Default Detection of Credit Card Clients

Literature Review – CKME136  
Harry Ragunathan

# Introduction

Credit-risk evaluation is a very challenging and important management science problem in the domain of financial analysis. Many techniques have been proposed, but neural networks have been receiving the high praise in default classification.

In this project I will attempt to classify credit card clients into defaulting on their next monthly payments or not defaulting. In order to do this, explanatory variables (including both demographic and past payment information) will be used in a neural network algorithm to predict whether or not a client will default.

# Literature Review

The following are summaries of the related papers I have reviewed to solve the problem of default detection of credit card clients.

## A Proposed Classification of Data Mining Techniques in Credit Scoring [1]

This article explores different types of data mining techniques that are employed for credit scoring. Methods include neural networks, Bayesian classifier, discriminant analysis, logistic regression, k-nearest neighbor, decision tree, survival analysis, fuzzy rule-based system, support vector machine, and hybrid models (integrating two or more existing models).

## Benchmarking state-of-the-art classification algorithms for credit scoring [2]

This article explores the effectiveness of different types of classification algorithms for credit scoring. It is an update 10 years after the original article was published since there has been numerous statistical advancements in predictive modeling. The results of this study consists of the performance estimates of 41 classifiers across 7 credit scoring data sets. Artificial neural networks still continues to be a very powerful individual classifier. There was a tendency that homogeneous ensemble classifiers outperform individual classifiers. Of these two groups the five best performing methods belong to the homogeneous ensemble family.

## Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation [3]

This paper has evaluated and contrasted three neural network rule extraction techniques – Neurorule, Trepan, and Nefclass, for credit-risk evaluation. This is relevant to my project as I will be utilizing neural networks to classify defaults for credit card clients. Experiments were conducted on three real-life financial credit-risk evaluation data sets. Both Neurorule and Trepan yielded a very good classification accuracy compared to the popular C4.5 algorithm and the logistic regression classifier.

# Dataset

The data used for this project has been provided by I-Cheng Yeh of Chung Hua University and Che-hui Lien of Tamkang University in Taiwan and consists of demographic information and past payment information of clients.

The following are the columns present in the dataset:   
X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.   
X2: Gender (1 = male; 2 = female).   
X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).   
X4: Marital status (1 = married; 2 = single; 3 = others).   
X5: Age (year).   
X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.   
X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.   
X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.

The dataset also contains a column “Y” where a value of 1 means that the client defaulted in the next month and 0 if they didn’t. I will predict this column in the evaluation step and use the value present to train the model in the training step.

The data can be obtained from the following URL: <http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>.

# Approach

The following is the approach I will take to attempt to solve the problem of default detection of credit card clients.

## Step 1: Prepare Data

The Data will be prepared in this step into training and testing sets in a 70/30 split. Neural networks require normalization and a min max method has been used.

## Step 2: Train Models

The models (the neural network model, the random forest model, and the decision tree model) will be trained using the training set prepared in step 1.

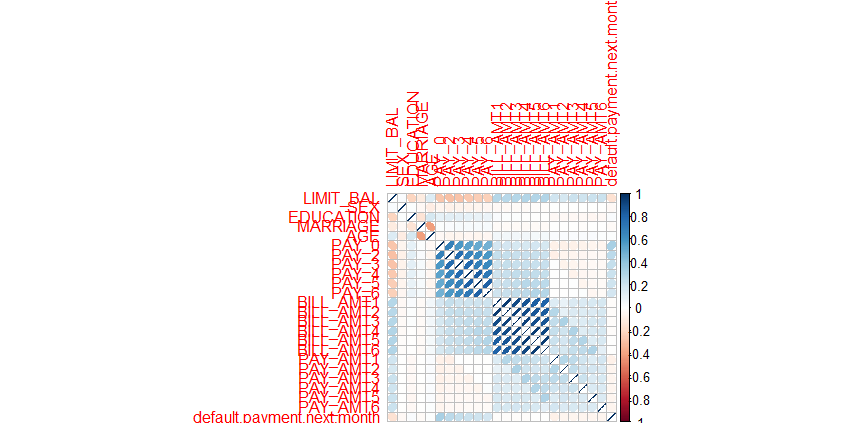
## Step 3: Evaluate Models

The models will be evaluated in this step using the testing set prepared in step 1. And conclusions on the results will be made.

Code used in the three steps can be found here: <https://github.com/xymosprime/CKME136/blob/master/CreditCard/CKME136_NN.Rmd>

# Results

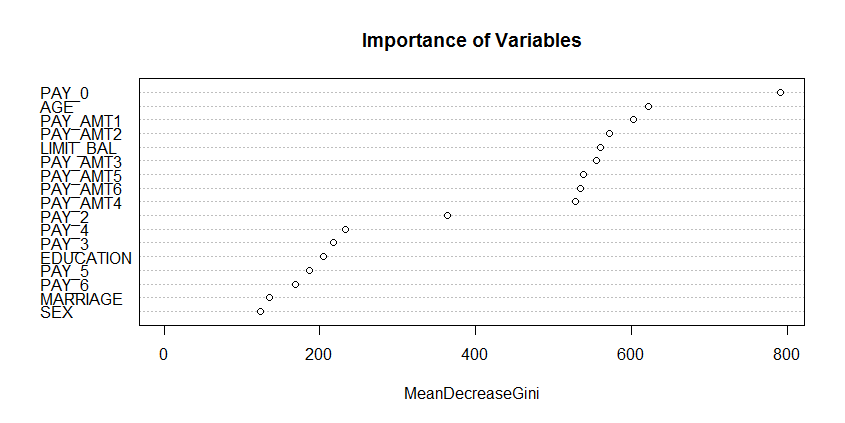
Before the models were built and explored a correlation between variables was taken to see if any columns could be ignored to safe computation time for some of the models.



As you can see that the *BILL\_AMT* variables have little to no effect on the prediction of *default.payment.next.month* and as such have been ignored in the Random Forest and Decision Tree models.

## Random Forest Model

Exploring the model highlights which variables have the largest effects on *default.payment.next.month*



Type of random forest: classification

Number of trees: 81

No. of variables tried at each split: 4

OOB estimate of error rate: 18.48%

Confusion matrix:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | class.error |
| 0 | 15415 | 978 | 0.05965961 |
| 1 | 2902 | 1704 | 0.63004776 |

The accuracy of the model was determined to be 81.5% ((15415+1704)/20999 x 100%).

When the model was applied to the testing set the following was the outcome:

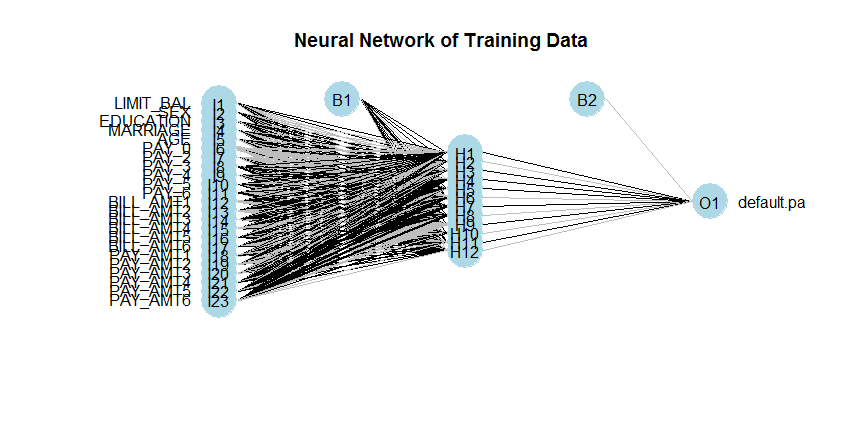
Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| rfPred | 0 | 1 |
| 0 | 6573 | 1266 |
| 1 | 398 | 764 |

Which results in a Prediction accuracy of 81.5% ((6573+764)/9001 x 100%)

## Neural Network Model

The Neural Network Model considered all of the variables except the ID variable which was already removed in the data preparation step. 12 Hidden Layers was chosen.



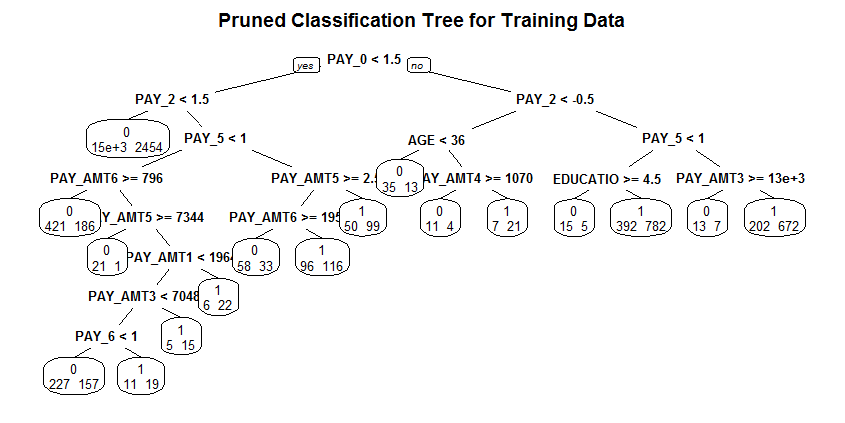
Neural Networks do not output prediction in 1 or 0 like Random Forests or Decision Trees. They give a prediction as a probability of being 1. The data is then rounded up if the predicted value is equal to or greater than 0.5, and rounded down if less than 0.5.

The accuracy of the model in correctly classifying whether a person would default or not was determined to be 82.3%.

This model was then applied to the Testing Set and a Prediction accuracy of 82.3% was concluded.

## Decision Tree Model

Considering every possible combination of 17 variables would provide a very convoluted, so I pruned the tree to only highlight the most important combinations (based on the given training set) to provide the best prediction model for *default.payment.next.month.*



Root node error: 4606/20999 = 0.21934

n= 20999

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CP | nsplit | rel error | xerror | xstd |
| 1 | 0.1799826 | 0 | 1.00000 | 1.00000 | 0.013019 |
| 2 | 0.0047764 | 1 | 0.82002 | 0.82002 | 0.012083 |
| 3 | 0.0032566 | 3 | 0.81046 | 0.82002 | 0.012083 |
| 4 | 0.0027139 | 4 | 0.80721 | 0.81937 | 0.012080 |
| 5 | 0.0015198 | 6 | 0.80178 | 0.81719 | 0.012067 |
| 6 | 0.0011579 | 8 | 0.79874 | 0.81633 | 0.012062 |
| 7 | 0.0010855 | 16 | 0.78789 | 0.81350 | 0.012046 |

|  |  |  |
| --- | --- | --- |
| dtPred | 0 | 1 |
| 0 | 15624 | 2860 |
| 1 | 769 | 1746 |

The accuracy of the model was determined to be 82.7%. Which can be determined in 2 ways:   
1. The root node error x rel error of the best cp x 100%  
2. The correctly classified rows / Total Rows of the set x 100%

(See Calculations below)

Accuracy of the model = 100 - ((0.21934 x 0.78789) x 100%) = 82.7%

Accuracy of the model = 100 - ((15624+1746)/20999) x 100% = 82.7%

Applying the model to the testing set yielded the following results:

|  |  |  |
| --- | --- | --- |
| dtPredTest | 0 | 1 |
| 0 | 6598 | 1280 |
| 1 | 373 | 750 |

A Prediction accuracy of 81.6% was determined from the confusion matrix (6598+750)/9001 x 100%)

# Model Comparison

# Conclusions

In conclusion it was determined that the Neural Network was the single best algorithm in the individual classifier category, which was in line with the literature review conducted. A prediction accuracy of 82.3% does indicate that the Neural Network is a good algorithm to use in default detection. The results were not as drastic as I would have liked, Neural Network only slightly edged out the other two reference models. Given more time I would have liked to explore some of the homogeneous ensemble classifiers, as the literature review highlighted that there was a tendency for them to outperform individual classifiers. The five best performing methods belonged to the homogeneous ensemble family.

# Bibliography

[1] Abbas Keramati and Niloofar Yousefi. *A Proposed Classification of Data Mining Techniques in Credit Scoring*. The Proceeding of the 2011 International Conference of Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia. 2011

[2] Stefan Lessmann, Hsin-Vonn Seow, Bart Baesens and Lyn C. Thomas. *Benchmarking state-of-the-art classification algorithms for credit scoring: A ten-year update*. Credit Research Centre, Conference Archive. 2013.

[3] Bart Baesens, Rudy Setiono, Christophe Mues and Jan Vanthienen. *Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation*. Management Science, v.49 n.3, p.312-329, March 2003.