Braden Bourgeois
ISDS 7302
Assignment 2
4/19/2025

Forecasting Bad Loans using Equity Data

Introduction and Business Understanding: Banks rely on predictions in order to assess loan risk and limit their losses. This assignment aims to see which modeling approach performs best at predicting whether a borrower will default on a loan, BAD = 1, using the equity data given. The business objective is to reduce loan risk exposure by identifying potentially bad loans early. At the same time, it is necessary to avoid false positives, which is rejecting a borrower who would have paid it back. In this assignment, a Decision Tree, Bootstrap Forest, Boosted Tree, and Neural Network model will be made, assessed, and compared across many different performance metrics.

Data Understanding: The original dataset included 5,960 records. 71 outliers were removed from the dataset. After looking at and excluding missing values, 2,628 rows were excluded from any analysis. The explanation tab in the Excel file gave us information on the predictors that would be used. It was quickly determined that CLNO would not be used as a predictor, as it is just a client's loan number and should not be used for prediction. After performing many initial models, the predictor named reason also seemed to worsen model performance and was not included in any predictive models. The dataset was split into 60% training, 20% validation, and 15% test sets.

While it was recommended to bin the continuous variables before modeling, this step is unnecessary for decision tree-based methods. After experimenting with binning variables and certain combinations of variables by using Partitioning Splits in JMP, models such as the Decision Tree, Bootstrap Forest, and Boosted Tree consistently performed worse than when they were not binned. After some research, I found that re-binning, especially on continuous variables, can actually reduce model performance by stripping away variation that tree-based algorithms are designed to exploit. My models using raw continuous inputs outperformed versions with manually binned variables. Variable splits used on predictors with interactions were used in the final models though.

Analysis: Four models were evaluated. The Decision Tree model was made first. This basic interpretable model showed lower predictive power compared to the other methods. It achieved moderate sensitivity and specificity but had the lowest AUC, as shown in Figure 2. The Bootstrap Forest was performed next. This model performed the best overall, achieving the highest AUC (0.8687). It also had the strongest lift curve performance, although they were all very close (Figure 3). It had near specificity and a strong cumulative gains curve (Figure 5), identifying high-risk loans early. The Boosted Tree was much better than the basic Decision Tree. This model showed improvements over the basic decision tree, it still lagged behind Bootstrap Forest in AUC and lift.

Many Neural Network models were performed next. These models were compared based on their misclassification rate on Validation data. The model chosen was a Model NTanH(3) with a slow learning rate, 100 tours, and the transform covariates option on. This model had the highest sensitivity (30%) as shown in Figure 1, making it the best at detecting bad loans. It correctly identified 30% of the bad loans. It also maintained a solid AUC of 0.8186 (Figure 2) and outperformed tree models in mid-range deciles of the lift curve (Figure 3).

Performance across all models was summarized using standard fit statistics (Figure 4), where the Bootstrap Forest stood out with the best validation error and R-squared.. The Neural

Network was second-best overall in terms of predictive metrics, benefiting from its higher sensitivity and AUC (Figure 1). While Neural Networks are sometimes prone to overfitting, this model performed consistently across validation and test sets, suggesting that proper tuning and the transformation of covariates worked. The Boosted Tree struggles with overfitting as the AUC drops from 0.9168 on training to 0.7664 on validation and the R^2 also drops significantly, from 0.5429 to 0.2545 (Figure 4). The training misclassification rate is only 3.6%, but validation jumps to 7.5%, which shows it fits to the training data much more tightly than the validation data. Overfitting was paid attention to and reduced as much as possible when making each model. The Decision Tree model had the weakest fit statistics and the lowest AUC.

Quality of Models: All models had near 100% specificity, meaning they made almost no false positive predictions. Sensitivity differed between the models. The Neural Network had the highest sensitivity at 30%, meaning it identified the most actual bad loans (Figure 1). The Bootstrap Forest had the strongest performance in terms of AUC, lift, and cumulative gains (Figures 2, 3, and 5). Despite having lower sensitivity, the Bootstrap Forest can rank risky borrowers early, making it valuable depending on the situation. The Boosted Tree was promising, but showed signs of overfitting, meaning it may generalize poorly. The Decision Tree in general, was underperforming across every metric.

Conclusion: Choosing the best model depends on the specific business goal in mind. If the priority is to identify as many risky borrowers as possible, the Neural Network is the best at that. If the goal is ranking high-risk applicants with high overall accuracy and stability, the Bootstrap Forest would be the better choice. Each model brings their different strengths, and the best choice depends on whether the business cares more about catching bad loans or avoiding turning away good customers.

Appendix

Figure 1 - Confusion Matrix

Model	Sensitivity	Specificity	F. Neg.	F. Pos.	Error	AUC
Decision Tree	26%	100%	7%	6%	7%	0.712
Bootstrap	16%	100%	8%	0%	8%	0.869
Boosted Tree	23%	100%	7%	7%	7%	0.766
Neural Netwrok	30%	100%	7%	5%	7%	0.819

Figure 2 - ROC Curves

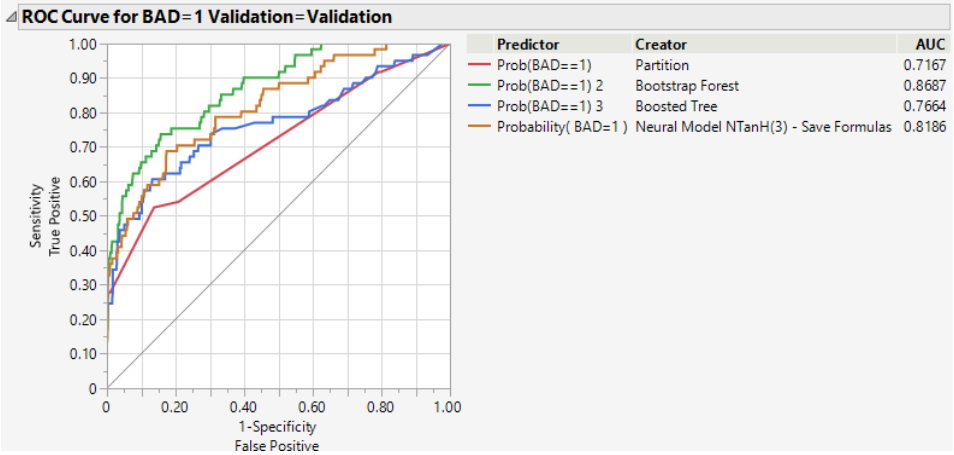

Figure 3 - Lift Curves

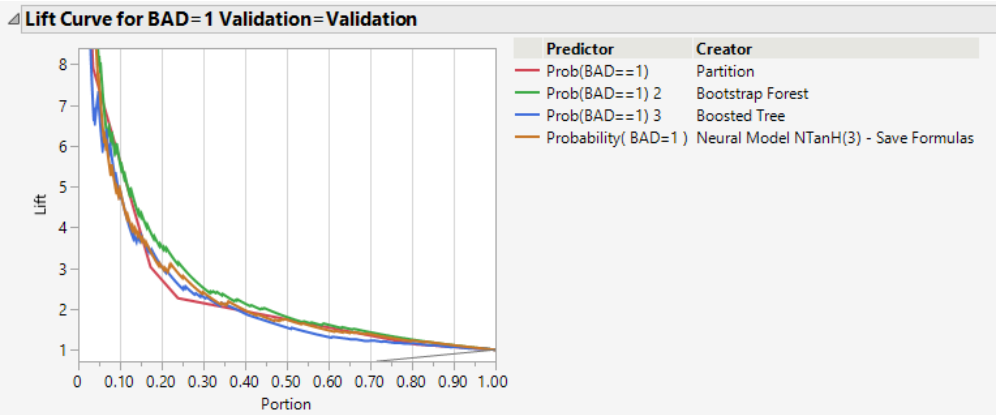

Figure 4 - Measures of Fit

Measures of Fit for BAD

Validation	Creator	.2 .4 .6 .8	Entropy RSquare	Generalized RSquare	Mean -Log p	RASE	Mean Abs Dev	Misclassification Rate	N	AUC
Training	Partition		0.3517	0.4111	0.1654	0.2008	0.0817	0.0441	1997	0.8187
Training	Bootstrap Forest		0.6915	0.7439	0.0787	0.1498	0.0582	0.0431	1997	0.9989
Training	Boosted Tree		0.4793	0.5429	0.1329	0.1786	0.0760	0.0381	1997	0.9168
Training	Neural Model NTanH(3) - Save Formulas		0.3013	0.3566	0.1783	0.2110	0.0968	0.0491	1997	0.8349
Validation	Partition		0.1571	0.2009	0.2616	0.2555	0.1053	0.0704	653	0.7167
Validation	Bootstrap Forest		0.3016	0.3691	0.2168	0.2455	0.1007	0.0781	653	0.8687
Validation	Boosted Tree		0.2017	0.2545	0.2478	0.2555	0.1104	0.0735	653	0.7664
Validation	Neural Model NTanH(3) - Save Formulas		0.2444	0.3044	0.2345	0.2468	0.1149	0.0674	653	0.8186
Test	Partition		0.1965	0.2496	0.2555	0.2576	0.1078	0.0733	682	0.7554
Test	Bootstrap Forest		0.3484	0.4223	0.2072	0.2411	0.0980	0.0748	682	0.8976
Test	Boosted Tree		0.2737	0.3394	0.2309	0.2506	0.1084	0.0733	682	0.8380
Test	Neural Model NTanH(3) - Save Formulas		0.2338	0.2936	0.2436	0.2538	0.1175	0.0704	682	0.8135

Figure 5 - Cumulative Gains Curves