Assignment 6

AutoML 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

April 2nd, 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 AutoML. This emerging technology, AutoML, refers to a tool which automates the critical processes of machine learning including data cleaning, feature engineering, model selection, model tuning, and accuracy measure generation (Madireddy et al., 2018). The number of AutoML tools increased by 300% in just 2 years (as of 2018) indicating the sheer demand and utility of these tools. They fill a market demand for smaller businesses and/or data analysts to have advanced analytics available to them (if they don't have access to a data scientist). While these tools don't focus on business understanding nor dataset creation (since these workflows are extremely difficult to automate), they do allow automation of all functions downstream of the dataset creation (Madireddy et al., 2018). There are already countless case studies demonstrating its effectiveness, but the impressiveness of the demand is seen in the predicted growth from a total market revenue of $270 million in 2019 to $15 billion in 2030 (a 5456% increase over 11 years)(Ozsubasi, 2024). Companies are seeing incredible increases in productivity from these tools including the following examples just from DataRobot: greatly reduced deployment time of models; greatly increased internet engagement; identifying churn with greater accuracy; and using the software to increase productivity (Ozsubasi, 2024). Thus, the purpose of this study is to explore the effectiveness of one of these tools, DataRobot, and to compare the automated ML process on the Universal Bank dataset to the previous manual ML analyses performed in SAS-EM (Fitch, 2024a, 2024b, 2024c). Many models were previously developed (total of 32, including SVMs, gradient boosting, random forest, and decision trees among others) but only some of the best models and exemplar model types from each analysis will be used as comparators.

<u>Exploratory Data Analysis and Preprocessing:</u>

No new features were used for preprocessing this dataset which was different from the previous analyses (Fitch, 2024a, 2024b, 2024c). All data exploration and preprocessing can be read in the previous reports. But a high-level summary will be given using examples from DataRobot and SAS Enterprise Miner (SAS-EM). This dataset was provided by the classroom for DATA 640 and was entitled "UniversalBank data.csv". It was uploaded to SAS-EM as a .sas7bdat filetype for the previous analyses but uploaded to DataRobot as a .csv filetype for this analysis (Figure 3). It contained 5,000 rows with 14 variables (Table 1, 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 1. There are no missing entries in the whole dataset (Figure 8). All ZIP codes belonged to California (Figure 4). One value (row 386) in column E (ZIP Code) was entered incorrectly as it only contained 4 characters (Table 2). 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 5, Figure 7). 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. In the SAS-EM models, the ZIP codes were transformed from their original 5-digit state to only the first 2 digits (this was not performed for the DataRobot models). 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 6); 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. It is finally critical to note that DataRobot will automatically perform some preprocessing functions (like feature engineering, imputation of missing entries, etc.) depending on the model so any of the subsequent models discussed may have some transformations performed on them (DataRobot, n.d.a).

Models and Methods:

The models developed were chosen in order to be roughly similar to previous models (in the type chosen) to show some level of comparability. Not all hyperparameters could be tuned the same as previous models, and some model types were not available. For example, DataRobot did not offer the option of a Support Vector Machine with a polynomial kernel, nor did it offer the option of heterogeneous ensemble models. Instead, 8 models were chosen from the available model list (Table 3) and the comparable SAS-EM models are seen in Table 5. In total, three weaker models were generated (Decision Tree, Logistic Regression, and Neural Network) and 5 more complex models (Gradient Boosting, Random Forest, 2 SVMs with different kernel types, and Naïve Bayes). The weaker models are in reference to the analysis performed in Fitch (2024b) comparing heterogenous Ensemble models versus the standalone models. The 2 SVM model types were linear kernel, and Nystroem kernel. The Nystroem kernel is a model which approximates the kernel for the entire dataset using just a subset (Yeshwanth, 2023). While the polynomial kernel SVMs were the best performing models in previous analyses, these were not available to use in DataRobot. The dataset split was different across the models was inconsistent with the automated approach. This was adjusted for each model slightly to test how the split affected the validation outcomes. It was decided to set all datasets splits at 80% training, 20% test (holdout) in order to stay consistent. This was also decided as many of the previous best performing models also used 80:20 splits. The naming convention of the dataset splits is training data (made of cross-validation and validation) and test data (called holdout). DataRobot uses a method of five-fold cross-validation where 5 versions of the model are iterated and the training data is split into 5 buckets. 20% of the training dataset (16% total) is then used as 1 bucket as validation dataset while the other 4 buckets are used to train the model. This occurs in each of the 5 iterations of models such that the validation “bucket” is always moved to a new position so that all of the training data is used to test and train the model (Figure 8). Each model generated was a direct comparator from the previous analyses (Previous Model in Table 3) to see how each performs using AutoML versus a manual approach.

DataRobot has several ways it handles having heavily imbalanced target variables. For binary target variables, it uses the LogLoss metric which is robust to imbalanced data (Haviland, 2021). It penalizes misclassifications differently based on the predicted probabilities. This causes the models to become more robust as they are punished for making incorrect predictions so they self-correct instead of just choosing the imbalanced class which is 90% likely to be correct by default. DataRobot allows the possibility of down sampling the dataset so that the target variable is more balanced. But when this was attempted for this dataset for these models, no major changes were observed. The tree-based models being used are naturally resistant to the effects of imbalanced datasets, so it was decided to stay with the current configurations. One further excursion will be discussed later in this analysis. No feature settings were deemed appropriate to tweak during modeling as this dataset was relatively simple and no major changes occurred in the SAS-EM models. It was desired to keep them mostly the same with only changes deemed appropriate by the automated features. The only difference in feature approach was that the zip code variable wasn’t bucketed by the first 2 digits for the DataRobot models. This was decided on to see if the automated analyses of DataRobot may find hidden trends using the full zip codes.

Results and Model Evaluation:

As previously, the selection criteria used to determine the best models were sensitivity (TPR) and F1 score (the harmonic mean of precision and TPR)(Fitch, 2024). 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. First, the DataRobot results will be discussed independently and then comparatively. The 8 models generated all showed relatively good results only 1 showing evidence of overfitting via TPR and one via F1 (with slight decreases in the test data versus training data). All others performed better showing the models generated were fairly robust and not just overly trained to the dataset. However, only 2 models showed a TPR of >91% which represents the baseline (as the personal loan rate was 9.6%, so any model with a TPR of >90.4% outperformed random chance. All models showed excellent overall accuracy with the lowest being only 88% and the average of all others being 97.4%. This demonstrates most of these models were making accurate predictions, not sensitive ones. This shows that they were guessing the “correct” answer was majority class. The champion model proposed by DataRobot was AML1, the Gradient Boosting tree, showing a TPR of 96.9%, a 5.2% increase in TPR on the test dataset from the train data, a test F1 of 95.4% and accuracy of 99.1%. This is already a strong on its own as it is robust, it is not overfit, and it offers a 6.5% increase from the baseline. The ROC curve and prediction distribution graph show the extremely high AUC demonstrating the effectiveness of this model (Figure 9). The latter graph shows the distribution and demonstrates again the heavy imbalance of the target variable. DataRobot generates its metrics automatically by calculating a determined threshold value. This was adjusted in order to increase the effectiveness of AML1 from its calculated value of 0.1538 to 0.01. This was done in order to maximize the desired outcomes of higher TPR. It could have been adjusted to 0.00001 to gain a TPR of 100%, however this drops accuracy/F1 and other critical metrics significantly. The value of 0.01 was found to still maintain a relatively high accuracy and F1 score of 84% and 96% respectively. This demonstrates the tradeoffs between TPR and accuracy again. When it is critical to determine the rare event in heavily imbalanced datasets, the demand for TPR can be increased. In this business case, the Universal Bank would need to perform a cost analysis to determine the tradeoff between the cost of marketing (and not receiving positive loan applicants, i.e. a false positive) and the revenue a personal loan customer brings (a true positive). The cost of marketing to the 15 false positive customers (the jump from having 0FN to 1FN) being less than the revenue of 1 true positive customer who took the loan would be worth it. The metrics for this AML1 were recorded as AML1* (Table 5). This model also had a similar ROC curve with only a slight shift, and the same prediction distribution with a different threshold mark (Figure 10).

The models were each compared directly to their corollary from previous analyses (Table 5). Across the board, the previous models performed much better on TPR with only AML1* performing 1% better, and all others performing worse (range of 5.3-23.0%). It should be noted that AML1 was only 2.1% worse before the modifications were made to increase TPR to 100%. That adjustment could be made to each of these models but for the matter of comparison, all models were kept at the baseline state since major adjustments were not made to the previous analyses to increase TPR. It should be noted these AutoML models did better in overall accuracy and in making more robust models. Almost every model had higher general accuracy (where they had higher false negative rates). The DataRobot models also all performed better on the test datasets meaning they were more robust. The SAS-EM models all performed worse than the DataRobot models, and worse on the test data. It is clear from these results, the DataRobot models have a much better potential. While overall accuracy may slightly decrease, this would be worth it in order to have a higher TPR and gain new customers. Most critically, the AutoML models are most robust and are likely to perform better on future data. As a final note, DataRobot offers many, many other features in order to understand the dataset, understand the models generated, and tune them for the desired outcomes. Of these abilities, 2 features should be highlighted. First, the hotspot visualization allows one to see how the data relates to itself, and the rules associated with rules-based models (Figure 11). This makes it easy to find trends and see how data is clustered in order to tailor marketing (or other business applications). For example, just from the first several rules, it can be determined marketing for the bank's personal loan should focus on individuals with incomes >$113k and families of >2.5. This paired with a few other rules would greatly reduce the marketing pool the bank should solicit saving large costs and efforts. Next, the speed vs accuracy graph demonstrates the tradeoffs made between performance and time (Figure 12). This is critical when preparing models for deployment into production environments. A model with extremely high accuracy but high testing times usually won't be usable, and a model with lower accuracy and high test times isn't worth using.

Conclusion, Limitations, and Improvements:

In conclusion, 8 AutoML predictive models were generated (and one modified) each with a direct comparator from previous analyses using SAS-EM. Of those models, all performed worse in TPR but better in overall accuracy and were more robust. The best performer AML1 was modified easily to AML1* to increase the importance of TPR (to identify more customers). This modification could easily be applied to all other models which would slightly decrease accuracy but increase TPR. The AutoML models were more efficient, more accurate, more tunable, more scalable, and more accessible. While the SAS-EM models performed natively in terms of TPR, the DataRobot models have a higher ceiling since minor tuning can greatly increase the TPR while maintaining acceptable accuracy/F1 levels. DataRobot offers accessible ways to understand the data and automate difficult tasks like feature engineering. This dataset was relatively simple and didn't require any data transformation, nor have missing values. However, the automated workflow for messy data with missing entries, and automated feature engineering is invaluable. Especially since there are countless needs when it comes to business applications, having a high level of flexibility in this software is critical. The DataRobot approach is much better than the SAS-EM approach for beginners, and for experts due to the configurability, automation, and robustness. SAS-EM may be the better approach when having a specific model design in mind ahead of time, if needing to choose a particular model design due to black-box regulations, or if particular features are needed due to business practices. Further analyses of this dataset must include modifying each of the 8 models to have a lower threshold (and thus increased TPR). Each of these models' results could then be further compared to the original SAS-EM models. A messier dataset should also be used in DataRobot, with missing entries, more variables, and potential for feature engineering. This would allow the opportunity to showcase its ability to automate model generation even with difficult data. This analysis was time-limited so further analysis could also include using DataRobot to generate models for a messier dataset, then remaking those models in SAS-EM with the ability to tweak them more thoroughly. A last future analysis should include using a model from SAS-EM and DataRobot in production to test how effective each one is (in terms of accuracy, TPR, and speed) as this is the ultimate test to determine how well the model is working. Based on this report, the DataRobot model would likely be easier to develop, have a higher TPR, and be more robust.

References:

DataRobot, (n.d.a). Data. DataRobot. https://docs.datarobot.com/en/docs/data/index.html

DataRobot (n.d.b). Training Sets, Validation Sets, and Holdout Sets. DataRobot. https://www.datarobot.com/wiki/training-validation-holdout/

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

Fitch, T. (2024c). Ensemble Modeling 2 on Universal Bank Dataset. GitHub. https://github.com/Capadetated/Ensemble-Modeling-2-on-Universal-Bank-Dataset/blob/main/Assignment3_Fitch.pdf

Haviland, L. (2021). How to Tackle Imbalanced Data with DataRobot. DataRobot. https://www.datarobot.com/blog/how-to-tackle-imbalanced-data-with-datarobot/

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

Madireddy, M., Ferruzza, G., & Sharma, A. (2018) An Overview of AutoML and Available Technology. Medium. https://medium.com/an-overview-of-automl-and-available-technology/an-overview-of-automl-and-available-technology-a6d3e4650c7c

Ozsubasi, I. (2024). 27 AutoML Statistics: Market Size, Adoption & Benefits. AI Multiple. https://research.aimultiple.com/automl-stats/

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

Yeshwanth, N. (2023). Nystrom Method: Empowering Large-Scale Kernel Machines. https://www.linkedin.com/pulse/nystrom-method-empowering-large-scale-kernel-machines-yeshwanth-n-1hr9f/

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 1. 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 2. All variables and the dataset variable statistics show no missing values and no major skew.

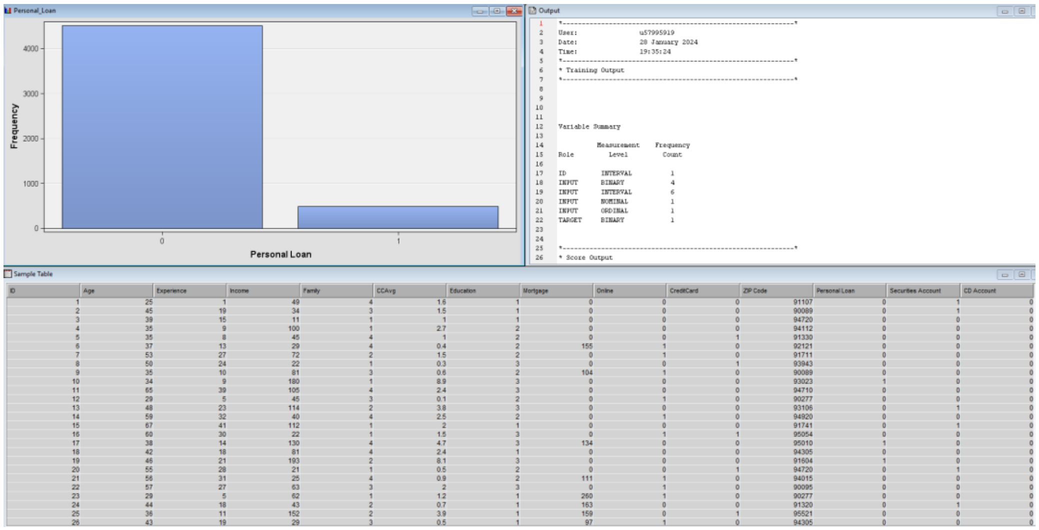

Figure 1. Graph from SAS Enterprise Miner showing the imbalance of the Personal Loan variable (left), the Variable Summary Table (right), and example data from UniversalBank dataset (bottom).

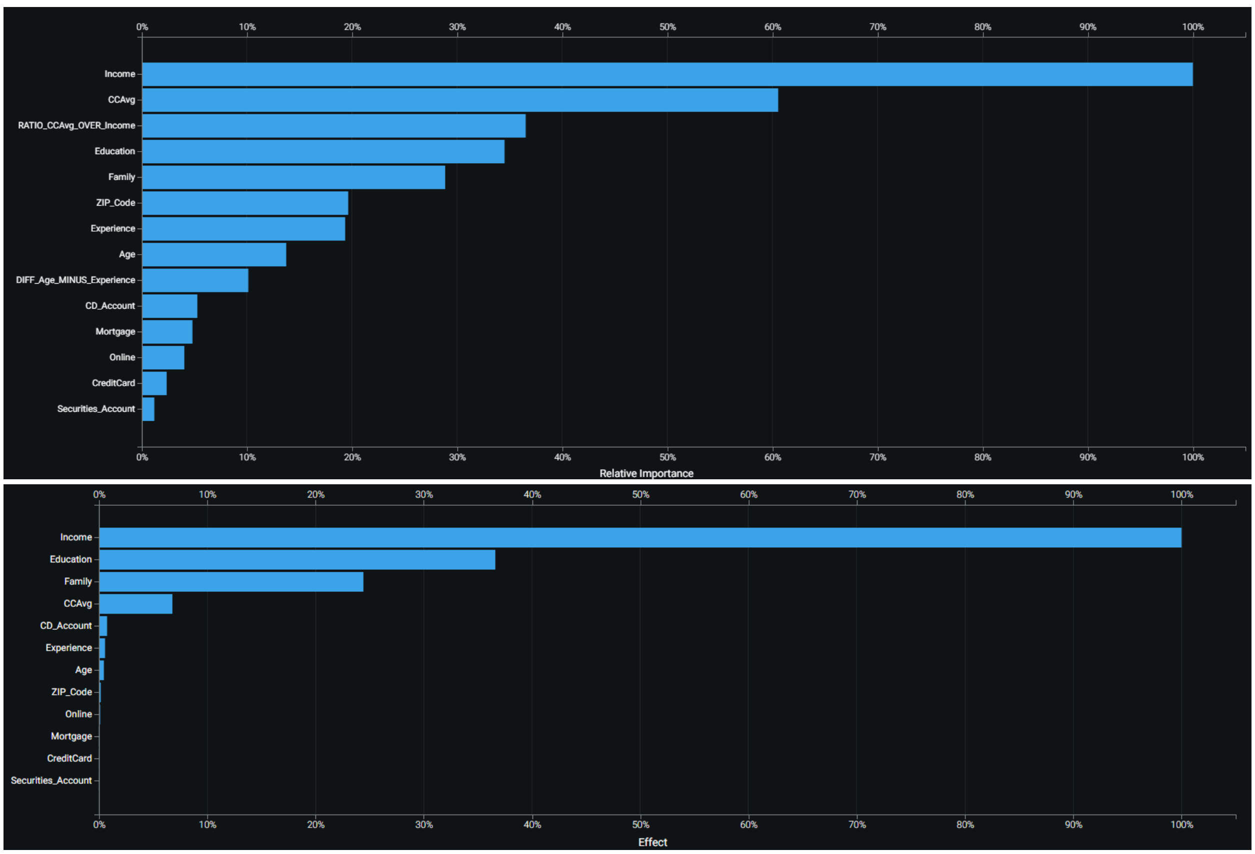

Figure 2. Graph from DataRobot showing the relative importance of each variable for the Gradient Boosting model (AML1) for the first 2,000 entries (top) versus all entries with feature engineering (bottom).

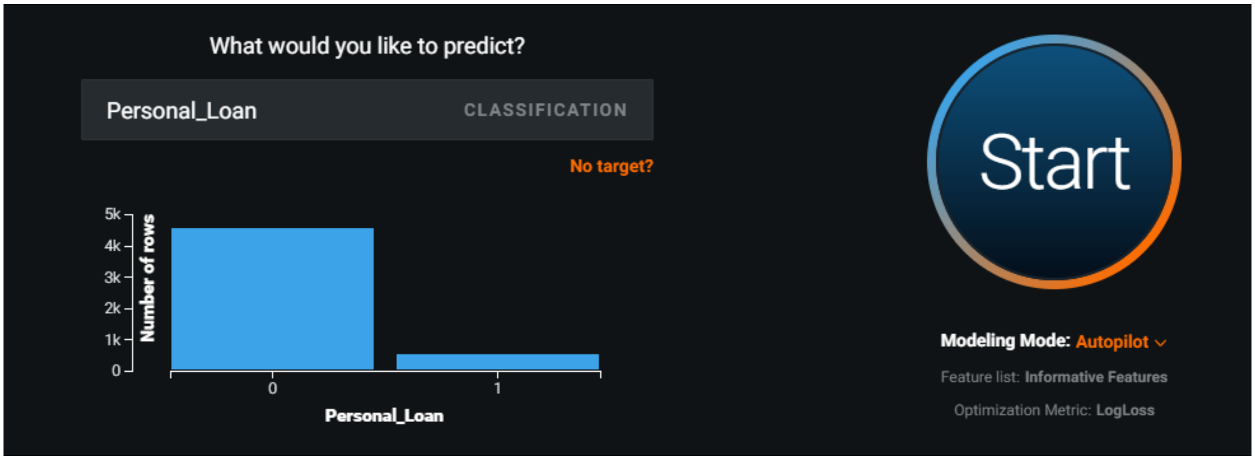

Figure 3. Graph from DataRobot showing the imbalance of the Personal Loan variable (left) and the function to begin automatically generating models using the Autopilot dropdown and using the LogLoss optimization metric.

Index	**Model Type**	**Model Description**	**DataRobot Model ID**	**Data Split**	**Previous Model**
AML1	Gradient Boosting	Gradient Boosted Greedy Trees Classifier with Early Stopping	M108	Tr: 80 T: 20	A2 4, 8, 12
AML2	Random Forest	RandomForest Classifier (Entropy)	M59	Tr: 80 T: 20	A2 5, 9, 12
AML3	Decision Tree	Decision Tree Classifier (Gini)	M55	Tr: 80 T: 20	A3 2
AML4	SVM	Nystroem Kernel SVM Classifier M22	M22	Tr: 80 T: 20	A1 10, 11, 12
AML5	Logistic Regression	Regularized Logistic Regression (L2)	M52	Tr: 80 T: 20	A3 4
AML6	SVM	Support Vector Classifier (Linear Kernel)	M113	Tr: 80 T: 20	A1 10, 11, 12
AML7	NN	Keras Slim Residual Neural Network Classifier using Training Schedule (1 Layer: 64 Units)	M48	Tr: 80 T: 20	A3 3
AML8	Naïve Bayes	Naïve Bayes Combined Classifier	M250	Tr: 80 T: 20	A3 1

Table 3. The 8 DataRobot models developed and all relevant parameters. The dataset splits were shortened to Tr for Training (Cross-Validation) and T for Test (Holdout) datasets. The previous models were shortened to A for Assignment, followed by the model number for that assignment. Where there were multiple models, the best performing model was highlighted in red and metrics only for that model will be used for comparison. Metrics and full model names can be seen in Table 5.

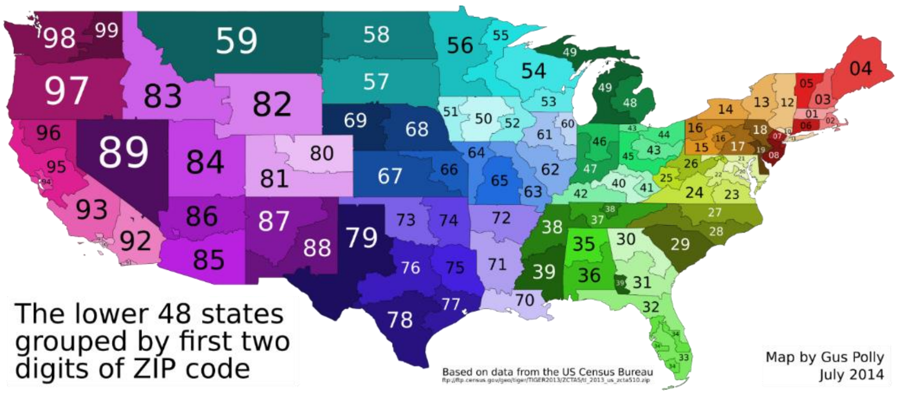

Figure 4. Map showing the lower 48 United States grouped by first 2 digits of ZIP code (All 90-96 ZIP codes can be seen in California)(Polly, 2014).

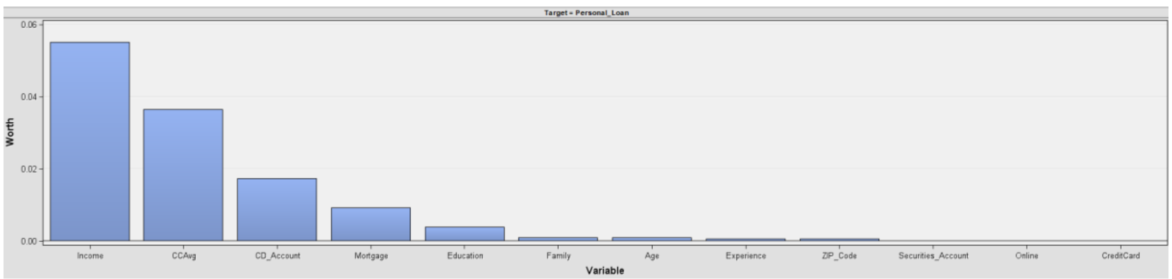

Figure 5. Variable worth for each variable in the UniversalBank dataset.

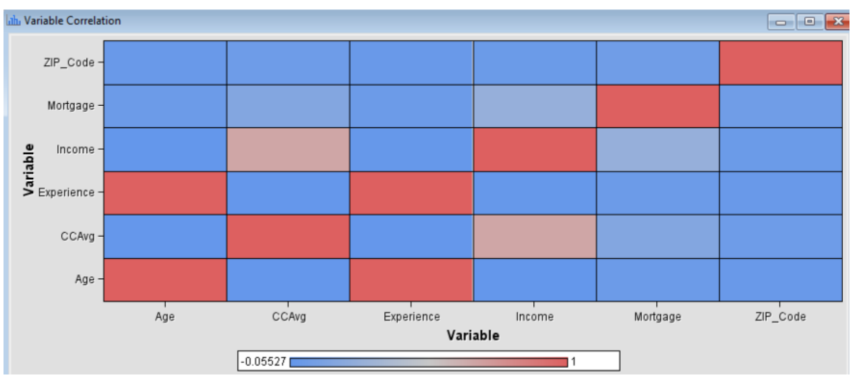

Figure 6. Variable correlation matrix for each variable in the UniversalBank dataset.

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

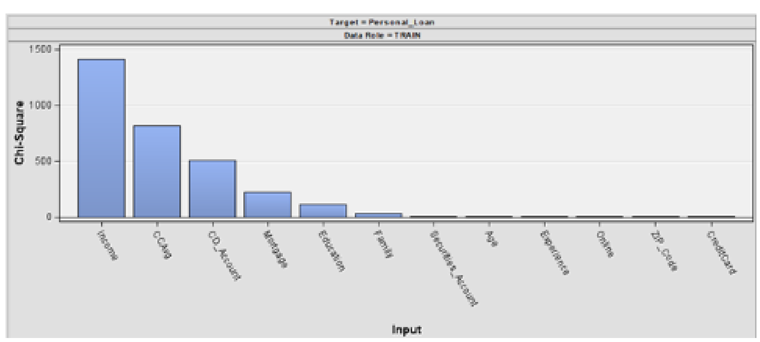

Figure 7. Chi-Square values for each variable in the UniversalBank dataset (in chart and table form).

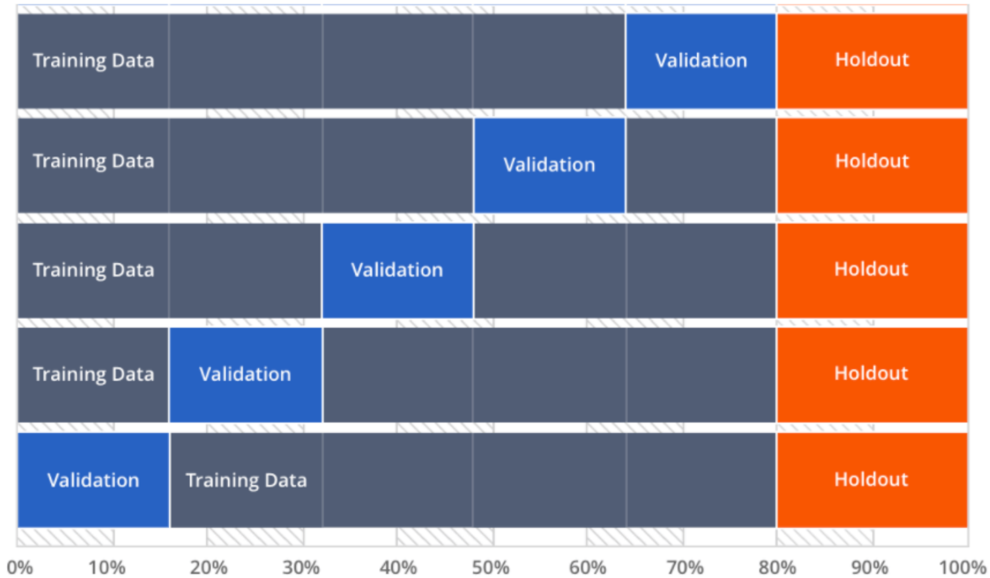

Figure 8. The five-fold cross-validation technique used by DataRobot (n.d.b).

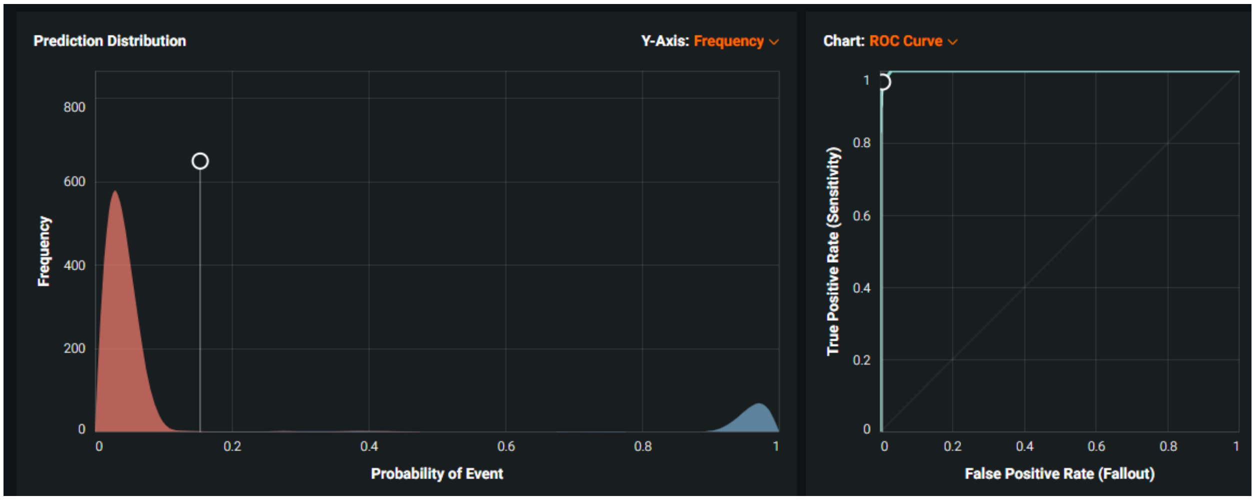

Figure 9. The prediction distribution graph for AML1, and the ROC curve with extremely high AUC.

Figure 10. The prediction distribution graph for AML1*, and the ROC curve with slightly improved AUC value, higher TPR, and slightly lowered accuracy/F1.

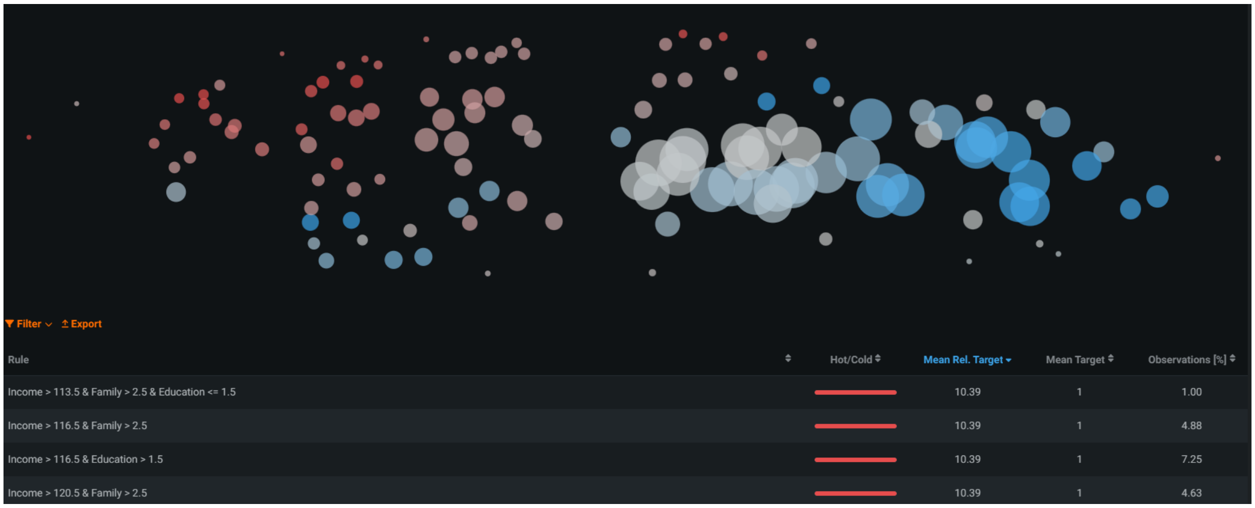

Figure 11. The hotspot visualization shows some of the decision tree rules where data is distributed.

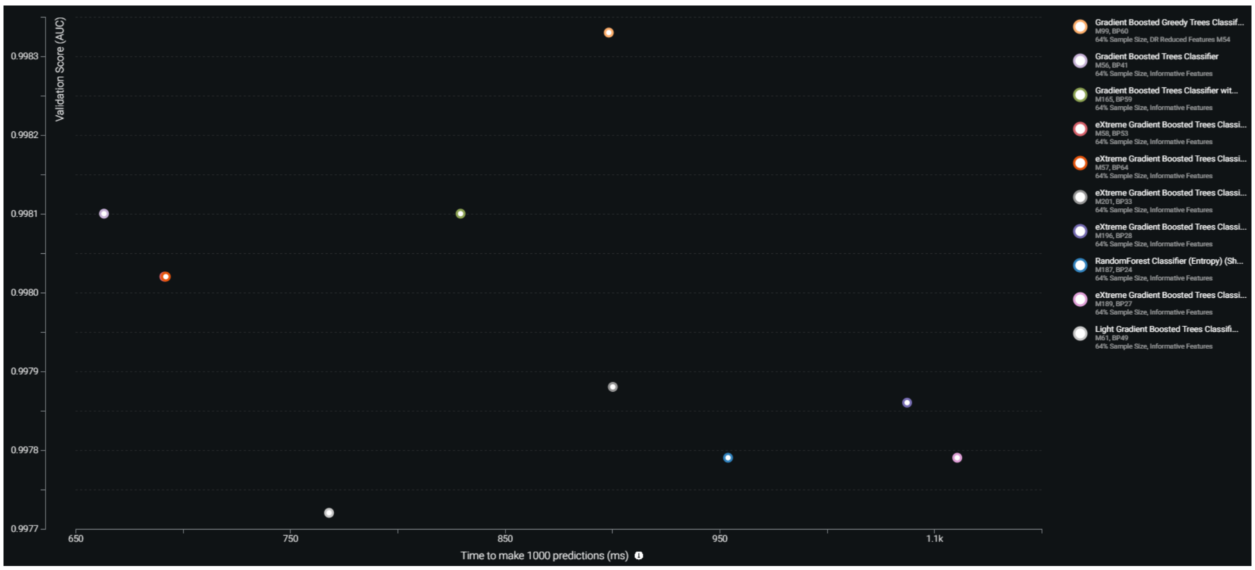

Figure 12. The speed vs accuracy visualization shows how each model has tradeoffs between accuracy performance and time to execute.

Index	Model Type	Model Description	Dataset	FN	TN	FP	TP	TPR	Overfitting TPR Check	Specificity	Precision	F1 Score	Overfitting F1 Check	AUC	Misclassification Rate	Accuracy
AML1	Gradient Boosting	Gradient Boosted Greedy Trees Classifier with Early Stopping	Training	32	3596	20	352	91.7%	5.2%	99.4%	94.6%	93.1%	2.3%	98.7%	1.3%	98.7%
			Test	3	898	6	93	96.9%		99.3%	93.9%	95.4%		99.9%	0.9%	99.1%
AML2	Random Forest	RandomForest Classifier (Entropy)	Training	59	3613	3	325	84.6%	7.0%	99.9%	99.1%	91.3%	3.3%	99.7%	1.6%	98.5%
			Test	8	902	2	88	91.7%		99.8%	97.8%	94.6%		99.7%	1.0%	99.0%
AML3	Decision Tree	Decision Tree Classifier (Gini)	Training	64	3608	8	320	83.3%	3.1%	99.8%	97.6%	89.9%	2.8%	98.8%	1.8%	98.2%
			Test	13	904	0	83	86.5%		100.0%	100.0%	92.7%		99.7%	1.3%	98.7%
AML4	SVM	Nystroem Kernel SVM Classifier M22	Training	65	3545	71	319	83.1%	3.4%	98.0%	81.8%	82.4%	3.6%	97.8%	3.4%	96.6%
			Test	13	890	14	83	86.5%		98.5%	85.6%	86.0%		98.6%	2.7%	97.3%
AML5	Logistic Regression	Regularized Logistic Regression (L2)	Training	81	3577	39	303	78.9%	5.5%	98.9%	88.6%	83.5%	1.3%	96.8%	3.0%	97.0%
			Test	15	890	14	81	84.4%		98.5%	85.3%	84.8%		98.3%	2.9%	97.1%
AML6	SVM	Support Vector Classifier (Linear Kernel)	Training	104	3561	55	280	72.9%	1.0%	98.5%	83.6%	77.9%	-0.7%	96.9%	4.0%	96.0%
			Test	25	887	17	71	74.0%		98.1%	80.7%	77.2%		97.4%	4.2%	95.8%
AML7	NN	Keras Slim Residual Neural Network Classifier using Training Schedule (1 Layer: 64 Units)	Training	71	3502	114	313	81.5%	-4.4%	96.8%	73.3%	77.2%	2.0%	96.5%	4.6%	95.4%
			Test	22	887	17	74	77.1%		98.1%	81.3%	79.1%		96.6%	3.9%	96.1%
AML8	Naïve Bayes	Naïve Bayes Combined Classifier	Training	101	3235	381	283	73.7%	1.3%	89.5%	42.6%	54.0%	2.0%	92.0%	12.1%	88.0%
			Test	24	815	89	72	75.0%		90.2%	44.7%	56.0%		93.0%	11.3%	88.7%

Table 4. Overview of predictive models shows 8 total models: 3 weaker standalone models (AML3, 5, and 7). The metrics shown in order are: the confusion matrix; the sensitivity/TPR; an overfitting check of test TPR minus train TPR; specificity; precision; F1 score; test F1 minus train F1 to check for overfitting; AUC; misclassification; and accuracy.

Index	Model Type	Model Description	Dataset	TPR	Overfitting TPR Check	F1 Score	Overfitting F1 Check	Accuracy	AutoML - Manual TPR
AML1	Gradient Boosting	Gradient Boosted Greedy Trees Classifier with Early Stopping	Training	91.7%	5.2%	93.1%	2.3%	98.7%	1.0%
			Test	96.9%		95.4%		99.1%	
AML1*	Gradient Boosting	Gradient Boosted Greedy Trees Classifier with Early Stopping. Modified threshold: 0.01	Training	99.0%	1.0%	80.7%	3.5%	95.5%	
			Test	100.0%		84.2%		96.4%	
A2: 4	Gradient Boosting	Gradient Boosting Model Default Settings	Training	98.0%	1.0%	97.0%	-1.0%	97.3%	
			Test	99.0%		96.0%		95.8%	
AML2	Random Forest	RandomForest Classifier (Entropy)	Training	84.6%	7.0%	91.3%	3.3%	98.5%	-5.3%
			Test	91.7%		94.6%		99.0%	
A2: 5	Random Forest	Random Forest Model Default Settings	Training	100.0%	-3.0%	100.0%	-4.0%	99.9%	
			Test	97.0%		96.0%		95.5%	
AML3	Decision Tree	Decision Tree Classifier (Gini)	Training	83.3%	3.1%	89.9%	2.8%	98.2%	-8.5%
			Test	86.5%		92.7%		98.7%	
A3: 2	Decision Tree	Decision Tree Model Default Settings	Training	94.0%	1.0%	96.0%	-1.0%	96.1%	
			Test	95.0%		95.0%		95.3%	
AML4	SVM	Nystroem Kernel SVM Classifier M22	Training	83.1%	3.4%	82.4%	3.6%	96.6%	-10.5%
			Test	86.5%		86.0%		97.3%	
A1: 10	SVM	SVM Active Set Polynomial	Training	97.0%	0.0%	98.0%	-3.0%	98.0%	
			Test	97.0%		95.0%		95.3%	
AML5	Logistic Regression	Regularized Logistic Regression (L2)	Training	78.9%	5.5%	83.5%	1.3%	97.0%	-8.6%
			Test	84.4%		84.8%		97.1%	
A3: 4	Logistic Regression	Logistic Regression Stepwise	Training	89.0%	4.0%	90.0%	0.0%	89.6%	
			Test	93.0%		90.0%		90.2%	
AML6	SVM	Support Vector Classifier (Linear Kernel)	Training	72.9%	1.0%	77.9%	-0.7%	96.0%	-23.0%
			Test	74.0%		77.2%		95.8%	
A1: 10	SVM	SVM Active Set Polynomial	Training	97.0%	0.0%	98.0%	-3.0%	98.0%	
			Test	97.0%		95.0%		95.3%	
AML7	NN	Keras Slim Residual Neural Network Classifier using Training Schedule (1 Layer: 64 Units)	Training	81.5%	-4.4%	77.2%	2.0%	95.4%	-17.9%
			Test	77.1%		79.1%		96.1%	
A3: 3	NN	Neural Network with no Hidden Units, no standardization	Training	97.0%	-2.0%	98.0%	-3.0%	98.3%	
			Test	95.0%		95.0%		95.3%	
AML8	Naïve Bayes	Naïve Bayes Combined Classifier	Training	73.7%	1.3%	54.0%	2.0%	88.0%	-17.0%
			Test	75.0%		56.0%		88.7%	
A3: 1	Naïve Bayes	Naïve Bayes Model with 10 Bins	Training	93.0%	-1.0%	90.0%	-5.0%	89.2%	
			Test	92.0%		85.0%		83.4%	

Table 5. Overview of the AutoML predictive models compared against those from Assignments 1-3 (A1, A2, A3)(Fitch, 2024). The best overall model previously was A2:4, a gradient boosting model with a 99% TPR, 96% F1 whereas the best DataRobot model was AML1* with a 100% TPR, 84% F1. The DataRobot models were slightly to much worse in TPR from every SAS-EM model except for the AML1* with 100% TPR.