Credit Card Default Model

Model Development Guide

Jonah Muniz

1.0 Introduction

The objective of this research paper is to understand which of the four models, Random Forest, Extra Gradient Boost (XGBoost), Logistic Regression, and Naiive Bayes perform the best when predicting if a customer will default on their credit. The Taiwan customer dataset is used throughout the paper. This report is structured as follows: Section 2.1 describes the data, Section 2.2 outlines data quality checks, Section 3.0 details feature engineering, Section 4.0 describes exploratory data analysis, Section 5.0 covers methods and results of the models, and Section 6.0 compares model results. The comparison of model performance will determine which model predicts customer credit default the best, thus meeting the objective of the paper.

2.0 The Data

2.1 Data Description

The data was acquired from a group of customer credit payments in Taiwan with a feature that indicates if the customer defaulted. The dataset was made publicly available January 26, 2016. The dataset contains 25 attributes and 30,000 observations and spans 6 months. The data dictionary table below outlines the 25 attributes incorporated in the dataset and their definitions.

Table 1: Data Dictionary

Dataset Attribute	Definition
ID	Observation Identifier
LIMIT_BAL	Observation's Credit Limit
SEX	Gender of the Observation: 1 = male, 2 = female
EDUCATION	Highest level of education completed: 1 = graduate school; 2 = university; 3 = high school; 4 = others
MARRIAGE	Marital status: 1 = married; 2 = single; 3 = others
AGE	Age of observation in years
PAY_0	Repayment status of month 1
PAY_2	Repayment status of month 2
PAY_3	Repayment status of month 3
PAY_4	Repayment status of month 4
PAY_5	Repayment status of month 5
PAY_6	Repayment status of month 6
BILL_AMT1	Bill amount after month 1
BILL_AMT2	Bill amount after month 2
BILL_AMT3	Bill amount after month 3
BILL_AMT4	Bill amount after month 4
BILL_AMT5	Bill amount after month 5
BILL_AMT6	Bill amount after month 6
PAY_AMT1	Payment amount affter month 1
PAY_AMT2	Payment amount affter month 2
PAY_AMT3	Payment amount affter month 3
PAY_AMT4	Payment amount affter month 4
PAY_AMT5	Payment amount affter month 5
PAY_AMT6	Payment amount affter month 6
Default	Binary variable indicating if the observation defualted

The next step after defining the dataset and its attributions is to perform an overall data quality check of the 30,000 observations.

2.2 Data Quality Checks

The first step in the data quality check is to confirm that the data found in the dataset reflects values consistent with the data dictionary definitions above. One way to perform this check is to run a numeric diagnostic leveraging the diagnose function within the dlookr package in R. The diagnostic revealed the min, quartile 1, mean, median, quartile 3, max, zero, min, and outliers. The diagnostic flagged several potential data quality issues.

For instance, the MARRIAGE feature should have three values: 1, 2, and 3. There were 54 instances in the dataset where MARRIAGE = 0. Instead of removing these customers from the dataset, the MARRIAGE value will be replaced with 3 which is other. The diagnostic also revealed that EDUCATION contained 14 instances of 0, which should not be possible based on the above data dictionary. All customers who have an EDUCATION value of 0 will be replaced with 4 which is other. Summary statistics for EDUCATION revealed instances that are greater than 4, which is not possible based on the above data dictionary. All instances where EDUCATION was greater than 4 were replaced and reassigned to 4 which is other. All EDUCATION values now fall within 1 and 4, which follows the data dictionary definition for this feature.

Repayment features also contained potential data quality issues. Although the Repayment features should have values ranging from -1-6, there were values of -2. These values do not meet the criteria for the Repayment features based on the data dictionary above. All -2 values are assumed to be the result of mislabeling. Instead of removing the samples from the dataset, all -2 repayment data points were replaced with -1. After correcting data quality issues, engineered features can be created.

3.0 Feature Engineering

There are additional features outside of the provided features in the data set that can be engineered to further capture customer's behavior. Approximately 45 additional features were engineered to enhance the dataset and capture customer behavior. Definitions and calculations for each feature that was engineered are below.

3.1 Demographic Feature Creation

The first engineered feature was binning of the continuous feature AGE to three distinct bins: Age_18_25, Age_26_40, and Age_41_100. For the Age_18_25 feature, if a customer's age fell within 18-25, they were assigned 1; if not, they were assigned 0. For the Age_26_40 feature, if a customer's age fell within 26-40, they were assigned 1; if not, they were assigned 0. Lastly, for the Age_41_100 feature, if a customer's age fell within 41-100, they were assigned 1; if not, they were assigned 0.

3.2 Payment Feature Creation

The average bill amount was then calculated to understand on average how much each customer has been charged in the last 6 months. The average across BILL_AMT1 – BILL_AMT6 was calculated to determine the Avg_Bill_Amt. The average pay amount was also calculated over the 6 months. The average across PAY_AMT1 – PAY_AMT6 was calculated to determine the Avg_Pay_Amt.

Payment ratio was then calculated. Payment ratio is the portion of the bill amount that the customer pays. The bill amount and pay amount lag one another so the first payment ratio considers BILL_AMT2 and PAY_AMT1. Pay_Ratio_1 is calculated by dividing PAY_AMT1 by BILL_AMT2. Pay_Ratio_2, 3, 4, and 5 follow a similar process. Average payment ratio (Avg_Pay_Ratio) is the average across Pay_Ratio_1 – Pay_Ratio_5.

Next, monthly utilization was calculated. Utilization is defined as the amount of the credit limit that is used. This is calculated by dividing BILL_AMT by LIMIT_BAL. Utilization_1 – Utilization_6 was calculated using the above equation. Average utilization, Avg_Utilization, was also calculated by taking the average across Utilization_1 – Utilization_6 for each customer. Utilization growth over 6 months was calculated for each customer to see if utilization has increased over the 6 months. The last utilization, Utilization_6, was subtracted by the original utilization, Utilization_1, to calculate the utilization growth over the 6 months.

Balance growth over the 6 months was calculated. This was calculated by taking the difference between the starting balance, BILL_AMT1, and the last balance, BILL_AMT6. Max bill amount was calculated by taking the maximum value across BILL_AMT1 – BILL_AMT6. Max payment amount was calculated by taking the maximum value across PAY_AMT1 – PAY_AMT6. Lastly, maximum delinquency was calculated by calculating the maximum value across PAY_1 – PAY_6.

3.3 Additional Feature Creation

In addition to the above features, there were several additional features engineered. Max utilization was calculated by calculating the maximum value across Utilization_1 – Utilization_6. Customers were further segmented using PAY_AMT to create three separate income bins: Low_Income, Medium_Income, and High_Income. Low_income customers are labeled 1 if their Max_Pay_Amt was greater or equal to 0 but less than or equal to $2,198. If the Max_Pay_Amt is out of this range, Low_income is 0. If a customer's Max_Pay_Amt is greater than or equal to $2,199 and less than or equal to $121,000 they were labeled 1, else 0 for the Medium_Income feature. Lastly, if a customer's Max_Pay_Amt is greater than or equal to $121,101 or less than or equal to $1,684,260 they were labeled 1, else 0 for the High_Income feature.

In summary, an additional 45 features were engineered in an effort to increase the amount of metrics available describing a customer's credit usage to better predict default. The next step after ensuring proper data quality and feature engineering is to perform exploratory analysis to better understand the data.

4.0 Exploratory Data Analysis

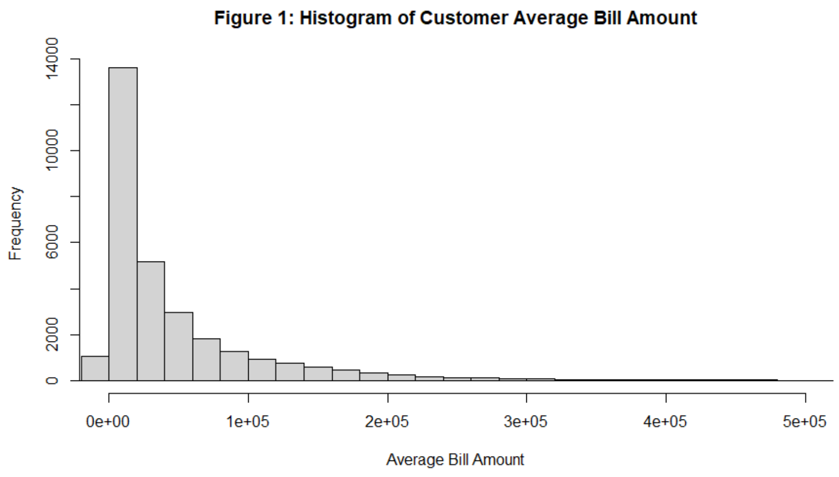
4.1 Traditional EDA

A summary table was created for the engineered features to act as a data quality check. Below is Table 2: Summary Statistics for Engineered Features.

Table 2: Summary Statistics for Engineered Features

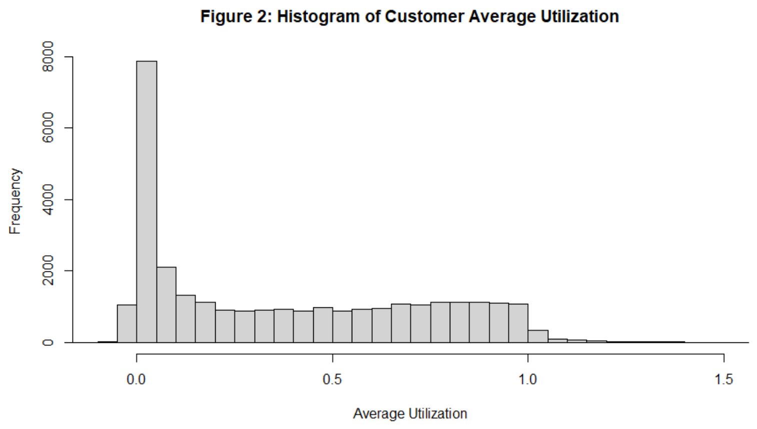
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
AGE_18_25	30,000	0.13	0.34	0	0	0	0	1
AGE_26_40	30,000	0.60	0.49	0	0	1	1	1
AGE_41_100	30,000	0.28	0.45	0	0	0	1	1
Avg_Bill_Amt	30,000	44,976.95	63,260.72	-56,043	4,781.3	21,051.8	57,104.4	877,314
Avg_Pay_Amt	30,000	5,275.23	10,137.95	0.00	1,113.29	2,397.17	5,583.92	627,344.30
Avg_Utilization	30,000	0.37	0.35	-0.23	0.03	0.28	0.69	5.36
Six_Month_Bal_Grwth	30,000	-12,351.57	43,922.42	-708,323	-19,793.8	-923	2,963	428,791
Six_Month_Utilization_Grwth	30,000	-0.11	0.30	-5.31	-0.18	-0.01	0.03	1.83
Max_Bill_Amt	30,000	60,572.44	78,404.81	-6,029	10,060	31,208.5	79,599	1,664,089
Max_Pay_Amt	30,000	15,848.23	37,933.56	0	2,198	5,000	12,100	1,684,259
Low_Income	30,000	0.25	0.43	0	0	0	1	1
Medium_Income	30,000	0.50	0.50	0	0	1	1	1
High_Income	30,000	0.25	0.43	0	0	0	0	1

Important features within the dataset were then analyzed to understand the distribution of values across all customers. Only a few of the data visualizations will be analyzed in this section. A more extensive analysis of features in the dataset can be found in the appendix. Customer average bill amount was analyzed and visualized to view the distribution of average bill amount across the customers in the dataset.



As can be seen in Figure 1 above, majority of the Average Bill Amounts present in the dataset are between $0 - $100,000 with a few customers exceeding $200,000 in Average Pay Amount. This is important to understand because Average Pay Amount is a proxy for income. Customers with a lower Average Pay Amount most likely have a lower income, which may be an indicator for Default.

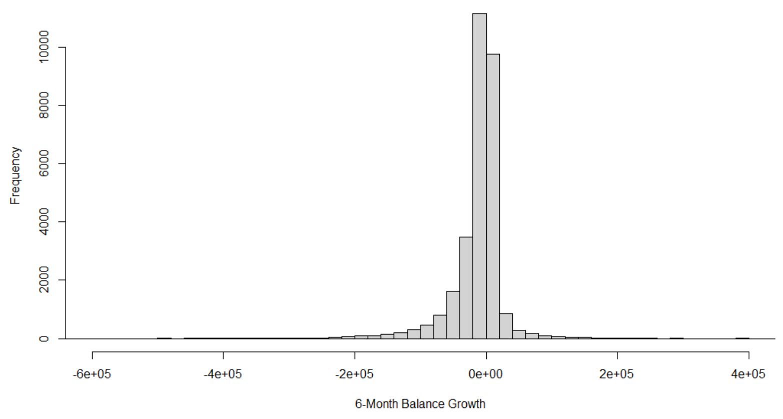
Average utilization was then analyzed to understand the distribution of average utilization among the customers. Average utilization may be an indicator of a customer risk of Default.



As can be seen in Figure 2 above, majority of the customers have an average utilization of slightly above 0. This means that majority of the customers in the data do not spend the total of their credit limit. Another interesting analysis from Figure 2 is that there are customers in the dataset who have an Average Utilization above 1. This means that customers are able to charge more than their credit limit on their cards. Charging more then what the bank determined was an adequate amount is a potentially important indicator that a customer may default.

The last feature exploratory data analysis that will be highlighted in this section is the customer 6-month balance growth. The histogram below displays the distribution of 6-month balance growth within the dataset.

Figure 3: Histogram of Customer 6-Month Balance Growth

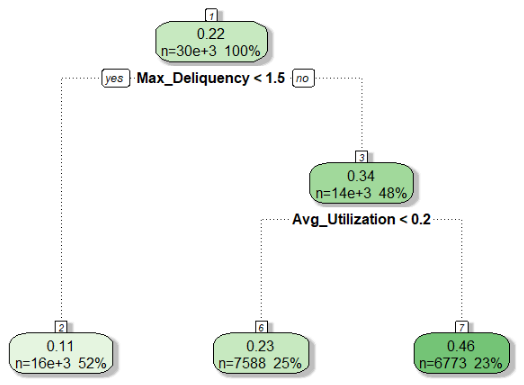


As can be seen in Figure 3 above, the majority of the 6-month balance growth falls within the -$10,000 and $10,000 range. This makes sense as some customers were able to pay off their balance amounts within the 6-month, so they see a negative 6-month balance growth while others did not pay off the monthly balances in full and accumulated a higher balance within the 6-months leading to a positive 6-month balance growth. It is interesting to see that there are a few customers who have a very large positive 6-month balance growth. A large 6-month balance growth indicates that a customer is struggling to pay off their monthly bill but are continuing to charge to their card. This type of behavior may be an important indicator of default. Additional histograms and boxplots analyzing engineered features within the dataset can be found in the appendix.

4.2 Model Based EDA

Now that a traditional EDA process has been conducted, a model-based approach can be taken by creating a dendrogram using the engineered features created in the previous section as these are the variables that will be used in the models. The dendrogram can be found below as Figure 4.

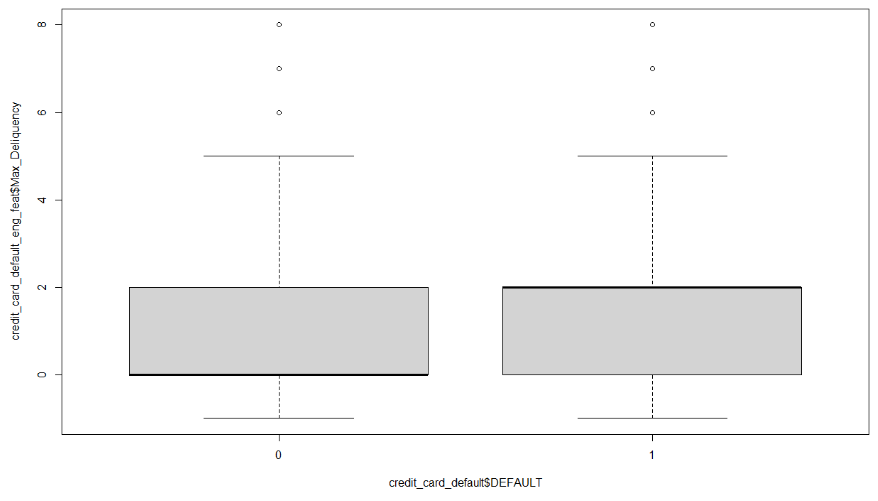
Figure 4: Dendrogram of Feautred Engineered Credit Card Default Dataset



Rattle 2022-Feb-27 21:22:05 jonah.muniz

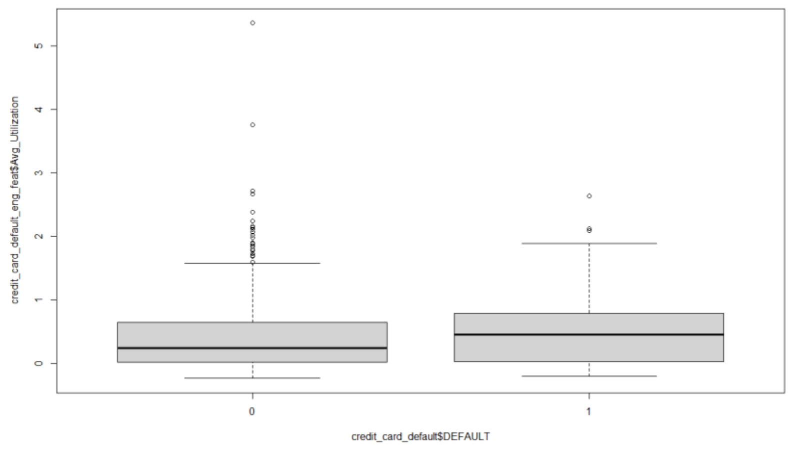
As can be seen in the figure above Max_Deliquency and Avg_Utilization are key features that can be used to differentiate the customers in the dataset. Figure 5 below is a box plot of these Max_Deliquency filtered by Default value.

Figure 5: Boxplot of Max_Deliquency by DEFAULT



As can be seen in the Figure 9 above, the mean Max_Deliquency value for DEFAULT = 0 is 0 and the mean Max_Deliquency value for DEFUALT = 1 is 2 which means on average the customers who default had a Max_Deliquency of 2 months. This makes sense as the longer a bill is unpaid the more likely a customer is to default. There are outliers in the DEFAULT = 0 customers where someone who does not default has a Max_Deliquency of 1-8 but majority of these customers have a Max_Deliquency of 0. Figure 6 below compares the Avg_Utilization to DEFAULT type to see if there is a correlation between the two variables.

Figure 6: Boxplot of Avg_Utilization by DEFAULT



As can be seen in Figure 6 above, the mean Avg_Utilization for the group of customers who did not default is lower than the mean for the group of customers who defaulted. There are outliers with both group of customers but majority of the customers who default have a higher Avg_Utilization.

5.0 Predictive Modeling: Methods and Results

Now that all the data quality, feature engineering and exploratory data analysis has been performed, four modeling approaches can be conducted and compared to see which performs the best at creating a credit risk model. The metrics of success that will be used to measure model success is True Positive Rate, False Positive Rate, Area Under the Curve (AUC), F1 Score, and Accuracy.

True Positive Rate is the probability that an actual positive, or default customer, will be predicted as positive, or default customer. A False Positive Rate is the ratio of negative events, non-default customers, that are wrongly categorized as default customers. AUC is the measure of the ability of a classifier to distinguish between classes. F1 Score is the harmonic mean of precision and recall. Lastly, the Accuracy of the model is the number of correct observations divided by the total number of observations.

5.1 Random Forest

The first model that will be modeled and tested is a random forest model. Random forest models are flexible and do not require extensive hyper-parameter tuning, which makes them one of the most used machine learning algorithms. It is also one of the most used machine learning algorithms due to its ability to be used in both classification and regression tasks. For this work the random forest algorithm will be used to classify customers on whether or not they will default. It is referred to as a random forest model due to its ability to build multiple “forests” of decision trees and ensemble the trees. It also leverages the “bagging” method to train the decision trees. The bagging method fits the model with different subsets of the training dataset then combines the predictions from all models to come to one classification. A random forest of 500 trees was created and trained using the bagging method on the train dataset. 3 variables were used at each split. Once the random forest algorithm was trained it was used to predict the default classification of the customers in the train and test dataset. A confusion matrix was then created for each dataset prediction to compare predicted default classification to actual classifications. The results of the random forest model can be seen below in Table 3.

Model #1: Random Forest Model (Train)

Actual	Predicted Class		Totals		Actual	Predicted Class		TP	1.00	TP+TN	1.98	AUC	0.97
Class	0	1			Class	0	1	TN	0.98	Precision	0.95	Sensitivity	1.00
0	11,755	184	11,939		0	0.98	0.02	Type I Error	0.02	Recall	1.00	Specificity	0.98
1	2	3,239	3,241		1	0.00	1.00	Type II Error	0.00	F1	0.99	Accuracy	99%

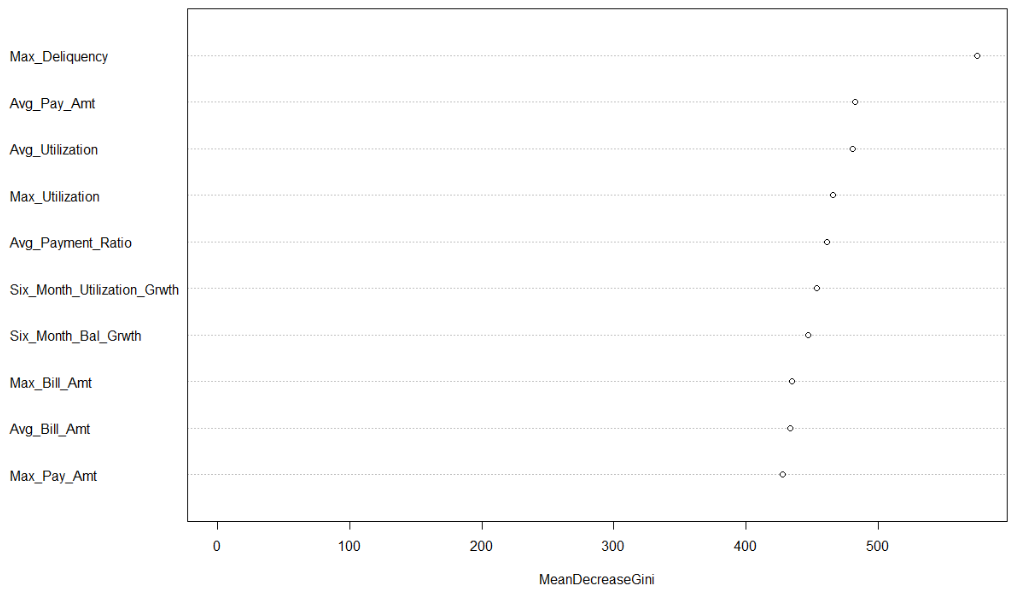
Model #1: Random Forest Model (Test)

Actual	Predicted Class		Totals		Actual	Predicted Class		TP	0.51	TP+TN	1.33	AUC	0.60
Class	0	1			Class	0	1	TN	0.82	Precision	0.26	Sensitivity	0.51
0	5,371	1,146	6,517		0	0.82	0.18	Type I Error	0.18	Recall	0.51	Specificity	0.82
1	395	411	806		1	0.49	0.51	Type II Error	0.49	F1	0.60	Accuracy	79%

Table 3: Confusion matrix and classification metrics for Model #1: Random Forest Model.

As can be seen in Table 3, it seems that the random forest model overfitted on the training dataset, as shown by the very high true positive rate and low false positive rate compared to the Test confusion matrix. The difference in accuracy also shows the performance variance between the train and test models. The feature importance plot below displays which variables impact the classification the most.

Figure 7: Top Ten Variable Importance



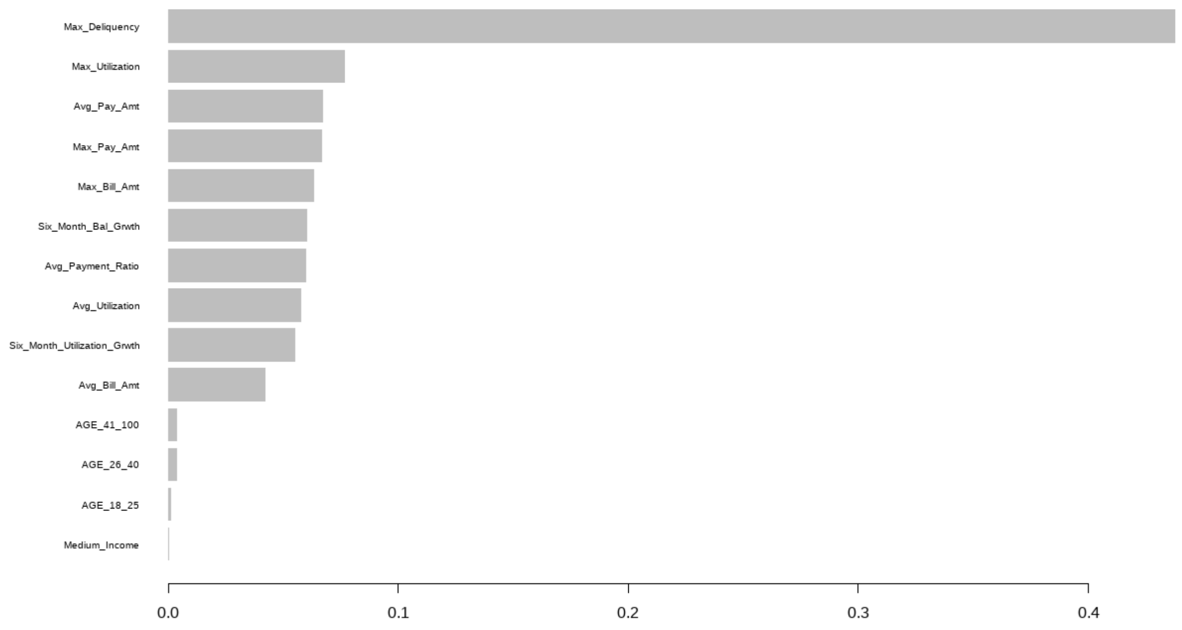
As can be seen in Figure 7 above, the top three important features in the random forest model are Max_Deliquency, Avg_Pay_Amt, and Avg_Utilization. Now that the confusion matrix for both the test and train dataset has been determined for the random forest model, the XGBoost model can be trained.

5.2 Gradient Boosting

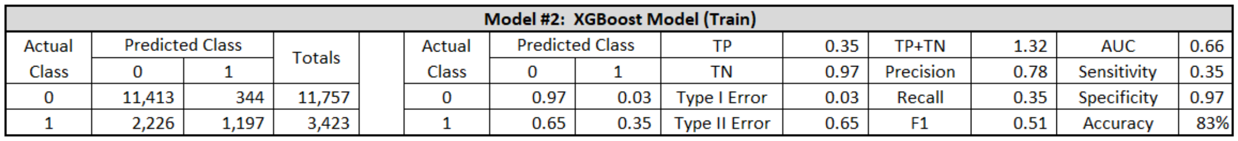
The second model that will be trained will be an Extreme Gradient Boosting model which is a type of gradient boosting model. Boosting is a sequential ensemble technique in which the model is improved by using information from previously grown models to improve the performance of subsequent models. Both the random forest model and the gradient boosting model ensemble trees to aggregate many individual models to come to a solution. However, the gradient model differs from the random forest because boosting prior model performance is used to improve the next model. Prior to training and testing the Extreme Gradient Boosting model, the training dataset was converted to a data matrix which is a requirement of the Extreme Gradient Boosting model. The same was done for the test dataset. Once the training and testing datasets were converted to data matrix the model parameters needed to be defined. The model parameters used were an objective function of multi:softprob, evaluation metric of mlogloss and the number of classes equals the length of the training dataset. Lastly the Extreme Gradient Boosting model was trained on 100 rounds.

A feature important plot was created to understand which features were the most important in the model for predicting customer default. The feature importance plot can be seen below as Figure 8.

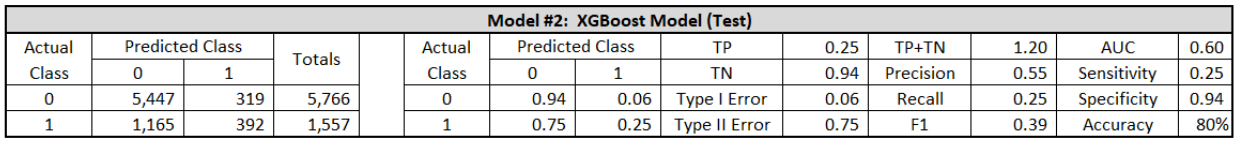
Figure 8: XGBoost Feature Importance Plot



As can be seen above in Figure 8, the top 5 most important features in the XGBoost model are Max_Deliquency, Max_Utilization, Avg_Pay_Amt, Max_Pay_Amt, and Avg_Bill_Amt. All five of the important features make sense as these features address how long a customer has not paid, how much of their credit limit they are using and overall, how high their bills are. Now that the XGBoost model has been trained and the feature importance plot has been created and analyzed, the model can now be used to predict default values for the train and test dataset and a confusion matrix can be create for each model. The results of the XGBoost train and test model performance can be seen below in Table 4.



Model #2: XGBoost Model (Train)													
Actual	Predicted Class		Totals		Actual	Predicted Class		TP	0.35	TP+TN	1.32	AUC	0.66
Class	0	1			Class	0	1	TN	0.97	Precision	0.78	Sensitivity	0.35
0	11,413	344	11,757		0	0.97	0.03	Type I Error	0.03	Recall	0.35	Specificity	0.97
1	2,226	1,197	3,423		1	0.65	0.35	Type II Error	0.65	F1	0.51	Accuracy	83%



Model #2: XGBoost Model (Test)													
Actual	Predicted Class		Totals		Actual	Predicted Class		TP	0.25	TP+TN	1.20	AUC	0.60
Class	0	1			Class	0	1	TN	0.94	Precision	0.55	Sensitivity	0.25
0	5,447	319	5,766		0	0.94	0.06	Type I Error	0.06	Recall	0.25	Specificity	0.94
1	1,165	392	1,557		1	0.75	0.25	Type II Error	0.75	F1	0.39	Accuracy	80%

Table 4: Confusion matrix and classification metrics for Model #2: XGBoost Model.

As can be seen above in Table 4, the train model outperformed the test model but the variance between model performance was reduced compared to the random forest model. This indicates that the XGBoost model is not overfitting as much as the random forest model. The accuracy of the train and test models are comparable at 83% and 80%. There is a reduction in True Positive Rate between Train and Test and False Positive Rate also increases between Train and Test. Both of these results are expected as the model was trained on the Train dataset and the Test data is new to the model.

5.3 Logistic Regression with Variable Selection

The third model that will be trained and tested will be a logistic regression model with variable selection. The logistic regression algorithm is a statistical method that predicts a binary outcome, such as yes or no or 0 or 1 for the current case. The addition of a variable selection method to logistic regression enables the “best” variables to be selected to create the “best” performing logistic regression model. Variable selection methods are implemented when there are a high number of features because it reduces the iterative process of selecting different variations of feature combinations to see which produces the best model. There are three variable selection processes that can be used to select the features: forward, backward, and stepwise variable selection.

Forward selection starts with only the intercept in the model and additional features are added to the model until the model no longer improves. Backward selection starts with all features included then removes one feature at a time. The process continues until the model performance no longer improves. Lastly, stepwise variable selection is a combination of the forward and backward methodology where a variable can be added or removed from the model until model performance no longer improves.

The table of important features from the XGBoost model was used as the initial list of variables for the logistic regression model. Table 5 displays the summary statistics of the initial list of important features.

Table 5: Summary of Important Feature List

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Max_Deliquency	15,180	0.68	1.07	0	0	2	8
Max_Utilization	15,180	0.49	0.43	-0.10	0.07	0.92	6.46
Avg_Pay_Amt	15,180	5,255.11	10,150.14	0.00	1,111.75	5,554.42	627,344.30
Max_Pay_Amt	15,180	15,620.73	35,279.24	0	2,195.8	12,200.8	1,215,471
Max_Bill_Amt	15,180	60,425.45	77,746.88	-2,900	10,050.8	79,119.5	823,540

Six_Month_Bal_Grwth	15,180	-12,495.35	44,045.77	-497,231	-20,094.2	2,962	399,983
Avg_Payment_Ratio	15,180	11.65	35.78	-605.46	0.05	1.00	2,687.00
Avg_Utilization	15,180	0.37	0.35	-0.20	0.03	0.69	4.47
Six_Month_Utilization_Grwth	15,180	-0.11	0.30	-5.31	-0.18	0.03	1.83
Avg_Bill_Amt	15,180	44,934.91	63,150.01	-56,043	4,789.1	56,880.5	592,432
AGE_41_100	15,180	0.28	0.45	0	0	1	1
AGE_26_40	15,180	0.60	0.49	0	0	1	1
AGE_18_25	15,180	0.13	0.34	0	0	0	1

A logistic model was trained on the above features and then stepwise variable selection was used on that model to select the best features to improve overall performance. Below in Table 6 is a summary table of the final logistic model with the final features after stepwise selection.

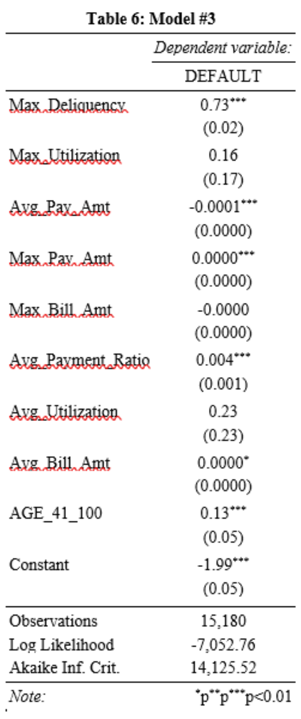
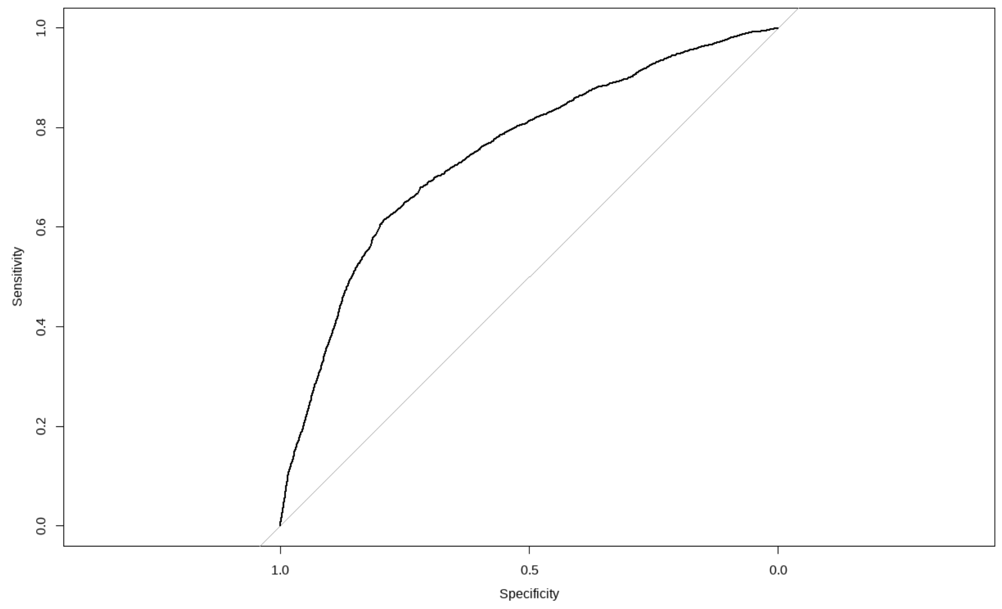


Table 6: Model #3

	Dependent variable:
	DEFAULT
Max_Deliquency	0.73***
	(0.02)
Max_Utilization	0.16
	(0.17)
Avg_Pay_Amt	-0.0001***
	(0.0000)
Max_Pay_Amt	0.0000***
	(0.0000)
Max_Bill_Amt	-0.0000
	(0.0000)
Avg_Payment_Ratio	0.004***
	(0.001)
Avg_Utilization	0.23
	(0.23)
Avg_Bill_Amt	0.0000*
	(0.0000)
AGE_41_100	0.13***
	(0.05)
Constant	-1.99***
	(0.05)
Observations	15,180
Log Likelihood	-7,052.76
Akaike Inf. Crit.	14,125.52
Note:	*p**p***p<0.01

The coefficients of the logistic regression model can be seen above in Table 6. The coefficients' magnitude and sign indicate how much an increase in each variable will affect the probability of the observation being a default customer if all other variables are held constant. Now that the model has been trained and can be used to generate predictions the threshold value must be determined to understand the ideal cutoff point to classify customers. An ROC curve, displayed below, can be used determine the ideal threshold value to classify the customers.

Figure 9: Logistic Regression ROC Curve



The threshold value is the point on the curve that is closest to the top left corner. The threshold value that was used to classify customers was 0.2864228. This means that all customers who had a predicted probability of default greater than 0.2864228 were assigned 1 and the rest were assigned 0. The performance of the predictions using the train and test datasets can be seen below in Table 7.

Model #3: Logistic Regression With Stepwise Feature Selection (Train)													
Actual	Predicted Class		Totals		Actual	Predicted Class		TP	0.46	TP+TN	1.34	AUC	0.75
Class	0	1			Class	0	1	TN	0.88	Precision	0.61	Sensitivity	0.46
0	9,306	1,318	10,624		0	0.88	0.12	Type I Error	0.12	Recall	0.46	Specificity	0.88
1	2,451	2,105	4,556		1	0.54	0.46	Type II Error	0.54	F1	0.58	Accuracy	62%

Model #3: Logistic Regression With Stepwise Feature Selection (Test)													
Actual	Predicted Class		Totals		Actual	Predicted Class		TP	0.43	TP+TN	1.32	AUC	0.71
Class	0	1			Class	0	1	TN	0.89	Precision	0.66	Sensitivity	0.43
0	4,385	531	4,916		0	0.89	0.11	Type I Error	0.11	Recall	0.43	Specificity	0.89
1	1,381	1,026	2,407		1	0.57	0.43	Type II Error	0.57	F1	0.56	Accuracy	74%

Table 7: Confusion matrix and classification metrics for Model #3: Logistic Regression With Stepwise Feature Selection Model.

As can be seen above, the Test dataset outperforms the Train dataset in overall accuracy which is a bit surprising given that the Train dataset was used to train the model and the Test dataset was never seen by the model. Diving deeper it can be seen that the Train dataset has a lower True Positive Rate compared to Test. The Train dataset has a lower False Positive Rate though compared to Test. Test outperforms Train slightly when comparing F1 Scores. Lastly, Train outperforms Test dataset in AUC. Overall, the Test dataset metrics will be used to evaluate the models as the Test dataset is a net new dataset to the model which models its performance in production better than the Train dataset.

5.4 Naïve Bayes

The fourth and final model that will be trained and tested will be a naïve bayes model. Naïve bayes models are a supervised non-linear classification algorithm. Naïve bayes classifiers are a family of simplistic probabilistic classifiers based in applying Bayes' theorem with the assumption that the occurrence of a certain feature is independent of the other features. Bayes' theorem gives the conditional probability of an event A given another event B has occurred. Using a naïve bayes model the train and test model were trained and the performance of the model can be seen below in Table 5.

Model #5: Naïve Bayes (Train)													
Actual	Predicted Class		Totals		Actual	Predicted Class		TP	0.40	TP+TN	1.26	AUC	0.67
Class	0	1			Class	0	1	TN	0.86	Precision	0.60	Sensitivity	0.40
0	8,657	1,384	10,041		0	0.86	0.14	Type I Error	0.14	Recall	0.40	Specificity	0.86
1	3,100	2,039	5,139		1	0.60	0.40	Type II Error	0.60	F1	0.52	Accuracy	70%

Model #5: Naïve Bayes (Test)													
Actual	Predicted Class		Totals		Actual	Predicted Class		TP	0.25	TP+TN	1.13	AUC	0.58
Class	0	1			Class	0	1	TN	0.87	Precision	0.82	Sensitivity	0.25
0	2,012	288	2,300		0	0.87	0.13	Type I Error	0.13	Recall	0.25	Specificity	0.87
1	3,754	1,269	5,023		1	0.75	0.25	Type II Error	0.75	F1	0.37	Accuracy	45%

Table 8: Confusion matrix and classification metrics for Model #4: Naïve Bayes Model.

As can be seen above in Table 8, the train model outperforms the test model. The accuracy of the train model was 70% compared to 45% in the test model. True Positive Rate also illustrates the difference in performance.

6.0 Comparison of Results

Below is a table comparing the four models, Random Forest, XGBoost, Logistic Regression and Naive Bayes, across the metrics of success described above.

Model Performance Comparison								
Metrics	Model #1 Random Forest		Model #2: XGBoost		Model #3: Logisitic Regression		Model #4: Naive Bayes	
	Train	Test	Train	Test	Train	Test	Train	Test
TP	1.00	0.51	0.35	0.25	0.46	0.43	0.40	0.25
FP	0.02	0.18	0.03	0.06	0.12	0.11	0.14	0.13
AUC	0.97	0.60	0.66	0.60	0.75	0.71	0.67	0.58
F1	0.99	0.60	0.51	0.39	0.58	0.56	0.52	0.37
Accuracy	99%	79%	83%	80%	62%	74%	70%	45%

As can be seen in the table above, the train and test performance of the four models can be compared across the five metrics of success. The two main metrics that will be used to determine the performance of the models are F1 score and Accuracy. The train model of the Random Forest model is very high in F1 score and Accuracy. As was mentioned above, this is due to overfitting of the model. The train model will be ignored as the test performance for the Random Forest model is dramatically different. When comparing the rest of the models the test models will only be considered. Comparing Accuracy

XGBoost performs the best compared to Random Forest, Logistic Regression, and Naïve Bayes. Comparing F1 Random Forest performs the best compared to the rest of the models. Due to this True Positive and False Positive Rate will be assessed. It is important to have a high True Positive Rate, but it is equally, if not more important to have a low False Positive Rate. Taking that into consideration, Random Forest model is the best performing model with a True Positive Rate of 0.51 and a False Positive Rate of 0.18. Although it does not have the lowest False Positive Rate, XGBoost, it does have the highest True Positive Rate and largest difference between True Positive Rate and False Positive Rate.

7.0 Conclusion

The objective of the paper is to use the Taiwan customer default dataset to understand out of the four models chosen, Random Forest, XGBoost, Logistic Regression with Variable Selection and Naiive Bayes which performed the best when predicting if a customer would default. Based on the above results the Random Forest model performed the best on the Test dataset. It had a True Positive Rate of 0.51, False Positive Rate of 0.18, F1 Score of 0.60, AUC of 0.60, and an overall Accuracy of 79%. To determine if the random forest model is performance is significant the True Positive Rate, False Positive Rate and Accuracy of the existing legacy model will need to be compared to the Random Forest model to truly understand if this performance is an upgrade to what exists today.

This process could be improved by increasing the number of months of data available. The current dataset used was limited to 6 months of data. If 12 months or even 24 months of data on a customer could be used to train and test the models the overall performance of the models to predict default would most likely improve. In addition to more data, the performance of the models would improve if parameter tuning was conducted. One example is with the Random Forest model. There was significant overfitting of the Random Forest model with the Train dataset. If the parameters of the Random Forest model could have been tuned the overfitting could have been mitigated, potentially leading to better performance on the Test dataset. Overall, the objective of the research paper was achieved. The four models were compared on their ability to predict a customer defaulting based on the Taiwan dataset and the Random Forest model performed the best on the Test dataset.

8.0 Appendix

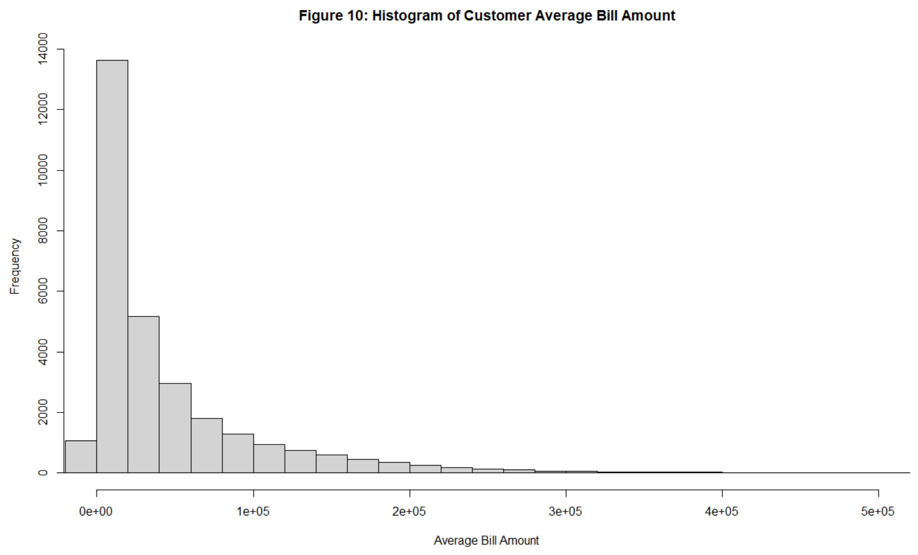


Figure 10: Histogram of Average Bill Amount per Customer

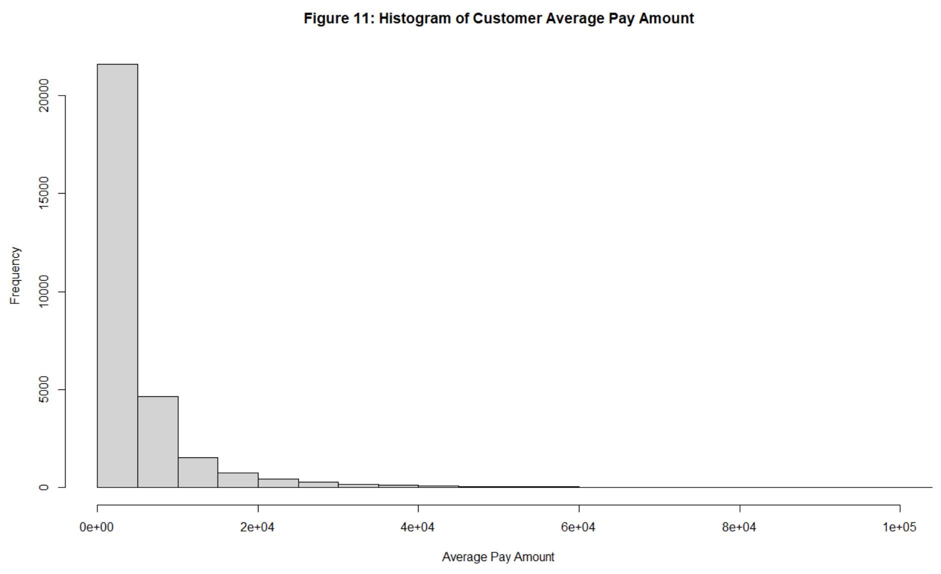


Figure 11: Histogram of Average Pay Amount per Customer

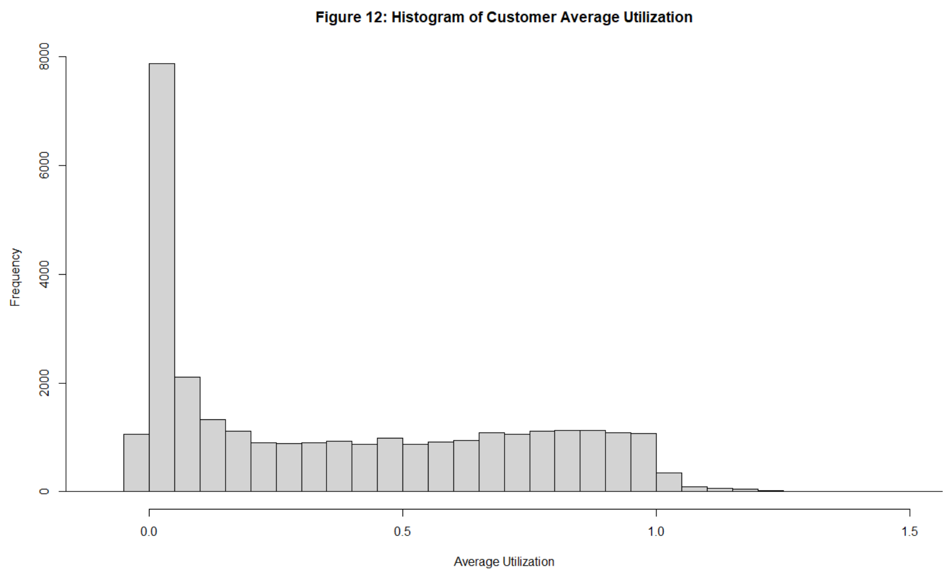


Figure 12: Histogram of Average Utilization per Customer

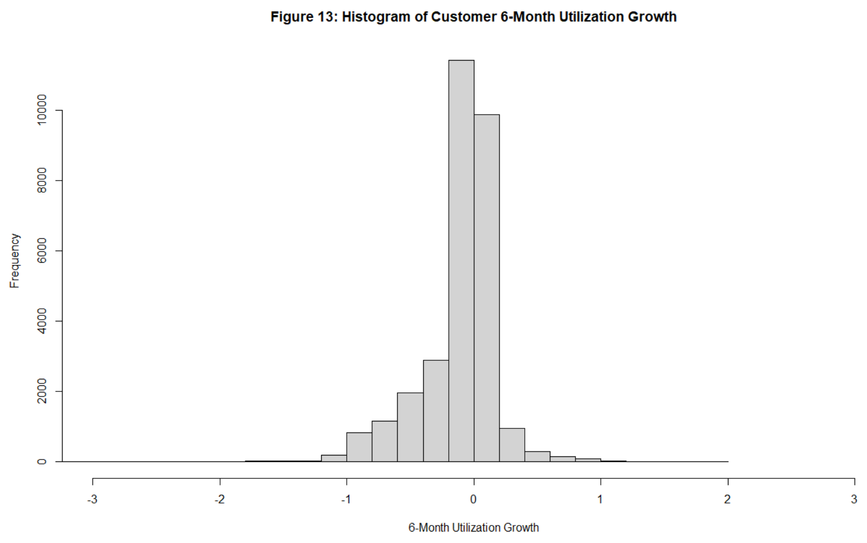


Figure 13: Histogram of 6-Month Utilization Growth

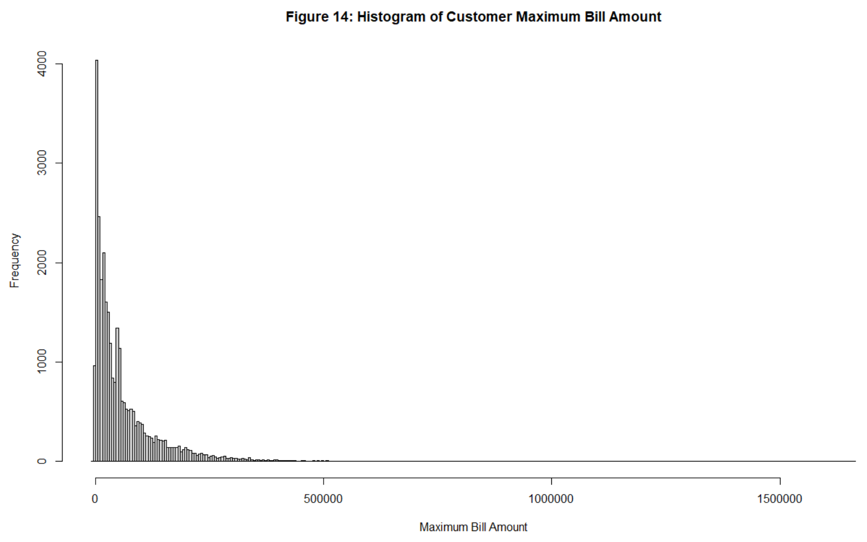


Figure 14: Histogram of Consumer Max Bill Amount