Assignment 3

Ensemble Modeling 2 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 20th, Spring 2024

Introduction:

The purpose of this report was to analyze the Universal Bank dataset and predict whether customers will accept the bank's personal loan using Ensemble Models (EMs) via the Ensemble node. EMs are classification prediction models that "average" the results of heterogenous "weaker" models in order to generate more accurate results (Srivastava, 2022). There are 4 methods the Ensemble node has: Average (averaging posterior probabilities), Maximum (taking the maximum of the posterior probabilities), Voting Average (averaging the posterior probabilities only for models in majority group), and Voting Proportion (the posterior probabilities calculated as a ratio of the number of models in the majority group versus total number of models)(Sarma, 2013). 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 the impact of having outliers in the data (SAS Software, 2017). Thus, EM models were developed and compared to the previous 2 analyses of SVM and EM modeling on this dataset in order to contrast their approaches and accuracy (Fitch, 2024a)(Fitch, 2024b). All background of this dataset was previously discussed; 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:

No new features were used for preprocessing this dataset which was different from the previous EM analysis (Fitch, 2024b). All data exploration and preprocessing can be read in the previous report. But a high-level summary will be given. 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 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). This was performed using a Transform Variables node (and the negative values in the Experience variable were corrected to positive). Then, since 1 value had only 4 digits, a correction was needed using a Replacement node. No feature engineering was deemed helpful for this dataset. 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 data was also sampled to have an equal distribution of positive/negative values of the target variable.

Models and Methods:

The cleaned/processed dataset was run using 4 weaker model types, and 3 iterations of the 4 ensemble node method types (Table 4)(Figure 2). First, the 4 weaker models were run in order to see how accurate each weaker model was on its own: Naïve Bayes, Decision Tree, Neural Network, and Logistic Regression. An 80:20 data partition was used since that ratio yielded the best overall results from the previous analysis. Then, each of those 4 weaker models were connected to each of the 4 ensemble methods. This was iterated 4 times originally to check results for different data partitions (80:20, 70:30, 60:40, and 55:45); 60:40 yielded the highest average sensitivity and 70:30 resulted the widest spread of sensitivity) and thus those 2 clusters were kept and run for each of the 4 ensemble methods (Figure 10). Finally, the 4 ensemble methods were run using 9 weaker models (the 4 used in the other clusters, as well as an SVM, 2 high performance decision trees (HPDT), a bagging, and a boosting model). These models were chosen in the anticipation that having more models would lead to more accurate results. The data partition was iterated multiple times to find the optimum accuracy measures (80:20 was chosen). The bagging/boosting models were models 6/15 respectively from the previous EM analysis chosen because they had the best sensitivity of the bagging/boosting models generated (Fitch, 2024b). The neural network model had hidden layers and standardization added (which the other NNs generated did not use). The SVM used interior point method of polynomial 2. The HPDT2 used default settings whereas the HPDT1 used a maximum branch of 5, maximum depth of 20,

and minimum categorial size of 2. Only certain models that generated SAS DATA step code could be connected to the ensemble node, so previously used models like random forests, gradient boosting, and active set SVMs could not be used (SAS Support, n.d.). If possible, they would have been for direct comparability between the previous analyses and this one.

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). The SVM had a cost of c = 1 applied. This penalized it for making incorrect classifications which forces the model 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. Thus, this approach was used as the first major approach to address the imbalance. Lastly, the cutoff criterion was finally used to optimize the models. This node was applied to all the models post ensemble node only (not weaker models feeding into the ensemble node). It was first explored using the default cutoff of 0.5 on all models and adjusted to an optimum value for each model (many were kept at or near 0.5, but several needed to be adjusted significantly).

Results and Model Evaluation:

As with the previous analysis, the selection criteria used to determine the best models were sensitivity (the true positive rate/TPR/Recall) and F1 score (the harmonic mean of precision and sensitivity)(Fitch, 2024b). The former was primarily 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.

Index	Model Type	Model Description	Data Role	FN	TN	FP	TP	Sensitivity / TPR	Specificity	Precision	F1 Score	Misclassification Rate	Accuracy
1	Naïve Bayes	1)_HP_BN_Classifier	Train	27	327	56	357	0.93	0.85	0.86	0.90	10.8%	89.2%
2	Naïve Bayes	1)_HP_BN_Classifier	Validate	8	73	24	88	0.92	0.75	0.79	0.85	16.6%	83.4%
3	Decision Tree	2)_Decision_Tree	Train	23	376	7	361	0.94	0.98	0.98	0.96	3.9%	96.1%
4	Decision Tree	2)_Decision_Tree	Validate	5	93	4	91	0.95	0.96	0.96	0.95	4.7%	95.3%
5	Neural Network	3)_Neural_Network	Train	10	380	3	374	0.97	0.99	0.99	0.98	1.7%	98.3%
6	Neural Network	3)_Neural_Network	Validate	5	93	4	91	0.95	0.96	0.96	0.95	4.7%	95.3%
7	Regression	4)_Regression	Train	41	344	39	343	0.89	0.90	0.90	0.90	10.4%	89.6%
8	Regression	4)_Regression	Validate	7	85	12	89	0.93	0.88	0.88	0.90	9.8%	90.2%
9	EM: Average	5)_Ensemble(AVG)70	Train	13	330	5	323	0.96	0.99	0.98	0.97	2.7%	97.3%
10	EM: Average	5)_Ensemble(AVG)70	Validate	7	138	7	137	0.95	0.95	0.95	0.95	4.8%	95.2%
11	EM: Max	6)_Ensemble(MAX)70	Train	6	267	68	330	0.98	0.80	0.83	0.90	11.0%	89.0%
12	EM: Max	6)_Ensemble(MAX)70	Validate	3	113	32	141	0.98	0.78	0.82	0.89	12.1%	87.9%
13	EM: Voting Average	7)_Ensemble(VOT_AVG)70	Train	8	309	26	328	0.98	0.92	0.93	0.95	5.1%	94.9%
14	EM: Voting Average	7)_Ensemble(VOT_AVG)70	Validate	6	130	15	138	0.96	0.90	0.90	0.93	7.3%	92.7%
15	EM: Voting Proportion	8)_Ensemble(VOT_PRO)70	Train	8	309	26	328	0.98	0.92	0.93	0.95	5.1%	94.9%
16	EM: Voting Proportion	8)_Ensemble(VOT_PRO)70	Validate	6	130	15	138	0.96	0.90	0.90	0.93	7.3%	92.7%
17	EM: Average	9)_Ensemble(AVG)60	Train	9	280	8	279	0.97	0.97	0.97	0.97	3.0%	97.0%
18	EM: Average	9)_Ensemble(AVG)60	Validate	6	181	11	186	0.97	0.94	0.94	0.96	4.4%	95.6%
19	EM: Max	10)_Ensemble(MAX)60	Train	4	236	52	284	0.99	0.82	0.85	0.91	9.7%	90.3%
20	EM: Max	10)_Ensemble(MAX)60	Validate	5	139	53	187	0.97	0.72	0.78	0.87	15.1%	84.9%
21	EM: Voting Average	11)_Ensemble(VOT_AVG)60	Train	8	267	21	280	0.97	0.93	0.93	0.95	5.0%	95.0%
22	EM: Voting Average	11)_Ensemble(VOT_AVG)60	Validate	6	166	26	186	0.97	0.86	0.88	0.92	8.3%	91.7%
23	EM: Voting Proportion	12)_Ensemble(VOT_PRO)60	Train	8	267	21	280	0.97	0.93	0.93	0.95	5.0%	95.0%
24	EM: Voting Proportion	12)_Ensemble(VOT_PRO)60	Validate	6	166	26	186	0.97	0.86	0.88	0.92	8.3%	91.7%
25	EM: Average	13)_Ensemble(AVG)80	Train	11	378	5	373	0.97	0.99	0.99	0.98	2.1%	97.9%
26	EM: Average	13)_Ensemble(AVG)80	Validate	3	92	5	93	0.97	0.95	0.95	0.96	4.2%	95.9%
27	EM: Max	14)_Ensemble(MAX)80	Train	0	290	93	384	1.00	0.76	0.81	0.89	12.1%	87.9%
28	EM: Max	14)_Ensemble(MAX)80	Validate	0	67	30	96	1.00	0.69	0.76	0.86	11.0%	89.0%
29	EM: Voting Average	15)_Ensemble(VOT_AVG)80	Train	10	378	5	374	0.97	0.99	0.99	0.98	3.9%	96.1%
30	EM: Voting Average	15)_Ensemble(VOT_AVG)80	Validate	3	91	6	93	0.97	0.94	0.94	0.95	2.3%	97.7%
31	EM: Voting Proportion	16)_Ensemble(VOT_PRO)80	Train	10	378	5	374	0.97	0.99	0.99	0.98	3.4%	96.6%
32	EM: Voting Proportion	16)_Ensemble(VOT_PRO)80	Validate	3	91	6	93	0.97	0.94	0.94	0.95	1.6%	98.4%

Table 1. Accuracy measures of the 16 models generated evaluated by TPR and F1 show that models 13 and 14 are the champions with a TPR of 0.97 / 1.00 and F1 of 0.96 / 0.86 on the validation data. These models show tradeoffs between higher overall accuracy versus true positive identification and thus cost-analysis must be performed to determine ROI of these tradeoffs.

The baseline TPR is 0.91 (since that's the likelihood of picking a negative customer in the population according to the total dataset). All models had a TPR >0.91 (average 0.96) on the validation data. The 4 weaker models all had TPR >0.91 (average 0.93); Average Ensemble models showed average TPR of 0.96; the 2 voting method models showed average TPR of (0.97); and the Maximum Ensembles showed 0.98 TPR. However, the inverse was true that the Voting models showed overall highest average accuracy and the Maximum models showed lowest overall lowest average accuracy. This is because the Maximum models were tuned to identify positive customers more accurately at the cost of incorrectly identifying false positives.

Thus, there are 2 proposed champion models: Model 14 had a TPR of 1.00, F1 of 0.86, and accuracy of 89.0%. This model identified every customer in the dataset who accepted the bank loan (however it also wrongly identified 30 false positives with a specificity of only 0.69). On the other hand, Model 13 had a TPR of 0.97, F1 of 0.96, and accuracy of 95.9%. This also makes sense for the method types since Average assumes equal contribution from all models and a smoothed estimate whereas Maximum gives strongest weight to the best model (and it had 9 to choose from). Since marketing to customers is a relatively low cost compared to the revenue a new customer generates when choosing a personal loan, Model 14 eliminates as many false negatives as possible. However, if the bank's cost analysis shows targeting marketing is too expensive and a more balanced approach is needed, then Model 13 should be chosen.

Overfitting is necessary to check for when modeling; in this case, 2 models saw a 5% accuracy decrease between train and validation datasets (Models 1&10) however all others were <3%. In fact, Models 14-16 saw slight increases in accuracy from train to validation. Similarly, we don't see dramatic decreases in ROC-AUC from training to validation datasets which would also indicate overfitting. On the other hand, all models had <3% increases or decreases in TPR between train and validation data (Table 1). Slight overfitting may have occurred with Models 1&10 in terms of accuracy; but, no major overfitting occurred for all models in terms of accuracy nor TPR. The methods used to account for the imbalanced target variable worked efficiently since all accuracy measure show positive results: sensitivity is above baseline for all models; F1 and accuracy are high for all models; ROC Index and Cumulative lift both show significantly better results of these models from baseline; and lastly no significant overfitting was observed from train to validation data (Figure 12, Figure 13, Figure 14)(see Figure 11 for example cutoff chart for Model 14). While the best test to determine if the imbalance methods were effective

would be to test the models on further data, the available metrics show they were effective (via cutoff, cost, and balanced sampled datasets). The cumulative lift and ROC charts similarly show that all models performed very well by showing they were all significantly lifted from using a random model (all cumulative lift was >1.96) and they all had AUC-ROC of >0.96.

To that end, F1 shows how TPR and precision are balanced. In this case, it is necessary to sacrifice some model specificity (the true negative rate) in order to increase TPR. When comparing this analysis to the previous SVM (Fitch, 2024a) and EM (Fitch, 2024b) analysis, we glean several insights. First, the average TPR, F1, and accuracy of the simple/weaker models is on average slightly above baseline but ~2% worse than all other model types. SVMs performed mediocrely with TPR/F1/accuracy of 95%, 94%, and 94%. Comparing the best standalone EM (gradient boosting) versus the best Ensemble node method (Maximum), shows a similar average TPR but higher overall accuracy for gradient boosting models. Meaning during exploration of overall accurate models, gradient boosting models show the best average metrics. However, the best individual models observed for TPR was model 14 (index 28). These results show a gradient boosting model is mostly likely to generate balanced results of accuracy/TPR. But if the bank deems that the cost/benefit ratio to losing a customer on this personal loan is higher than marketing costs, then a highly sensitive model like Model 14 should be chosen (Table 5).

Conclusion, Limitations, and Improvements:

In conclusion, 16 predictive models were created (4 weaker models, 12 EMs) to predict which bank customers were going to accept the personal loan from the bank. Of those, all models showed a higher sensitivity than randomly guessing which customers would likely choose the loan (baseline = 90%). Two champion models were recommended: Model 13 had a balance between decent TPR and overall accuracy (97% and 95.9%). Model 14 had a perfect TPR of

100% but slightly lower overall accuracy of 89%. These were also compared to the previous analyses; SVMs being the worst overall, gradient boosting having the best results for both TPR and accuracy, and ensemble maximum having the best TPR results. The bank would need a cost analysis to know how much cost marketing, time, or employees spending time cold-calling customers would be worth it in order to gain a new customer. If the personal loan revenue is higher than the cost of marketing, then the bank should always choose the more sensitive model. Even if the marketing doesn't succeed at gaining new customers specific to this personal loan, the advertising may help gain new customers for other bank commodities like a CD or brokerage account (or decrease churn). In implementation, there are always tradeoffs to be found (between accuracy/complexity and ease of implementation/understanding). 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 how conclusions were reached). Model 14 may prove too complex in a production environment since it is made of 9 other models. As with the previous analyses, this modeling is limited by the dataset. More data would be helpful to test the models to prove their accuracy/TPR. Further variables should also be garnered (debt at bank, total debt, total savings, and savings/month would prove invaluable). The dataset could also be explored using only the most important variables. Ensemble nodes also are highly permutative (many more/different weaker models could be added, or the strongest ensemble models could be added into one ensemble model). Before deciding on one model, more iterations of these ensemble node models should be made. Through this method, the champion Model 14 should be further tweaked to increase accuracy while maintaining perfect sensitivity. Thus, Universal Bank should adjust model 14, the ensemble maximum model, to increase accuracy 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. (2024a). SVM Modeling on Universal Bank Dataset. GitHub. https://github.com/Capadetated/SVM-Modeling-on-Universal-Bank-Dataset/blob/main/Assignment1_Fitch.pdf

Fitch, T. (2024b). Ensemble Modeling on Universal Bank Dataset. GitHub. https://github.com/Capadetated/Ensemble-Modeling-on-Universal-Bank-Dataset/blob/main/Assignment2_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

Sarma, K. (2013). Chapter 7: Comparison and Combination of Different Models. In *Predictive Modeling with SAS Enterprise Miner*. SAS.

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

SAS Support. (n.d.). SAS Enterprise Miner Tools Production of Score Code HPDM Nodes corrected. SAS Support. https://support.sas.com/kb/57/addl/fusion_57062_1_sas_enterprise_miner_tools_production_of_score_code_hpdm_nod.pdf

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/

Appendix:

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.

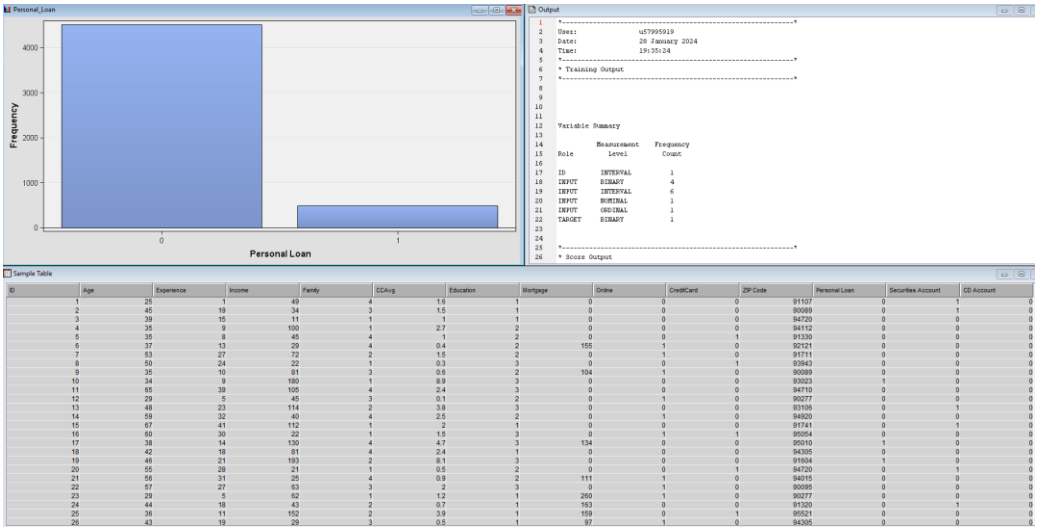

Figure 1. Graph showing the imbalance of the Personal Loan variable (left), the Variable Summary Table (right), and example data from UniversalBank dataset (bottom).

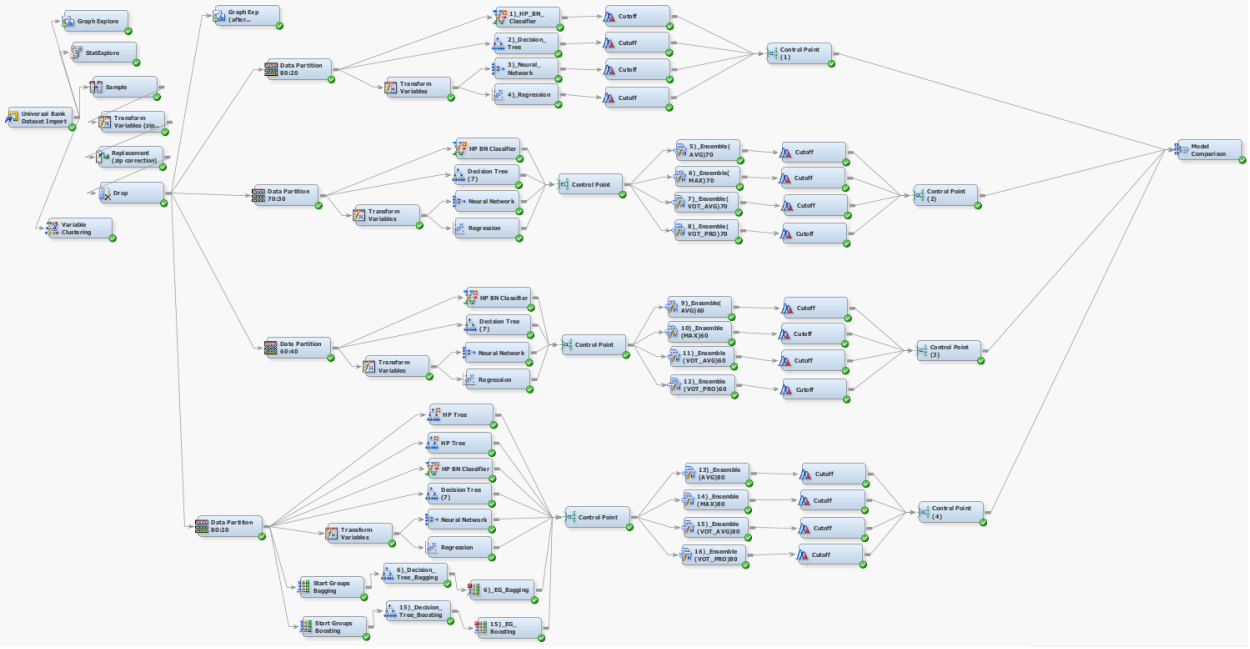

Figure 2. SAS Enterprise Miner Diagram of Ensemble Modeling using the ensemble node. 4 clusters of models exist: 4 standalone weaker models (top), using the 4 methods of ensemble node using a 70:30 data partition (top middle), using the 4 methods of ensemble node using a 60:40 data partition (bottom middle), using the 4 methods of ensemble node using 8 weaker models together (bottom).

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.

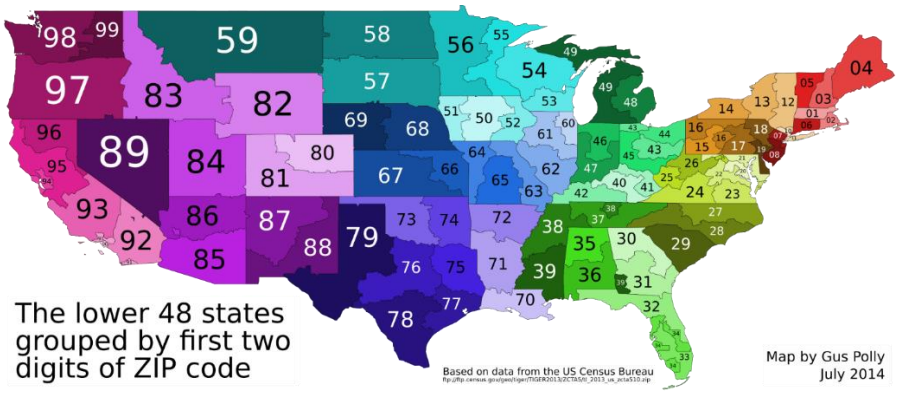

Figure 5. Map showing the lower 48 United States grouped by first 2 digits of ZIP code (All 90-96 ZIP codes can be seen in California)(Polly, 2014).

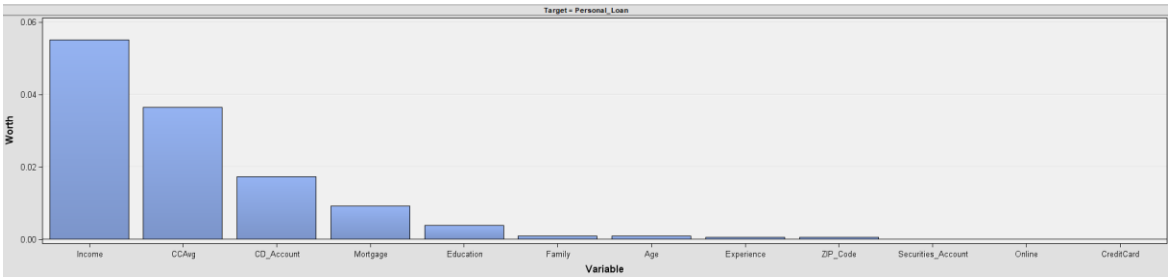

Figure 6. Variable worth for each variable in the UniversalBank dataset.

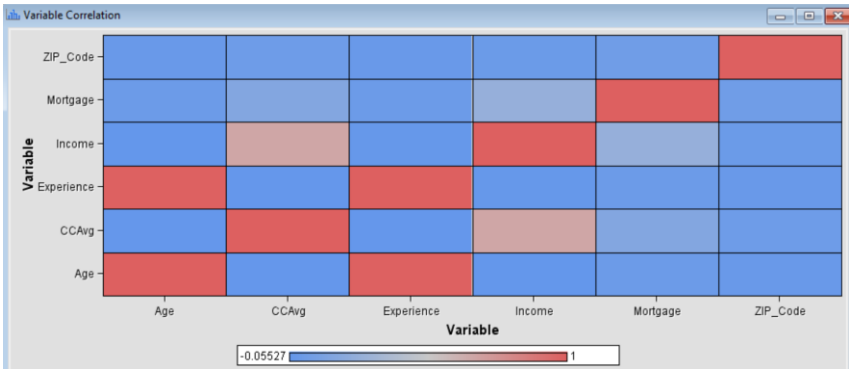

Figure 7. Variable correlation matrix for each variable in the UniversalBank dataset.

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

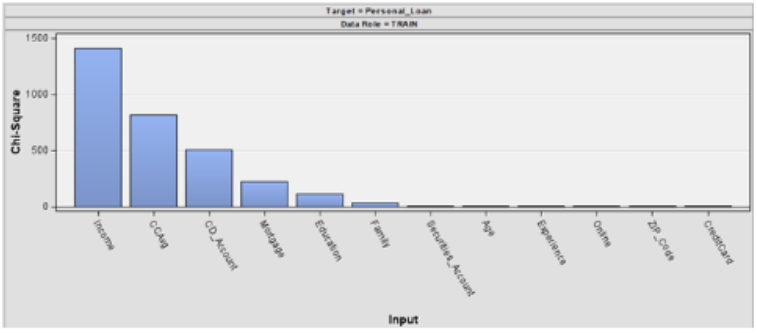

Figure 8. Chi-Square values for each variable in the UniversalBank dataset (in chart and table form).

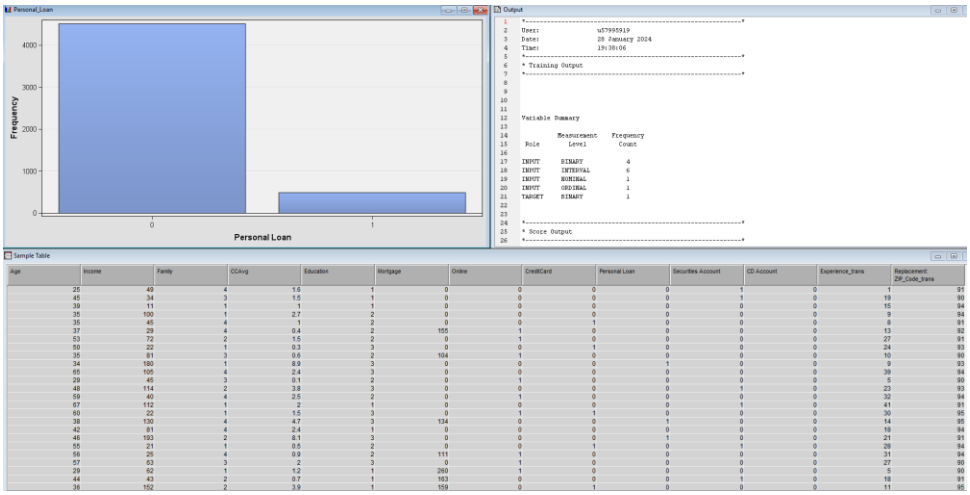

Figure 9. The dataset after preprocessing occurred. The Variable Summary Table (right), and example data from UniversalBank dataset (bottom) can be seen with new variables introduced.

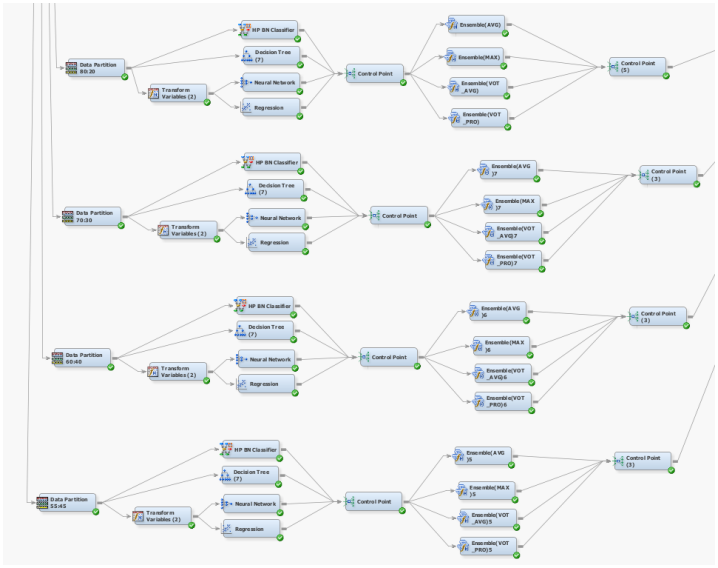

Figure 10. SAS Enterprise Miner Diagram of EM using the ensemble node: 4 clusters of models were iterated: using a data partition of 80:20, 70:30, 60:40, and 55:45. The 70:30 and 60:40 partitions were chosen because the latter had the highest average TPR and the former had the widest spread of TPR. This image was not the final diagram but was simply used as proof of concept (see Figure 2 for final diagram).

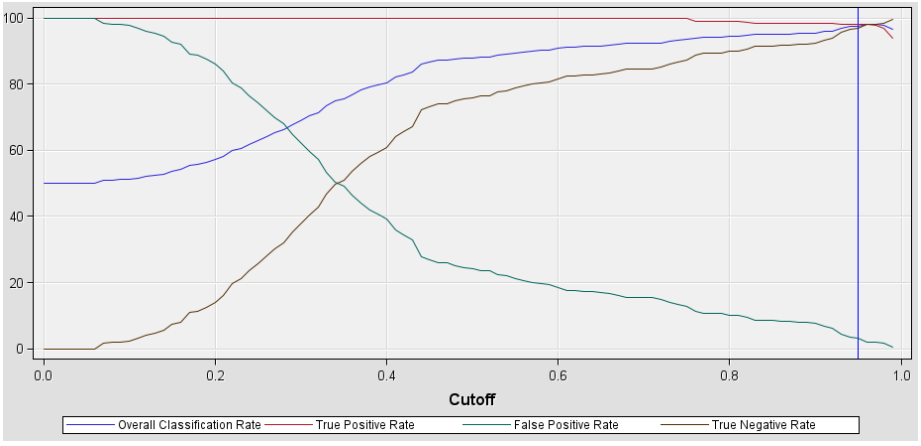

Figure 11. The overall rates of classification, TP, FP, and TN using a cutoff of 0.95 for model 14.

Index	Data Partition (train: val)	Diagram Title	Model Type	Model Settings:	Ensemble Node: Posterior Probabilities
1	80:20	1)_HP_BN_Classifier	Naïve Bayes (NB)	10 Bins	N/A
2	80:20	2)_Decision_Tree	Decision Tree (DT)	Max depth: 6 Max branch: 2 Min Category size: 5	N/A
3	80:20	3)_Neural_Network	Neural Network (NN)	No hidden units. No standardization.	N/A
4	80:20	4)_Regression	Logistic Regression	Logistic Regression: stepwise.	N/A
5	70:30	5)_ Ensemble(AVG)70	Ensemble Node Combining: • Naïve Bayes • Decision Tree • Neural Network • Regression	Same settings as each simple model above. Ensemble nodes used only posterior probabilities settings.	Average
6	70:30	6)_ Ensemble(MAX)70			Maximum
7	70:30	7)_ Ensemble(VOT_AVG)70			Voting Average
8	70:30	8)_ Ensemble(VOT_PRO)70			Voting Proportion
9	60:40	9)_ Ensemble(AVG)60	Ensemble Node Combining: • Naïve Bayes • Decision Tree • Neural Network • Regression	Same settings as each simple model above. Ensemble nodes used only posterior probabilities settings.	Average
10	60:40	10)_ Ensemble(MAX)60			Maximum
11	60:40	11)_ Ensemble(VOT_AVG)60			Voting Average
12	60:40	12)_ Ensemble(VOT_PRO)60			Voting Proportion
13	80:20	13)_ Ensemble(AVG)80	Ensemble Node Combining: 1. SVM 2. High Performance DT1 3. High Performance DT2 4. Naïve Bayes 5. Decision Tree 6. Neural Network 7. Regression 8. Bagging 9. Boosting	Same settings as each simple model above. Ensemble nodes used only posterior probabilities settings. Other model settings discussed in caption.	Average
14	80:20	14)_ Ensemble(MAX)80			Maximum
15	80:20	15)_ Ensemble(VOT_AVG)80			Voting Average
16	80:20	16)_ Ensemble(VOT_PRO)80			Voting Proportion

Table 4. Overview of predictive models shows 4 weaker models (1-4), and the 4 types of Ensemble node methods iterated 3 times: 1. a 70:30 data partition 2. a 60:40 data partition 3. an 80:20 data partition with 9 weaker models (DT2 used default decision tree settings, whereas DT1 used enhanced decision tree settings of: maximum branches of 5, maximum depth of 20, and a minimum category size of 2; NN used hidden layers and standardization; SVM used interior point method of polynomial 2).

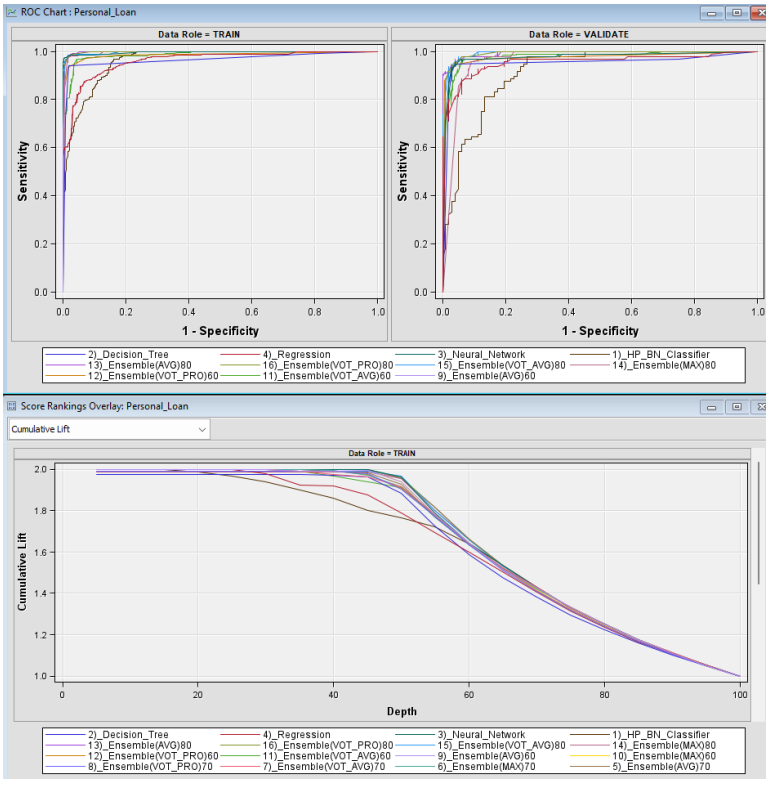

Figure 12. ROC Chart and Cumulative Lift Chart of Models 1-16.

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Train: Cumulative Lift
Y	CUT17	Ensmbl20	9)_Ensemble(AVG)60	Personal_L...		2
	CUT16	Ensmbl3	11)_Ensemble(VOT_AVG)60	Personal_L...		2
	CUT15	Ensmbl2	10)_Ensemble(MAX)60	Personal_L...		2
	CUT22	Ensmbl7	16)_Ensemble(VOT_PRO)80	Personal_L...		1.997396
	CUT20	Ensmbl8	13)_Ensemble(AVG)80	Personal_L...		1.997396
	CUT21	Ensmbl6	15)_Ensemble(VOT_AVG)80	Personal_L...		1.997396
	CUT8	Neural	3)_Neural_Network	Personal_L...		1.997396
	CUT9	Reg	4)_Regression	Personal_L...		1.997396
	CUT	HPBNC2	1)_HP_BN_Classifier	Personal_L...		1.997396
	CUT14	Ensmbl16	5)_Ensemble(AVG)70	Personal_L...		1.997024
	CUT12	Ensmbl18	7)_Ensemble(VOT_AVG)70	Personal_L...		1.997024
	CUT11	Ensmbl17	6)_Ensemble(MAX)70	Personal_L...		1.997024
	CUT18	Ensmbl4	12)_Ensemble(VOT_PRO)60	Personal_L...		1.991632
	CUT13	Ensmbl19	8)_Ensemble(VOT_PRO)70	Personal_L...		1.98984
	CUT19	Ensmbl5	14)_Ensemble(MAX)80	Personal_L...		1.984633
	CUT10	Tree10	2)_Decision_Tree	Personal_L...		1.976071

Figure 13. Fit statistics from the model comparison node using Cumulative Lift as the selection criterion.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Train: Roc Index
Y	CUT22	Ensmbl7	16)_Ensemble(VOT_PRO)80	Personal_L...		0.999
	CUT20	Ensmbl8	13)_Ensemble(AVG)80	Personal_L...		0.999
	CUT21	Ensmbl6	15)_Ensemble(VOT_AVG)80	Personal_L...		0.996
	CUT8	Neural	3)_Neural_Network	Personal_L...		0.996
	CUT19	Ensmbl5	14)_Ensemble(MAX)80	Personal_L...		0.995
	CUT15	Ensmbl2	10)_Ensemble(MAX)60	Personal_L...		0.993
	CUT14	Ensmbl16	5)_Ensemble(AVG)70	Personal_L...		0.992
	CUT17	Ensmbl20	9)_Ensemble(AVG)60	Personal_L...		0.992
	CUT12	Ensmbl18	7)_Ensemble(VOT_AVG)70	Personal_L...		0.99
	CUT11	Ensmbl17	6)_Ensemble(MAX)70	Personal_L...		0.99
	CUT16	Ensmbl3	11)_Ensemble(VOT_AVG)60	Personal_L...		0.987
	CUT18	Ensmbl4	12)_Ensemble(VOT_PRO)60	Personal_L...		0.986
	CUT13	Ensmbl19	8)_Ensemble(VOT_PRO)70	Personal_L...		0.984
	CUT10	Tree10	2)_Decision_Tree	Personal_L...		0.966
	CUT	HPBNC2	1)_HP_BN_Classifier	Personal_L...		0.962
	CUT9	Reg	4)_Regression	Personal_L...		0.961

Figure 14. 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	Ensmbl8	13)_Ensemble(AVG)80	0.03627	0.021254	0.02086	0.03500
	Ensmbl7	16)_Ensemble(VOT_PRO)80	0.04145	0.016868	0.01956	0.03344
	Ensmbl6	15)_Ensemble(VOT_AVG)80	0.04145	0.018619	0.01956	0.03395
	Ensmbl20	9)_Ensemble(AVG)60	0.04427	0.034121	0.02951	0.04380
	Neural	3)_Neural_Network	0.04663	0.014346	0.01695	0.04038
	Tree10	2)_Decision_Tree	0.04663	0.037045	0.03911	0.04456
	Ensmbl16	5)_Ensemble(AVG)70	0.04844	0.034637	0.02683	0.04251
	Ensmbl19	8)_Ensemble(VOT_PRO)70	0.07266	0.036233	0.05067	0.04196
	Ensmbl18	7)_Ensemble(VOT_AVG)70	0.07266	0.037655	0.05067	0.05691
	Ensmbl4	12)_Ensemble(VOT_PRO)60	0.08333	0.034939	0.05035	0.04785
	Ensmbl3	11)_Ensemble(VOT_AVG)60	0.08333	0.040003	0.05035	0.06153
	Reg	4)_Regression	0.09845	0.075962	0.10430	0.07440
	Ensmbl17	6)_Ensemble(MAX)70	0.12111	0.073155	0.11028	0.09067
	Ensmbl2	10)_Ensemble(MAX)60	0.15104	0.074291	0.09722	0.11715
	Ensmbl5	14)_Ensemble(MAX)80	0.15544	0.083744	0.12125	0.12116
	HPBNC2	1)_HP_BN_Classifier	0.16580	0.074944	0.10821	0.11437

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

Index	Analysis	Model Type	Model Description	Data Role	Sensitivity (TPR)	F1 Score	Accuracy	Avg. TPR	Avg. F1	Avg. Accuracy
1	A3	Naïve Bayes	1)_HP_BN_Classifier	Train	93.0%	89.6%	89.2%			
2	A3	Naïve Bayes	1)_HP_BN_Classifier	Validate	91.7%	84.6%	83.4%			
3	A3	Decision Tree	2)_Decision_Tree	Train	94.0%	96.0%	96.1%			
4	A3	Decision Tree	2)_Decision_Tree	Validate	94.8%	95.3%	95.3%	93.0%	91.4%	91.1%
5	A3	Neural Network	3)_Neural_Network	Train	97.4%	98.3%	98.3%			
6	A3	Neural Network	3)_Neural_Network	Validate	94.8%	95.3%	95.3%			
7	A3	Regression	4)_Regression	Train	89.3%	89.6%	89.6%			
8	A3	Regression	4)_Regression	Validate	92.7%	90.4%	90.2%			
25	A3	EM: Average	13)_Ensemble(AVG)80	Train	97.1%	97.9%	97.9%			
26	A3	EM: Average	13)_Ensemble(AVG)80	Validate	96.9%	95.9%	95.9%			
27	A3	EM: Max	14)_Ensemble(MAX)80	Train	100.0%	89.2%	87.9%			
28	A3	EM: Max	14)_Ensemble(MAX)80	Validate	100.0%	86.5%	89.0%			
29	A3	EM: Voting Average	15)_Ensemble(VOT_AVG)80	Train	97.4%	98.0%	96.1%	98.0%	93.3%	95.2%
30	A3	EM: Voting Average	15)_Ensemble(VOT_AVG)80	Validate	96.9%	95.4%	97.7%			
31	A3	EM: Voting Proportion	16)_Ensemble(VOT_PRO)80	Train	97.4%	98.0%	96.6%			
32	A3	EM: Voting Proportion	16)_Ensemble(VOT_PRO)80	Validate	96.9%	95.4%	98.4%			
11	A3	EM: Max	6)_Ensemble(MAX)70	Train	98.2%	89.9%	89.0%			
12	A3	EM: Max	6)_Ensemble(MAX)70	Validate	97.9%	89.0%	87.9%			
19	A3	EM: Max	10)_Ensemble(MAX)60	Train	98.6%	91.0%	90.3%	**98.0%**	87.3%	**87.3%**
20	A3	EM: Max	10)_Ensemble(MAX)60	Validate	97.4%	86.6%	84.9%			
27	A3	EM: Max	14)_Ensemble(MAX)80	Train	100.0%	89.2%	87.9%			
28	A3	EM: Max	14)_Ensemble(MAX)80	Validate	100.0%	86.5%	89.0%			
7	A2	Gradient Boosting	4)_Gradient_Boosting	Train	97.6%	97.3%	97.3%			
8	A2	Gradient Boosting	4)_Gradient_Boosting	Validate	98.6%	95.9%	95.8%			
15	A2	Gradient Boosting	8)_Gradient_Boosting	Train	98.4%	96.9%	96.9%	**98.0%**	95.6%	**95.5%**
16	A2	Gradient Boosting	8)_Gradient_Boosting	Validate	97.9%	95.4%	95.3%			
23	A2	Gradient Boosting	12)_Gradient_Boosting	Train	98.4%	96.9%	96.9%			
24	A2	Gradient Boosting	12)_Gradient_Boosting	Validate	97.9%	95.4%	95.3%			
1	A1	SVM	1)_SVM_IP_polynomial(2)	Train	95.1%	96.9%	97.0%			
2	A1	SVM	1)_SVM_IP_polynomial(2)	Validate	94.0%	94.0%	94.0%			
3	A1	SVM	2)_SVM_AS_polynomial(5)	Train	95.1%	96.9%	97.0%			
4	A1	SVM	2)_SVM_AS_polynomial(5)	Validate	94.0%	94.0%	94.0%	95.0%	94.4%	94.4%
5	A1	SVM	3)_SVM_AS_polynomial(10)	Train	97.1%	98.0%	98.0%			
6	A1	SVM	3)_SVM_AS_polynomial(10)	Validate	96.9%	95.4%	95.3%			

Table 5. Overview of the best predictive models from Assignment 1 (A1)(Fitch, 2024a), Assignment 2 (A2)(Fitch, 2024b), and Assignment 3 (this analysis) show the average TPR, F1, and Accuracy (based on validation datasets) have tradeoffs between overall accuracy and sensitivity. The ensemble max model showed perfect sensitivity at the tradeoff of lower overall accuracy while the gradient boosting showed the best values of both sensitivity and accuracy. All metrics can be expressed as percentage or decimal out of 1 (and for Table 1).