DECISION SCIENCES INSTITUTE

A Decision-Tree-Based Classifier for Credit Assessment Problems under a Big Data Environment

Ching-Chin Chern
National Taiwan University
Email: cchern@ntu.edu.tw

Weng U Lei
National Taiwan University
Email: r01725019@ntu.edu.tw

Shu-Yi Chen
Ming Chuang University
Email: maxchen@mail.mcu.edu.tw

ABSTRACT

This study proposes a Decision-Tree Credit Assessment Approach (DTCAA) to solve the credit assessment problem under a big data environment. Decision tree model is selected because of its interpretability and easily understanding rules, as well as its competitive performance capability. By acquiring a large-volume dataset from one of the biggest car collateral loan companies in Taiwan, the efficiency and the validity of DTCAA are verified through several experiments.

KEYWORDS: Credit Assessment, Decision Tree, Big Data, Data Mining, Data Pre-processing, Record linkage

INTRODUCTION

Credit assessment refers to the process of recognizing potential risks and excluding them from the system when a creditor investigates an applicant of a customer credit loan. The credit assessment problems can be treated as a supervised “classification” problem, which refers to a problem of building a classification model (or function) from pre-classified historical data. The “classification” algorithm first obtained input historical data and corresponding pre-defined output value (target value) from the dataset, and then built a model (classifier) that is able to categorize the training data, which in turn was used to infer an appropriate class for a future incoming applicant.

In the past, finance institutes has adopted different data mining and machine learning algorithms to detect rapidly changed fraudulent patterns or identify specific risky segments. Based on the accumulated application data, finance institutes build their own credit assessment models and establish related knowledge. The methodology based on different data mining algorithms, such as SVM (Chen & Li, 2010), decision tree (Wang et al, 2012), logistic regression (Siddiqi, 2005; Yap et al, 2011), etc. builds different predictive models.

However, rather strict assumptions of using the logistic regression model make the adoption of a logistic-regression credit scorecard model very difficult if not inaccurate for violating the assumptions (Hosmer, 2013). Moreover, when the new coming data causes concept shifting, a

whole new logistic model has to be constructed to ensure its usability, which incurred huge computation efforts. Finally, the interpretation of a logistic-regression credit score is not intuitive and the conversion from the score to its actual meaning is a complicated and struggling process.

Considering various credit assessment applications and crucial information derived from cumulated data (such as the applications in the past three years, etc.), researches used conventional data mining techniques as tools to conduct credit scoring (Chen & Li, 2010). The problem of the aforementioned data mining techniques is that their running time is satisfied only when a relatively small dataset was introduced in the process. Previous researches showed that even for a small dataset, the parameters significantly affected the running times of conventional data mining techniques (Sahin & Duman, 2011). While data mining techniques usually assume that the pattern is consistent over time, the actual situation is that fraudsters change their strategies to commit the frauds and constant changes of external environment also cause inconsistency in the accumulated historical data.

Taking the advantages of the well-structured and relatively simple computational properties, this study will focus on modifying a decision tree model to solve the credit assessment problem under a big data environment by considering different perspectives. The objective is to apply this modified big-data decision tree approach to a real dataset obtained from one of the biggest car collateral loan companies in Taiwan to help them solving their credit assessment problem.

The rest of the paper is organized as follows. Section 2 describes the problem. Section 3 presents our decision-tree-rule-based credit assessment approach (DTCAA) to solve the credit assessment problem under a big data environment efficiently. Section 4 compares the results obtained with our heuristic algorithm to those obtained with other heuristic methods, in order to evaluate the DTCAA's efficiency and optimality. Finally, section 5 offers our conclusions.

PROBLEM DESCRIPTION

Credit assessment for a consumer credit loan application in a finance institute calculates the creditworthiness of an individual's ability to honor his or her financial obligation. The credit assessment result of a consumer credit loan application is a class assigned to the applicant. For example, after the credit assessment process, an applicant may be classified as 'AA+', meaning excellent credit record without any possibility of default, or as 'CC–', meaning the worst credit record with a very high possibility of default. Hence, credit assessment is performed to obtain the class of an applicant whenever a new consumer credit loan application is submitted. The objective of this study is to build a credit assessment model that can calculate the creditworthiness of an applicant through evaluating an applicant's personal information and historical payment records.

For the credit assessment problem considered in this study, three types of target variables are defined as Y_1: binary form, Y_2: multi-classes and Y_3: continuous form. For instance, there are six different classes in Y_2, included (1) three good-credit classes and (2) three bad-credit classes. The three good-credit classes and three bad-credit classes are defined as following:

1. Applicants who are approved and will pay off the loan without being late;
2. Applicants who are approved and will pay off the loan with early payment;
3. Applicants who are approved but will not take their loan offers;
4. Applicants who are approved and will pay off the loans with some late payment records;

5. Applicants who are denied the loan applications;
6. Applicants who will default on their loans if they are approved.

Before describing the detail of a decision tree classifier (Quinlan, 1986; Chang & Sheng, 2008), we need the following definitions.

Definition 1: The training data set T has m distinct attributes A_1, A_2, ..., A_m and one target variable Y_j and n is the number of training objects, $\{x_1, x_2, \ldots, x_n\}$, in T.

Definition 2: Each training object in T consists of the values of the attributes and one target variable. The training object t can be described as $<(A_1, a_1), (A_2, a_2), \ldots,(A_m, a_m), y_j>$, where a_i is the value of A_i in t and y_j is the class of t.

In a credit assessment problem, attributes refers to the applicants' personal and demographic information provided when they submit the loan applications. Their preceding financial situations and abilities of payment will be checked through a central financial institution, responsible to gather credit records from all banking or financial companies in a country. In the end, the result provides to the finance institute a detailed history of the applicant's general demand on loan and credit class (score).

Learning from data is feasible only when the features of the data are distinguishable. Therefore, feature selection and feature extraction are applied on the data to enhance classification performance. Feature selection refers to a process of selecting representative attributes. Feature extraction, on the other hand, transforms the original attributes to another form. These techniques are used as dimension reduction method. After the preparation steps, the data are ready for a learning algorithm to generate prediction models.

The goal of a learning algorithm is to minimize the classification error. For example, a binary classification algorithm generates its model from a dataset; this model will generate predictions, denoted as y_j^, such that the error will be least compared to y_j. A common error measurement of a learning model is to verify its misclassification rate, which is the count of false positive and false negative divided by number of records of the dataset.

Another important criterion regarding to the classification result is sensitivity. Sensitivity is defined as true positive (TP) divided by the sum of true positive (TP) and false negative (FN). The denominator of sensitivity, (TP + FN), is the actual positive case. Sensitivity indicates an ability to correctly identify the positive case of a model. The higher the sensitivity, the less Type-II error is occurs by applying the model. Lower sensitivity suggests a poor performance in identifying bad customer.

Decision tree is an algorithm that classified records by conjunctive rules. There are several paradigms in decision tree classifiers. ID3 and C4.5 are some of the classifiers that applied information theory to separate data. It iteratively calculated the entropy of a set T and compared the information gain from a split.

An entropy, denoted as $H(T)$, was defined as $H(T) = -\Sigma p(y)\log(y)$, where T is the set, Y is a set of target classes in T, y belongs to Y, $p(y)$ is the probability of object x_i in T classified as y. Information gain is the amount of uncertainty reduced because of the split. Define $IG(A) = H(T) - \Sigma p(i)\log(i)$, where A is an attribute which the split is happened, I is a subset split from set T at attribute A. Therefore, a decision tree will select a locally best attribute (i.e. highest information

gain) as a splitting criterion. This step requires $O(n^2)$ of time because every variable has to be checked in determining the best one. By comparing information gains of all attributes, a decision tree will select an attribute with the maximum information gain as the node to split the data into two or more subsets. The procedure iteratively proceeds until a full tree is developed.

Pruning might be necessary after a tree is built because of overfitting problems caused by a decision tree with too many levels or too many nodes. Therefore, pruning the tree is a way to prevent a decision tree to be constructed with a high complexity. There are two ways to avoid an oversized tree: pre-pruning approach controls the number of a tree's nodes before the tree is built; post-pruning approach evaluates and finds the least important sub-trees to prune after the decision tree is built. Mahmood et al. (Mahmood et al, 2010) have proposed various pruning heuristics for comparisons. In our study, limiting the number of objects at every leaf node, a simple yet effective pruning method, is applied at pruning step under a big data environment.

To build models without overfitting problems, different approaches are adopted and considered. Different models are evaluated by the test data that are not used in model training to ensure that the evaluation can be inferred to the reality. Therefore, validation data are used to check if the model's misclassification rates meet the requirement and validation technique also considers the probability the worst case when a model's complexity is high. In addition, *k*-fold validation is also widely used, which divides a dataset into *k* subsets and takes *k*-1 subsets as training dataset with the remaining one as the validation dataset. Therefore, the model will be trained *k* times and each iteration used the *i*-th subset one at a time. While conventional data mining technique uses *k*-fold validation as a tool to enhance data usage, it is not appropriate under a big data environment, since it is already a time-consuming process through only one scan. The computational time of *k*-fold validation under a big data environment will be intolerable.

As data are generating from various sources and channels, analytics can benefit from consolidating them into a single table. The problem is that it takes only a few seconds to collect and store many gigabytes of raw data but it takes days, if not months, to extract information from the massive data. This is also known as the "Volume" property of big data environment (McAfee & Brynjolfsson, 2012). To handle a huge volume of data, some tried to solve it using a distributed system, but the problem still exists as tremendous data is already occupied in the distributed system.

Another issue is that a dataset changes in both vertical and horizontal aspects. Vertical change refers to the velocity of new data generation, while horizontal means that attributes of a dataset is not static, with addition of new attributes or removal of the existing ones (McAfee & Brynjolfsson, 2012). Taking a house mortgage application as an example, there are enormous historical application data stored in a bank's database, including the applicants' demographic and credit information, records of payments and times of defaults, which are accumulated through the years with multiple dimensions. These data can hardly be analyzed since millions of records are multiplied by hundreds of attributes, which exceed the capability of any analysis tool.

Credit assessment problem can leverage from data mining methods to generate a predictive solution. However, when building a decision tree under a big data environment, there are several problems from various aspects of the credit assessment issue. In order to efficiently and effectively analyze the big data, the proposed algorithm must be able to retrieve the most representative data as soon as possible.

THE DECISION-TREE CREDIT ASSESSMENT APPROACH (DTCAA)

Since this study aims at solving the credit assessment problem under a big data environment, it is unrealistic to use all the attributes and all the objects to generate the decision-tree-based classifier. This study extract the most discriminative attributes as nodes to partition data before generating the classifier. Once a classifier is built, prediction is conducted for the future incoming data to be assigned classes (scores). To address the aforementioned issues, we propose a decision tree data mining approach, called the Decision Tree Credit Assessment Approach (DTCAA) with the following three major steps:

Step 1: Data Analysis and Preprocessing;

Step 2: Decision Tree Model Building; and

Step 3: Prediction and Scoring.

Raw data are collected from different sources and integrated into a single dataset for further usage in Step 1. In order to validate the performance independently, DTCAA also partitions the dataset into two separated subsets in Step 1. In the second step, DTCAA establishes decision tree models from the dataset produced by Step 1, validates the models to prevent them from overfitting and selects the best decision tree model for future prediction.

Finally, DTCAA predicts and scores the upcoming data by using the previously chosen model in the Step 2. DTCAA calculates a probability for each class at the leaf node and selects the maximum among different classes as the prediction result. Therefore, we can treat the probability as a score assigned to the predicted class. For example, when the decision tree model is used to classify an applicant, DTCAA will generate the final predicted class with a probability of defaulting, as well as a specific rule to classify this record. The rule provides a better understanding about the class assigned to an individual than a score with the same coefficients of the variables without differentiating applicants, yet it still accompanies a probability of defaulting as a score. In other words, DTCAA is designed to deal with the credit assessment problem, which needs rules and scores to make the consumer credit load decision.

In order to build an effective decision tree model, this step will consolidate different types of raw data and transform them into a suitable form to be used in building a decision tree. Since the study treat a credit assessment problem as a classification problem, DTCAA defines a target variable first and then identifies different groups of tables, attributes and objects that can be consolidated for the data mining purposes. Different consolidation techniques will be applied so that a concise and coherent dataset can be used in the next step. After performing these three consolidations, we can proceed with partitioning the dataset into two separated subsets.

Data consolidation refers to a process that integrates data from different sources into a single dataset. This process is required since there are different types of data stored in different databases, e.g. customer personal information, application form, customers' transactional history, etc. These databases provide information with different dimensions and are critical in decision-makings. However, when integrating the data, a great deal of human expert involvement is required in the consolidation process, which is the source of human-factored error. As a result, several techniques are adopted to generate an appropriate dataset.

The necessity of attribute consolidations is related to the "Volume" property of big data. Due to the direction shift of financial policy or system design, attributes with the same meaning but

created at different stages appear as different attributes even in the same dataset. It happens as well when the meanings or the representation codes of some particular attributes have changed over time. Moreover, highly correlated attributes would also be an issue, especially in a regression model due to the multicollinearity issue among the independent variables.

The object consolidation exists because of data inconsistency or redundancy. For example, multiple queries for an applicant's credit record are saved in the dataset, in which all of the queries are in similar formats but with different values because they are from different finance institutes. To deal with the data inconsistency or redundancy issue, object consolidation or replacement is necessary to generate a concise and coherent dataset for credit assessment.

To deal with definition inconsistency among attributes or data inconsistency, human expert involvement may become the only choice. Integrating the attributes with the same meaning is feasible only when the meaning of representation codes is consistent within multiple versions of data. For data inconsistency of the credit assessment problem considered in this study, we applied the same techniques mentioned previously, such as combining different objects by their defined value in the same attribute.

We then partition the dataset constructed in the previous steps into a training dataset and a validation dataset. The former is used to build a scoring model or a classifier using different data mining techniques, and the latter is used to validate different parameters/models and choose the most suitable one. After this step, we can adopt different data mining techniques by using the same training dataset to build different classifiers.

Next, a decision tree classifier is built based on the training dataset and validate the classifier using the validation dataset. However, the credit assessment problem considered in this study usually involves hundreds of attributes, which are not all essential to identify characteristics of the target variable. Furthermore, correlations among attributes cause multicollinearity and inaccuracy in some data-mining models. Before a decision tree classifier is built, an attribute selection mechanism selects most discriminative attributes to build a classifier so that time complexity is significantly decreased without sacrificing the accuracy.

There are many attribute selection mechanisms in some data mining process. Weight Of Evidence (*WOE*) with Information Value (*IV*), one of the attribute selection methods, measures the ability of separating bad customers from good ones among different classes in an attribute (Siddiqi, 2005). The *WOE* of a class is formulated as $WOE_{class=i}$ = ln(Relative Frequency of "Good" in class *i* / Relative Frequency of "Bad" in class *i*) $\times$ 100% and Information Value (*IV*) of an attribute is formulated as $IV = \Sigma_i$[(% of "Good"$_i$ – % of "Bad"$_i$) $\times$ ln(Relative Frequency of "Good" in class *i* / Relative Frequency of "Bad" in class *i*)].

In order to make sure that the classifier is usable, a rather loose criterion, 70% accuracy, is used to filter inappropriate models. After that, we generate different models using different parameters, such as the minimum number of leaf nodes, different attributes, etc., to determine the best classifier. Finally, we compare different models using other criteria to confirm that a final model is appropriate in multiple dimensions.
To build a decision tree, two main issues remain: first to choose the appropriate attribute to be the node of splitting and second to determine the criteria of splitting the node. For the first issue, many mechanisms, such as C4.5, Classification and Regression Trees **(**CART), etc., calculate all the possibilities at different splitting points at one time and determine the maximal probability of the locally best attribute as the corresponding splitting point.

For the second issue of node splitting criteria, some problems arise if the class distribution of the target variable is not balanced. Therefore, a skewed-insensitive splitting mechanism should be adopted when there are imbalance situations. A splitting criterion called Hellinger Distance Decision Tree (HDDT), which is based on a measurement of distributional divergence, is proposed by Cieslak and Chawla (Cieslak & Chawla, 2008).

The Hellinger Distance is proved to be useful for its skewed-insensitive property and its better performance than the traditional decision tree algorithm, like C4.5 or CART, under imbalance situations. Therefore, The Hellinger Distance is applied in this study to construct decision tree for the imbalance credit assessment dataset.

An appropriate model should be able to consistently perform, and have powerful ability to predict the future dataset. To identify performance among different data mining models, there are different perspectives and criteria. Validation techniques can be used to evaluate the generalization performance of models. As a validation subset is built, evaluation will be applied here using the validation dataset. If the error between training data and validation data is high, overfitting should be taken into account.

Another issue is their predictive ability. In general, misclassification rate is widely used in evaluating classification models. The problem of misclassification rate, however, is that it could not present its Type-I / Type-II error. For the credit assessment problem in this study, Type-II error is crucial since the company would pay a lot, when the model misidentified a customer who is risky to default as a good customer. Sensitivity is therefore applied in the evaluation process.

COMPUTATIONAL ANALYSIS

In order to demonstrate the applicability of DTCAA on a real-world credit assessment problem, a dataset is acquired from one of the three biggest car collateral loan companies in Taiwan, whose business is mainly focusing on secured customer credit loan by using cars as collaterals. The company provides multiple tables, including application data (e.g. interest rate, collateral value, periods of payment), applicants' personal information (e.g. demographic information, working condition), cosigners' information, applicants' credit condition and transactional data (e.g. payment situations, period of being paid, stability) during the period from 2007 to 2013.

According to the previous discussion in Section 2, there are three types of definitions on the target variable (Y_1, Y_2, Y_3), representing the defaulted behaviors of applicants. After a long preprocessing step, 68 different attributes have been generated from the raw database. On the other hand, there are nearly 180,000 objects (applications) extracted into dataset CAT_B. Through the preprocessing step of DTCAA, the sizes of the datasets are reduced to about 100MB in CAT_B, compared to 3GB, the size of the original data set.

The experiments are conducted on a desktop computer with 16GB RAM, Intel® Core™ i7-2600 3.40GHz CPU and running Windows® 7 Enterprise. WEKA [10] is a well-known and open-sourced data mining software written in Java and provides numerous machine-learning algorithms, variable selection mechanism, etc., which will be applied in the data mining process. However, some variable selection mechanisms and data mining algorithms, particularly *WOE*/*IV* method and HDDT, are not included in WEKA. This study, therefore, modifies HDDT provided

from the Cieslak and Chawla's website (Cieslak & Chawla, 2008), calculates *WOE*/*IV* by a series of SQL commands and incorporates both into WEKA.

The experiments compare the results of four data mining methods, including Decision Tree (HDDT and C4.5), Logistic Regression and Artificial Neural Network. Since different parameters are needed for these data mining models, their corresponding settings must be different. For Decision Tree (HDDT and C4.5), we need to determine one parameter, the number of objects at every leaf node when performing pre-pruning to the decision tree. Since there is not optimal setting for this parameter, we test and compare the results of five different settings: 2, 5, 50, 100 and 1000. The more objects at each leaf node, the smaller and the less accurate the final tree is. On the other hand, the Logistic Regression and ANN are performed using the WEKA's (Hall et al, 2009) default settings and will automatically search the best setting for the final model.

As mentioned in Section 2, three important metrics are used to compare the experiment results: misclassification rate, sensitivity and training time. These metrics reflect usability and accuracy of a model. In additions, the interpretability of the result is also a crucial criterion when applying different data mining methods in practice and thus will be included in discussion.

The experiments show that the misclassification rate is improved a little whenever an attribute selection mechanism is adopted. However, these attribute selection mechanisms do not outperform each other. It all depends on the combinations of the attribute selection mechanisms and their corresponding parameters. For example, *WOE*/*IV* with *IV*>0.05 performs better with HDDT than with C4.5 while on the opposite, *WOE*/*IV* with *IV*>0.01 performs better with C4.5 than with HDDT. For logistic regression (LR), however, the performance of *WOE*/*IV* with *IV*>0.1 is better than *WOE*/*IV* with *IV*>0.05. On the contrary, sensitivities decrease consistently among various models when *WOE*/*IV* is applied. The results also show that information gain is a good attribute selection mechanism in terms of misclassification and sensitivity.

Among the four data mining models, the experiments indicate that, under various scenarios, decision tree model provides a better result than LR and ANN. The misclassification rates, as well as sensitivities, are close among HDDT, C4.5, LR and ANN in the same scenario. The training times of HDDT and C4.5 are less than those of LR and ANN. In fact, the logistic regression model, a popular credit scoring solution, does not outperform decision tree (HDDT, C4.5). Against the common believe that decision tree performs badly, decision tree actually generates a competitive prediction model in this case. Furthermore, the complicated processes in attribute selection, transformation and interpretation of logistic regression model make it less favorable to the users. The ANN, moreover, is not a suitable model in solving the credit assessment problem due to its black-boxed property and long training time.

The choice of parameters for decision tree (HDDT or C4.5) in this study focuses on the number of objects at a leaf node. The experiments suggest that a 100-object leaf node to be the best choice for HDDT while a fifty-object leaf node for C4.5 outperforms the other candidates. In general, the experiments suggest a fifty-object leaf node to be a better choice among the five candidates for both HDDT and C4.5.

The observations of the results suggest that decision tree with 50 objects at a leaf node provides competitive accuracies, very short training time, as well as easily understandable rules. The variable selection mechanisms help data mining methods to reduce the training times greatly, whereas they also reduce the complexities of data. However, the sensitivities among

different models and scenarios are relatively low to identify most of the risky customers due to the noises in the data source.

In conclusion, the results have shown that DTCCA can be treated as a framework in solving a credit assessment problem. In this section, we have designed the experiments based on three factors, target variables, decomposition of multi-class method and variable selection mechanism. Four data mining models are applied to verify whether the proposed HDDT is a competitive candidate to split nodes in a decision tree approach. For a decision tree approach, we also test five different numbers of objects at each leaf node to determine a suitable parameter in practice.

CONCLUSIONS

This study proposes a Decision-Tree-Based Credit Assessment Approach (DTCAA), a framework for solving a credit assessment problem under a big data environment by three steps. First, a series of data preprocessing procedures is applied in order to generate a complete dataset from different data sources, which is considered the biggest challenge under a big data environment. Second, HDDT is adopted to generate a decision tree model. Finally, the resulted decision tree is in turn used to predict and score a new application.

A dataset is acquired from one of the three biggest car collateral loan companies in Taiwan, whose business is mainly focusing on secured customer credit loan by using cars as collaterals, to demonstrate the applicability of DTCAA on a real-world credit assessment problem. As results, DTCAA is shown through the experiments to be able to provide competitive results in terms of misclassification rate, sensitivity and training time. Most importantly, DTCAA is able to generate an easily understandable and interpretable result than ANN, a black-boxed model, or logistic regression, which needs to meet a lot of assumptions. This study confirms that decision tree is indeed a good choice in solving credit assessment problem under a big data environment. By applying DTCAA, the decision tree model can generate a competitive performance as well as provide a clear and easily understandable rule within a fast training time though a series of procedures to reduce the complexity under a big data environment is definitely necessary. This study also demonstrates that the proposed DTCAA is actually applicable in practice since we have applied the DTCAA on a real world dataset obtained from one of the three biggest car collateral loan companies in Taiwan. The experiments show that DTCAA can help them solving their credit assessment problem and produces understandable and interpretable rules that the car collateral loan company can easily use to determine the class of an applicant.

ACKNOWLEDGEMENTS

This research was sponsored by the National Science Committee of Taiwan, under project number: NSC 103-2410-H-002-099-MY3.

REFERENCES

Chang, N. & Sheng, O. R. L. (2008). Decision-Tree-Based Knowledge Discovery: Single- vs. Multi-Decision-Tree Induction. *INFORMS Journal on Computing*, 20(1), 46-54.

Chen, F. L. & Li, F. C. (2010). Combination of Feature Selection Approaches with SVM in Credit Scoring. *Expert Systems with Applications*, 37(7), 4902-4909.

Cieslak, D. A. & Chawla, N. V. (2008). *Learning Decision Trees for Uunbalanced Data Machine Learning and Knowledge Discovery in Databases*. 1st ed: Springer, 241-256.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1), 10–18.

Hosmer Jr, D. W., Lemeshow, S. & Sturdivant, R. X. (2013). Applied Logistic Regression. Retrieved from Wiley.com.

Mahmood, A. M., Gudapati, P., Kavuluru, V. G. & Kuppa, M. R. (2010). A New Pruning Approach For Better And Compact Decision Trees. *International Journal on Computer Science & Engineering*, 2551-2558.

McAfee, A. & Brynjolfsson, E. (2012). Big Data: the Management Revolution. *Harvard Business Review*, 90(10), 60-66.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1), 81-106.

Rahmani, M., Hashemi, S., Hamzeh, A. & Sami, A. (2009). Agent Based Decision Tree Learning: A Novel Approach. *International Journal of Software Engineering and Knowledge Engineering*, 19(7), 1015-1022.

Sahin, Y. & Duman, E. (2011). Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. *Proceedings of the International Multi-Conference of Engineers and Computer Scientists*.

Siddiqi, N. (2005). Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring. 3rdt ed: John Wiley & Sons.

Wang, G., Ma, J., Huang, L. & Xu, K. (2012). Two Credit Scoring Models based on Dual Strategy Ensemble Trees. *Knowledge-Based Systems*, 26(1), 61-68.

Yap, B. W., Ong, S. H. & Husain, N. H. M. (2011). Using Data Mining to Improve Assessment of Credit Worthiness via Credit Scoring Models. *Expert Systems with Applications*, 38(10), 13274-13283,.