10 : IF A2 > 4 and A3 ≤ 9320 and A6 ≤ 0 THEN bad
RULE11 : IF A2 > 4 and A3 ≤ 9320 and A6 > 0 and A10 ≤ 1 THEN good
RULE12 : IF A2 > 4 and A3 ≤ 9320 and A6 > 0 and A10 > 1 THEN bad
RULE13 : IF A2 > 4 and A3 > 9320 THEN bad

Figure 4.7. Rule set extracted for Hidden_Rule2

We aim to develop a target-rule base, revealing how the neural network decide if a customer is good credit applicant. In the fourth step, we will combine rules with target class which are generated in the third step. We pick rules which produce the target class among all. However, in both Figure 4.6 and Figure 4.7, Rule1 and Rule9 are totally the same. We pick only one from these two. As we use C4.5 decision tree, we do not need to simplify generated rules because C4.5 composes pruned decision tree. Thus, we only need to find overlapping or redundant rules because these rules are extracted from different subsets of Input2. As a result, it is possible that some rules overlap other rules. As we have only two functions in the second step, our produced rules do not overlap each other and allow us to use them directly as the rule base. Our final rule-base is shown in Figure 4.8.

RULE1 : IF A2 ≤ 4 THEN good
RULE2 : IF A2 > 4 and A12 > 2 and A7 ≤ 80000 and A4 > 1223 and A9 ≤ 1 and A6 > 0 THEN good
RULE3 : IF A2 > 4 and A12 > 2 and A7 ≤ 80000 and A4 > 1223 and A9 > 1 THEN good
RULE4 : IF A2 > 4 and A12 > 2 and A7 > 80000 and A8 ≤ 28 THEN good
RULE5 : IF A2 > 4 and A3 ≤ 9320 and A6 > 0 and A10 ≤ 1 THEN good

Figure 4.8. Rules extracted from trained MLP on Input2

When we apply these five rules on Input2, the performance results obtained are given in detail in the table below. As performance measure, we use precision which corresponds to accuracy. We aim to generate most accurate rules. As seen in Table 4.8, three of five generated rules give good prediction performance, where other two produces average accuracy. These results indicate, the target rule-base we developed mostly reveal the decision criterion of the classifier we use.

Table 4.8. Rule-base performance results

Rules	Precision
Rule1	0.96
Rule2	0.5641
Rule3	1
Rule4	0.6023
Rule5	0.3863

4.5. Experimental Results for Probability of Default Estimation

PD estimation is the third module of our proposed approach. PD indicates the risk of credit applicant to willing to pay the loan back. It ranges between zero and one, where PD increases, risk increases. In this work, we apply boosted logistic regression on Input2 to calculate PD for each customer. In theory, the threshold value to determine bad customer is 0.5. Customers who have PD value above threshold are accepted as bad customer. However, in real sector applications, threshold value is determined by credit experts under different conditions such as economical parameters and company internal specifications. In this work, we evaluate boosted logistic regression result according to theoretical approach where threshold is 0.5. In the calibration phase, we scale the uncalibrated PD values, produced by logistic regression according to the portfolio default average then actual PD values are obtained.

4.5.1. Boosted Logistic Regression

Logistic regression is well-known PD estimation technique based discriminant analysis. In this work, we use boosted logistic regression on Weka where as input we use MLP results of Input2. As stated earlier, MLP produces value between zero to one and if output is greater than 0.5, we accept that applicant as bad customer. Thus, we use MLP results as input for LR and produce PD from those outputs. In Table 4.9, the results of logistic regression on MLP results are given above the results of logistic regression applied directly dataset2.

Table 4.9. Logistic Regression performance evaluation

Dataset	**Performance Measures**				
	fn_rate	**fp_rate**	**ms_rate**	**Acc**	**Mcc**
MLP – LR	0,25	0,2361	0,2461	0,7539	0,3232
LR	0,3913	0,07639	0,3027	0,7558	0,2956

According to the results it is obvious that logistic regression produces exciting performance on MLP results for Input2. When we apply logistic regression directly on

Input2, discretized probability of default values obtained by logistic regression fail to classify good customers, on the other hand, when we apply logistic regression on the MLP results for the same dataset, the classification performance has great increase both for good and bad customers.

4.5.2. PD Calibration

In this phase, we calibrate the PD values according to the actual portfolio average. However, we do not have actual portfolio average but as Turkey credit score is given as B+, we assume portfolio default average as five per cent. We obtain sample data default average directly and calibrate the PD values according to assumed portfolio average. These calibrated PD values are used for scorecard development in the following subsection.

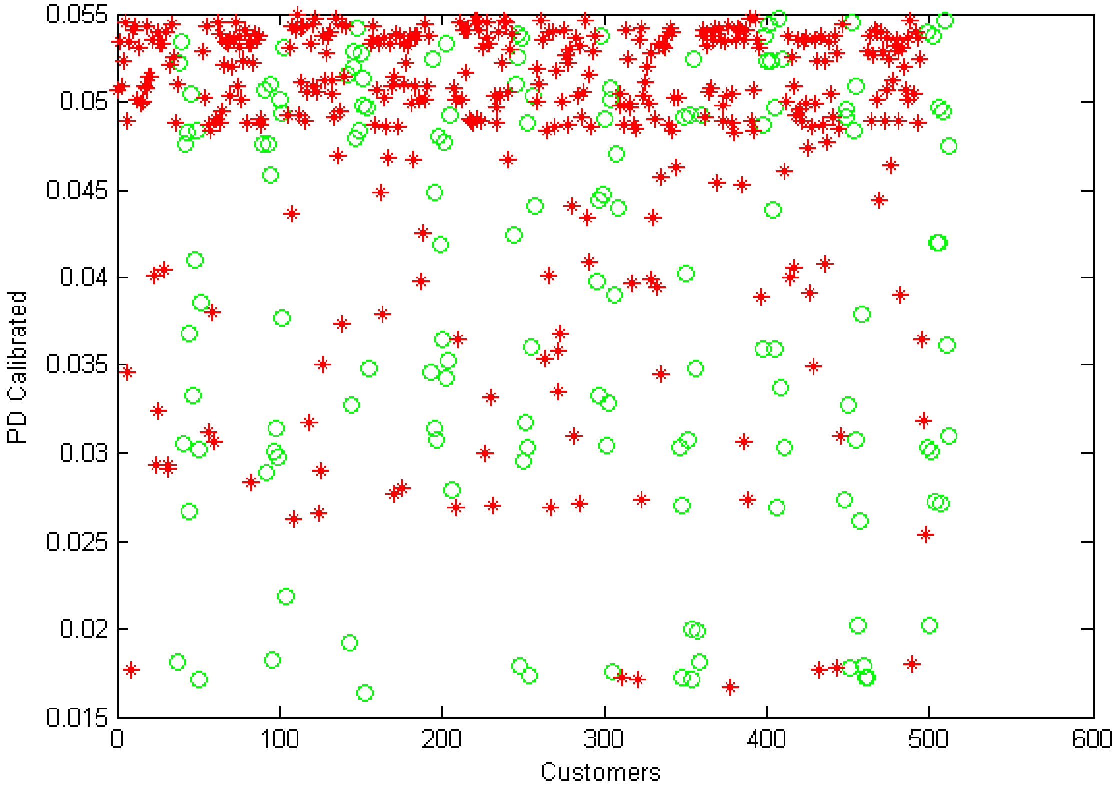


Figure 4.9. Calibrated PD distribution for Input2

Calibrated PD distribution of Input2 is given above where x dimension corresponds to customers and y dimension to calibrated PD values. Unfilled circles indicate good customers however stars indicate bad customers. As seen, good customers and bad customers have a significant separation between each.

4.6. Experimental Results for Scorecard

Scorecard development is the last module of the proposed model. In his phase, we aim to segment calibrated PD values obtained in the previous subsection. First of all, we apply k-means clustering algorithm on calibrated PD values of our customer portfolio, then we evaluate our results comparing with average SME score against country score.

4.6.1. k-Means Clustering

We implement k-means algorithm on Matlab and apply on calibrated PD values of our portfolio with different k values. As stated earlier, S&P use 10 buckets to segment credit risks, in the real market, 16 and 20 are also preferred. Because when number of segments decrease, risk score ranges become flexible, thus, 16 and 20 are average number used in real market. In this work, we try 10, 16 and 20 segments to group the customer. Results are given in Table 4.10 in detail for *Scorecard1* with 10 segments like S&P. The first row indicates the segment label, second one indicates the mean of the calibrated PD values of customers in each segment which is also corresponds to the average default risk of the segment. The last row shows how many customers are segmented into each group. Also, number of customer distribution is available on Figure 4.10.

Table 4.10. Scorecard1: PD values for 10 customer segments

Segment	**Seg1**	**Seg2**	**Seg3**	**Seg4**	**Seg5**	**Seg6**	**Seg7**	**Seg8**	**Seg9**	**Seg10**
Avg Risk	0,15	1,87	2,7	3,48	4,2	4,8	5,12	5,5	5,76	5,94
Num of Customer	30	50	32	24	23	80	81	48	98	46

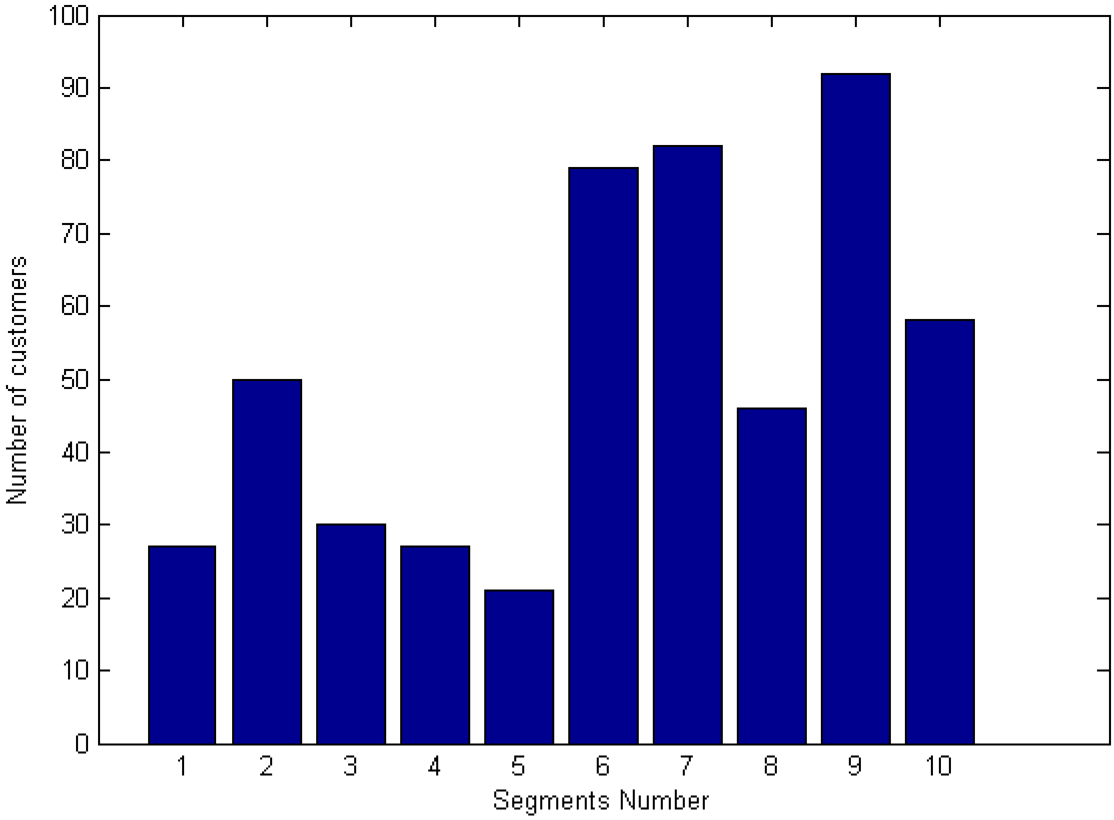


Figure 4.10. Segment-customer distribution for 10 segments

When we use 16 segments, that is called as *Scorecard2* since now, the results are given in table 4.11 in detail. As seen from the results, when number of segments increases, more detailed score-card development is possible.

Table 4.11. Scorecard2: PD values for 16 segments

Segment	**Seg1**	**Seg2**	**Seg3**	**Seg4**	**Seg5**	**Seg6**	**Seg7**	**Seg8**
Avg Risk	1,75	2,02	2,72	3,08	3,51	3,94	4,35	4,67
Num of Customer	21	6	22	33	25	24	17	19
Segment	**Seg9**	**Seg10**	**Seg11**	**Seg12**	**Seg13**	**Seg14**	**Seg15**	**Seg16**
Avg Risk	4,88	5	5,07	5,15	5,23	5,31	5,37	5,44
Num of Customer	64	45	37	13	28	38	66	54

At last, we segment our customer portfolio into 20 segments and called *Scorecard3* to visualize customer risk behavior in detail. As seen in Table 4.12, detailed segmentation give us chance to make more accurate decision when granting customer. This becomes more important especially high growth of SME customer volume in the real market.

Table 4.12. Scorecard3: PD values for 20 segments

Segment	**Seg1**	**Seg2**	**Seg3**	**Seg4**	**Seg5**	**Seg6**	**Seg7**	**Seg8**	**Seg9**	**Seg10**
Avg Risk	1,81	2,91	3,47	3,94	4,35	4,67	4,88	5	5,07	5,15
Num of Customer	27	50	30	24	17	19	64	45	37	13

Segment	**Seg11**	**Seg12**	**Seg13**	**Seg14**	**Seg15**	**Seg16**	**Seg17**	**Seg18**	**Seg19**	**Seg20**
Avg Risk	5,23	5,28	5,32	5,34	5,36	5,37	5,39	5,41	5,45	5,47
Num of Customer	26	14	21	22	9	24	16	25	21	8

5. CONCLUSION

In this research, we focus on credit risk analysis which becomes very important in real financial market in the recent years. Also credit risk analysis will become obligatory for credit firms in the near future as BDDK declared. Thus, we propose a modular comprehensive credit risk analysis method for Turkish SME customer portfolio, which covers each credit risk analysis need; customer classification, rule-base extraction, PD estimation and scorecard development.

As a preprocess for our proposed method, we investigate how to reduce our real-life feature space in order to prevent from increasing time and space complexity and also discard non-informative variables. We compare performance of both feature selection algorithms to eliminate the most informative features and feature extraction algorithms to map our sample space into lower dimensional space. For this comparison, different techniques are applied on original dataset that produces a number of reduced input sets. As comparison criterion, we give these reduced datasets to three different classifiers. According to the experiments done, we can conclude that SVM-RFE, which is a feature selection methods, is the optimal method to reduce input data volume and should be used as SME credit risk analysis pre-processing method.

In customer classification aspect, we aim to find optimal classifier that classifies our customer portfolio successfully. We apply kNN, MLP and SVM on the all input sets. The optimal classifier is selected which maximizes Mcc on the other hand minimizes fp_rate and fn_rate. MLP produces good classification performance on the optimal input where both fp_rate and fn_rate optimal. Thus, both high credit debtors and low unpredicted losses criterions are almost reached.

In the second module, we investigate the underlying circumstances of the main classifier decision mechanism. For rule-extraction purpose, we use a neural rule extractor which successfully reveals that under which conditions a credit applicant is classified as good.

In probability of default estimation module, we propose a cascaded MLP model which is followed by boosted logistic regression. Logistic regression is applied on MLP results. Thus, the success of MLP to acquire the underlying information is also added to the PD estimation model. According to the experiments done, cascaded MLP-LR model outperforms classical LR model. On the other hand, the resulted PD values could not be used directly as real PD values hence a calibration phase is added into the third module with five per cent of portfolio default average assumption.

In the last module, we focus on scorecard development which segments customers into risk-related groups according to their calibrated probability of default values. Our proposed model is able to produce a scorecard which consists of a predefined number of customer segments.

When we compare our proposed model with other related studies, we see that our proposed model covers most of the credit risk analysis domain’s needs that any of other studies has covered. Also, we see that our proposed model produces better results in term of probability of default estimation and outperforms the classical model stated in other studies. On the other hand, SME behavior changes in different countries according to economical, regional, cultural and sector-specific differences. For this reason, we could not validate other model with any of other countries SME databases.

REFERENCES

1. Bank for International Settlements (BIS), http://www.bis.org.

2. Banking Regulation and Supervision Agency (BDDK), http://www.bddk.org.tr/turkce/Basel-II/Basel-II.aspx.

3. Haimowitz, I. J. and T. K. Keyes, "*Handbook of Datamining and Knowledge Discovery*", Oxford University Press, 2002.

4. US Small Business Administration, http://www.sba.gov/services/contractingopportunities/sizestandardstopics/index.html.

5. OECD, Statistics Directorate, "*Towards better Structural Business and SME Statistics*", SBS Expert Meeting, November 2005.

6. Ankara Ticaret Odası, "*Basel II : Kobi'lerin Kredi Riski Ve Derecelendirilmesi*", 2007.

7. Türkiye Bankalar Birliği Basel II Yönlendirme Komitesi, "*Risk Yönetimi ve Basel II'nin Kobi'lere Etkileri"*, 2004.

8. Fantazzini, D., S. Figini, "*Random Survival Forests Model for SME Credit Risk Measurement*", Methodology and Computing in Applied Probability, Springer, 2008.

9. Pang, S., Y. Wang, Y. Bai, "*Credit Scoring Model Based On Neural Network*", Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 2002.

10. Yeh, I. C., C.Lien, “*The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card client*s”, Expert Systems with Applications, Volume 36, Issue 2, Part 1, March 2009, Pages 2473-2480.

11. Yang, C. G., X. B. Duan, “*Credit Risk Assessment in Commercial Banks Based on SVM using PCA*”, Proceeding of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 2008.

12. Zhou, J., T. Bai, “*Credit Risk Assessment using Rough Set Theory and GA-Based SVM*”, The Third International Conference on Grid and Pervasive Computing-Workshop, 2008.

13. Wei, L., J. Li, Z. Chen, “*Credit Risk Evaluation Using Support Vector Machine and Mixture of Kernel*”, Lecture Notes in Computer Science, Computational Science – ICCS 2007 .

14. Galindo, J., P. Tamayo, “*Credit Risk Assessment Using Statistical and Machine Learning: Basic Methodology and Risk Modeling Applic*ations”, Computational Economics 15:107-143, 2000.

15. German Credit Dataset,
http://archive.ics.uci.edu/datasets/Statlog+(German+Credit+Data)

16. Kaya, M. E., F. Gürgen, N. Okay, “*An Analysis of Support Vector Machines for Credit Risk Modeling*”, PAKDD 2007 DMBiz Workshop, China, IOS Press, 2008.

17. Gaganis, C., F. Pasiouras, C. Spathis, C. Zopounidis, “*A Comparison of Nearest Neighbors, Discriminant and Logit Models for Auditing Decisions*”, Intelligent Systems in Accounting, Finance and Management 15, 23-40, 2007

18. Altman, E. I. AND G. Sabato, “*Modeling Credit Risk for SMEs: Evidence from the US Market*”, http://pages.stern.nyu.edu/~ealtman.

19. Alpaydin, E., *Introduction to Machine Learning*, MIT Press, 2004.

20. Han, J. and M. Kamber, “*Data Mining Consepts and Techniques*”, Academic Press, 2001.

21. Huang, S., M. O. Ward and E. A. Rundensteine, “*Exploration of Dimensionality Reduction for Text Visualization”*, Proceedings of the Third International Conference on Coordinated & Multiple Views in Exploratory Visualization, 2005.

22. Ahmad, F. K., S. Deris, N. M. Norwawi, N. H. Othman, “*A Review of Feature Selection Techniques via Gene Expression Profiles*”, Information Technology, 2008.

23. Bachrach R. G., A. Navot, N. Tishby, “*Margin Based Feature Selection - Theory and Algorithms*”, Proceedings of the 21 st International Conference on Machine Learning, 2004, Canada.

24. Kojadinovic, I. and Wottka, T., “*Comparison between a filter and a wrapper approach to variable subset selection in regression problems*”, Submitted to *ESIT 2000,* Germany.

25. Huang J., Y. Cai and X. Xu, “*A Filter Approach To Feature Selection Based On Mutual Information*” Proc. 5th IEEE Int. Conf. on Cognitive Informatics (ICCI'06).

26. Berger, H., D. Merkl, M. Dittenbach, “*Exploiting Partial Decision Trees for Feature Subset Selection in e-Mail Categorization*”, Proceedings of the 2006 ACM symposium on Applied computing,2006.

27. Duan K, J. C. Rajapakse, “*Multiple SVM-RFE for Gene Selection in Cancer Classification With Expression Data*”, NanoBioscience, IEEE Transactions, 2005.

28. Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1993.

29. WEKA 3: Datamining Software in Java, http://www.cs.waikato.ac.nz/ml/weka.

30. Chen, X., and J. C. Jeong, "*Enhanced Recursive Feature Elimination*", IEEE Sixth International Conference on Machine Learning and Applications, 2007.

31. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "*Gene selection for cancer classification using support vector machines*," *Mach. Learn.*, vol.46, no. 1–3, pp. 389–422, 2002.

32. Thang, Y., Y. Zhang, Z. Huang, "*Development of Two-Stage SVM-RFEGene Selection Strategy forMicroarray Expression Data Analysis*", IEEE/ACM Transactions On Computational Biology and Bioinformatics,2007.

33. C. Furlanello, M., Serafini, S., Merler, G. Jurman, "*Gene Selection and Classification by Entropy-based Recursive Feature Elimination*", Proceedings of the International Joint Conference on Neural Networks, 2003

34. Tsai, F. S., K. L. Chan,"*Dimensionality Reduction Techniques for Data Explor*ation", Proc. IEEE 6th International Conference on Information, Communications and Signal Processing, 2007,Singapore.

35. Hiden, H.G., Willis, M.J., Tham, M.T., Turner, P., Montague, G.A., "*Non-linear principal components analysis using genetic programming", ,* Second International Conference On Genetic Algorithms in Engineering Systems: Innovations and Applications, 1997. GALESIA 97.

36. Lee, J. K., K.H. Kim, T. Y. Kim, W. H. Choi, "*Nonlinear principle component analysis using local probability***",** Proceedings KORUS 2003, The 7th Korea-Russia International Symposium on Science and Technology.

37. Takiguchi, T., Ariki, Y., "*Robust Feature Extraction using Kernel P***CA",** IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006 Proceedings.

38. The MathWorks, http://www.mathworks.com

39. Wu, N. and J. Zhang, "*Factor Analysis Based Anomaly Detection*", Proceedings of the 2003 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY June 2003.

40. Oreški, D. and P. Peharda, "*Application of Factor Analysis in Course Evaluation*", Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, June 23-26, Cavtat, Croatia.

41. SAS: Business Intelligence Software and Predictive Analytics, http://www.sas.com.

42. Weinberger, K. Q., K. S. Lawrence, "*Fast Solvers and Efficient Implementation for Distance Metric Learning*", Proceedings of the 25 th International Conference on Machine Learning, Helsinki, Finland, 2008.

43. Junli C., J. Licheng, "*Classification Mechanism of Support Vector Machines*"**,** Proceedings of 5th International Conference on Signal Processing, WCCC-ICSP, 2000.

44. Sato, M., H. Tsukimoto, "*Rule Extraction from Neural Networks via Decision Tree Induction*", Proceedings of IJCNN '01, International Joint Conference on Neural Networks, 2001.

45. Boz, O. , "*Extracting Decision Trees From Trained Neural Networks*", ACM-SIGKDD '02 July 23-26,2002, Edmonton, Alberta, Canada.

46. Craven M. W**.,** **"***Extracting Comprehensible Models From Trained Neural Networks*"**,** PhD thesis, Department of Computer Sciences, University of Wisconsin-Madison, http://www.biostat.wisc.edu/~craven.

47. Cai, Y. D., Feng, K. Y., Lu, W. C., Chou, K. C., "*Using LogitBoost Classifier to Predict Protein Structural Classes*", Journal of Theoretical Biology 238 (2006), 172–176.

48. Kaya, M. E., F. Gürgen, N. Okay, "*An Application of Support Vector Machines and Logistic Regression for Credit Risk Analysis*", International Journal of Information Technology and Intelligent Computing, Vol. 1, No. 4, 2008.

49. Standard and Poor, International Rating Company,
http://www.standardandpoors.com.

50. Lund, O., M. Nielsen, C. Lundegaard, C. Keşmir, S. Brunak, "*Immunological Bioinformatics",*The MIT Press, London, 2005.