

### **Classifying Credit Risk for Home Equity Loan Customers**

In this assignment, I wanted to investigate these factors in conjunction with several other variables in a dataset containing the 5960 applicants for home equity loans and their reported credit risk as the binary response variable contained in the BAD column, being either “Good Risk” or “Bad Risk”. Overall, there were 12 predictor variables, comprising a blend of both continuous and categorical data with varying number of levels:

LOAN: The total value of the loan.

MORTDUE: Mortgage due.

VALUE: (continuous variable).

REASON: Either Home Improvement or Debt Consolidation.

JOB: Broad job category (Self, Sales, ProfExe, Office, Mgr, or Other).

VOJ: Years on the job.

DEROG: Number of derogatory reports.

DELINQ: Number of delinquent trade line.

CLAGE: Age of oldest trade line.

NINQ: Number of recent credit inquiries.

CLNO: Number of trade lines.

DEBTINC: Debt-to-Income as a percentage.

Home Equity Loans generally have low rates, given their long repayment terms (NerdWallet, 2023). In exchange for these favorable terms, the applicant’s house is set as collateral for the loan, in which case failure to repay the loan would result in the lender foreclosing the home. Typically, applicant credit history, income, and their home’s market value all factor into both the sum of the loan and the stipulated interest rate. This dataset offers a broader image of how

other factors can play into an applicant's perceived credit risk, like their profession and the intended purpose of their loan.

This dataset presents a problem that has not been addressed in previous assignments since there are many instances of missing data. The JMP software comes equipped with the unique "informative missing" feature when running predictive models that serves to address this issue by treating rows with missing data as a separate class than the rest, effectively running the models twice to evaluate how the omission of some responses effects the predictions. In this way, analysts can explore the question as to why respondents might be omitting certain answers when applying for home equity loans. In particular, the Debt/Income ratio variable, DEBTINC, contains many instances of missing data, where 1267 of the 4693 responses were left blank for this variable, amounting to just over a quarter of all applicants.

My focus in this assignment is to explore how well predictive models perform when there is missing data by comparing models with the "Informative Missing" option selected to their otherwise identical counterparts for each method. So, I used three predictive modeling techniques with both versions: the Boosted Tree (Boosted Decision Tree), the Bootstrap Forest, and the Boosted Neural Network.

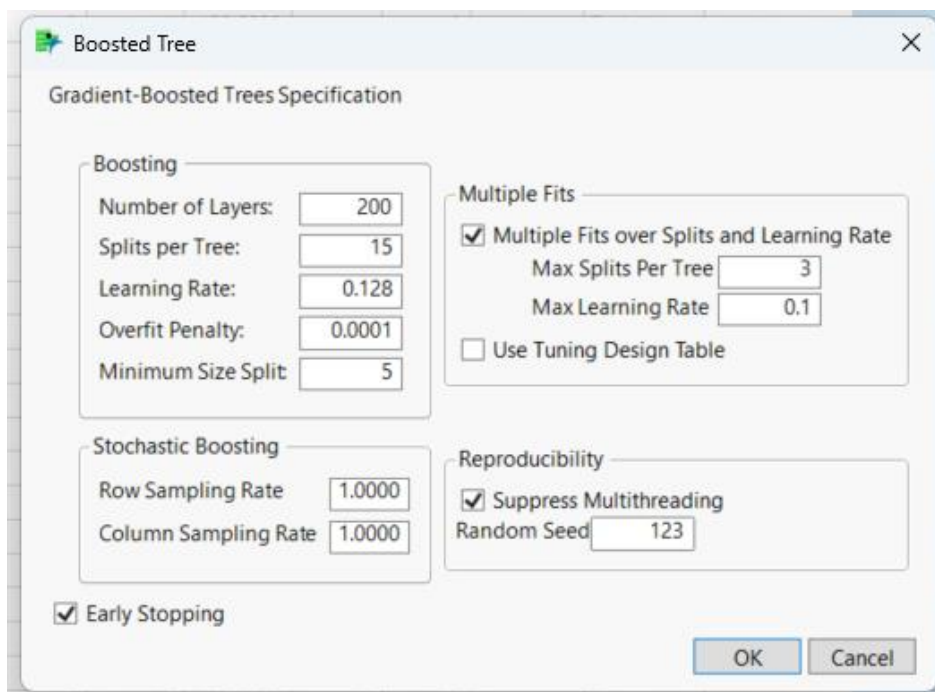
The Boosted Tree and Boosted Neural Network both share the commonality of boosting in their constructions. Boosting is the idea of having a model learn from its mistakes (misclassified observations in this case, but otherwise known as residuals in a regression problem) and rebuild the model by giving the errors more weight in each sequential iteration of the model until the models predictions on the validation data of the data cease to improve. The majority vote of all the weighted "boosts" predictions in this classification problem are then taken to produce the final version of the boosted model's predictions.

The Bootstrap Forest (commonly called the Random Forest method), unlike the other two, employs a bagging method of ensembling where instead of weighted averages (for regression problems) and weighted majority vote (in this case for classification problems), the model produce final predictions for the applicants credit risk as a simple majority vote for each "tree", essentially as a collection of many decision trees without applying any weights on

misclassifications (equivalent to residuals in regression problems). The idea of taking a collection of many independent trees, in principle, serves to avoid overfitting.

One particular criticism of the ordinary Decision Trees is how the individual trees tend to be highly correlated, which could lead to overfitting the model and poorly performing on new data. Luckily, the boosted version of Decision Trees avoids this phenomenon since it runs a large number of models and then takes the majority vote or averages the predictions. In this assignment, I specified for all Decision Trees (labeled as Boosted Trees in my tables) to have 200 “boosts” for this reason, which should also help to yield very similar results if rerun.

The screenshot below shows the model dialogue for the Boosted Tree where I maintained the default values for many of the options and set the random seed to 123. The “Early Stopping” option helps ensure that JMP stop the model when it stops improving its predictive capacity on the validation data, as opposed to running all 200 boosts, thus partly controlling for how computationally intensive this method can become.



In addition to the 3 models explained earlier, I also created 2 models that don't offer the "Informative Missing" option when constructing the models: A Nominal Logistic model and a K-Nearest Neighbors (KNN) model. The reasoning behind including the Nominal Logistic model in this discussion is to show the vast improvement in predictions derived from a simpler methodology to more complex methods, as well as to showcase the true power of the "Informative Missing" feature in JMP. KNN models, although simpler than ensemble methods, are a powerful method that can handle missing data in JMP by using the "nearest neighbor imputation" method. In this method, missing values are filled by assigning rows to certain classes for missing values in categorical variables and an average for continuous variables (the number of neighbors used to impute the values is set as K's value in the model dialogue box in JMP). The default K value for the model dialogue was 10. In simple terms, with a K value of 10, JMP would impute the missing value for an applicant's response by looking at the nearest 10 "neighbors" for this data point and replace the missing value with the majority vote or average, dependent on the type of data for each column.

Since this dataset contains cross sectional data, and not a time series dataset, my cross-validation column had a 60-20-20 split for the training, validation, and testing splits, respectively. I also used a random seed of 123 for the purpose of replicability of my models.

I ran all 8 models into the model comparison tool where my focus was on the Misclassification Rates for the Test split. Since the R-squares and RASE, which are only informative for regression problems, were logically empty, I copied the Misclassification rates into the simplified table below, using Excel to highlight where three results stood out:

Validation	Creator	Misclassification Rate
Training	Binominal Logistic	0.4835
Training	Bosted Tree Informative Missing	0.0218
Training	Boosted Tree	0.1342
Training	Bootstrap Forest	0.08
Training	Bootstrap Forest Informative Missing	0.0431
Training	Boosted Neural Network	0.46
Training	K-Nearest Neighbors	0
Training	Boosted Neural Network Informative Missing	0.0626
Validation	Binominal Logistic	0.4681




Validation	Boosted Tree Informative Missing	0.0713
Validation	Boosted Tree	0.1242
Validation	Bootstrap Forest	0.1376
Validation	Bootstrap Forest Informative Missing	0.0931
Validation	Boosted Neural Network	0.4513
Validation	K-Nearest Neighbors	0.0587
Validation	Boosted Neural Network Informative Missing	0.0747
Test	Binominal Logistic	0.4815
Test	Boosted Tree Informative Missing	0.0805
Test	Boosted Tree	0.1426
Test	Bootstrap Forest	0.1233
Test	Bootstrap Forest Informative Missing	0.1091
Test	Boosted Neural Network	0.4664
Test	K-Nearest Neighbors	0.0663
Test	Boosted Neural Network Informative Missing	0.0831

The first major takeaway is just how much better each method was at making predictions on the Test data when the Informative Missing option was selected. The best model where this option was selected was the highlighted Boosted Tree model where a misclassification rate of 0.0805, which was slightly better than Boosted Neural Network model with the Informative Missing option enable, which misclassified 0.083 of the Test split.

For these two methods, the models lacking this feature performed worse, however not to the same degree. The normal Boosted Tree had a misclassification rate of 0.1426, whereas the normal Boosted Neural Network model's misclassification rate was a vastly worse 0.4664.

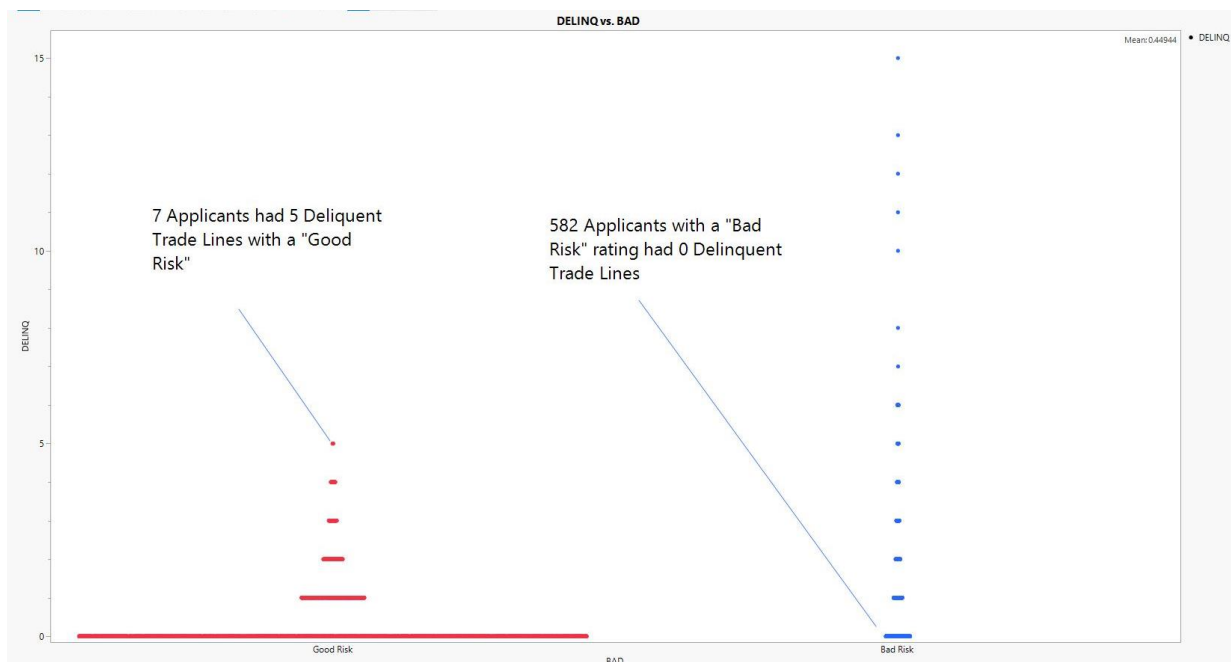
The best model overall was the K-Nearest Neighbors, which did not have an Informative Missing option available when building the model. Unfortunately, KNN models are less interpretable and naturally do not offer the ability to assess variable importance in JMP by virtue of their design.

So, in tune with the original assignment, the best model that proves the power of the Informative Missing feature in JMP is the Boosted Tree model, which I will be using to explore each variable's importance by their Independent Uniform Inputs:

Column	Main Effect	Total Effect	
DELINQ	0.236	0.543	
REASON	0.028	0.453	
DEROG	0.056	0.33	

Column	Main Effect	Total Effect	
JOB	0.033	0.201	
CLNO	0.021	0.156	
NINQ	0.015	0.116	
CLAGE	0.017	0.099	
YOJ	0.013	0.095	
DEBTINC	0.021	0.084	
VALUE	0.006	0.026	
LOAN	0.006	0.023	
MORTDUE	0.005	0.02	

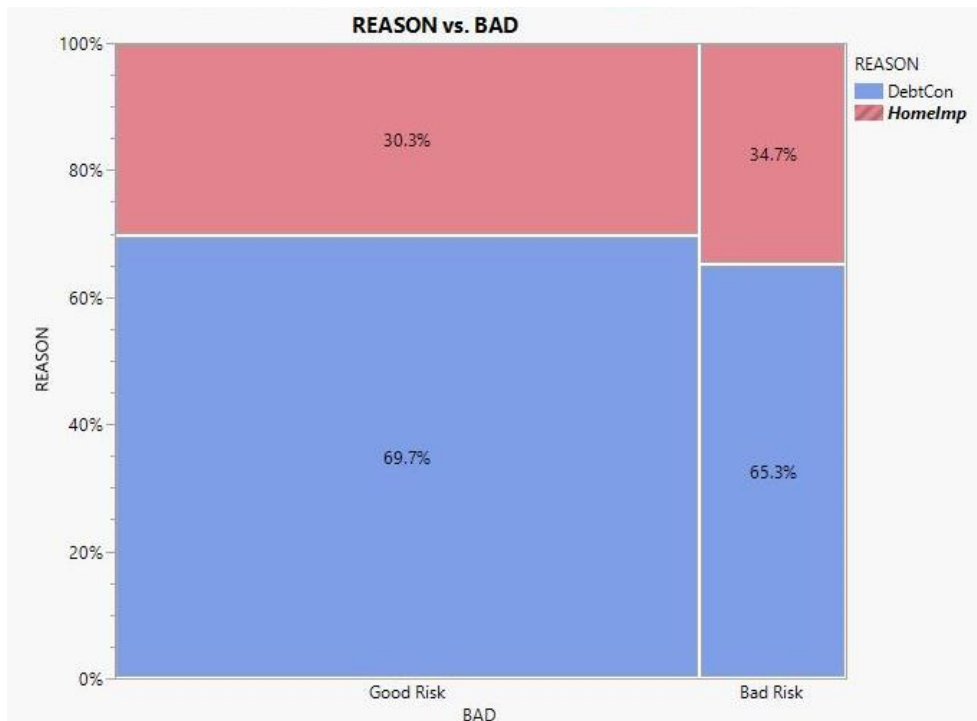
DELINQ (Number of Delinquent Trade Lines) was the most important variable in this model. The idea of delinquency on trade lines refers to where the borrower misses payments for line of credit such as mortgage, car loans, student loans, credit cards, or personal loans (Investopedia, 2022). Logically, the higher the number of delinquent lines, the worse the borrower's credit score would be, which is detrimental to their reputation. So, lenders would feel less inclined to trust the applicant to repay their home equity loan in a timely manner, if at all.



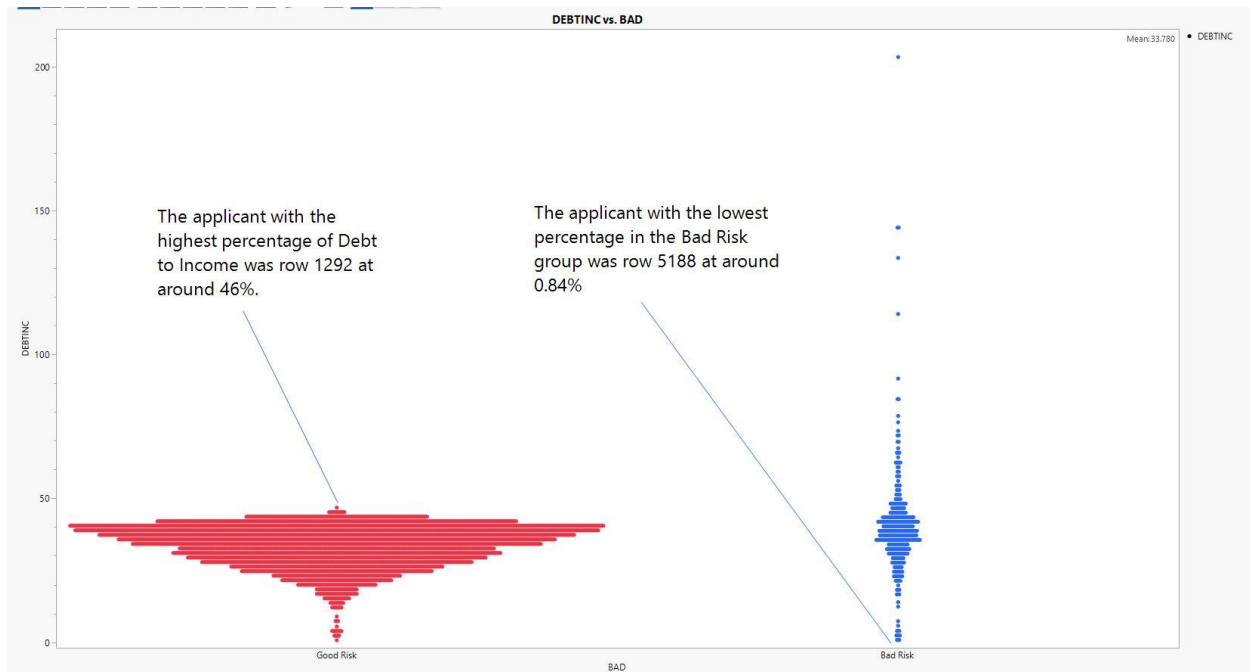
So, what other responses did these 7 Applicants with the highest DELINQ value in their group have?

		BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC
•	4764	Good Risk	26800	213165	282068	HomeImp	ProfExe	5	•	4	476.7283	•	55	31.18
•	4765	Good Risk	20000	126324	171450		Mgr	26	•	5	329.5667	1	28	•
•	4766	Good Risk	21300	134578	176968		Mgr	25	•	5	334.0659	0	27	31.88
•	4767	Good Risk	22600	131098	176119		Mgr	26	•	5	339.677	1	28	33.06
•	4768	Good Risk	22800	127010	171758		Mgr	25	•	5	335.6157	0	27	32.37
•	4769	Good Risk	23000	213000	280000	HomeImp	ProfExe	7	•	5	468.8667	•	56	•
•	4770	Good Risk	26400	220432	283978	HomeImp	ProfExe	6	•	5	480.356	•	56	33.71
•	4771	Good Risk	26900	215692	286555	HomeImp	ProfExe	8	•	5	468.1781	•	55	30.55
•	4772	Bad Risk	1500	•	•			•	•	•	•	•	•	•
•	4773	Bad Risk	2000	22608	•			18	•	•	•	•	•	•

These 7 applicants all either had Mgr or ProfExe listed as their Job. All of the ProfExe in this situation listed their Reason as Home Improving, so the empty spaces for the Mgr (assumed to be Managers) applicants might be intentional so as to not disclose that their loan is for DebtCon (Debt Consolidation) purposes. In fact, the reason for each loan was identified as the second most important contributing factor in an applicant's credit risk by the model. Surprisingly though, the proportion of the two reasons (Home Improvement and Debt Consolidation) were fairly similar between the risk groups as shows in the mosaic plot below:



Earlier in this paper, I mentioned that the DEBTINC (Debt-to-Income as a percentage) column was missing data for over a quarter of the applicants. The Graph Below show the distribution of Respondent's Debt to Income Percentage by which group they belong to:



So, while there were no applicants with a Good Risk rating that had a DEBTINC of over 46%, that is not to say that many individuals did not disclose their answer and could actually have a much higher value that would intuitively lead to them having a bad risk. 481 applicants with a Good Risk had no recorded value for their DEBTINC, whether intentionally or not.

Given this information, I wanted to predict how the model would classify a client's credit risk if they had the following responses for each variable, with two missing values:

LOAN: 31200, MORTDUE: 16800, VALUE: "." (missing), REASON: DebtCon, JOB: Other, YOJ: 12, DEROG: 1, DELINQ: 2, CLAGE: 110, NINQ: 4, CLNO: 20, DEBTINC: "." (missing)

**For the Boosted Tree Model with the Informative Missing Feature:**



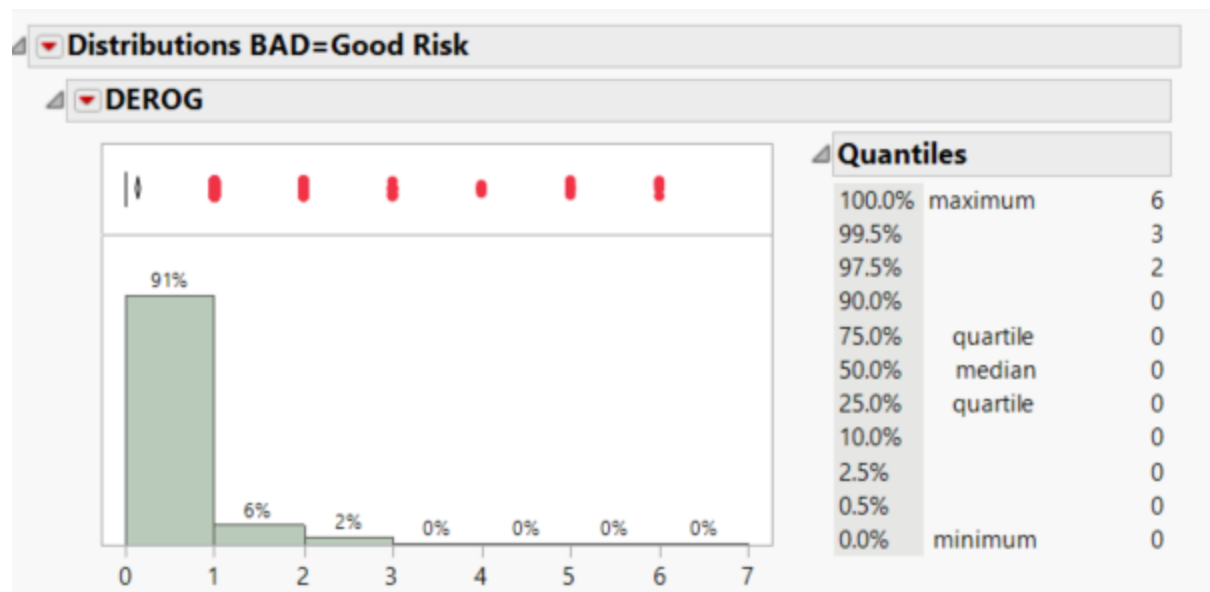


Based on the model's prediction profiler, I would not approve the client's loan request.

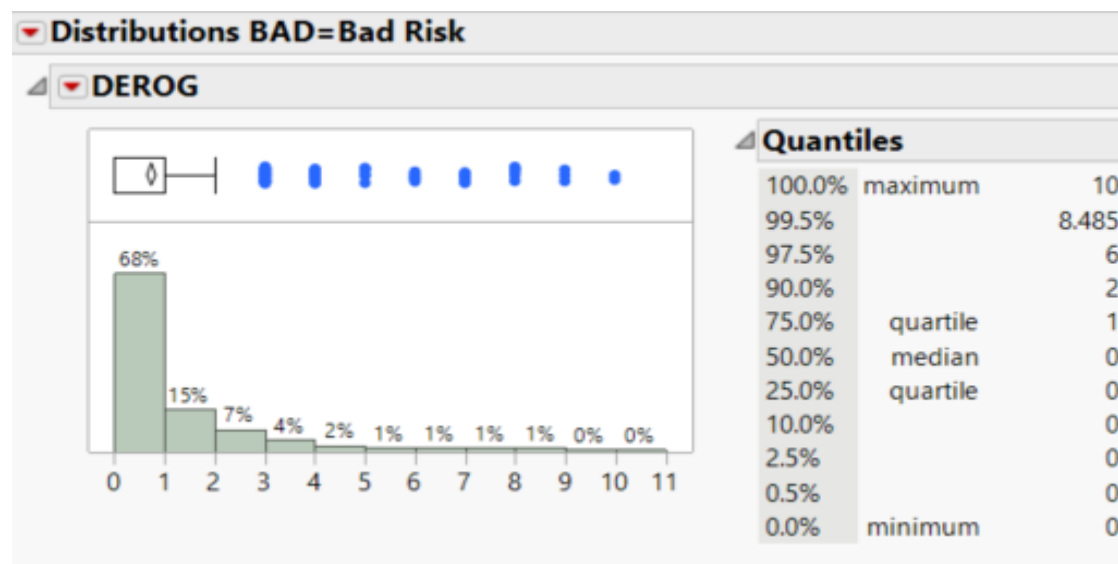
Now, to discuss why their application would be declined, it's important to reevaluate their responses in the context of each variable importance to the model's predictions. DELINQ had a total effect of 0.543 in the model. The client had a value of 2, which is on the lower end of the spectrum. So, their other responses must have been overall detrimental impact on their credit risk, particularly when it came to the REASON, DEROG, and JOB variables.

DEROG refers to the number of Derogatory reports for the applicant. Even a single derogatory report can have a lasting, negative impact on someone's credit history (Experian, 2022). So, the number of derogatory reports differs from the number of delinquent lines in terms of how severe the past failures to repay loans were, with the number of derogatory reports corresponding to the number of accounts that have been charged off, collection accounts with a status of repossession, foreclosure, and bankruptcy.

The percentage of applicants with even 1 derogatory report greatly different between the two credit risk classes:



6% of applicants with a good credit risk had 1 derogatory report, compared to 15% of those with a bad credit risk.



Considering how the K-Nearest Neighbors model outperformed the Boosted Tree models at predicting credit risks in the testing split, I wanted to test out what the KNN model would predict for the hypothetical applicant by filling in a new row under the existing dataset. Doing so auto-populates the prediction formula columns for all models that I have used. Although I can't see the profilers assess variable importance in this model, the KNN model rated this client as having a "Good Risk" while every other model has this client classified as a belonging to the

“Bad Risk” group. My interest here is in determining whether this difference would lead me to suspect that the boosted tree model misclassified the applicant.

So, I used the confusion matrices from both models to see which one had the highest total misclassification rate for the Bad Risks.

#### **Confusion Matrix for K-Nearest Neighbors on Test Split:**

<b>Actual BAD</b>	<b>Predicted Count</b>	
	<b>Good Risk</b>	<b>Bad Risk</b>
Good Risk	960	6
Bad Risk	73	153

<b>Actual BAD</b>	<b>Predicted Rate</b>	
	<b>Good Risk</b>	<b>Bad Risk</b>
Good Risk	0.994	0.006
Bad Risk	0.323	0.677

#### **Confusion Matrix for Boosted Tree with Informative Missing Feature on Test Split:**

<b>Actual BAD</b>	<b>Predicted Count</b>	
	<b>Good Risk</b>	<b>Bad Risk</b>
Good Risk	941	25
Bad Risk	71	155

<b>Actual BAD</b>	<b>Predicted Rate</b>	
	<b>Good Risk</b>	<b>Bad Risk</b>
Good Risk	0.974	0.026
Bad Risk	0.314	0.686

From the confusion matrices, it is apparent that both models had less predictive accuracy when it came to predicting true Bad Risks than true positives. However, the KNN model’s rate for accurately predicting Bad Risks was 0.677 compared to the Boosted Tree models rate of 0.686. Similarly, the KNN models misclassified actual Bad Risks as “Good” at a rate of 0.323, while the Boosted Tree’s equivalent misclassification rate was slightly lower at 0.314. So, considering that the Boosted Tree model was better at classifying true Bad Risks in the testing split, and that its prediction of “Bad Risk” for the hypothetical applicant matches the predictions for all other models besides the KNN model, I will still support the decision to deny the client’s request for a

Home Equity Loan based on the prediction profile output from the Boosted Tree model with the Informative Missing feature enabled.

## References

Kagan, Julia. "Trade Line: Definition, How It Works, and Included Records." *Investopedia*, Investopedia, [www.investopedia.com/terms/t/trade-line.asp](https://www.investopedia.com/terms/t/trade-line.asp). Accessed 10 Oct. 2023.

Millerbernd, Annie. "Personal Loan vs. Home Equity Loan: Which Is Best?" *NerdWallet*, 16 Feb. 2023, [www.nerdwallet.com/article/loans/personal-loans/home-equity-loan-vs-personal-loan](https://www.nerdwallet.com/article/loans/personal-loans/home-equity-loan-vs-personal-loan).

White, Jennifer. "What Does 'Derogatory' Mean on a Credit Report?" *Experian*, Experian, 29 June 2022, [www.experian.com/blogs/ask-experian/what-the-term-derogatory-means-in-a-credit-report/](https://www.experian.com/blogs/ask-experian/what-the-term-derogatory-means-in-a-credit-report/).