CLASSIFIER PERFORMANCES FOR CREDIT RISK ANALYSIS

by

Erkan Çetiner

B.S., Computer Engineering, Işık University, 2008

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

Graduate Program in Computer Engineering

Boğaziçi University

2011

CLASSIFIER PERFORMANCES FOR CREDIT RISK ANALYSIS

APPROVED BY:

Prof. Fikret Gürgen …….

(Thesis Supervisor)

Prof. Taflan Gündem …….

Prof. Nesrin Okay ……..

DATE OF APPROVAL: 24.05.2011

ACKNOWLEDGEMENTS

I am heartily thankful to my supervisor, Prof. Fikret Gürgen, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. With his significant contribution , I was able to finish this thesis work .

Secondly, I would like to thank Prof. Nesrin Okay and Prof. Taflan Gündem who agreed to participate in my thesis examining committee.

I would like to present my special thanks to TÜBİTAK BİDEB, which gave me scholarship during my graduate study and encourage me to work on development of science . Also I am appreciated to them since they helped me to be aware about importance of science and technology culture for people .

I am also grateful to Mr. Muharrem İşeri and Mr. Berk Ekim Paşmakoğlu for their vital suggestions and help in constructing my datasets used in thesis work .

Finally, I would like to send my deepest thanks to my family, who are always encouraging me in this study with their whole love and belief . Without their strong belief to me, I could not be able to finish this work.

ABSTRACT

CLASSIFIER PERFORMANCES FOR CREDIT RISK ANALYSIS

Credit Risk Analysis (CRA) is an important and challenging data mining problem in financial analysis domain which is commonly used by many financial organizations. It became one of the most important and hot concept in finance sector since the real market's credit volume has significant growths while economies have fluctuations which has great impacts on financial organizations. CRA aims to decrease future losses by estimating the potential risk and eliminating the new credit proposal if the risk is higher than a defined tolerance value. In this thesis study, it is aimed to compose a combined classification model to have any little improvement of classification performance when it is compared with existing classifiers. This comparison is based on performance metrics such as accuracy, Receiver Operating Characteristics (ROC) and precision. Any little improvement in accuracy and optimality which seems insignificant, will reduce losses in a large loan portfolio and save very significant amounts which can be defined in terms of billions of dollars. Three different datasets are used in this study. Those are German Dataset, a national bank dataset and a synthetic dataset from a data mining tool. WEKA and GeneXproTools softwares are used to make experiments. Single classification techniques Support Vector Machine (SVM) is applied to German Dataset with different kernel functions. Logistic Regression (LR) is applied on synthetic data with different cross-validation. Finally, LR, SVM, Neural Networks, Naive Bayes and Dynamic Bayesian approaches are compared each other on real-life bank dataset. A hybrid approach proposed which combines best-performed single classifiers inside as a unique classifier. Results show that combined classification impact performance in terms of improvement among other single classification techniques.

ÖZET

KREDİ RİSK ANALİZİ İÇİN SINIFLANDIRMA TEKNİKLERİNİN PERFORMANSI

Kredi Risk Analizi (KRA) finansal alanda faaliyet gösteren kuruluşlar için önemli ve bir o kadar da zor bir veri madenciliği problemidir. Global kredi hacminin önemli ölçüde büyümesi ve bu büyümenin yanında ekonomik dalgalanmaların yaşanması, bu problemi daha önemli ve güncel bir noktaya getirmiştir. KRA potansiyel riskleri değerlendirir ve gelecekteki kayıpları azaltmak amacıyla yeni gelen kredi tekliflerini daha önceden belirlenmiş bir tolerans değeri ile karşılaştırarak elemine eder. Tez çalışmasında, kredi risk analizi yararına kullanılan sınıflandırma ölçütlerinin performanslarını ufak dahi olsa geliştirecek bir kombine sınıflandırma yöntemi oluşturmak amaçlanmıştır. Performans değerlerini karşılaştırmak için tutarlılık oranı, Alıcı İşletim Karakteristiği (AİK) ve hassaslık metrikleri kullanılmıştır. Tutarlılık oranında ya da optimal sonuçta önemsiz gibi gözükecek en ufak iyileştirme bile borç portföyünde kayıpları azaltacak ve değeri milyar dolarlar ile ifade edilebilecek tasarruflar sağlayacaktır. Bu tez çalışmasında üç farklı veri tabanı kullanılmıştır. Bunlar Alman Veritabanı, bir bankadan elde edilen veri tabanı ve özel bir yazılımın içinde bulunan veritabanıdır. Deneyler WEKA ve GeneXproTools yazılımları üzerinde gerçekleştirilmiştir. Destek Vektör Makinesi (DVM) sınıflandırma tekniği Alman Veritabanına uygulanmıştır. Lojistik Regresyon (LR) tekniği, GeneXproTools yazılımı üzerinde bulunan veritabanına farklı çapraz doğrulama yöntemleri eşliğinde uygulanmıştır. Son olarak DVM, LR, Sinir Ağları (NN), Basit Bayes Yaklaşımı ve Dinamik Bayes Yaklaşımı; bankadan elde edilen veritabanı üzerinde farklı örneklemeler için uygulanıp ortalama değerler alınarak performans değerleri incelenmiş ve önceki sonuçlara göre daha iyi veriler elde edildiği görülmüştür. Bu çalışmada, basit oylama modellemesi ile performans açısından iyi sonuçlar veren sınıflandırma tekniklerini bir araya getirilerek, hibrid bir sınıflandırma tekniği oluşturulmuş ve bu yeni yaklaşımın diğer sınıflandırma teknikleri ile performans açısından karşılaştırıldığında birçoğundan daha iyi sonuçlar verdiği görülmüştür.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

ABSTRACT iv

ÖZET v

LIST OF FIGURES viii

LIST OF TABLES ix

1. INTRODUCTION 1

    1.1 Credit Risk and Credit Risk Analysis 1

    1.2 Literature Review 4

    1.3 Proposed Method 6

    1.4 Datasets 10

    1.5 Outline 13

2. DIMENSION REDUCTION 14

    2.1 Feature Selection 16

        2.1.1 Decision Tree 18

3. CREDIT RISK CLASSIFICATION 25

    3.1 Logistic Regression 25

    3.2 Support Vector Machines 27

    3.3 Artificial Neural Networks 29

    3.4 Naive Bayes Approach 31

    3.5 Dynamic Bayesian Approach 32

        3.5.1 Dynamic Bayesian Approach to Credit Risk Analysis 33

4. EXPERIMENTS and RESULTS 35

    4.1 Performance Metrics 35

    4.2 Experiments 39

        4.2.1 SVM on German Dataset 39

4.2.2 Logistic Regression on Credit Approval Data..... 40

4.2.3 Experiments on Real-life Bank Dataset 43

4.3 Results..... 45

5. CONCLUSION 48

REFERENCES..... 50

LIST OF FIGURES

Figure 1.1. Proposed Approach Step – 1. 7

Figure 1.2. Proposed Approach Step – 2. 7

Figure 1.3. Proposed Approach Step – 3. 8

Figure 1.4. Proposed Approach Step – 4. 9

Figure 2.1. Big P Small n Problems. 15

Figure 2.2. Cosine Similarity. 16

Figure 2.3. Feature Selection Process with Validation. 18

Figure 2.4. A Simple Decision Tree. 20

Figure 2.5. ID3 Decision Tree Algorithm. 21

Figure 2.6. Entropy Function. 23

Figure 3.1. Logistic Curve. 26

Figure 3.2. Support Vector Machine. 27

Figure 3.3. Typical Neural Network. 30

Figure 3.4. Model of Computing Neuron. 30

Figure 3.5. Modelling Dynamic Bayesian Structure. 34

Figure 4.1. Confusion Matrix. 35

Figure 4.2. ROC Graph. 38

Figure 4.3. Generating ROC Points. 38

Figure 4.4. 5-Fold Cross Validation. 41

Figure 4.5. 8-Fold Cross Validation. 41

Figure 4.6. 10-Fold Cross Validation. 42

Figure 4.7. ROC Curve – 10-Fold Cross Validation. 42

Figure 4.8. Logistic Fit Curve. 43

Figure 4.9. Confusion Matrix for 10-Fold Cross Validation. 43

LIST OF TABLES

Table 4.1. Experiment on German Dataset with SVM. 40

Table 4.2. Experiment on Credit Approval Dataset with Logistic Regression. 40

Table 4.3. Confusion Matrix for 10-Fold Cross Validation. 43

Table 4.4. Experiment Results on Bank Dataset-1. 44

Table 4.5. Experiment Results on Bank Dataset-2. 45

1. INTRODUCTION

Credit is pervasive and a common feature of all global economies. Credit card issuers attempt to persuade almost everybody to live on credit, and at least some credit card holders find themselves living beyond their means and almost in perpetual credit. Individuals borrow to fullfil immediate physical needs such as house, furniture, car or consumer durables or to meet consumption expenditure for things as varied as marriage, education or holidays. Businesses borrow to facilitate expansion or to meet working capital requirements, among other things. Governments borrow to keep themselves afloat and hope to repay loans from future tax revenues or further loans. Central governments and big businesses borrow from abroad, and these debts, if not managed properly, can plunge the debtor country into a foreign exchange crisis. Examples are Argentina 2002, Asia 1997 and so on.

Credit can be defined as a transaction between two parties in which one (the creditor or lender) supplies money or monetary equivalent goods, services, in return for a promise of future payment by the other (the debtor or borrower). Such transactions normally include the payment of interest to the lender. The borrower is responsible for the borrowings he makes and he should face the consequences [1]. Naturally, some debtors do not pay back the credit as promised, creating a credit loss, sometimes even making the creditor bankrupt. That is the reason why credit risk analysis is an emerging issue. A proper credit risk analysis will bring to light the probability of credit loss arising out of genuine business factors.

1.1. Credit Risk and Credit Risk Analysis

Credit risk exists whenever a product or service is availed with a promise to settle payment in future, as agreed upon. The definition of credit risk is the probability of the loss (due ton on-recovery) emanating the credit extended, as a result of the non-fullfilment of contractual obligations arising from unwillingness or inability of the

counter-party or for any other reason [1]. Credit Risk Analysis (CRA) is the study from the perspective of a supplier of credit of a present/prospective claim on other economic agents in the form of debt, including trade receivables, loans, and public securities among others. Credit risk analysis is valid in a variety of decision contexts. Financial institutions have to answer following questions:

- What are the risks involved in extending a loan to a particular client?
- Given the risk level, how should it be priced?
- What about repayment capability?
- Is the management capable?
- Which economic & industrial factors will impact the performance of client?

Answering these questions is a crucial step for adjusting portfolios of financial organizations. Credit risk analysis plays a major role to help these organizations in finding answers of those questions [1]. And the principles how to manage credit risk was defined by Basel 2 Accord.

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 [2]. 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. The Committee provides regular cooperation on banking supervisory matters and famous for Basel 2 Capital Accord which is studied in this work. While financial institutions have faced difficulties over the years for a multitude of reasons, the major cause of serious banking problems continues to be directly related to lax credit standards for borrowers and counterparties, poor portfolio risk management or a lack of attention to changes in economic or other circumstances that can lead to a deterioration in the credit standing of a bank's counterparties. This experience is common in both G-10 and non-G-10 countries.

The goal of credit risk management is to maximise a bank's risk-adjusted rate of return by maintaining credit risk exposure within acceptable parameters. Banks need to manage the credit risk inherent in the entire portfolio as well as the risk in individual credits or transactions. Banks should also consider the relationships between credit risk and other risks. The effective management of credit risk is a critical component of a comprehensive approach to risk management and essential to the long-term success of any banking organisation.

For most banks, loans are the largest and most obvious source of credit risk; however, other sources of credit risk exist throughout the activities of a bank, including in the banking book and in the trading book, and both on and off the balance sheet. Banks are increasingly facing credit risk (or counterparty risk) in various financial instruments other than loans, including acceptances, interbank transactions, trade financing, foreign exchange transactions, financial futures, swaps, bonds, equities, options, and in the extension of commitments and guarantees, and the settlement of transactions.

Since exposure to credit risk continues to be the leading source of problems in banks world-wide, banks and their supervisors should be able to draw useful lessons from past experiences. Banks should now have a keen awareness of the need to identify, measure, monitor and control credit risk as well as to determine that they hold adequate capital against these risks and that they are adequately compensated for risks incurred. The Basel Committee is issuing this document in order to encourage banking supervisors globally to promote sound practices for managing credit risk. Although the principles contained in this paper are most clearly applicable to the business of lending, they should be applied to all activities where credit risk is present.

A further particular instance of credit risk relates to the process of settling financial transactions. If one side of a transaction is settled but the other fails, a loss may be incurred that is equal to the principal amount of the transaction. Even if one party is simply late in settling, then the other party may incur a loss relating to missed investment opportunities. Settlement risk (i.e. the risk that the completion or settlement of a financial transaction will fail to take place as expected) thus includes elements of liquidity, market, operational and reputational risk as well as credit risk. The level of risk is determined by the particular arrangements for settlement. Factors in such arrangements that have a

bearing on credit risk include: the timing of the exchange of value; payment/settlement finality; and the role of intermediaries and clearing houses [3].

1.2. Literature Review

The models for credit risk evaluation mainly include two categories: traditional statistical model and artificial intelligence model. Traditional statistical models include multiple discriminant analysis (MDA) model and the logistic regression model. The advantage of statistical is its obvious explanatory, but the preconditions of application are too strict, such as discriminant analysis model requires that the distribution of data analyzed is multivariate normal with covariance. With the development of information technology, some classification and prediction algorithms based on artificial intelligence and machine learning technologies are also applied to the credit risk assessment [4], including artificial neural networks and decision trees.

Data mining is the process of discovering actionable and meaningful patterns, profiles, and trends by sniffing through very large databases using pattern recognition technologies, such as, decision trees, neural networks, k-nearest neighborhood, and association rules. Its dominant goal is generating non-obvious yet useful information for decision makers from large-scale data [5].

Due to importance of credit risk analysis, many methods were widely applied to credit risk measurement tasks: linear discriminant analysis, logit analysis, probit analysis, linear programming, integer programming, support vector machine, genetic algorithm, and some hybrid models. Gallo[6] showed that artificial neural networks represent an easily customizable tool for modelling learning behaviour of agents and for studying a lot of problems very difficult to analyze with standard economic models in 2006. Shachmurove [7] stated that artificial neural networks have the ability to analyze complex patterns quickly and with a high degree of accuracy in his work at 2002.

Satchidananda [8] compared the efficacy of the decision trees with logistic regression and stated that decision trees have a good performance among logistic regression. Yu, Huang, Cai and Hu [9] developed a work on a comparative study for data

mining algorithms such as logistic regression, support vector machines and decision trees on credit risk evaluation and they stated that logistic regression outperforms other classification techniques at 2010.

Wong, Wang and Lai [10] worked on a new fuzzy support vector machine to evaluate credit risk. Empirical tests on three public datasets show that it can have 5etwo discriminatory power than the standard support vector machine and the fuzzy support vector machine if appropriate kernel and membership generation method are chosen. Hao, Lin, and Tsai [11] worked also on support vector machine on fuzzy hyper-plane and searched for its application on credit risk. The experimental results show that the proposed method performs fairly well. However, there are several limitations that may restrict the use of proposed fuzzy SVM models for its applications. Those are membership generation and computational complexity. Also, Kaya, Gürgen and Okay obtained good results for PD estimation by logistic regression on German Credit Dataset [12].

Zhang, Orgun, Baxter, and Lin [13] developed an application of element oriented analysis based credit scoring on Brazilian credit card dataset. Their model is applicable in credit risk assessment in the context of the accuracy measures from the existing network on the same dataset and the benchmark models. Huang, Chen, Hsu, Chen W., Wu [14] made a market comparative study between support vector machines and neural networks. They applied their method on Taiwan's and USA's banks datasets and the results showed that support vector machines achieved accuracy comparable to that of backpropagation neural networks.

Doumpos, Zopounidis [15] combined different classification techniques in a stacked generalization approach. Results showed that combination approach has the potential for improving the efficiency of credit risk models over individual approaches. 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 [16]. Campos, Ruiz, Agell, and Angulo [17] worked on innovative soft computing techniques for

financial credit risk measurement. Kotsiantis [18], proposed a hybrid model to evaluate credit risk which is also a basis for this thesis work.

1.3. Proposed Method

As described in previous section, most of classification techniques are used to classify credit risk analysis data to have beter and accurate results. But most of them are applied as a single classification model and results are compared each other after experiments.

Recently in the area of machine learning and data mining, the concept of combining classifiers is proposed as a new direction for the improvement of the performance of individual classifiers. These classifiers could be based on a variety of classification methodologies, and could achieve different rates of correctly classified individuals. The goal of classification result integration algorithms is to generate more certain, precise and accurate system results.

Numerous methods have been suggested for the creation of ensemble of classifiers. Mechanisms that are used to build ensemble of classifiers include [16]:

- Using different subsets of training data with a single learning method
- Using different training parameters with a single training method (e.g., using
- different initial weights for each neural network in an ensemble)
- Using different learning methods.

According to this new trend, this study is based on combining several classification techniques to have better performance results in comparison with single classification techniques. To accomplish this aim, a modular approach is applied which can be seen in Figure 1.1, Figure 1.2, Figure 1.3 and Figure 1.4.

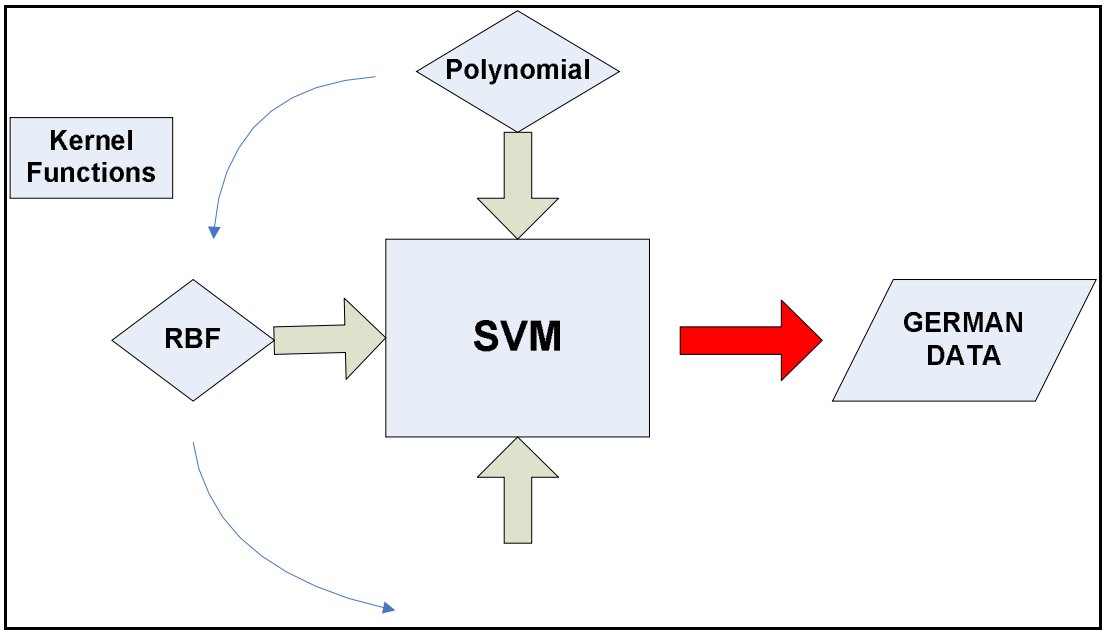


Figure 1.1 Proposed Approach Step – 1

At first step as in Figure 1.1, German dataset was used. With three different types of kernel functions, support vector machine classifier has been applied. One of them was chosen whose performance values outperform the others.

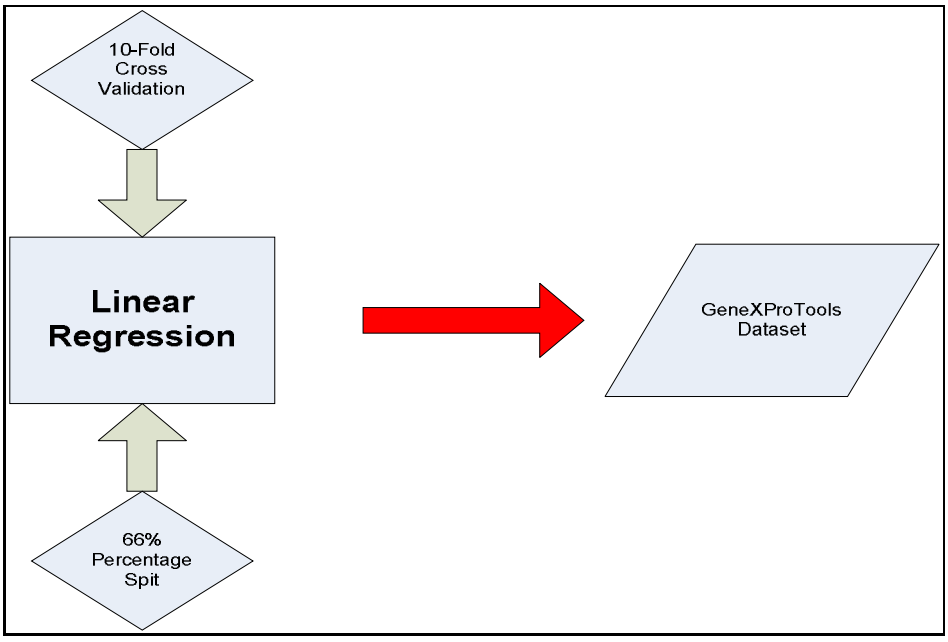


Figure 1.2 Proposed Approach Step – 2

At second step as in Figure 1.2, logistic regression with different cross-validations and percentage splits was applied to GeneXproTools' dataset. Among performance values, the most fitted one was chosen for next step.

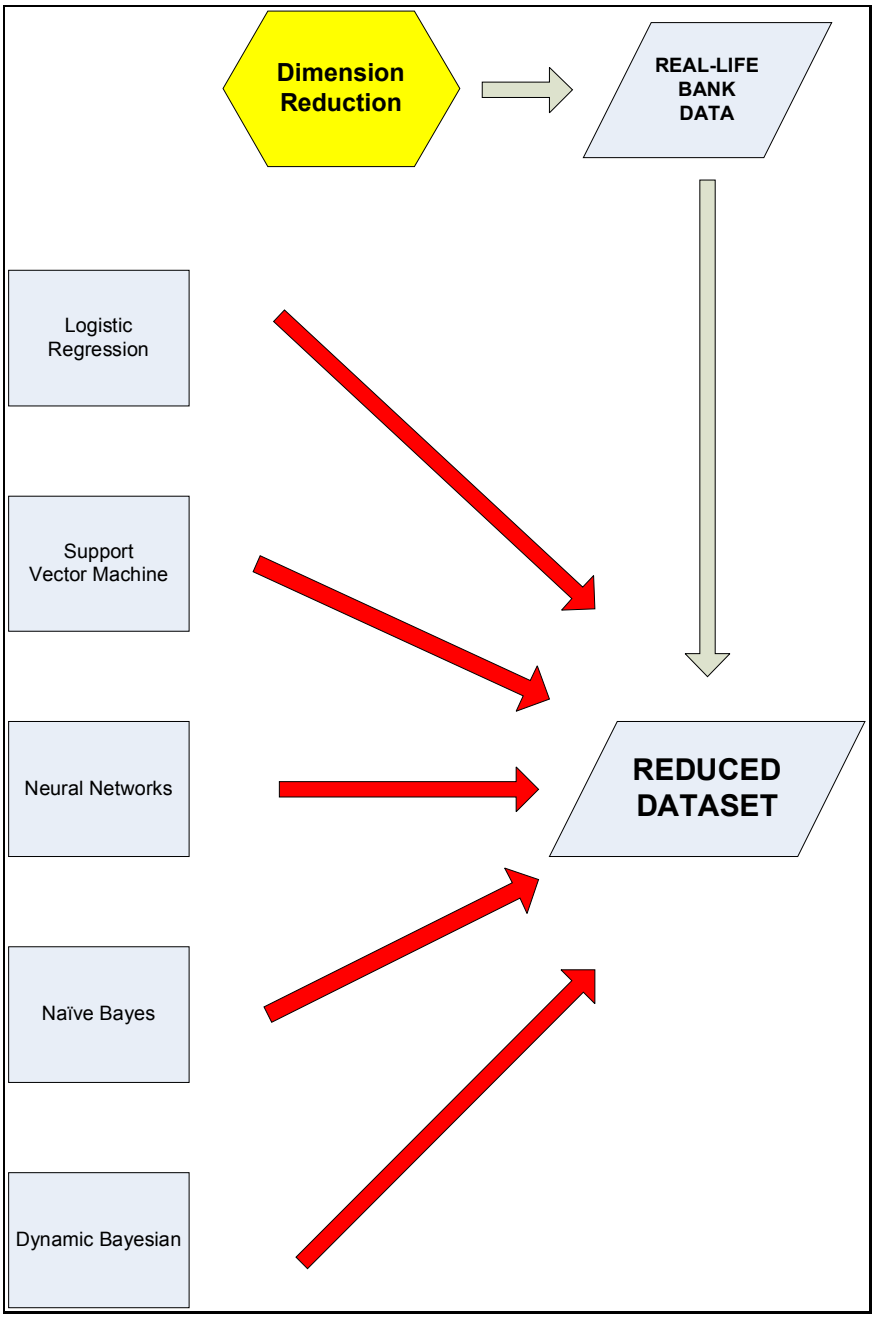


Figure 1.3 Proposed Approach Step – 3

All three datasets are very similar in this thesis work. They include similar attributes with similar number of total instances. Thus, the outcomes from first and second step are used as inputs at third step. In this step, bank dataset was reduced after dimension reduction first, then on this reduced dataset, SVM and LR from first two steps were applied. Performances are compared also with neural network, naive bayes approach and dynamic bayesian approach as in Figure1.3. Among results, to construct a hybrid model for future work, some of classifiers are selected and some of them are discarded.

Our aim in this study is creating a hybrid classifier which has any little improvement on classification accuracy. After applying five different classification techniques to real-life bank dataset, we have results in terms of accuracy. To strengthen this accuracy measure, we applied simple voting mechanism with weighted classifiers and have a final result for our proposed method in this thesis work. Following Figure 1.4 shows the procedure:

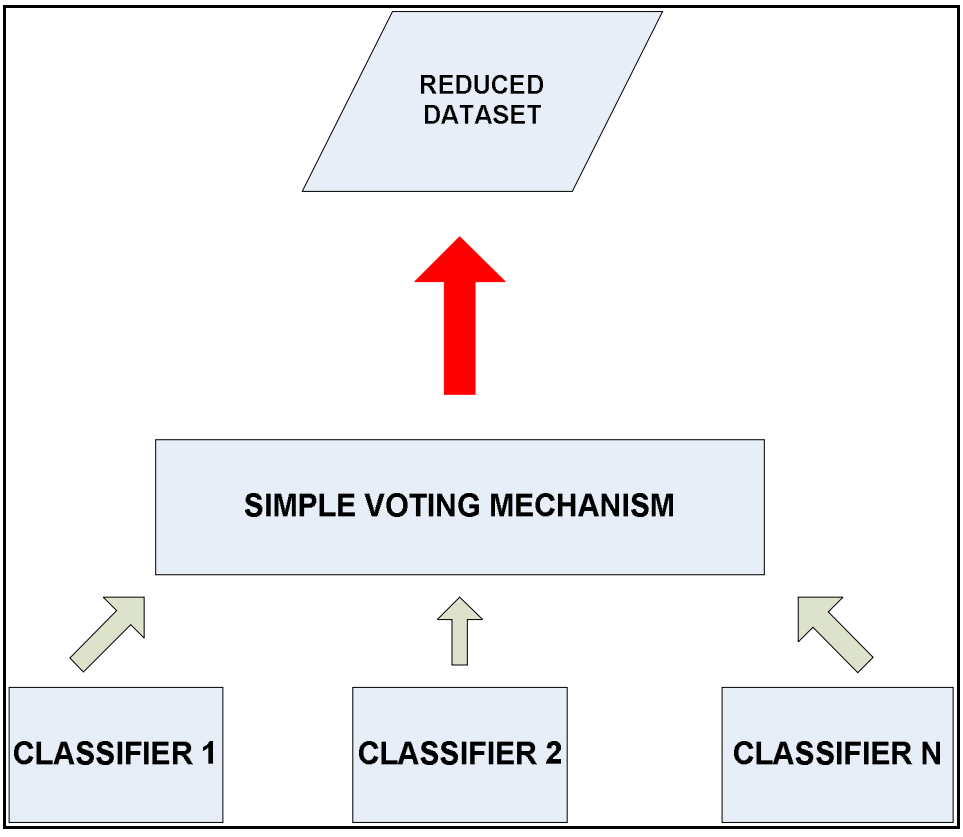


Figure 1.4 Proposed Approach – Final Step

1.4. Datasets

Three different datasets are used in three step of this thesis work. At first step German Dataset is used. German Dataset is a public access credit database that this thesis study utilizes [19]. Every row of the database corresponds to a credit customer with each element of every row being a feature for the customer. More precisely, each column represents an attribute for the customers. In fact an attribute is nothing but a kind of predictive information like credit history, annual income, credit purpose, credit amount and etc… Attributes are used by the loan officers and experts in order to decide whether the credit applicant is creditworthy or not. German dataset has continuous (numerical) and discrete (categorical or symbolic) attributes.

German dataset has 1000 data instances regarding retail credit applications. 300 of the instances are of the bad borrowers and 700 of them are of the good borrowers. German dataset contains a total of 20 attributes 7 of which are numerical and 13 of which are categorical. Some of the attributes presented in German Database are as follows: status of existing checking account (categorical), credit history (categorical), credit purpose (categorical), credit amount (numerical), present employment since (categorical) and so on.

Since both databases have numerical and categorical data together, an adjustment for the datasets is necessary in order to make them appropriate for the classifiers of our interest namely, LR and k-NN. For this reason the numerical data are converted to categorical format. In other words, continuous values of the attributes are converted to discrete categories according to discrete intervals. For example, the attribute regarding *credit amount*, is a continuous number representing the amount of the loan that the credit applicant wants to borrow. It is denoted as *attribute 4* in German dataset. Since the credit amount may take a wide range of continuous values, the values of this attribute should be converted to discrete categorical values in accordance with appropriate intervals as follows:

- If the credit amount is smaller than 1000 DM (Deutsche Mark), it is converted to category 1 and encoded as C041 indicating the 1st category of attribute 4.
- Else if the credit amount is between 1000 DM and 4000 DM, it is converted to category 2 and encoded as C042 indicating the 2nd category of attribute 4.
- Else if the credit amount is between 4000 DM and 10000 DM, it is converted to category 3 and encoded as C043 indicating the 3rd category of attribute 4.
- Else if the credit amount is greater than 10000 DM, it is converted to category 4 and encoded as C044 indicating the 4th category of attribute 4.

For both datasets, the numbers of columns representing input features (attributes) are rather high and it should be decreased to a reasonable number. This should be done intuitively by trial and error method and also decision tree mechanism. Alternatively, factor analysis can be used to see which factors are more predictive. Principal component analysis or independent component analysis can also be used for dimension reduction. However in this case components take the place of the factors and it is not very straightforward to see which factors should be used in the model.

Second dataset used in this thesis study is from special software "GeneXproTools 4.0". This dataset is a Credit Approval Data. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. There are a total of 690 instances, 307 of which are successful "1" applications and 383 unsuccessful "0". The total attribute number is 18.

Training and testing datasets used and generated by standard random partitioning of the original dataset. All missing values in the original dataset were replaced by the average. For training, a total of 460 cases were used, whereas the remaining 230 cases were used for testing.

A total of nine attributes in the original dataset are nominal and therefore had to be converted to numbers in order to simplify classification operations. The original dataset was obtained through the UCI Machine Learning Repository and is available for download [20].

Third dataset is a real-life dataset from a Turkish Bank. It consists data of individual loans for a period between 2009 and 2010. The data was taken randomly without any sampling methodology, and it is a small subset of the original data.

It has 1066 samples at total. Dataset includes 24 attributes both categorical and nominal 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.742 of them are “good” customers and 324 of them are “bad” customers. For the purpose of classification clarifyness all attributes are converged to categorical values. Due to confidentiality regulations, name of attributes are converted to symbols.

The dataset consists of 24 features; six of them are *categorical* and the others are *continuous* variables. Continuous data is information that is going on or extending without interruption and a variable can have any of the data points. Money, age or ranges, such as between minus one and one etc, are accepted as continuous variables. On the other hand, categorical data corresponds to groups or categories which are resulted from placing individuals into groups or categories. 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 applicant. 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 12 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.

1.5. Outline

Previous parts include information about Credit Risk and Credit Risk Analysis concepts. Also it described proposed approach used in this study, literature review and datasets of this work. Following chapters describe thesis study in detail. Chapter 2 describes dimension reduction concept and its necessity for this study. Chapter 3 includes classification techniques Logistic Regression, Support Vector Machines, Neural Networks and Bayesian Approach in detail context. Chapter 4 describes experiments on datasets and their results among performance metrics. Finally, in Chapter 5, detailed analysis, comments and conclusions take place.

2. DIMENSION REDUCTION

When data objects that are the subject of analysis using machine learning techniques are described by a large number of features (i.e. the data is high dimension) it is often beneficial to reduce the dimension of the data. Dimension reduction can be beneficial not only for reasons of computational efficiency but also because it can improve the accuracy of the analysis. The set of techniques that can be employed for dimension reduction can be partitioned in two important ways; they can be separated into techniques that apply to supervised or unsupervised learning and into techniques that either feature selection or feature extraction [22].

Data analysis problems where the data objects have a large number of features are becoming more prevalent areas such as multimedia data analysis and bioinformatics. In these situations it is often beneficial to reduce the dimension of the data (describe it in less features) in order to improve the efficiency and accuracy of data analysis. Statisticians sometimes talk of problems that are “Big p Small n”; these are extreme examples of situations where dimension reduction (DR) is necessary because the number of explanatory variables p exceeds (sometimes greatly exceeds) the number of samples n [23]. From a statistical point of view, it is desirable that the number of examples in the training set should significantly exceed the number of features used to describe those examples as in Figure 2.1. In theory the number of examples needs to increase exponentially with the number of features if inference is to be made about the data. In practice this is not the case as real high-dimension data will only occupy a manifold in the input space so the implicit dimension of the data will be less than the number of features p. For this reason data sets as depicted in Figure 2.1.b can still be analysed. Nevertheless, traditional algorithms used in machine learning and pattern recognition applications are often susceptible to the well-known problem of the curse of dimensionality [24], which refers to the degradation in the performance of a given learning algorithm as the number of features increases. To deal with this issue, dimension reduction techniques are often applied as a data pre-processing step or as part of the data analysis to simplify the data model. This typically involves the identification of a suitable low-dimensional representation for the original high-dimensional dataset. By working

with this reduced representation, tasks such as classification or clustering can often yield more accurate and readily interpretable results, while computational costs may also be significantly reduced.

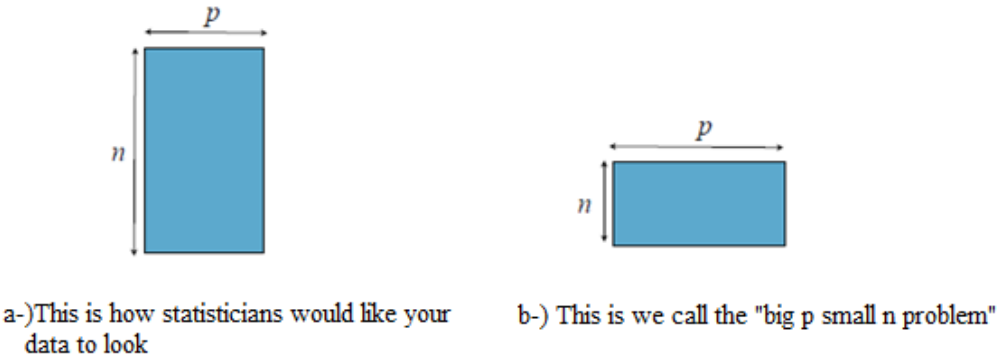


Figure 2.1 Big p small n problems are problems where the number of features in a dataset is large compared with the other objects

The motivation for dimension reduction can be summarised as follows:

- The identification of a reduced set of features that are predictive of outcomes can be very useful from a knowledge discovery perspective.
- For many learning algorithms, the training and/or classification time increases directly with the number of features.
- Noisy or irrelevant features can have the same inuence on classification as predictive features so they will impact negatively on accuracy.
- Things look more similar on average the more features used to describe them (see Figure 2.2). The example in the figure shows that the resolution of a similarity measure can be worse in 20D than in a 5D space.

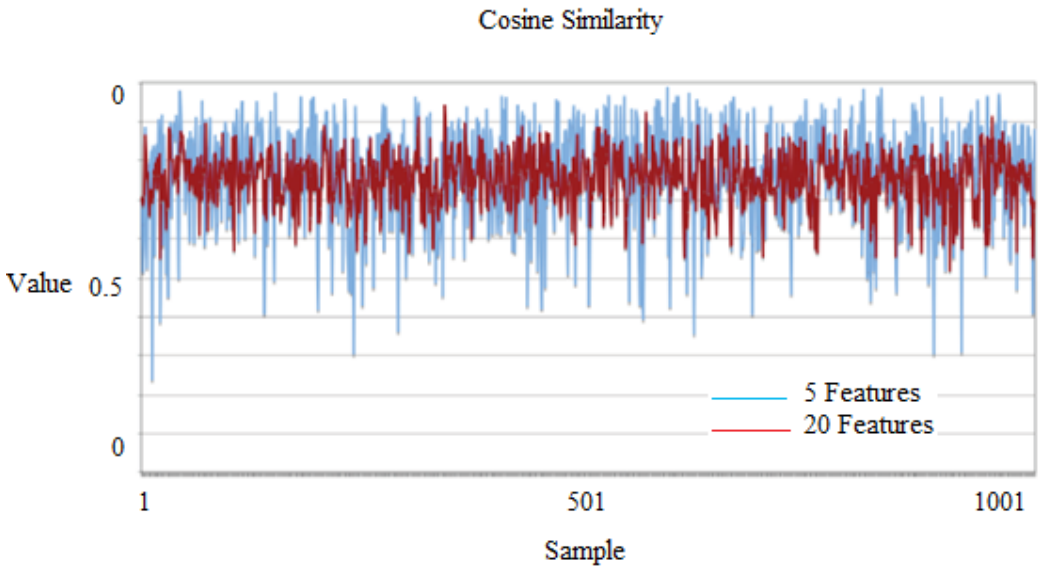


Figure 2.2. Cosine Similarity Between Randomly Generated Data Objects by 5 and 20 Features

The more dimensions used to describe objects the more similar on average things appear. This figure shows cosine similarity between randomly generated data objects described by 5 and 20 features.It is clear thatin 20 dimensions similarity has a lower variance than in 5.

2.1. Feature Selection

The majority of real-world classification problems require supervised learning where the underlying class probabilities and class-conditional probabilities are unknown, and each instance is associated with a class label. In real-world situations, relevant features are often unknown *a priori*. Therefore, many candidate features are introduced to better represent the domain. Unfortunately many of these are either partially or completely irrelevant/redundant to the target concept. A relevant feature is neither irrelevant nor redundant to the target concept; an irrelevant feature does not affect the target concept in any way, and a redundant feature does not add anything new to the target concept [25]. In many applications, the size of a dataset is so large that learning might not work as well before removing these unwanted features. Reducing the number

of irrelevant/redundant features drastically reduces the running time of a learning algorithm and yields a more general concept. This helps in getting a better insight into the underlying concept of a real-world classification problem [26]. *Feature selection* methods try to pick a subset of features that are relevant to the target concept. Feature selection is defined by many authors by looking at it from various angles. But as expected, many of those are similar in intuition and/or content. The following lists those that are conceptually different and cover a range of definitions.

- *Idealized*: Find the minimally sized feature subset that is necessary and sufficient to the target concept [27].
- *Classical*: Select a subset of M features from a set of N features, $M < N$, such that the value of a criterion function is optimized over all subsets of size M [28].
- *Improving Prediction Accuracy*: The aim of feature selection is to choose a subset of features for improving prediction accuracy or decreasing the size of the structure without significantly decreasing prediction accuracy of the classifier built using only the selected features [26].
- *Approximating Original Class Distribution*: The goal of feature selection is to select a small subset such that the resulting class distribution, given only the values for the selected features, is as close as possible to the original class distribution given all feature values [26].

Feature selection attempts to select the minimally sized subset of features according to the following criteria. The criteria can be:

- The classification accuracy does not significantly decrease; and
- The resulting class distribution, given only the values for the selected features, is as close as possible to the original class distribution, given all features.

Figure 2.3 shows the steps of feature selection process.

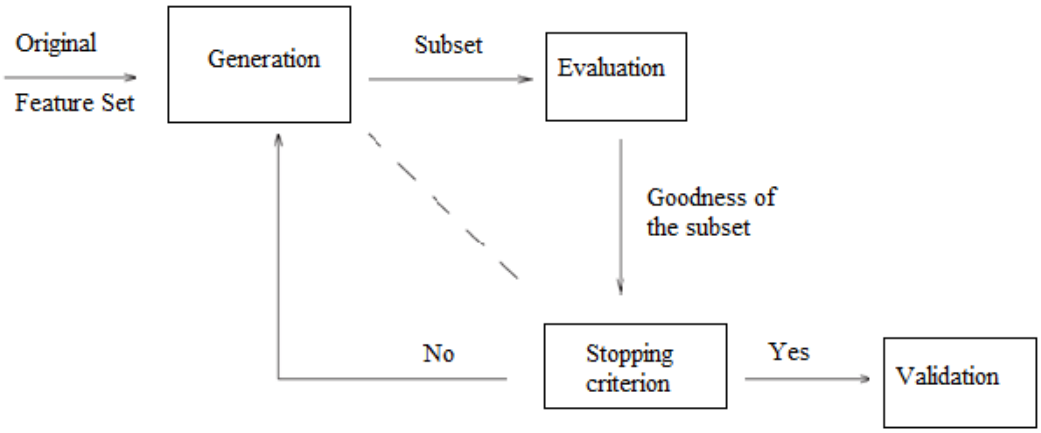


Figure 2.3 Feature Selection Process with Validation

Many approaches have been proposed for feature selection such as the well known methods of decision tree. Decision tree utilizes tree induction algorithm with the entropy as an evaluation measure [29]. Following section includes more details about decision trees.

2.1.1. Decision Tree

Decision trees are powerful and popular tools for classification and prediction. The attractiveness of decision trees is due to the fact that, in contrast to neural networks, decision trees represent *rules*. Rules can readily be expressed so that humans can understand them or even directly used in a database access language like SQL so that records falling into a particular category may be retrieved.

In some applications, the accuracy of a classification or prediction is the only thing that matters. In such situations we do not necessarily care how or why the model works. In other situations, the ability to explain the reason for a decision is crucial. In marketing one has describe the customer segments to marketing professionals, so that they can utilize this knowledge in launching a successful marketing campaign. This domain's experts must recognize and approve this discovered knowledge, and for this we need good descriptions. There are a variety of algorithms for building decision trees that share the desirable quality of interpretability. A well known and frequently used over the

years is C4.5 which is also used in this thesis study. In WEKA software J48 decision tree, which corresponds to C4.5 is used for feature selection purposes on experiments.

Decision tree is a classifier in the form of a tree structure (see Figure 2.4), where each node is either:

- A *leaf node* - indicates the value of the target attribute (class) of examples, or
- A *decision node* - specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test.

A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the classification of the instance.

Decision tree induction is a typical inductive approach to learn knowledge on classification. The key requirements to do mining with decision trees are:

- *Attribute-value description*: object or case must be expressible in terms of a fixed collection of properties or attributes. This means that we need to discretize continuous attributes, or this must have been provided in the algorithm.

- *Predefined classes (target attribute values)*: The categories to which examples are to be assigned must have been established beforehand (supervised data).
- *Discrete classes*: A case does or does not belong to a particular class, and there must be more cases than classes.
- *Sufficient data*: Usually hundreds or even thousands of training cases.

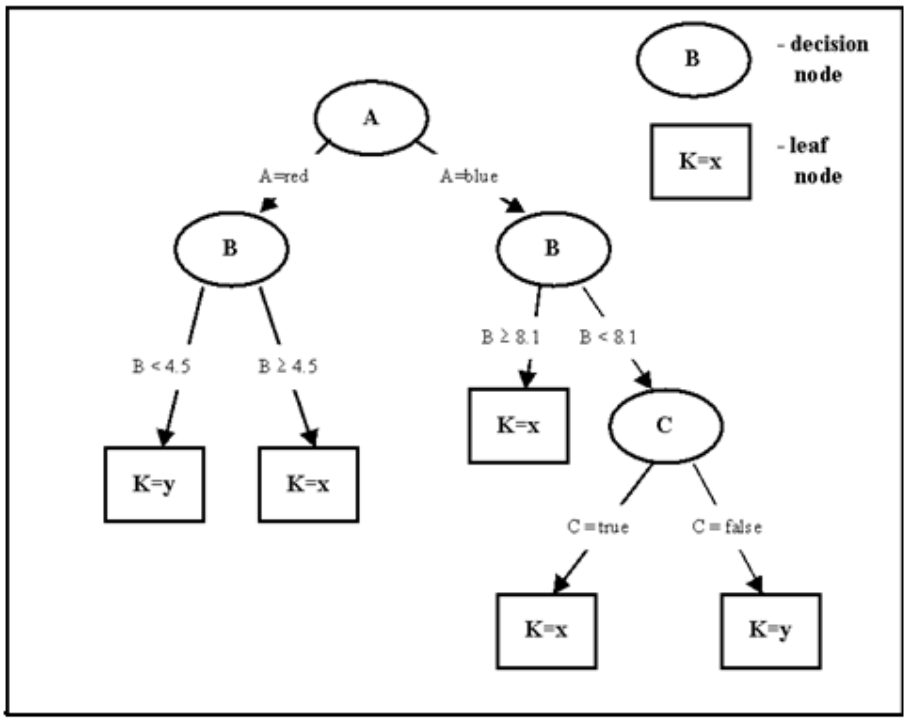


Figure 2.4. A Simple Decision Tree

Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. Decision tree programs construct a decision tree *T* from a set of training cases.

J. Ross Quinlan originally developed ID3 at the University of Sydney. ID3 is based on the Concept Learning System (CLS) algorithm [30].

Function ID 3

Input : (R: a set of non-target attributes,

C: the target attribute,

S: a training set) returns a decision tree;

Begin

If S is empty, return a single node with value

Failure;

If S consists of records all with same value for the target attribute, return a single leaf node with that value;

If R is empty, then return a single node with the value of the most frequent of the values og the target attribute that are found in records of S;[in that case there may be errors, examples that will be improperly classified];

Let A will be the attribute with largest Gain(A,S) among attributes in R;

Let $\{ a_j | j = 1,2,\ldots,m\}$ be the values of attribute A;

Let $\{ S_j | j = 1,2,\ldots,m\}$ be the subsets of S consisting respectively of records with value a_j for A;

Return a tree with root labeled A and arcs labeled a_1, a_2, …, a_m going respectively to the trees (ID3(R-{A}, C, S_1), ID3(R-{A}, C, S_2), …, ID3(R-{A}, C, S_m);

Recursively apply ID3 to subsets $\{S_j | j = 1,2,\ldots,m\}$ until they are empty

end

Figure 2.5. ID3 Decision Tree Algorithm

ID3 searches through the attributes of the training instances and extracts the attribute that best separates the given examples. If the attribute perfectly classifies the

training sets then ID3 stops; otherwise it recursively operates on the m (where m = number of possible values of an attribute) partitioned subsets to get their "best" attribute. The algorithm uses a greedy search, that is, it picks the best attribute and never looks back to reconsider earlier choices. ID3 may misclassify data.

The central focus of the decision tree growing algorithm is selecting which attribute to test at each node in the tree. For the selection of the attribute with the most inhomogeneous class distribution the algorithm uses the concept of entropy.

The estimation criterion in the decision tree algorithm is the selection of an attribute to test at each decision node in the tree. The goal is to select the attribute that is most useful for classifying examples. A good quantitative measure of the worth of an attribute is a statistical property called *information gain* that measures how well a given attribute separates the training examples according to their target classification. This measure is used to select among the candidate attributes at each step while growing the tree.

In order to define information gain precisely, we need to define a measure commonly used in information theory, called entropy that characterizes the (im) purity of an arbitrary collection of examples. Given a set S, containing only positive and negative examples of some target concept (a 2 class problem), the entropy of set S relative to this simple, binary classification is defined as:

$$\text{Entropy}(S) = - p_p \log_2 p_p - p_n \log_2 p_n \qquad (2.1)$$

where p_p is the proportion of positive examples in S and p_n is the proportion of negative examples in S. In all calculations involving entropy we define 0log0 to be 0.

To illustrate, suppose S is a collection of 25 examples including 15 positive and 10 negative examples [15+, 10-]. Then the entropy of S relative to this classification is

$$\text{Entropy}(S) = - (15/25) \log_2 (15/25) - (10/25) \log_2 (10/25) = 0.97 \qquad (2.2)$$

Notice that the entropy is 0 if all members of S belong to the same class. Note the entropy is 1 (at its maximum) when the collection contains an equal number of positive and negative examples. If the collection contains unequal numbers of positive and negative examples, the entropy is between 0 and 1. Figure 2.6 shows the form of the entropy function relative to a binary classification, as p_+ varies between 0 and 1.

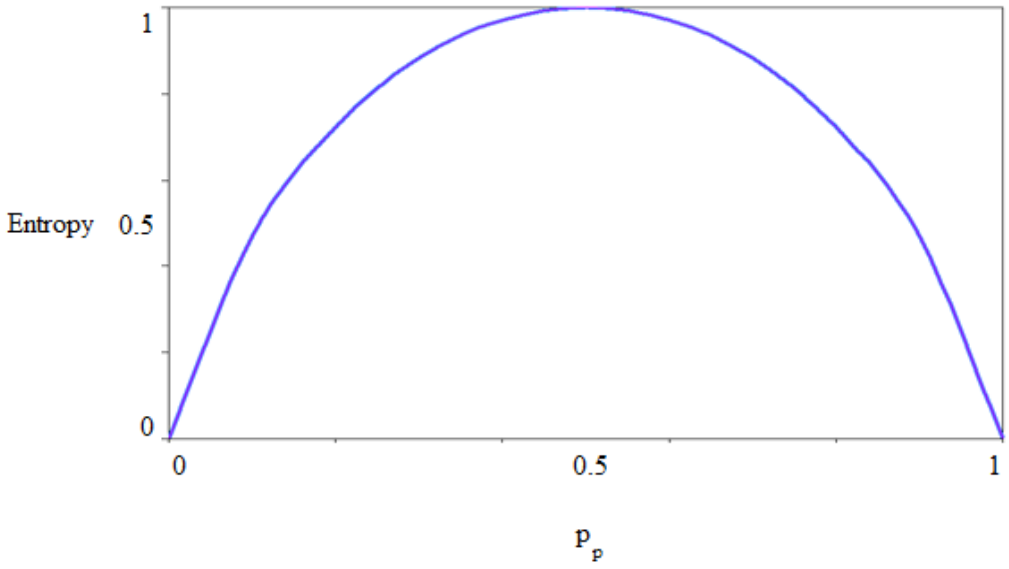


Figure 2.6. The Entropy Function relative to a binary classification

One interpretation of entropy from information theory is that it specifies the minimum number of bits of information needed to encode the classification of an arbitrary member of S (i.e., a member of S drawn at random with uniform probability). For example, if p_p is 1, the receiver knows the drawn example will be positive, so no message need be sent, and the entropy is 0. On the other hand, if p_p is 0.5, one bit is required to indicate whether the drawn example is positive or negative. If p_p is 0.8 then a collection of messages can be encoded using on average less than 1 bit per message by assigning shorter codes to collections of positive examples and longer codes to less likely negative examples.

Given entropy as a measure of the impurity in a collection of training examples, we can now define a measure of the effectiveness of an attribute in classifying the training data. The measure we will use, called *information gain*, is simply the expected

reduction in entropy caused by partitioning the examples according to this attribute. More precisely, the information gain, *Gain (S, A)* of an attribute *A*, relative to a collection of examples S, is defined as

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|Sv|}{|S|} Entropy(Sv) \quad (2.3)$$

where *Values(A)* is the set of all possible values for attribute *A*, and S_v is the subset of *S* for which attribute *A* has value *v* (i.e., S_v = {s Î S | *A(s)* = *v*}). First term in the equation for *Gain* is just the entropy of the original collection *S* and the second term is the expected value of the entropy after S is partitioned using attribute *A*. The expected entropy described by this second term is simply the sum of the entropies of each subset S_v, weighted by the fraction of examples $|S_v|/|S|$ that belong to S_v. *Gain (S, A)* is therefore the expected reduction in entropy caused by knowing the value of attribute *A*. Put another way, *Gain(S, A)* is the information provided about the target attribute value, given the value of some other attribute A. The value of *Gain(S, A)* is the number of bits saved when encoding the target value of an arbitrary member of *S*, by knowing the value of attribute *A*.

The process of selecting a new attribute and partitioning the training examples is now repeated for each non-terminal descendant node, this time using only the training examples associated with that node. Attributes that have been incorporated higher in the tree are excluded, so that any given attribute can appear at most once along any path through the tree. This process continues for each new leaf node until either of two conditions is met:

- Every attribute has already been included along this path through the tree, or
- The training examples associated with this leaf node all have the same target attribute value (i.e., their entropy is zero).

3. CREDIT RISK CLASSIFICATION

This chapter describes several classification techniques in detail which are used in this study as a basis for experiments. Those data mining algorithms are logistic regression, support vector machines, neural networks, naive bayesian approach and dynamic bayesian approach.

3.1. Logistic Regression

In statistics, logistic regression is used for prediction of the probability of occurrence of an event by fitting data to a logit function logistic curve. It is a generalized linear model used for binomial regression. Like many forms of regression analysis, it makes use of several predictor variables that may be either numerical or categorical. For example, the probability that a person has a heart attack within a specified time period might be predicted from knowledge of the person's age, sex and body mass index. Logistic regression is used extensively in the medical and social sciences fields, as well as marketing applications such as prediction of a customer's propensity to purchase a product or cease a subscription.

An explanation of logistic regression begins with an explanation of the logistic function:

$$f(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}} \tag{3.1}$$

A graph of the function is shown in Figure 3.1. The input is z and the output is $f(z)$. The logistic function is useful because it can take as an input any value from negative infinity to positive infinity, whereas the output is confined to values between 0 and 1. The variable z represents the exposure to some set of independent variables, while $f(z)$ represents the probability of a particular outcome, given that set of explanatory

variables. The variable z is a measure of the total contribution of all the independent variables used in the model and is known as the logit.

The variable z is usually defined as:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_k x_k, \qquad (3.2)$$

where β_0 is called the "intercept" and β_1, β_2, β_3, and so on, are called the "regression coefficients" of x_1, x_2, *and x3* respectively. The intercept is the value of z when the values of all independent variables are zeros (e.g. the value of z in someone with no risk factors). Each of the regression coefficients describes the size of the contribution of that risk factor. A positive regression coefficient means that the explanatory variable increases the probability of the outcome, while a negative regression coefficient means that the variable decreases the probability of that outcome; a large regression coefficient means that the risk factor strongly influences the probability of that outcome, while a near-zero regression coefficient means that that risk factor has little influence on the probability of that outcome.

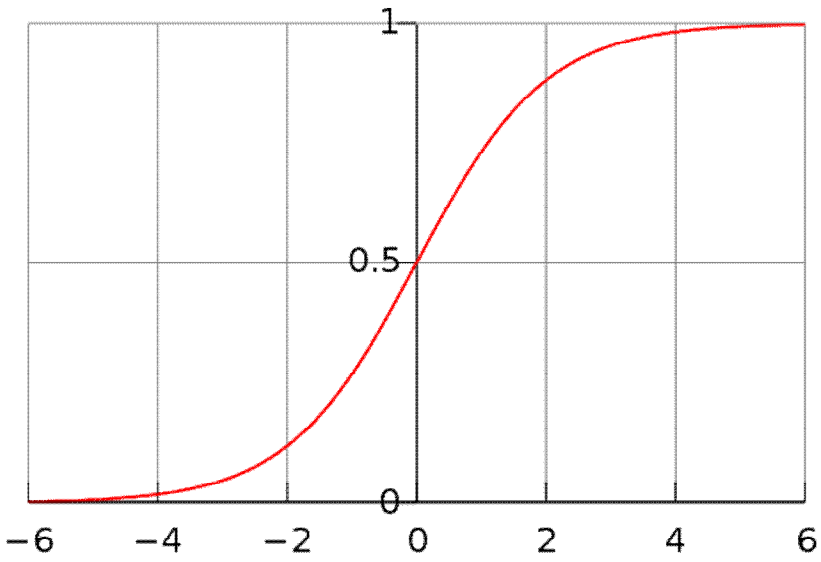


Figure 3.1. Logistic Curve

Logistic regression is a useful way of describing the relationship between one or more independent variables (e.g., age, sex, etc.) and a binary response variable, expressed as a probability, that has only two values, such as having cancer ("has cancer" or "doesn't have cancer"). These are binary outcome measures. It is particularly useful where the dataset is very large, and the predictor variables do not behave in orderly ways, or obey the assumptions required of discriminant analysis [31].

The results of logistic regression are somewhat mystifying, since the original variable of interest (such as survival) disappears and is replaced by the logit. But with a good understanding of how to convert logit values into probabilities, this method can be powerful tool.

3.2. Support Vector Machines

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 3.2.

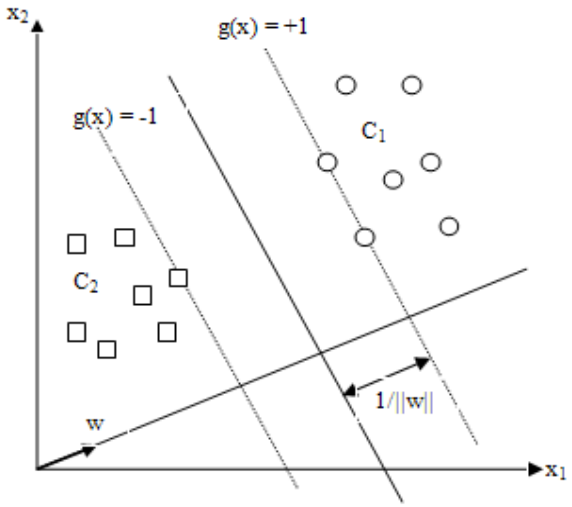


Figure 3.2. Support Vector Machine

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*= (*xt, rt)* where *x* is the input, *r* is the target value and *r*Î {-1, +1}. The separating hyperplane can be expressed as function of *x* [32]:

$$g(x) = w^t x^t + w_0 \tag{3.3}$$

where $g(x)$ is 1 for $RT = 1$ and $g(x)$ - 1 for $rt = 1$. The distance of x_t to the hyperplane that we want to maximize should be greater than ρ for t. The formula is given in Equation 3.4. Thus, in order to obtain maximum margin, we should minimize $\| w \|$ [32].

$$\frac{r^t \left(w^t x^t + w_0 \right)}{\|w\|} \geq p \tag{3.4}$$

The cost function in SVM is obtained as stated in Equation 3.5 where the problem is converted into an optimization problem.

(3.5)

When the minimum *w* is found, the maximum margin can be calculated by the formula given in Equation 3.6.

$$p = \frac{1}{\|w\|} \tag{3.6}$$

The data points which lie on the margin are called as *support vectors* which satisfy the formula given in (3.7).

$$r^t \left(w^t + w_0 \right) = 1 \tag{3.7}$$

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.

3.3. Artificial Neural Networks

Artificial neural networks (ANN) are calculable structures consisting of a number of the same types elements, which execute relatively simple functions. Processes in ANS sometimes are associated with ones, which take place in the nervous system of living organisms [33].

The key difference between neural networks and other problem-solving methods is that neural networks in a real sense learn by example rather than having to be programmed with specific, preconceived rules. In other words, neural networks can be seen as a non-parametric statistical procedure that uses the observed data to estimate the unknown function [34].

The most important feature of neural networks is their ability to learn. Just like human brain, neural networks can learn by example and dynamically modify themselves to fit the data presented. Furthermore, neural models are also able to learn from very noisy, distorted, or incomplete sample data, which render other methods useless [35].

An artificial neural network consists of an input layer, the inner (hidden) layers and an output layer as in Figure 3.3. The input layer serves the purpose of taking in information (e.g. specific indicator values) and passing it on to the downstream neurons via the connections shown in Figure 3.3. These links are assigned weights in an artificial neural network and thus control the flow of information [36].

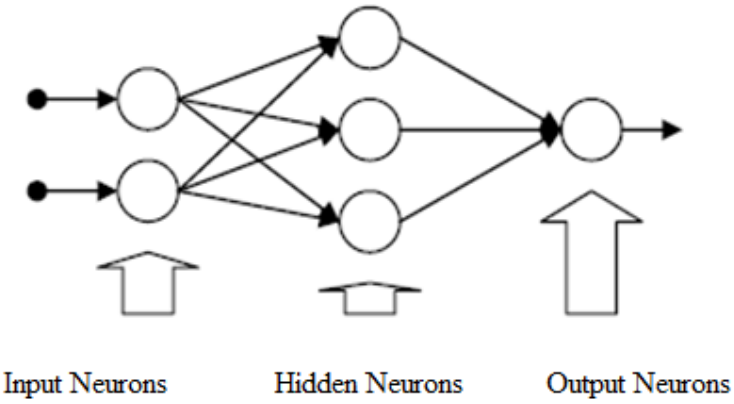


Figure 3.3. Typical Neural Network

The base unit of any neural network is the neuron (processor). Each unit or neuron has a number of inputs that are combined to produce a single output [37]. Model of computing neuron is illustrated in Figure 3.4. [35]. Each neuron is able to sum many inputs, whether these inputs are from a database or from other neurons, with each input modified by an adjustable weight w_i. The sum of these weighted inputs s is added to an adjustable threshold for the neuron and then passed through a modifying (transfer) function f that determines the final output y [34].

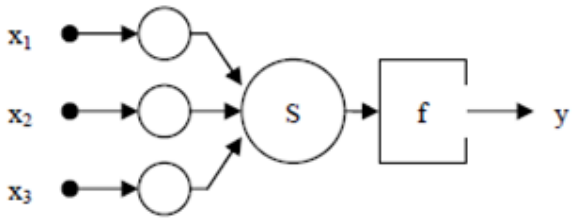


Figure 3.4. Model of Computing Neuron

In most real world problems, the neural network is never 100% correct. ANN are programmed to learn up to a given threshold of error.After the neural network learns up to the error threshold, the weight adaptation mechanism is turned off, and the net is tested on known cases it has not seen before. The application of the neural network to unseen cases gives the true error rate. In a well-trained neural network, the error threshold and the true error should be identical [5].

3.4. Naive Bayes Approach

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model".

In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

In spite of their naive design and apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix [38].

Abstractly, the probability model for a classifier is a conditional model

$$p(C|F_1,\ldots,F_n) \tag{3.8}$$

over a dependent class variable C with a small number of outcomes or *classes*, conditional on several feature variables F_1 through F_n. The problem is that if the number

of features *n* is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes' theorem, we write

$$p(C|F_1,\ldots,F_n) = \frac{p(C)\ p(F_1,\ldots,F_n|C)}{p(F_1,\ldots,F_n)}. \quad (3.9)$$

In plain English the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}. \quad (3.10)$$

3.5. Dynamic Bayesian Approach

Efficient algorithms exist that perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables (*e.g.* speech signals or protein sequences) are called dynamic bayesian networks**.** Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

Dynamic bayesian network is a Bayesian network that represents sequences of variables. These sequences are often time-series (for example, in speech recognition) or sequences of symbols (for example, protein sequences). The hidden Markov model and the Kalman filter can be considered as the simplest dynamic Bayesian networks [41].

3.5.1 Dynamic Bayesian Approach to Credit Risk Analysis

The Bayesian model is a result of formal encoding of the domain expert knowledge, the factors influencing the phenomenon and their mutual interactions.

Economic model being the objective of this study has been created outside the system. Its aim is to analyze the standing of the cuctomers from the point of view of its creditworthiness.

At first the primary goal is defining descriptive parameters of the model – the nodes of Bayesian network. At this stage, the domain expert has a crucial work to do. The topology of network and its detailed nature is then specified. Unspecified parameters, lacking or misdirected connections between nodes result in inconsistencies of model, ambiguity of causal relationships and finally in inaccuracy of expertise.

The process of network structure generation, given nodes, can be boosted using dedicated computer programs. Statistical dependencies between model parameters support the expert to eliminate redundant inter-node connections, preventing from non-convergence of propagation algorithms due to overloaded structure. Iterative generation process typically leads to computationally tractable structure.

The next step, quantitative specification of model, requires filling-up conditional probability tables. The computer aided, statistical analysis of historical data can reveal strength of causal dependencies between nodes. During all procedures the domain expert can moderate generation process. Figure 3.5 describe the nature of attributes which takes place in modelling. Among these nodes, dynamic bayesian approach proceeds with the sequence of these variables.

Two factors taken into account in our research for sequenced variable input: Reliability and Indebtedness. For reliability issue, “Previous Debts” are taken into account. For indebtedness, “Assets” and “Current Debts” are taken into account. Since our real-life ban dataset does not fit so much with the nature of dynamic bayesian approach, only correlated attributes are chosen for experiment purposes.

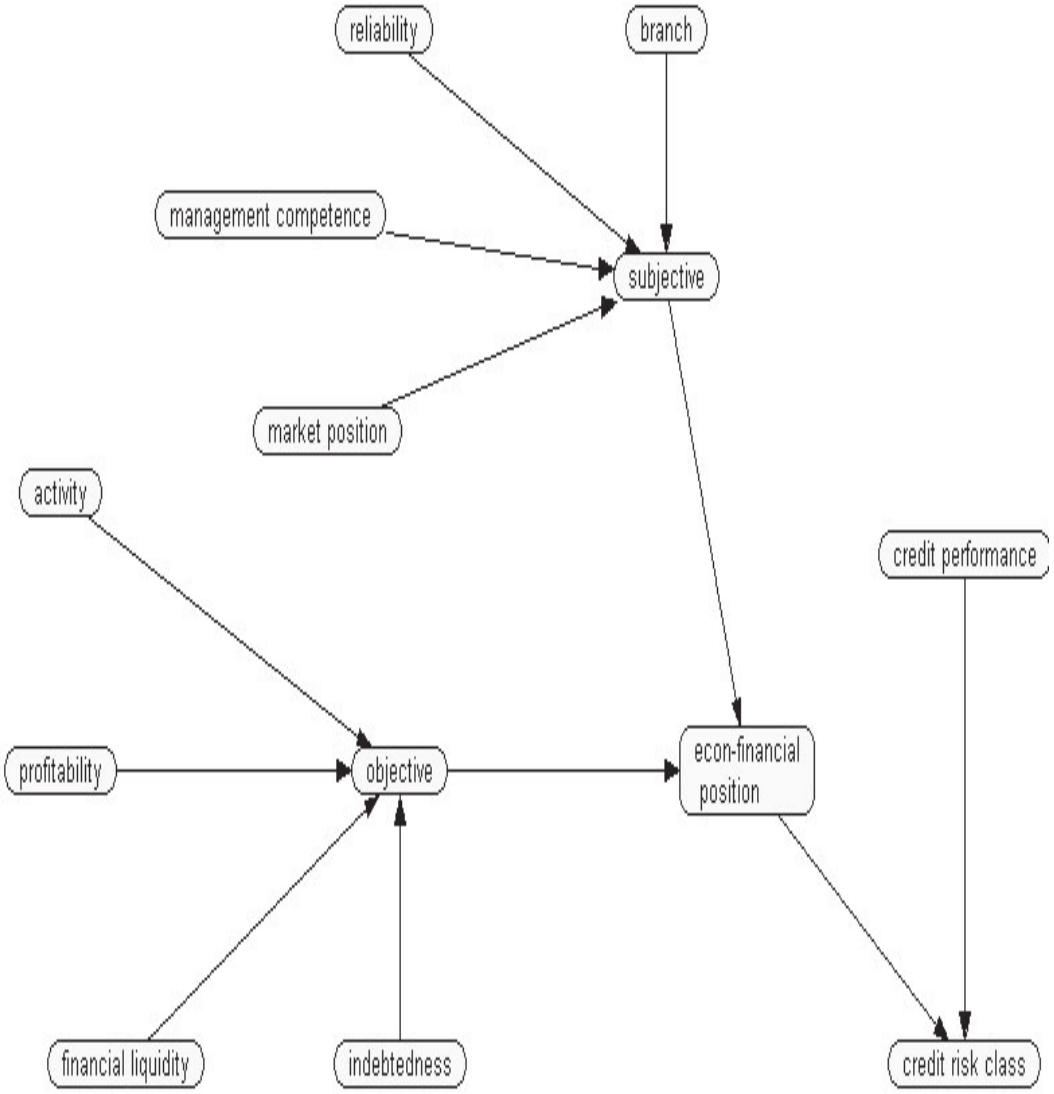


Figure 3.5 Modelling Dynamic Bayesian Structure

4. EXPERIMENTS & RESULTS

Our aim in this study is comparing performances of different data mining classifiers. By using three different datasets, experiments have been applied and outcomes were calculated. To make comprehensible comments on those outcomes, it is a crucial step to define performance metrics used in this thesis work. Since comparison of performances based on these metrics, without understanding them properly, it is impossible to see improvements for future work analysis.

4.1. Performance Metrics

Different performance metrics are used in this thesis study. Classification results are shown on confusion matrix. An example confusion matrix is shown on Figure 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.

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

Figure 4.1. Confusion Matrix

In our experiment, we use three measures: accuracy, precision, and Type I Accuracy to assess the effectiveness of classification algorithms. According to classification accuracy of the algorithms, one can evaluate how accurately an algorithm will classify future data, that is, data on which the algorithm has not been trained. Accuracy measures also help in the comparison of different classification models. Since it is worse to class a customer as "good" when they are "bad" (Type I Accuracy), than to class a customer as "bad" when they are "good" (Type II Accuracy), so we adopt Type I Accuracy rate as an important measure to assess different classification algorithms in credit evaluation.

$$Accuracy = \frac{t_pos + t_neg}{pos + neg} \tag{4.1}$$

$$\Pr ecision(pos) = \frac{t_pos}{t_pos + f_pos} \tag{4.2}$$

$$\Pr ecision(neg) = \frac{t_neg}{t_neg + f_neg} \tag{4.3}$$

$$Type\ \ I\ \ error\ \ rate = \frac{f_neg}{pos + neg} \tag{4.4}$$

Where *t_pos* is the number of true positives (*bad* customers that were correctly classified as such), *pos* is the number of positive (*bad*) samples, *t_neg* is the number of true negatives (*good* customers that were correctly classified as such), *neg* is the number of negative (*good*) samples, *f_neg* is the number of false negatives (*bad* customers that were incorrectly labeled as *good*).

Holdout and cross-validation are two common techniques for assessing classifier accuracy, based on randomly sampled partitions of the given data. In the holdout method, the given data are randomly divided into two independent sets, a training set and a test set. Typically two thirds of the data are allocated to the training sets, and the remaining one third is allocated to the test set. In *k*-fold cross-validation, the initial data are randomly partitioned into *k* mutually exclusive subsets or "folds," S_1, S_2... S_k, each of approximately equal size. Training and testing is performed *k* times. In iteration *i*, the

subset S_i is reserved as the test set, and the remaining subsets are collectively used to train the classifier. Both of these techniques have been applied with Logistic Regression and SVM classifiers with different subsets.

A receiver operating characteristics (ROC) graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC graphs have long been used in signal detection theory to depict the tradeoff between hit rates and false alarm rates of classifiers [39]. ROC analysis has been extended for use in visualizing and analyzing the behavior of diagnostic systems.

ROC graphs are two-dimensional graphs in which tp_rate is plotted on the Y axis and fp_rate is plotted on the X axis. An ROC graph depicts relative tradeoffs between benefits (true positives) and costs (false positives). Figure 4.2 shows an ROC graph with five classifiers labeled A through E. A discrete classifier is one that outputs only a class label. Each discrete classifier produces an (fp rate, tp rate) pair corresponding to a single point in ROC space. The classifiers in Figure 4.2 are all discrete classifiers. Several points in ROC space are important to note. The lower left point (0, 0) represents the strategy of never issuing a positive classification; such a classifier commits no false positive errors but also gains no true positives. The opposite strategy, of unconditionally issuing positive classifications, is represented by the upper right point (1, 1). The point (0, 1) represents perfect classification. D's performance is perfect as shown. Informally, one point in ROC space is better than another if it is to the northwest (tp rate is higher, fp rate is lower, or both) of the first. Classifiers appearing on the left-hand side of an ROC graph, near the X axis, may be thought of as ''conservative'': they make positive classifications only with strong evidence so they make few false positive errors, but they often have low true positive rates as well. Classifiers on the upper right-hand side of an ROC graph may be thought of as ''liberal'': they make positive classifications with weak evidence so they classify nearly all positives correctly, but they often have high false positive rates. In Figure 4.2, A is more conservative than B. Many real world domains are dominated by large numbers of negative instances, so performance in the far left-hand side of the ROC graph becomes more interesting.

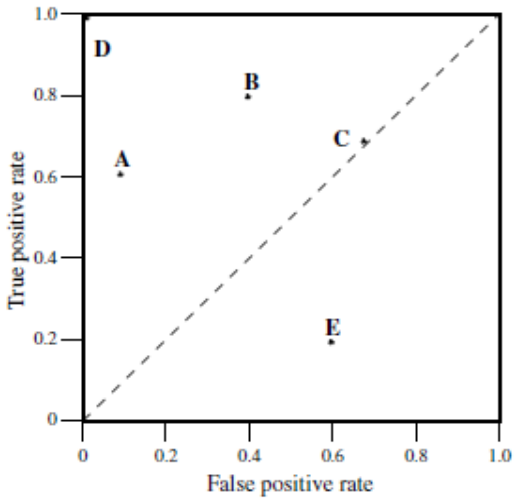


Figure 4.2. ROC Graph

By using following algorithm, ROC points have been generated for both negative and positive examples.

Algorithm: Efficient method for generating ROC points
Inputs: L, the set of test examples; f(i), the probabilistic classifier`s estimate that example i is positive; P and N, the number of positive and negative examples.
Outputs: R, a list if ROC points increasing by f_p rate.
Require: P > 0 and N > 0
L_{sorted} ← L sorted decreasing by f scores
FP ← TP ← 0
R ← 0
f_{prev} ← - ∞
i ← 1
while i ≤ | L_{sorted} | do
if f(i) ≠ f_{prev} then
push ($\frac{FP}{N}$, $\frac{TP}{P}$) onto R
f_{prev} ← f(i)
end if
if L_{sorted}[i] is a positive example then
TP ← TP + 1
else /*i is a negative example */
FP ← FP + 1
end if
i ← i + 1

end while

push ($\frac{FP}{N}$, $\frac{TP}{P}$) onto R /*This is (1,1) */

Figure 4.3. Generating ROC Points

ROC graphs are a very useful tool for visualizing and evaluating classifiers. They are able to provide a richer measure of classification performance than scalar measures such as accuracy, error rate or error cost. Because they decouple classifier performance from class skew and error costs, they have advantages over other evaluation measures such as precision-recall graphs and lift curves. However, as with any evaluation metric, using them wisely requires knowing their characteristics and limitations. It is hoped that this article advances the general knowledge about ROC graphs and helps to promote better evaluation practices in the pattern recognition community.

4.2. Experiments

Following sections include information about experiments of three different datasets with different classification techniques described in Chapter 3 of this study. Results of these experiments are evaluated in Section 4.3.

4.2.1. SVM on German Dataset

German Dataset is trained on WEKA software with Support Vector Machine. There are two steps in this experiment. At first step, dataset is originally used without any dimension reduction operation. At second step, dimension reduction is used by applying C4.5 decision tree on german dataset. Then reduced dataset is trained on WEKA software.

The other aspect of SVM experiment is related with kernel functions. When a nonlinear decision boundary is required, the hyperplane theory is maintained by the use of kernel functions. The original input space, X, is projected to another higher dimension dot product space F, called feature space, via a nonlinear map φ: $Rm \rightarrow Fd$ ($d>>m$), where linear separation may be possible. A linear separation in feature space corresponds to a nonlinear separation in the original input space. Kernels are usually employed when: (i) the input data belong to a space where they cannot be classified through hyperplanes; or (ii) the set containing the data does not have a structure of Euclidean space, so it is necessary to represent the data in a Euclidean space in which it is already possible to be classified [15]. In this experiment 3 different kernel functions are used with SVM. Those

are Linear Kernel, Polynomial Kernel and RBF Kernel functions. Following table, Table 4.1 shows results of SVM experiment on German Dataset.

Table 4.1. Experiment on German Dataset with SVM

	DATASET	Performance Measures			
		Accuracy	Precision-P	Precision-N	Type I Accuracy
SVM-Linear	Original	80.5	0.731	0.826	0.134
SVM-Polynomial	Original	78.4	0.681	0.815	0.142
SVM-RBF	Original	82.8	0.768	0.846	0.117
SVM-Linear	Reduced	81.6	0.752	0.835	0.127
SVM-Polynomial	Reduced	79.4	0.711	0.817	0.142
SVM-RBF	Reduced	83.2	0.777	0.849	0.115

4.2.2. Logistic Regression on Credit Approval Data

Second experiment takes place on special data mining software GeneXProTools 4.0. In this software, logistic regression was applied on credit approval data with different cross-validation numbers and results are recorded as in Table 4.2.

Table 4.2. Experiment on Credit Approval Dataset with Logistic Regression

LOGISTIC REGRESSION	DATASET	Performance Measures			
		Accuracy	Precision-P	Precision-N	Type I Accuracy
5-fold	Credit Approval Data	84,5	0.89	0.794	0.097
8-fold	Credit Approval Data	84,5	0.89	0.794	0.097
10-fold	Credit Approval Data	86,3	0.908	0.812	0.089

Following figures Figure 4.4, Figure 4.5 and Figure 4.6 shows the graphical distribution of different cross validation values. Figure 4.7 and Figure 4.8 shows the

logistic fit curve and ROC curve for 10-fold cross validation. Figure 4.9 represents the confusion matrix for 10-fold cross validation [42].

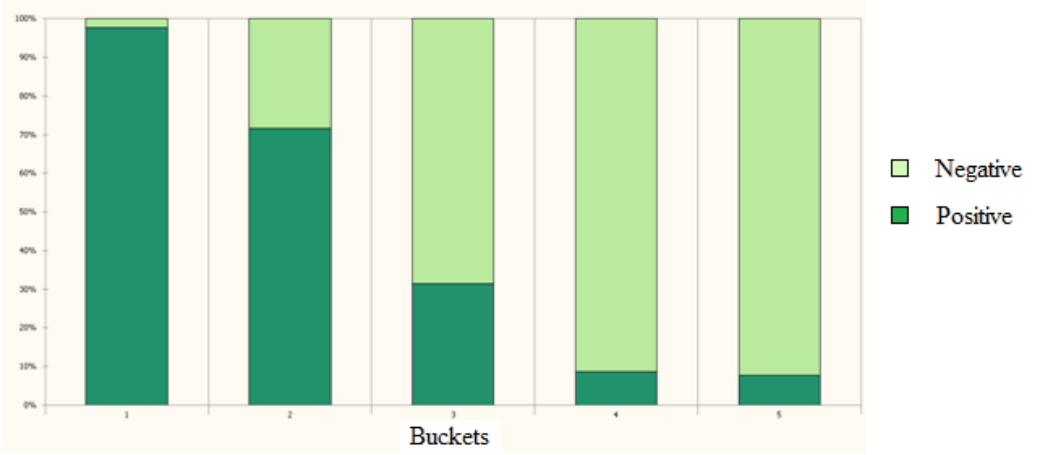


Figure 4.4. 5-fold Cross Validation

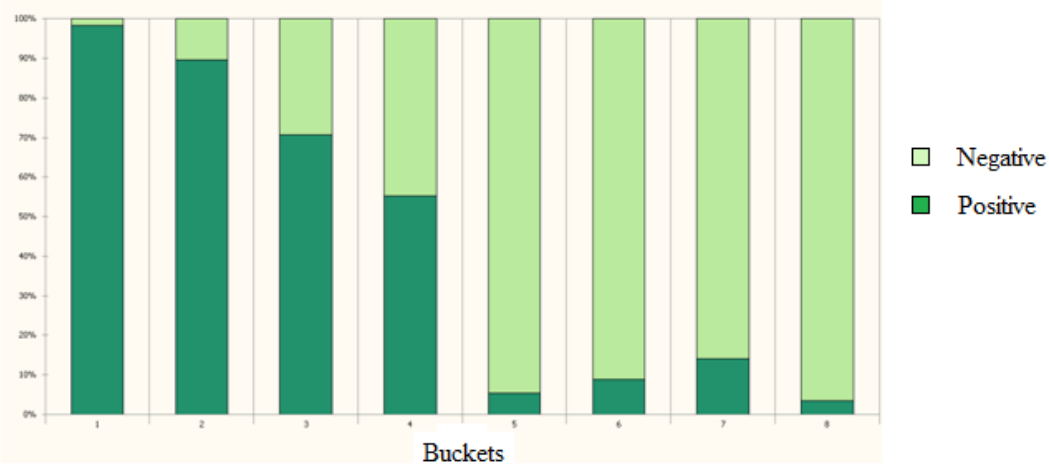


Figure 4.5. 8-fold Cross Validation

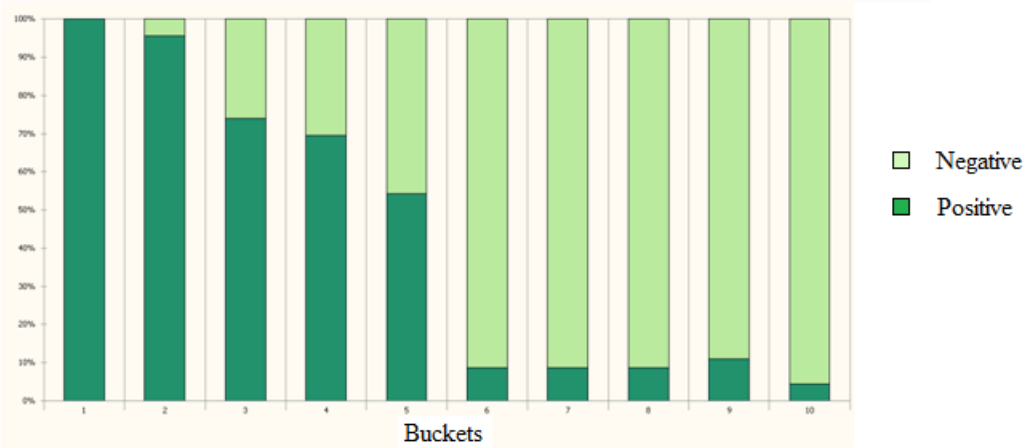


Figure 4.6. 10-fold Cross Validation

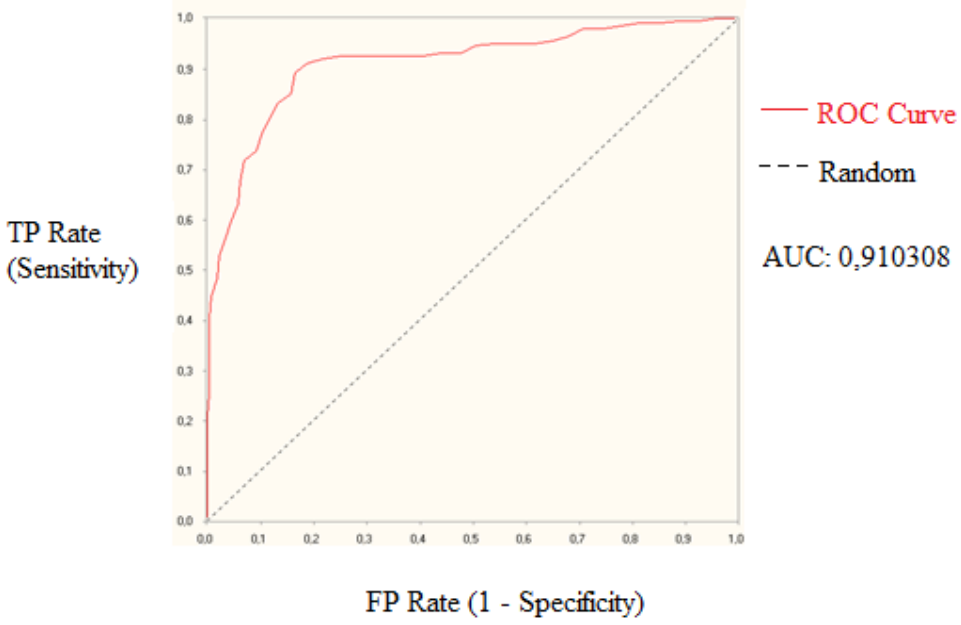


Figure 4.7. ROC Curve – 10-fold Cross Validation

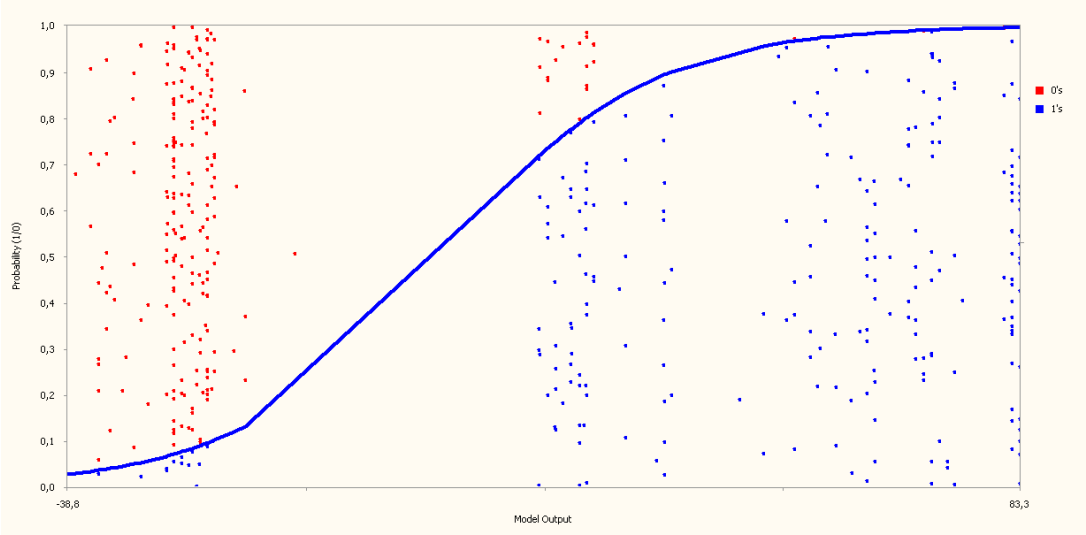


Figure 4.8. Logistic Fit Curve – 10-Fold Cross Validation

Table 4.3. Confusion Matrix for 10-Fold Cross Validation

	Yes (predicted)	No (predicted)	Total
Yes (actual)	40,00% (TP)	3,48% (FN)	43,48% (APC)
No (actual)	12,39% (FP)	44,13% (TN)	56,52% (ANC)
Total	52,39% (PP)	47,61% (NP)	100,00%

4.2.3 Experiments on Real-life Bank Dataset

Final experiments take place on Turkish Bank Dataset. As it was described before at proposed approach section, first two experiments are a basis fort he final experiment. From first experiment related to suppot vector machines, SVM – RBF is chosen after the dimension reduction operation. From second experiment, logistic regression with 10-fold cross validation is chosen. Both of these candidates are compared with neural networks, Naive Bayes approach and dynamic bayesian approach at this part. Among the performance comparison of this experiment, the proposed approach was developed by three of the classifier. Those classifiers are SVM, LR and Naive Bayes approach. They are gathered together as a unique classifier with different weights. Then proposed

approach was compared with other single classifiers as last experiment of the study to see if there is an improvement due to hybrid approach.

All experiments at this step are constructed on WEKA software. As in first experiment, dimension reduction operation has been applied as feature selection. C4.5 decision tree algoritm run on dataset and the total number of attributes are decreased from 25 to 12. All classification techniques are applied on this reduced dataset. Results are shown in Table 4.4.

Table 4.4. Experiment Results on Bank Dataset-1

	DATASET	Performance Measures			
		Accuracy	Precision-P	Precision-N	Type I Accuracy
SVM-RBF – 10-k	BankData	77	0.642	0.80	0.143
Logistic Regression – 10-k	BankData	75.6	0.620	0.801	0.149
Neural Network	BankData	71.2	0.53	0.78	0.151
Naive Bayes	BankData	75.4	0.617	0.80	0.150
Dynamic Bayesian Approach	BankData	72.0	0.56	0.79	0.150

Table 4.5 Experiments Results on Bank Dataset-2

	DATASET	Performance Measures			
		Accuracy	Precision-P	Precision-N	Type I Accuracy
SVM-RBF – 10-k	BankData	77	0.642	0.80	0.143
Logistic Regression – 10-k	BankData	75.6	0.620	0.801	0.149
Neural Network	BankData	71.2	0.53	0.78	0.151
Naive Bayes	BankData	75.4	0.617	0.80	0.150
Dynamic Bayesian Approach	BankData	72.0	0.56	0.79	0.150
Proposed Hybrid Approach	BankData	75.9	0.635	0.80	0.146

4.3 Results

Three experiments have been applied on three different datasets. First experiment is constructed on German Dataset with different kernel functions of support vector machines. Second experiment is constructed on a credit approval data with special software by using different cross-validation number with logistic regression. Final experiment has been applied on a real-life dataset with support vector machine – rbf kernel, logistic regression, neural network, naive bayes approach and dynamic bayesian approach. Then those single classifiers are compared with proposed hybrid classifier.

First experiment shows two important points. First one is related with the selection of kernel function. Choosing the most appropriate kernel highly depends on the problem at hand - and fine tuning its parameters can easily become a tedious and cumbersome task. The choice of a Kernel depends on the problem at hand because it depends on what we are trying to model. A polynomial kernel, for example, allows us to model feature conjunctions up to the order of the polynomial. Radial basis functions allow to pick out circles (or hyperspheres) - in constrast with the linear kernel, which allows only to pick out lines (or hyperplanes). The motivation behind the choice of a

particular kernel can be very intuitive and straightforward depending on what kind of information we are expecting to extract about the data [40]. First experiment shows this significant issue by trying different kernel functions. The best result was taken with RBF kernel function. Second aspect is related with importance of dimension reduction. It does not only have efficiency in terms of computation time but also helps to improve accuracy and optimality by targetting the most important attributes for classification.

Second experiment is related with different cross-validation numbers together with logistic regression. The goal of cross-validation is to estimate the expected level of fit of a model to a dataset that is independent of the data that were used to train the model. It can be used to estimate any quantitative measure of fit that is appropriate for the data and model. For example, for binary classification problems as in our study, each case in the validation set is either predicted correctly or incorrectly. In this situation the misclassification error rate can be used to summarize the fit, although other measures like positive predictive value could also be used. When the value being predicted is continuously distributed, the mean squared error, root mean squared error or median absolute deviation could be used to summarize the errors. This creates a greate advantage in comparison. At our experiment, 10-fold cross validation gives beter results among others. But there is a trade-off in this point. The computation time increase with increased k number. For small datasets, this can be negligible, but for huge datasets which consists millions of data, the computation time can be a significant problem.

Last experiment is the most important one, since it creates a basis for the future work of this study. SVM-RBF from experiment 1 and LR – 10k from experiment 2 has been chosen. To have a clear and appropriate comparison results, 3 more data mining techniques added to experiment. Those are neural network, naive bayes approach and dynamic bayesian approach. Results show that, SVM-RBF outperforms other classifiers in terms of accuracy. Logistic regression and naive bayes approach models also show very good results. They are so closed to SVM. But neural network and dynamic bayesian approach did not perform well in this experiment. One other bad issue for neural network is its computational time. Since the other classifiers give results in terms of seconds,

neural network classifier gives results in tens of minutes. When we think the dimension of our dataset, it has 1066 samples and this shows that neural network model can meet with a difficulty in terms of computation time when it has to deal with much more data. Also, the reason why dynamic bayesian approach could not perform well related with the nature of the dataset attributes. Dynamic bayesian approach could give better results with data which has time sequence variables. In other words, since our dataset attributes does not correlated each other and haven`t changed time to time, we could not get good performance results in terms of accuracy with dynamic bayesian approach.

We can see that the SVM algorithm obtained average classification accuracy 77%, That is mean the SVM algorithm is more stable and reliable than the LR algorithm for the credit data set. Furthermore, when the training set is a smaller size, such as training set take 10%, 20%, or 30% of total data set, the SVM algorithm got the highest classification accuracy among the four algorithms, so we have the reason to believe that the SVM algorithm is of higher generalization ability.

Results shows that SVM, LR and Naive Bayes classifiers could be used in a hybrid manner together. Those are gathered into one unique classifier as described in proposed approach. Simple weighting used for each of them and experiment on real-life bank dataset applied. When we look results, this proposed hybrid approach gives better results compared to most of the single classifier out of SVM. It has 75.9% accuracy performance and this is very close to SVM`s results. So it can be said that this hybrid approach could be a useful tool for classifying a customer in credit risk domain. However, it must be remembered that, dimension and attributes of a dataset plays major role for performance calculation and this approach may give different results for other types of data or very huge datasets.

5. CONCLUSION

In this study, 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 credit firms, which combined different data mining classifers inside. The main reason for combining these classifiers is integrating their robustness, generalization ability and the better interpretational capability, and applies them in individual credit risk evaluation. We aimed any little improvement in terms of accuracy and optimality which are vital aspects for finance sector. So small improvements may have seen unimportant but for this era, this means billions of dollars.

For a pre-process period, we aimed to maximize fittness of the classifiers which are used in this study. For that reason, first two experiments on public credit datasets are applied. First experiment helped us to select appropriate kernel function with support vector machine and second experiment shows us the appropriate cross-validation number for logistic regression. Since we aimed to combine the effectiveness of those classifiers, these steps are crucial and have a basis role.

Before applying proposed method, other important concept helped us to increase optimality of data and also decreased the computation time. Dimension reduction plays a major role since it focuses on most important attributes for our classification purposes. By applying it as feature selection, we get a new subset of original data with most relative attributes for decision making mechanism.

Proposed method based on two important concepts. To have a hybrid view on credit risk analysis, first thing is training of different classifiers on different subsets of the dataset with different optionalities. This study based on this approach. Second thing is combining those outcomes to have a robust classifier which outperforms the single classifiers. This mechanism provided after the first step.

Experiment shows that SVM outperforms other classifiers in terms of accuracy and optimality. Also LR and Naive Bayes approach has very closed results. Only NN and dynamic bayesian approach did not perform well. Results shows that SVM-RBF and LR with 10-fold cross validation, supported with dimension reduction, outperforms standard SVM and LR. So, it has been expected that combination of those classifiers also outperforms their single results. Proposed approach combined SVM, LR and Naive Bayes classifiers inside. Results of this approach shows that it is more effective then single clasifiers such as LR, NN, and dynamic bayesian networks. Only SVM outperforms hybrid approach since it has great generalization ability due to its greedy nature.

This approach could be tried with more different single classifiers. Also, this sudy only based on a credit risk data which has small sample size. Result may difer with huge size of datasets or with other context of data such as military or medical data. Future work of this study will be based on those two concepts.

REFERENCES

1. Joseph, C., *Credit Risk Analysis:A Tryst with Strategic Prudence*, Tata McGraw-Hill Education, USA, 2006.

2. Altmen, E., H. Fraydman, E. D. Kao, "Introducing recursive partitioning for financial classification:the case of financial distress", *Banking and Finance*, Vol. 40, pp. 269-291, 1985.

3. Yu, H., C. X. Wan, "Knowledge Management in e-commerce: A data mining perspective", *Proceedings of International Conference on Management of e-commerce and e-government*, pp. 152-155, 2009.

4. Gallo, C., C, Letizia, G. Stazio, "Artificial Neural Networks in Financial Modelling", University of Foggia Press, Foggia, 2006.

5. Shachmurave, Y. "Applying Artificial Neural Networks to Business Economics & Finance", University of Pennsylvania Press, Pennsylvania, 2002.

6. Sotchidananda, S. S., "Comparing the Efficacy of the Decision Trees with Logistic Regression for Credit Risk Analysis", *Head Risk Solutions & Research*, Publications of HP India, 2006.

7. Hong, Y., X. Huang, X. Hu, H. Cai, "A Comparative Study on Data Mining Algorithms for Individual Credit Risk Evaluation", *Management of e-Commerce and e-Government (ICMeCG), Fourth International Conference at China*, pp. 35-38, 2010.

8. Yangqioo, W., W. Shauyang, K. K. Lei, "A New Fuzzy SVM to Evaluate Credit Risk", *IEEE Transactions on Fuzzy Systems*, Vol. 13, No. 6, 2005.

9. Dei-ti, H., L. Min-Shiu, L. Tsai, "A New SVM with Fuzzy Hyper-Plane and its application to evaluate Credit Risk", *Intelligent Systems Design and Applications*, *ISDA '08 Eighth International Conference*, 2008.

10. Zang, Y., M. A. Orgun, R. Baxter, W. Lin. "An application of element oriented analysis based credit scoring", *Industrial Conference on Data Mining*, Australia, 2010.

11. Huang, Z., H. Chan, C. Hsu, W. Chan, S. Wu, "Credit Rating Analysis with Support Vector Machines & Neural Networks: A Market Comparative Study", The University of Arizona Press, Arizona, 2003.

12. Doumpos, M., C. Zopounidis, "Model combination for credit risk assessment:A stacked generalization approach", University of Crete Press, Greece , 2006.

13. Gaganiz, C., F. Pasioures, C. Spathis, C. Zopounidis. "A comparison of nearest neighbors, discriminant and Logit Models for Auditing Decisions", *Intelligent Systems in Accounting, Finance & Management*, Vol. 15, No. 2, pp. 23-40, 2007.

14. Kaya, M. E., F. Gürgen, N. Okay. "An Analysis of Support Vector Machines for Credit Risk Modeling", *PAKDD 2007 DMBiz Workshop*, China, 2008.

15. Campos, R., F. J. Ruiz, N. Agell, C. Angulo, "Financial Credit Risk Measurement Prediction using innovatiove soft-computing techniques", Wessex Institute of Technology Press, Wessex, 2004.

16. Kotsiantis, S., "Credit Risk Analysis Using a Hybrid Data Mining Model", *International Journal of Intelligent Systems Technologies and Applications,* Vol. 2, No. 4, pp. 345 - 356*,* 2007.

17. *Bank for International* Settlements, 1999, http://www.bis.org, accessed at March 2009.

18. *Principles for the Measurement of Credit Risk* , 1999, http://www.bis.org, accessed at March 2010.

19. *German Dataset - Statlog Project Databases*, Prof. Hans Hofmann, ftp://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/german, accessed at March 2010.

20. *Credit Approval Data*, GepSOFT Co., 2011, http://www.gepsoft.com/, accessed at April 2011.

21. Private Bank Dataset, SoftTech Co., 2010, http://www.softtech.com/, accessed at May 2010.

22. Cunningham, P., "Dimension Reduction", University College Dublin Press, Dublin, 2007.

23. West, M., "Bayesian factor regression models in the "Large p, small n" paradigm", *Bayesian Statistics*, *Proceedings of the 7th Valencia Internetional Meetings*, Vol. 7, No. 2, pp. 23-73, 2003.

24. Bellman, R., "Adaptive Control Processes: A Guided Tour", Princeton University Press, Princeton, 1961.

25. John, G. H., R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem", *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 121-129, 1994.

26. Keller, D. and M. Sehami, "Toward Optimal Feature Selection", *Proceedings of International Conference on Machine Learning*, 1996.

27. Kira, K. and L. A. Rendall, "The feature selection problem: Traditional Methods and a new algorithm", *Proceedings of Ninth National Conference on Artificial Intelligence*, pp. 129-134, 1992.

28. Narendra, P. M. and K. Fukunaga, "A Branch and Bound Algorithm for feature selection", *IEEE Transtactions on Computers*, Vol.26, No. 9, pp. 917-922, 2004.

29. Beryor, 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.

30. Quinlan J. R., *Programs for Machine Learning,* Morgan Kauffman Publishers Inc., California, 1993.

31. "Logistic Regression", 2001, http://www.statgun.com/tutorials/logistic-regression.html, accessed at March 2008.

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

33. Shunin, Y. N., "Neural Networks Modelling of Business Situations And Decision-Making Analysis" , *Computer Modelling and New Technologies*, Vol. 9, No. 7, pp. 17-26, 2005.

34. Tal, B., "Background Information on Neural Network Based System of Leading Indicators", *Economics & Strategy*, Vol.14, No. 1, pp. 1-4, 2003.

35. Handzic, M., F. Tjondrowibawe, J. Yeo, "How Neural Networks Can Help Loan Officer to Make Better Informed Application Decisions", *Informing Science InSITE: Where Parallels Intersect*, Vol.27, No. 2, pp. 97-108, 2003.

36. Thonabauer, G., "Rating Models and Validation: Guidelines on Credit Risk Management", 2004, http://www.fma.gv.at, accessed at March 2010.

37. Nguyen, H. G., "Using Neural Network in Predicting Corporate Failure", *Journal of Social Sciences*, Vol. 4, No.7, pp. 199-202, 2005.

38. Webb, G. I., J. Boughton, and Z. Wang, *Not So Naive Bayes:Aggregating One-Dependence Estimators,* Monash University Press, Melbourne, 2005.

39. Fawcett, T., "An Introduction to ROC Analysis", *Pattern Recognition Letters*, Institute For the Study of Learning & Expertise, Palo Alto, USA, 2005.

40. "Kernel Functions for SVM" 2010, http://www.crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html, accessed at March 2010.

41. Ghahramani, Z., "Learning Dynamic Bayesian Networks*", Lecture Notes In Computer Science*, Vol. 1387, No. 4, pp. 168-197, 1997.

42. "Credit Risk Assesment", 2011, http://www.gepsoft.com, accessed at April 2011.