Corporate Financial Analysis with efficient Logistic Regressions and Hybrids of Neuro-Genetic networks

LOUKERIS N.
C.C.F.E.A.
University of Essex
Wivenhoe park, CO4 3SQ, UK or Ag. Ioannou Xenou 9, 73100 Chania, Crete, Greece
United Kingdom

Abstract:- Financial institutions, portfolio managers and investors demand strong analytical methods of corporate finance to maintain lucrative investment portfolios. The volatility of stock prices, affected partially by the vast accounting data and the level of efficiency in the financial market require support by accurate decision making to increase the value of investments. Logistic regressions in Econometrics achieve significant results in financial analysis of companies, whilst Artificial Intelligence-as nonlinear regression systems- provides efficient corporate financial evaluations in longer computation time.

Keywords:- Logistic Regressions, Hybrid Systems, Neural Networks, Genetic Algorithms, Financial Analysis

1 Introduction

Decision making in assets and portfolio management seeks efficient methods of financial analysis to make the difference, providing higher profitability to investors. The complexity of financial and accounting data includes hidden information regarding the real value of corporations. Thus Econometrics with various models of Logistic regression offers financial analysis of corporations with classifications of high precision. Furthermore Artificial Intelligence approaching nonlinearly systems behaviour deploys vigorous methods of hybrid Neural Networks with Genetic Algorithms optimization in corporate finance, **[1]**. The determination of the most efficient methods in Corporate Financial Analysis is the objective of this research.

2 The Regression Models

2.1 Multinomial Logistic Regression- Logistic function in WEKA

The Multinomial Logistic Regression-MLR model with a ridge estimator was used to elaborate for n instances at m attributes, the dimension in matrix B of parameters is m*(k-1),. The probability for class j with the exception of the last class is:

$$P_j(X_i) = e^{XiBj} / ((\Sigma\, e^{Xi*Bj})+1)$$

$$j=1,..(k-1)$$

financial analysis of corporations. In the Multinomial Logistic Regression given k classes The probability of last class is:

$$1-(\underset{j=1,..(k-1)}{\Sigma}\, P_j(X_i)) = 1/((\underset{j=1,..(k-1)}{\Sigma}\, e^{Xi*Bj})+1)$$

Consequently the multinomial log-likelihood will have negative values as:

$$L = -\underset{i=1,..n}{\Sigma}\, [\underset{j=1,..(k-1)}{\Sigma}\, (Y_{ij} * \ln(P_j(X_i))) + (1 - \underset{j=1,..(k-1)}{(\Sigma Y_{ij})} * \ln(1 - \underset{j=1,..(k-1)}{\Sigma}$$

$$P_j(X_i))] + ridge * B^2$$

The Quasi-Newton Method is implemented to seek the optimized values of m*(k-1) variables, aiming to find the matrix B for which L is minimised, **[2]**. The initial Logistic Regression algorithm, **[3]**, does not compute instance weights, an adjusted algorithm was implemented through WEKA platform calculate the instance weights. The missing values are replaced with ReplaceMissingValuesFilter, and nominal attributes are transformed into numeric attributes with NominalToBinaryFilter.

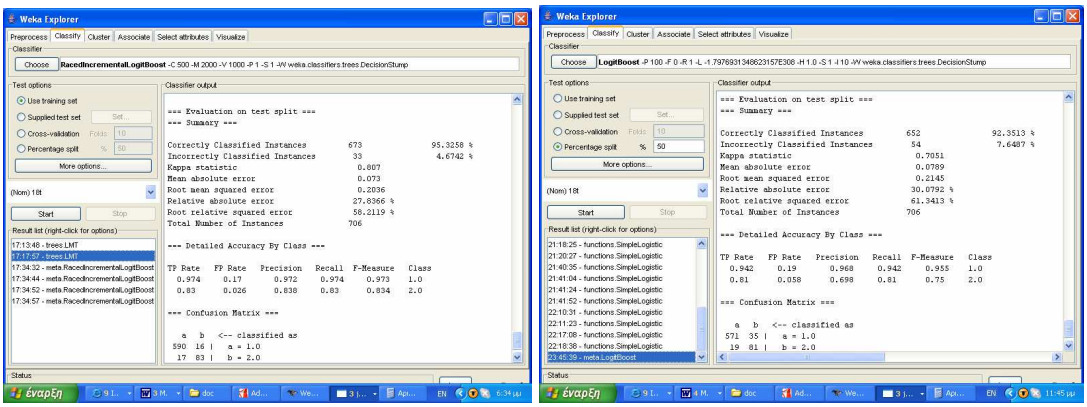

Figure 1. The Multinomial Logistic Regression results in split of 50% to create training set (left), The Additive Logistic Regression -Logitboost (right)

2.2 Additive Logistic Regression -Logitboost in WEKA

Based on the research of **[4]**, **[5]** Boosting is a considerably important development in the classification domain, where it applies in a sequential order a classification algorithm to reweighted versions of the training data, and on the next step takes a weighted majority vote of the sequence of produced classifiers, causing a accelerating improvement in performance. Additive modeling and maximum likelihood, for the two-class problem with boosting, consist an approximation to additive modeling on the logistic scale using maximum Bernoulli likelihood as a criterion. Direct multi-class generalizations based on multinomial likelihood are derived that exhibit performance comparable to other recently proposed multi-class generalizations of boosting in most situations, and far superior in some. The general form of additive models is: $P_j(X_i) = \alpha + \Sigma f(X_i))$, describing the Additive Regression models as well, whilst AdaBoost M1 model is described by: $F(x) = \Sigma c_m f_m(x)$.

LogitBoost is a form of Additive Logistic Regression, having the ability to boost very simple learning schemes even in cases of multiple classes, **[6]**, performing in a superior manner than AdaBoost M1 algorithm. LogitBoost boosts schemes for numeric prediction, to create a combined classifier that predicts a categorical class, regression scheme as the base learner in multi-class problems, elaborating efficient internal cross-validation to determine appropriate number of iterations. LogitBoost is a form of Additive an activity that AdaBoost M1 does not perform. The classification process is implemented through a Logistic Regression, having the ability to boost very simple learning schemes even in cases of multiple classes, **[6]**, performing in a superior manner then AdaBoost M1 algorithm. LogitBoost boosts schemes for numeric prediction, to create a combined classifier that predicts a categorical class, an activity that AdaBoost M1 does not perform. The classification process is implemented through a regression scheme as the base learner in multi-class problems, elaborating efficient internal cross-validation to determine appropriate number of iterations.

2.3 Simple Logistic

Another way to create Linear Logistic Regression is through SimpleLogistic function of WEKA platform. The SimpleLogistic function as a Linear Logistic Regression implements a LogitBoost algorithm implementing ordinal regression functions as base learners to fit the logistic models. Cross-validation is used to acquire the optimal number of LogitBoost iterations that offer automatic attribute selection, **[7]**.

2.4 Logistic Model Trees-LMT

Logistic Model Trees are classification trees implementing in their leaves logistic regression functions at the leaves. Logistic Model Trees may process binary and multi-class target variables, numeric and nominal attributes and missing values, **[7]**.

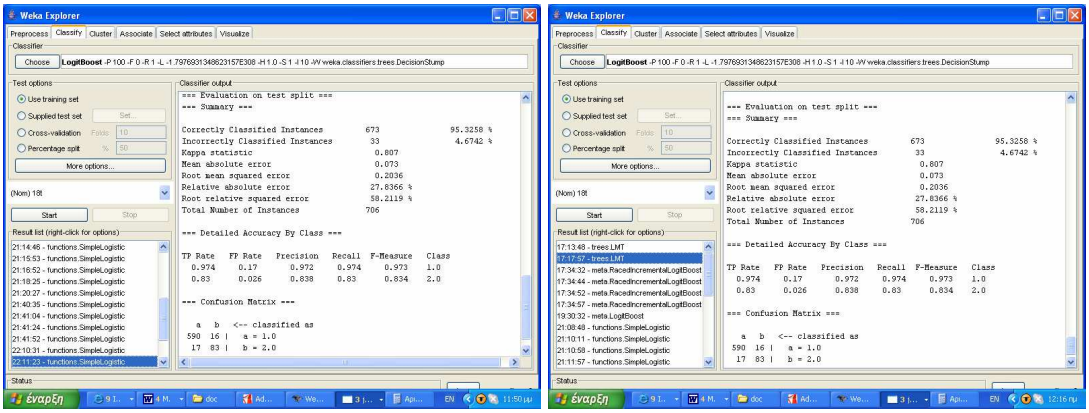

Figure 3. The Linear Logistic Regression- SimpleLogistic results as 50% split training set (left), and ii) Logistic Model Trees results (right)

3 The Hybrid Neuro-Genetic Networks

Based on the previous research of Loukeris and Matsatsinis (2006[b]) the results of 10 different neural networks architectures were used: 1)Principal Component Analysis networks-PCA, 2) Recurrent networks, 3)Time Lag Recurrent Network – TLRN, 4) Support Vector Machine – SVM, 5) Kohonen Shelf Organizing Maps-SOFIM, 6) Jordan Elman networks, 7) Multi Layer Perceptrons – MLP, 8) Generalized Feed Forward – GFF, 9) Modular networks, 10) Radial Basis Function Network – RBFN, in different topologies. The optimal hybrid Neuro-Genetic networks were selected out of these 10 models providing 4 excellent hybrid models for corporate financial analysis, table 2.

The Genetic Algorithms selected the significant inputs in the Neural Networks, requiring multiple training on the network to find the optimal input combination with the lowest error. Genetic Algorithms were used on each layer with different topologies. On-Line learning updated the weights of hybrid neuro-genetic nets, after the presentation of each exemplar. Genetic Algorithms optimized the a) Processing Elements, b) Step Size and c) Momentum Rate. Output layer was chosen to implement Genetic Algorithms in all networks, optimizing the value of the Step size and the Momentum.

Data came by 1411 companies from the loan department of a Greek commercial bank, with the following 16 financial indices from the period 1994-1997: 1) EBIT/Total Assets**,** 2) Net Income/Net Worth, 3) Sales/Total Assets, 4) Gross Profit/Total Assets, 5) Net Income/Working Capital, 6)Net Worth/Total Liabilities 7)Total Liabilities/Total assets, 8) Long Term Liabilities /(Long Term Liabilities + Net Worth), 9)Quick Assets/Current Liabilities 10)(Quick Assets-Inventories)/Current Liabilities, 11)Floating Assets/Current Liabilities, 12)Current Liabilities/Net Worth, 13) Cash Flow/Total Assets, 14)Total Liabilities/Working Capital, 15)Working Capital/Total Assets, 16) Inventories/Quick Assets, and a 17th index with initial classification, done by bank executives. Test set was 50% of overall data, and training set 50% as well.

4 Results of Regressions

In the Logistic function of WEKA (Multinomial Logistic Regression) the output debug information to the console D was set to false, the maximum number of iterations to perform, M, was by default -1, hence the algorithm iterates until convergence, and the ridge value in the log-likelihood ridge, R, was 1E-08.

The Multinomial Logistic Regression as a Logistic function revealed an adequate convergence where the initially categorized as healthy companies by human experts were classified as healthy at a rate of 82.43% (582 companies), table 1, the healthy companies classified falsely as in distress were a rate of 1.84% (13 cases), and the distressed companies classified as healthy were 4.10% (29 cases), whilst the distressed companies that were classified as in distress were 11.67% (82 cases). The model needed 0.39 seconds to be built, producing 664 correctly classified instances (94.051 %) in 42 incorrectly classifications (5.949 %). The Kappa statistic that measures interobserver variability was quite good at 0.7615, and Mean Absolute Error 0.0761, when the Root Mean Squared Error was 0.2194, indicating the cost function in the form of Mean Square Error as 0.0481 revealing a satisfactory fitness of the network output to the desired output, the Relative

Absolute Error was 29.0656 %, and the Root Relative Squared Error at 60.2752 %.

The LogitBoost, as a Logistic Regression, used the Decision Stump base classifier, the debug was false, the likelihoodThreshold that is a threshold on improvement in likelihood was initially -1.79E308, the number of folds for internal cross-validation was by default 0 indicating that means no cross-validation is performed, the number of iterations to be performed was set to 10, with 1 run for internal cross-validation, the random number seed was set to 1, the Shrinkage parameter was 1.0 to reduce overfitting, and finally the weight threshold for weight pruning was initially to 100.

Logit Boost had a satisfactory convergence, since the initially characterized healthy companies, by loan experts, were classified through Logistic Regression as healthy in a proportion of 82.43 % (582 companies), with 13 misclassified healthy companies in the category of the distressed companies, 30 companies initially categorized as in distress were put in the healthy category, and 81 distressed companies were classified as in distress. The performed iterations were 10, whilst the time taken to build model was very short at: 0.63 seconds. The evaluation on test split produced 663 correctly classified instances (93.9093 %), and 43 incorrect classifications (6.0907 %), the Kappa statistic was satisfactory at 0.7549, whilst the Mean Absolute Error was 0.0852, the Root Mean Squared Error received 0.2227 with the highest cost function of MSE at 0.04959, between all logistic regressions, in a well fitted network output to the desired output, and Relative Absolute Error reached 32.5318 %, with a Root Relative Squared Error at 61.1839 % given that the training set was on the 50% split of the initial 1411 companies.

SimpleLogistic function performed the logistic regression model using LogitBoost. The errorOnProbabilities was not selected since its use did not provide any significant difference in the several tests we did with it, the RMSE without it was lower at 0.2036, whilst the RMSE error increased to 0.2088 when it was chosen. The heuristicStop was 50, activating the heuristic algorithm for greedy stop while cross-validation is used to LogitBoost iterations, causing a stop to LogitBoost in case that new error minimum has not been reached in the last heuristicStop iterations, and accelerating computing time. The maximum number of iterations for LogitBoost was selected 500. The fixed number of iterations for LogitBoost was 0, whilst the number of LogitBoost iterations to be cross-validated or the stopping criterion on the training set should be used was set on true value.

SimpleLogistic function had an accurate convergence with 83.56% of the companies initially characterized as healthy by bank executives to be classified as healthy by the model (590 cases), 2.26% of the healthy companies were put in the distress category (16 cases), 2.40% of companies in distress were classified as healthy (17 cases), and 23.51% of in distress companies were classified as in distress (83 cases). The model required 13.78 seconds to be built, revealing 673 correctly classified instances (95.3258 %), with 33 incorrectly classified instances (4.6742 %). The Kappa statistic was quite adequate at 0.807, MAE 0.073, the RMSE 0.2036 offering the cost function as the Mean Square Error at 0.041 was satisfactory, but compared to the other logistic regression models had the highest value, with a satisfactory fitness of the network output to the desired output nevertheless. RAE was 27.8366 %, and RRSE at 58.2119 %.

In Logistic Model Trees- LMT minimization of error on probabilities instead of misclassifying the error when cross-validating the number of LogitBoost iterations, through procedure errorOnProbabilities was not selected, since it required vast computing times in the experiments. With fastRegression a use heuristics avoided cross-validating the number of Logit-Boost iterations at every node. In case of fitting the logistic regression functions at a node, LMT determines the number of LogitBoost iterations to run, which was cross-validated at every node in the tree. This heuristic cross-validates the number only once and then uses it at every node in the tree, without decreasing accuracy and by significantly improving runtime. The minimum number of instances at which a node is considered for splitting was 15. The fixed number of iterations for LogitBoost was -1 causing cross-validation to this number. The splitting criterion on residuals of LogitBoost was not selected.

The Logistic Model Tree classified the healthy companies according to bank experts in the class of healthy at a rate of 83.56% (590 cases), whilst some healthy companies were classified as in distress at a rate 2.26% (16 cases), the misclassifications included companies in distress which were categorized as healthy at a rate 2.40% (17 cases), finally the distressed companies were classified as in distress at a rate 23.51% (83 cases). With split 50% for the training data the Logistic Model Tree had 6 leaves and its size was 11, whilst the time taken to build model was 54.92 seconds.

The correctly classified instances were 673 (95.3258 %) and the incorrectly 33 (4.6742 %). Kappa statistic was very good at 0.807, with a MAE at 0.073, RMSE 0.2036 supplying the cost function with MSE at 0.0414 at an adequate level, in the lowest values among the four different regression types, providing an excellent fitness of the network output to the desired output, RAE was 27.8366 %, and RRSE 58.2119 %.

Table 1. Results for the regressions as training set splitted to 50% of overall data

Regressions	0->0	0->1	1->0	1->1	Miscla s.	Correct class.	K-stat	MAE	MSE	RMSE	RAE	RRSE	Time
Logistic	582	13	29	82	42	664	0.7615	0.076	0.0481	0.2194	29.06%	60.27%	0.39
	82.43%	1.84%	4.10%	1.67%	5.94%	94.05%		1					sec
Logit boost	582	13	30	81	43	663	0.7549	0.085	0.0495	0.2227	32.53%	61.18%	0.63
	82.43%	1.84%	4.24%	1.47%	6.097%	93.9%		2					sec
Simple	590	16	17	83	33	673	0.807	0.073	0.0414	0.2036	32.75%	62.60%	13.78
Logistic	83.56%	2.26%	2.40%	23.51%	4.67%	95.3%							sec
Logistic Mod	590	16	17	83	33	673	0.807	0.073	0.0414	0.2036	27.83%	58.21%	54.92
Trees	83.56%	2.26%	2.40%	23.51%	4.67%	95.3%							sec

Table 2. Networks with excellent performance

Hybrid Network	Active Confusion Matrix				MSE	NMSE	r	Time
Layers	0->0	0->1	1->0	1->1				
Jordan Elman 1	100	0	0	100	0.029	0.113	0.96	2 h 01’
SOFM 1	100	0	0	100	0.042	0.042	0.979	5 h 01’
Modular 3	100	0	0	100	0.013	0.054	0.972	12 h 09’
SVM 1000 epoc.	100	0	0	100	0.849	2.672	0.677	4 h 13’

5 Comparison

5.1 The hybrid optimal results

Hybrids of Neural Networks with Genetic Algorithms for optimization on genes solutions produced 5 independent confusion matrices with correct classification at a level 100% that resulted in the same form. It is obvious that the most excellent Hybrid Neural Network with Genetic Algorithms optimization was: Jordan Elman with 1 hidden layer, table 2, with a very low MSE, the second lower of 43 networks that were deployed in this research, its NMSE was very low at 0.042 and correlation coefficient r was very high at 0.960, whilst the time to converge was the fastest of all at 2 hours and 1’. The second better network was SOFM with 1 hidden layer converged slower in 5 hours and 1’ whilst it had the lowest MSE of all: 0.010, and the highest r 0.979. Another hybrid neural network that had a quick convergence was Modular network with 3 hidden layers concluded its convergence in 2 hours 9’, the MSE was 0.013, the lowest of nets, and r 0.972. Finally SVM – 1000 epochs that converged in 4 hours 13’ with a very high cost function at MSE of 0.849 and the lowest r at 0.677.

5.2 The optimal Regressions results

The optimal financial evaluation of corporations was achieved by Simple Logistic regression of which the performance provided the lowest cost function expressed by Mean Square Error, a result which was also achieved by Logistic Model Trees, and the confusion matrix had the highest convergence. The computation time for Simple Logistic regression was 13.78 seconds higher than those of Logistic and Logit boost regressions, and significantly lower than the processing time of Logistic Model Trees which was extended at 54.92 seconds. Simle Logistic regression and Logistic Model Trees regression achieved the same Kappa statistic, at 0.807 which was the highest among all, the lowest Root Mean Square Error, whilst the Relative Absolute Error of Simple Logistic regression was the highest, and the RAE of Logistic Model Trees the lowest of the four regressions. The

lowest Root Relative Square Error was achieved by Logistic Model Trees, followed by Logistic regression, by Logit Boost and finally by Simple Logistic regression.

5.3 Comparative analysis of results in regressions and hybrid models

The hybrid neuro-genetic networks produced a totally converged confusion matrix to the initial classifications of loan executives, whilst the logistic regression models could not achieve the same accuracy in their classifications. Also two Hybrid networks: the Modular with 3 layers and the Jordan/Elman with 1 layer had the lowest Mean Square Error at 0.013 and 0.029 providing an excellent fitness of the network output to the desired output, whilst the Simple Logistic regression and the Logistic Model Trees followed with 0.0414 each, followed by SOFM hybrid with 0.042, then came Logistic Regression with 0.0481 and Logit boost regression with 0.0495 and last SVM hybrid with a huge MSE at 0.807 indicating an unsatisfactory fitted network output to the desired output.

The correlation coefficient r met high values in three hybrids: SOFM had 0.979 an almost perfect fitness of the model to the data, the Modular had 0.972, the Jordan/Elman 0.96 and finally the SVM received a 0.677 with a moderate fitness of the model to data, on the other hand the logistic regressions had significant values in Kappa statistic, indicating interobserver agreement in results-similar to correlation coefficient- where the Simple Logistic regression and Logistic Model Trees had 0.807, Logistic regression had 0.7615 and finally Logit boost ranked last with a 0.7549.

The computation time is obvious in favour of the logistic regressions models since their results were given very fast from 0.39 seconds-Logistic regression- until 54.92 seconds for the Logistic Model Trees, whilst the hybrid neuro genetic systems were extremely time consuming, demanding from 2 hours and 1 minute - Jordan/Elman hybrid- until 12 hours and 9 minutes –Modular hybrids.

6 Conclusions-Future Research

The comparative analysis of logistic regressions and the optimal architectures of hybrid neuro-genetic networks indicated clearly that Jordan/Elman, SOFM, and Modular neuro-genetic hybrids had an excellent performance in financial classifications which was time consuming at computations, followed by Simple Logistic regression and Logistic Model Trees that produced classifications in a well fitted network output to the desired output with very short processing time requirements. In the future a thorough cross examination to the hybrid systems and regressions can reveal the most reliable methods in Corporate Financial Analysis

7 Bibliography

[1]. Loukeris N., Matsatsinis N., (2006)-Corporate Financial Evaluation and Bankruptcy Prediction implementing Artificial Intelligence methods-, WSEAS Transactions in Business and Economics, Issue 4, Volume 3, April

[2]. Witten I., Frank E. (2000)- Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations- Morgan Kaufmann Publishers, Department of Computer Science, University of Waikato, Hamilton, New Zealand

[3]. le Cessie S. and van Houwelingen J.C. (1992)- Ridge Estimators in Logistic Regression. Applied Statistics, Vol. 41, (1), 191-201.

[4]. Freund, Y. and Schapire, R.(1996), Experiments with a new boosting algorithm. In Machine Learning: Proceedings of the Thirteenth International Conference, 148–156.

[5]. Freund, Y. and Schapire, R.(1997), A decision-theoretic generalization of on-line learning and an application to boosting. ,Jour. of Comp. and Sys. Sci., 55(1), 119–139.

[6]. Friedman J., T. Hastie and R. Tibshirani (1998)- Additive Logistic Regression: a Statistical View of Boosting- Technical report, Stanford University

[7]. Landwehr N., Hall M. & Frank E. (2003)- Logistic Model Trees-, ECML.