**University of Maryland University College**

**DATA 640 – Predictive Modeling**

**SUMMER 2017**

**Assignment 5 - Ensembles**

# Testing the Kaggle Car Dataset with Ensemble and Predictive Models to discover the best Model in Selecting Kicked Cars.

**John Parsons**

jparsons20@student.umuc.edu

**Professor Knode**

**Abstract**

The objectives for this project are to test several Ensemble Models from Maldonado et. al. (2014) and compare these with previous models using the Kaggle Kicked Car Data set. Ensemble models generally give a better overall prediction when compared to other predictive analytical models and this will be tested to see how well it performs with the Kaggle dataset. The optimization template will be used to see which parameters for Bagging, Boosting, Gradient Boosting and HP Forest Models will be used for this study. The top four models will be selected from seven groups of models and the best model will be tested with the original Kaggle Test data set to determine how well it performs in this study. The Kaggle Test Data set has 48,707 entries and all Target variables are 0 or classified as NOT kicked.

**Introduction**

The CARVANA lemon car training data set from Kaggle will be used for this project in creating the best predictive models and Ensemble Models to determine which cars will be kicked by using the SEMMA approach. This Kaggle data set was used for Support Vector Machines (SVM) Assignment Three and will be briefly discussed. The dataset contains one binary dependent variable (**IsBadBuy**) and a total of 33 independent variables (such as **Auction, Make, VehOdo and VNZIP**). The **IsBadBuy** Dependent variable is skewed with 8,976 to 64,007 entries (12.3 to 87.7 %) (Figure 2A) classified as kicked or not kicked cars in this dataset. A model can be 87% correct in selecting all cars as **not kicked** and still have a better than average prediction, but this does not help the consumer or car dealers in not buying cars that are a bad investment. The list of variables, inputs, levels and description can be seen in Figure 1A (Appendix) (Please note that all Figures that end in a vowel will be at the end of this document).

## <u>Data Cleansing and Preparation</u>

Modifications were made to this dataset and all **NULL** values were replaced with empty cells in MS Excel and then the **NULL** transformed data set was uploaded into SAS to begin this study. Literature and class notes discuss how Ensembles and Decision Trees are easy to initiate and can handle missing values and outliers, which will be tested in this study (Rush and Baker, ND). The Kaggle training file has a total of 72,983 data entries and the Fit Statistics for this modified data base can be seen in Figure 1. Please note the missing values for **PRIMEUNIT** and **AUCGUART** variables as shown in Figure 2 and the number of variables with missing values (16 variables). A screenshot of this modified database can be seen in Figure 2. **PRIMEUNIT** and **AUCGUART** variables were rejected during the SAS File Save and this dataset only has a total of 32 variables. The **PunchDate** and **RefID** Variables were rejected for this study and these were also rejected for the SVM Models.

**Figure 1: Fit Statistics for all 32 variables for the SAS Kicked Training Database.**

Variables - Ids

(none) | not | Equal to | Apply | Reset

Columns: Label | Mining | Basic | Statistics

| Name | Role | Level | Drop | Number of Levels | Percent Missing | Minimum | Maximum | Mean | Standard Deviation | Skewness | Kurtosis |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Auction | Input | Nominal | No | 3 | 0 | . | . | . | . | . | . |
| BYRNO | Input | Interval | No | . | 0 | 835 | 99761 | 26345.84 | 25717.35 | 2.129225 | 3.474184 |
| Color | Input | Nominal | No | 16 | 0.010961 | . | . | . | . | . | . |
| IsBadBuy | Target | Binary | No | 2 | 0 | . | . | . | . | . | . |
| IsOnlineSale | Input | Binary | No | 2 | 0 | . | . | . | . | . | . |
| MMRAcquisitionAuctionAveragePric | Input | Interval | No | . | 0.024663 | 0 | 35722 | 6128.909 | 2461.993 | 0.463641 | 1.593728 |
| MMRAcquisitionAuctionCleanPrice | Input | Interval | No | . | 0.024663 | 0 | 36859 | 7373.636 | 2722.492 | 0.466501 | 1.651147 |
| MMRAcquisitionRetailAveragePrice | Input | Interval | No | . | 0.024663 | 0 | 39080 | 8497.034 | 3156.285 | 0.209214 | 0.680599 |
| MMRAcquisitonRetailCleanPrice | Input | Interval | No | . | 0.024663 | 0 | 41482 | 9850.928 | 3385.79 | 0.1763 | 0.924827 |
| MMRCurrentAuctionAveragePrice | Input | Interval | No | . | 0.431607 | 0 | 35722 | 6132.081 | 2434.568 | 0.522583 | 1.529939 |
| MMRCurrentAuctionCleanPrice | Input | Interval | No | . | 0.431607 | 0 | 36859 | 7390.682 | 2686.249 | 0.535525 | 1.56562 |
| MMRCurrentRetailAveragePrice | Input | Interval | No | . | 0.431607 | 0 | 39080 | 8775.723 | 3090.703 | 0.201356 | 0.63885 |
| MMRCurrentRetailCleanPrice | Input | Interval | No | . | 0.431607 | 0 | 41062 | 10145.39 | 3310.254 | 0.19478 | 0.847008 |
| Make | Input | Nominal | No | 21 | 0 | . | . | . | . | . | . |
| Model | Input | Nominal | No | 21 | 0 | . | . | . | . | . | . |
| Nationality | Input | Nominal | No | 4 | 0.006851 | . | . | . | . | . | . |
| PurchDate | Rejected | Interval | No | . | 0 | . | . | . | . | . | . |
| RefId | Rejected | Interval | No | . | 0 | 1 | 73014 | 36511.43 | 21077.24 | -.000203 | -1.19996 |
| Size | Input | Nominal | No | 12 | 0.006851 | . | . | . | . | . | . |
| SubModel | Input | Nominal | No | 21 | 0 | . | . | . | . | . | . |
| TopThreeAmericanName | Input | Nominal | No | 4 | 0.006851 | . | . | . | . | . | . |
| Transmission | Input | Binary | No | 2 | 0.012332 | . | . | . | . | . | . |
| Trim | Input | Nominal | No | 20 | 4.367863 | . | . | . | . | . | . |
| VNST | Input | Nominal | No | 21 | 0 | . | . | . | . | . | . |
| VNZIP1 | Input | Interval | No | . | 0 | 2764 | 99224 | 58043.06 | 26151.64 | -0.10353 | -1.68869 |
| VehBCost | Input | Interval | No | . | 0 | 1 | 45469 | 6730.934 | 1767.846 | 0.715931 | 8.144378 |
| VehOdo | Input | Interval | No | . | 0 | 4825 | 115717 | 71500 | 14578.91 | -0.45315 | -0.19874 |
| VehYear | Input | Nominal | No | 10 | 0 | . | . | . | . | . | . |
| VehicleAge | Input | Nominal | No | 10 | 0 | . | . | . | . | . | . |
| WarrantyCost | Input | Interval | No | . | 0 | 462 | 7498 | 1276.581 | 598.8468 | 2.070831 | 9.964808 |
| WheelType | Input | Nominal | No | 3 | 4.348958 | . | . | . | . | . | . |
| WheelTypeID | Input | Nominal | No | 4 | 4.342107 | . | . | . | . | . | . |

**Figure 2: Screenshot of the Kaggle Kicked Care Data Base.**

Preview

| ty | Size | TopThreeAmericanName | MMRAcquisitionAuctionAveragePric | MMRAcquisitionAuctionCleanPrice | IsBadBuy | MMRAcquisitionRetailAveragePrice | MMRAcquisitonRetailCleanPrice | MMRCurrentAuctionAveragePrice | MMRCurrentAuctionCleanPrice | MMRCurrentRetailAveragePrice | MMRCurrentRetailCleanPrice | PRIMEUNIT | AUCGUART | BYRNO | VN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ASI... | MEDIUM | OTHER | 8155 | 9829 | 0 | 11636 | 13600 | 7451 | 8552 | 11597 | 12409 | | | 21973 | |
| AN ... | LARGE TR... | CHRYSLER | 6854 | 8383 | 0 | 10897 | 12572 | 7456 | 9222 | 11374 | 12791 | | | 19638 | |
| AN ... | MEDIUM | CHRYSLER | 3202 | 4760 | 0 | 6943 | 8457 | 4035 | 5557 | 7146 | 8702 | | | 19638 | |
| AN ... | COMPACT | CHRYSLER | 1893 | 2675 | 0 | 4658 | 5690 | 1844 | 2646 | 4375 | 5518 | | | 19638 | |
| AN ... | COMPACT | FORD | 3913 | 5054 | 0 | 7723 | 8707 | 3247 | 4384 | 6739 | 7911 | | | 19638 | |
| ASI... | MEDIUM | OTHER | 3901 | 4908 | 0 | 6706 | 8577 | 4709 | 5827 | 8149 | 9451 | | | 19638 | |
| ASI... | MEDIUM | OTHER | 2966 | 4038 | 0 | 6240 | 8496 | 2980 | 4115 | 6230 | 8603 | | | 19638 | |

## Predictive Models Developed

This study was divided into three sections to find the best models for this dataset. The first section is the optimization of the Bagging, Boosting, Gradient Boosting and HP Forest models. The output for the HP Forest Modifications can be seen in Figures 4AA, 5A and 6A. The list of parameters tested for each optimized model can be seen in Figure 3A and the Confusion Matrix Results with the best models are highlighted in yellow for Figure 7A. The data in this group was not transformed or imputed before running the models.

The second section created a total of seven groups (Models 1A to 1G) of models which tested individual predictive models and Ensemble Models. The model modifications and names of each model for these seven groups can be seen in **Figures 5, 6 and 7.**

**Figure 5: Model Template for Groups A to B:**

| Final Analytical Models | Model Modifications |
|---|---|
| **Group A Models** | |
| **1A Decision Tree** | Standard Decision Tree connected to the data source directly using default settings and Misclassification for Assessment. |
| **2A Step Log Reg** | Logistic Regression Model connected to a Data Partition Node (70:20:10) and the Imputation Node set to median value for input variables. The Data Partition and Imputation settings are the same for all models in this diagram. The selection criteria is set to Validation Misclassification and all other settings are default. |
| **3A SVM Linear** | The HP SVM Node was also connected to the Data Partition and Imputation node. The Interior Linear Settings were used for this node. |
| **4A Neural Net** | The Neural Net Node was also connected to the Data Partition and Imputation Node. The Neural Net had a total of 3 hidden units with 1 layer. |
| **Ensemble 1A** | Ensemble Model was connected to the Decision Tree, Logistic Regression, SVM and Neural Network Nodes |
| **Group B Models** | |
| **1B Bagging with Decision Tree** | The Start Node was changed to Bagging with and Index count of 10 and Bagging % set to 10 (default). These settings will be used for all Bagging Diagrams. The Decision Tree with default settings and Stop Node was used for this model. |
| **2B Boosting** | The Start Node was changed to Boosting and the Index Count was changed to 20 and the Decision Tree (default settings) and Stop Nodes were connected. |
| **3B Bagging with Regression** | Maldonado et. al. (2014) said most models could be connected to these models. The Previous Bagging Settings were used and the Stepwise Logistic Regression Model replaced the Decision Tree. The Validation Misclassification Criteria was used for selection criteria and was connected to the Data Partition Node (same settings as before). |
| **4B Boosting with Regression** | The Previous Boosting Settings were used and the Stepwise Logistic Regression Model replaced the Decision Tree. The Validation Misclassification Criteria was used for selection criteria and was connected to the Data Partition Node (same settings as before). |
| **5B Boosting with Neural Nets** | The Previous Boosting Settings were used and the Neural Network Model replaced the Decision Tree. The Neural Net had 3 hidden unites with one layer. The Misclassification Criteria was used for selection criteria and was connected to the Data Partition Node (same settings as before). |
| **5B Gradient Boosting** | The Gradient Boosting Node with 100 N Iterations was connected directly to the data node. Remaining settings used the default settings. |
| **6B HP Forest** | The HP Forest Node was attached directly to the data and the optimization settings were used. This had a total 70 trees, 20 variables and 3 for smallest number of observations. |
| **Ensemble 2B** | The Decision Tree Bagging, Boosting Logistic Regression Bagging and Boosting and Gradient Boosting Models were attached to the Ensemble node. |

**Figure 6: Model Template for Groups C, D, E and F.**

Predictive Models Criteria: Part B

| Final Analytical Models | Model Modifications |
|---|---|
| **Group C Neural Network Models (all connected to Data Partition (70:20:10 Split). This Model came from Maldanodo et. al. (2014)** | |
| **1C Neural Net** | Standard Neural Net Node was used for this model and this had a total of 3 hidden units with only 1 layer. |
| **2C Neural Net** | Standard Neural Net Node was used for this model and this had a total of 10 hidden units with only 1 layer. |
| **3C Neural Net** | Standard Neural Net Node was used for this model and this had a total of 30 hidden units with only 1 layer. |
| **4C Neural Net** | Standard Neural Net Node was used for this model and this had a total of 50 hidden units with only 1 layer. |
| **1C Ensemble** | All four Neural Nets (1C to 4C) were connected to this Ensemble. |
| **2C Ensemble** | Only three Neural Nets (2C-4C) were connected to this Ensemble. |
| **3C Ensemble** | Only two Neural Nets (3C-4C) were connected to this Ensemble. |
| **Group D Models or Ensemble Rotation Forest which came from Maldanodo et. al. (2014).** | |
| **4DD Ensemble Model** | This is a Rotation Forest which has five rows of a Sample Node, SAS Code (1 to 5), Principal Component Nodes, and Decision Trees that are all connected to the Ensemble Node 4DD. The default settings were used for the sample node (100%), Principal Components Node. The default settings were used for the Decision Tree but the Misclassification criteria was selected for this node. The SAS Code used the same code except the keep_flag value was changed to represent the correct code or row. The SAS Diagram can be found in the Figure 1BB |
| **Group E Models or Sequential Boot Strap Aggregating Algorithms.** | |
| **5EE SEQ Final Model** | This is a Sequential Boot Strap Aggregation Model that came from the Maldonado et. al. (2014) article. This sequential iteration assigns a larger weight to incorrect observations. The data set was connected to three rows of a Sample Node (100%), Decision Tree with default settings (criteria set to misclassification) and the SAS Codes (1 to 3). All SAS Codes had the same SAS Code as referenced in Figure 3BB but the Frequency was changed from 32 down to 15. The last Decision Tree was connected to Final Model Compare |
| **Group E Models or Sequential Boot Strap Aggregating Algorithms that replaces the Decision Tree with the HP Forest Node.** | |
| **6FF Final HP Forest Model** | The Sequential Boot Strap Aggregation Model from before was duplicated except the HP Forest Nodes were used instead of the Decision Trees. The HP Forest Nodes were set to 70 number of trees, 20 number of variables and 3 for the smallest number of observations in the node. All other settings were identical. |

**Figure 7: Model Template for Groups G.**

Predictive Models Criteria: Part C

| Final Analytical Models | Model Modifications |
|---|---|
| **7GG Stacked Forest Model from Czika et al. (2016).** | |
| **1st Set of Stacked Models** | Three HP Neural Nets were created for this Stacked Model. The first NN was connected to the Best Transformed Node and was set to 10 hidden units with only 1 Layer. The Second NN was also attached to the Best Transformed Node but also to the Imputation Node (set to Median). The Third NN was attached to the Forward Logistic Regression (Selection Criteria None) which was Best Transformed and Imputed. |
| **2nd Set of Stacked Models** | The second set of Stacked Models was attached directly to the data set and then the default settings were used for the Gradient Boosting Model, HP Forest Model (Maximum Number of Trees set to 100) and the Support Vector Machine Model. The two HP Tree Models were set to default but had the Target Criterion set to Entropy or Chi Square. All Stacked Models were attached to a Model Compare Node. |

Some models in groups A, B, C and F had the Data Partition Node set to a 70:20:10 split for the training, validation and testing set. Models A and F also had the **Imputation** and **Transform Nodes** added for several models to replace the missing values with the median value and to transform (Max Transform or Best) which smoothed out the skewed variables in this data set. A total of 14 variables had standard deviations higher than 3.0 and the highest one was 26,151 for the **VINZIP1** variable.

## Ensemble Model Results

The complete results for the seven groups and top four models can be seen in Figure 8A

**Figure 8: Fit Statistics and Confusion Matrix for the top four models.**

| Top Four Models from Final Model Comparison Node | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Precision | Train/Vald. Misclassification |
|---|---|---|---|---|---|---|---|---|
| #1 IS 5EE_SEQ Final Decision Tree TRAIN | 16 | 63995 | 12 | 8960 | 0.429 | 1.000 | 0.999 | 0.0004 |
| #2 HPDMForest5 6FF_Final HP Forest TRAIN | 19 | 63946 | 61 | 8957 | 0.763 | 0.999 | 0.993 | 0.0011 |
| #3 HPNNA3 3G_30 HU HP Neural TRAIN | 5576 | 63179 | 828 | 3400 | 0.129 | 0.987 | 0.804 | 0.0878 |
| #4 nsmbl7 2B_Ensemble TRAIN | 8973 | 64007 | 0 | 3 | 0.000 | 1.000 | 1.000 | 0.1230 |

**Figure 9: Fit Statistics for the Rotation Forest, Sequential Bootstrap and Stacked Model.**

| Model 4: Ensemble Rotatinon Forest | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Precision | Train Misclassification |
|---|---|---|---|---|---|---|---|---|
| 4DD Ensemble Rotation Forest | 8976 | 64007 | 0 | 0 | 0.000 | 1.000 | . | 0.1230 |

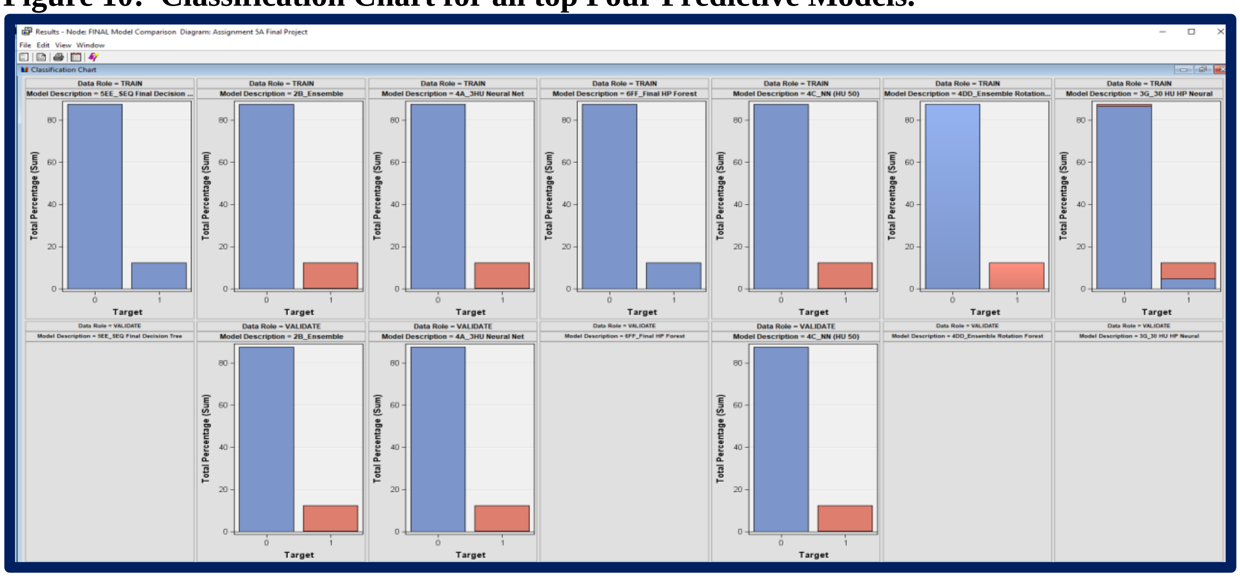
| Model 5: Sequential Bootstrap Aggregated Algorithm | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Precision | Train Misclassification |
|---|---|---|---|---|---|---|---|---|
| 5EE_SEQ Final Decision Tree Boostrap | 16 | 63995 | 12 | 8960 | 0.429 | 1.000 | 0.999 | 0.0004 |

| Model 6: SEQ Final HP Forest | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Precision | Train Misclassification |
|---|---|---|---|---|---|---|---|---|
| 6FF_Final HP Forest SEQ | 19 | 63946 | 61 | 8957 | 0.763 | 0.999 | 0.993 | 0.0011 |

| Model 7: Stacked Model | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Precision | Train Misclassification |
|---|---|---|---|---|---|---|---|---|
| 1G 10 HU HP Neural Net | 6105 | 62973 | 1034 | 2871 | 0.145 | 0.984 | 0.735 | 0.098 |
| 2G 15 HU HP Neural Net | 5362 | 62709 | 1298 | 3614 | 0.195 | 0.980 | 0.736 | 0.091 |
| 3G 30 HU HP Neural Net | 5576 | 63179 | 828 | 3400 | 0.129 | 0.987 | 0.804 | 0.088 |
| Gradient Boosting | 8976 | 64007 | 0 | 0 | 0.000 | 1.000 | #DIV/0! | 0.123 |
| 1G Entropy HP Tree | 8645 | 63854 | 153 | 331 | 0.017 | 0.998 | 0.684 | 0.121 |
| 1G CHI Square HP Tree | 8951 | 64003 | 4 | 25 | 0.000 | 1.000 | 0.862 | 0.123 |
| 1G HP Forest | 6840 | 63768 | 239 | 2136 | 0.034 | 0.996 | 0.899 | 0.097 |
| 1G HP SVM | 8815 | 63940 | 67 | 161 | 0.008 | 0.999 | 0.706 | 0.122 |

and 9A. The results for the top four models of this project and several select Ensemble models can be found in Figures 8 and 9. The best models for predicting the most True Positives with the lowest Misclassification Rates came from the sequential boosted bootstrap aggregating models. The best model for this study is **5EE_SEQ Final** with 8,960 True Positives and a 0.0004 Misclassification Rate and **6_FF Final HP** Forest with 8,957 True Positives and a 0.0011 Misclassification Rate. Model **6_FF** has the same diagram flow as 5EE but with HP Forest Nodes. These algorithms with Decision Trees and even HP Forest Nodes attached to the diagram flow came from the Maldonado Article (2014) (Figure 13A). The Specificity (True Negatives) was 1 and 0.99 while the Sensitivity (True Positives was a little high or 0.429 and 0.763). These numbers were a little distorted due to the very small sample size for the False Negatives and False Positives. The Classification chart in Figure 10 shows that Models **5_EE** and **6_FF** (1st and 4th top diagrams) shows these models did the best job in correctly identifying the kicked and not kicked values for the Target variables. The third best model was **3G** or the Neural Net Model and this identified a total of 3,400 True Positives with a Misclassification Rate of 0.0878. This model came from the Stacked Model design from Czika, Maldonado and Liu (2016).

**Figure 10: Classification Chart for all top Four Predictive Models.**




Ensemble Model 2B came in fourth place with a Misclassification Rate of 0.1230 and only correctly identified 3 True Positives and had 8,973 False Positives. The Ensembles for this study did not do a good job in identifying the True Positives and had a low Sensitivity Rate, but did a good job in identifying the False Positives. The Gradient Boosting and Neural Net Models had very similar results as the Ensemble models and did not pick a lot of the True Positives but did a good job in identifying True Negatives.

Model **5EE** Sequential Boosted Bootstrap Algorithm and **3G** Neural Net Model was tested using the Kaggle Test Data which had 48,707 attributes and all Dependent variables were 0 or NOT Kicked. The Null Values were replaced with and empty cell and had the same parameters as the training data. This model misidentified a total of 1,780 samples or 3.66% of the samples and was not as efficient in separating the True Positives from the True Negatives, but still did a good job in identifying True Negatives. Model 3G misidentified 3,086 of these attributes and was slightly lower than the training data set from the previous model.

**Figure 11: Scoring Results for Model 5EE.**

Class Variable Summary Statistics

Data Role=SCORE Output Type=CLASSIFICATION

| Variable | Numeric Value | Formatted Value | Frequency Count | Percent |
|---|---|---|---|---|
| I_IsBadBuy | . | 0 | 46927 | 96.3455 |
| I_IsBadBuy | . | 1 | 1780 | 3.6545 |

**Figure 12: Scoring Results for Model 3G Neural Net**

Class Variable Summary Statistics

Data Role=SCORE Output Type=CLASSIFICATION

| Variable | Numeric Value | Formatted Value | Frequency Count | Percent |
|---|---|---|---|---|
| I_IsBadBuy | . | 0 | 45621 | 93.6642 |
| I_IsBadBuy | . | 1 | 3086 | 6.3358 |

## Conclusions

The goal of this study was to optimize the Bagging, Boosting, Gradient Boosting and HP Forest Models and test several Ensemble Models in determining which cars would be Kicked or True Positives. The best models in this study are the sequential boosted bootstrap aggregating algorithms for **5EE** and **6FF.** This type of model was also used for the previous assignment and had the best predictive results for the training data set. The sequential iteration of steps with assigning different weights based on the previous prediction of being right or wrong had the best outcome for this study (Maldonado et al. 2016). These types of models are winning data competitions and are very flexible in their approach in handling missing variables and outliers. However, these models are classified as black boxes because they are almost impossible to understand and even banned from certain institutions like banking (Rush and Baker, ND.). Figure 14A shows a Decision Tree from Model 1A and this can be easily interpreted, but it is not as accurate as these other models. This is the area that needs to be optimized as these black box models are being developed and becoming good predictive models.

This is the Final Assignment for MS 640 and has been an incredible 12 weeks. SAS Miner 14.2 is a phenomenal program in its approach for using the SEMMA methodology in creating predictive models. The SAS Community is very diligent in answering our questions as we grow in this ever-changing field. These 12 weeks have been incredibly challenging, but very rewarding to me and I look forward to learning more about SAS during my remaining time at the University of Maryland University College.

## References

Abbot, D. (2014). Applied Predictive Analytics. Principals and Techniques for the Professional Data Analyst. John Wiley & Sons, Inc. Indianapolis, Indiana. Chapters 3-4.

Czika, W., Maldonado, M. and Liu, L. (20.16). Ensemble Modeling: Recent Advances and Applications. Paper SAS3120-2016. Retrieved from: https://support.sas.com/resources/papers/proceedings16/SAS3120-2016.pdf

Kaggle, (ND). CARVANA, Don't Get Kicked! Retrieved from: https://www.kaggle.com/c/DontGetKicked/data

Kattamuri, S. (2013). Predictive Modeling with SAS Enterprise Miner. Practical Solutions for Business Applications. SAS Institute, Inc. Cary, NC, USA. Second Edition.

Maldonado, M., Dean, J., Czika, W. and Haller, S. (2014). Leveraging Ensemble Models in SAS Enterprise Miner. Paper SAS133-2014. Retrieved from: https://support.sas.com/resources/papers/proceedings14/SAS133-2014.pdf

Rush, M. and Baker, T. Ensemble Models and Partitioning Algorithms in SAS Enterprise Miner. Retrieved from:https://communities.sas.com/kntur85557/attachments/kntur85557/library/2042/1/Ensemble%20Models%20and%20Partitioning%20Algorithms%20in%20SAS%C2%AE%20Enterprise%20Miner%20-%20N....pdf

SAS books (ND). Data mining using SAS Enterprise Miner: A case study approach. SAS Institute Inc., Cary, N.C. Chapter 1. Retrieved from http://support.sas.com/documentation/cdl/en/emcs/66392/PDF/default/emcs.pdf

SAS books. (2016). Applied Analytics Using SAS Enterprise Miner Course Notes. SAS Institute, Inc. Cary, NC, USA. Chapters 1-6.

Wujek, B. (October 20, 2015). Random forest and support vector machines getting the most from your classifiers [Web]. Retrieved from: https://www.youtube.com/watch?v=EOxwpnbFqIU

Wujek, B. (2015). Tip: Getting the most from your Random Forest. Retrieved from: https://communities.sas.com/t5/tkb/articleprintpage/tkb-id/library/article-id/757?nobounce

## Appendix

**Figure 1A: CARVAN Kicked Car Data Variable Names and Descriptions.**

| Variable Name | Role | Level | Description |
|---|---|---|---|
| AUCGUART | Removed | Nominal | Auction level guarantee (Green light - Guaranteed/arbitrable, Yellow Light - caution/issue, red light - sold as is). |
| Auction | Input | Nominal | Auction provider of purchased vehicle |
| BYRNO | Input | Interval | Unique number assigned to the buyer that purchased the vehicle. |
| Color | Input | Nominal | Vehicle Color. |
| IsBadBuy | Target | Binary | Identifies if the kicked vehicle was an avoidable purchase (0=good, 1=lemon) |
| IsOnlineSale | Input | Binary | Identifies if vehicle was purchased online. |
| MMRA_1* | Input | Interval | average condition at time of purchase. |
| MMRA_2* | Input | Interval | above Average condition at time of purchase. |
| MMRA_3* | Input | Interval | retail market in average condition at time of purchase. |
| MMRA_4* | Input | Interval | retail market in above average condition at time of purchase. |
| MMRC_5* | Input | Interval | in average condition, as of current day. |
| MMRC_6* | Input | Interval | in the above condition, as of current day. |
| MMRC_7* | Input | Interval | retail market in average condition as of current day. |
| MMRC_8* | Input | Interval | retail market in above average condition as of current day. |
| Make | Input | Nominal | Vehicle Manufacturer. |
| Model | Input | Nominal | Vehicle Model. |
| Nationality | Input | Nominal | The Manufacturer's country. |
| PRIMEUNIT | Removed | Interval | Identifies if the vehicle would have a higher demand than a standard purchase. |
| PunchDate | Rejected | Interval | Date the Auction vehicle was Purchased. |
| RefID | Rejected | Interval | Unique (sequential) vehicle number. |
| Size | Input | Nominal | Vehicle Size (Compact, SUV, etc.). |
| SubModel | Input | Nominal | Vehicle Sub model. |
| TopThreeAmerican Name | Input | Nominal | Identifies if the manufacturer is one of the top three American manufacturers. |
| Transmission | Input | Nominal | Transmission type (Automatic, Manual). |
| Trim | Input | Nominal | Vehicle Trim Level. |
| VNST | Input | Nominal | State where the car was purchased. |
| VNZIP1 | Input | Interval | Zip code where the car was purchased. |
| VehBCost | Input | Interval | Acquisition cost paid for the vehicle at time of purchase. |
| VehOdo | Input | Interval | The vehicles odometer reading. |
| VehYear | Input | Nominal | The manufacturer's year of the vehicle. |
| VehAge | Input | Nominal | Age of vehicle at time of sale. |
| WarrantyCost | Input | Interval | Warranty price (term=36month and millage=36K). |
| WheelType | Input | Nominal | Vehicle wheel type (Alloy, Covers). |
| WheelTypeID | Input | Nominal | The type id of the vehicle wheel |

**NOTES: MMRA_1 to MMRA_8 entire variable name is listed below and all entries description begins with "Acquisition price for this vehicle in".**
MMRA_1: is MMRAcquisitionAuctionAveragePrice
MMRA_2: is MMRAcquisitionAuctionCleanPrice
MMRA_3: is MMRAcquisitionRetailAveragePrice
MMRA_4: is MMRAcquisitionRetailCleanPrice
MMRA_5: is MMRCurrentAuctionAveragePrice
MMRA_6: is MMRCurrentAuctionCleanPrice
MMRA_7: is MMRCurrentRetailAveragePrice
MMRA_8: is MMRCurrentRetailCleanPrice

**Figure 2A: The IsBadBuy Skewed Distribution for the Target Variable (SAS JMP13).**

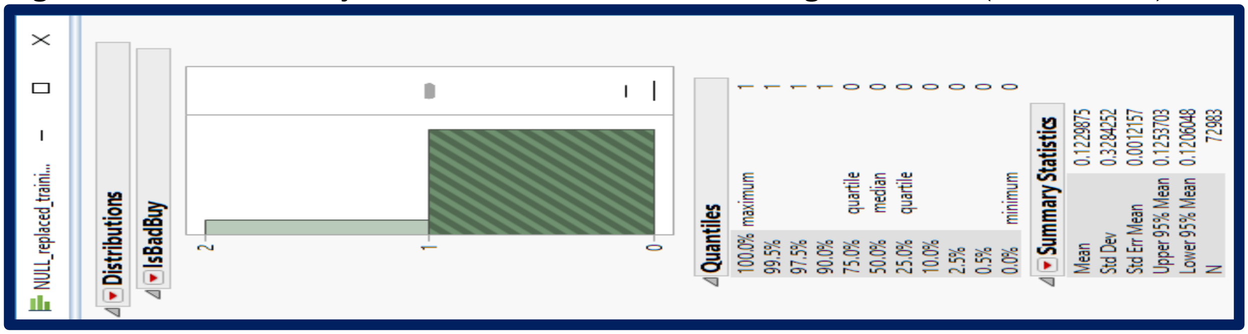
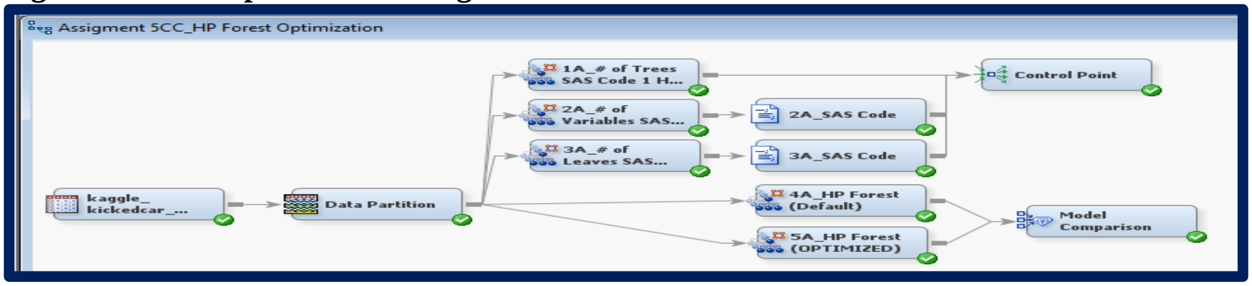



**Figure 3A: Optimization Template for Bagging, Boosting, Gradient and Forest Models.**

| | |
|---|---|
| **Bagging Optimization Model Criteria for Decision Trees 1A to 1D (Model 1A)** | |
| **Pre-Bagging Models** | **Model Modifications** |
| **1A_Decision Tree** | Start Groups Mode set to Bagging and Index count set to 5. |
| **1B_Decision Tree** | Start Groups Mode set to Bagging and Index count set to 10. |
| **1C_Decision Tree** | Start Groups Mode set to Bagging and Index count set to 20. |
| **1D_Decision Tree** | Start Groups Mode set to Bagging and Index count set to 30. |
| *Please note all Decision Trees were connected to an End Group node and all other settings kept the default settings (percent is 10% and standard Decision Tree). | |
| **Bagging Optimization Model Criteria for Decision Trees 2A to 2F (Model 2A)** | |
| **Pre-Bagging Models** | **Model Modifications** |
| **2A_Decision Tree** | Start Groups Mode set to Bagging and percentage count set to 7. |
| **2B_Decision Tree** | Start Groups Mode set to Bagging and percentage count set to 8. |
| **2C_Decision Tree** | Start Groups Mode set to Bagging and percentage count set to 9. |
| **2D_Decision Tree** | Start Groups Mode set to Bagging and percentage count set to 10. |
| **2E_Decision Tree** | Start Groups Mode set to Bagging and percentage count set to 11. |
| **2F_Decision Tree** | Start Groups Mode set to Bagging and percentage count set to 12. |
| *Please note all Decision Trees were connected to an End Group node and all other settings kept the default settings (Index count set to 10 and standard Decision Tree). | |
| **Boosting Optimization Model Criteria for Decision Trees 3A to 3F (Model 3A)** | |
| **Pre-Boosting Models** | **Model Modifications** |
| **3A_Decision Tree** | Start Groups Mode set to Boosting and Index count set to 3. |
| **3B_Decision Tree** | Start Groups Mode set to Boosting and Index count set to 5. |
| **3C_Decision Tree** | Start Groups Mode set to Boosting and Index count set to 10. |
| **3D_Decision Tree** | Start Groups Mode set to Boosting and Index count set to 15. |
| **3E_Decision Tree** | Start Groups Mode set to Boosting and Index count set to 20. |
| **3F_Decision Tree** | Start Groups Mode set to Boosting and Index count set to 30 |
| *Please note all Decision Trees were connected to an End Group node and all other settings kept the default settings. | |
| **Gradient Boosting Optimization Models 4A to 4D (Model 4A).** | |
| **Pre-Gradient Boosting Models** | **Model Modifications** |
| **4A_Grad. Boosting** | Gradient Boosting Node Series Options N Iterations set to 100 |
| **4B_Grad. Boosting** | Gradient Boosting Node Series Options Shrinkage set to 0.09 |
| **4C_Grad. Boosting** | Gradient Boosting Node Series Options Shrinkage set to 0.08 |
| **4D_Grad. Boosting** | Gradient Boosting Node Series Options Shrinkage set to 0.07 |
| *Please note all Gradient Boosting Nodes were connected to an Control Point node and all other settings kept the default settings (Only the N Iterations and Shrinkage values were adjusted in these models with N Iterations set to 100). | |

**Figure 4A: HP Optimization Diagram**




**Please note that SAS Code for SAS NODES 2A and 3A can be found in Figure 3BB.**

**Figure 4AA: HP Forest Selection of Best Number of Trees**

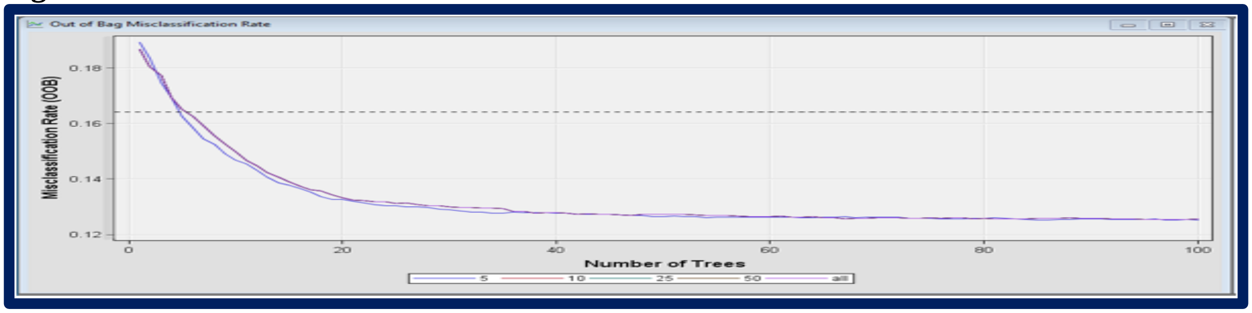



**Figure 5A: HP Forest Selection of the best Number of Variables**

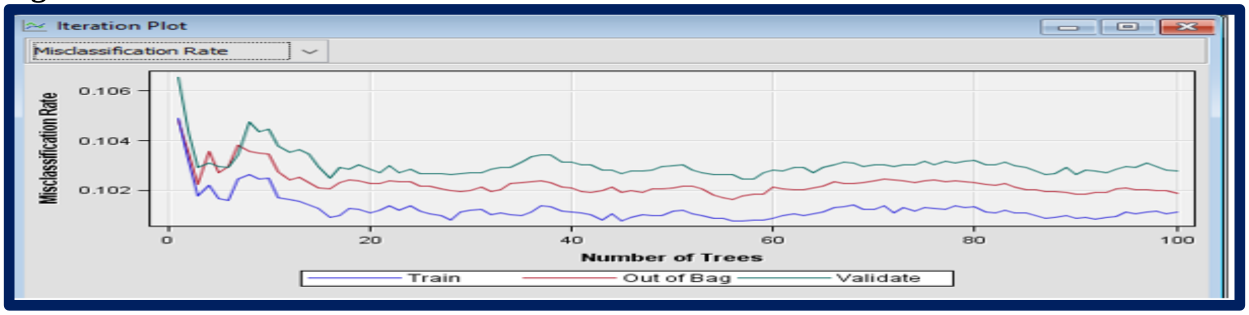



**Figure 6A: HP Forest Selection of the Best Number of Leaves**

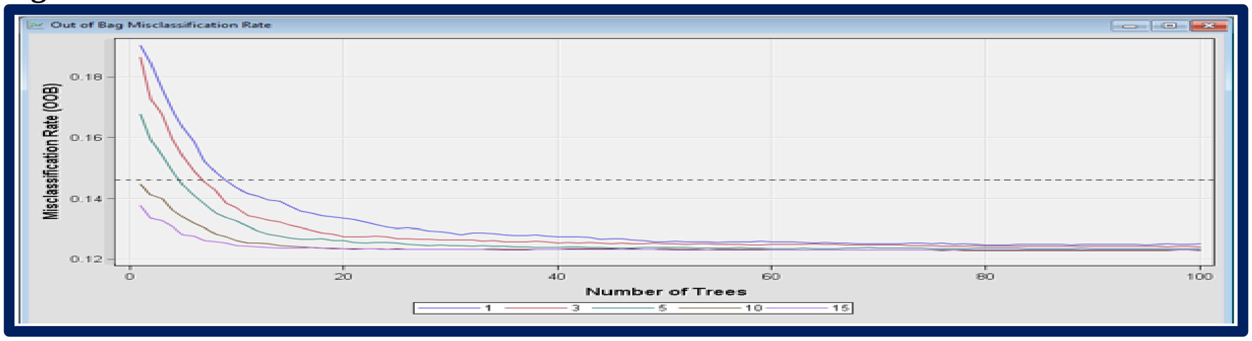


**Figure 7A: Confusion Matrix Results for the Bagging, Boosting, Gradient and Forest Optimization Results.**

| Bagging Optimization Model Decision Trees 1A to 1E From Figure AA (Model 1) | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Train ROC Index | Train Gini Coeff. | Train Misclassification |
|---|---|---|---|---|---|---|---|---|---|
| 1A_End (5,10) Groups Bagging Model TRAIN | 6856 | 63682 | 325 | 2120 | 0.0453 | 0.9949 | 0.707 | 0.414 | 0.0984 |
| 1B_End (10,10) Groups Bagging Model TRAIN | 6856 | 63682 | 325 | 2120 | 0.0453 | 0.9949 | 0.707 | 0.414 | 0.0984 |
| 1C_End (20,10) Groups Bagging Model TRAIN | 6856 | 63682 | 325 | 2120 | 0.0453 | 0.9949 | 0.707 | 0.414 | 0.0984 |
| 1D_End (30,10) Groups Bagging Model TRAIN | 6856 | 63682 | 325 | 2120 | 0.0453 | 0.9949 | 0.707 | 0.414 | 0.0984 |
| | | | | | | | | | |
| Bagging Optimization Model Decision Trees 2A to 2E From Figure AA (Model 2) | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Train ROC Index | Train Gini Coeff. | Train Misclassification |
| 2A_End (10,7) Groups Bagging Model TRAIN | 6976 | 63600 | 407 | 2000 | 0.0551 | 0.9936 | 0.740 | 0.479 | 0.1012 |
| 2B_End (10,8) Groups Bagging Model TRAIN | 6974 | 63568 | 439 | 2002 | 0.0592 | 0.9931 | 0.743 | 0.486 | 0.1016 |
| 2C_End (10,9) Groups Bagging Model TRAIN | 6979 | 63622 | 385 | 1997 | 0.0523 | 0.9940 | 0.746 | 0.492 | 0.1009 |
| 2D_End (10,10) Groups Bagging Model TRAIN | 6830 | 63451 | 556 | 2146 | 0.0753 | 0.9913 | 0.739 | 0.478 | 0.1006 |
| 2E_End (10,11) Groups Bagging Model TRAIN | 6859 | 63455 | 552 | 2117 | 0.0745 | 0.9914 | 0.742 | 0.484 | 0.1012 |
| 2F_End (10,12) Groups Bagging Model TRAIN | 6868 | 63535 | 472 | 2108 | 0.0643 | 0.9926 | 0.740 | 0.480 | 0.1015 |
| | | | | | | | | | |
| Bagging Optimization Model Decision Trees 3A to 3F From Figure AA (Model 3) | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Train ROC Index | Train Gini Coeff. | Train Misclassification |
| 3A_End (3) Groups Boosting Model TRAIN | 1431 | 24783 | 39224 | 7545 | 0.9648 | 0.3872 | 0.781 | 0.561 | 0.5571 |
| 3B_End (5) Groups Boosting Model TRAIN | 7058 | 51080 | 12927 | 1918 | 0.6468 | 0.7980 | 0.775 | 0.550 | 0.2738 |
| 3C_End (10) Groups Boosting Model TRAIN | 6452 | 57065 | 6942 | 2524 | 0.5183 | 0.8915 | 0.811 | 0.621 | 0.1835 |
| 3D_End (15) Groups Boosting Model TRAIN | 4582 | 38983 | 25024 | 4394 | 0.8452 | 0.6090 | 0.839 | 0.678 | 0.4057 |
| 3E_End (20) Groups Boosting Model TRAIN | 6321 | 60607 | 3400 | 2655 | 0.3498 | 0.9469 | 0.855 | 0.710 | 0.1332 |
| 3F_End (30) Groups Boosting Model TRAIN | 4010 | 24662 | 39345 | 4966 | 0.9075 | 0.3853 | 0.881 | 0.762 | 0.5940 |
| | | | | | | | | | |
| Gradient Boosting Optimization Models 4A to 4D from Figure AA (Model 4) | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Train ROC Index | Train Gini Coeff. | Train Misclassification |
| 4A_Gradient Boosting (100) TRAIN | 8542 | 63954 | 53 | 434 | 0.0062 | 0.9992 | 0.762 | 0.525 | 0.1178 |
| 4B_Gradient Boosting (100, 0.09) TRAIN | 8615 | 63965 | 42 | 361 | 0.0049 | 0.9993 | 0.762 | 0.525 | 0.1186 |
| 4C_Gradient Boosting (100, 0.08) TRAIN | 8754 | 63990 | 17 | 222 | 0.0019 | 0.9997 | 0.759 | 0.519 | 0.1202 |
| 4D_Gradient Boosting (100, 0.07) TRAIN | 8976 | 64007 | 0 | 0 | 0.0000 | 1.0000 | 0.629 | 0.257 | 0.1230 |
| | | | | | | | | | |
| Random Forest Model Comparison (1A to 5A) from Figure AA | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | ROC Index | TRAINING Misclassif. | VALID Misclassification |
| Forest 4A_HP Forest DEF_TRAIN | 2718 | 25368 | 234 | 871 | 0.0793 | 0.9909 | 0.788 | 0.101 | . |
| Forest 4A_HP Forest DEF_VALIDATE | 2042 | 18994 | 208 | 651 | 0.0924 | 0.9892 | 0.746 | . | 0.1011 |
| Forest5A_HP Forest OPTIMIZED TRAIN | 2735 | 25466 | 136 | 854 | 0.0474 | 0.9947 | 0.857 | 0.098 | . |
| Forest5A_HP Forest OPTIMIZED VALIDATE | 2071 | 19072 | 130 | 622 | 0.0591 | 0.9932 | 0.748 | . | 0.0984 |

**Figure 8A: Fit Statistic Results for the top four Models and Models A to C.**

| Top Four Models from Final Model Comparison Node | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Precision | Train/Vald. Misclassification |
|---|---|---|---|---|---|---|---|---|
| #1 IS 5EE_SEQ Final Decision Tree TRAIN | 16 | 63995 | 12 | 8960 | 0.429 | 1.000 | 0.999 | 0.0004 |
| #2 HPDMForest5 6FF_Final HP Forest TRAIN | 19 | 63946 | 61 | 8957 | 0.763 | 0.999 | 0.993 | 0.0011 |
| #3 HPNNA3 3G_30 HU HP Neural TRAIN | 5576 | 63179 | 828 | 3400 | 0.129 | 0.987 | 0.804 | 0.0878 |
| #4 nsmbl7 2B_Ensemble TRAIN | 8973 | 64007 | 0 | 3 | 0.000 | 1.000 | 1.000 | 0.1230 |
| | | | | | | | | |
| **Models 1: Default Fit Statistics for Models 1A to 4A (Control)** | **False Negative** | **True Negative** | **False Positive** | **True Positive** | **Sensitivity** | **Specificity** | **Precision** | **Train/Vald Misclassification** |
| 1A_Decision Tree | 6856 | 63682 | 325 | 2120 | 0.0453 | 0.9949 | 0.867 | 0.0984 |
| Reg 2A_Step_Trans Regression TRAIN | 6282 | 44804 | 0 | 0 | ` | 1.000 | . | 0.1230 |
| Reg 2A_Step_Trans Regression VALIDATE | 1795 | 12801 | 0 | 0 | 0.000 | 1.000 | . | 0.1230 |
| HPSVM 3A_HP Lin (INT)_T SVM TRAIN | 6096 | 44757 | 47 | 186 | 0.008 | 0.999 | 0.798 | 0.1203 |
| HPSVM 3A_HP Lin (INT)_T SVM VALIDATE | 1772 | 12748 | 53 | 23 | 0.029 | 0.996 | 0.303 | 0.1250 |
| Neural5 4A_3HU Neural Net TRAIN | 6223 | 44743 | 61 | 59 | 0.010 | 0.999 | 0.492 | 0.1230 |
| Neural5 4A_3HU Neural Net VALIDATE | 1780 | 12792 | 9 | 15 | 0.005 | 0.999 | 0.625 | 0.1226 |
| ENSEMBLE 1A TRAIN | 6242 | 44804 | 0 | 40 | 0.000 | 1.000 | 1.000 | 0.1222 |
| ENSEMBLE 1A VALIDATE | 1793 | 12801 | 0 | 2 | 0.000 | 1.000 | 1.000 | 0.1223 |
| | | | | | | | | |
| **Model 2: Bagging, Boosting, Gradient and Neural Network Models to 5BB_25HU.** | **False Negative** | **True Negative** | **False Positive** | **True Positive** | **Sensitivity** | **Specificity** | **Precision** | **Train/Vald Misclassification** |
| 1B_End Bagging Model | 6868 | 63535 | 472 | 2108 | 0.064 | 0.993 | 0.817 | 0.1000 |
| 2B_End_Boosting Model | 6321 | 60607 | 3400 | 2655 | 0.350 | 0.947 | 0.438 | 0.1332 |
| 3B_Step Bagging End Groups (TRAIN ONLY) | 6282 | 44804 | 0 | 0 | 0.000 | 1.000 | #DIV/0! | 0.1230 |
| Reg2 4B_Step Log Regression | 6282 | 44804 | 0 | 0 | 0.000 | 1.000 | . | 0.1230 |
| Reg2 4B_Step Log Regression VALIDATE | 1795 | 12801 | 0 | 0 | 0.000 | 1.000 | . | 0.1230 |
| Neural6 5B_3HU Neural Network | 6282 | 44804 | 0 | 0 | 0.000 | 1.000 | . | 0.1230 |
| Neural6 5B_3HU Neural Network VALIDATE | 1795 | 12801 | 0 | 0 | 0.000 | 1.000 | . | 0.1230 |
| Boost 5B_Gradient Boosting | 8542 | 63954 | 53 | 434 | 0.006 | 0.999 | 0.891 | 0.1178 |
| HPDMForest 6B_HP Forest | 6840 | 63768 | 239 | 2136 | 0.034 | 0.996 | 0.899 | 0.0970 |
| Ensemble 2A VALIDATION | 8973 | 64007 | 0 | 3 | 0.000 | 1.000 | 1.000 | 0.1200 |
| Ensemble 2A VALIDATION | 1794 | 12801 | 0 | 1 | 0.000 | 1.000 | 1.000 | 0.1200 |
| | | | | | | | | |
| **Model 3: Neural Net Models with Ensembels TRAINING RESULTS ONLY** | **False Negative** | **True Negative** | **False Positive** | **True Positive** | **Sensitivity** | **Specificity** | **Precision** | **Train Misclassification** |
| Neural 1C_NN (3HU)_(DEF) TRAIN | 6282 | 44804 | 0 | 0 | 0.000 | 1.000 | . | 0.1230 |
| Neural2 2C_NN (10 HU) TRAIN | 6276 | 44795 | 9 | 6 | 0.001 | 1.000 | 0.400 | 0.1039 |
| Neural3 3C_NN (HU 30) TRAIN | 6281 | 44803 | 1 | 1 | 0.000 | 1.000 | 0.500 | 0.1230 |
| Neural4 4C_NN (HU 50) TRAIN | 6273 | 44793 | 11 | 9 | 0.002 | 1.000 | 0.450 | 0.1230 |
| Ensmble 1C_Ensemble All NN TRAIN | 6282 | 44804 | 0 | 0 | 0.000 | 1.000 | . | 0.1230 |
| Ensmble 2C_Ensemble NN (10-30 HU) TRAIN | 6280 | 44799 | 5 | 2 | 0.001 | 1.000 | 0.286 | 0.1230 |
| Ensmble 3C_Ensemble NN (10-50 HU) TRAIN | 6281 | 44799 | 5 | 1 | 0.001 | 1.000 | 0.167 | 0.1230 |

**Figure 9A: Fits statistics for Models D to G.**

| Model 4: Ensemble Rotatinon Forest | False Negative | True Negative | False Positive | True Positive | Sensitivity | Specificity | Precision | Train Misclassification |
|---|---|---|---|---|---|---|---|---|
| 4DD Ensemble Rotation Forest | 8976 | 64007 | 0 | 0 | 0.000 | 1.000 | . | 0.1230 |
| | | | | | | | | |
| **Model 5: Sequential Bootstrap Aggregated Algorithm** | **False Negative** | **True Negative** | **False Positive** | **True Positive** | **Sensitivity** | **Specificity** | **Precision** | **Train Misclassification** |
| 5EE_SEQ Final Decision Tree Boostrap | 16 | 63995 | 12 | 8960 | 0.429 | 1.000 | 0.999 | 0.0004 |
| | | | | | | | | |
| **Model 6: SEQ Final HP Forest** | **False Negative** | **True Negative** | **False Positive** | **True Positive** | **Sensitivity** | **Specificity** | **Precision** | **Train Misclassification** |
| 6FF_Final HP Forest SEQ | 19 | 63946 | 61 | 8957 | 0.763 | 0.999 | 0.993 | 0.0001 |
| | | | | | | | | |
| **Model 7: Stacked Model** | **False Negative** | **True Negative** | **False Positive** | **True Positive** | **Sensitivity** | **Specificity** | **Precision** | **Train Misclassification** |
| 1G 10 HU HP Neural Net | 6105 | 62973 | 1034 | 2871 | 0.145 | 0.984 | 0.735 | 0.098 |
| 2G 15 HU HP Neural Net | 5362 | 62709 | 1298 | 3614 | 0.195 | 0.980 | 0.736 | 0.091 |
| 3G 30 HU HP Neural Net | 5576 | 63179 | 828 | 3400 | 0.129 | 0.987 | 0.804 | 0.088 |
| Gradient Boosting | 8976 | 64007 | 0 | 0 | 0.000 | 1.000 | #DIV/0! | 0.123 |
| 1G Entropy HP Tree | 8645 | 63854 | 153 | 331 | 0.017 | 0.998 | 0.684 | 0.121 |
| 1G CHI Square HP Tree | 8951 | 64003 | 4 | 25 | 0.000 | 1.000 | 0.862 | 0.123 |
| 1G HP Forest | 6840 | 63768 | 239 | 2136 | 0.034 | 0.996 | 0.899 | 0.097 |
| 1G HP SVM | 8815 | 63940 | 67 | 161 | 0.008 | 0.999 | 0.706 | 0.122 |

**Figure 10A: Fit Statistics for the top Four Predictive Models.**

Fit Statistics

| Predecessor Node | Model Node | Model Description | Target Variable | Train: Misclassification Rate | Selection Criterion: Valid: Misclassification Rate | Test: Misclassification Rate | Valid: Roc Index | Test: Roc Index |
|---|---|---|---|---|---|---|---|---|
| Tree5 | Tree5 | 5EE_SEQ Final Decision Tree | IsBadBuy | .0003837 | . | . | . | . |
| HPDMFore... | HPDMFore... | 6FF_Final HP Forest | IsBadBuy | 0.001096 | . | . | . | . |
| MdlComp | HPNNA3 | 3G_30 HU HP Neural | IsBadBuy | 0.087746 | . | . | . | . |
| MdlComp5 | Ensmbl7 | 2B_Ensemble | IsBadBuy | 0.122946 | 0.12291 | 0.123134 | 0.801 | 0.82 |
| Ensmbl6 | Ensmbl6 | 4DD_Ensemble Rotation Forest | IsBadBuy | 0.122988 | . | . | . | . |
| MdlComp13 | Neural5 | 4A_3HU Neural Net | IsBadBuy | 0.123008 | 0.122568 | 0.122312 | 0.677 | 0.691 |
| MdlComp8 | Neural4 | 4C_NN (HU 50) | IsBadBuy | 0.123008 | 0.122842 | 0.123134 | 0.669 | 0.68 |

**Figure 11A: ROC Chart for the top four Predictive Models.**

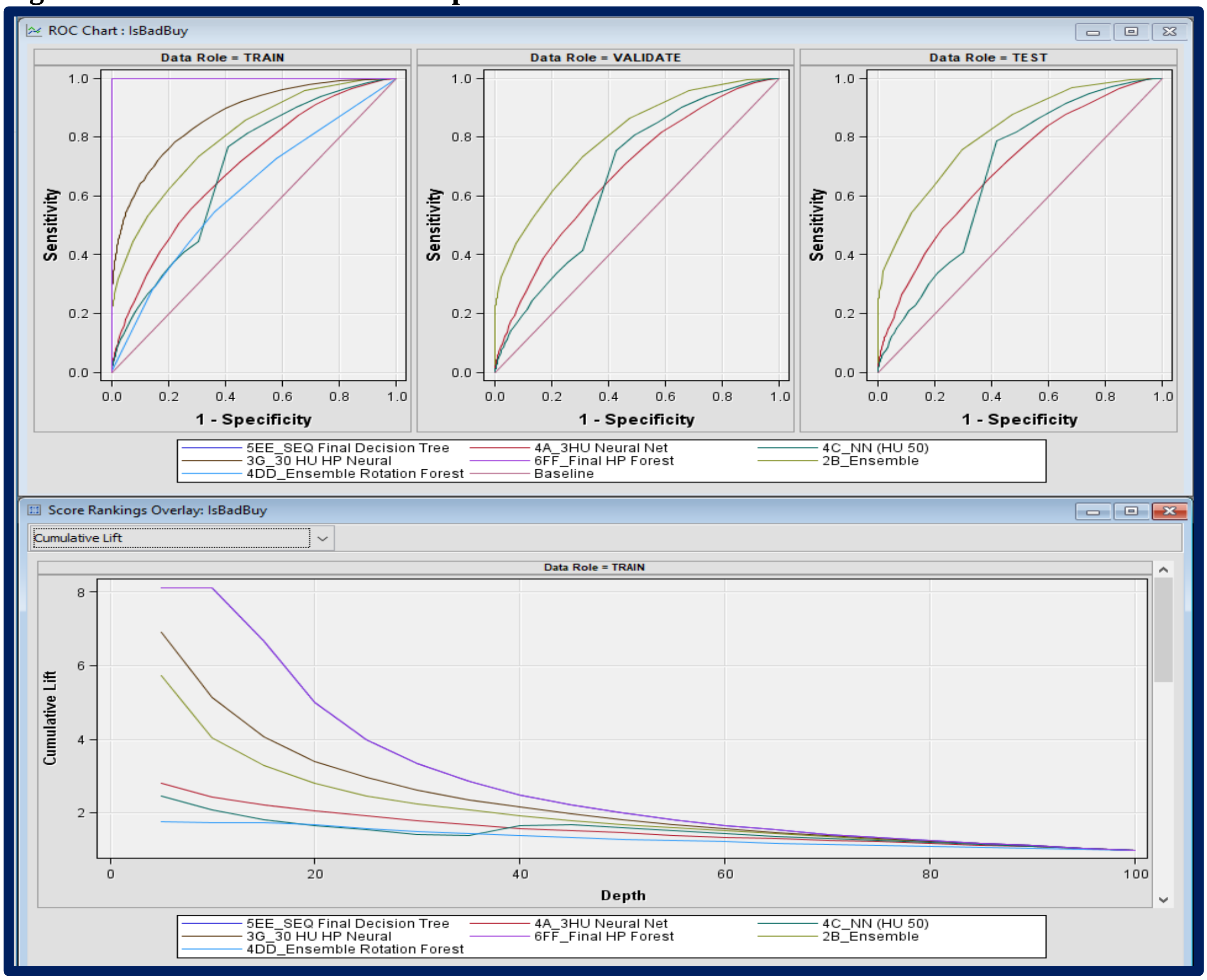


**Figure 12A: Optimization Diagram for Bagging, Boosting and Gradient Boosting.**

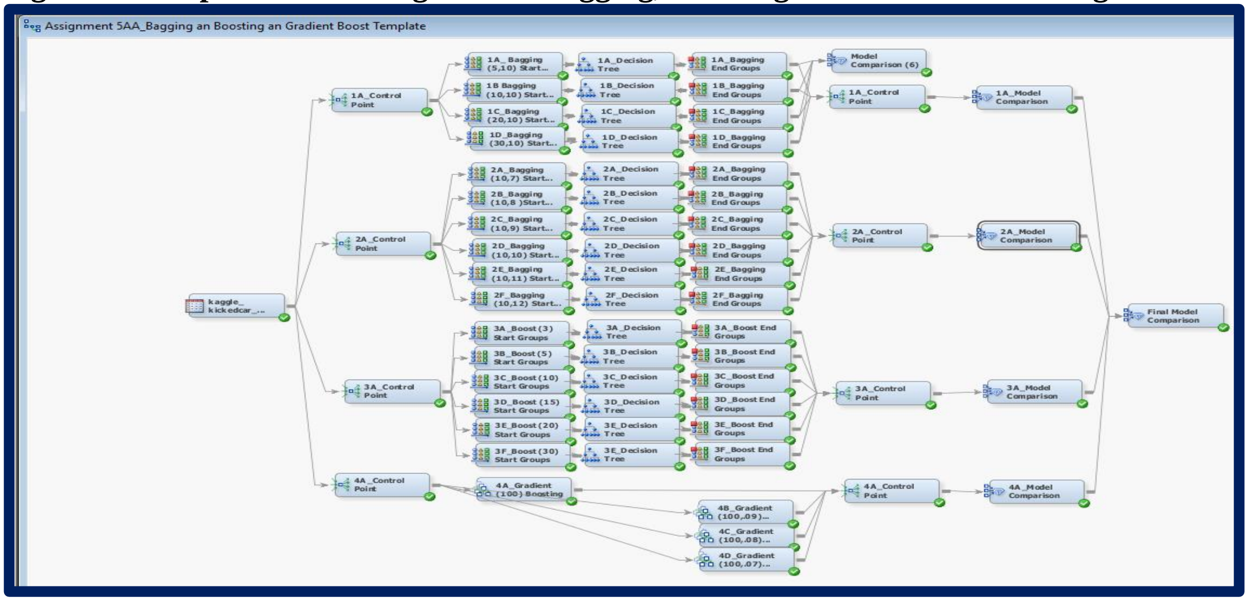



**FIGURE 13A: Ensemble Model Final Diagram.**

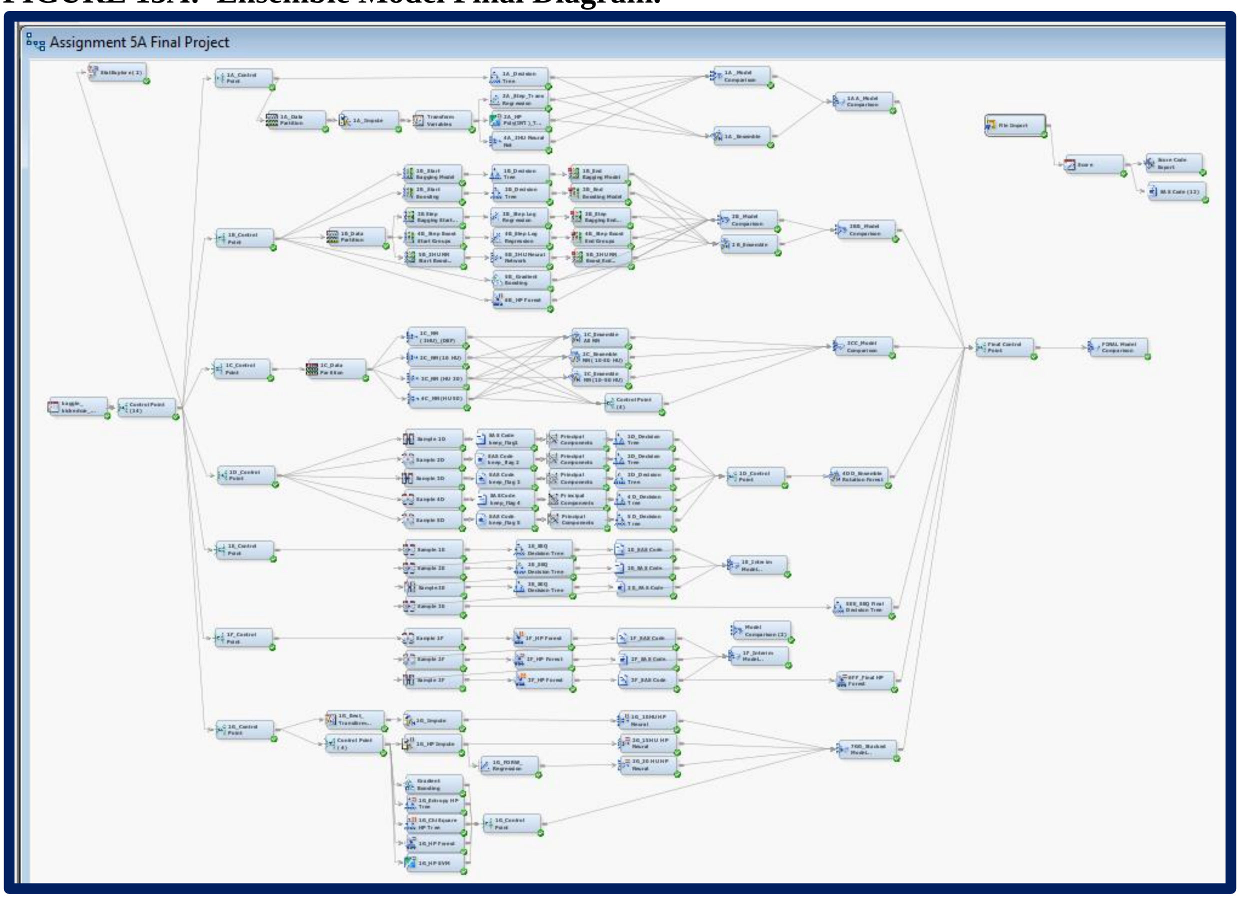
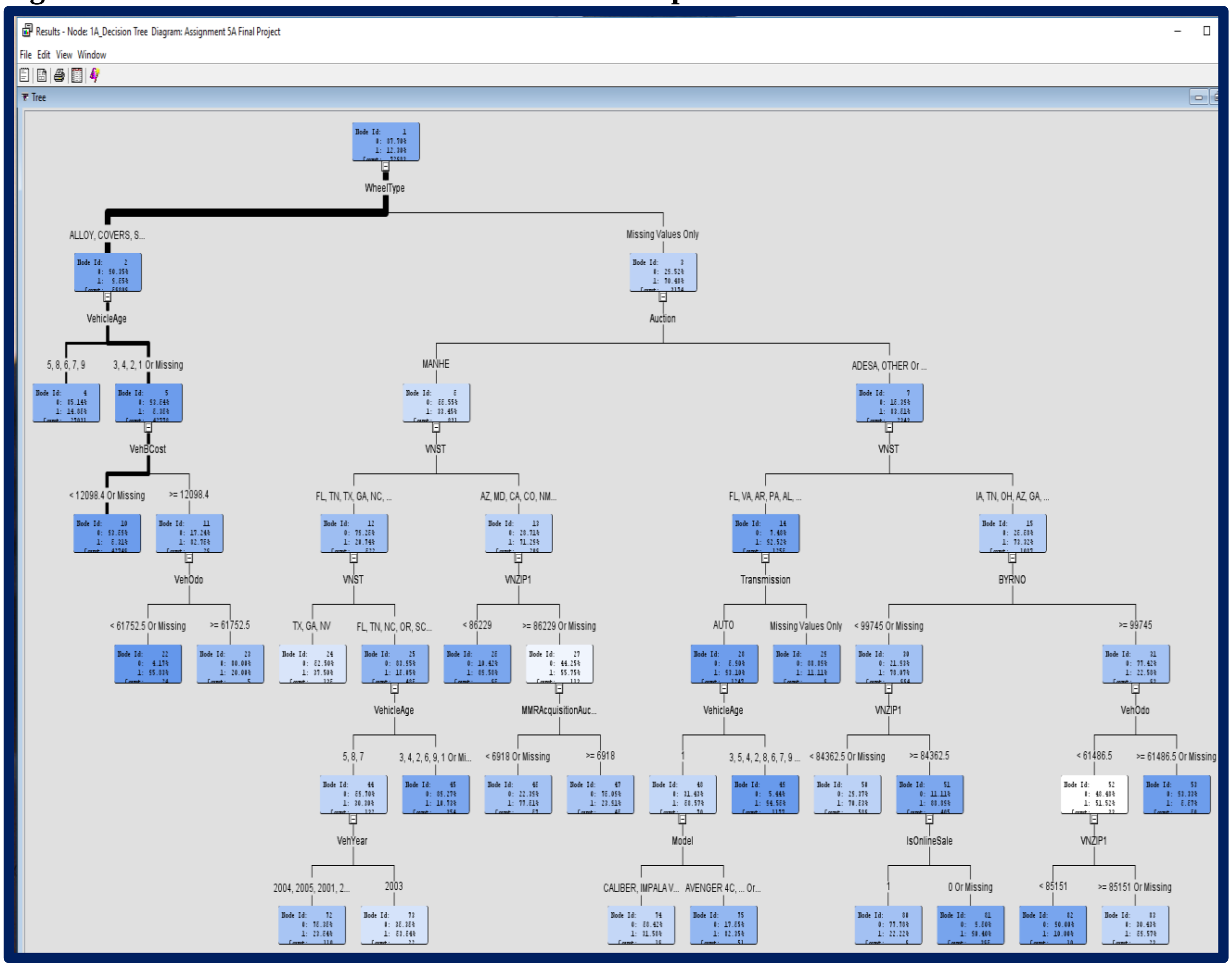


**Figure 14A: Decision Tree from Model 1A in Group A.**



# APPENDIX B

**Figure 1BB: SAS Code for Rotation Forest**

| SAS Code for Rotation Forest for 1 to 5 Codes (Replace the keep_flag with the correct number. Code came from Maldanodo et. al. (2014) |
|---|
| %macro keepsomevars(randomseed=12345,n_groups=5,keep_flag=1); data _null_; set &EM_IMPORT_DATA_CMETA(where=(role="INPUT")) end=eof; call symput ('my_input'!!strip(put(_N_, BEST.)),strip(NAME)); if eof then call symput('total_inputs', strip(put(_N_, BEST.))); run; %do i = 1 %to &total_inputs; %let x =%sysevalf(%sysfunc(ceil(%sysfunc(ranuni(&randomseed))*&n_groups))); %put variable myinput&i is &&my_input&i with random value &x; %if "&x" ne "&keep_flag" %then %do; %put Variable &&my_input&i will be rejected.; %EM_METACHANGE(name=&&my_input&i, role=REJECTED); %end; %end; %mend keepsomevars; %keepsomevars(randomseed=12345,n_groups=5,keep_flag=1) |

**Figure 2BB: SAS Code for HP Forest SAS CODES 2A.**

SAS Code for Number of Variables for HP Forest (Wujek (2015))

```
%macro hpforestStudy (nVarsList=10,maxTrees=200);

  %let nTries = %sysfunc(countw(&nVarsList.));
   /* Loop over all specified number of variables to try */
  %do i = 1 %to &nTries.;
   %let thisTry = %sysfunc(scan(&nVarsList.,&i));

   /* Run HP Forest for this number of variables */
   proc hpforest data=&em_import_data maxtrees=&maxTrees. vars_to_try=&thisTry.;
    input %EM_INTERVAL_INPUT /level=interval;
    target %EM_TARGET / level=binary;
    ods output fitstatistics=fitstats_vars&thisTry. ;
   run;
   /* Add the value of varsToTry for these fit stats */
   data fitstats_vars&thisTry.;
    length varsToTry $ 8;
    set fitstats_vars&thisTry.;
    varsToTry = "&thisTry.";
   run;

   /* Append to the single cumulative fit statistics table */
   proc append base=fitStats data=fitstats_vars&thisTry.;
   run;
  %end;
%mend hpforestStudy;

%hpforestStudy(nVarsList=5 10 25 50 all,maxTrees=100);
/* Register the data set for use in the em_report reporting macro */
%em_register(type=Data,key=fitStats);
data &em_user_fitStats;
   set fitStats;
run;
%em_report(viewType=data,key=fitStats,autodisplay=y);
%em_report(viewType=lineplot,key=fitStats,x=nTrees,y=miscOOB,group=varsToTry,d
escription=Out of Bag Misclassification Rate,autodisplay=y);
```

**Figure 3BB: SAS Code for HP Forest SAS CODES 3A and the Ensemble Rotation Forest.**

| **SAS Code for Number of Leaves for HP Forest (Wujek (2015))** |
|---|
| <pre>  %macro hpforestStudy (leafsizeList=5,maxTrees=200);<br><br> %let nTries = %sysfunc(countw(&leafsizeList.));<br> /* Loop over all specified number of variables to try */<br> %do i = 1 %to &nTries.;<br>  %let thisTry = %sysfunc(scan(&leafsizeList.,&i));<br><br>   /* Run HP Forest for this number of variables */<br>   proc hpforest data=&em_import_data maxtrees=&maxTrees. leafsize=&thisTry.;<br>      input %EM_INTERVAL_INPUT /level=interval;<br>      target %EM_TARGET / level=binary;<br>      ods output fitstatistics=fitstats_vars&thisTry. ;<br>   run;<br><br>   /* Add the value of varsToTry for these fit stats */<br>   data fitstats_vars&thisTry.;<br>      length leafsize $ 8;<br>      set fitstats_vars&thisTry.;<br>      leafsize = "&thisTry.";<br>   run;<br>   /* Append to the single cumulative fit statistics table */<br>   proc append base=fitStats data=fitstats_vars&thisTry.;<br>   run;<br>  %end;<br>%mend hpforestStudy;<br>%hpforestStudy(leafsizeList=1 3 5 10 15,maxTrees=100);<br>/* Register the data set for use in the em_report reporting macro */<br>%em_register(type=Data,key=fitStats);<br>data &em_user_fitStats;<br>   set fitStats;<br>run;<br>%em_report(viewType=data,key=fitStats, SAS Code for Number of Trees for HP Forest<br>(Maldonado et al. (2014))autodisplay=y);<br>%em_report(viewType=lineplot,key=fitStats,x=nTrees,y=miscOOB,group=leafsize,descriptio<br>n=Out of Bag Misclassification Rate,autodisplay=y);</pre> |
| **SAS Code for Sequential Boosted Bootstrap Aggregating Algorithms for all the Code Weights and Targets Node (Maldonado et al. (2014))** |
| <pre>data &EM_EXPORT_TRAIN;<br> set &EM_IMPORT_DATA;<br>if _freq_ = . then _freq_=1;<br>if f_%EM_TARGET ne i_%EM_TARGET then _freq_=_freq_*32<br>;<br>run</pre> |