Assignment 2

Ensemble Modeling 1 on Universal Bank Dataset:

Predicting Personal Loan Clientele

Theodore Fitch

Department of Data Analytics, University of Maryland Global Campus

DATA 640: Predictive Modeling

Dr. Steven Knode

February 13th, Spring 2024

Introduction:

The purpose of this analysis was to analyze the Universal Bank dataset and predict whether customers will accept the bank's personal loan using Ensemble Models (EMs). EMs are classification prediction models that "average" the results of multiple "weaker" models in order to generate more accurate results (Srivastava, 2022). The primary methods used in this analysis were Bagging (decision tree modeling using sampling with replacement), Boosting (iteratively training weaker models and penalizing incorrect classifications), Gradient Boosting (the Boosting technique but optimizes the cost function), and Random Forest (generating decision trees using a random subset of variable inputs)(SAS Software, 2017; Srivastava, 2022). Like Support Vector Machines (SVMs), EMs can be difficult to interpret and prone to overfitting, and thus also difficult to implement in production environments (due to computational demands or black box requirements in regulated industries)(CFPB, 2022). But, EMs also are incredibly accurate and reduce impact of having outliers in the data (SAS Software, 2017). Thus, EM models were developed and compared to the previous analysis of SVM modeling on this dataset in order to contrast their approaches and accuracy (Fitch, 2024). All background of this dataset was discussed in Fitch (2024); to summarize, it is necessary for banks to be able to predict which of their customers would be likely to accept a personal loan. The SAS SEMMA method was still used as a general approach (Shafique and Qaiser, 2014).

Exploratory Data Analysis and Preprocessing:

This dataset was provided by the classroom for DATA 640 and was entitled "UniversalBank data.csv". It was uploaded to SAS Enterprise Miner as a .sas7bdat filetype (Figure *2*). It contained 5,000 rows with 14 variables (Table 2, Figure 1). There were 7 interval variables (columns A-M, except for the nominal/ordinal attributes), 1 nominal (education), 1 ordinal (family), and 5 binary variables: (columns J-N). The first column was effectively ignored as it was an index. The target variable column J, “Personal Loan” was heavily imbalanced (480 entries of 5,000; rate of 9.6%)(Figure 1). Other variables can be seen in Table 2. There are no missing entries in the whole dataset (Figure 3, Figure 4). All ZIP codes belonged to California (Figure 5). One value (row 386) in column E (ZIP Code) was entered incorrectly as it only contained 4 characters (Table 3). There were 52 negative values found for column C (Experience) as well as 60 values of “0”. No significant outliers were observed for the numeric variables (age, income, mortgage, etc.) and thus skew was not expected to be an issue for this dataset. When creating graphs to explore the Chi-Square and Worth of each variable, Income, CCAvg, CD_Account, Mortgage, Education, and Family were found to be the 6 most important variables with Chi-Squared values ranging from 1,411-30 (Figure 6, Figure 8). The next 6 variables combined Chi-Squared values were <35 showing the significance of first variables.

No missing values existed in this dataset thus no imputations were necessary. The ZIP codes were transformed from their original 5-digit state to only the first 2 digits. This was thought helpful since the first 2 digits of a ZIP code indicate a general location in a state. In this case, all ZIP codes belonged to California locations: 90, 91, and 94 being city dense locations and 92, 93, 95, and 96 being rural (Polly, 2014). While ZIP codes can have significant demographic variation (especially within cities), it was thought that rural locations and city-dense locations would be the best way to categorize the ZIP codes. This was performed using a Transform Variables node. Then, since 1 value had only 4 digits, a correction was needed using a Replacement node. The first Transform Variables node was also used to take the absolute value of all negative values in the Experience variable (as these were deemed to be misinputs). No feature engineering was deemed helpful for this dataset. A Drop node was then used to clear unnecessary variables from the dataset (several were created during the usage of the Transform Variables node, and the ID was also dropped as this was simply an index (Figure 9). A variable correlation matrix was used to determine that age and experience were strongly correlated (Figure 7); however when analyzed further by contrasting model comparison results dropping age, and then dropping experience, there was no significant change in final results as compared to keeping both variables in the models. The last new feature to this analysis was the data was sampled to have an equal distribution of positive/negative values of the target variable.

Models and Methods:

The cleaned/processed dataset was run using the 4 model types with 3 iterations (Table 4). In addition, the 3 best performing models from the SVM analysis were retained for contrast (Fitch, 2024). The 3 clusters of 4 models were iterated in order to find the most accurate model using a wide variety of methods. First, the data was partitioned using a 70:30 split (55:45 was also tested but yielded suboptimal results) and all default settings on the methods were used (Models 4-7). Next, the data was partitioned using an 80:20 split and also using all default settings (Models 8-11). Since the second cluster of models yielded better results than the first, an 80:20 data partition was used for the third cluster. The gradient boosting model was increased from 10 to 100 iterations. The random forest model was increased from 10 to 1,000 iterations. The iterations were increased on these models with the anticipation this would increase accuracy by forcing the models to iterate their testing much further. Since EMs work by “voting” the most accurate answer across the models, it was thought that increasing the number of voters would also increase accuracy. Model 14 had the rule to create subtrees changed from Assessment (using the fit statistics of the tree to split) to N (choosing the tree which had the most rules). This was thought it may make the model more accurate to force the EM to choose larger, more detailed rulesets. Model 15 was given a maximum depth of 10 instead of 5 with the expectation iterative decision trees that were allowed a total of 10 rules would be more accurate; in addition, the minimum categorical size was set to 10 with the anticipation that this would force the model to not focus on the outliers but focus the general trends.

In order to account for the target variable being heavily imbalanced, several approaches were explored to decrease likelihood of false negatives (Cao et al., 2013). Firstly, the SVMs had a cost of c = 1 applied to every model. This penalizes them for making incorrect classifications which forces the models to make fewer false negatives. The second approach taken was exploring sampling the dataset. This technique was used to balance the imbalanced dataset by subtracting negative values until there are equal entries between positive & negative instances in the target variable. When exploring this technique, the model comparison results showed better results in accuracy and sensitivity (and other measures explored)(Figure 10, Figure 11, Figure *2*). Thus, this approach was used as the primary approach to address the imbalance. Lastly, the cutoff criterion was finally used to optimize the models. This node was not applied to all the SVMs (since they should not have been changed from their state in the previous analysis); but, it was applied to all relevant EMs.

Results and Model Evaluation:

Index	Model Type	Model Description	Data Role	FN	TN	FP	TP	Sensitivity	Specificity	Precision	F1 Score	Misclassification Rate	Accuracy
1	SVM	1)_SVM_IP_polynomial(2)	TRAIN	13	261	3	251	0.95	0.99	0.99	0.97	0.03	97.0%
2	SVM	1)_SVM_IP_polynomial(2)	VALIDATE	13	203	13	203	0.94	0.94	0.94	0.94	0.06	94.0%
3	SVM	2)_SVM_AS_polynomial(5)	TRAIN	13	261	3	251	0.95	0.99	0.99	0.97	0.03	97.0%
4	SVM	2)_SVM_AS_polynomial(5)	VALIDATE	13	203	13	203	0.94	0.94	0.94	0.94	0.06	94.0%
5	SVM	3)_SVM_AS_polynomial(10)	TRAIN	11	379	4	373	0.97	0.99	0.99	0.98	0.02	98.0%
6	SVM	3)_SVM_AS_polynomial(10)	VALIDATE	3	91	6	93	0.97	0.94	0.94	0.95	0.05	95.3%
7	Gradient Boosting	4)_Gradient_Boosting	TRAIN	8	325	10	328	0.98	0.97	0.97	0.97	0.03	97.3%
8	Gradient Boosting	4)_Gradient_Boosting	VALIDATE	2	135	10	142	0.99	0.93	0.93	0.96	0.04	95.8%
9	Random Forest	5)_HP_Forest	TRAIN	1	335	0	335	1.00	1.00	1.00	1.00	0.00	99.9%
10	Random Forest	5)_HP_Forest	VALIDATE	4	136	9	140	0.97	0.94	0.94	0.96	0.04	95.5%
11	Bagging	6)_EG_Bagging	TRAIN	18	313	22	318	0.95	0.93	0.94	0.94	0.06	94.0%
12	Bagging	6)_EG_Bagging	VALIDATE	6	136	9	138	0.96	0.94	0.94	0.95	0.05	94.8%
13	Boosting	7)_EG_Boosting	TRAIN	29	313	22	307	0.91	0.93	0.93	0.92	0.08	92.4%
14	Boosting	7)_EG_Boosting	VALIDATE	20	135	10	124	0.86	0.93	0.93	0.89	0.10	89.6%
15	Gradient Boosting	8)_Gradient_Boosting	TRAIN	6	365	18	378	0.98	0.95	0.95	0.97	0.03	96.9%
16	Gradient Boosting	8)_Gradient_Boosting	VALIDATE	2	90	7	94	0.98	0.93	0.93	0.95	0.05	95.3%
17	Random Forest	9)_HP_Forest	TRAIN	0	383	0	384	1.00	1.00	1.00	1.00	0.00	100.0%
18	Random Forest	9)_HP_Forest	VALIDATE	3	90	7	93	0.97	0.93	0.93	0.95	0.05	94.8%
19	Bagging	10)_EG_Bagging	TRAIN	35	373	10	349	0.91	0.97	0.97	0.94	0.06	94.1%
20	Bagging	10)_EG_Bagging	VALIDATE	6	92	5	90	0.94	0.95	0.95	0.94	0.06	94.3%
21	Boosting	11)_EG_Boosting	TRAIN	5	342	41	379	0.99	0.89	0.90	0.94	0.06	94.0%
22	Boosting	11)_EG_Boosting	VALIDATE	4	86	11	92	0.96	0.89	0.89	0.92	0.08	92.2%
23	Gradient Boosting	12)_Gradient_Boosting	TRAIN	6	365	18	378	0.98	0.95	0.95	0.97	0.03	96.9%
24	Gradient Boosting	12)_Gradient_Boosting	VALIDATE	2	90	7	94	0.98	0.93	0.93	0.95	0.05	95.3%
25	Random Forest	13)_HP_Forest	TRAIN	0	383	0	384	1.00	1.00	1.00	1.00	0.00	100.0%
26	Random Forest	13)_HP_Forest	VALIDATE	3	91	6	93	0.97	0.94	0.94	0.95	0.05	95.3%
27	Bagging	14)_EG_Bagging	TRAIN	192	191	194	190	0.50	0.50	0.49	0.50	0.50	49.9%
28	Bagging	14)_EG_Bagging	VALIDATE	41	45	56	51	0.55	0.45	0.48	0.51	0.50	50.3%
29	Boosting	15)_EG_Boosting	TRAIN	0	384	0	384	1.00	1.00	1.00	1.00	0.00	100.0%
30	Boosting	15)_EG_Boosting	VALIDATE	5	143	2	143	0.97	0.99	0.99	0.98	0.04	96.4%

Table 1. Accuracy measures of the 15 models generated evaluated by sensitivity and F1 show that model 4 is the champion with a sensitivity of 0.99 and F1 of 0.96 on the validation data.

Fifteen models were generated; the selection criteria used to determine the best models were sensitivity (the true positive rate) and F1 score (the harmonic mean of precision and sensitivity). The former was used because the cost of a false negative is high; the income for a bank to identify a positive customer would outweigh the price of marketing to multiple customers. F1 on the other hand shows the balance between increasing the true positive rate and overall accuracy. Only 2 models (7, 14/boosting, bagging respectively) showed a lower sensitivity/F1 than baseline. All other models had a sensitivity >0.94 (average 0.96). Model 14 had poor performance showing the subtree method (N) performed far worse than Assessment (which makes sense since the assessment method chooses trees based upon fit metrics rather than size of the tree). The champion model was chosen based upon best sensitivity of the validation dataset: model 4, gradient boosting method had a sensitivity of 0.99 and F1 of 0.96 (missing only 2 positive customers of 144). Since marketing to customers is a relatively low cost compared to the revenue a new customer generates when choosing a personal loan, this model was chosen to eliminate as many false negatives as possible. The methods used to account for the imbalanced target variable worked efficiently since the sensitivity/F1 of model 4 was 0.85/0.9 in the unsampled data with no cutoff (see Figure 12 for example cutoff chart for model 4)(accuracy measures increased for all models using balanced data). This shows the importance of using methods to increase accuracy of positive predictions (via cutoff, cost, or balanced datasets).

Examining each class of model, gradient boosting models performed the best (all had sensitivity of >0.98). This was followed by random forests, SVMs, boosting, and lastly bagging. All model types performed similarly in terms of overall accuracy with the gradient boosting models performing best in sensitivity and F1. To that end, F1 shows how sensitivity and precision are balanced. In this case, it is necessary to sacrifice some model specificity (the true negative rate) in order to increase sensitivity. Compared to the SVM models built for the previous analysis, the ensemble models performed both better (gradient boosting, random forest) and worse (boosting, bagging). Model 3 previously showed accuracy of 98% and sensitivity of 0.92 (Fitch, 2024) without usage of the sampling technique to account for the imbalanced target variable. With sampling used, accuracy lowered to 95% but sensitivity rose to 97%. While sensitivity was not previously assessed for all models, this increase for this model alone shows 1. the effectiveness of sampling 2. how well the SVMs performed compared to the others. Compared to the unsampled data (Figure 11), the sampled models showed slightly lower accuracy but significantly higher sensitivity which was deemed more important for this business case. The cumulative lift and ROC charts similarly show that all models performed very well (with exception of 14) by showing they were all significantly lifted from using a random model and they all had AUC-ROC of >0.97 respectively (Figure 13, Figure 14, Figure 15). Overfitting is clearly not an issue for these models either as we don't see dramatic drops in model accuracy/sensitivity from the training data to the validation data. Similarly, we don't see dramatic decreases in ROC-AUC from training to validation datasets which would also indicate overfitting (Figure 13). Lastly, it is critical to note that the 80:20 data partition performed better than the 70:30 (but this is primarily because the boosting model (7) performed poorly for the 70:30 partition. The parameters being changed did not affect the gradient boosting and random forest models (8,9,12,13) but made the boosting model better and the bagging much worse.

Conclusion, Limitations, and Improvements:

In conclusion, 15 predictive models were created (3 SVMs, 12 EMs) to predict which bank customers were going to accept the personal loan from the bank. Of those, all models except for 7 and 14 showed a higher sensitivity than randomly guessing which customers would likely choose the loan (baseline = 90%). The champion was a gradient boosting model (model 4) showing a 95.8% accuracy, 0.99 specificity, and 0.96 F1. The gradient boosting model type performed the best, yielding average sensitivity 1% better than the next category (random forest) and 17% better than the worst model type (bagging). Model 14 performed poorly because decision trees should be based upon fit statistics in order to split instead of forcing larger trees to be chosen (bigger doesn’t always mean better). In implementation of predictive models, there are always tradeoffs to be found (between accuracy/complexity and ease of implementation/ understanding). Such tradeoffs cannot be avoided and need to be addressed. Some models implemented cannot be too complex because companies don’t have hardware to support them (or because they are in regulated industries and need to be explained).

It’s critical to note this dataset was limited. The models generated should be validated on further customer datasets. Further variables should also be garnered (debt at bank, total debt, total savings, and savings/month would prove invaluable). As with the previous analysis, several improvements to this analysis can be made. The champion model should be further tweaked to increase accuracy (and ideally increase sensitivity). A cost analysis would need to be performed by the bank to determine how the model should be tuned. Assuming that marketing to customers is cheaper than the cost of losing a customer on the personal loan, then the model should be tweaked to increase sensitivity to 100%. In all likelihood, each customer would generate enough revenue to the marketing to multiple false positives. The bank should also explore the tradeoffs between accuracy of this model and computation power/energy/time to support using it. Thus, Universal Bank should adjust model 4, the gradient boosting model, to increase sensitivity and apply finding the clients most likely to need and accept the personal loan.

References:

Consumer Financial Protection Bureau (CFPB). (2022). *CFPB acts to protect the public from black-box credit models using complex algorithms*. https://www.consumerfinance.gov/about-us/newsroom/cfpb-acts-to-protect-the-public-from-black-box-credit-models-using-complex-algorithms/

Fitch, T. (2024). SVM Modeling on Universal Bank Dataset. GitHub. https://github.com/Capadetated/SVM-Modeling-on-Universal-Bank-Dataset/blob/main/Assignment1_Fitch.pdf

Knode, S. (2024). universal bank description.docx. University of Maryland Global Campus DATA 640 Learning Portal. Retrieved January 17th, 2024.

Polly, G. (2014). *The US grouped by first two zip code digits*. Imgur. https://imgur.com/NJGcg6v

SAS Software. (2017). *Decision trees, boosting trees, and random forests: A side-by-side comparison*. YouTube. https://www.youtube.com/watch?v=gehNcYRXs4M&ab_channel=SASSoftware

Shafique, U., & Qaiser, H. (2014). A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, *12*(1), 217-222.

Srivastava, T. (2022). *Support Vector Machine - simplified*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2014/10/support-vector-machine-simplified/

Srivastava, T. (2020). *Basics of Ensemble Learning explained in simple English*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/

<u>Appendix:</u>

Variable Name	Variable Meaning	Variable Type
Age	**Customer's age in completed years**	**Interval**
Experience	**Number of years of professional experience**	**Interval**
Income	**Annual income of the customer ($000)**	**Interval**
ZIPCode	**Home address ZIP code**	**Interval**
Family	**Family size of the customer**	**Ordinal**
CCAvg	**Average spending on total credit cards per month ($000)**	**Interval**
Education	**Education level: 1. Undergraduate 2. Graduate 3. Advanced/Professional**	**Nominal**
Mortgage	**Value of house mortgage ($000)**	**Interval**
Personal Loan	**Did this customer accept the personal loan offered in the last campaign?**	**Binary**
Securities Account	**Does the customer have a securities account with the bank?**	**Binary**
CD Account	**Does the customer have a certificate of deposit account with the bank?**	**Binary**
Online	**Does the customer use internet banking facilities?**	**Binary**
CreditCard	**Does the customer use a credit card issued by UniversalBank?**	**Binary**

Table 2. The 13 dataset variable descriptions and variable types generated based on the description given by Knode (2024).

Name	Role	Level	Number of Levels	Percent Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis
ID	ID	Interval	.	0	1	5000	2500.5	1443.52	0	-1.2
Income	Input	Interval	.	0	8	224	73.7742	46.03373	0.841339	-0.04424
Mortgage	Input	Interval	.	0	0	635	56.4988	101.7138	2.104002	4.756797
Family	Input	Ordinal	4	0	.	.	.	.	.	.
Securities_Accou	Input	Binary	.	.	.	.	.	.	.	.
ZIP_Code	Input	Interval	.	.	.	.	.	.	.	.
Online	Input	Binary	2	0	.	.	.	.	.	.
CCAvg	Input	Interval	.	0	0	10	1.937938	1.747659	1.598443	2.646706
CD_Account	Input	Binary	.	.	.	.	.	.	.	.
Age	Input	Interval	.	0	23	67	45.3384	11.46317	-0.02934	-1.15307
Experience	Input	Interval	.	0	-3	43	20.1046	11.46795	-0.02632	-1.12152
Education	Input	Nominal	3	0	.	.	.	.	.	.
CreditCard	Input	Binary	2	0	.	.	.	.	.	.
Personal_Loan	Target	Binary	.	.	.	.	.	.	.	.

Table 3. All variables and the dataset variable statistics show no missing values and no major skew.

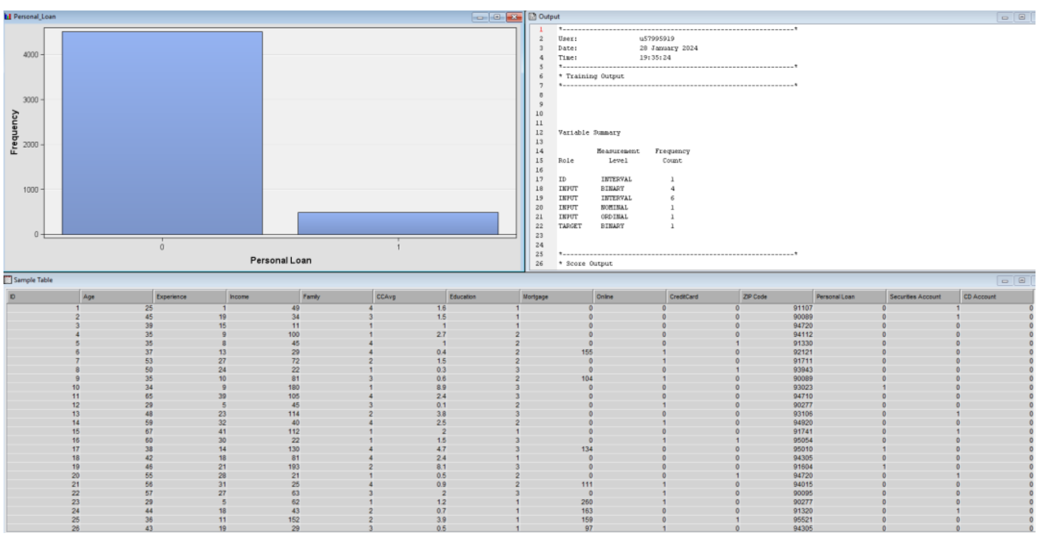


Figure 1. Graph showing the imbalance of the Personal Loan variable (left), the Variable Summary Table (right), and example data from UniversalBank dataset (bottom).

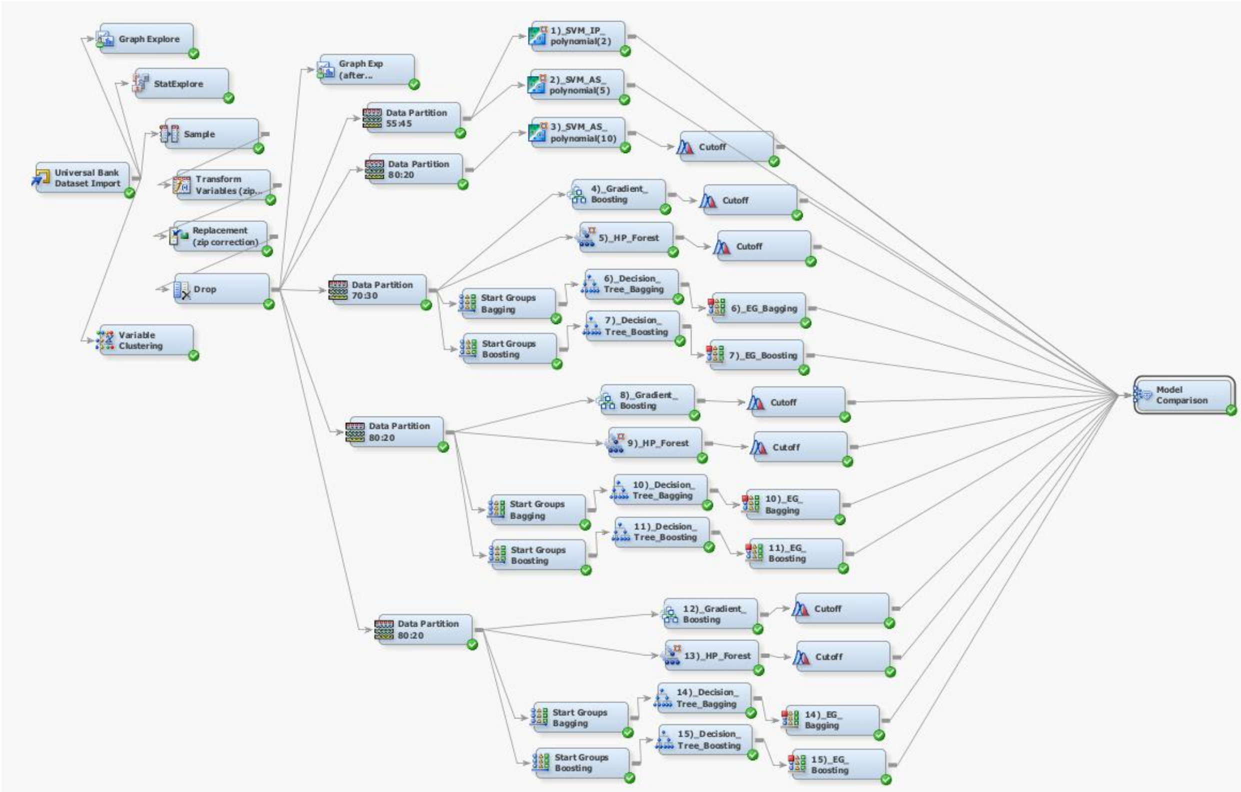


Figure 2. SAS Enterprise Miner Diagram of Ensemble Modeling.

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	CD_Account	INPUT	2	0	0	93.96	1	6.04
TRAIN	CreditCard	INPUT	2	0	0	70.60	1	29.40
TRAIN	Education	INPUT	3	0	1	41.92	3	30.02
TRAIN	Family	INPUT	4	0	1	29.44	2	25.92
TRAIN	Online	INPUT	2	0	1	59.68	0	40.32
TRAIN	Securities_Account	INPUT	2	0	0	89.56	1	10.44
TRAIN	Personal_Loan	TARGET	2	0	0	90.40	1	9.60

Figure 3. Class variable summary statistics.

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Age	INPUT	45.3384	11.46317	5000	0	23	45	67	-0.02934	-1.15307
CCAvg	INPUT	1.937938	1.747659	5000	0	0	1.5	10	1.598443	2.646706
Experience	INPUT	20.1046	11.46795	5000	0	-3	20	43	-0.02632	-1.12152
Income	INPUT	73.7742	46.03373	5000	0	8	64	224	0.841339	-0.04424
Mortgage	INPUT	56.4988	101.7138	5000	0	0	0	635	2.104002	4.756797
ZIP_Code	INPUT	93152.5	2121.852	5000	0	9307	93437	96651	-12.5002	486.2043

Figure 4. Interval variable summary statistics.

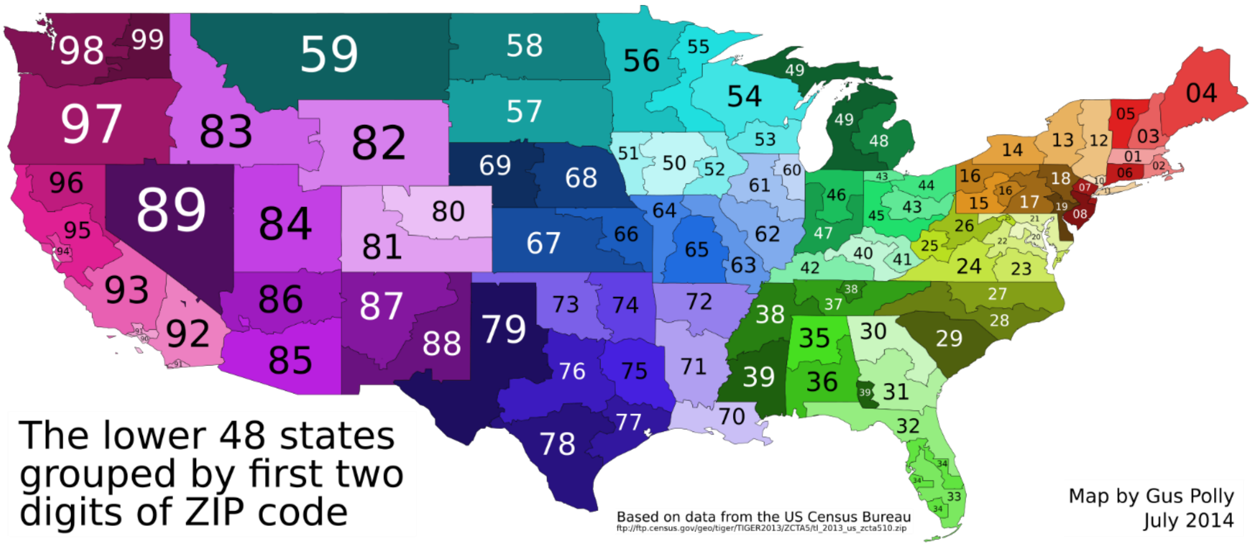


Figure 5. Map showing the lower 48 United States grouped by first 2 digits of ZIP code (All 90-96 ZIP codes can be seen in California)(Polly, 2014).

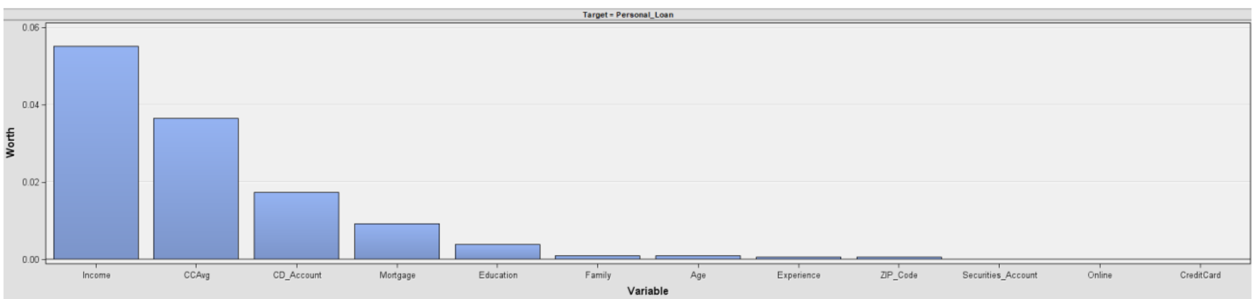


Figure 6. Variable worth for each variable in the UniversalBank dataset.

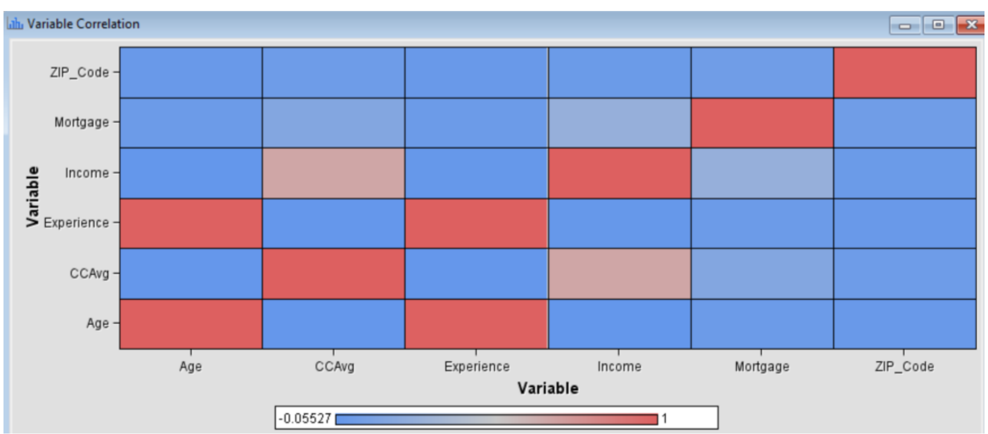
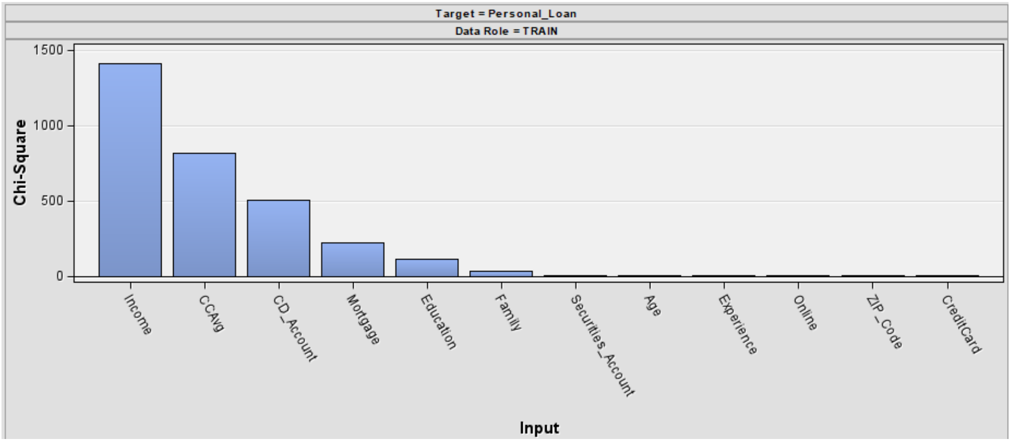


Figure 7. Variable correlation matrix for each variable in the UniversalBank dataset.



Chi-Square Statistics
(maximum 500 observations printed)

Data Role=TRAIN Target=Personal_Loan

Input	Chi-Square	Df	Prob
Income	1410.6154	4	<.0001
CCAvg	817.4473	4	<.0001
CD_Account	500.4019	1	<.0001
Mortgage	219.3955	4	<.0001
Education	111.2399	2	<.0001
Family	29.6761	3	<.0001
Securities_Account	2.4099	1	0.1206
Age	0.6125	4	0.9617
Experience	0.4612	4	0.9772
Online	0.1971	1	0.6571
ZIP_Code	0.1062	1	0.7445
CreditCard	0.0392	1	0.8430

Figure 8. Chi-Square values for each variable in the UniversalBank dataset (in chart and table form).

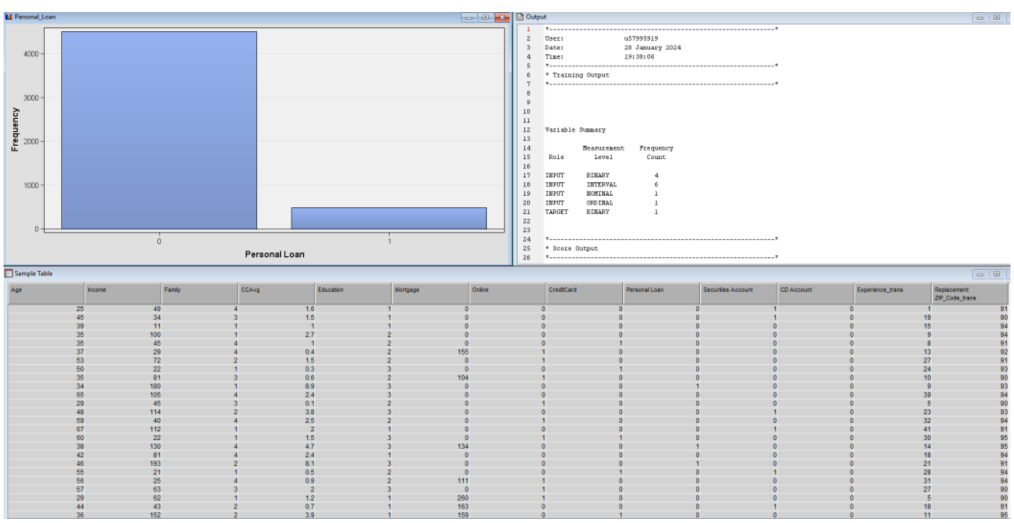


Figure 9. The dataset after preprocessing occurred. The Variable Summary Table (right), and example data from UniversalBank dataset (bottom) can be seen with new variables introduced.

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	EndGrp6	EndGrp6	15)_EG_Bo...	Personal_L...		0.008982
	CUT5	HPDMFore...	13)_HP_Fo...	Personal_L...		0.010978
	CUT3	HPDMFore...	9)_HP_For...	Personal_L...		0.010978
	CUT2	Boost3	12)_Gradie...	Personal_L...		0.013972
	CUT4	Boost2	8)_Gradient...	Personal_L...		0.01497
	CUT6	HPDMForest	5)_HP_For...	Personal_L...		0.015313
	HPSVM2	HPSVM2	3)_SVM_AS...	Personal_L...		0.016966
	CUT7	Boost	4)_Gradient...	Personal_L...		0.017976
	HPSVM8	HPSVM8	2)_SVM_AS...	Personal_L...		0.018206
	HPSVM5	HPSVM5	1)_SVM_IP...	Personal_L...		0.018206
	EndGrp3	EndGrp3	10)_EG_Ba...	Personal_L...		0.018962
	EndGrp	EndGrp	6)_EG_Bag...	Personal_L...		0.020639
	EndGrp4	EndGrp4	11)_EG_Bo...	Personal_L...		0.045908
	EndGrp2	EndGrp2	7)_EG_Boo...	Personal_L...		0.052597
	EndGrp5	EndGrp5	14)_EG_Ba...	Personal_L...		0.096806

Figure 10. The fit statistics of the model comparison node showing the results (not using the sampling technique to control for having a heavily imbalanced target variable) show better results in model accuracy but worse results in model sensitivity than not using sampling.

Ind	Model Type	Model Description	Data Role	Target	FN	TN	FP	TP	Sensitivity	Specificity	Precision	F1 Score	Misclassification Rate	Accuracy
1	SVM	1)_SVM_IP_polynomial(2)	TRAIN	Personal_Loan	30	2478	7	233	0.89	1.00	0.97	0.93	0.014	98.7%
1	SVM	1)_SVM_IP_polynomial(2)	VALIDATE	Personal_Loan	25	2019	16	192	0.88	0.99	0.92	0.90	0.018	98.2%
2	SVM	2)_SVM_AS_polynomial(5)	TRAIN	Personal_Loan	30	2478	7	233	0.89	1.00	0.97	0.93	0.014	98.7%
2	SVM	2)_SVM_AS_polynomial(5)	VALIDATE	Personal_Loan	25	2019	16	192	0.88	0.99	0.92	0.90	0.018	98.2%
3	SVM	3)_SVM_AS_polynomial(10)	TRAIN	Personal_Loan	39	3606	9	344	0.90	1.00	0.97	0.93	0.012	98.8%
3	SVM	3)_SVM_AS_polynomial(10)	VALIDATE	Personal_Loan	8	896	9	89	0.92	0.99	0.91	0.91	0.017	98.3%
4	Gradient Boosting	4)_Gradient_Boosting	TRAIN	Personal_Loan	57	3158	5	278	0.83	1.00	0.98	0.90	0.018	98.2%
4	Gradient Boosting	4)_Gradient_Boosting	VALIDATE	Personal_Loan	22	1352	5	123	0.85	1.00	0.96	0.90	0.018	98.2%
5	Random Forest	5)_HP_Forest	TRAIN	Personal_Loan	8	3163	0	327	0.98	1.00	1.00	0.99	0.002	99.8%
5	Random Forest	5)_HP_Forest	VALIDATE	Personal_Loan	19	1353	4	126	0.87	1.00	0.97	0.92	0.015	98.5%
6	Bagging	6)_EG_Bagging	TRAIN	Personal_Loan	30	3128	35	305	0.91	0.99	0.90	0.90	0.019	98.1%
6	Bagging	6)_EG_Bagging	VALIDATE	Personal_Loan	11	1337	20	134	0.92	0.99	0.87	0.90	0.021	97.9%
7	Boosting	7)_EG_Boosting	TRAIN	Personal_Loan	25	3054	109	310	0.93	0.97	0.74	0.82	0.038	96.2%
7	Boosting	7)_EG_Boosting	VALIDATE	Personal_Loan	17	1295	62	128	0.88	0.95	0.67	0.76	0.050	95.0%
8	Gradient Boosting	8)_Gradient_Boosting	TRAIN	Personal_Loan	54	3607	8	329	0.86	1.00	0.98	0.91	0.016	98.5%
8	Gradient Boosting	8)_Gradient_Boosting	VALIDATE	Personal_Loan	10	900	5	87	0.90	0.99	0.95	0.92	0.015	98.5%
9	Random Forest	9)_HP_Forest	TRAIN	Personal_Loan	13	3615	0	370	0.97	1.00	1.00	0.98	0.003	99.7%
9	Random Forest	9)_HP_Forest	VALIDATE	Personal_Loan	10	904	1	87	0.90	1.00	0.99	0.94	0.011	98.9%
10	Bagging	10)_EG_Bagging	TRAIN	Personal_Loan	76	3602	13	307	0.80	1.00	0.96	0.87	0.022	97.8%
10	Bagging	10)_EG_Bagging	VALIDATE	Personal_Loan	17	903	2	80	0.82	1.00	0.98	0.89	0.019	98.1%
11	Boosting	11)_EG_Boosting	TRAIN	Personal_Loan	59	3491	124	324	0.85	0.97	0.72	0.78	0.046	95.4%
11	Boosting	11)_EG_Boosting	VALIDATE	Personal_Loan	10	869	36	87	0.90	0.96	0.71	0.79	0.046	95.4%
12	Gradient Boosting	12)_Gradient_Boosting	TRAIN	Personal_Loan	52	3607	8	331	0.86	1.00	0.98	0.92	0.015	98.5%
12	Gradient Boosting	12)_Gradient_Boosting	VALIDATE	Personal_Loan	9	900	5	88	0.91	0.99	0.95	0.93	0.014	98.6%
13	Random Forest	13)_HP_Forest	TRAIN	Personal_Loan	13	3615	0	370	0.97	1.00	1.00	0.98	0.003	99.7%
13	Random Forest	13)_HP_Forest	VALIDATE	Personal_Loan	10	904	1	87	0.90	1.00	0.99	0.94	0.011	98.9%
14	Bagging	14)_Decision_Tree_Bagging	TRAIN	Personal_Loan	383	3615	0	0	0.00	1.00	0.00	0.00	0.096	90.4%
14	Bagging	14)_Decision_Tree_Bagging	VALIDATE	Personal_Loan	97	905	0	0	0.00	1.00	0.00	0.00	0.097	90.3%
15	Boosting	15)_Decision_Tree_Boosting	TRAIN	Personal_Loan	65	3609	6	318	0.83	1.00	0.98	0.90	0.000	100.0%
15	Boosting	15)_Decision_Tree_Boosting	VALIDATE	Personal_Loan	12	905	0	85	0.88	1.00	1.00	0.93	0.009	99.1%

Figure 11. Example fit statistics without the sampling technique show a decrease in overall sensitivity and F1.

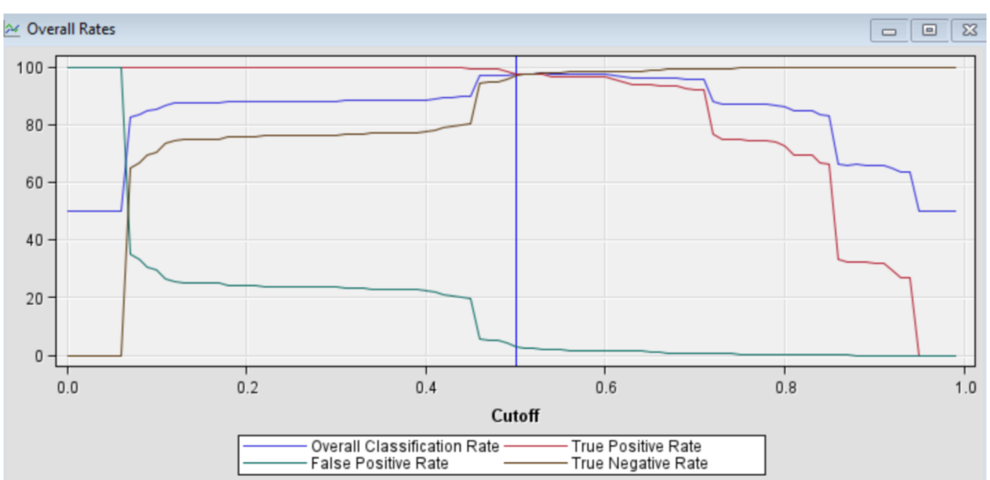


Figure 12. The overall rates of classification, TP, FP, and TN using a cutoff of 0.5 for model 4.

Index	Data Partition (train: val)	Diagram Title	Model Type	Optimization Method	Method Setting
1	55:45	1)_SVM_IP_polynomial(2)	SVM	Interior Point	2nd degree
2	55:45	2)_SVM_AS_polynomial(5)	SVM	Interior Point	2nd degree
3	80:20	3)_SVM_AS_polynomial(10)	SVM	Active Set	2nd degree
4	70:30	4)_Gradient_Boosting	Gradient Boosting	N/A	Default
5	70:30	5)_HP_Forest	Random Forest	N/A	Default
6	70:30	6)_Decision_Tree_Bagging	Decision Tree Bagging	N/A	Default
7	70:30	7)_Decision_Tree_Boosting	Decision Tree Boosting	N/A	Default
8	80:20	8)_Gradient_Boosting	Gradient Boosting	N/A	Default
9	80:20	9)_HP_Forest	Random Forest	N/A	Default
10	80:20	10)_Decision_Tree_Bagging	Decision Tree Bagging	N/A	Default
11	80:20	11)_Decision_Tree_Boosting	Decision Tree Boosting	N/A	Default
12	80:20	12)_Gradient_Boosting	Gradient Boosting	N/A	100 iterations
13	80:20	13)_HP_Forest	Random Forest	N/A	1,000 iterations
14	80:20	14)_Decision_Tree_Bagging	Decision Tree Bagging	N/A	Subtree method: N
15	80:20	15)_Decision_Tree_Boosting	Decision Tree Boosting	N/A	Max depth: 10 Min categorical size: 10

Table 4. Overview of created predictive models shows 3 SVMs (from previous analysis) and 12 EMs. The EMs are the same 4 models iterated 3 times using a different dataset partition, and different method settings.

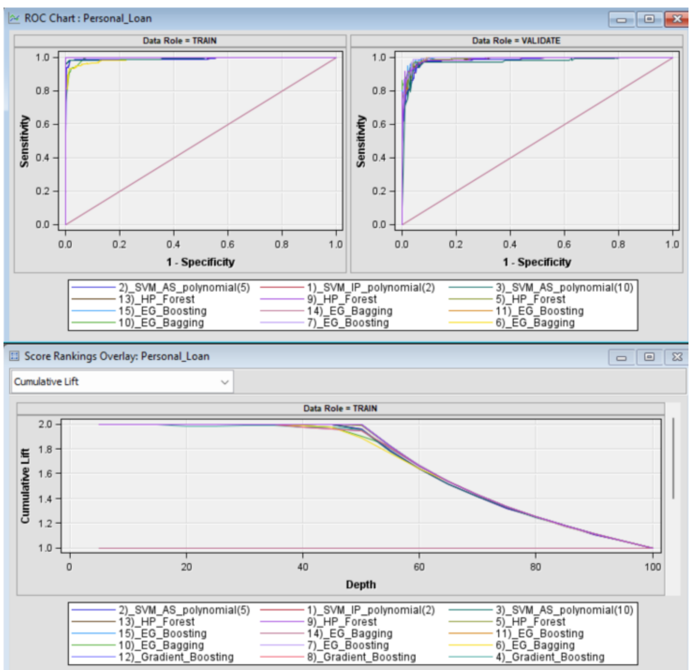


Figure 13. ROC Chart and Cumulative Lift Chart of Models 1-15.

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Cumulative Lift
Y	EndGrp3	EndGrp3	10)_EG_Ba...	Personal_L...		2.010417
	EndGrp6	EndGrp6	15)_EG_Bo...	Personal_L...		2.010417
	CUT5	HPDMFore...	13)_HP_Fo...	Personal_L...		2.010417
	CUT3	HPDMFore...	9)_HP_For...	Personal_L...		2.010417
	EndGrp4	EndGrp4	11)_EG_Bo...	Personal_L...		2.010417
	CUT4	Boost2	8)_Gradient...	Personal_L...		2.010417
	CUT2	Boost3	12)_Gradie...	Personal_L...		2.010417
	CUT	HPSVM2	3)_SVM_AS...	Personal_L...		2.010417
	CUT6	HPDMForest	5)_HP_For...	Personal_L...		2.006944
	EndGrp2	EndGrp2	7)_EG_Boo...	Personal_L...		2.006944
	CUT7	Boost	4)_Gradient...	Personal_L...		2.006944
	HPSVM5	HPSVM5	1)_SVM_IP...	Personal_L...		2
	HPSVM8	HPSVM8	2)_SVM_AS...	Personal_L...		2
	EndGrp	EndGrp	6)_EG_Bag...	Personal_L...		1.98154
	EndGrp5	EndGrp5	14)_EG_Ba...	Personal_L...		1

Figure 14. Fit statistics from the model comparison node using Cumulative Lift as the selection criterion.

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Roc Index
Y	EndGrp3	EndGrp3	10)_EG_Ba...	Personal_L...		0.994
	EndGrp6	EndGrp6	15)_EG_Bo...	Personal_L...		0.994
	EndGrp2	EndGrp2	7)_EG_Boo...	Personal_L...		0.994
	CUT6	HPDMForest	5)_HP_For...	Personal_L...		0.991
	CUT7	Boost	4)_Gradient...	Personal_L...		0.991
	EndGrp4	EndGrp4	11)_EG_Bo...	Personal_L...		0.991
	CUT3	HPDMFore...	9)_HP_For...	Personal_L...		0.99
	CUT5	HPDMFore...	13)_HP_Fo...	Personal_L...		0.989
	CUT4	Boost2	8)_Gradient...	Personal_L...		0.986
	CUT2	Boost3	12)_Gradie...	Personal_L...		0.986
	EndGrp	EndGrp	6)_EG_Bag...	Personal_L...		0.984
	HPSVM5	HPSVM5	1)_SVM_IP...	Personal_L...		0.979
	HPSVM8	HPSVM8	2)_SVM_AS...	Personal_L...		0.979
	CUT	HPSVM2	3)_SVM_AS...	Personal_L...		0.973
	EndGrp5	EndGrp5	14)_EG_Ba...	Personal_L...		0.5

Figure 15. Fit statistics from the model comparison node using ROC Index as the selection criterion.

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	EndGrp6	15)_EG_Boosting	0.03627	0.02375	0.00000	0.04140
	Boost	4)_Gradient_Boosting	0.04152	0.04956	0.02683	0.05366
	HPDMForest	5)_HP_Forest	0.04498	0.01390	0.00149	0.04184
	HPDMForest3	13)_HP_Forest	0.04663	0.01172	0.00000	0.04288
	Boost2	8)_Gradient_Boosting	0.04663	0.04919	0.03129	0.06281
	Boost3	12)_Gradient_Boosting	0.04663	0.04919	0.03129	0.06281
	HPSVM2	3)_SVM_AS_polynomial(10)	0.04663	0.12103	0.01956	0.12471
	HPDMForest2	9)_HP_Forest	0.05181	0.01172	0.00000	0.04392
	EndGrp	6)_EG_Bagging	0.05190	0.03779	0.05961	0.04114
	EndGrp3	10)_EG_Bagging	0.05699	0.03350	0.05867	0.03781
	HPSVM5	1)_SVM_IP_polynomial(2)	0.06019	0.12000	0.03030	0.12697
	HPSVM8	2)_SVM_AS_polynomial(5)	0.06019	0.12001	0.03030	0.12698
	EndGrp4	11)_EG_Boosting	0.07772	0.04467	0.05997	0.06176
	EndGrp2	7)_EG_Boosting	0.10381	0.03641	0.07601	0.04869
	EndGrp5	14)_EG_Bagging	0.50259	0.25001	0.49935	0.25003

Figure 16. Fit statistics of the model comparison node for Models 1-15.