Assignment 4: Ensemble Model Development 1 Using SAS Enterprise Miner

Daanish Ahmed

DATA 640 9041

Fall 2017

dsahmed2334@yahoo.com

Professor Sounak Chakraborty

UMUC

November 19, 2017

Introduction

Ensemble models are some of the most accurate predictive models to use for classification problems. They involve creating several distinct models and averaging their outputs to arrive at a single result. Ensemble models are typically more accurate than individual models, but they can only achieve this when each model is distinct from one another ("Ensemble models," n.d.). These models work best on binary targets, and some of the most effective ensemble methods include bagging, boosting, gradient boosting, and random forests. In this assignment, I will create several ensemble models using SAS Enterprise Miner on a dataset about automobile purchases. I will focus on categorizing vehicles based on whether they are bad purchases, which will help my organization provide higher quality vehicles for consumers. I will build one model for each of the four ensemble types mentioned. I also plan on creating additional diagrams that use sampling, transformations, and variable selection—allowing me to study their impact on accuracy. I will then examine the effects of preprocessing by excluding data preparation from one of the models. Finally, I will compare the results to see which method is the most accurate, as well as which method can identify the highest number of bad vehicle purchases.

My dataset contains 72,983 observations and 34 variables. The target variable, "is bad buy," indicates whether the vehicle was a bad purchase. It is a binary variable with the values 0 (not bad) and 1 (bad). Some of the inputs include make, model, vehicle year, purchase date, warranty cost, and transmission (see Figure 4 in Appendix). The dataset contains 15 numeric variables and 19 character variables. Of the numeric variables, 2 are binary and 13 are interval. The binary variables are "is bad buy" and "is online sale." Interval inputs include vehicle year, vehicle cost, and warranty cost. All character inputs are nominal, and they include wheel type, transmission, and model. I set the variables "ref ID," "wheel type ID," and "BYRNO" to rejected

because they are IDs and are not useful for the analysis. I also found that vehicle year is strongly correlated with vehicle age. Vehicle age is a better indicator of a vehicle's actual age, while vehicle year is just a "model year" label. Thus, I kept vehicle age and set vehicle year to rejected.

According to the descriptive statistics (see Figure 4 in Appendix), there are six variables with missing values. The input with the highest percentage of missing values is trim (4.37%). This figure also shows that warranty cost has the highest skewness of 2.07 (excluding BYRNO since it was rejected). I found that the target variable is skewed as well, since there are only 8976 vehicles that are labeled as bad purchases. Furthermore, some variables appear to have outliers. For instance, warranty cost, odometer reading, and vehicle cost have values more than three standard deviations from the mean. Additionally, some variables such as current retail clean price are listed as nominal character variables when they are supposed to be numeric interval inputs. It is impossible to set these variables to the interval type. Inspecting the dataset reveals that some of their values are listed as "NULL," which is written in text format (see Figure 6 in Appendix). Thus, I will have to create new variables that exclude these "NULL" cases.

Data Preparation

I began data preparation by handling issues such as outliers, missing values, and errors. I used the Replacement node to eliminate outliers by setting the interval variable default limits method to "standard deviations from the mean." When using the default cutoff value of 3.0, this node will replace all values more than 3 standard deviations from the mean. I also used this node to handle errors, namely the "NULL" issue described earlier. I used the replacement editor to find all "NULL" cases and set their replacement value to unknown, and then I set the class variable

unknown levels replacement to "missing value." Next, I used the Transform Variables node to create new variables from the inputs which previously included "NULL" values. I used the formula editor to create copies of variables such as "current retail clean price" (see Figure 5 in Appendix). Since these new variables no longer contain any character entries, they are now labeled correctly as numeric interval inputs. Finally, I used the Impute node to handle missing values. I set the class variable default input method to "tree surrogate," and I set the interval variable default input method to "mean" to compute a replacement for all missing values.

Next, I used the StatExplore node to create a bar graph showing the importance of each input variable (see Figure 7 in Appendix). According to this image, the variables with the highest worth are wheel type, model, vehicle age, sub model, and current auction clean price. Likewise, the least important variables are online sale, transmission, nationality, color, and auction. When looking at the variable distributions, we see that several inputs are skewed and may benefit from transformations (see Figure 8 in Appendix). For instance, warranty cost has a clear positive skew, suggesting that most vehicles have cheaper warranties. Using the "max normal" transform will reduce this skewness. Furthermore, size and VNST have spikes in their distributions—which can be resolved by creating dummy variables. However, this would create hundreds of dummy variables since some variables contain very high numbers of levels. Thus, I will use variable selection to limit the number of inputs for all models that use transformations.

In this analysis, I will create three similar diagrams to explore the effects of sampling, transformations, and variable selection. My first diagram will involve creating the four ensemble models after completing the steps described so far (see Figure 1 in Appendix). My second diagram will iterate on my first diagram by using sampling prior to each model (see Figure 2 in Appendix). My third diagram will expand on my second diagram by using transformations and variable

selection nodes after sampling (see Figure 3 in Appendix). Sampling will be done using the Sample node, and it serves two purposes. Since my dataset contains over 70,000 cases, sampling will firstly reduce the training time. Secondly, creating a weighted sample will balance the cases of my target variable and reduce its skewness. These weighted samples were created by selecting the "level based" sampling criterion and setting level selection to the rarest level. I set the rarest level to take up 40% of all cases, resulting in samples with 22,440 cases and 8976 cases of bad buys. Each Sample node uses a different random seed to ensure that every model uses different data. This will avoid any issues from using the same data in all models. In the end, I will compare the models in each diagram to see if sampling has any effect on the overall accuracy.

In my third diagram, I will perform transformations and variable selection using the Transform Variables and Variable Selection nodes. For the bagging and boosting models, I will use the "max normal" method to minimize the skewness of interval inputs. For the gradient boosting and random forest models, I will use the "best" transform for interval inputs—which has often produced more accurate models in past assignments. All models use dummy indicators for class inputs to allow handling of nominal variables. I will then attach Variable Selection nodes using default properties to reduce the number of dummy variables. These models will determine the impact of transformations and variable selection on accuracy. One disadvantage of variable selection is that it may limit the number of available inputs for models such as random forest.

Model Development

With data preparation complete, I can now create my ensemble models. The first four models use one of the following methods: bagging, boosting, gradient boosting, and random forest.

Bagging, or "Bootstrap Aggregating," involves building many different decision trees where each tree uses a random sample of the data and replaces those cases from the original dataset (Knode, 2016a). This process results in some cases being used in multiple trees while others are not used at all. Boosting is an iterative process that adjusts the weights of subsequent trees such that incorrect classifications are given more weight while correct predictions have less weight (Srivastava, 2015). Gradient boosting, a variation of boosting, involves computing the error from the previous tree and attempting to correct that error in later trees (Ihler, 2012). Random forests consist of many decision trees made from random data samples, where each iteration of the tree uses a random subset of the available input variables (Knode, 2016a).

The bagging and boosting models were created using three nodes: Start Groups, Decision Tree, and End Groups. The Start Groups node had its mode set to "bagging" and "boosting" for each of the respective models. For the index count (i.e. number of iterations), I kept the default value of 10 due to extremely long training times lasting several minutes. I then attached a Decision Tree node to each model followed by an End Groups node, using the default properties for each. For the gradient boosting and random forest models, I used the Gradient Boosting and HP Forest nodes respectively—keeping the default properties for each node. These four models are the same across all three diagrams, with the only difference being the inclusion of sampling, transformation, and variable selection nodes for the appropriate diagrams. My fifth model will be a random forest model that does not use any preprocessing (see Figure 3 in Appendix). Its purpose is to evaluate whether data preparation will improve the results. I will not use data partitioning in any diagram because ensemble models are usually evaluated on the training data (Knode, 2016a).

I will now describe several important measures for evaluating the results. The ROC curve shows the sensitivity of each model compared to its specificity (see Figure 9 in Appendix). Since

we want to maximize the number of true positives, we prefer our model to have a higher sensitivity. Next, the classification chart shows a bar graph with the number of correct and incorrect classifications for the majority and minority classes (see Figure 10 in Appendix). This is useful for visualizing each model's ability to classify 1's and 0's. The fit statistics include several measures, of which the most important is misclassification rate (see Figure 11 in Appendix). A more accurate model will have a low misclassification rate. Finally, the classification table shows the number of true positives, false positives, true negatives, and false negatives in each model (see Figure 12 in Appendix). We want to minimize the error from false positives and negatives while maximizing the number of correct predictions—especially from true positives. Unfortunately, the results provide little information for finding the most effective inputs. Ensemble models are black box technologies which are difficult to interpret ("Ensemble models," n.d.), which is a major disadvantage when compared to individual models. However, it is likely that "wheel type" will have the greatest impact on the target due to having the highest worth (see Figure 7 in Appendix).

Results

I will now assess the results of my analysis to determine which method is the most effective at classifying bad vehicle purchases. I will use Model Comparison nodes to analyze all models in each diagram, beginning with my first diagram (see Figure 1 in Appendix). According to the ROC curve, boosting has the highest sensitivity per specificity, followed by random forest, bagging, and gradient boosting (see Figure 9 in Appendix). This suggests that boosting is the most effective model for identifying the minority class, while gradient boosting is the weakest. The classification chart reveals that boosting identified the highest number of true positives, while gradient boosting identified the fewest (see Figure 10 in Appendix). However, boosting also misclassified a massive

number of 0's. And even though this model identified the highest number of true positives, it still failed to classify a significant amount of 1's. Gradient boosting accurately identified more 0's than any other model, yet it hardly identified any 1's. The bagging and random forest models have more balanced results, but they are still very weak at identifying true positives.

The fit statistics indicate that the random forest is the most accurate model with a misclassification of 9.88%, while boosting is by far the least accurate with a misclassification of 36.3% (see Figure 11 in Appendix). This number is significantly worse than the second least accurate model, gradient boosting, which has a misclassification of only 12.3%. The classification table (see Figure 12 in Appendix) corresponds with my findings from the classification chart. This figure shows that boosting identified the highest number of true positives while having the lowest number of false negatives. But it performed poorly at identifying 0's, as evidenced by having 22,492 false positives. Gradient boosting classified every value as a 0—meaning that it correctly identified all 0's but failed to classify any 1's. I would not recommend using the gradient boosting model due to its inability to identify bad vehicle purchases. And though boosting can identify the highest number of 1's, I cannot recommend it either due to its terrible overall accuracy. Of the remaining models, I would suggest using the random forest model since it has the highest accuracy. But when classifying true positives, there is significant room for improvement.

Next, I will examine the models from my second diagram that uses weighted samples (see Figure 2 in Appendix). Its ROC curve is very different from the previous diagram's ROC curve, since it lists random forest as the strongest model and boosting as the weakest (see Figure 13 in Appendix). The fit statistics show that boosting has the highest misclassification rate of 50.0%, while random forest has the lowest rate of 27.5% (see Figure 14 in Appendix). These results resemble those from the previous diagram, but the accuracy rates are now significantly worse.

According to the classification table and classification chart (see Figures 15 and 16 in Appendix), boosting once again identifies the highest number of true positives (6212) and the lowest number of true negatives by far (5004). However, gradient boosting no longer classifies every value as a 0 and has successfully identified 4196 true positives. Thus, it is possible that the gradient boosting model has a better fit on the current data sample than on the full dataset. But overall, all four models perform better at classifying 1's but are significantly weaker at classifying 0's. This causes every model to have a much lower accuracy, and thus I do not recommend using this sampling method on my dataset. Sampling may have weakened the results by reducing the number of cases available in each model. Based on these results, I would recommend the random forest model if focusing on overall accuracy. But if focusing on bad vehicle purchases, I would suggest using gradient boosting since it has the highest number of true positives.

Now I will evaluate the models from my third diagram that uses sampling, transformations, and variable selection (see Figure 3 in Appendix). The ROC curve once again shows that the random forest has the highest sensitivity per specificity while boosting has the lowest (see Figure 17 in Appendix). The fit statistics show that boosting has the highest misclassification of 46.9% while random forest has the lowest rate of 27.4% (see Figure 18 in Appendix). These models are slightly more accurate than those in the second diagram, except for gradient boosting which has a higher misclassification of 35.8%. Gradient boosting used the "best" transform, suggesting that this transformation may not be a good fit for this model. It is also possible that variable selection lowered its accuracy by limiting the number of input variables. Based on the classification table and classification chart (see Figures 19 and 20 in Appendix), boosting still has the highest number of true positives (5167) and the lowest number of true negatives by far (6743). Overall, these models are slightly better at identifying 0's compared to the second diagram, but they are also

slightly worse at identifying 1's. Still, the accuracy improved when using transformations and variable selection in most cases. But these models still succumb to the shortcomings from sampling, making them weaker than my original set of models. The random forest has the highest overall accuracy, and it identified the highest number of true positives aside from the boosting model (the least accurate model). Thus, I still recommend using the random forest model.

Finally, I will evaluate my last model—a random forest without any data preparation (see Figure 3 in Appendix). This model will determine whether preprocessing improves the classification accuracy. The fit statistics reveal that this model has a misclassification rate of 9.96% (see Figure 21 in Appendix), which is slightly worse than my original random forest's misclassification of 9.88% (see Figure 11 in Appendix). The classification table shows that this model has a slightly higher number of true positives, but its increase of false positives outweighs its increase of true positives (see Figure 22 in Appendix). These findings show that preprocessing has slightly improved the accuracy, but it has also slightly reduced the sensitivity.

Conclusion

In my analysis, I created five ensemble models across three diagrams to classify bad vehicle purchases. My goal was to obtain not only the most accurate model, but also the model that can identify the highest number of bad buys. I found that the random forest consistently had the highest accuracy regardless of sampling, transformations, or variable selection. When focusing on true positives, boosting always had the highest sensitivity by far. However, it also had the highest misclassification rate due to its ineffectiveness at classifying 0's. Due to its terrible accuracy, I do not recommend the boosting model for classifying bad vehicle purchases. Excluding this model,

the random forest often had one of the highest sensitivity rates—though gradient boosting had higher sensitivity when using weighted data samples. But since gradient boosting fails to classify 1's when using the full dataset, I ultimately recommend the random forest model.

Through this analysis, I found that my sampling method significantly worsened the results. Although it improved each model's sensitivity, it also drastically lowered the specificity and accuracy. As such, I would suggest using the entire dataset instead. Transformations and variable selection, on the other hand, slightly improved each model's accuracy. However, these nodes slightly reduced the number of true positives identified. I may recommend using transformations and variable selection to maximize accuracy, but this should be done without sampling. Finally, I found that data preparation helps to improve accuracy but with a minimal loss of sensitivity. My fifth model—a random forest with no preprocessing nodes—had a slightly lower accuracy than my original random forest but also had a slightly higher number of true positives.

The main limitation of my analysis is the poor sensitivity of my models. Even my most accurate models failed to achieve sensitivity rates of at least 80%—regardless of data preparation, sampling, transformations, or variable selection. Although ensemble models are usually more accurate than individual models, it is possible that my dataset is not the best fit for using bagging, boosting, gradient boosting, or random forests. It is likely that other models, such as SVM or neural networks, may produce stronger results. Thus, one suggestion for further analysis would be to use combine other types of models using the Ensemble node. These heterogeneous models might provide a closer balance between accuracy and sensitivity. Another idea for future analysis would involve experimenting with a validation or test set to evaluate the results. Though ensemble models are typically evaluated on the training set, it is possible that using this approach may yield helpful insights to help my organization effectively classify bad vehicle purchases.

References

Ensemble models and portioning algorithms in SAS Enterprise Miner. (n.d.). Retrieved November 6, 2017, from http://www.sas.com/apps/webnet/video-sharing.html?bcid=4363855671001

Ihler, A. (Performer) (2012, November 7). *Ensembles (3): Gradient boosting* [Web]. Retrieved November 7, 2017, from https://www.youtube.com/watch?v=sRktKszFmSk

Knode, S. (2016a, November 1). *Ensemble Models (bagging, boosting, random forest)*. Lecture presented at UMUC. Retrieved November 6, 2017.

Knode, S. (2016b, November 3). *SAS Enterprise Miner Ensemble Models (bagging, boosting, random forest)*. Lecture presented at UMUC. Retrieved November 7, 2017.

Srivastava, T. (2015, August 2). Basics of Ensemble Learning Explained in Simple English. Retrieved November 6, 2017, from https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/

Appendix

Relevant SAS Enterprise Miner Output Images

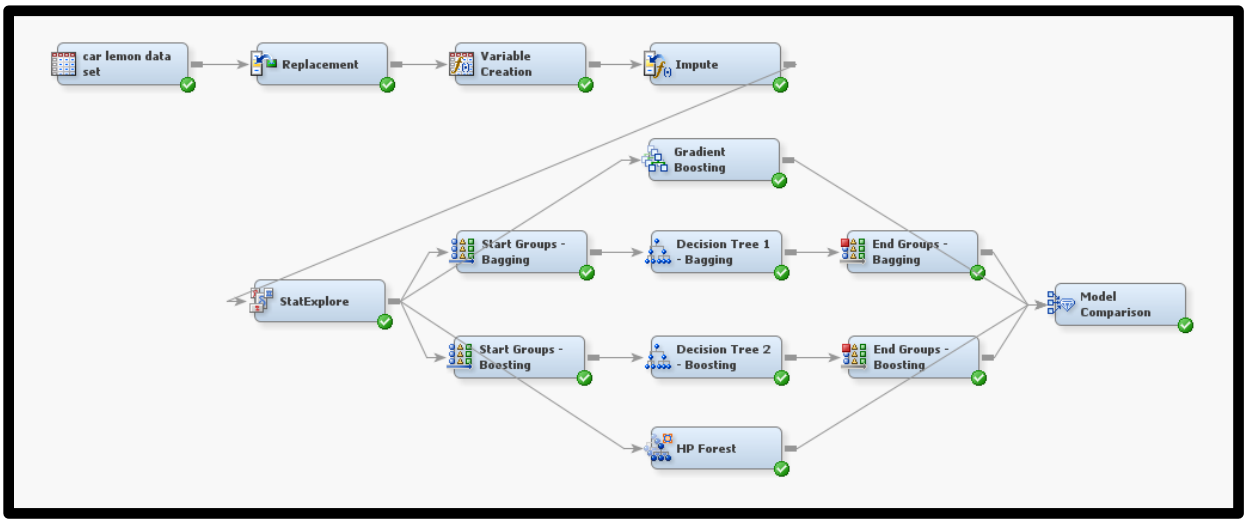


Figure 1. Initial Process Flow Diagram for Car Lemon Ensemble Models.

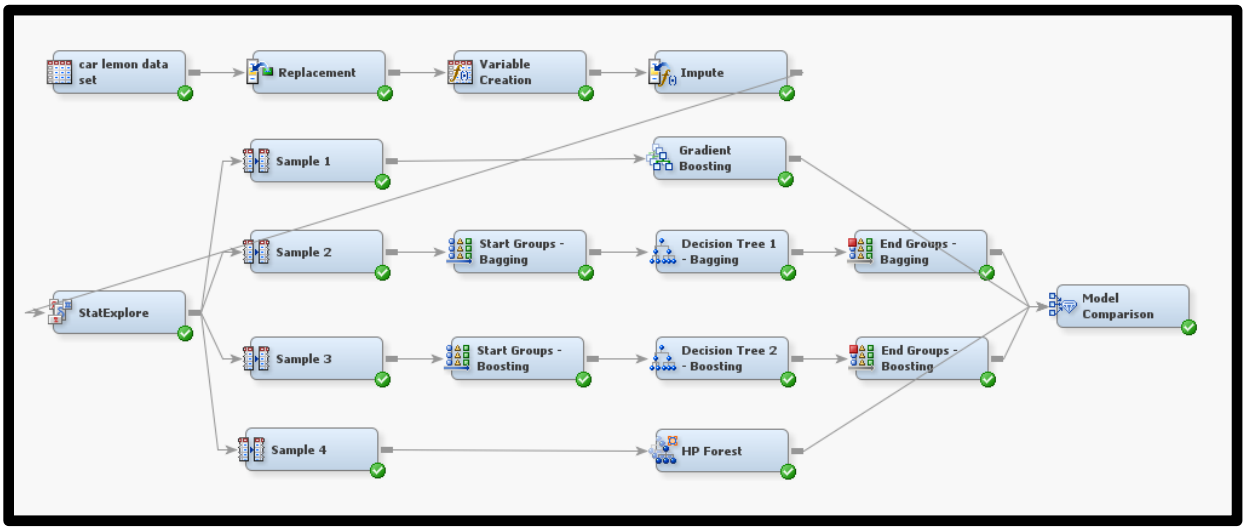


Figure 2. Process Flow Diagram for Ensemble Models with Sampling.

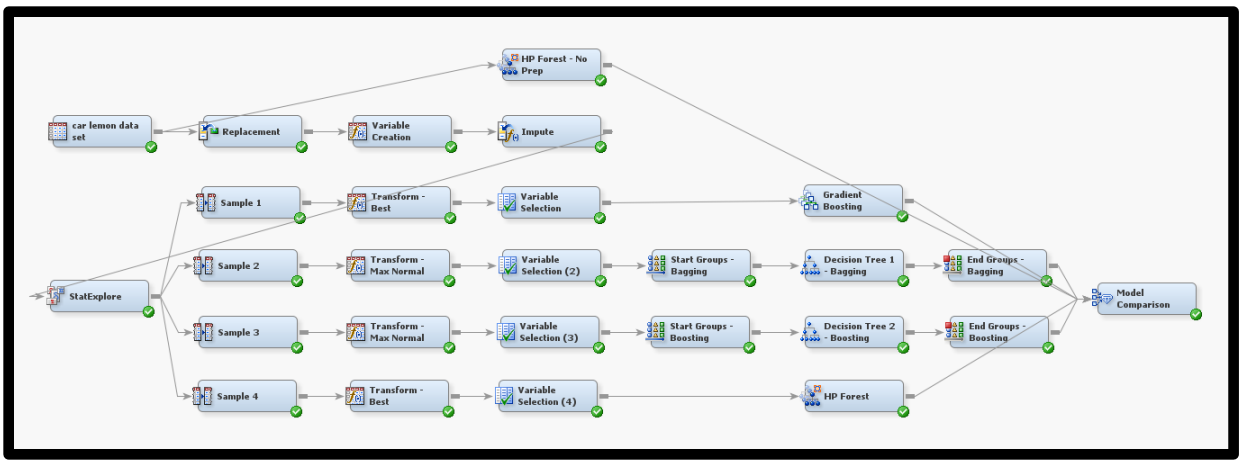


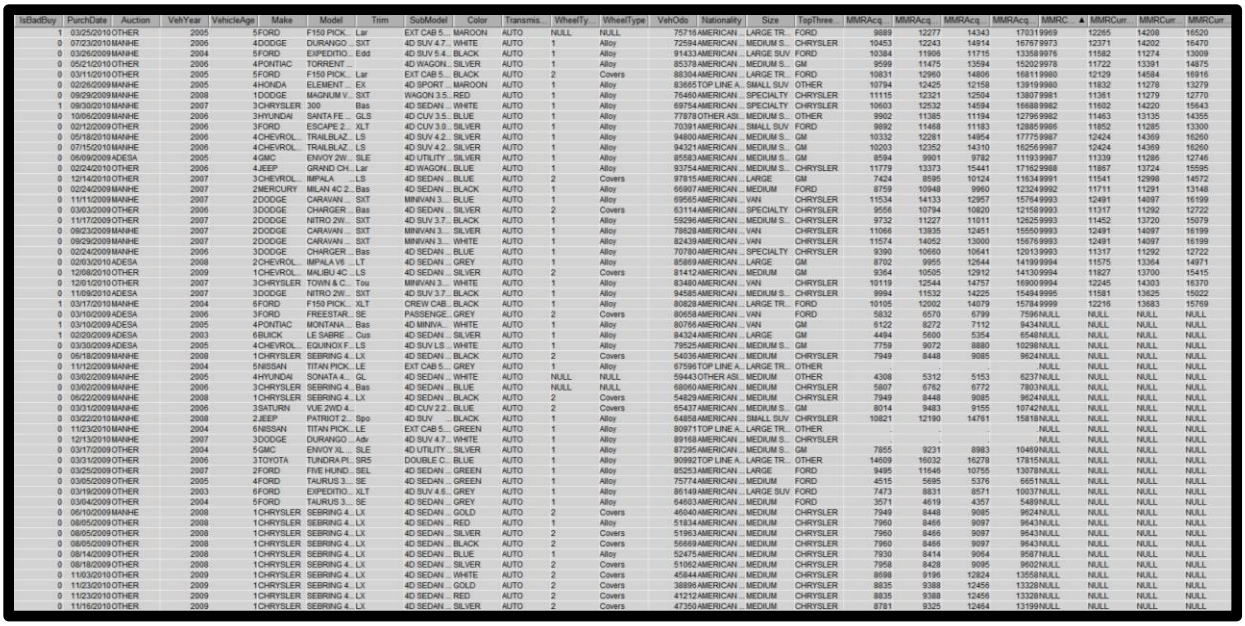
Figure 3. Process Flow Diagram with Sampling, Transformations, and Variable Selection.

Name	Role	Level	Type	Number of Levels	Percent Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis
AUCGUART	Input	Nominal	Character	3	0	.	.	.	.	.	.
Auction	Input	Nominal	Character	3	0	.	.	.	.	.	.
BYRNO	Rejected	Interval	Numeric	.	0	835	99761	26345.84	25717.35	2.129225	3.474184
Color	Input	Nominal	Character	17	0	.	.	.	.	.	.
IsBadBuy	Target	Binary	Numeric	2	0	.	.	.	.	.	.
IsOnlineSale	Input	Binary	Numeric	2	0	.	.	.	.	.	.
MMRAcquisitionAuctionAveragePric	Input	Interval	Numeric	.	0.024663	0	35722	6128.909	2461.993	0.463641	1.593728
MMRAcquisitionAuctionCleanPrice	Input	Interval	Numeric	.	0.024663	0	36859	7373.636	2722.492	0.466501	1.651147
MMRAcquisitionRetailAveragePrice	Input	Interval	Numeric	.	0.024663	0	39080	8497.034	3156.285	0.209214	0.680599
MMRAcquisitonRetailCleanPrice	Input	Interval	Numeric	.	0.024663	0	41482	9850.928	3385.79	0.1763	0.924827
MMRCurrentAuctionAveragePrice	Input	Nominal	Character	21	0	.	.	.	.	.	.
MMRCurrentAuctionCleanPrice	Input	Nominal	Character	21	0	.	.	.	.	.	.
MMRCurrentRetailAveragePrice	Input	Nominal	Character	21	0	.	.	.	.	.	.
MMRCurrentRetailCleanPrice	Input	Nominal	Character	21	0	.	.	.	.	.	.
Make	Input	Nominal	Character	21	0	.	.	.	.	.	.
Model	Input	Nominal	Character	21	0	.	.	.	.	.	.
Nationality	Input	Nominal	Character	5	0	.	.	.	.	.	.
PRIMEUNIT	Input	Nominal	Character	3	0	.	.	.	.	.	.
PurchDate	Time ID	Interval	Numeric	.	0	.	.	.	.	.	.
RefId	Rejected	Interval	Numeric	.	0	1	73014	36511.43	21077.24	-.000203	-1.19996
Size	Input	Nominal	Character	13	0	.	.	.	.	.	.
SubModel	Input	Nominal	Character	21	0	.	.	.	.	.	.
TopThreeAmericanName	Input	Nominal	Character	5	0	.	.	.	.	.	.
Transmission	Input	Nominal	Character	3	0.00137	.	.	.	.	.	.
Trim	Input	Nominal	Character	20	4.367863	.	.	.	.	.	.
VNST	Input	Nominal	Character	21	0	.	.	.	.	.	.
VNZIP1	Input	Interval	Numeric	.	0	2764	99224	58043.06	26151.64	-0.10353	-1.68869
VehBCost	Input	Interval	Numeric	.	0	1	45469	6730.934	1767.846	0.715931	8.144378
VehOdo	Input	Interval	Numeric	.	0	4825	115717	71500	14578.91	-0.45315	-0.19874
VehYear	Rejected	Interval	Numeric	.	0	2001	2010	2005.343	1.731252	-0.33736	-1.28193
VehicleAge	Input	Interval	Numeric	.	0	0	9	4.176644	1.71221	0.393616	-0.20927
WarrantyCost	Input	Interval	Numeric	.	0	462	7498	1276.581	598.8468	2.070831	9.964808
WheelType	Input	Nominal	Character	4	0	.	.	.	.	.	.
WheelTypeID	Rejected	Nominal	Character	5	0	.	.	.	.	.	.

Figure 4. Descriptive Statistics of Car Lemon Dataset.

Name ▲	Type	Length	Format	Level	Formula	Label	Role	Report
NEW_CurrentAuctionAveragePric	Numeric	8		Interval	input(REP_MMRCurrentAuctionAveragePric, best32.)		Input	No
NEW_CurrentAuctionCleanPrice	Numeric	8		Interval	input(REP_MMRCurrentAuctionCleanPrice, best32.)		Input	No
NEW_CurrentRetailAveragePrice	Numeric	8		Interval	input(REP_MMRCurrentRetailAveragePrice, best32.)		Input	No
NEW_CurrentRetailCleanPrice	Numeric	8		Interval	input(REP_MMRCurrentRetailCleanPrice, best32.)		Input	No

Figure 5. Interval Variable Creation Using Transform Variables Formula Editor.



IsBadBuy	PurchDate	Auction	VehYear	VehicleAge	Make	Model	Trim	SubModel	Color	Transmis...	WheelTy...	WheelType	VehOdo	Nationality	Size	TopThree...	MMRAcq...	MMRAcq...	MMRAcq...	MMRAcq...	MMRC... ▲	MMRCurr...	MMRCurr...	MMRCurr...
1	03/25/2010	OTHER	2005	5	FORD	F150 PICK...	Lar	EXT CAB 5...	MAROON	AUTO	NULL	NULL	75716	AMERICAN ...	LARGE TR...	FORD	9889	12277	14343	17031	9969	12265	14208	16520
0	07/23/2010	MANHE	2006	4	DODGE	DURANGO ...	SXT	4D SUV 4.7...	WHITE	AUTO	1	Alloy	72594	AMERICAN ...	MEDIUM S...	CHRYSLER	10453	12243	14914	16767	9973	12371	14202	16470
0	03/26/2009	MANHE	2004	5	FORD	EXPEDITIO...	Edd	4D SUV 5.4...	BLACK	AUTO	1	Alloy	91433	AMERICAN ...	LARGE SUV	FORD	10384	11906	11715	13358	9976	11582	11274	13009
0	05/21/2010	OTHER	2006	4	PONTIAC	TORRENT ...	..	4D WAGON...	SILVER	AUTO	1	Alloy	85378	AMERICAN ...	MEDIUM S...	GM	9599	11475	13594	15202	9978	11722	13391	14875
0	03/11/2010	OTHER	2005	5	FORD	F150 PICK...	Lar	EXT CAB 5...	BLACK	AUTO	2	Covers	88304	AMERICAN ...	LARGE TR...	FORD	10831	12960	14806	16811	9980	12129	14584	16916
0	02/26/2009	MANHE	2005	4	HONDA	ELEMENT ...	EX	4D SPORT ...	MAROON	AUTO	1	Alloy	83665	TOP LINE A...	SMALL SUV	OTHER	10794	12425	12158	13919	9980	11832	11278	13279
0	09/29/2009	MANHE	2008	1	DODGE	MAGNUM V...	SXT	WAGON 3.5...	RED	AUTO	1	Alloy	76460	AMERICAN ...	SPECIALTY	CHRYSLER	11115	12321	12504	13807	9981	11361	11279	12770
1	09/30/2010	MANHE	2007	3	CHRYSLER	300	Bas	4D SEDAN ...	WHITE	AUTO	1	Alloy	69754	AMERICAN ...	SPECIALTY	CHRYSLER	10603	12532	14594	16688	9982	11602	14220	15643
0	10/05/2009	MANHE	2006	3	HYUNDAI	SANTA FE ...	GLS	4D CUV 3.5...	BLUE	AUTO	1	Alloy	77878	OTHER ASI...	MEDIUM S...	OTHER	9902	11385	11194	12796	9982	11463	13135	14355
0	02/12/2009	OTHER	2006	3	FORD	ESCAPE 2...	XLT	4D CUV 3.0...	SILVER	AUTO	1	Alloy	70391	AMERICAN ...	SMALL SUV	FORD	9892	11468	11183	12885	9986	11852	11285	13300
0	05/18/2010	MANHE	2006	4	CHEVROL...	TRAILBLAZ...	LS	4D SUV 4.2...	SILVER	AUTO	1	Alloy	94800	AMERICAN ...	MEDIUM S...	GM	10332	12281	14954	17775	9987	12424	14369	16260
0	07/15/2010	MANHE	2006	4	CHEVROL...	TRAILBLAZ...	LS	4D SUV 4.2...	SILVER	AUTO	1	Alloy	94321	AMERICAN ...	MEDIUM S...	GM	10203	12352	14310	16256	9987	12424	14369	16260
0	06/09/2009	ADESA	2005	4	GMC	ENVOY 2W...	SLE	4D UTILITY ...	SILVER	AUTO	1	Alloy	85583	AMERICAN ...	MEDIUM S...	GM	8594	9901	9782	11193	9987	11339	11286	12746
0	02/24/2010	OTHER	2006	4	JEEP	GRAND CH...	Lar	4D WAGON...	BLUE	AUTO	1	Alloy	93754	AMERICAN ...	MEDIUM S...	CHRYSLER	11779	13373	15441	17162	9988	11867	13724	15595
0	12/14/2010	OTHER	2007	3	CHEVROL...	IMPALA	LS	4D SEDAN ...	BLUE	AUTO	2	Covers	97815	AMERICAN ...	LARGE	GM	7424	8595	10124	11634	9991	11541	12998	14572
0	02/24/2009	MANHE	2007	2	MERCURY	MILAN 4C 2...	Bas	4D SEDAN ...	BLACK	AUTO	1	Alloy	66907	AMERICAN ...	MEDIUM	FORD	8759	10948	9960	12324	9992	11711	11291	13148
0	11/11/2009	MANHE	2007	2	DODGE	CARAVAN ...	SXT	MINIVAN 3...	WHITE	AUTO	1	Alloy	69565	AMERICAN ...	VAN	CHRYSLER	11534	14133	12957	15764	9993	12491	14097	16199
0	03/03/2009	OTHER	2006	3	DODGE	CHARGER ...	Bas	4D SEDAN ...	SILVER	AUTO	2	Covers	63114	AMERICAN ...	SPECIALTY	CHRYSLER	9556	10794	10820	12158	9993	11317	11292	12722
0	11/17/2009	OTHER	2007	2	DODGE	NITRO 2W...	SXT	4D SUV 3.7...	BLACK	AUTO	1	Alloy	59296	AMERICAN ...	MEDIUM S...	CHRYSLER	9732	11227	11011	12625	9993	11452	13720	15079
0	09/23/2009	MANHE	2007	2	DODGE	CARAVAN ...	SXT	MINIVAN 3...	SILVER	AUTO	1	Alloy	78628	AMERICAN ...	VAN	CHRYSLER	11066	13935	12451	15550	9993	12491	14097	16199
0	09/29/2009	MANHE	2007	2	DODGE	CARAVAN ...	SXT	MINIVAN 3...	WHITE	AUTO	1	Alloy	82439	AMERICAN ...	VAN	CHRYSLER	11574	14052	13000	15676	9993	12491	14097	16199
0	02/24/2009	MANHE	2006	3	DODGE	CHARGER ...	Bas	4D SEDAN ...	BLUE	AUTO	1	Alloy	70780	AMERICAN ...	SPECIALTY	CHRYSLER	9390	10660	10641	12913	9993	11317	11292	12722
0	02/03/2010	ADESA	2008	2	CHEVROL...	IMPALA V6 ...	LT	4D SEDAN ...	GREY	AUTO	1	Alloy	85869	AMERICAN ...	LARGE	GM	8702	9955	12644	14199	9994	11575	13364	14971
0	12/08/2010	OTHER	2009	1	CHEVROL...	MALIBU 4C ...	LS	4D SEDAN ...	SILVER	AUTO	2	Covers	91412	AMERICAN ...	MEDIUM	GM	9364	10505	12912	14130	9994	11827	13700	15415
0	12/01/2010	OTHER	2007	3	CHRYSLER	TOWN & C...	Tou	MINIVAN 3...	WHITE	AUTO	1	Alloy	83480	AMERICAN ...	VAN	CHRYSLER	10119	12544	14757	16900	9994	12245	14303	16370
0	11/09/2010	ADESA	2007	3	DODGE	NITRO 2W...	SXT	4D SUV 3.7...	BLACK	AUTO	1	Alloy	94585	AMERICAN ...	MEDIUM S...	CHRYSLER	9994	11532	14225	15494	9995	11581	13625	15022
1	03/17/2010	MANHE	2004	5	FORD	F150 PICK...	XLT	CREW CAB ...	BLACK	AUTO	1	Alloy	80828	AMERICAN ...	LARGE TR...	FORD	10105	12002	14079	15784	9999	12216	13683	15769
0	03/10/2009	ADESA	2006	3	FORD	FREESTAR ...	SE	PASSENGE...	GREY	AUTO	2	Covers	80658	AMERICAN ...	VAN	FORD	5832	6579	6799	7596	NULL	NULL	NULL	NULL
1	03/10/2009	ADESA	2005	4	PONTIAC	MONTANA ...	Bas	4D MINIVA...	WHITE	AUTO	1	Alloy	80766	AMERICAN ...	VAN	GM	6122	8272	7112	9434	NULL	NULL	NULL	NULL
1	02/20/2009	ADESA	2003	6	BUICK	LE SABRE ...	Cus	4D SEDAN ...	SILVER	AUTO	1	Alloy	84324	AMERICAN ...	LARGE	GM	4494	5600	5354	6548	NULL	NULL	NULL	NULL
0	03/30/2009	ADESA	2005	4	CHEVROL...	EQUINOX F...	LS	4D SUV LS ...	WHITE	AUTO	1	Alloy	79525	AMERICAN ...	MEDIUM S...	GM	7759	9072	8880	10298	NULL	NULL	NULL	NULL
0	06/18/2009	MANHE	2008	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	BLACK	AUTO	2	Covers	54036	AMERICAN ...	MEDIUM	CHRYSLER	7949	8448	9085	9624	NULL	NULL	NULL	NULL
0	11/12/2009	MANHE	2004	5	NISSAN	TITAN PICK...	LE	EXT CAB 5...	GREY	AUTO	1	Alloy	67596	TOP LINE A...	LARGE TR...	OTHER	.	.	.	.	NULL	NULL	NULL	NULL
0	03/02/2009	MANHE	2005	4	HYUNDAI	SONATA 4 ...	GL	4D SEDAN ...	WHITE	AUTO	NULL	NULL	59443	OTHER ASI...	MEDIUM	OTHER	4308	5312	5153	6237	NULL	NULL	NULL	NULL
0	03/02/2009	MANHE	2006	3	CHRYSLER	SEBRING 4 ...	Bas	4D SEDAN ...	BLUE	AUTO	NULL	NULL	68060	AMERICAN ...	MEDIUM	CHRYSLER	5807	6762	6772	7803	NULL	NULL	NULL	NULL
0	06/22/2009	MANHE	2008	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	BLACK	AUTO	2	Covers	54829	AMERICAN ...	MEDIUM	CHRYSLER	7949	8448	9085	9624	NULL	NULL	NULL	NULL
0	03/31/2009	MANHE	2006	3	SATURN	VUE 2WD 4 ...	..	4D CUV 2.2...	BLUE	AUTO	2	Covers	65437	AMERICAN ...	MEDIUM S...	GM	8014	9483	9155	10742	NULL	NULL	NULL	NULL
0	03/22/2010	MANHE	2008	2	JEEP	PATRIOT 2...	Spo	4D SUV	...	BLACK	AUTO	1	Alloy	64858	AMERICAN ...	SMALL SUV	CHRYSLER	10821	12190	14761	15818	NULL	NULL	NULL	NULL
0	11/23/2010	MANHE	2004	6	NISSAN	TITAN PICK...	LE	EXT CAB 5...	GREEN	AUTO	1	Alloy	80971	TOP LINE A...	LARGE TR...	OTHER	.	.	.	.	NULL	NULL	NULL	NULL
0	12/13/2010	MANHE	2007	3	DODGE	DURANGO ...	Adv	4D SUV 4.7...	WHITE	AUTO	1	Alloy	89168	AMERICAN ...	MEDIUM S...	CHRYSLER	.	.	.	.	NULL	NULL	NULL	NULL
0	03/17/2009	OTHER	2004	5	GMC	ENVOY XL ...	SLE	4D UTILITY ...	SILVER	AUTO	1	Alloy	87295	AMERICAN ...	MEDIUM S...	GM	7855	9231	8983	10459	NULL	NULL	NULL	NULL
0	03/31/2009	OTHER	2006	3	TOYOTA	TUNDRA PI...	SR5	DOUBLE C...	BLUE	AUTO	1	Alloy	90992	TOP LINE A...	LARGE TR...	OTHER	14609	16032	16278	17815	NULL	NULL	NULL	NULL
0	03/25/2009	OTHER	2007	2	FORD	FIVE HUND...	SEL	4D SEDAN ...	GREEN	AUTO	1	Alloy	95253	AMERICAN ...	LARGE	FORD	9495	11646	10755	13078	NULL	NULL	NULL	NULL
0	03/05/2009	OTHER	2005	4	FORD	TAURUS 3...	SE	4D SEDAN ...	GREEN	AUTO	1	Alloy	75774	AMERICAN ...	MEDIUM	FORD	4515	5695	5376	6651	NULL	NULL	NULL	NULL
0	03/19/2009	OTHER	2003	6	FORD	EXPEDITIO...	XLT	4D SUV 4.6...	GREY	AUTO	1	Alloy	86149	AMERICAN ...	LARGE SUV	FORD	7473	8831	8571	10037	NULL	NULL	NULL	NULL
0	03/04/2009	OTHER	2004	5	FORD	TAURUS 3...	SE	4D SEDAN ...	GREY	AUTO	1	Alloy	64603	AMERICAN ...	MEDIUM	FORD	3571	4619	4357	5489	NULL	NULL	NULL	NULL
0	06/10/2009	MANHE	2008	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	GOLD	AUTO	2	Covers	46040	AMERICAN ...	MEDIUM	CHRYSLER	7949	8448	9085	9624	NULL	NULL	NULL	NULL
0	08/05/2009	OTHER	2008	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	RED	AUTO	1	Alloy	51834	AMERICAN ...	MEDIUM	CHRYSLER	7960	8466	9097	9643	NULL	NULL	NULL	NULL
0	08/05/2009	OTHER	2008	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	SILVER	AUTO	2	Covers	51963	AMERICAN ...	MEDIUM	CHRYSLER	7960	8466	9097	9643	NULL	NULL	NULL	NULL
0	08/05/2009	OTHER	2008	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	BLACK	AUTO	2	Covers	56669	AMERICAN ...	MEDIUM	CHRYSLER	7960	8466	9097	9643	NULL	NULL	NULL	NULL
0	08/14/2009	OTHER	2008	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	BLUE	AUTO	1	Alloy	52475	AMERICAN ...	MEDIUM	CHRYSLER	7930	8414	9064	9587	NULL	NULL	NULL	NULL
0	08/18/2009	OTHER	2008	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	SILVER	AUTO	2	Covers	51952	AMERICAN ...	MEDIUM	CHRYSLER	7958	8428	9095	9602	NULL	NULL	NULL	NULL
0	11/03/2010	OTHER	2009	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	WHITE	AUTO	2	Covers	45844	AMERICAN ...	MEDIUM	CHRYSLER	8698	9196	12824	13558	NULL	NULL	NULL	NULL
0	11/23/2010	OTHER	2009	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	GOLD	AUTO	2	Covers	38896	AMERICAN ...	MEDIUM	CHRYSLER	8835	9388	12456	13328	NULL	NULL	NULL	NULL
0	11/23/2010	OTHER	2009	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	RED	AUTO	2	Covers	41212	AMERICAN ...	MEDIUM	CHRYSLER	8835	9388	12456	13328	NULL	NULL	NULL	NULL
0	11/15/2010	OTHER	2009	1	CHRYSLER	SEBRING 4 ...	LX	4D SEDAN ...	SILVER	AUTO	2	Covers	47350	AMERICAN ...	MEDIUM	CHRYSLER	8781	9325	12464	13199	NULL	NULL	NULL	NULL

Figure 6. Car Lemon Dataset with NULL Errors.

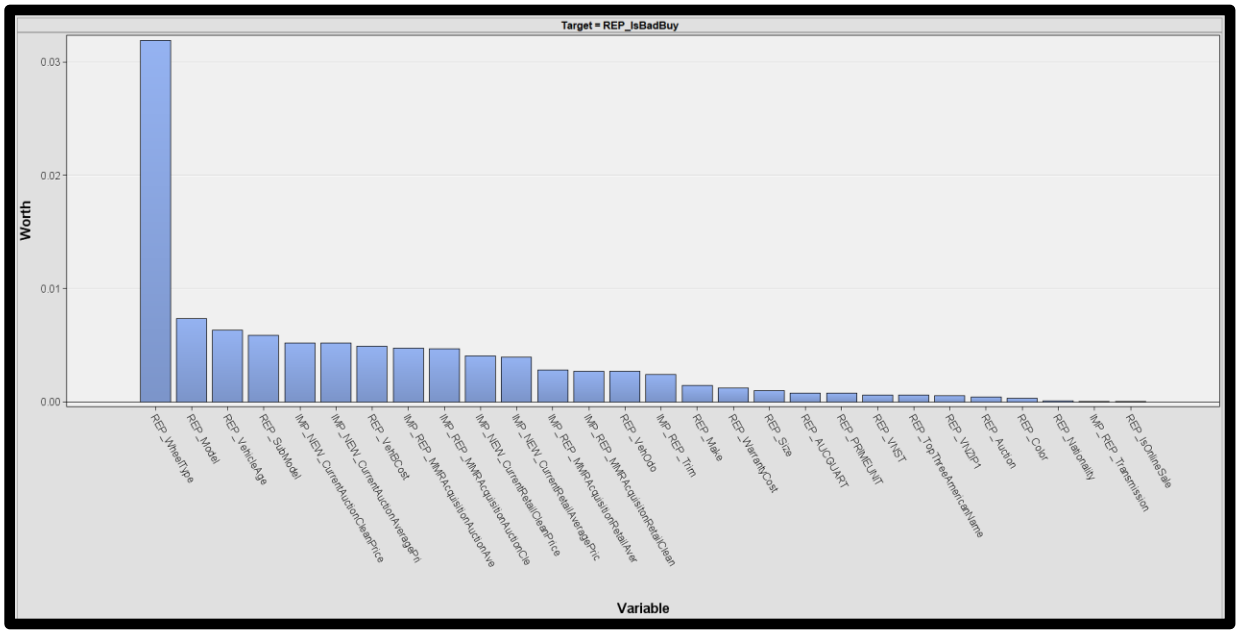


Figure 7. StatExplore Plot of Input Variable Worth.

Figure 8. Distributions of Variables in Car Lemon Dataset.

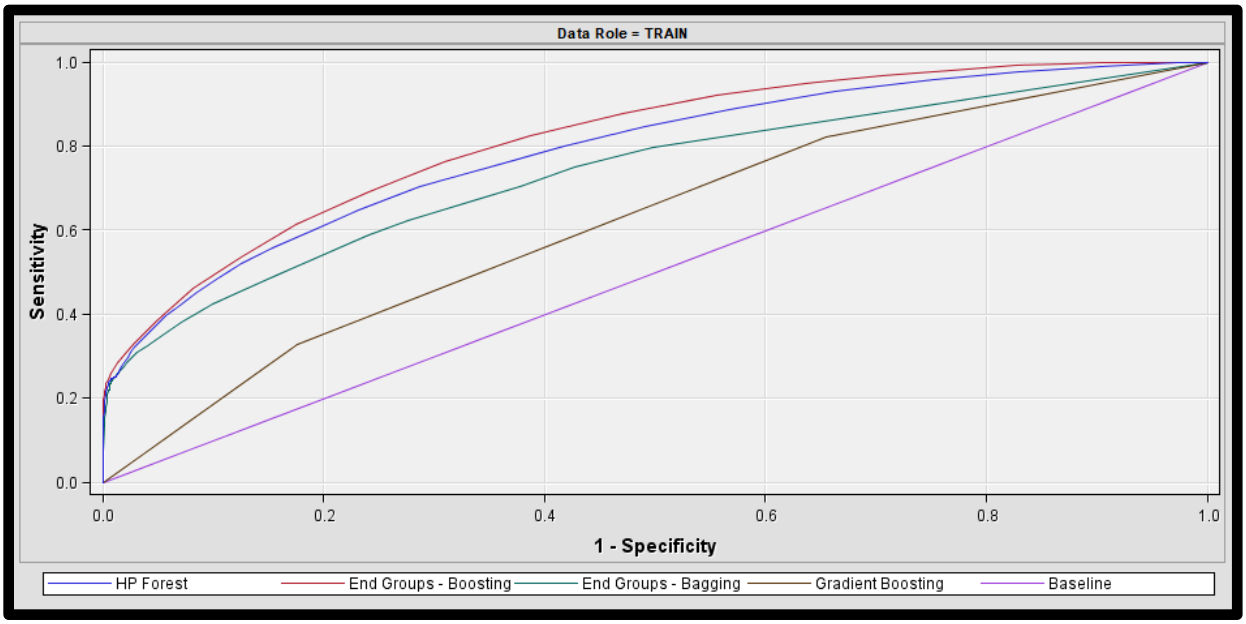


Figure 9. ROC Curve for Models in Initial Process Flow Diagram.

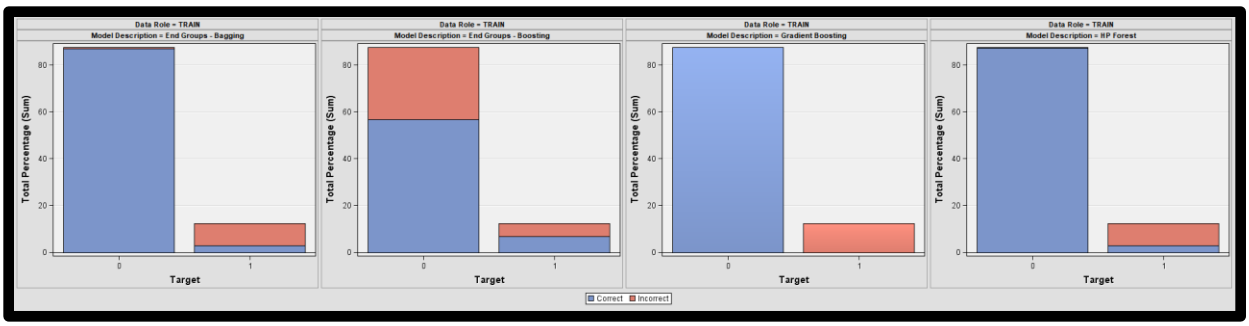


Figure 10. Classification Chart for Models in Initial Process Flow Diagram.

Selected Model	Predecessor Node	Model Node	Model Description ▲	Target Variable	Target Label	Selection Criterion: Train: Misclassification Rate	Train: Average Squared Error	Train: Divisor for ASE	Train: Maximum Absolute Error	Train: Sum of Frequencies
	EndGrp	EndGrp	End Groups - Bagging	REP_IsBad...	Replaceme...	0.100571	0.087049	145966	0.929135	72983
	EndGrp2	EndGrp2	End Groups - Boosting	REP_IsBad...	Replaceme...	0.363455	0.172631	145966	0.704396	72983
	Boost	Boost	Gradient Boosting	REP_IsBad...	Replaceme...	0.122988	0.107366	145966	0.882822	72983
Y	HPDMForest	HPDMForest	HP Forest	REP_IsBad...	Replaceme...	0.098804	0.085702	145966	0.974614	72983

Figure 11. Fit Statistics for Models in Initial Process Flow Diagram.

Event Classification Table
Model Selection based on Train: Misclassification Rate (_MISC_)

Model Node	Model Description	Data Role	Target	Target Label
EndGrp	End Groups - Bagging	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy
EndGrp2	End Groups - Boosting	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy
Boost	Gradient Boosting	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy
HPDMForest	HP Forest	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy

False Negative	True Negative	False Positive	True Positive
6868	63535	472	2108
4034	41515	22492	4942
8976	64007	0	0
6872	63668	339	2104

Figure 12. Classification Table for Models in Initial Process Flow Diagram.

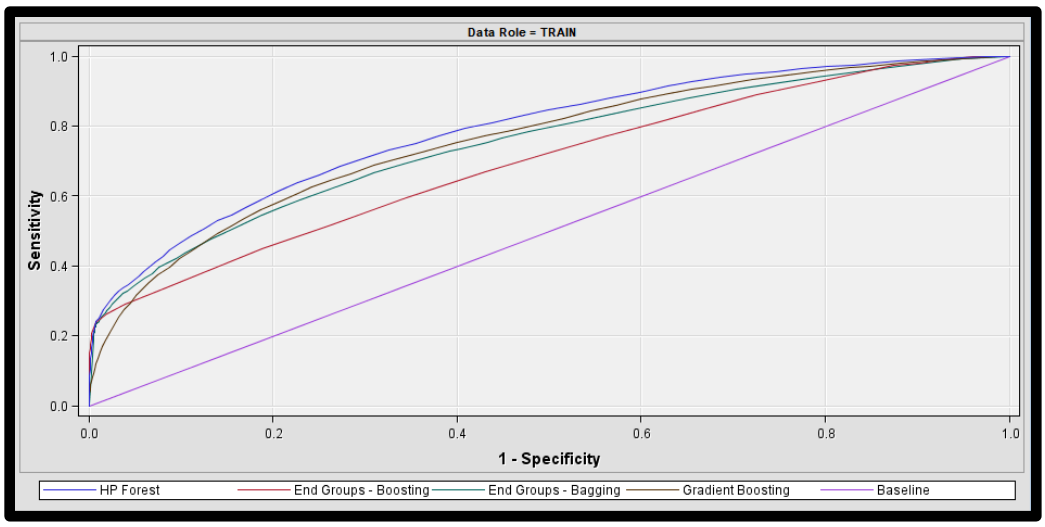


Figure 13. ROC Curve for Models with Sampling.

Selected Model	Predecessor Node	Model Node	Model Description ▲	Target Variable	Target Label	Selection Criterion: Train: Misclassification Rate	Train: Average Squared Error	Train: Divisor for ASE	Train: Maximum Absolute Error	Train: Sum of Frequencies
	EndGrp	EndGrp	End Groups - Bagging	REP_IsBad...	Replaceme...	0.30918	0.189837	44880	0.956063	22440
	EndGrp2	EndGrp2	End Groups - Boosting	REP_IsBad...	Replaceme...	0.500178	0.231253	44880	0.712571	22440
	Boost	Boost	Gradient Boosting	REP_IsBad...	Replaceme...	0.288547	0.195944	44880	0.876068	22440
Y	HPDMForest	HPDMForest	HP Forest	REP_IsBad...	Replaceme...	0.274911	0.182981	44880	0.893592	22440

Figure 14. Fit Statistics for Models with Sampling.

Event Classification Table
Model Selection based on Train: Misclassification Rate (_MISC_)

Model Node	Model Description	Data Role	Target	Target Label
EndGrp	End Groups - Bagging	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy
EndGrp2	End Groups - Boosting	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy
Boost	Gradient Boosting	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy
HPDMForest	HP Forest	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy

False Negative	True Negative	False Positive	True Positive
4906	11432	2032	4070
2764	5004	8460	6212
4780	11769	1695	4196
4981	12276	1188	3995

Figure 15. Classification Table for Models with Sampling.

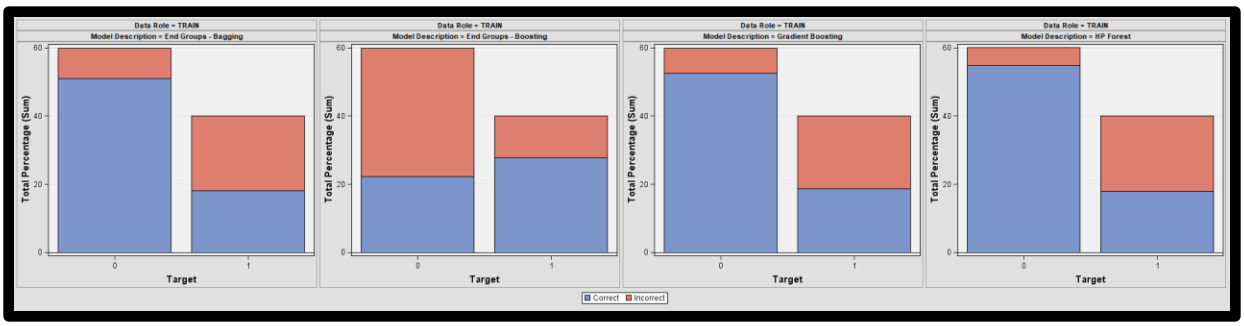


Figure 16. Classification Chart for Models with Sampling.

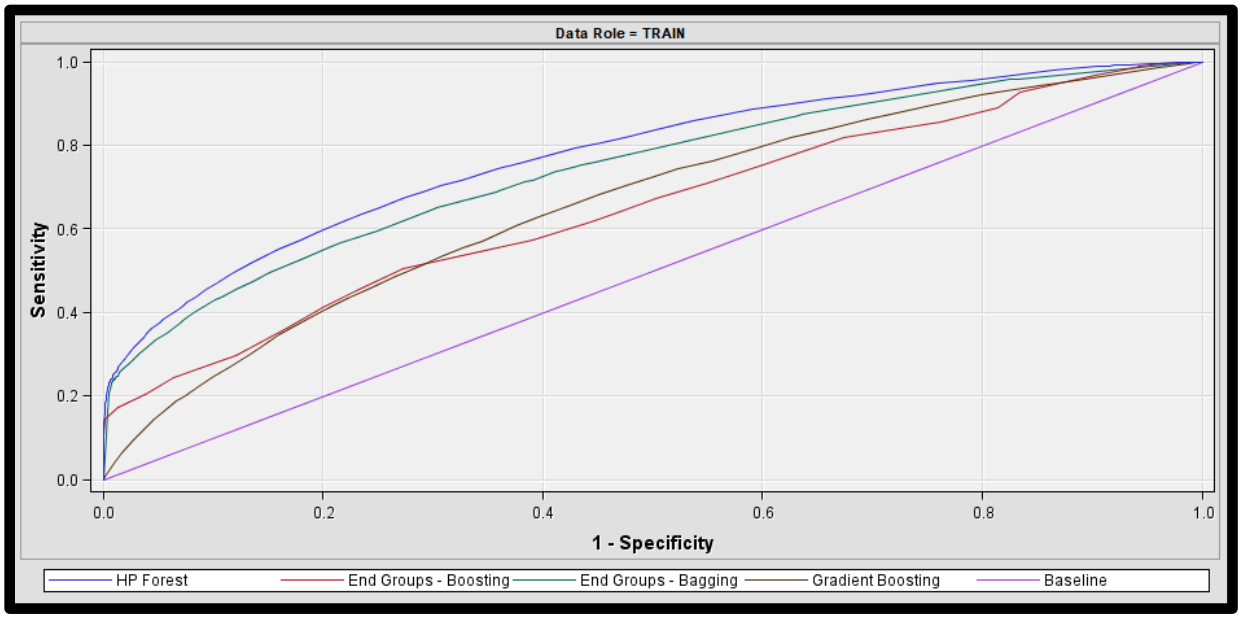


Figure 17. ROC Curve for Models with Sampling, Transforms, and Variable Selection.

Selected Model	Predecessor Node	Model Node	Model Description ▲	Target Variable	Target Label	Selection Criterion: Train: Misclassification Rate	Train: Average Squared Error	Train: Divisor for ASE	Train: Maximum Absolute Error	Train: Sum of Frequencies
	EndGrp	EndGrp	End Groups - Bagging	REP_IsBad...	Replaceme...	0.301248	0.191123	44880	0.946435	22440
	EndGrp2	EndGrp2	End Groups - Boosting	REP_IsBad...	Replaceme...	0.469251	0.235435	44880	0.661701	22440
	Boost	Boost	Gradient Boosting	REP_IsBad...	Replaceme...	0.358066	0.22198	44880	0.801505	22440
Y	HPDMForest	HPDMForest	HP Forest	REP_IsBad...	Replaceme...	0.274287	0.181057	44880	0.93545	22440

Figure 18. Fit Statistics for Models with Sampling, Transforms, and Variable Selection.

Event Classification Table
Model Selection based on Train: Misclassification Rate (_MISC_)

Model Node	Model Description	Data Role	Target	Target Label
EndGrp	End Groups - Bagging	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy
EndGrp2	End Groups - Boosting	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy
Boost	Gradient Boosting	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy
HPDMForest	HP Forest	TRAIN	REP_IsBadBuy	Replacement: IsBadBuy

False Negative	True Negative	False Positive	True Positive
5941	12645	819	3035
3809	6743	6721	5167
5880	11313	2151	3096
4889	12198	1266	4087

Figure 19. Classification Table for Models with Sampling, Transforms, and Variable Selection.

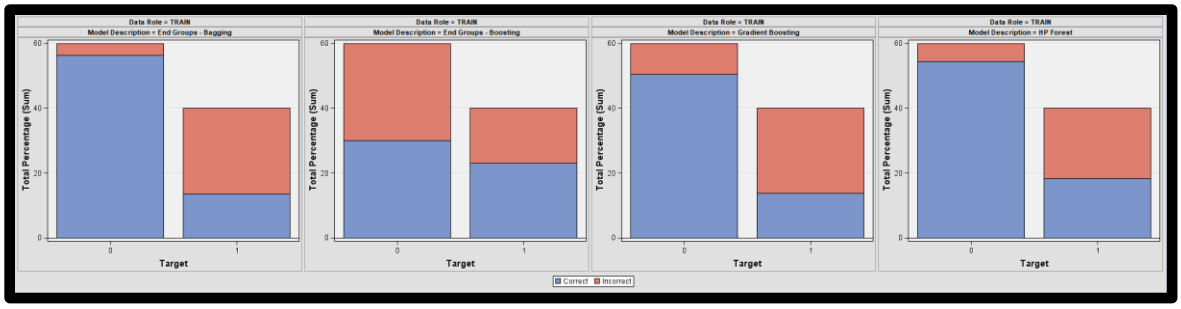


Figure 20. Classification Chart for Models with Sampling, Transforms, and Variable Selection.

Target	Target Label	Fit Statistics	Statistics Label ▲	Train	Validation	Test
IsBadBuy		_ASE_	Average Squared Error	0.085185	.	.
IsBadBuy		_DIV_	Divisor for ASE	145966	.	.
IsBadBuy		_DISF_	Frequency of Classified Cases	72983	.	.
IsBadBuy		_MAX_	Maximum Absolute Error	0.982691	.	.
IsBadBuy		_MISC_	Misclassification Rate	0.099626	.	.
IsBadBuy		_WRONG_	Number of Wrong Classifications	7271	.	.

Figure 21. Fit Statistics for Random Forest Model with no Data Preparation.

Event Classification Table

Data Role=TRAIN Target=IsBadBuy Target Label=' '

False Negative	True Negative	False Positive	True Positive
6818	63554	453	2158

Figure 22. Classification Table for Random Forest Model with no Data Preparation.