Bret Peterson

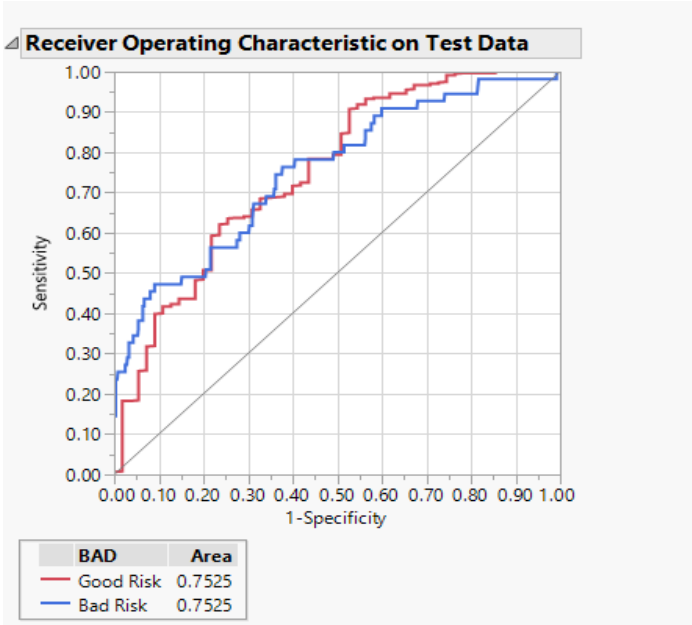
BAN-525

6/30/2019

Final Project: Predicting “Bad” Credit Risk

When a bank considers whether to approve a loan applicant, one of the most important factors for the bank to consider is whether that applicant will be a “bad” credit risk and default on the loan. I will use previous applicant data titled, “Equity.JMP”, in order to create a model to better predict the credit risk of a loan applicant. I will use a Nominal Logistic model as my base model to compare other models to, the second model I will run will be a Boosted Tree model, and the final model I will run will be the Bootstrap Forest. After running these models, using JMP’s unique, “informative missing variable” selected, I will then run the same models with it deselected as this may change my results. Once the models have been calculated, I will then conduct a model comparison to determine which model predicts “Bad Credit Risk” the best. Following the selection of the most accurate model, I will then analyze the individual variables to calculate which variable(s) have the largest impact on the model.

Before conducting my first model, I created a Validation column splitting the 60/20/20, using a fixed random seed of 123 (to be able to repeat model results). Next, I chose my first method to analyze the data, the Nominal Logistic (NL) model. I chose BAD as the independent variable to be tested as we are trying to determine bad credit risk. I tested BAD against all variables in the dataset (excluding BAD).



The second model I chose to model the data was the Boosted Trees method. This method is advantageous in that it can be used for both classification and regression problems; however, Boosted Trees can sometimes overfit the data. Again, I chose “BAD” as my independent variable and tested it against the remaining variables, using “Validation” as my validation column. After the specification page came up, I kept all the JMP default values. I did select “Multiple Fits over Splits and Learning Rate” so that JMP will repeat the testing process and potentially select a better model. I also selected “Suppress Multithreading” and entered a random seed of 123 in order to be able to reproduce my results. Also, I select the “informative missing” option which fills in missing information from the dataset.

Boosted Tree for BAD: Informative Missing Selected

Boosted Tree for BAD

Specifications

Target Column:	BAD	Number of training rows:	3576
Validation Column:	Validation	Number of validation rows:	1192
Number of Layers:	50	Number of test rows:	1192
Splits per Tree:	3		
Learning Rate:	0.1		
Overfit Penalty:	0.0001		

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.4094	0.3979	0.3964	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.5273	0.5290	0.5191	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.2852	0.3252	0.3052	∑ -Log(ρ[j])/n
RMSE	0.2855	0.3126	0.2992	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.1945	0.2141	0.2024	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1077	0.1351	0.1216	∑ (ρ[j]≠ρMax)/n
N	3576	1192	1192	n

Confusion Matrix

Training

Actual BAD	Predicted Count Good Risk	Predicted Count Bad Risk
Good Risk	2869	36
Bad Risk	349	322

Validation

Actual BAD	Predicted Count Good Risk	Predicted Count Bad Risk
Good Risk	905	12
Bad Risk	149	126

Test

Actual BAD	Predicted Count Good Risk	Predicted Count Bad Risk
Good Risk	936	13
Bad Risk	132	111

After running the Boosted Tree model with the “informative missing” option selected, I also ran it a second time deselecting this option.

Boosted Tree for BAD: Informative Missing Deselected

Boosted Tree for BAD

Specifications

Target Column:	BAD	Number of training rows:	3576
Validation Column:	Validation	Number of validation rows:	1192
Number of Layers:	50	Number of test rows:	1192
Splits per Tree:	3		
Learning Rate:	0.1		
Overfit Penalty:	0.0001		

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.1639	0.0733	0.0823	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.2364	0.1153	0.1255	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.4036	0.5005	0.4641	∑ -Log(ρ[j])/n
RMSE	0.3528	0.4052	0.3859	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.2620	0.3018	0.2855	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1720	0.2173	0.1946	∑ (ρ[j]≠ρMax)/n
N	3576	1192	1192	n

Confusion Matrix

Training

Actual BAD	Predicted Count Good Risk	Predicted Count Bad Risk
Good Risk	2897	8
Bad Risk	601	70

Validation

Actual BAD	Predicted Count Good Risk	Predicted Count Bad Risk
Good Risk	913	4
Bad Risk	255	20

Test

Actual BAD	Predicted Count Good Risk	Predicted Count Bad Risk
Good Risk	944	5
Bad Risk	227	16

The third model that I chose to model the data was the Bootstrap or Decision Tree method. I chose a random forest model as it creates an accurate model without an excess need to use hyper-parameter tuning and can be used on both classification and regression data. Using the same specifications as the previous models, I used BAD as my independent variable and tested it against the remaining variables, using “Validation” as my validation column. Once at the Bootstrap Specification window, I left all JMP defaults but did select “Suppress Multithreading” and applied a random seed of 123 in order to be able to repeat the results. I also selected the “informative missing” option.

Bootstrap for BAD: Informative Missing Selected

Bootstrap Forest for BAD

Specifications

Target Column:	BAD	Training Rows:	3576
Validation Column:	Validation	Validation Rows:	1192
		Test Rows:	1192
Number of Trees in the Forest:	100	Number of Terms:	12
Number of Terms Sampled per Split:	3	Bootstrap Samples:	3576
		Minimum Splits per Tree:	10
		Minimum Size Split:	5

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.6843	0.5222	0.5315	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.7809	0.6528	0.6535	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.1524	0.2581	0.2369	$\sum$ -Log(ρ[j])/n
RMSE	0.2002	0.2801	0.2655	$\sqrt{\sum(y[j]-\rho[j])^2/n}$
Mean Abs Dev	0.1203	0.1718	0.1587	$\sum \lvert y[j]-\rho[j]\rvert/n$
Misclassification Rate	0.0481	0.1166	0.0923	$\sum$ (ρ[j]≠ρMax)/n
N	3576	1192	1192	n

Confusion Matrix

Training

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	2891	14
Bad Risk	158	513

Validation

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	892	25
Bad Risk	114	161

Test

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	929	20
Bad Risk	90	153

After running the Bootstrap model with the “informative missing” option selected, I also ran it a second time deselecting this option.

Bootstrap for BAD: Informative Missing Deselected

Equity - Bootstrap Forest of BAD 2 - JMP Pro

Bootstrap Forest for BAD

Specifications

Target Column:	BAD	Training Rows:	3576
Validation Column:	Validation	Validation Rows:	1192
		Test Rows:	1192
Number of Trees in the Forest:	27	Number of Terms:	12
Number of Terms Sampled per Split:	3	Bootstrap Samples:	3576
		Minimum Splits per Tree:	10
		Minimum Size Split:	5

Overall Statistics

Measure	Training	Validation	Test	Definition
Entropy RSquare	0.4007	0.2797	0.2938	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.5182	0.3948	0.4040	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.2893	0.3891	0.3572	∑ -Log(ρ[j])/n
RMSE	0.2992	0.3577	0.3398	√ ∑(y[j]-ρ[j])²/n
Mean Abs Dev	0.1985	0.2492	0.2311	∑ |y[j]-ρ[j]|/n
Misclassification Rate	0.1451	0.2039	0.1795	∑ (ρ[j]≠ρMax)/n
N	3576	1192	1192	n

Confusion Matrix

Training

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	2905	0
Bad Risk	519	152

Validation

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	917	0
Bad Risk	243	32

Test

Actual	Predicted Count	
BAD	Good Risk	Bad Risk
Good Risk	948	1
Bad Risk	213	30

The next step I took after running the three models; Nominal Logistic, Boosted Trees, and Bootstrap, was to run a model comparison to determine which model is the best predictor of Bad Credit. In doing so, I used JMP's model comparison tool, selecting "Prob(BAD)" for all three models (and the informative missing deselected) as my Y-Predictor and assigned Validation to group.

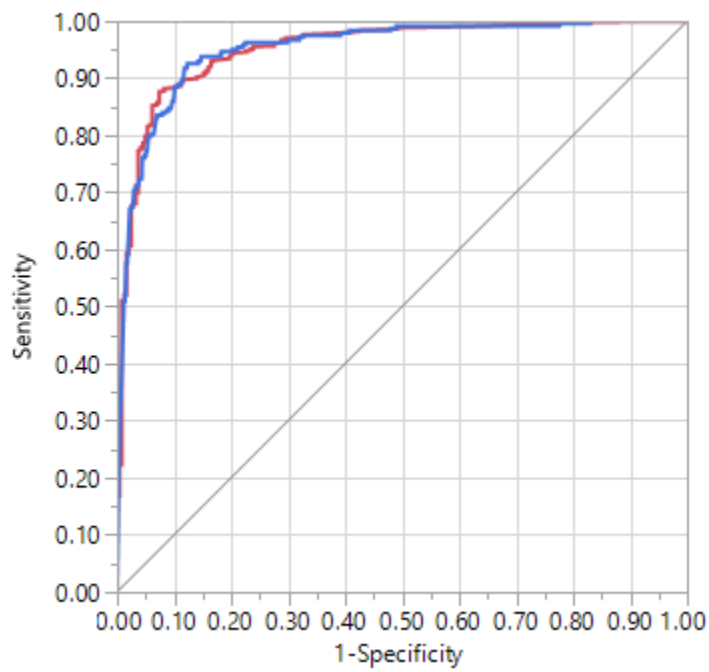
Model Comparison

Validation	Creator	Entropy RSquare	Generalized RSquare	Mean -Log p	RMSE	Mean Abs Dev	Misclassification Rate	N
Training	Bootstrap Forest	0.5508	0.6661	0.2169	0.2444	0.1672	0.0808	3576
Training	Boosted Tree	0.2914	0.3960	0.3421	0.3141	0.2512	0.1398	3576
Training	Bootstrap Forest	0.6843	0.7809	0.1524	0.2002	0.1203	0.0481	3576
Training	Boosted Tree	0.4094	0.5273	0.2852	0.2855	0.1945	0.1077	3576
Training	Fit Nominal Logistic	0.2263	0.2813	0.2317	0.2470	0.1240	0.0736	2053
Validation	Bootstrap Forest	0.3682	0.4968	0.3413	0.3267	0.2295	0.1569	1192
Validation	Boosted Tree	0.3380	0.4631	0.3576	0.3229	0.2605	0.1258	1192
Validation	Bootstrap Forest	0.5222	0.6528	0.2581	0.2801	0.1718	0.1166	1192
Validation	Boosted Tree	0.3979	0.5290	0.3252	0.3126	0.2141	0.1351	1192
Validation	Fit Nominal Logistic	0.2880	0.3558	0.2268	0.2468	0.1246	0.0740	649

Validation	Creator	Entropy RSquare	Generalized RSquare	Mean -Log p	RMSE	Mean Abs Dev	Misclassification Rate	N
Test	Bootstrap Forest	0.4022	0.5253	0.3023	0.3046	0.2088	0.1409	1192
Test	Boosted Tree	0.3196	0.4341	0.3441	0.3147	0.2525	0.1267	1192
Test	Bootstrap Forest	0.5315	0.6535	0.2369	0.2655	0.1587	0.0923	1192
Test	Boosted Tree	0.3964	0.5191	0.3052	0.2992	0.2024	0.1216	1192
Test	Fit Nominal Logistic	0.1739	0.2173	0.2365	0.2490	0.1255	0.0680	662

After analyzing the model comparison on the unbiased test data, it is easy to determine that the "Bootstrap Forest" model is the most accurate model having the highest R-square (0.5315), a low RMSE (0.1587), and a low misclassification rate (0.0923).

Knowing that the Bootstrap Forest is my best model, I will next interpret that model.



BAD	Area
Good Risk	0.9528
Bad Risk	0.9528

Looking at the Area Under the Curve (AUC), the Bootstrap forest appears to be very accurate.

I also wanted to determine the variable importance of the model. In order to do this, I looked at the prediction profilers and then investigated the individual uniform inputs of the variables.

Variable Importance: Independent Uniform Inputs

Column	Main Effect	Total Effect
DEBTINC	0.536	0.59
DELINQ	0.194	0.231
CLAGE	0.056	0.087
DEROG	0.054	0.084
VALUE	0.024	0.051
CLNO	0.01	0.024
MORTDUE	0.005	0.017
NINQ	0.007	0.015
JOB	0.007	0.014
LOAN	0.006	0.014
YOJ	0.005	0.012
REASON	0.004	0.008

As can be seen in the table above, the two variables with the largest impact on the model are DEBTINC and DELINQ, making up for 59% and 23% of the impact. Together, the two variables combined account for 82% of the impact in the model proving that these two variables are the highest predictors of bad credit risk.