Assignment 2 - Ensemble I Models using SAS Enterprise Miner on the Universal Bank Dataset

Tanushree Kumar | tanushree.kumar@outlook.com

DATA 640 - Fall 2024

Professor Steve Knode

University of Maryland Global Campus

Due: September 14, 2024

Introduction and Data Set Description

The objective of this analysis is to develop and evaluate different ensemble models using SAS Enterprise Miner to predict whether a customer will accept a personal loan based on the customer's demographic and financial data. The goal is to determine which SVM model provides the best performance in classifying whether a customer is likely to take the loan, addressing imbalances in the dataset while exploring model variations.

The problem domain revolves around personal loan marketing in the banking sector. Banks like Universal Bank provide various financial services and often promote personal loans to existing customers. However, not all customers may be inclined to accept such offers. By analyzing customer demographics, financial information, and historical data, predictive models can be created to identify potential loan acceptors. Such models are critical for banks to target marketing efforts efficiently, increase loan acceptance rates, and reduce costs associated with customer acquisition.

The analysis focuses on predicting whether a customer will accept a personal loan offer, which involves binary classification. Given the nature of the problem—making accurate predictions in a dataset with a heavily imbalanced target variable—ensemble methods like bagging, boosting, random forest, and gradient boosting are particularly well-suited. Each of these algorithms operates on the principle of combining multiple weak learners (often decision trees) to form a strong predictive model.

The target variable is imbalanced, with a significant portion of the customers not accepting the loan offer. Single classifiers, such as logistic regression or decision trees, can struggle in this situation by favoring the majority class and delivering poor predictive performance on the minority class. Ensemble methods, particularly boosting and cost-sensitive

random forests, allow for the integration of techniques to account for this imbalance. For example, boosting focuses more on misclassified examples, which can help improve the model's ability to identify loan acceptances.

Decision trees, while powerful, can be prone to overfitting, especially in high-dimensional spaces. Bagging, also known as Bootstrap Aggregating, helps reduce variance by averaging predictions from multiple independent models. Similarly, random forests add randomness by selecting different subsets of both data and features, thereby improving the model's ability to generalize and reducing the likelihood of overfitting.

Boosting methods, such as gradient boosting, sequentially build models where each subsequent model corrects the errors of the previous ones. This iterative approach is powerful in improving accuracy, especially when the relationships between features and the target variable are complex. Since the analysis involves interactions between financial and demographic factors, boosting can capture these complexities better than simpler models.

Random forests and gradient boosting provide flexibility by tuning hyperparameters such as the number of trees, depth of trees, learning rate, and feature importance. This flexibility allows us to build models that are highly accurate while also offering insights into which features are the most important in driving the loan acceptance decision. For example, understanding whether income or education level has a higher impact on the decision-making process can be valuable for the bank in personalizing loan offers.

Ensemble methods, with their ability to aggregate multiple models and reduce errors associated with variance or bias, are a robust solution for this financial prediction problem. These methods offer the adaptability and accuracy needed to deal with complex relationships in data and imbalanced classification tasks, making them the most appropriate choice for this analysis.

The dataset is sourced from the class portal and consists of 5000 rows and 14 columns. Each row represents a customer, and each column represents a demographic or financial attribute, except for the target variable. The target variable is Personal Loan, which is binary with 1 = customer accepted the loan, 0 = customer did not accept the loan. This variable is highly imbalanced, with a large proportion of customers declining the offer, approximately 10% of customers accepted the loan, while 90% did not. The key features and their relevant details can be seen in the table below showing no missing values. The dataset also has no significant outliers which also results in minimal skew that is within an acceptable range.

Feature Name	**Feature Description**	**Missing Cases**	**Skewness**
ID	Customer ID	0	0
Age	Customer's age in completed years	0	-0.02934
Experience	Experience: Number of years of professional experience	0	-0.02632
Income	Annual income of the customer ($000)	0	0.841339
ZIPCode	Home address ZIP code	0	-
Family	Family size of the customer	0	-
CCAvg	Average spending on credit cards per month ($000)	0	1.598443
Education	Education level; 1 = Undergrad, 2 = Graduate, 3 = Advanced/Professional	0	-
Mortgage	Value of mortgage if applicable ($000)	0	2.104002
Personal Loan	Target variable; Did this customer accept the personal loan offered in the last campaign?	0	-
Securities Account	Does the customer have a securities account with the bank? (binary)	0	-
CD Account	Does the customer have a Certificate of Deposit (CD) account with the bank? (binary)	0	-
Online	Does the customer use internet banking facilities? (binary)	0	-

Credit Card	Does the customer have a credit card issued by UniversalBank? (binary)	0	-

Table 1. Table of key features and statistical values.

(Knode, 2024a)

Data Cleansing and Preparation

Based on the dataset, approximately 10% of customers accepted the loan, while 90% did not. This imbalance in the target variable will be addressed in the modeling process using techniques such as resampling, cost-sensitive learning, or cutoff adjustment. The resampling method uses oversampling or undersampling methods to balance the dataset. Cost-sensitive learning works by adjusting the cost function to penalize misclassification of the minority class. The cutoff adjustment works by altering the classification cutoff threshold from the default 0.5 value to better classify the minority class.

There were no missing values present in the dataset which meant no imputations were performed. However, if there were any missing values, the mean or the mode of the specific column would have been imputed to handle the missing data. There were also no significant outliers present, which were gathered from observing the minimum, maximum, and mean values. None of the variables were extremely skewed as they had a skewness value between the [-2,2] range, which is deemed acceptable. Thus, no transformations were required either. However, if some variables were skewed, the recommended approach would be to develop models without transforming any variables and then redevelop those models after transforming the highly skewed variables. The ‘ZIPCode’ column was removed as it wasn't deemed necessary to the models. Overall, minimal feature engineering was required since the dataset was in good condition with no errors.

A correlation matrix was developed and examined to see if there were any highly correlated features. The matrix can be seen in Figure _7 in Appendix A. It can be observed from the matrix that the 'Age' and 'Experience' variables are highly correlated with a correlation value of 0.994215. Besides that strong correlation, there is another correlation between 'Income' and 'CCAvg' with a value of 0.645984. A temporary sample model was used to assess if dropping any of the variables would give drastic results, however, no significant result changes were noticed so the variables remained in the dataset.

Development of Predictive Models

The data was partitioned into a training and validation split of 70% and 30% respectively. Starting with most of the data in the training set allows for better and more accurate models, but also increases the likelihood of overfitting. To check if any of the models are overfitting, the accuracy for the training and validation set will be observed since an overfitting model performs well on the training set but poorly on the validation set.

Twelve models were developed in total, with 3 models for each of the four model types: bagging, boosting, random forest, and gradient boosting. For all of these models, a cost function was applied to adjust for the skewed target population. This was done using the Decision Table and the Decision Matrix. The cost function essentially forces the model to find and correctly classify the rare cases (Knode, 2016b). However, a caveat of this technique is that it may result in more false positives but will also identify more true positives.

Ideally, it would be good practice to experiment with different cost values and evaluate how they affect the model's performance. The goal is to find a balance that aligns with the business goals, maximizing the detection of the minority class without overly sacrificing accuracy in the majority class. Choosing the right cost value is essential as a false negative

(rejecting a good loan) might have a higher financial impact than a false positive (approving a bad loan). Thus, the cost ratio should reflect the relative impact of these misclassifications. Thus, twelve models were made to make each of the 3 models for each model type with a different cost function value. The first four models were made with a cost function value of 2 to emphasize the impact of class imbalance. The specific value of 2 was selected to show that misclassifying a loan acceptance, positive class, as a loan rejection, negative class, should be considered significantly more costly than the reverse error.

In many cases, misclassifying a minority class, such as predicting that a customer will not accept a loan when they actually would, might have more serious consequences. For example, if the bank wrongly rejects loan acceptance, it could result in lost business opportunities. Therefore, assigning a higher cost, in this case, 2, to false negatives ensures that the model becomes more sensitive to predicting the positive class accurately.

Models 5-8 were made with a cost function value of 4 and models 9-12 were made with a cost function value of 6. By assigning a higher cost to false negatives, the models will be penalized more for missing the minority class, effectively making the models more sensitive to it. This helps to address the imbalance by increasing the recall which is the ability to predict the minority class correctly. All model types used their default settings.

Results

Below is a table that shows the summary and relevant statistics for all the models. As mentioned earlier in the data cleansing and preparation section, no imputing or transformations were performed. The table includes the model number; model name; training/validation set; imbalanced target adjustment which is the cost function; accuracy; misclassification rate; raw numbers for TP, FP, TN, and FN; sensitivity; precision; F1 score; and ROC index.

The sensitivity measurement measures how well the rare cases are being identified, in other words, how sensitive the model is to the minority case. The precision measurement helps ensure that the sensitivity is not achieved just by declaring cases to be positive. The ROC index measurement tells if the model is good enough to be used. This can be assessed by seeing the value: a value of .9 or better is an excellent model; a value of 0.8 is a pretty good model; and a value of 0.7 is subpar. The F1 value is a harmonic mean of sensitivity and precision values and can be used as a tiebreaker when models are close to other factors. Each of these measures helps to decide which, if any, of the models is best suited to be the champion model. The sensitivity, precision, and F1 values were all calculated with their respective formulas using the TP, FP, TN, and FN values. The results table can be seen below:

Model No.	Model Type	Train/Validate	Cost (C)	Accuracy	Misclassification Rate	TP	FP	TN	FN	Sensitivity (recall)	Precision	F1 Score	ROC Index
1	Bagging	Train	2	97.46	0.025443	246	0	3163	89	0.73	1.00	0.85	0.99
1	Bagging	Validate	2	97.60	0.023968	109	0	1357	36	0.75	1.00	0.86	0.99
2	Boosting	Train	2	97.63	0.023728	252	0	3163	83	0.75	1.00	0.86	1.00
2	Boosting	Validate	2	97.67	0.023302	110	0	1357	35	0.76	1.00	0.86	1.00
3	Random Forest	Train	2	99.74	0.002573	326	0	3163	9	0.97	1.00	0.99	1.00
3	Random Forest	Validate	2	98.47	0.015313	126	4	1353	19	0.87	0.97	0.92	1.00
4	Gradient Boosting	Train	2	98.23	0.017724	278	5	3158	57	0.83	0.98	0.90	0.99
4	Gradient Boosting	Validate	2	98.20	0.017976	123	5	1352	22	0.85	0.96	0.90	0.99
5	Bagging	Train	4	97.46	0.025443	246	0	3163	89	0.73	1.00	0.85	0.99
5	Bagging	Validate	4	97.60	0.023968	109	0	1357	36	0.75	1.00	0.86	0.99
6	Boosting	Train	4	97.63	0.023728	252	0	3163	83	0.75	1.00	0.86	1.00
6	Boosting	Validate	4	97.67	0.023302	110	0	1357	35	0.76	1.00	0.86	1.00
7	Random Forest	Train	4	99.74	0.002573	326	0	3163	9	0.97	1.00	0.99	1.00
7	Random Forest	Validate	4	98.47	0.015313	126	4	1353	19	0.87	0.97	0.92	1.00
8	Gradient Boosting	Train	4	98.23	0.017724	278	5	3158	57	0.83	0.98	0.90	0.99
8	Gradient Boosting	Validate	4	98.20	0.017967	123	5	1352	22	0.85	0.96	0.90	0.99
9	Bagging	Train	6	97.46	0.025443	246	0	3163	89	0.73	1.00	0.85	0.99
9	Bagging	Validate	6	97.60	0.023968	109	0	1357	36	0.75	1.00	0.86	0.99
10	Boosting	Train	6	97.63	0.023728	252	0	3163	83	0.75	1.00	0.86	1.00
10	Boosting	Validate	6	97.67	0.023302	110	0	1357	35	0.76	1.00	0.863	1.00
11	Random Forest	Train	6	99.74	0.002573	326	0	3163	9	0.97	1.00	0.99	1.00
11	Random Forest	Validate	6	98.47	0.015313	126	4	1353	19	0.87	0.97	0.92	1.00
12	Gradient Boosting	Train	6	98.23	0.017724	278	5	3158	57	0.83	0.98	0.90	0.99
12	Gradient Boosting	Validate	6	98.20	0.017976	123	5	1352	22	0.85	0.96	0.90	0.99

Table 2. Table of model comparison with all the relevant aspects.

As can be observed in the table, all the models had a ROC value of either 0.99 or 1. Figures 8, 9, and 10 in Appendix A show the ROC curves for all 12 models in order. Just by looking at the curves themselves, it can be said that all of the models performed very well. By looking at the accuracy, misclassification rates, and sensitivity values, it can be observed that overfitting did not occur for any of the 12 models since the values for all three criteria did not have drastic differences between the training and validation sets.

Based on the metrics provided for all 12 ensemble models, the champion model is determined by evaluating performance on accuracy, precision, recall (sensitivity), and the ROC Index. These metrics help balance correct predictions (true positives and true negatives) with minimizing errors (false positives and false negatives).

Model No. 4 (Random Forest with Cost = 2) emerges as the champion model due to its overall balanced performance across key metrics, particularly on the validation dataset with an accuracy of 98.47%. The other metrics of the model indicate that it offers a strong balance between sensitivity (the model correctly identifies 87% of positive cases) and precision (97% of the cases predicted as positive are positive), while also achieving the highest F1 score and near-perfect ROC Index (1.00), which shows excellent discrimination between the positive and negative classes. The low misclassification rate of 0.0153 further supports this model as the most reliable in terms of minimizing errors.

In this analysis, cost values of 2, 4, and 6 were explored. Model 4, with a cost value of 2, outperformed other models in terms of accuracy and precision, while maintaining high recall and F1 scores. Increasing the cost to 6 did not significantly improve performance as Models 9 through 12 performed at the same relative level. The cost value of 2 provided the optimal balance, as it led to minimal false positives and false negatives while ensuring that the model

could generalize well to new data. Overall, it can be said that increasing the cost level does not cause any drastic changes to the models.

Comparing the different ensemble models reveals that while boosting and bagging models also performed well, they generally had lower sensitivity and F1 scores compared to the random forest model. For example, Boosting (Model 2) achieved a similar accuracy (97.67%), but its lower sensitivity (76%) and slightly lower F1 score (0.86) made it less optimal than Model 4. The Random Forest method's ability to handle both variance and bias, combined with the use of a cost-sensitive approach, allowed it to excel in both precision and recall, making it the best choice for this classification problem.

Conclusions

The analysis aimed to develop and evaluate several ensemble models to predict whether a customer would accept a personal loan offer based on demographic and financial data. Through the use of bagging, boosting, random forest, and gradient boosting algorithms, it was determined that Model 4 (Random Forest with Cost = 2) provided the best performance in balancing accuracy, precision, recall, and overall predictive power. This model was identified as the champion model for its ability to generalize well to unseen data while effectively addressing the imbalance in the target variable.

One limitation of this analysis is the dataset itself, particularly the imbalance in the target variable. Although cost-sensitive learning was applied to adjust for the imbalance, the challenge of predicting the minority class of loan acceptances persists, and the effectiveness of the model could vary in a real-world scenario. Additionally, while ensemble methods like random forests and gradient boosting offer strong predictive power, they tend to be computationally intensive and may not scale well with larger datasets or in real-time prediction environments. Another

limitation lies in the fact that the models were tuned using default hyperparameters, and deeper hyperparameter tuning could lead to further improvements in performance. Lastly, the analysis assumes that the features provided are the most relevant for predicting loan acceptance, but there may be other latent variables not captured in the dataset that could enhance predictive performance.

Several improvements can be made in future analyses to enhance the model's accuracy and interpretability. Future work could include more extensive hyperparameter optimization for all ensemble models, particularly for the random forest and gradient boosting models, to fine-tune model parameters such as tree depth, learning rate, and the number of estimators. The inclusion of additional derived features or the incorporation of external data (e.g., credit scores, customer behavior data) could provide more context to improve the model's ability to predict loan acceptance. While cost-sensitive learning was used in this analysis, other approaches such as resampling or cutoff adjustment could be used; or more advanced undersampling techniques could be explored to balance the dataset and further improve sensitivity. Future iterations could experiment with more advanced ensemble methods or alternative algorithms such as XGBoost or deep learning-based methods, which have shown potential in handling imbalanced classification problems with complex data relationships.

References

Decision Tree Node. (n.d.). Documentation.sas.com; SAS Help Center. https://documentation.sas.com/doc/en/emref/15.1/n0cx4ud03paymdn1kargegadueml.htm

Gradient Boosting Node. (n.d.). Documentation.sas.com; SAS Help Center. https://documentation.sas.com/doc/en/emref/15.1/n0t6j7sk2xn3mon1e7ulvypjppew.htm

HP Forest Node. (n.d.). Documentation.sas.com; SAS Help Center. Retrieved September 14, 2024, from https://documentation.sas.com/doc/en/emref/15.1/p1uhmtoprigyvkn147i1tw9e2ax0.htm

Knode, S. (2016a). Adjusting for skewed target population [Vimeo]. In *Vimeo*. https://vimeo.com/186471846

Knode, S. (2016b). Cost function to adjust for skewed target population [Vimeo]. In *Vimeo*. https://vimeo.com/189827241

Knode, S. (2017a). Correlation Matrix with SAS Enterprise Miner [Vimeo]. In *Vimeo*. https://vimeo.com/233193992

Knode, S. (2017b). Ensemble Models - bagging_boosting_random forest with validation [Vimeo]. In *Vimeo*. https://vimeo.com/242743630

Knode, S. (2024a). *Universal bank description*. University of Maryland Global Campus. https://learn.umgc.edu/d2l/le/content/1226105/viewContent/33213198/View. Dataset description.

Knode, S. (2024b). *UniversalBank data*. University of Maryland Global Campus. https://learn.umgc.edu/d2l/le/content/1226105/viewContent/33213199/View. CSV file for the dataset.

Appendix A

All figures and visualizations mentioned in the report can be seen below.

Figure 1

Figure of the dataset.

Sample Table

ID	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Online	CreditCard	Personal Loan	Securities Account	CD Account
1	25	1	49	4	1.6	1	0	0	0	0	1	0
2	45	19	34	3	1.5	1	0	0	0	0	1	0
3	39	15	11	1	1	1	0	0	0	0	0	0
4	35	9	100	1	2.7	2	0	0	0	0	0	0
5	35	8	45	4	1	2	0	0	1	0	0	0
6	37	13	29	4	0.4	2	155	1	0	0	0	0
7	53	27	72	2	1.5	2	0	1	0	0	0	0
8	50	24	22	1	0.3	3	0	0	1	0	0	0
9	35	10	81	3	0.6	2	104	1	0	0	0	0
10	34	9	180	1	8.9	3	0	0	0	1	0	0

Figure 2

Figure of the dataset showing no missing values and summary statistics with no missing and high skew values.

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

Figure 3

Figure of the graph of the Chi-square values.

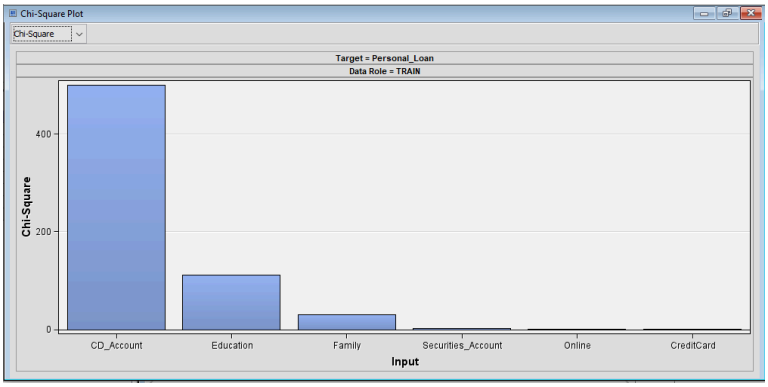

Figure 4

Figure of the graph of the variable worth.

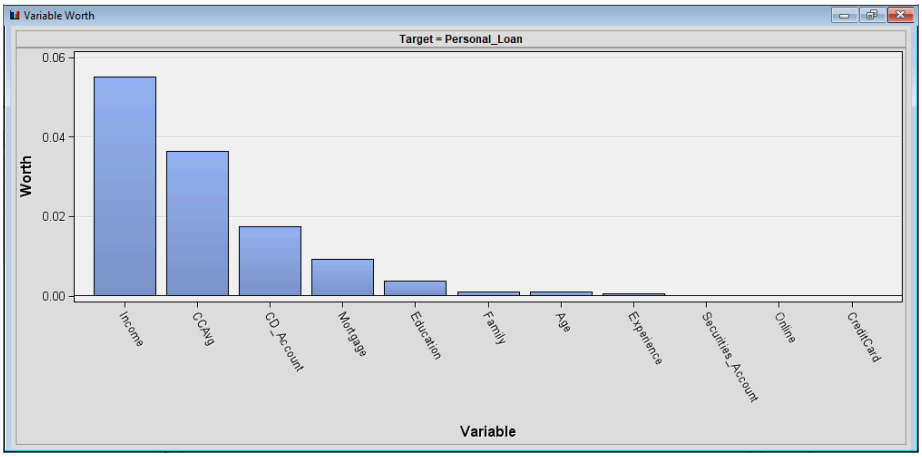

Figure 5

Figure of the cost matrix used for the cost function, the same method was used for the additional cost functions.

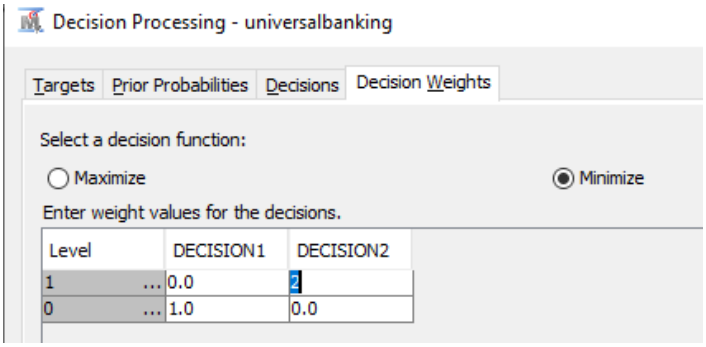

Figure 6

Figure of the Ensemble Models 1 diagram.

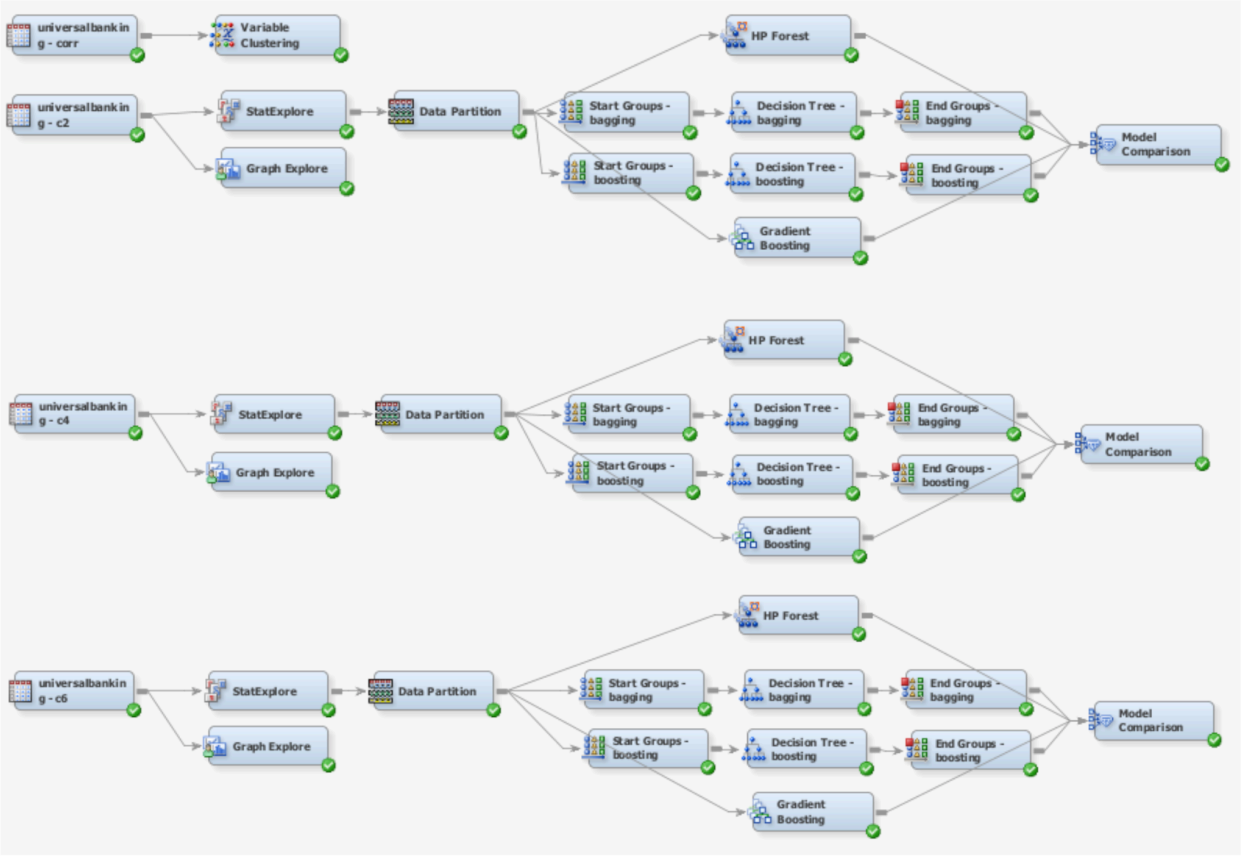

Figure 7

Figure of the correlation matrix.

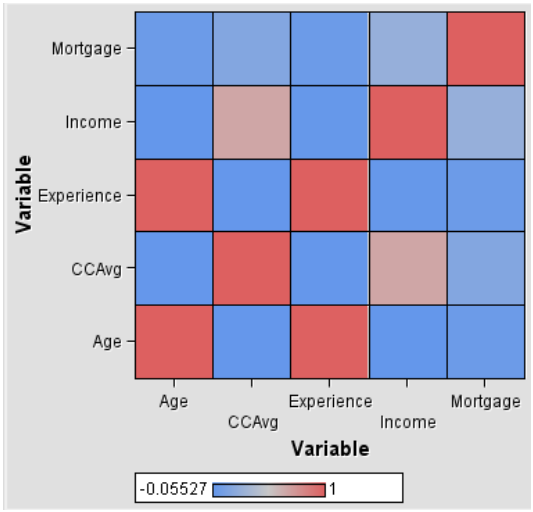

Figure 8

Figure of the ROC curve for models 1-4.

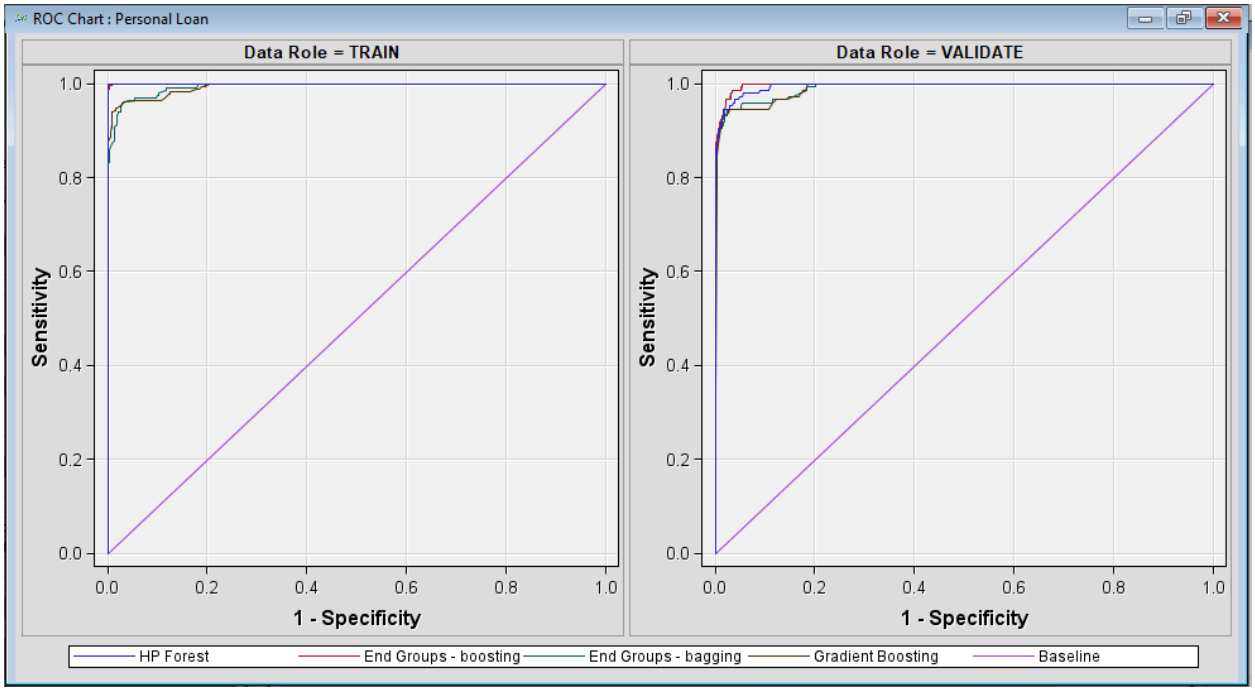

Figure 9

Figure of the ROC curve for models 5-8.

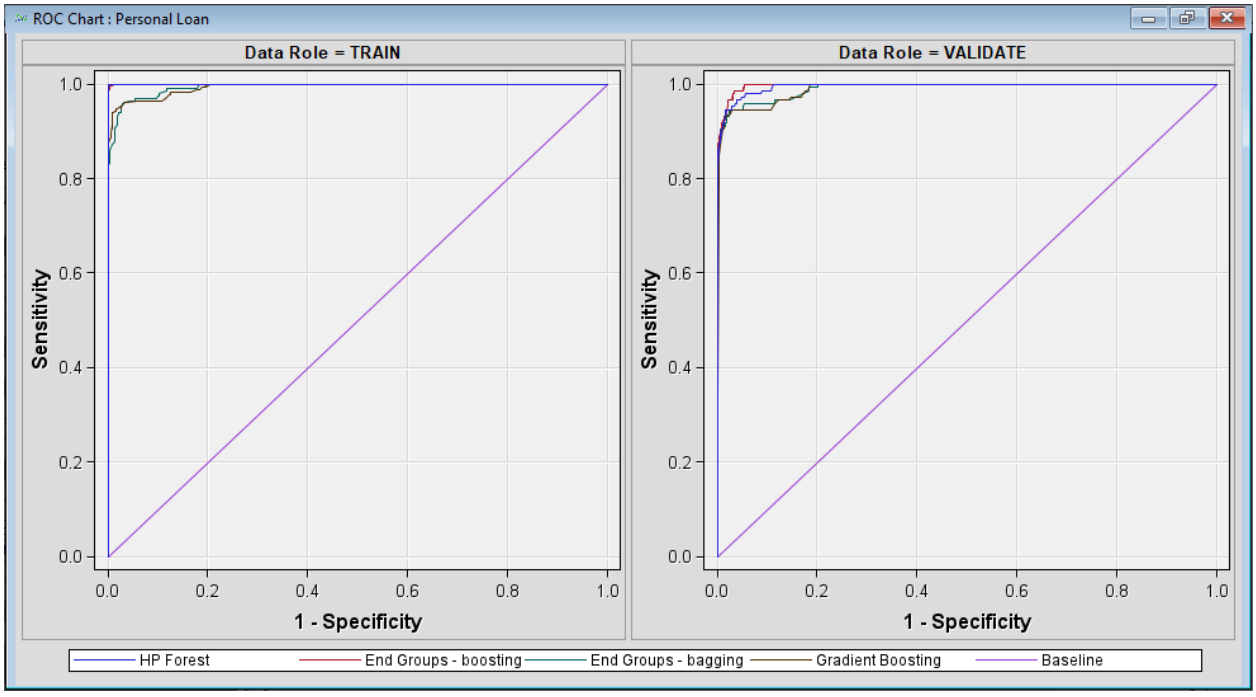

Figure 10

Figure of the ROC curve for models 9-12.