CREDIT SCORING USING THE CLUSTERED SUPPORT VECTOR MACHINE

Terry Harris
Credit Research Unit
Department of Management Studies
The University of the West Indies, Cave Hill Campus
Cave Hill Campus, P.O. Box 64, Barbados, W.I
Tel: (246) 417-4302; Fax: (246) 438-9167
terry.harris@cavehill.uwi.edu

ABSTRACT

This work investigates the practice of credit scoring and introduces the use of the Clustered Support Vector Machine (CSVM) for credit scorecard development. This recently designed algorithm addresses some of the limitations noted in the literature that is associated with traditional nonlinear Support Vector Machine (SVM) based methods for classification. Specifically, it is well known that as historical credit scoring datasets get large, these nonlinear approaches while highly accurate become computationally expensive. Accordingly, this study compares the CSVM with other nonlinear SVM based techniques and shows that the CSVM can achieve comparable levels of classification performance while remaining relatively cheap computationally.

Key words: Credit Risk; Credit Scoring; Clustered Support Vector Machine; Support Vector Machine

1 INTRODUCTION

In recent years, credit risk assessment has attracted significant attention from managers at financial institutions around the world. This increased interest has been in no small part caused by the weaknesses of existing risk management techniques that have been revealed by the recent financial crisis (Harris, 2013; Wang, Yan, & Zhang, 2011). Addressing these concerns, over past decades credit scoring has become increasingly important as financial institutions move away from the traditional manual approaches to this more advanced method, which entails the building of complex statistical models (Huang, Chen, & Wang, 2007; Zhou, Lai, & Yu, 2010).

Many of the statistical methods used to build credit scorecards are based on traditional classification techniques such as logistic regression or discriminant analysis. However, in recent times non-linear approaches[1], such as the kernel support vector machine, have been applied to credit scoring. These methods have helped to increase the accuracy and reliability of many credit scorecards (Bellotti & Crook, 2009; Yu, 2008). Nevertheless, despite these advances credit analyst at financial institutions are pressed to continually pursue improvements in classifier performance in an attempt to mitigate the credit risk faced by their institutions. However, many of the improvements in classifier performances remain unreported due to the proprietary nature of industry led credit scoring research which attempts to find more efficient and effective algorithms.

In the wider research community, the recent vintages of non-linear classifiers (e.g the kernel support vector machine) have received a lot of attention and have been critiqued for, *inter alia*, their large time complexities. In fact the best-known time complexity for training a kernel based support vector machine is still quadratic (Bordes, Ertekin, Weston, & Bottou,

[1] This has been applied because credit-scoring data is often not linearly separable.

2005). As a result, when applied to credit scoring substantial computational resources are consumed when training on reasonably sized real world datasets. Accordingly, efforts to develop and apply new classifiers to credit scoring, which are capable of separating nonlinear data while remaining relatively inexpensive computationally, are well placed.

This paper investigates the suitability for credit scoring of a recently developed support vector machine based algorithm that has been proposed by Gu and Han (2013). Their clustered support vector machine has been shown to offer comparable performance to kernel based approaches while remaining cheap in terms of computational time. Furthermore, this study makes some novel adjustments to their implementation and explores the use of radius basis function (RBF) kernels in addition to the linear kernel posited by Gu and Han.

The remainder of this paper is presented as follows. Section 2 outlines a brief review of the literature concerning the field of credit scoring and sets the stage for the proposed CVSM model for credit scoring that is presented in Section 3. The details of the historic clients' loan dataset and modeling method are highlighted in Section 4. Section 5 presents the study results, and Section 6 discusses the findings, presents conclusions, and outlines possible directions for future research.

2 BACKGROUND

2.1 Overview

Credit scoring has been critical in permitting the exceptional growth in consumer credit over the last decades. Indeed without accurate, automated credit risk assessment tools, lenders could not have expanded their balance sheets effectively over this time. This section presents a brief review of the relevant literature that has emerged in this space.

2.2 What is Credit Scoring?

Credit scoring can be viewed as a method of measuring the risk attached to a potential customer, by analyzing their data to determine the likelihood that the prospective borrower will default on a loan (Abdou & Pointon, 2011). According to Hand and Jacka (1998), Eisenbeis (1978) and Hand et al. (2005) credit scoring can also be described as the statistical technique employed to convert data into rules that can be used to guide credit granting decisions. As a result, it represents a critical process in a firm's credit management toolkit. Durand (1941) posited that the procedure includes collecting, analyzing and classifying different credit elements and variables in order to make credit granting decisions. He noted that to classify a firm’s customers, the objective of the credit evaluation process, is to reduce current and expected risk of a customer being “bad” for credit. Thus credit scoring is an important technology for banks and other financial institutions as they seek to minimize risk.

2.3 Problems Associated with Credit Scoring

Crook (1996) and Bailey (2004) posited that credit scorecards can be criticized due to the fact that they fail to include all variables that are informative of a potential client’s likelihood to default and this can lead to the problem of misclassification. Also speaking to this issue, Baesens et al. (2003) noted that credit scoring is a notoriously difficult task as the data collected on and from past customers is often not easily separable. This is in-part due to the nature of the credit assessment exercise, as there is an asynchrony of information between the applicant and the firm, as loan applicants often have more knowledge of their own creditworthiness than credit providers. The financial institution is therefore challenged to gather this information about the applicant. Here, despite the best efforts of the firm it is almost impossible to record every aspect of a client’s life that may result in their default. Hence, credit scorecards often produce higher misclassification rates than other classification problems.

Another criticism of credit scoring is that some variables can be used as proxies for, as they are highly correlated with, legally forbidden model variables. In this way, care must be taken to ensure that these variables are treated appropriately, as in some countries the use of variables that are coextensive with legally prohibited attributes (e.g. race) are also outlawed, thereby adding a layer of complexity to the process (Hand & Henley, 1997). Furthermore, credit scoring has been noted to disadvantage immigrants due to their limited credit history in their country of residence and a lack of information transference from their country of origin.

Credit scoring models can also be expensive to buy and maintain. Compounding this issue is the fact that credit scorecards routinely go out-of-date. This is because, in the case of a parametric model, the learnt weights are assumed to be constant over time. However, this is not the case in reality as the class distribution of creditworthy individuals shifts periodically. Accordingly, this results in a diminution of the accuracy of the credit-scoring model over time (Hand, 2006).

2.4 Size and Time Complexity Constraints

Henley (1994) and Mays (1995) noted that in building practical scoring models, a wide range of statistical and more recently non-linear methods have been used. Here, the use of more complex non-linear techniques, such as neural networks, and support vector machines, to build credit scoring applications has seen significant increases in the reported accuracy and performance on benchmark datasets (Baesens et al., 2003). Irwin et al. (1995) and Paliwal and Kumar (2009) agreed that such advanced statistical techniques provide a superior alternative to traditional statistical methods, such as discriminant analysis, probit analysis and logistic regression, when building practical models. This point of view was also espoused by Masters (1995) who believed that the use of sophisticated techniques, such as neural networks, was essential because of the capability to model credit scoring data that exhibit interactions and curvature. This can be contrasted with traditional linear techniques, such as, linear/logistic regression and linear discriminant analysis.

However the computational costs (time) associated with most of these nonlinear techniques can outweigh the benefits associated with increased classification performance, as the size of the historical clients dataset gets large. This is because many of these algorithms grow exponentially with increasing problem size. Furthermore, increasing computational power offers little in addressing this problem. To illustrate this, consider the fact that if the best known algorithm for solving a given credit scoring problem has a time complexity on the order of 2^n (stated mathematically O(2^n)), where the variable *n* represents the size of the training set and allowing one unit of time to equal one millisecond, then this algorithm can process in one second a maximum input of size of approximately 9.96 as shown in equation (1);

$$\begin{gathered} 2^n = 1{,}000, \\ n\log(2) = \log(1{,}000), \\ n = \frac{\log(1{,}000)}{\log(2)}, \\ n = 9.96, \end{gathered} \tag{1}$$

Now, suppose that the firm wishes to increase the size of its training dataset and decides to purchase a newly designed micro-processor that is able to achieve a tenfold speedup in processing time. This new micro-processor chip would only increase the maximum solvable problem size in one second by 3.32 (as is shown in Equation 2).

$$\begin{gathered} 2^n = 10{,}000, \\ n\log(2) = \log\,(10{,}000), \\ n = \frac{\log(10{,}000)}{\log(2)}, \\ n = 13.28. \end{gathered} \tag{2}$$

This is not very significant! Furthermore it can be contrasted with the increased performance to be derived should a better classification algorithm be applied to the problem. If a new algorithm is capable of transforming the time complexity from O(2^n) to O(*n*) then the maximum size of the problem solvable in one second would be 1,000 (n = 1,000) on the old micro-processor and 10,000 using the new micro-processor chip. Clearly, this is significantly greater than the performance possible using the older algorithm on the faster micro-processor.

As a result, the development and application of more computationally efficient algorithms in the credit scoring space is becoming increasingly more important as the sizes of historical datasets grow. The recently posited Clustered Support Vector Machine reduces the Rademacher complexity of the state-of-the-art SVM based classifier to an upper bound equivalent to the term $15k\sqrt{\log n\,/n}$ where k represents the number of clusters. The interested reader is invited to consult Gu and Han(2013) for further details. In the next section the authors make a contribution to literature by describing the development of the CSVM for credit risk assessment.

3 Clustered Support Vector Machine for Credit Scoring

To build a CSVM classifier from a historical client dataset $S = \{(x_{(i),}\ y_{(i)});\ i = 1,\ldots, m\}$, where m represents the number of instances, ignoring the labels (the $y_{(i)}$'s) partition S into k clusters using K-means such that $\{c^{(j)}; j = 1,\ldots, k\}$. To do this the K-means algorithm assigns every training example $x_{(i),}$ to its closest centroid.[2] Following this, the CSVM classifier for a cluster j can be represented as the linear combination of the attributes of the applicants in the cluster represented by, x's, multiplied by some cluster specific weights, w's, plus a noise term b as is shown in (3).

$$z^{(j)} = w_1^{(j)}x_1^{(j)} + w_2^{(j)}x_2^{(j)} + \cdots + w_n^{(j)}x_n^{(j)} + b^{(j)3} \qquad (3)$$

[2] The initial centroids are randomly selected.

[3] Here the subscript is used to demote the individual variables as opposed to a specific training example. In this paper the use of parenthesis in the subscript will indicate training examples while their absence will denote a specific variable. E.g. $x_{(i)}$ denotes training example i while x_i indicates independent variable i.

where the n denotes the number of client feature variables. Since the $w^{(j)}$'s and $x^{(j)}$'s can be represented as column vectors (3) can be written as;

$$z^{(j)} = w^{(j)\,T}x^{(j)} + b^{(j)}. \tag{4}$$

For each cluster the CSVM learns the parameters $w^{(j)}$ and $b^{(j)}$ and tries to find a hyperplane that maximizes the margin between the creditworthy and un-creditworthy individuals in the cluster. Hence, when given an individual training example $(x_{(i)}^{(j)}, y_{(i)}^{(j)})$, such that $y_{(i)}^{(j)} \in$ {-1,1}, the cluster specific functional margin $\hat{\gamma}^{(j)}$ can be defined for the i'th training example as follows;

$$\hat{\gamma}^{(j)} = y_{(i)}^{(j)}\left(w^{(j)\,T}x^{(j)} + b^{(j)}\right). \tag{5}$$

Furthermore, to confidently predict each training example in the cluster the functional margin needs to be large. And this therefore means that, $w^{(j)\,T}x^{(j)} + b^{(j)}$ must be a large positive number when $y_{(i)}^{(j)}$= 1, and a large negative number when $y_{(i)}^{(j)}$= -1. Thus, the functional margin with respect to the cluster $c^{(j)}$; is necessarily the smallest of the functional margins in the cluster, as in (6).

$$\hat{\gamma}^{(j)} = \min_{i=1,\dots,m} \hat{\gamma}_{(i)}^{(j)} \tag{6}$$

Considering a positive case, where $x_{(i)}^{(j)}$ corresponds to the label $y_{(i)}^{(j)}$ = 1, the geometric distance between this point and the decision boundary, $\gamma_{(i)}^{(j)}$, is a vector orthogonal to the separating hyperplane. Thus, to find the value of $\gamma_{(i)}^{(j)}$, the corresponding point on the decision boundary is located by recognizing that $w^{(j)}/\|w^{(j)}\|$ is a unit vector pointing in the same direction as $w^{(j)}$. As a result, the relevant point on the separating hyperplane can be computed by evaluating the equation $x_{(i)}^{(j)}$- $\gamma_{(i)}^{(j)}$ $w^{(j)}/\|w^{(j)}\|$. In addition, since this point is on the decision boundary, it will satisfy $w^{(j)\,T}x^{(j)} + b^{(j)}$ = 0, as

$w^{(j)\,T}\left(x^{(j)}_{(i)} - \gamma^{(\mathrm{j})}_{(\mathrm{i})}\frac{w^{(j)}}{\|w^{(j)}\|}\right) + b^{(j)} = 0$. And this can be reduced to, $w^{(j)\,T}x^{(j)}_{(i)} - \gamma^{(\mathrm{j})}_{(\mathrm{i})}\frac{w^{(j)\,T}w^{(j)}}{\|w^{(j)}\|} + b^{(j)} = 0$, since, $w^{(j)\,T}w^{(j)}/\|w^{(j)}\| = \|w^{(j)}\|^2/\|w^{(j)}\| = \|w^{(j)}\|$, $\gamma^{(\mathrm{j})}_{(\mathrm{i})}$ can be solved for as $\gamma^{(\mathrm{j})}_{(\mathrm{i})} = (\frac{w^{(j)}}{\|w^{(j)}\|})^T x^{(j)}_{(i)} + \frac{b^{(j)}}{\|w^{(j)}\|}$. Hence, the general representation, taking into account cases of negative training examples, gives the equation $\gamma^{(\mathrm{j})}_{(\mathrm{i})} = y^{(j)}_{(i)}\ [(\frac{w^{(j)}}{\|w^{(j)}\|})^T x^{(j)}_{(i)} + \frac{b^{(j)}}{\|w^{(j)}\|}]$. Finally, recognizing that when $\|w^{(j)}\| = 1$, the geometric margin is equal to the functional margin, the minimization problem, as in (7), can be re-expressed with respect to the geometric margin.

$$\gamma^{(\mathrm{j})} = \min_{i=1,\dots,m} \gamma^{(\mathrm{j})}_{(\mathrm{i})} \tag{7}$$

As a result, in order to find the decision boundary that maximizes the geometric margin for a cluster $c^{(j)}$ the optimization problem shown below must be solved,

$$\begin{gathered} \max_{\gamma^{(\mathrm{j})}, w^{(j)}, b^{(j)}} \ \gamma^{(\mathrm{j})}, \\ s.t.\ y^{(j)}_{(i)}\left(w^{(j)\,T}x^{(j)} + b^{(j)}\right) \geq \gamma^{(\mathrm{j})}, i = 1, \dots, m, \\ \| w^{(j)} \| = 1. \end{gathered} \tag{8}$$

Since the constraint $\|w^{(j)}\| = 1$ is non-convex, the equation (8) is transformed thereby making it more suitable for convex-optimization. To achieve this recognize that if, $\hat{\gamma}^{(\mathrm{j})} = 1$, then $\hat{\gamma}^{(\mathrm{j})}/\|w^{(j)}\| = 1/\|w^{(j)}\|$, and maximizing this is equivalent to minimizing $\|w^{(j)}\|^2$. Furthermore, to avoid over-fitting the cluster data, a regularization term $\xi^{(\mathrm{j})}$, is added coupled with the constant C used to signify a turning parameter that weights the significance of misclassification. In addition, at this point the global reference vector w is added to the optimization problem to leverage information between clusters. Accordingly, the primal form of the general optimization problem is represented as follows;

$$\begin{gathered} \min_{\gamma^{(\mathrm{j})}, w^{(j)}, \ w, \ b^{(j)}} \ \frac{1}{2} \| w \|^2 + \frac{1}{2}\sum_{j=1}^{k} \| w^{(j)} - w \|^2 + C \sum_{j=1}^{k}\sum_{i=1}^{m^{(j)}} \xi^{(\mathrm{j})}_{(\mathrm{i})}, \\ s.t.\ y^{(j)}_{(i)}\left(w^{(j)\,T}x^{(j)}_{(i)} + b^{(j)}\right) \geq 1 - \xi^{(\mathrm{j})}_{(\mathrm{i})}, i = 1, \dots, m^{(j)}, \forall j, \\ \xi^{(\mathrm{j})}_{(\mathrm{i})} \geq 0, i = 1, \dots, m^{(j)}, \forall j. \end{gathered} \tag{9}$$

4 METHODOLOGY

4.1 Data

A German credit scoring dataset was taken from the UCI Machine Learning Repository. This dataset consists of 700 examples of creditworthy applicants and 300 examples of customers who should not have been granted credit. In addition, it presents twenty (20) features for each credit applicant comprising the following categories: the status of the client's existing checking account, the duration of the credit period in months, the client's credit history, the purpose for the credit, the credit amount requested, the client's savings account/bonds balance, the client's present employment status, the client's personal (marital) status and sex, whether the client is a debtor or guarantor of credit granted by another institution, the number of years spent at present residence, the type of property possessed by client, the client's age in years, whether the client has other installment plans, the client's housing arrangements (whether they own their home, rent, or live for free), the number of existing credits the client has at the bank, the client's job, the number of people for whom the client is liable to provide maintenance for, whether the client has a telephone, and whether the client is a foreign worker.

4.2 Experimental Approach

The data were pre-processed so as to transform all categorical data into numerical data for analysis. In addition, the data were normalized so as to improve the performance of the CSVM and the other seven (7) classifiers developed as comparators. All told, the classifiers developed in this paper include the following; logistic regression (LR), K means plus logistic regression (K means + LR), clustered support vector machine with a RBF kernel (CSVM-RBF), K means plus support vector machine with a RBF kernel (K means + SVM-RBF), support vector machine with a RBF kernel (SVM-RBF), linear clustered support vector machine (CSVM-linear), K means plus support vector machine with a linear kernel (K means + SVM-linear), and a linear support vector machine (SVM-linear).

To begin model building, the data-file was randomly split into two data-file—test (20%), and training and cross validation (80%). The withheld test dataset was exclusively used to test the performance of the classification models developed. This approach gives some intuition as to the performance of the models in real world settings. The training and cross-validation dataset was used to develop the models for each classifier type.

In total 35 credit scoring models were built for each classifier type. The classifiers mean performance and standard deviation are reported and discussed in the results section.

4.3 Measures

It has been previously noted that when building and reporting on credit scoring models, it is prudent to make a distinction between metrics used during (i) training phase and (ii) the reporting phase (Harris, 2013). The reason for this being that one needs to be clear as to which metric(s) was (were) used to select model parameters. Consistent with Harris (2013) the term evaluation-metric will be used when referring to the metric used during the training phase, and the term performance-metric used to refer to the measure used to report models performance at the reporting phase.

The Area under the Receiver Operating Characteristic (ROC) curve (AUC) is designated as the primary model evaluation metric and performance metric in this study. The AUC makes use of the ROC curve, which is a two dimensional measure of classification performance where the sensitivity (10) (i.e. the proportion of actual positives predicted as positive) and the specificity (11) (i.e. the proportion of actual negatives that are predicted as negative), are plotted on the Y and X axis, respectively. The AUC measure is highlighted as in (12) below where, S_1, represents the sum of the ranks of the creditworthy clients. Here, a score of 100% indicates that the classifier is able to perfectly discriminate between the classes, and a score of 50% indicates a classifier of insignificant discriminatory quality.

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False negative}} \tag{10}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}} \tag{11}$$

$$\text{AUC} = \frac{(s_1 - \text{Sensitivity}) * [(\text{Sensitivity} + 1) * 0.5]}{\text{Sensitivity} * \text{Specificity}} \tag{12}$$

A number of other performance metrics are also used to report the performances of the classifiers developed in this paper. For example, Test accuracy, as in (13) is also reported as it measures how accurately the credit applicants on a withheld test dataset are classified.

$$\text{Test accuracy} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} + \frac{\text{True Negative}}{\text{False Negative} + \text{True Negative}} \tag{13}$$

An arguably more meaning full measure of classifier performance is the balanced accuracy (BAC) as in (14). This measure avoids the misleading affects on accuracy caused by imbalanced datasets by showing the arithmetic mean of sensitivity and specificity. Since skewed datasets are a common occurrence with real world credit scoring datasets this measure may be more relevant.

$$\text{BAC} = \frac{\text{Specificity} + \text{Sensitivity}}{2} \tag{14}$$

5 Results and Analysis

It has been widely noted that credit-scoring is a difficult task as credit data is very often not easily separable. The nature of the credit assessment exercise entails asynchrony of information between the applicant and the assessor. As a result, credit analysts are responsible for gathering pertinent information about the loan applicant. However, very often the best efforts of the analyst are insufficient to appraise every aspect of a client’s life. Hence, credit-scoring usually results in higher misclassification rates than would normally be considered acceptable (Baesens et al., 2003). The reader is asked to bear this in mind when interpreting the results presented.

5.1 Classifier performances

Table 1 presents the performances of the CSVM classifier in addition to seven (7) other comparator classification methods built using the German dataset. In total thirty-five credit-scoring models for each classifier were built. The withheld test dataset was used to report the mean and standard deviation values for each performance metric. Here results presented in Table 1 suggest that the models built were indeed predictive of creditworthiness as indicated by AUC on the withheld test dataset.

Insert Table 1 here

5.2 Significances of AUC Differences

The ANOVA analysis for the eight model types is highlighted in Table 2. There the results indicate a significant difference between one or more of the classifiers (i.e. the groups) when comparing mean AUC scores ($F = 3.284$, $p < 0.05$).

Insert Table 2 here

Accordingly, a Bonferroni test was computed to determine which classifiers were performing significantly different from each other (Levene's statistic = 1.444; $p = 0.187$). Table 3 illustrates the results of this testing and shows that the only significant difference was between the mean AUC scores of the logistic regression models and the SVM models with a linear kernel function. In terms of performance the CSVM models (both linear and RBF) showed comparable AUCs to the other classifiers as there was no significant difference between them and the other classifiers in terms of AUC.

Insert Table 3 here

5.3 Training time

Consistent with the author's expectations, the average training time for the linear CSVM model was considerably shorter than that of the other models (Please see Table 1), particularly the K means + SVM (linear and RBF kernels), SVM-RBF, and the K means + LR models. It is interesting that the base line SVM linear out performs the CSVM-linear in terms of training time. However, the results indicate that the linear CSVM consistently outperforms its direct comparators, which are the K means + SVM-linear, SVM-RBF, and K means + SVM-RBF.

6 CONCLUSION

This paper introduces the use of the CSVM for credit scoring. The CSVM represents a possible solution to the limitations of the current crop classifiers used in practice. Prior work has noted that as datasets get large nonlinear approaches become increasingly computationally expensive. As a result, the search for more computationally efficient algorithm has intensified in recent years as data analyst seek to discover patterns in datasets of increasing size and complexity without seeding classifier performance.

The results of the paper suggest that the CSVM compare well with nonlinear SVM based techniques in terms of AUC, while outperforming them in terms of training time. It is the CSVM's cutting edge performance coupled with its comparatively cheap computational cost that makes it an interesting algorithm in the credit scoring space.

The future work of this author will seek to improve the classification performance of the CSVM algorithm in terms of AUC and mean model training time. In addition, other metrics will be used as the primary model evaluation metric. Furthermore, future studies will consider the impact of extending the clustered approach to other classification techniques such as random forest.

REFERENCES

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

Baesens, B., van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*, 1082-1088.

Bailey, M. (2004). *Consumer credit quality: underwriting, scoring, fraud prevention and collections*: White Box Publishing.

Bellotti, T., & Crook, J. (2009). Support vector machines for credit scoring and discovery of significant features. *Expert Systems with Applications, 36*(2), 3302-3308.

Bordes, A., Ertekin, S., Weston, J., & Bottou, L. o. (2005). Fast kernel classifiers with online and active learning. *The Journal of Machine Learning Research, 6*, 1579-1619.

Crook, J. N. (1996). Credit scoring: An overview. Working Paper Series. University of Edinburgh.

Durand, D. (1941). *Risk Elements in Consumer Instalment Financing*. NY: National Bureau of Economic Research.

Eisenbeis, R. A. (1978). Problems in applying discriminant analysis in credit scoring models. *Journal of Banking & Finance, 2*(3), 205-219.

Gu, Q., & Han, J. (2013). *Clustered support vector machines.* Paper presented at the Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics.

Hand, D. J. (2006). Classifier technology and the illusion of progress. *Statistical Science, 21*(1), 1-14.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society), 160*(3), 523-541.

Hand, D. J., & Jacka, S. D. (1998). *Statistics in finance*: Arnold London.

Hand, D. J., Sohn, S. Y., & Kim, Y. (2005). Optimal bipartite scorecards. *Expert Systems with Applications, 29*(3), 684-690.

Harris, T. (2013). Quantitative credit risk assessment using support vector machines: Broad versus Narrow default definitions. *Expert Systems with Applications, 40*(11), 4404-4413.

Henley, W. E. (1994). *Statistical aspects of credit scoring.* Open University.

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

Irwin, G. W., Warwick, K., & Hunt, K. J. (1995). *Neural network applications in control*.

Masters, T. (1995). *Advanced algorithms for neural networks: A C++ sourcebook*: John Wiley & Sons, Inc.

Mays, E. (1995). *Handbook of credit scoring*: Global Professional Publishing.

Paliwal, M., & Kumar, U. A. (2009). Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications, 36*(1), 2-17.

Wang, Z., Yan, S. C., & Zhang, C. S. (2011). Active learning with adaptive regularization. *Pattern Recognition, 44*(10-11), 2375-2383.

Yu, L. (2008). *Bio-inspired credit risk analysis: computational intelligence with support vector machines*: Springer.

Zhou, L., Lai, K. K., & Yu, L. (2010). Least squares support vector machines ensemble models for credit scoring. *Expert Systems with Applications, 37*(1), 127-133.

Table 1

Showing Comparative Classifier Performances

Classifier		Training Accuracy	Test Accuracy	BAC	AUC	Training Time (Seconds)
1) K means + LR						
	Mean	77.625	74.700	69.526	68.868	0.103
	S.D	0.995	1.716	3.021	2.758	0.005
2) LR						
	Mean	70.675	68.900	71.955	70.855	0.035
	S.D	0.337	1.798	2.699	2.875	0.004
3) CSVM-RBF						
	Mean	84.525	77.100	69.834	69.234	0.071
	S.D	2.897	2.114	3.775	3.172	0.038
4) K means + SVM-RBF						
	Mean	83.250	76.500	69.000	68.614	0.141
	S.D	2.494	1.604	3.871	3.119	0.184
5) SVM-RBF						
	Mean	83.400	78.000	70.654	69.526	0.122
	S.D	2.236	1.843	3.269	2.915	0.021
6) CSVM-linear						
	Mean	79.300	76.300	71.387	70.219	0.029
	S.D	0.565	2.477	2.551	2.830	0.004
7) K means + SVM-linear						
	Mean	82.233	76.381	69.238	68.752	0.107
	S.D	2.514	2.105	3.790	3.089	0.046
8) SVM-linear						
	Mean	78.950	78.700	69.779	69.133	0.017
	S.D	0.404	1.045	2.449	2.942	0.042

Table 2
Showing summary the ANOVA computed for the eight groups of classifiers

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	202.732	7.000	28.962	3.284	0.002
Within Groups	2398.705	272.000	8.819		
Total	2601.437	279.000			

Table 3
Showing comparisons of the classifiers using Bonferroni's method

Bonferroni	Classifier (I)	1							2						
	Classifier (J)	2	3	4	5	6	7	8	1	3	4	5	6	7	8
Mean Difference (I-J)		-1.987	-0.366	0.254	-0.658	-1.351	0.254	0.735	1.987	1.621	2.241	1.329	0.636	2.241	2.722
Std. Error		0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710
Sig.		0.154	1.000	1.000	1.000	1.000	1.000	1.000	0.154	0.649	0.050	1.000	1.000	0.050	0.004
95% Confidence Interval	Lower Bound	-4.227	-2.606	-1.986	-2.898	-3.591	-1.986	-1.505	-0.252	-0.619	0.001	-0.911	-1.603	0.001	0.482
	Upper Bound	0.252	1.874	2.493	1.582	0.889	2.493	2.974	4.227	3.861	4.480	3.569	2.876	4.480	4.961

	Classifier (I)	3							4						
	Classifier (J)	1	2	4	5	6	7	8	1	2	3	5	6	7	8
Mean Difference (I-J)		0.366	-1.621	0.620	-0.292	-0.985	0.620	1.101	-0.254	-2.241	-0.620	-0.912	-1.605	0.000	0.481
Std. Error		0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710
Sig.		1.000	0.649	1.000	1.000	1.000	1.000	1.000	1.000	0.050	1.000	1.000	0.688	1.000	1.000
95% Confidence Interval	Lower Bound	-1.874	-3.861	-1.620	-2.532	-3.225	-1.620	-1.139	-2.493	-4.480	-2.859	-3.151	-3.844	-2.240	-1.759
	Upper Bound	2.606	0.619	2.859	1.948	1.255	2.859	3.340	1.986	-0.001	1.620	1.328	0.635	2.240	2.721

Table 3 Continued

	Classifier (I)	5							6						
	Classifier (J)	1	2	3	4	6	7	8	1	2	3	4	5	7	8
Mean Difference (I-J)		0.658	-1.329	0.292	0.912	-0.693	0.912	1.393	1.351	-0.636	0.985	1.605	0.693	1.605	2.086
Std. Error		0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710
Sig.		1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.688	1.000	0.688	0.100
95% Confidence Interval	Lower Bound	-1.582	-3.569	-1.948	-1.328	-2.933	-1.328	-0.847	-0.889	-2.876	-1.255	-0.635	-1.547	-0.635	-0.154
	Upper Bound	2.898	0.911	2.532	3.151	1.547	3.151	3.632	3.591	1.603	3.225	3.844	2.933	3.844	4.325

	Classifier (I)	7							8						
	Classifier (I)	1	2	3	4	5	6	8	1	2	3	4	5	6	7
Mean Difference (I-J)		-0.254	-2.241	-0.620	0.000	-0.912	-1.605	0.481	-0.735	-2.722	-1.101	-0.481	-1.393	-2.086	-0.481
Std. Error		0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710	0.710
Sig.		1.000	0.050	1.000	1.000	1.000	0.688	1.000	1.000	0.004	1.000	1.000	1.000	0.100	1.000
95% Confidence Interval	Lower Bound	-2.493	-4.480	-2.859	-2.240	-3.151	-3.844	-1.759	-2.974	-4.961	-3.340	-2.721	-3.632	-4.325	-2.721
	Upper Bound	1.986	-0.001	1.620	2.240	1.328	0.635	2.721	1.505	-0.482	1.139	1.759	0.847	0.154	1.759