Statistical Learning-Classification

Project Title: CLASSIFYING TAIWANESE CREDIT CARD DEFAULT STATUSES USING MODEL OPTIMIZATION TECHNIQUES

Project Number: 14

Group Members:

Surname, First Name	Student ID	STAT 441	STAT 841	CM 763	Your Dept. e.g. STAT, ECE, CS
Chow, Alaric	20517917	☒	□	□	STAT
Mehta, Karan	20512167	☒	□	□	STAT
Shin, Yoon Soo	20516955	☒	□	□	STAT
Tam, Melody	20518019	☒	□	□	STAT

Your project falls into one of the following categories. Check the boxes which describe your project the best.

1. □ **Kaggle project.** Our project is a Kaggle competition.
   - This competition is active □ inactive □.
   - Our rank in the competition is . . . .
   - The best Kaggle score in this competition is . . . . . . , and our score is . . . .
2. □ **New algorithm.** We developed a new algorithm and demonstrated (theoretically and/or empirically) why our technique is better (or worse) than other algorithms.
3. ☒ **Application.** We applied known algorithm(s) to some domain.

   ☒ We applied the algorithm(s) to our own research problem.

   ☒ We tried to reproduce results of someone else's paper.

   □ We used an existing implementation of the algorithm(s).

   □ We implemented the algorithm(s) ourself.

Our most significant contributions are (List at most three):

(a) . Detected a machine learning technique that best predicts the occurrence of defaulting.

(b) . Applied tuning and resampling methods to determine the optimal parameters for each technique.

(c) . Achieved comparable final results found in the referenced paper written by Yeh, I-Cheng, and Che-Hui Lien, which states that the best performing model is a Neural Network model with a 83% accuracy on their validation dataset.

List the name of programming languages, tools, packages, and software that you have used in this project:

.
Programming Language: R
Packages: Caret, CaTools, doParallel, kernlab, klaR, MASS, modelr, pROC, rpart, randomForest
Software: RStudio, Microsoft Office Suite

UNIVERSITY OF WATERLOO
Faculty of Mathematics

CLASSIFYING TAIWANESE CREDIT CARD DEFAULT STATUSES USING MODEL OPTIMIZATION TECHNIQUES

STAT 441 - Statistical Learning - Classification
Ali Ghodsi

Prepared by
Alaric Chow, ID 20517917
Karan Mehta, ID 20512167
Yoon Soo Shin, ID 20516955
Melody Tam, ID 20518019
April 19, 2018

Table of Contents

1. Data and Problem

The default of credit card clients classification problem is the research problem that we will be examining, implementing existing algorithms to predict whether clients are credible through binary default statuses. The data is taken from the University of California Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients). It contains 30,000 observations, from which we split the data into 24,000 training observations and 6,000 testing observations. Each observation represents a credit card client in Taiwan, and their information on default payments, demographic factors, credit data, and bill statements from April 2005 to September 2005.

The main goals of this study are to:

1) Detect a machine learning technique that best predicts the occurrence of defaulting.
2) Apply tuning and resampling methods to determine the optimal parameters for each technique.
3) Achieve comparable final results found in the referenced paper written by Yeh, I-Cheng, and Che-Hui Lien, which states that the best performing model is a Neural Network model with a 83% accuracy on their validation dataset.

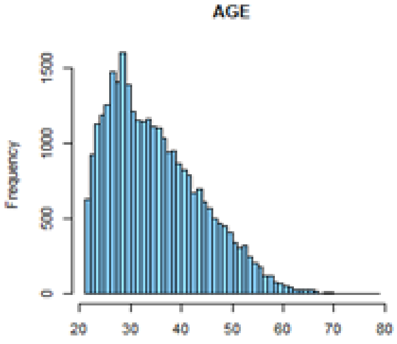
The provided data contains 23 independent variables. The numerical variables include Limit Balance, Age, monthly bill statement amounts from April 2005 to September 2005 (6 variables), and monthly previous bill payment amounts from April 2005 to September 2005 (6 variables). The categorical variables are Sex, Education, Marital Status, and monthly repayment statuses from April 2005 to September 2005 (6 variables).

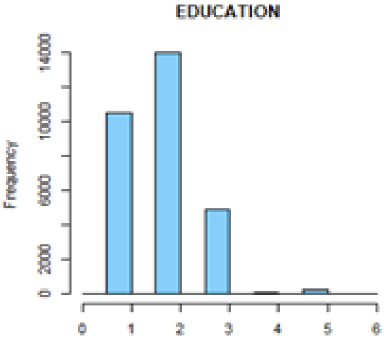
Refer to the link provided above to find a detailed explanation on the levels of each categorical variables. The research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable.

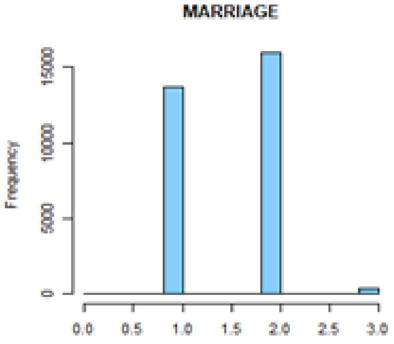
2. Exploratory Data Analysis

To better understand the dataset structure and variables, we decided to perform a series of analyses relating to how each variate affects the default status.

In particular, we wanted to determine if certain variables would require data pre-processing to improve the quality of our results. A series of histograms were used to illustrate certain numeric variables. The histogram for *Age* appeared left-skewed and as a result, we decided to band age into intervals of 5 years to account for lower frequencies and reduce the effects of observation outliers. As well, the histogram for *Education* showed low frequencies for the values of four to six, which correspond to either “other” or “unknown” values. The histogram for *Marriage* also showed low frequencies for values zero and three, which correspond to either “other” or unclassified. For these reasons, we decided to group these values together as one category labeled as “other” for both variates. The graphs can be seen below.







In our preliminary analyses, we also examined the relations between variables. This included looking at a correlation plot between bill payment, bill statement, and payment status variables. The plot was intuitively reasonable, as payment statuses strongly correlate with each other since individuals will tend to exhibit similar payment habits month to month. Bill amounts also show high positive correlations for the same reason.

Box plots were also examined for the population defining variables, *Age*, *Education*, *Sex*, and *Marriage*. From these plots, we concluded that across different categories for *Education, Sex, and Marriage*, and intervals for *Age*, the default status and limit balance is as expected for each respective category. Ultimately, different levels of *Education* and *Marriage* statuses will vary in limit balance and default status, but these differences are reasonable in each case. In addition, *Sex*, *Education*, *Marriage*, default status, and payment statuses were all coded as factors since these variables are categorical in nature and models will better recognize these variables as such.

3. Methodology and Approach

3.1. Reasoning for Using Caret Package

There are many modeling packages available in R for different classifying techniques. However, the main issue is that they were created by different entities thus; inconsistencies exist as to how the models are specified and how the predictions are made. Caret allows for standardized tuning and parameter evaluations for numerous machine learning models originating from different libraries. This allows for a more ideal environment for the comparison of the models we highlight in this project.

3.2. Optimizing Parameters using Receiver Operating Curves (ROC)

The ROC plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) for all possible thresholds, and uses the generated curve to evaluate the performance of a classifier. The area under the ROC or area under the curve (AUC) is the metric used for evaluating the performance of a classifier. The AUC can be translated into the probability that a classifier will rank a randomly chosen positive observation higher than a randomly chosen negative observation (assuming that positive ranks higher than negative). Thus, the larger the area is under a curve, the better the classifier. We utilize ROC in the tuning process by creating a profile of AUC scores for different parameter values of a specific model. This aids us in choosing the optimal parameters for each classifier. Due to its ability to showcase all thresholds, the AUC does not fall victim to unbalanced classes such as the dataset featured in this paper, which is the case with accuracy rates. Additionally, the ROC plot can allow us to evaluate optimal models at any given type I error cut-off.

3.3. Resampling Data using Repeated K-fold Cross Validation

K-Fold Cross Validation randomly splits the data into K equally sized parts. We removed the k^{th} block of the data and evaluated our selected classifier on the rest. The k^{th} block is retained as the validation set for testing the evaluated model. This is repeated for all K parts and those results are then averaged. We chose Repeated K-fold Cross Validation because we believe that by using different variations of the partitioning process in K-fold Cross Validation, we can reduce potential overfitting when evaluating our models. This is possible because we believe that our dataset encapsulates the bulk of possible variate combinations that we will need to predict. In the tuning and training of each model for this project, resampling was done using 10-Fold Cross Validation, repeated 5 times.

3.4. Machine Learning Techniques

The following machine learning techniques were selected for further analysis. The models were fit to both the full and reduced datasets, where the reduced dataset includes variables deemed relevant based on stepwise regression model fitting.

3.4.1. Generalized Linear Model (GLM)

Logistic Regression is a simple model with very easy to interpret results. It operates best when the labels are linear and will predict and train very quickly. The added benefit of this technique is that we can easily interpret the significance of each parameter unlike some machine learning techniques. This will be useful in identifying key variables for predicting default if necessary.

3.4.2. K-Nearest Neighbour (KNN)

KNN classifies and predicts based on feature similarities, where the K closest entries from the data point we wish to predict is analyzed. The most common classification is taken as the prediction. This model generalizes the data during the testing phase instead of the training phase, allowing it to respond quickly to changes even in real time. Additionally, this model handles outliers very well. However, it is sensitive to features with low relevance since it weighs all features equally relevant. Although it handles outliers very well, it is sensitive to the any localized abnormalities that it may encounter.

3.4.3. Random Forest (RF)

Random Forest utilizes ensemble learning in the form of numerous decision trees, where the final classification is based on the majority result. Each decision tree is created by sampling with replacement from the original dataset. Due to this method of sampling, decision trees are also decorrelated to some extent. However, random forests can overfit noisy datasets and due to the nature of training, the model is not easily interpretable if we wish to identify relevant predictors.

3.4.4. Neural Network (NNet)

Neural network utilizes backpropagation to update and optimize the weights, which will then tune the activation number in each neuron. The neuron with the highest activation number in the output layer is the classified output. Historically, the model yields fairly accurate results; however it takes a long time to fit and requires a large training dataset. Additionally, the model is not interpretable, and parameters cannot be fine tuned without rerunning.

3.4.5. Linear Support Vector Machine (LinearSVM)

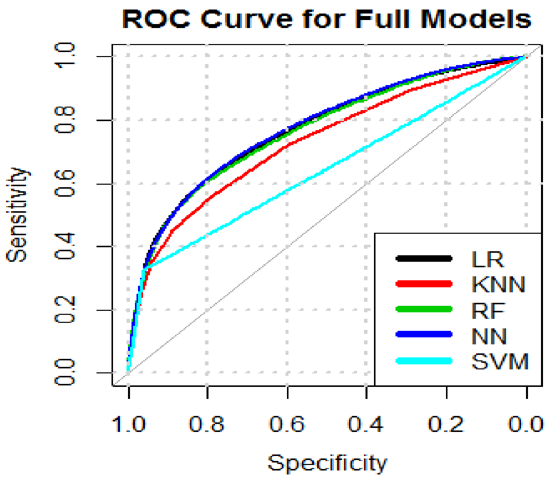
Support Vector Machine, with the help of a linear kernel, finds a hyperplane that discriminates between classes, and its goal is to have maximum separation (margin) between the classes. The Linear SVM model works well when when the data is in a high-dimensional space due to its compatibility with kernel methods. However, this classifier has issues if the data is noisy with odd outliers and overlapping classes.

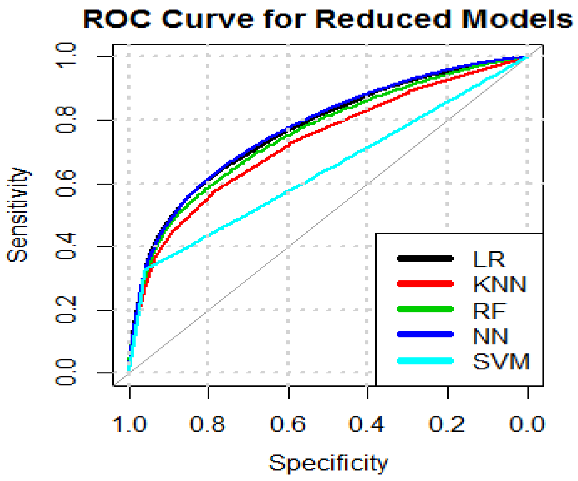
4. Analysis of Results

From our final results, the reduced and full versions of Neural Network have the best performing AUC results. The full model has a mean AUC value of 0.7732, mean sensitivity value of 0.9477, and mean specificity value of 0.3629. The reduced model has an mean AUC value of 0.7749, mean sensitivity value of 0.9484, and mean specificity value of 0.3647. The results are very similar for both models.

ROC Results			
Full Dataset		**Reduced Dataset**	
	Mean		**Mean**
Full KNN	0.7302	**Reduced KNN**	0.7315
Full Logistic Regression	0.7693	**Reduced Logistic Regression**	0.7697
Full Random Forest	0.7663	**Reduced Random Forest**	0.7581
Full Neural Network	0.7732	**Reduced Neural Network**	0.7749
Full SVM Linear	0.6737	**Reduced SVM Linear**	0.6784

We can look at the following ROC plot to verify our results. Notice that the Neural Network ROC curve is superior to all other models at all potential cutoff points. Similar results can be found for the reduced Neural Network model. Therefore, we conclude that the reduced Neural Network model is our best performing model.





5. Conclusion

Based on the data abstract found on UCI Machine Learning Repository for our dataset, it was said that, “Artificial Neural Network is the only one that can accurately estimate the real probability of default” (Yeh, I-Cheng, and Che-Hui Lien). This is consistent with our results which indicate that the best performing model is the Neural Network model.

The reduced Neural Network model achieved a prediction accuracy of 82.48% on our test dataset. In parallel, the paper written by Yeh, I-Cheng, and Che-Hui Lien cited on the UCI Machine Learning Repository achieved an accuracy rate of 83% on their validation dataset. We believe that the methodology outlined in this paper can efficiently replicate established results, while utilizing alternate resampling and parameter optimization techniques.

6. References

Benyamin, Dan. “CitizenNet Blog.” *A Gentle Introduction to Random Forests, Ensembles, and Performance Metrics in a Commercial System*, 9 Nov. 2012, blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics.

Bowne-Anderson, Hugo. “Preprocessing in Data Science (Part 1).” *DataCamp Community*, 26 Apr. 2016, www.datacamp.com/community/tutorials/preprocessing-in-data-science-part-1-centering-scaling-and-knn.

Brownlee , Jason. “Save And Finalize Your Machine Learning Model in R.” *Machine Learning Mastery*, 22 Sept. 2016, machinelearningmastery.com/finalize-machine-learning-models-in-r/.

Brownlee, Jason. “Get Your Data Ready For Machine Learning in R with Pre-Processing.” *Machine Learning Mastery*, 22 Sept. 2016, machinelearningmastery.com/pre-process-your-dataset-in-r/.

Brownlee, Jason. “K-Nearest Neighbors for Machine Learning.” *Machine Learning Mastery*, 22 Sept. 2016, machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/.

“Default of Credit Card Clients Data Set.” *UCI Machine Learning Repository: Default of Credit Card Clients Data Set*, 26 Jan. 2016, archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients.

Howson, Ian. “Models: A List of Available Models in Train in Caret: Classification and Regression Training.” *R Package Documentation*, 1 Apr. 2018, rdrr.io/cran/caret/man/models.html.

Kelly, Ryan. “Bagging, Random Forests, Boosting.” *Bagging, Random Forests, Boosting*, 14 July 2014, www.rmdk.ca/boosting_forests_bagging.html.

Kuhn, Max. “Predictive Modeling with R and the Caret Package.” *Universidad De Castilla-La Mancha*, 2013, www.edii.uclm.es/~useR-2013/Tutorials/kuhn/user_caret_2up.pdf+.

Prabhakaran, Selva. “Caret Package - A Practical Guide to Machine Learning in R.” *Machine Learning Plus*, 25 Mar. 2018, www.machinelearningplus.com/machine-learning/caret-package/.

Yeh, I-Cheng, and Che-Hui Lien. “The Comparisons of Data Mining Techniques for the Predictive Accuracy of Probability of Default of Credit Card Clients.” *Expert Systems with Applications*, vol. 36, no. 2, 2009, pp. 2473–2480., doi:10.1016/j.eswa.2007.12.020.