Credit Card Analysis

Vincent Pun

Abstract

Knowing when to accept or reject a credit application is an important decision that credit card companies have to make each day. It has been determined that granting credit to a bad customer is five times the cost of rejecting a good customer. To reduce costs and improve predictive accuracy, both traditional logistic regression and modern unsupervised learning methods such as autoencoders are explored in this paper, and this is used to make classification predictions on a German credit card company's dataset.

Keywords Credit cards, classification, logistic regression, unsupervised pre training, neural network, autoencoder, activation function, precision, recall

Introduction

The purpose of this research is to evaluate the use of both supervised and unsupervised learning techniques to classify customers as either good or bad in terms of their overall credit risk. The original dataset contains 1,000 observations and 21 variables that describe customer financial and demographic characteristics; exactly 30% of these observations are classified as "bad" customers, which is recorded in the "class" attribute. Historically, traditional methods such as logistic regression have been used to make this determination. Thus, we are conducting this research to learn whether the use of unsupervised learning methods will help produce more accurate predictions and reduce costs compared to our existing methods.

Literature review

Scalability and data size are known challenges in studies involving rarities and class imbalances (Pang, Shen, Cao, and Hengel, 2020). The determination of optimal proportions when the relative cost of a false-positive result is very much greater than the cost of a false-negative result was explored in studies regarding clinical medicine (Zweig and Campbell, 1993). Differences between activation functions and their ability to converge towards optimal solutions is explored in a paper by LeCun, Bottou, Orr, and Muller that was written in 1998. Cross-entropy and squared error in error training were previously compared while optimizing training of artificial neural networks (Golik, Doetsch, and Ney, 2013).

Methods

The first step in this research is to prepare the data. We remove the “female single” level from the personal_status attribute, as there are no observations for single females in this dataset. It is observed that there are levels in the “purpose” attribute that occur infrequently, and these are consolidated into a new level called “other.” Also, logarithmic transformation is performed on the “credit_amount” feature to reduce skewness prior to modeling. Histograms, quantile-quantile plots, and boxplots are generated to visualize credit_amount before and after transformation.

Figure 1: Distribution, Q-Q Plot, and Boxplot of credit_amount

The prepared data is then input to a baseline cross-validated logistic regression model, which will be used to assess the effectiveness of using autoencoders to reconstruct our model inputs. A function is created to predict the class attribute using all twenty-one explanatory variables, and this model is evaluated using precision, recall, F1 Score, and cost. An optimal cutoff between false-positive and false-negative classifications is complex and is calculated based on the knowledge that we have on the costs of accepting bad customers in addition to the proportion of bad customers within the dataset (see Appendix). Given that it is five times as expensive to grant bad customers credit than it is to erroneously reject a good customer, we focus on achieving higher precision rather than recall. F1 Score performs well when precision and recall are similar values, so this may not be the most appropriate evaluation method in this context.

When preparing the dataset for building autoencoders, the data is converted into a design matrix by expanding factor attributes to sets of dummy variables. It is then normalized on a common scale from 0 to 1 to prevent variables from having too much influence. There are now 44 explanatory variables that will be used when building the model. These observations are split into two subsets for training (80%) and testing (20%) purposes.

Autoencoders help visualize the credit data in a lower-dimension space as a set of embeddings. Unsupervised artificial neural networks are used to decode data, and this should help reduce risks of overfitting when building predictive models. This pretrained data is then used to see whether it generates better results. Three unique sequential models are built, and various hyperparameters such as the number of hidden layers, number of nodes, and loss functions are tested. Increasing the number of layers and nodes per layer is a method of increasing overall code size, which can make predictive models more powerful (Dertat, 2017). Also, the tanh activation function is preferred over the sigmoid activation function for this dataset, for it produced more accurate results across all modeling solutions. In general, we found that binary cross-entropy was a better loss function than mean squared error (MSE), as it is used to predict targets between 0 and 1. MSE would be better suited for regression problems where the desired output values are continuous, which is not the case for this study. Embedding vectors are generated by fitting the sequential models with the training data subsets, and they are tested as replacement input variables to the original classification model.

Additionally, an entirely unsupervised learning environment is used for comparison, where a model is trained only on observations of good customers. In result, this predictive model displays much higher MSE values for observations where we have bad customers, as they are introduced as anomalies compared to the fitted data. We use this to visualize the precision/recall tradeoff, and we can see that MSE rises as the cost of improving overall precision.

Figure 2: Precision Recall Tradeoff

Results

Model	**Hidden Layers**	**Nodes**	**Activation Function**	**Loss Function**	**F1 Score**	**Precision**	**Recall**	**Cost**
Baseline Logistic Regression	--	--	--	--	0.532	0.370	0.920	116.00 (best)
Model 1	3	15, 10, 15	Tanh	MSE	0.489	0.330	0.990	126.40
Model 2	5	30, 20, 10, 20, 30	Tanh	MSE	0.475	0.310	1.000	132.80
Model 3	5	30, 20, 10, 20,	Tanh	Binary Cross-en	0.490	0.330	0.990	125.80

		30		tropy				
Unsupervised Model	5	30, 20, 10, 20, 30	Tanh	Binary Cross-entropy	0.142	0.909	0.077	125.00

When regressing on the lower-dimension output generated by the autoencoder models, we observe that the baseline model is still the best performer. Increasing the code size does not seem to have a clear positive impact on predictive results, as seen between Model 1 and Model 2. However, the models that use binary cross-entropy as its loss function appear to outperform those that utilized MSE in terms of minimizing cost.

The fully unsupervised model utilizes a manually-determined MSE threshold, which is adjusted to minimize the total cost in this experiment. We set the MSE threshold at 3.2, which allows us to correctly identify 69 out of 70 bad customers in the test subset and correctly identify 10 out of 130 good customers. Despite having a high number of false-negative results, this model produces the second most cost-efficient results.

Figure 3: Unsupervised Learning MSE Determination

Conclusions

After comparing the performances of autoencoder-based solutions, we find that the traditional logistic regression model is still the best performer. Given these results, it is not recommended to use the pretraining-based classifiers that have been demonstrated in this research. We learned that anomaly detection is possible with these methods, but it may not be necessary if the purposes of this study are to improve prediction accuracy and reduce costs.

Appendix

Cost Cutoff Determination:

$m = (\frac{false-positive\ cost}{false-negative\ cost}) \times (\frac{1-P}{P})$ P = probability of being a bad customer

Figure 1 - credit_amount - Distribution, Quantile-Quantile Plot, Barplot:

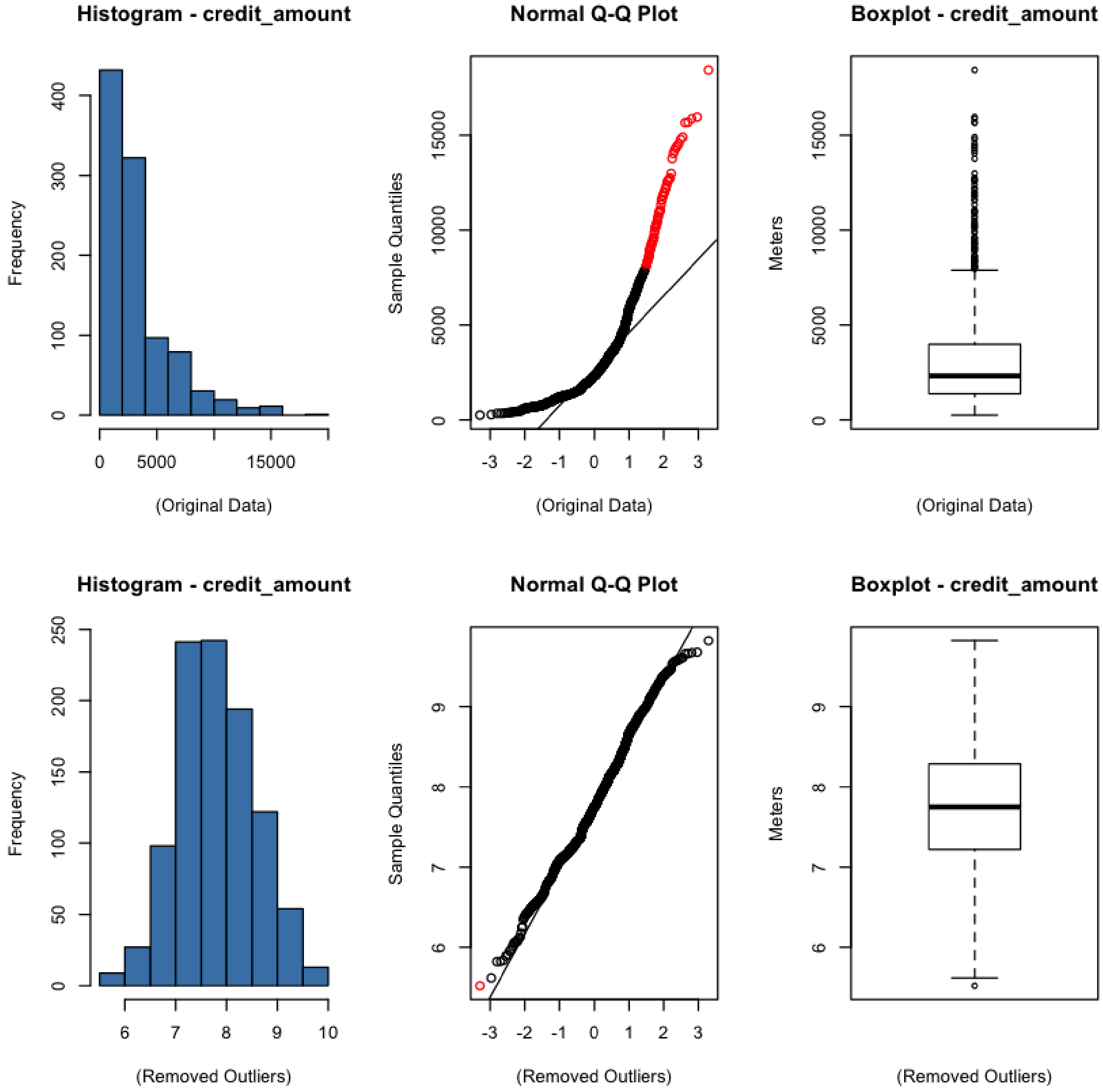


Figure 2 - Precision/Recall Tradeoff:

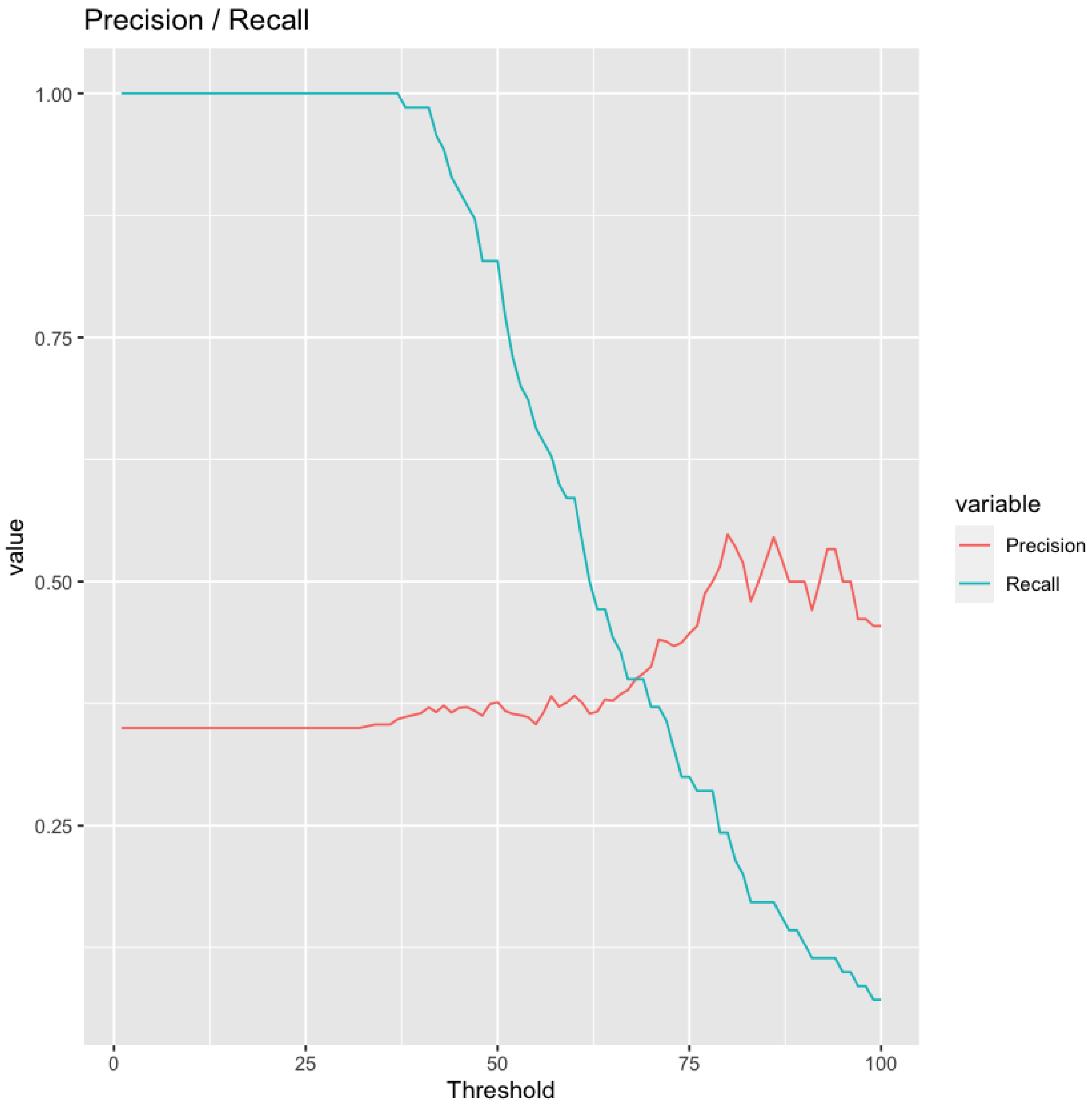
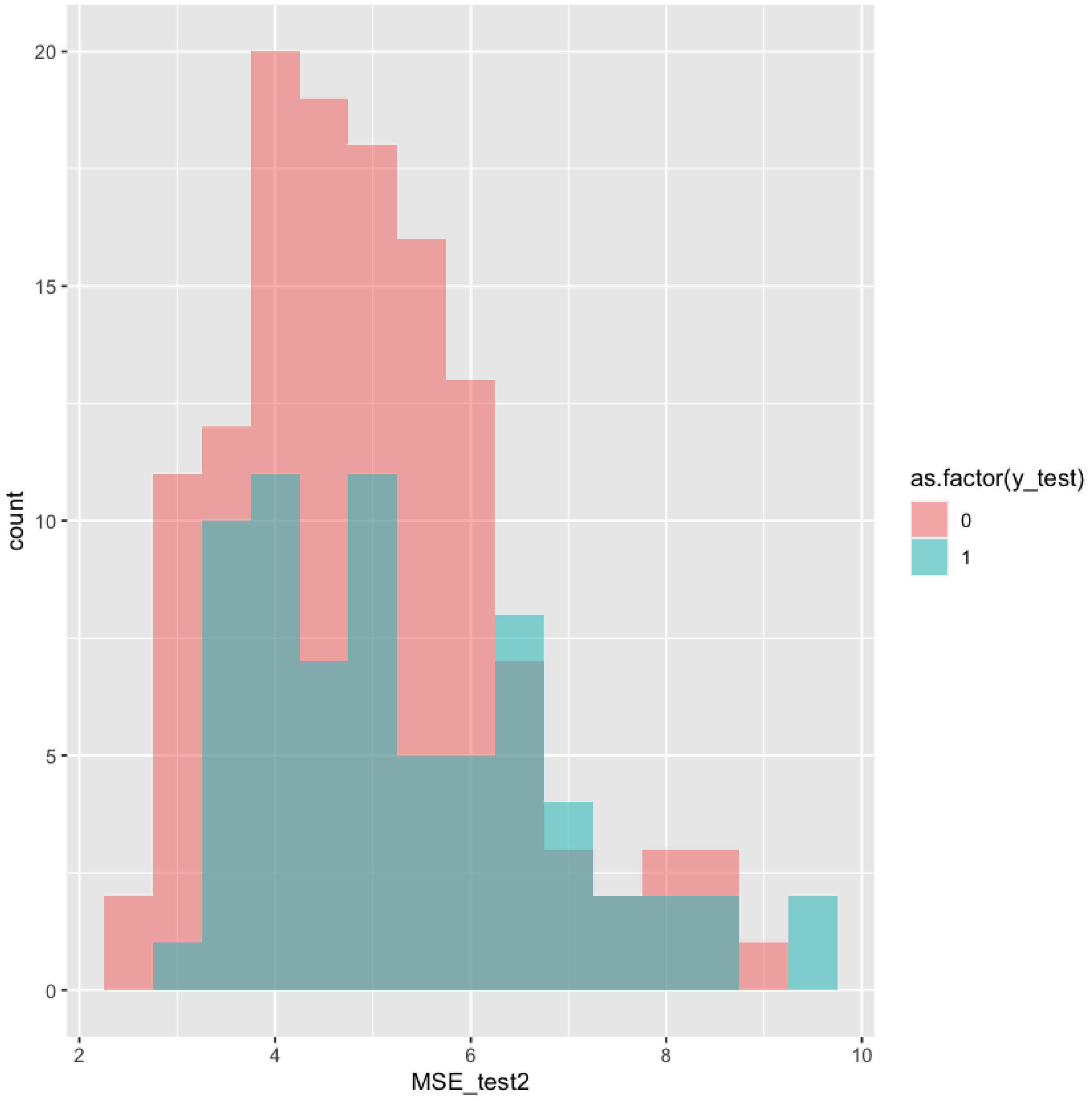


Figure 3 - Unsupervised Learning MSE Determination



References

Géron, Aurélien. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (second ed.), Sebastopol, Calif.: O'Reilly.

LeCun Y., Bottou L., Orr G.B., Müller K.R. (1998) Efficient BackProp. In: Orr G.B., Müller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 1524. Springer, Berlin, Heidelberg.

Hofmann, Hans. UCI Machine Learning Repository: Statlog (German Credit Data) Data Set. Institut f"ur Statistik und "Okonometrie, November 17, 1994. https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

Dertat, Arden. "Applied Deep Learning - Part 3: Autoencoders." Medium. Towards Data Science, October 8, 2017. https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798.

Zweig, Mark H. and Gregory Campbell. 1993. Receiver-Operating Characteristic (ROC) Plots: A Fundamental Evaluation Tool in Clinical Medicine. Clinical Chemistry, 39:4, 561-577. Retrieved from the World Wide Web on February 28, 2021, at http://www.floppybunny.org/robin/web/virtualclassroom/dss/articles/roc%20curves/roc_as_evaluation_tool_zweig_campbell_1993.pdf

Pang, Guansong, Chunhua Shen, Longbing Cao, and Anton van den Hangel. 2020. "Deep Learning for Anomaly Detection: A Review," ACM Computing Surveys. Retrieved from the World Wide Web, February 28, 2021, at https://arxiv.org/pdf/2007.02500.pdf

Ney, Hermann. "Cross-Entropy vs. Squared Error Training: a Theoretical and Experimental Comparison." RWTH Aachen University, August 2013. https://www.researchgate.net/publication/266030536_Cross-Entropy_vs_Squared_Error_Training_a_Theoretical_and_Experimental_Comparison

Versloot, Christian. "ReLU, Sigmoid, Tanh: Activation Functions for Neural Networks." MachineCurve, January 17, 2021. https://www.machinecurve.com/index.php/2019/09/04/relu-sigmoid-and-tanh-todays-most-used-activation-functions/.