Summer 2019

Capstone 498

Lauren Camero
NORTHWESTERN UNIVERSITY

Introduction:

This report contains a data survey, a data quality check, and an initial exploratory data analysis for the credit card data set. The data are modeled to classify whether a customer will default on their credit. To build a model that classifies whether a customer will default on their credit, an exploratory data analysis (EDA) of the data set is needed to be conducted. First, the data documentation was reviewed to understand the types of variables collected. Descriptive statistics were performed on the variables when possible. Otherwise the remaining variables were tabulated and examined for null entries or errors.

Five models were built, and results were compared. The five models selected were Random Forests, Gradient Boosting, Logistic, Support Vector Machines, Naïve Bayes. The five models were then compared using binary performance metrics to select the best model.

Data:

The data set contains 30,000 records and 30 variables from customer default payments in Taiwan. The data dictionary is illustrated in Figure 1 below.

Figure 1: Data Dictionary

Limit Balance	Amount of credit including both individual and consumer credit from family (supplementary card)
Sex	Gender (1 = male, 2 = female)
Education	(1 = graduate school, 2 = university, 3 = high school, 4 = other)
Marriage	Marital Status (1 = married, 2 = single, 3 = other)
Age	Age in years
Pay 0	Repayment status in September 2005 (-1 = pay duly, 1 = payment delay for one month, 2 = payment delay for 2 months, … , 9 = payment delay for 9 months)
Pay 1	Repayment status in August 2005 (-1 = pay duly, 1 = payment delay for one month, 2 = payment delay for 2 months, … , 9 = payment delay for 9 months)
Pay 2	Repayment status in July 2005 (-1 = pay duly, 1 = payment delay for one month, 2 = payment delay for 2 months, … , 9 = payment delay for 9 months)
Pay 3	Repayment status in June 2005 (-1 = pay duly, 1 = payment delay for one month, 2 = payment delay for 2 months, … , 9 = payment delay for 9 months)
Pay 4	Repayment status in May 2005 (-1 = pay duly, 1 = payment delay for one month, 2 = payment delay for 2 months, … , 9 = payment delay for 9 months)
Pay 5	Repayment status in April 2005 (-1 = pay duly, 1 = payment delay for one month, 2 = payment delay for 2 months, … , 9 = payment delay for 9 months)
Pay 6	Repayment status in March 2005 (-1 = pay duly, 1 = payment delay for one month, 2 = payment delay for 2 months, … , 9 = payment delay for 9 months)
Bill Amount 1	Bill amount for September 2005
Bill Amount 2	Bill amount for August 2005
Bill Amount 3	Bill amount for July 2005
Bill Amount 4	Bill amount for June 2005
Bill Amount 5	Bill amount for May 2005
Bill Amount 6	Bill amount for April 2005
Pay Amount 1	Amount paid in September 2005
Pay Amount 2	Amount paid in August 2005
Pay Amount 3	Amount paid in July 2005
Pay Amount 4	Amount paid in June 2005
Pay Amount 5	Amount paid in May 2005
Pay Amount 6	Amount paid in April 2005

Data Survey:

At first glance, the dataset has no missing values. Below are the descriptive statistics of the continuous variables (bill, payment amounts, and limit balance) and tabular counts of the categorical variables (sex, education, marriage, age brackets, and repayment statuses.

There are no missing values in the data, but quite a few errors. There seems to be no errors in the first category, gender.

Figure 2: Gender Descriptive Statistics

Sex	Male	Female
	11888	18112

However, Education has 3 categories besides the ones described in the data dictionary: 0, 5, and 6. This is an error and correct these datapoints by categorizing these 345 records as “Other.”

Figure 3: Education Descriptive Statistics

Education	High School	University	Masters	Other	0	5	6
	10585	14030	4917	123	14	280	51

Similarly, there is a “0” category in the Marriage field. The assumption is that this is a mistake and correct the 54 records by renaming them as “Other” if labeled “0.”

Figure 4: Marriage Descriptive Statistics

Marriage	Married	Single	Other	0
	13659	15964	323	54

Our target variable, Default, is below in Figure 5.

Figure 5:Default Descriptive Statistics

Default	No	Yes
	23364	6636

Next is the categorical variables of repayment statuses and there are a -2 and 0 that are not described in the data dictionary. This could mean that the user paid forward a month meaning they overpaid their bill. At this point, there is no need to deduce if the -2 payment status is useful information and decide to

change all -2 and -1 payment statuses to 0. Additionally, the data shows a field names Pay_0, not Pay_1 as listed in the dictionary.

Figure 6: Payment Status Descriptive Statistics

	-2	-1	0	1	2	3	4	5	6	7	8
Pay 0	2759	5686	14737	3688	2667	322	76	26	11	9	19
Pay 2	3782	6050	15730	28	3927	326	99	25	12	20	1
Pay 3	4085	5938	15764	4	3819	240	76	21	23	27	3
Pay 4	4348	5687	16455	2	3159	180	69	35	5	58	2
Pay 5	4546	5539	16947	2	0	178	84	17	4	58	1
Pay 6	4895	5740	16286	2	0	184	49	13	19	46	2

The descriptive statistics of the bill amounts for each month show that there are negative bill amounts. It seems that there should be no reason for a negative bill amount unless the customer overpaid their bill in a previous month. Based on the observations, this logic is correct and these negative datapoints are kept unaltered.

Figure 7: Bill Amount Descriptive Statistics

Bill Amount 1	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
	(165,580)	3,559	22,382	51,223	67,091	964,511
Bill Amount 2	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
	(69,777)	2,985	21,200	49,179	64,006	983,931
Bill Amount 3	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
	(157,264)	2,666	20,089	47,013	60,165	1,664,089
Bill Amount 4	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
	(170,000)	2,327	19,052	43,263	54,506	891,586
Bill Amount 5	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
	(170,000)	2,327	19,052	43,263	54,506	891,586
Bill Amount 6	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max.**
	(339,603)	1,256	17,071	38,872	49,198	961,664

The Monthly Payment Amount's descriptive statistics are in Figure 8, and there are no errors or outliers.

Figure 8: Payment Amount Descriptive Statistics

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Pay Amount 1	0	1,000	2,100	5,664	5,006	873,552
Pay Amount 2	0	833	2,009	5,921	5,000	1,684,259
Pay Amount 3	0	390	1,800	5,226	4,505	896,040
Pay Amount 4	0	296	1,500	4,826	4,013	621,000
Pay Amount 5	0	253	1,500	4,799	4,032	426,529
Pay Amount 6	0	118	1,500	5,216	4,000	528,666

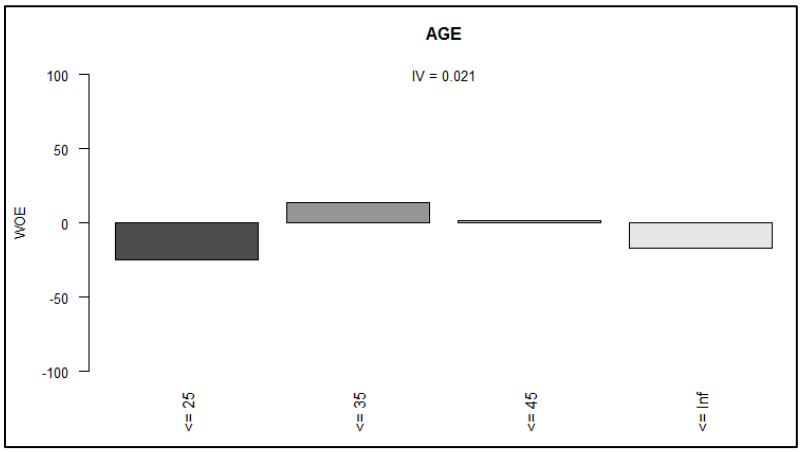
Next are the descriptive statistics for Age. Although there are no obvious outliers or errors, it may be better for the model to have age binned to form a categorical variable.

Figure 9: Age Descriptive Statistics

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Age	21	28	34	35	41	79

A weight of evidence binning strategy classifies the continuous variable, Age. This is a supervised tree segmentation. This type of binning fits the data characteristics of the Default indicator. Figure 10 is the binning segmentation of the Age_bin feature created for modelling.

Figure 10: Tree binning of Age



Finally, the Limit Balance has no errors or outliers.

Figure 11: Limit Balance Descriptive Statistics

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Limit Balance	10000	50000	140000	167484	240000	1000000

Feature Engineering:

New variables are created and displayed in Figure 12.

Figure 12:Additional Variables Created

avg_bill_amt	average monthly bill amount over 6 months
avg_pmt_amt	avergage payment amount over 6 months
pmt_ratio1	ratio of payment in September to bill in August
pmt_ratio2	ratio of payment in August to bill in July
pmt_ratio3	ratio of payment in July to bill in June
pmt_ratio4	ratio of payment in June to bill in May
pmt_ratio5	ratio of payment in May to bill in April
avg_pmt_ratio	avgerage of payment ratios
max_bill_amt	maximum amount billed over 6 months
max_pmt_amt	maximum amount paid over 6 months
util	ratio of balance to limit in September
util2	ratio of balance to limit in August
util3	ratio of balance to limit in July
util4	ratio of balance to limit in June
util5	ratio of balance to limit in May
util6	ratio of balance to limit in April
avg_util	average utilzation over 6 months
bal_growth_6mo	indicator of increase in balance from April to September
util_growth_6mo	indicator of increase in utilization from April to September
max_DLQ	max delinquency in repayment status over 6 months
min_pmt_ratio	minimum amount paid over 6 months
education by age	interactive variable created by the multiplication of education and age

After creating these variables and fixing errors summary statistics for all variables are examined to see if there are any further issues. There are no errors or outliers.

Figure 13: Summary Statistics for Model Variables

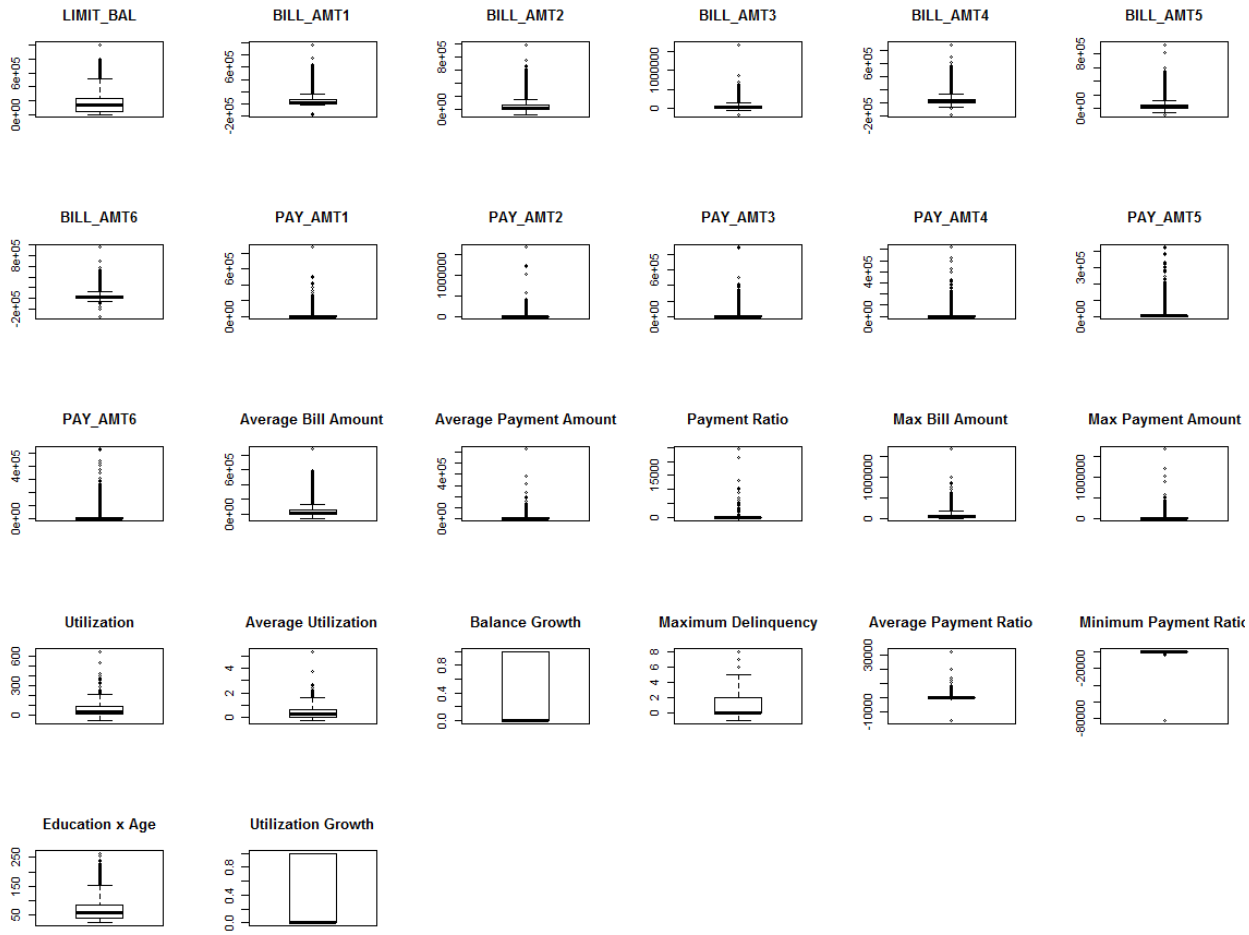
Summary Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
ID	30,000	15,000.50	8,660.40	1	7,500.8	15,000.5	22,500.2	30,000
LIMIT_BAL	30,000	167,484.30	129,747.70	10,000	50,000	140,000	240,000	1,000,000
SEX	30,000	1.60	0.49	1	1	2	2	2
EDUCATION	30,000	1.84	0.74	1	1	2	2	4
MARRIAGE	30,000	1.56	0.52	1	1	2	2	3
AGE	30,000	35.49	9.22	21	28	34	41	79
PAY_1	30,000	0.26	0.85	-1	0	0	0	8
PAY_2	30,000	0.19	0.91	-1	0	0	0	8
PAY_3	30,000	0.17	0.91	-1	0	0	0	8
PAY_4	30,000	0.11	0.88	-1	0	0	0	8
PAY_5	30,000	0.07	0.84	-1	0	0	0	8
PAY_6	30,000	0.06	0.85	-1	0	0	0	8
BILL_AMT1	30,000	51,223.33	73,635.86	-165,580	3,558.8	22,381.5	67,091	964,511
BILL_AMT2	30,000	49,179.08	71,173.77	-69,777	2,984.8	21,200	64,006.2	983,931
BILL_AMT3	30,000	47,013.15	69,349.39	-157,264	2,666.2	20,088.5	60,164.8	1,664,089
BILL_AMT4	30,000	43,262.95	64,332.86	-170,000	2,326.8	19,052	54,506	891,586
BILL_AMT5	30,000	40,311.40	60,797.16	-81,334	1,763	18,104.5	50,190.5	927,171
BILL_AMT6	30,000	38,871.76	59,554.11	-339,603	1,256	17,071	49,198.2	961,664
PAY_AMT1	30,000	5,663.58	16,563.28	0	1,000	2,100	5,006	873,552
PAY_AMT2	30,000	5,921.16	23,040.87	0	833	2,009	5,000	1,684,259
PAY_AMT3	30,000	5,225.68	17,606.96	0	390	1,800	4,505	896,040
PAY_AMT4	30,000	4,826.08	15,666.16	0	296	1,500	4,013.2	621,000
PAY_AMT5	30,000	4,799.39	15,278.31	0	252.5	1,500	4,031.5	426,529
PAY_AMT6	30,000	5,215.50	17,777.47	0	117.8	1,500	4,000	528,666
DEFAULT	30,000	0.22	0.42	0	0	0	0	1
u	30,000	0.50	0.29	0.0000	0.25	0.49	0.75	1.00
train	30,000	0.51	0.50	0	0	1	1	1
test	30,000	0.24	0.43	0	0	0	0	1
validate	30,000	0.25	0.43	0	0	0	0	1
data.group	30,000	1.74	0.83	1	1	1	2	3
target	30,000	1.78	0.42	1	2	2	2	2
age_bin	30,000	39.66	9.05	25	35	35	45	55
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_pmt_amt	30,000	5,275.23	10,137.95	0.00	1,113.29	2,397.17	5,583.92	627,344.30
pmt_ratio1	30,000	5.85	254.10	-498	0.04	0.1	0.5	24,437
pmt_ratio2	30,000	12.42	798.11	-385	0.04	0.1	0.5	100,000
pmt_ratio3	30,000	4.23	658.64	-82,150	0.03	0.05	0.4	60,000
pmt_ratio4	30,000	7.68	364.75	-4,307	0.02	0.04	0.3	31,000
pmt_ratio5	30,000	13.89	1,007.70	-185	0.03	0.04	0.4	162,000
max_bill_amt	30,000	60,572.44	78,404.81	-6,029	10,060	31,208.5	79,599	1,664,089
max_pmt_amt	30,000	15,848.23	37,933.56	0	2,198	5,000	12,100	1,684,259
util	30,000	42.38	41.15	-61.99	2.20	31.40	82.98	645.53
util2	30,000	41.11	40.46	-139.55	1.83	29.61	80.65	638.05
util3	30,000	39.22	39.64	-102.51	1.60	27.31	75.51	1,068.86
util4	30,000	35.95	36.87	-137	1.4	24.2	66.8	515
util5	30,000	33.31	35.05	-88	1.1	21.2	60.2	494
util6	30,000	31.86	34.53	-151	0.8	18.5	58.2	389
avg_util	30,000	0.37	0.35	-0.23	0.03	0.28	0.69	5.36
bal_growth_6mo	30,000	0.38	0.49	0	0	0	1	1
max_DLQ	30,000	0.61	1.15	-1	0	0	2	8
avg_pmt_ratio	30,000	8.81	304.44	-16,429.80	0.04	0.09	0.50	32,400.00
min_pmt_ratio	30,000	-3.29	475.91	-82,150	0	0.02	0.04	2
education_by_age	30,000	66.63	36.14	21	37	58	84	264
util_growth_6mo	30,000	0.38	0.49	0	0	0	1	1

Exploratory Data Analysis:

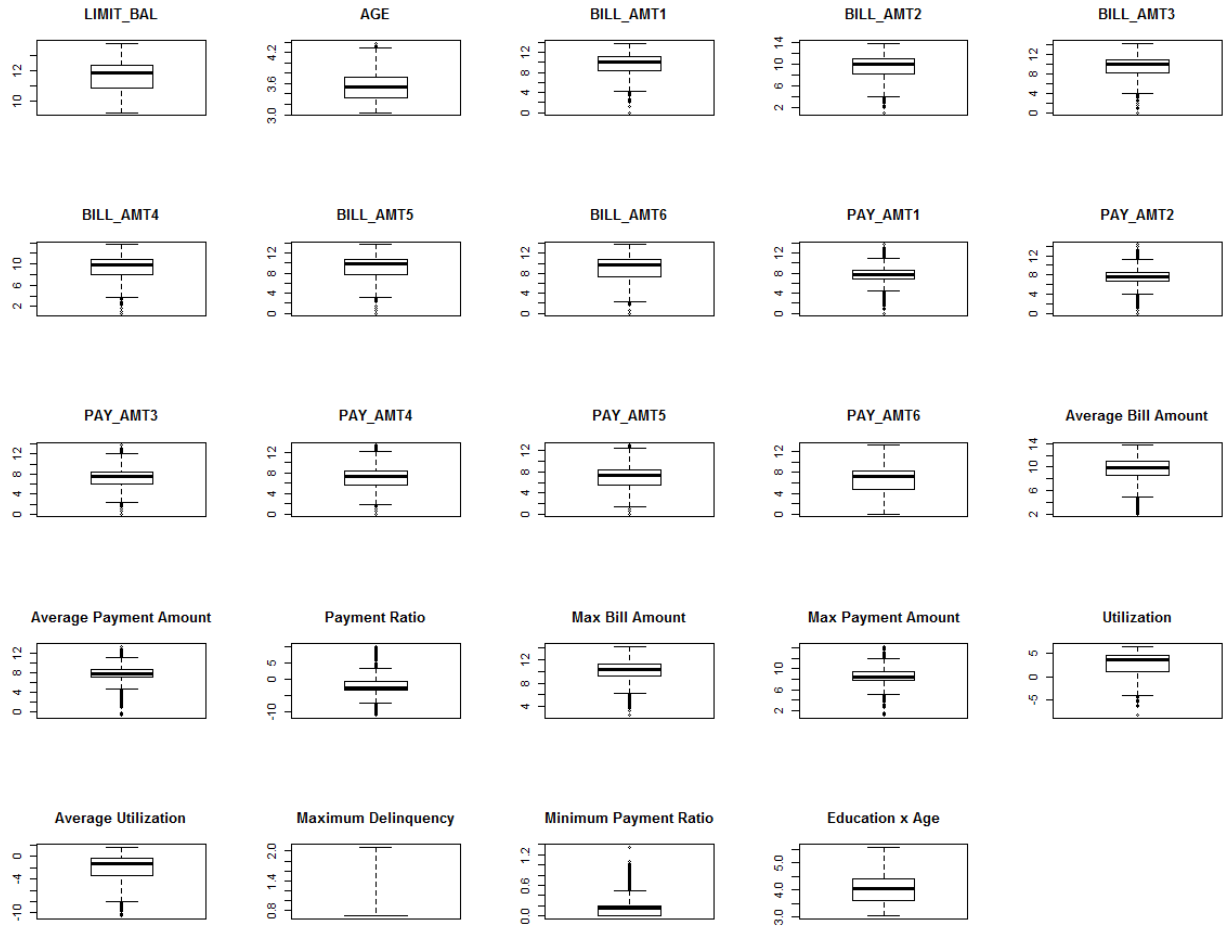
Next, the distributions of the variables are observed by visualizing their distributions in a boxplot matrix. The variables were mostly skewed to the right and could be improved if the variables were transformed by taking the natural log. Only Minimum Payment Risk was skewed to the left. Therefore, that variables were transformed using the square root.

Figure 14: Boxplots of Continuous Variables



The natural log is calculated for the following variables: balance, age, bill amount 1-6, payment amount 1-6, average bill amount, average payment amount, payment ratio, max bill amount, utilization, average utilization, max delinquency, and education by age. The boxplot matrix shown in Figure 15.

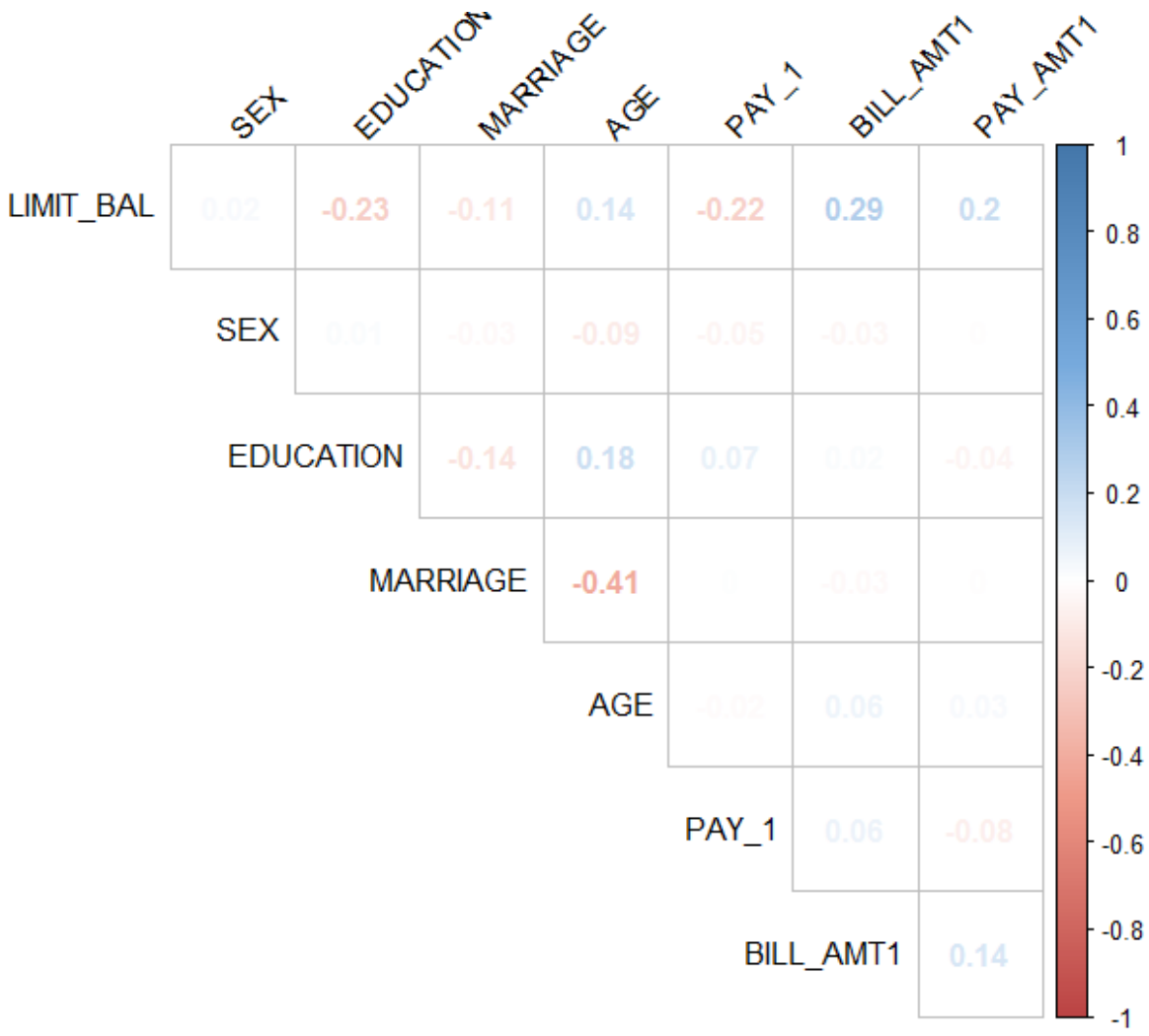
Figure 15: Boxplots of Transformed Continuous Variables



After reviewing Figure 15, the transformations benefited the normality of the continuous variables and transformed the model variables.

Next, a correlation matrix to visualize the relationship between the variables is illustrated in Figure 16.

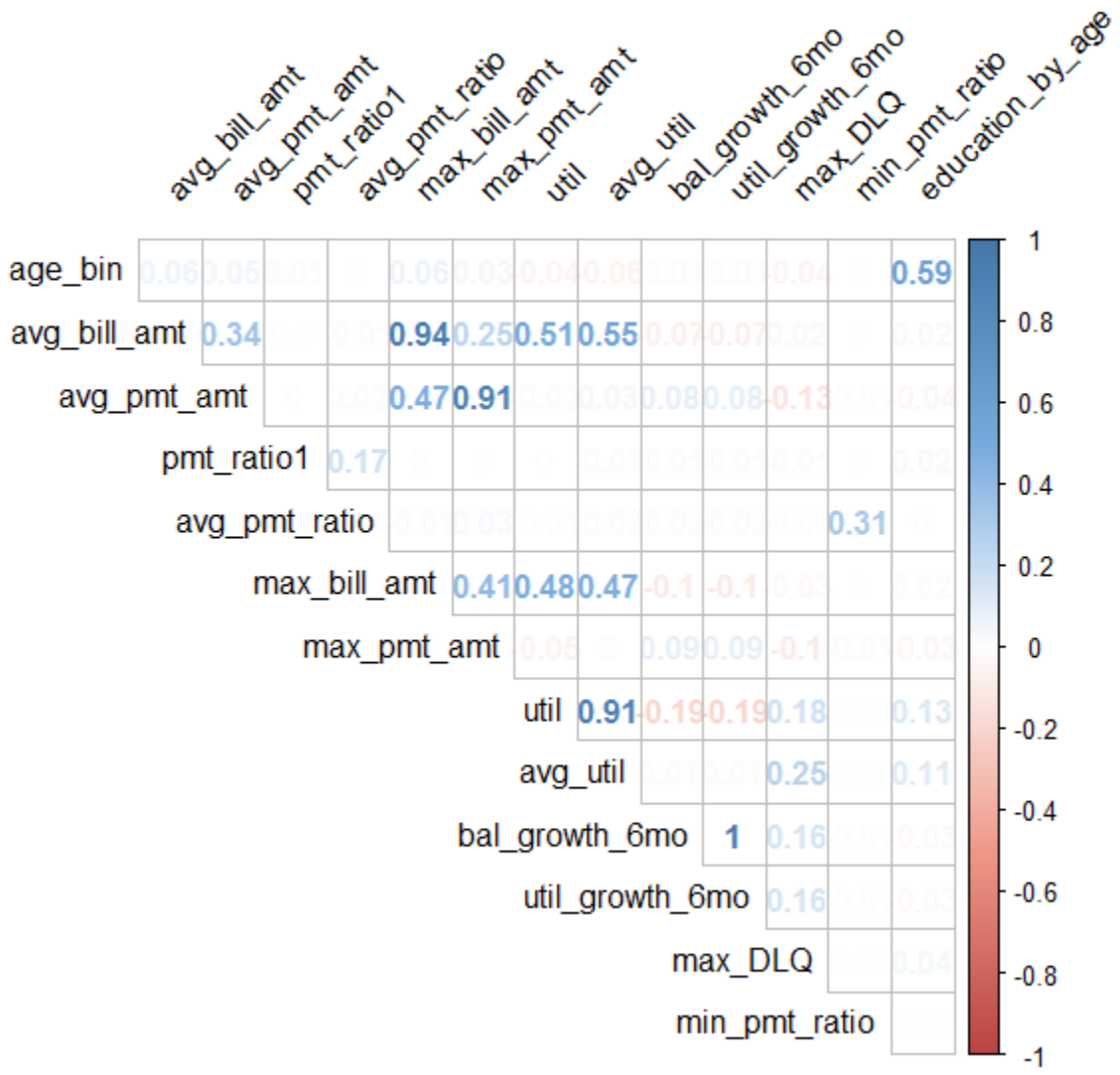
Figure 16: Correlation Matrix of Continuous Variables



Surprisingly, there are no strong correlations between the raw variables. The strongest correlations in the matrix are the negative relationship between marriage and age. Limit Balance and Bill Amount in September have a correlation of 0.29 which is intuitive since a high bill would only work for customers that have a high limit. Surprisingly, the payment amount and the bill amount in September have barely any correlation meaning that not many customers have paid their last bill.

Figures 17 illustrates the correlation plot of features that have been calculated.

Figure 17: Correlation Matrix of Calculated Features

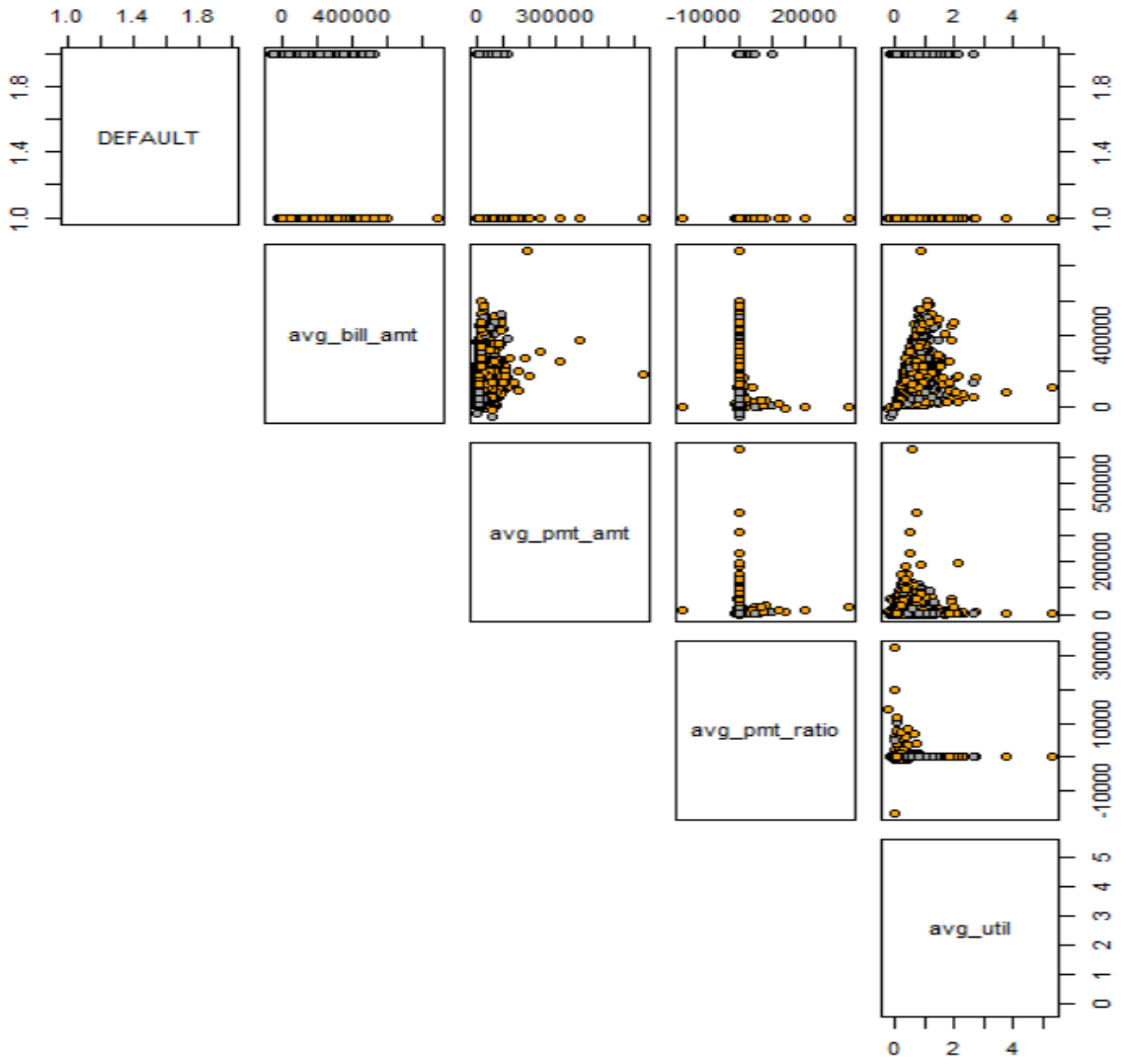


Since most of these variables are calculations or interactions of multiple features, it is unsurprising that the correlation between variables is much higher than the matrix of raw variables. The correlation between average utilization and average bill amount is 0.55. Also, the average payment amount and max bill amount has a correlation of 0.47. This means that, although the bill and payment amounts were not highly correlated in September, they were correlated on average over a 6-month period.

The correlation plot does not have any variables that are very highly correlated so there is no need to exclude certain features from the model to prevent multicollinearity.

To confirm this, Figure 18 represents a scatter plot matrix

Figure 18: Scatterplot Matrix



The grey points signify that the customer did not default and the yellow means they did. Similar to the correlation plot, there are no strong linear relationships with our calculated variables that should be excluded due to multicollinearity.

Finally, One Rule Machine Learning Classification Algorithm (OneR) discretizes all numeric data into categorical bins of equal length based on clusters. OneR binning is used for the education categorical variable. Discretized binning may help the performance of some models.

Figure 19: OneR Binning of Limit Balance

		Default	
Limit Balance		0	1
9010	130000	10389	4152
130000	1000000	12975	2484

The OneR binning method is used to bin the limit balance variable to see if this feature would perform better in any of the models. Based on this algorithm, we create a Limit Balance bin feature.

Predictive Modeling: Methods and Results

The following is a breakdown of the dataset into training, testing, and validation groups. Since the goal is to determine if a customer will default on their credit load, the dependent variable is a binary target and requires a logistic or categorical model.

Figure 20: Cross-Validation

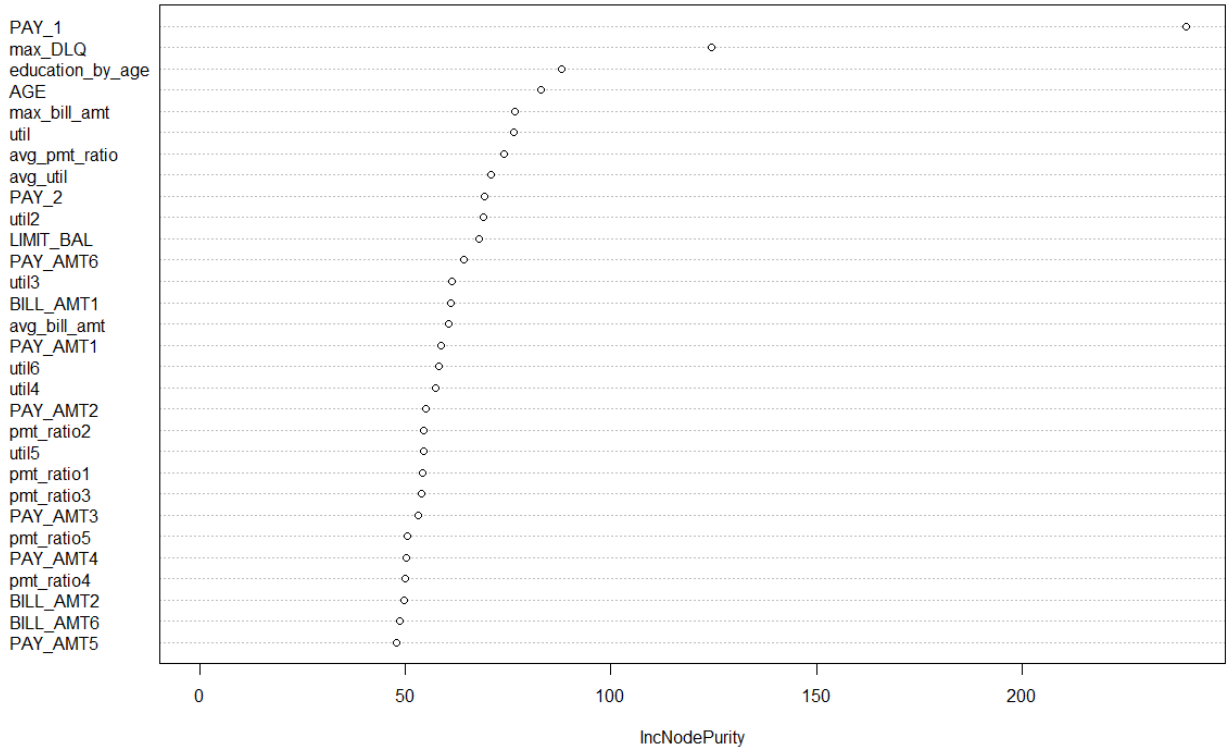
	Train	Test	Validate
Dataset breakdown	15180	7323	7497

Random Forest

First model is a Random Forest. For this model, all raw and engineered features were included. Below is a variable importance plot that illustrates the results from the random forest model. The variable

importance plot is calculated with the Gini index and 'purity' at each node split. A higher purity means the data are more easily classified.

Figure 21: Variable Importance Plot



As seen in Figure 21, the model calculates that the five most important variables are repayment status in August, max delinquency, interaction of education and age, age, and maximum amount billed. Since the raw variable, Pay_1, is an important variable, this model is kept. The model explains 19.83% of the variance in the binary variable, Default.

Figure 22 illustrates results from the training set on the Random Forest model. The Area Under the ROC Curve (AUC) of this model was 0.98. AUC measures how well predictions are ranked. An AUC of 0.98 represents an excellent classifier.

Figure 22: Model 1 Training Results

Model #1: Random Forest - Training Dataset												
Actual	Predicted Class		Totals	Actual	Predicted Class		TP	0.97	TP+TN	1.97	AUC	0.98
Class	0	1		Class	0	1	TN	1.00	Precision	1.00	Sensitivity	0.97
0	11,754	3	11,757	0	1.00	0.00	Type I Error	0.00	Recall	0.97	Specificity	1.00
1	105	3,318	3,423	1	0.03	0.97	Type II Error	0.03	F1	0.98		

Figure 23 shows that when switching to the testing data set, the AUC drops to a 0.66. This means that the model is overfitting to the training data set and will not perform as well on another sample.

Additionally, the Type II error for the testing data set has increased from 0.03 to 0.63. Type II error is a false negative result. This means that the model has increased the number of Defaults predicted to 0 when it should be a 1.

Figure 23: Random Forest Testing Results

Model #1: Random Forest - Testing Dataset												
Actual	Predicted Class		Totals	Actual	Predicted Class		TP	0.37	TP+TN	1.31	AUC	0.66
Class	0	1		Class	0	1	TN	0.94	Precision	0.63	Sensitivity	0.37
0	5,438	328	5,766	0	0.94	0.06	Type I Error	0.06	Recall	0.37	Specificity	0.94
1	988	570	1,558	1	0.63	0.37	Type II Error	0.63	F1	0.51		

Gradient Boosting

Next is the Gradient Boosting model. Boosting is an ensemble learning algorithm to reduce the high variance by averaging lots of models fitted on bootstrapped data samples to avoid overfitting. The model generates 10,000 trees with a learning rate, or shrinkage parameter, of 0.01.

Figure 24 illustrates a feature importance plot. Repayment status in August, max delinquency, maximum amount billed, average payment amount, and ratio of balance to limit in September are the most important features in this model.

Figure 24: Relative Influence Plot

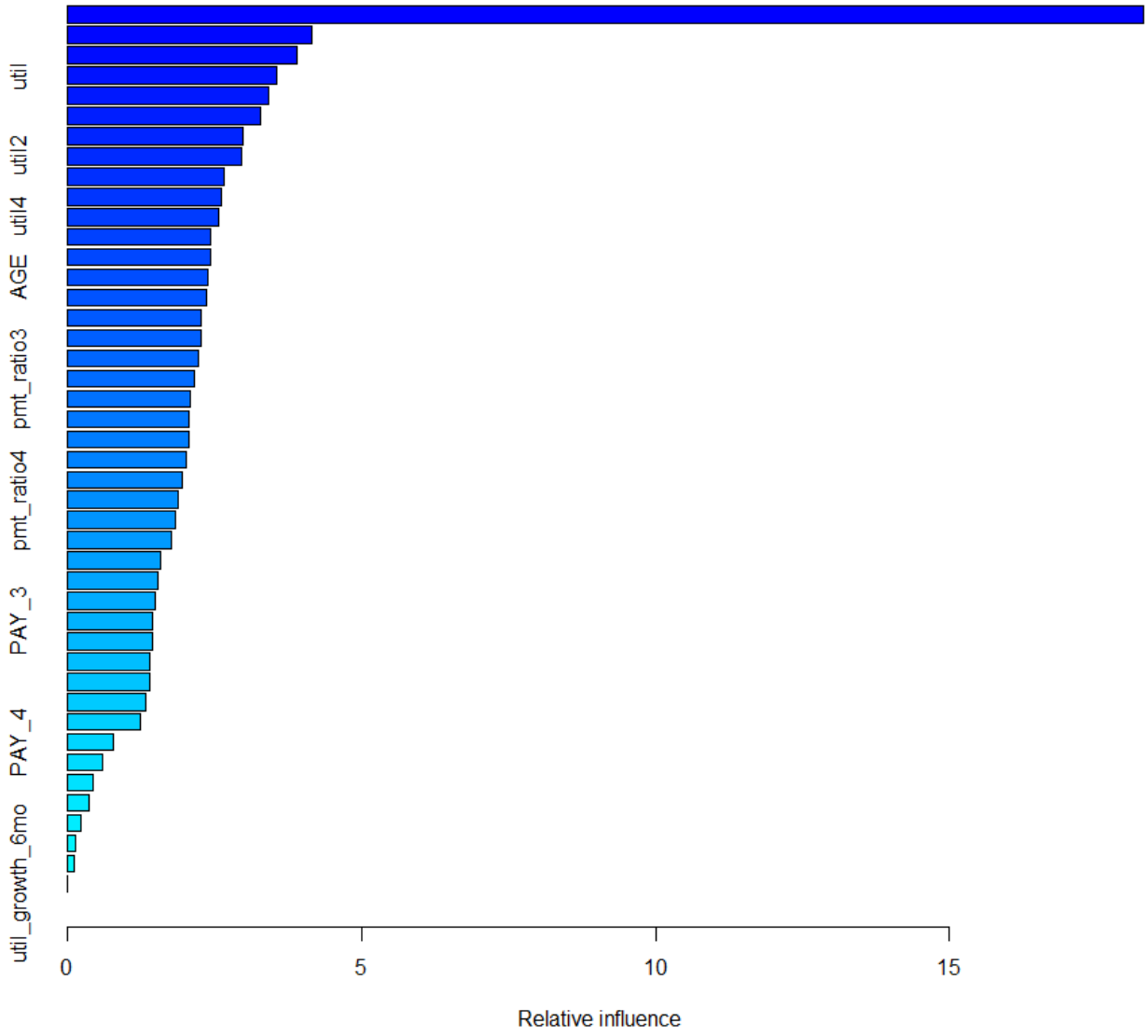


Figure 25 represents the model results from the training dataset for the Gradient Boosting model.

Figure 25:Training data set model results

Model #2: Gradient Boosting Model - Training Dataset												
Actual Class	Predicted Class 0	Predicted Class 1	Totals	Actual Class	Predicted Class 0	Predicted Class 1	TP	0.97	TP+TN	1.97	AUC	0.97
0	11,745	12	11,757	0	1.00	0.00	TN	1.00	Precision	1.00	Sensitivity	0.97
1	100	3,323	3,423	1	0.03	0.97	Type I Error	0.00	Recall	0.97	Specificity	1.00
							Type II Error	0.03	F1	0.98		

The Type II increased from 0% to 6% when switching from the training to testing data set.

Figure 26: Testing data set model results

Model #2: Gradient Boosting Model - Testing Dataset												
Actual	Predicted Class		Totals	Actual	Predicted Class		TP	0.42	TP+TN	1.36	AUC	0.63
Class	0	1		Class	0	1	TN	0.94	Precision	0.65	Sensitivity	0.42
0	5,412	354	5,766	0	0.94	0.06	Type I Error	0.06	Recall	0.42	Specificity	0.94
1	901	657	1,558	1	0.58	0.42	Type II Error	0.58	F1	0.57		

Logistic Regression

After reviewing the most important features from the Gradient Boost and Random Forest models, a simple logistic regression model is created. The following features are included in the model: repayment status in August, max delinquency, average payment amount, ratio of balance to limit in September, interaction of education and age, age, and maximum amount billed. A stepwise selection logistical model by forward, backward, and both directions.

Figure 27: Logistic regression coefficients

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-1.9720	0.1011	-19.5060	< 2e-16	***
PAY_1	0.6588	0.0347	18.9890	< 2e-16	***
max_DLQ	0.3634	0.0259	14.0580	< 2e-16	***
avg_pmt_amt	0.0000	0.0000	-8.5190	< 2e-16	***
age_bin	0.0067	0.0023	2.8970	0.0038	**
util	0.0014	0.0005	2.6290	0.0086	**

The most significant variables based on p-value were the repayment status in August, max delinquency, average payment amount, age, the interaction between age and education, and the ratio of balance to limit in September. Figure 28 represents the model results from the training dataset for the logistic model.

Normally, a stepwise model selection process would require looking into the AIC and BIC results of all models: forwards, backward, and both directions. However, all stepwise selection models resulted in the same model without cutting out any of the 5 selected features.

Figure 28: AIC and BIC Model Results

	AIC	BIC
Forward	13857.72	13903.49
Backward	13857.72	13903.49
Both	13857.72	13903.49

The results from the training data set are shown below. The AUC for this model is lower than the random forest model at 0.64. The Type II error for this model is very high as well, 0.66. The precision of the logistic model is 0.70. Precision is the positive prediction value meaning that about 70% of the estimated Default values of 1 were actually 1. This is also lower than the random forest model.

Figure 29: Train results logistic model

Model #3: Logistic Regression Model - Training Dataset												
Actual	Predicted Class		Totals		Actual	Predicted Class		TP	0.34	TP+TN	1.29	AUC	0.64
Class	0	1			Class	0	1	TN	0.96	Precision	0.70	Sensitivity	0.34
0	11,263	494	11,757		0	0.96	0.04	Type I Error	0.04	Recall	0.34	Specificity	0.96
1	2,271	1,152	3,423		1	0.66	0.34	Type II Error	0.66	F1	0.49		

The precision of the logistic model decreased when using the test data set. All other metrics are very close between the two sample data sets. The Specificity is the proportion of actual negatives that are correctly identified. This means that the Default was 0 and the predicted value was 0. Between the two sample data sets, the Specificity decreased only slightly from 0.96 to 0.95.

Figure 30: Test results logistic model

Model #3: Logistic Regression Model - Testing Dataset												
Actual	Predicted Class		Totals	Actual	Predicted Class		TP	0.35	TP+TN	1.30	AUC	0.65
Class	0	1		Class	0	1	TN	0.95	Precision	0.67	Sensitivity	0.35
0	5,506	260	5,766	0	0.95	0.05	Type I Error	0.05	Recall	0.35	Specificity	0.95
1	1,018	539	1,557	1	0.65	0.35	Type II Error	0.65	F1	0.50		

Support Vector Machine

Next is a supervised learning model. Supervised Vector Machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification. All untransformed variables were included in the Support Vector Machine Model.

Figure 31 shows the decision boundary between the max delinquency and the interaction between age and education. An SVM classification plot is a scatterplot uses the data from the Support Vector Machine Fit to draw a line between the class regions. Picking these two variables, the model is classifying a default on the diagonal line is shown below.

Figure 31: SVM classification between util and education by age

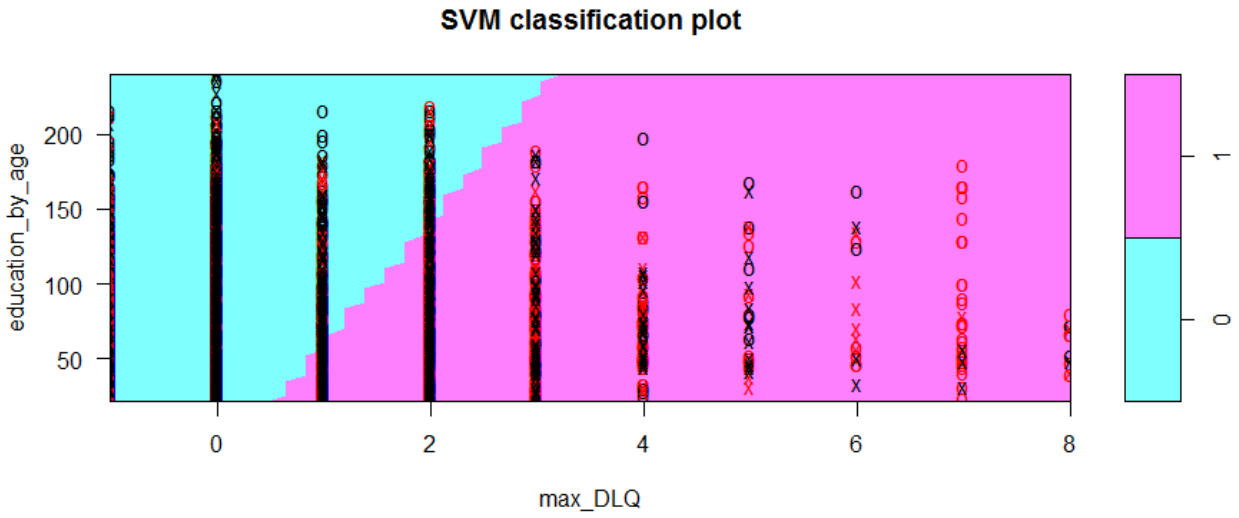


Figure 32 represents the model results from the training dataset for the Support Vector Machine model. The AUC for this model was 0.95.

Figure 32:Training results for SVM model

Model #4: SVM Model - Training Dataset												
Actual	Predicted Class		Totals	Actual	Predicted Class		TP	0.96	TP+TN	1.93	AUC	0.95
Class	0	1		Class	0	1	TN	0.97	Precision	0.91	Sensitivity	0.96
0	11,432	325	11,757	0	0.97	0.03	Type I Error	0.03	Recall	0.96	Specificity	0.97
1	145	3,278	3,423	1	0.04	0.96	Type II Error	0.04	F1	0.96		

The Type II decreased from 3% to 8% when switching from the training to testing data set. The AUC decreased to 0.64.

Figure 33: Testing results for SVM model

Model #4: SVM Model - Testing Dataset												
Actual	Predicted Class		Totals	Actual	Predicted Class		TP	0.39	TP+TN	1.31	AUC	0.64
Class	0	1		Class	0	1	TN	0.92	Precision	0.57	Sensitivity	0.39
0	5,312	454	5,766	0	0.92	0.08	Type I Error	0.08	Recall	0.39	Specificity	0.92
1	954	604	1,558	1	0.61	0.39	Type II Error	0.61	F1	0.53		

Naive Bayes

The final model is a Naïve Bayes Model. A Naïve Bayes classifier uses a family of probabilistic classifiers that use Bayes theorem and strong independence assumptions between the features. The following features are included in the model: repayment status in August, max delinquency, average payment amount, ratio of balance to limit in September, interaction of education and age, age, and maximum amount billed.

Figure 34: Density Plot or Pay_1

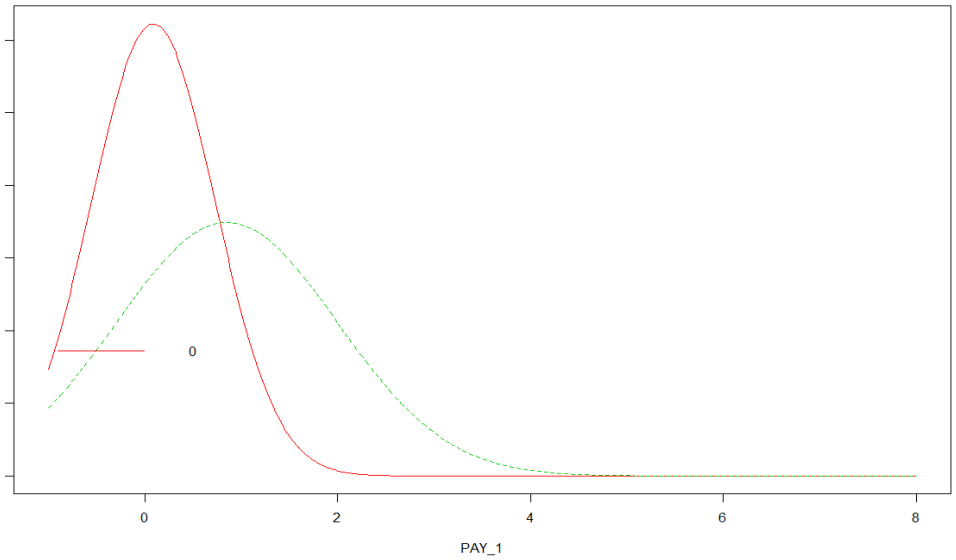


Figure 34 illustrates a density plot of repayment status in August. The plot shows that the default would not likely happen if the customer has missed a payment in the last couple of months. Figure 35 illustrates the results from the training dataset for the Naïve Bayes model.

The AUC dropped significantly compared to other metrics and the Type 1 Error is very high. Similarly, the precision is much lower than other models. The F1 score is the harmonic average between the precision and recall. The F1 score is best when it is closest to 1, so the 0.68.

Figure 35: Naive Bayes Training Dataset

Model #5: Naïve Bayes Model - Training Dataset												
Actual	Predicted Class		Totals	Actual	Predicted Class		TP	0.96	TP+TN	1.07	AUC	0.53
Class	0	1		Class	0	1	TN	0.11	Precision	0.24	Sensitivity	0.96
0	1,304	10,453	11,757	0	0.11	0.89	Type I Error	0.89	Recall	0.96	Specificity	0.11
1	123	3,300	3,423	1	0.04	0.96	Type II Error	0.04	F1	0.68		

The table below shows the results the testing data in the Naïve Bayes model. The results are almost identical between the training and testing datasets. Therefore, the model is consistent, but not that accurate.

Figure 36: Naive Bayes Testing Dataset

Model #5: Naïve Bayes Model - Testing Dataset												
Actual Class	Predicted Class		Totals	Actual Class	Predicted Class		TP	0.96	TP+TN	1.07	AUC	0.53
	0	1			0	1	TN	0.11	Precision	0.23	Sensitivity	0.96
0	658	5,108	5,766	0	0.11	0.89	Type I Error	0.89	Recall	0.96	Specificity	0.11
1	61	1,496	1,557	1	0.04	0.96	Type II Error	0.04	F1	0.68		

Comparison of Results

Figure 33 exemplifies the results of the four models. Examining these results, it seems Random Forest is the best model between these variables.

Figure 37: Results Table

Model #1: Random Forest - Testing Dataset					
TP	0.37	TP+TN	1.31	AUC	0.66
TN	0.94	Precision	0.63	Sensitivity	0.37
Type I Error	0.06	Recall	0.37	Specificity	0.94
Type II Error	0.63	F1	0.51		
Model #2: Gradient Boosting Model - Testing Dataset					
TP	0.42	TP+TN	1.36	AUC	0.63
TN	0.94	Precision	0.65	Sensitivity	0.42
Type I Error	0.06	Recall	0.42	Specificity	0.94
Type II Error	0.58	F1	0.57		
Model #3: Logistic Regression Model - Testing Dataset					
TP	0.35	TP+TN	1.30	AUC	0.65
TN	0.95	Precision	0.67	Sensitivity	0.35
Type I Error	0.05	Recall	0.35	Specificity	0.95
Type II Error	0.65	F1	0.50		
Model #4: SVM Model - Testing Dataset					
TP	0.39	TP+TN	1.31	AUC	0.64
TN	0.92	Precision	0.57	Sensitivity	0.39
Type I Error	0.08	Recall	0.39	Specificity	0.92
Type II Error	0.61	F1	0.53		
Model #5: Naïve Bayes Model - Testing Dataset					
TP	0.96	TP+TN	1.07	AUC	0.53
TN	0.11	Precision	0.23	Sensitivity	0.96
Type I Error	0.89	Recall	0.96	Specificity	0.11
Type II Error	0.04	F1	0.68		

Model #1: Random Forest - Training Dataset					
TP	0.97	TP+TN	1.97	AUC	0.98
TN	1.00	Precision	1.00	Sensitivity	0.97
Type I Error	0.00	Recall	0.97	Specificity	1.00
Type II Error	0.03	F1	0.98		
Model #2: Gradient Boosting Model - Training Dataset					
TP	0.97	TP+TN	1.97	AUC	0.97
TN	1.00	Precision	1.00	Sensitivity	0.97
Type I Error	0.00	Recall	0.97	Specificity	1.00
Type II Error	0.03	F1	0.98		
Model #3: Logistic Regression Model - Training Dataset					
TP	0.34	TP+TN	1.29	AUC	0.64
TN	0.96	Precision	0.70	Sensitivity	0.34
Type I Error	0.04	Recall	0.34	Specificity	0.96
Type II Error	0.66	F1	0.49		
Model #4: SVM Model - Training Dataset					
TP	0.96	TP+TN	1.93	AUC	0.95
TN	0.97	Precision	0.91	Sensitivity	0.96
Type I Error	0.03	Recall	0.96	Specificity	0.97
Type II Error	0.04	F1	0.96		
Model #5: Naïve Bayes Model - Training Dataset					
TP	0.96	TP+TN	1.07	AUC	0.53
TN	0.11	Precision	0.24	Sensitivity	0.96
Type I Error	0.89	Recall	0.96	Specificity	0.11
Type II Error	0.04	F1	0.68		

Conclusion

The five models in this report were Random Forest, Gradient Boosting, Logistic, Support Vector Machines, and Naïve Bayes classification models to detect if a customer will default on their loan. With more time, the features going into the model should be tweaked to improve the performance of each of the models. For example, the features in the logistic and Naïve Bayes models were selected based on the variable importance selection plots from the Gradient Boosting and Random Forest models. With more time, additional variables should included to test if this improves the performance metrics of the model.

Bibliography

Thomas, L. C. *Consumer Credit Models: Pricing, Profit, and Portfolios*. Oxford University Press, 2009.

Appendix: R Code:

```
# Lauren Camero


# 06.17.2019

################################################################################ LOAD PACKAGES

# install.packages("randomForest")
# install.packages("caret")
# install.packages("lattice")
# install.packages("gbm")
# install.packages("e1071")
# install.packages("jtools")
library(corrplot)
library(ggplot2)
library(randomForest)
library(lattice)
library(ggplot2)
library(caret)
library(rpart)
library(gbm)
library(MASS)
library(e1071)
library(jtools)
library(ROCR)
library(naivebayes)
library(e1071)
library(jtools)
library(PerformanceAnalytics)
library('OneR')
library('woeBinning')
# install.packages("flux")
library(flux)
# install.packages("pROC")
library(pROC)
library(PRROC)

################################################################################ LOAD DATA

# set up the file path
my.path <- 'C:\\Users\\lcamero\\Downloads\\';
my.file <- paste(my.path,'credit_card_default.RData',sep='');


# Read the RData object using readRDS();
credit_card_default <- readRDS(my.file)
cc <- credit_card_default

################################################################################ EXPORE DATA

# examine the data
str(cc)
table(cc$data.group)
head(cc)
summary(cc)

# check descriptive statistics
table(cc$SEX)
table(cc$EDUCATION)
table(cc$MARRIAGE)
summary(cc$AGE)
table(cc$PAY_0)
```

```
table(cc$PAY_2)
table(cc$PAY_3)
table(cc$PAY_4)
table(cc$PAY_5)
table(cc$PAY_6)
table(cc$DEFAULT)
summary(cc$BILL_AMT1)
summary(cc$BILL_AMT2)
summary(cc$BILL_AMT3)
summary(cc$BILL_AMT4)
summary(cc$BILL_AMT4)
summary(cc$BILL_AMT6)
summary(cc$PAY_AMT1)
summary(cc$PAY_AMT2)
summary(cc$PAY_AMT3)
summary(cc$PAY_AMT4)
summary(cc$PAY_AMT5)
summary(cc$PAY_AMT6)
summary(cc$LIMIT_BAL)


################################################################################ FEATURE ENGINEERING

# Correct categrorical data
cc$EDUCATION <- ifelse(cc$EDUCATION == 0, 4,
                  ifelse(cc$EDUCATION == 5, 4,
                      ifelse(cc$EDUCATION == 6, 4,cc$EDUCATION)))
cc$MARRIAGE <- ifelse(cc$MARRIAGE == 0, 3,cc$MARRIAGE)
cc$PAY_0 <- ifelse(cc$PAY_0 == -2, -1,
              ifelse(cc$PAY_0 == -1, 0, cc$PAY_0))
cc$PAY_2 <- ifelse(cc$PAY_2 == -2, -1,
              ifelse(cc$PAY_2 == -1, 0, cc$PAY_2))
cc$PAY_3 <- ifelse(cc$PAY_3 == -2, -1,
              ifelse(cc$PAY_3 == -1, 0, cc$PAY_3))
cc$PAY_4 <- ifelse(cc$PAY_4 == -2, -1,
              ifelse(cc$PAY_4 == -1, 0, cc$PAY_4))
cc$PAY_5 <- ifelse(cc$PAY_5 == -2, -1,
              ifelse(cc$PAY_5 == -1, 0, cc$PAY_5))
cc$PAY_6 <- ifelse(cc$PAY_6 == -2, -1,
              ifelse(cc$PAY_6 == -1, 0, cc$PAY_6))
colnames(cc)[colnames(cc)=="PAY_0"] <- "PAY_1"

################################################################################ BINNING MODEL

# Bin age using woe.tree.binning
library(woeBinning)
library(OneR)

cc$target <- abs(as.numeric(cc$DEFAULT)-2);
age.tree <- woe.tree.binning(df=cc,target.var=c('target'),pred.var=c('AGE'))

# WOE plot for age bins;
woe.binning.plot(age.tree)
# Note that we got different bins;

# Score bins on data frame;
tree.df <- woe.binning.deploy(df=cc,binning=age.tree)
head(tree.df)
table(tree.df$age.in.years.binned)

# See the WOE Binning Table
woe.binning.table(age.tree)

#rewrite AGE with new bins
cc$age_bin <- ifelse(cc$AGE <= 25, 25,
                ifelse(cc$AGE <= 35, 35,
```

```
                ifelse(cc$AGE <= 45, 45, 55)))

############################################################################## CALCULATE FEATURES

# average bill amount over 6 months
cc$avg_bill_amt <- (cc$BILL_AMT1 + cc$BILL_AMT2 + cc$BILL_AMT3 + cc$BILL_AMT4 + cc$BILL_AMT5 + cc$BILL_AMT6)/6

# average payment amount
cc$avg_pmt_amt <- (cc$PAY_AMT1 + cc$PAY_AMT2 + cc$PAY_AMT3 + cc$PAY_AMT4 + cc$PAY_AMT5 + cc$PAY_AMT6)/6

# Payment Ratio
cc$pmt_ratio1 <- cc$PAY_AMT1/(ifelse(cc$BILL_AMT2 == 0,1,cc$BILL_AMT2))
cc$pmt_ratio2 <- cc$PAY_AMT2/(ifelse(cc$BILL_AMT3 == 0,1,cc$BILL_AMT3))
cc$pmt_ratio3 <- cc$PAY_AMT3/(ifelse(cc$BILL_AMT4 == 0,1,cc$BILL_AMT4))
cc$pmt_ratio4 <- cc$PAY_AMT4/(ifelse(cc$BILL_AMT5 == 0,1,cc$BILL_AMT5))
cc$pmt_ratio5 <- cc$PAY_AMT5/(ifelse(cc$BILL_AMT6 == 0,1,cc$BILL_AMT6))

# Average Payment Ratio
cc$avg_pmt_ratio <- (cc$pmt_ratio1 + cc$pmt_ratio2 + cc$pmt_ratio3 + cc$pmt_ratio4 + cc$pmt_ratio5)/5

# fix payment ratio nulls to 100
cc[is.na(cc)] <- 100

# max bill amount
cc$max_bill_amt <- pmax(cc$BILL_AMT1,cc$BILL_AMT2,cc$BILL_AMT3,cc$BILL_AMT4,cc$BILL_AMT5,cc$BILL_AMT6)

# max payment amount
cc$max_pmt_amt <- pmax(cc$PAY_AMT1,cc$PAY_AMT2,cc$PAY_AMT3,cc$PAY_AMT4,cc$PAY_AMT5,cc$PAY_AMT6)

# Utilization
cc$util <- cc$BILL_AMT1/cc$LIMIT_BAL
cc$util2 <- cc$BILL_AMT2/cc$LIMIT_BAL
cc$util3 <- cc$BILL_AMT3/cc$LIMIT_BAL
cc$util4 <- cc$BILL_AMT4/cc$LIMIT_BAL
cc$util5 <- cc$BILL_AMT5/cc$LIMIT_BAL
cc$util6 <- cc$BILL_AMT6/cc$LIMIT_BAL

# average utilization
cc$avg_util <- (cc$util + cc$util2 + cc$util3 + cc$util4 + cc$util5 + cc$util6)/6

# balance growth over 6 months and convert to binary
cc$bal_growth_6mo <- cc$BILL_AMT6 > cc$BILL_AMT1
cc$bal_growth_6mo <- 1 * cc$bal_growth_6mo

# utilization growth over 6 months and convert ot binary
cc$util_growth_6mo <- cc$util6 > cc$util
cc$util_growth_6mo <- 1 * cc$util_growth_6mo

# max delinquency
cc$max_DLQ <- pmax(cc$PAY_1,cc$PAY_2,cc$PAY_3,cc$PAY_4,cc$PAY_5,cc$PAY_6)

# scale the utilization
summary(cc$util)
cc$util <- cc$util*100
cc$util2 <- cc$util2*100
cc$util3 <- cc$util3*100
cc$util4 <- cc$util4*100
cc$util5 <- cc$util5*100
cc$util6 <- cc$util6*100

# calculate minimum payment ratio
cc$min_pmt_ratio <- pmin(cc$pmt_ratio1, cc$pmt_ratio2, cc$pmt_ratio3, cc$pmt_ratio4, cc$pmt_ratio5)

# Create calculated field based on the interaction between age and education
cc$education_by_age <- cc$EDUCATION*cc$AGE
```

```
################################################################################ EDA

# summary statistics of new variables
summary(cc$avg_bill_amt)
summary(cc$avg_pmt_amt)
summary(cc$pmt_ratio1)
summary(cc$pmt_ratio2)
summary(cc$pmt_ratio3)
summary(cc$pmt_ratio4)
summary(cc$pmt_ratio5)
summary(cc$avg_pmt_ratio)
summary(cc$max_bill_amt)
summary(cc$max_pmt_amt)
summary(cc$util)
summary(cc$util2)
summary(cc$util3)
summary(cc$util4)
summary(cc$util5)
summary(cc$util6)
summary(cc$avg_util)
summary(cc$bal_growth_6mo)
summary(cc$util_growth_6mo)
summary(cc$max_DLQ)
summary(cc$min_pmt_ratio)
summary(cc$education_by_age)

# use stargazer to
# out.path <- 'C:\\Users\\lcamero\\Downloads\\';
# file.name <- 'summary_statistics.html';
# stargazer(cc, type=c('html'),out=paste(out.path,file.name,sep=''),
#         title=c('Summary Statistics'),
#         align=TRUE, digits=2, digits.extra=2, initial.zero=TRUE, median=TRUE)


# EDA
# # recheck descriptive statistics of changed variables
# table(cc$EDUCATION)
# table(cc$MARRIAGE)
# table(cc$PAY_1)
# table(cc$PAY_2)
# table(cc$PAY_3)
# table(cc$PAY_4)
# table(cc$PAY_5)
# table(cc$PAY_6)
# table(cc$DEFAULT)

################################################################################ TRANSFORM VARIABLES

# review the boxplots
par(mfrow=c(5,6))
boxplot(cc$LIMIT_BAL, main = "LIMIT_BAL")
boxplot(cc$BILL_AMT1, main = "BILL_AMT1")
boxplot(cc$BILL_AMT2, main = "BILL_AMT2")
boxplot(cc$BILL_AMT3, main = "BILL_AMT3")
boxplot(cc$BILL_AMT4, main = "BILL_AMT4")
boxplot(cc$BILL_AMT5, main = "BILL_AMT5")
boxplot(cc$BILL_AMT6, main = "BILL_AMT6")
boxplot(cc$PAY_AMT1, main = "PAY_AMT1")
boxplot(cc$PAY_AMT2, main = "PAY_AMT2")
boxplot(cc$PAY_AMT3, main = "PAY_AMT3")
boxplot(cc$PAY_AMT4, main = "PAY_AMT4")
boxplot(cc$PAY_AMT5, main = "PAY_AMT5")
boxplot(cc$PAY_AMT6, main = "PAY_AMT6")
boxplot(cc$avg_bill_amt, main = "Average Bill Amount")
boxplot(cc$avg_pmt_amt, main = "Average Payment Amount")
boxplot(cc$pmt_ratio1, main = "Payment Ratio")
```

```
boxplot(cc$max_bill_amt, main = "Max Bill Amount")
boxplot(cc$max_pmt_amt, main = "Max Payment Amount")
boxplot(cc$util, main = "Utilization")
boxplot(cc$avg_util, main = "Average Utilization")
boxplot(cc$bal_growth_6mo, main = "Balance Growth")
boxplot(cc$max_DLQ, main = "Maximum Delinquency")
boxplot(cc$avg_pmt_ratio, main = "Average Payment Ratio")
boxplot(cc$min_pmt_ratio, main = "Minimum Payment Ratio")
boxplot(cc$education_by_age, main = "Education x Age")
boxplot(cc$util_growth_6mo, main = "Utilization Growth")

# review the log of the boxplots
par(mfrow=c(5,5))
boxplot(log(cc$LIMIT_BAL), main = "LIMIT_BAL")
boxplot(log(cc$BILL_AMT1), main = "BILL_AMT1")
boxplot(log(cc$BILL_AMT2), main = "BILL_AMT2")
boxplot(log(cc$BILL_AMT3), main = "BILL_AMT3")
boxplot(log(cc$BILL_AMT4), main = "BILL_AMT4")
boxplot(log(cc$BILL_AMT5), main = "BILL_AMT5")
boxplot(log(cc$BILL_AMT6), main = "BILL_AMT6")
boxplot(log(cc$PAY_AMT1), main = "PAY_AMT1")
boxplot(log(cc$PAY_AMT2), main = "PAY_AMT2")
boxplot(log(cc$PAY_AMT3), main = "PAY_AMT3")
boxplot(log(cc$PAY_AMT4), main = "PAY_AMT4")
boxplot(log(cc$PAY_AMT5), main = "PAY_AMT5")
boxplot(log(cc$PAY_AMT6), main = "PAY_AMT6")
boxplot(log(cc$avg_bill_amt), main = "Average Bill Amount")
boxplot(log(cc$avg_pmt_amt), main = "Average Payment Amount")
boxplot(log(cc$pmt_ratio1), main = "Payment Ratio")
boxplot(log(cc$max_bill_amt), main = "Max Bill Amount")
boxplot(log(cc$max_pmt_amt), main = "Max Payment Amount")
boxplot(log(cc$util), main = "Utilization")
boxplot(log(cc$avg_util), main = "Average Utilization")
boxplot(log(cc$max_DLQ), main = "Maximum Delinquency")
boxplot(sqrt(cc$min_pmt_ratio), main = "Minimum Payment Ratio")
boxplot(log(cc$education_by_age), main = "Education x Age")


# transform the variables into the log
cc$LIMIT_BAL_log <- log(cc$LIMIT_BAL)
cc$AGE_log <- log(cc$AGE)
cc$BILL_AMT1_log <- log(cc$BILL_AMT1)
cc$BILL_AMT2_log <- log(cc$BILL_AMT2)
cc$BILL_AMT3_log <- log(cc$BILL_AMT3)
cc$BILL_AMT4_log <- log(cc$BILL_AMT4)
cc$BILL_AMT5_log <- log(cc$BILL_AMT5)
cc$BILL_AMT6_log <- log(cc$BILL_AMT6)
cc$PAY_AMT1_log <- log(cc$PAY_AMT1)
cc$PAY_AMT2_log <- log(cc$PAY_AMT2)
cc$PAY_AMT3_log <- log(cc$PAY_AMT3)
cc$PAY_AMT4_log <- log(cc$PAY_AMT4)
cc$PAY_AMT5_log <- log(cc$PAY_AMT5)
cc$PAY_AMT6_log <- log(cc$PAY_AMT6)
cc$avg_bill_amt_log <- log(cc$avg_bill_amt)
cc$avg_pmt_amt_log <- log(cc$avg_pmt_amt)
cc$pmt_ratio1_log <- log(cc$pmt_ratio1)
cc$max_bill_amt_log <- log(cc$max_bill_amt)
cc$max_pmt_amt_log <- log(cc$max_pmt_amt)
cc$util_log <- log(cc$util)
cc$avg_util_log <- log(cc$avg_util)
cc$max_DLQ_log <- log(cc$max_DLQ)
cc$min_pmt_ratio_sqrt <- sqrt(cc$min_pmt_ratio)
cc$education_by_age_log <- log(cc$education_by_age)

################################################################################ VISUALIZE VARIABLES
```

```
plot(cc$AGE~ cc$LIMIT_BAL)
p<-ggplot(cc, aes(x=LIMIT_BAL)) +
 geom_histogram(color="black", fill="white")
p
p<-ggplot(cc, aes(x=BILL_AMT1)) +
 geom_histogram(color="black", fill="white")
p
neg_bill <- subset(cc, cc$BILL_AMT1 <= -1)
p<-ggplot(neg_bill, aes(x=BILL_AMT1)) +
 geom_histogram(color="black", fill="white")
p
p<-ggplot(cc, aes(x=SEX, y=BILL_AMT1)) +
 geom_boxplot()
p
summary(cc$LIMIT_BAL)
cc$balance_6 <- -cc$BILL_AMT6 + cc$PAY_AMT6
cc$balance_5 <- -cc$BILL_AMT5 + cc$PAY_AMT5 + cc$balance_6
cc$balance_4 <- -cc$BILL_AMT4 + cc$PAY_AMT4 + cc$balance_5
cc$balance_3 <- -cc$BILL_AMT3 + cc$PAY_AMT3 + cc$balance_4
cc$balance_2 <- -cc$BILL_AMT2 + cc$PAY_AMT2 + cc$balance_3
cc$balance_1 <- -cc$BILL_AMT1 + cc$PAY_AMT1 + cc$balance_2
cc$above_bal <- cc$LIMIT_BAL < cc$BILL_AMT1
sum(cc$train)
sum(cc$test)
sum(cc$validate)
cc$DEFAULT <- as.factor(cc$DEFAULT)

############################################################################## DISCRETE BINNING

# oneR binning for Limit Balance
options(scipen = 999)
bin.4 <- optbin(cc$DEFAULT ~ cc$LIMIT_BAL,method=c('logreg'));
table(bin.4)
aggregate(cc$DEFAULT, by=list(LIMIT_BAL=bin.4[,1]), FUN=mean)
cc$LIMiT_BAL_bin <- ifelse(cc$LIMIT_BAL >= 130000,1,0)

############################################################################## Correlation Matrix

cc_cor <- cor(cc[, (names(cc) %in% c("age_bin","avg_bill_amt","avg_pmt_amt","pmt_ratio1","avg_pmt_ratio"
                            ,"max_bill_amt","max_pmt_amt","util","avg_util","bal_growth_6mo"
                            ,"util_growth_6mo","max_DLQ","min_pmt_ratio","education_by_age"))])
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
# corrplot(cc_cor, method = "number", type = "upper", col = col(200), tl.col = "black", tl.srt = 45, diag = FALSE)

# use a scatter plot matrix to observe the correlations
cc_sp <- cc[, (names(cc) %in% c("DEFAULT", "avg_bill_amt","avg_pmt_amt","avg_pmt_ratio","avg_util","min_pmt_ratio"))]
pairs(cc_sp[,1:5],
    pch = 21,
    lower.panel = NULL,
    bg = c("orange", "dark grey")
    [unclass(cc_sp$DEFAULT)])

# Giant corrplot table
# par(mfrow=c(1,1))
# cc.cor <- cor(cc[c(2,6,7,13,14,15,16,17,18,19,20,21,22,23,24,25,33,34,35,40,41,42,43,49,50,51)])
# corrplot(cc.cor)
# names(cus) <- toupper(names(cus))
# cus_cor <- cor(cus[, !(names(cus) %in% c("FLT_ORIG_DT",
"AIRPORT_PROCESS","INFLIGHT_CREW","AIRPORT_STAFF","INFLIGHT_SERVICE","OPD_FLT_IND"))])
# col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
# corrplot(cus_cor, method = "number", type = "upper", col = col(200), tl.col = "black", tl.srt = 45, diag = FALSE)
# out.path <- 'C:\\Users\\lcamero\\Downloads\\';
# file.name <- 'corr_matrix.html';
# cor.matrix <- cc.cor
# stargazer(cor.matrix, type=c('html'),out=paste(out.path,file.name,sep=''),
#         align=TRUE, digits=2, title='Correlation Matrix')
```

```
################################################################################ MODEL

# pick the model that is only the training data set
cc_train <- subset(cc, train == 1)
cc_validate <- subset(cc, validate == 1)
cc_test <- subset(cc, test == 1)

################################################################################ RANDOM FOREST

# try a random forest
ranfor <- randomForest(DEFAULT ~ LIMIT_BAL + SEX + EDUCATION + MARRIAGE
                + AGE + PAY_1 + PAY_2 + PAY_3 + PAY_4 + PAY_5 + PAY_6
                + BILL_AMT1 + BILL_AMT2 + BILL_AMT3 + BILL_AMT4 + BILL_AMT5
                + BILL_AMT6 + PAY_AMT1 + PAY_AMT2 + PAY_AMT3 + PAY_AMT4
                + PAY_AMT5 + PAY_AMT6 + age_bin + avg_bill_amt + avg_pmt_amt + pmt_ratio1
                + pmt_ratio2 + pmt_ratio3 + pmt_ratio4 + pmt_ratio5 + avg_pmt_ratio
                + max_bill_amt + max_pmt_amt + util + util2 + util3 + util4 + util5 + util6 + avg_util
                + bal_growth_6mo + util_growth_6mo + max_DLQ + min_pmt_ratio + education_by_age + LIMiT_BAL_bin
                , data = cc_train, importance = TRUE)

# show model results
ranfor
summary(ranfor)

# calculate the expected values
cc_train$DEFAULT_ranfor <- predict(ranfor, cc_train)

library(dplyr)
true_pos_ranfor <- cc_train %>%  filter(DEFAULT == 1)
true_pos_ranfor <- true_pos_ranfor %>%  filter(as.numeric(DEFAULT_ranfor) == 1)
false_pos_ranfor <- true_pos_ranfor %>%  filter(DEFAULT_ranfor == 0)
true_neg_ranfor <- cc_train %>%  filter(DEFAULT == 0)
true_neg_ranfor <- true_neg_ranfor %>%  filter(DEFAULT_ranfor == 0)
false_neg_ranfor <- true_neg_ranfor %>%  filter(DEFAULT_ranfor == 1)

nrow(true_pos_ranfor)
nrow(false_pos_ranfor)
nrow(true_neg_ranfor)
nrow(false_neg_ranfor)

# create a varibale importance plot
(VI_F=importance(ranfor))
varImp(ranfor)
varImpPlot(ranfor,type=2)
cc_train$DEFAULT_ranfor <- predict(ranfor, cc_train)
PRROC_obj <- roc.curve(scores.class0 = as.numeric(cc_train$DEFAULT), weights.class0=as.numeric(cc_train$DEFAULT_ranfor),
                curve=TRUE)
plot(PRROC_obj)
roc_obj <- roc((1-cc_test$DEFAULT),(1-cc_test$xDEFAULT))
auc(roc_obj)

# Build some example data
# Observed (truth) data as presence-absence (1-0)
# Predicted data as values ranging from 0 to 1

# Install the ROCR package
library('ROCR')

# AUC function
fun.auc <- function(pred,obs){
  # Run the ROCR functions for AUC calculation
  ROC_perf <- performance(prediction(pred,obs),"tpr","fpr")
  ROC_sens <- performance(prediction(pred,obs),"sens","spec")
  ROC_err <- performance(prediction(pred, labels=obs),"err")
  ROC_auc <- performance(prediction(pred,obs),"auc")
```

```
 # AUC value
 AUC <- ROC_auc@y.values[[1]] # AUC
 # Mean sensitivity across all cutoffs
 x.Sens <- mean(as.data.frame(ROC_sens@y.values)[,1])
 # Mean specificity across all cutoffs
 x.Spec <- mean(as.data.frame(ROC_sens@x.values)[,1])
 # Sens-Spec table to estimate threshold cutoffs
 SS <- data.frame(SENS=as.data.frame(ROC_sens@y.values)[,1],SPEC=as.data.frame(ROC_sens@x.values)[,1])
 # Threshold cutoff with min difference between Sens and Spec
 SS_min_dif <- ROC_perf@alpha.values[[1]][which.min(abs(SS$SENS-SS$SPEC))]
 # Threshold cutoff with max sum of Sens and Spec
 SS_max_sum <- ROC_perf@alpha.values[[1]][which.max(rowSums(SS[c("SENS","SPEC")]))]
 # Min error rate
 Min_Err <- min(ROC_err@y.values[[1]])
 # Threshold cutoff resulting in min error rate
 Min_Err_Cut <- ROC_err@x.values[[1]][which(ROC_err@y.values[[1]]==Min_Err)][1]
 # Kick out the values
 round(cbind(AUC,x.Sens,x.Spec,SS_min_dif,SS_max_sum,Min_Err,Min_Err_Cut),3)
}

# Run the function with the example data
observations <- as.numeric(cc_train$DEFAULT)
predictions <- as.numeric(predict(ranfor, cc_train))
fun.auc(predictions, observations)

# Run the function with the example data
observations <- as.numeric(cc_test$DEFAULT)
predictions <- as.numeric(predict(ranfor, cc_test))
fun.auc(predictions, observations)

################################################################################ GRADIENT BOOSTING

# run a gradient boosting model
boost = gbm(DEFAULT ~ LIMIT_BAL + SEX + EDUCATION + MARRIAGE
        + AGE + PAY_1 + PAY_2 + PAY_3 + PAY_4 + PAY_5 + PAY_6
        + BILL_AMT1 + BILL_AMT2 + BILL_AMT3 + BILL_AMT4 + BILL_AMT5
        + BILL_AMT6 + PAY_AMT1 + PAY_AMT2 + PAY_AMT3 + PAY_AMT4
        + PAY_AMT5 + PAY_AMT6 + age_bin + avg_bill_amt + avg_pmt_amt + pmt_ratio1
        + pmt_ratio2 + pmt_ratio3 + pmt_ratio4 + pmt_ratio5 + avg_pmt_ratio
        + max_bill_amt + max_pmt_amt + util + util2 + util3 + util4 + util5 + util6 + avg_util
        + bal_growth_6mo + util_growth_6mo + max_DLQ + min_pmt_ratio + education_by_age + LIMiT_BAL_bin
        ,data = cc_train, distribution = "gaussian",n.trees = 10000
        , shrinkage = 0.01, interaction.depth = 4)

# show model results
boost
summary(boost)

# calculate the expected values
cc_train$xdefault2 <- predict(boost, cc_train,n.trees = 10000)
cc_test$xdefault2 <- predict(boost, cc_test,n.trees = 10000)
cc_validate$xdefault2 <- predict(boost, cc_validate,n.trees = 10000)

# check the actual default
table(cc_train$xdefault2)
table(cc_test$xdefault2)
table(cc_validate$xdefault2)

cc_train$DEFAULT_boost <- predict(ranfor, cc_train)
PRROC_obj <- roc.curve(scores.class0 = as.numeric(cc_train$DEFAULT_ranfor), weights.class0=as.numeric(cc_train$DEFAULT),
                curve=TRUE)
plot(PRROC_obj)

# Run the function with the example data
observations <- as.numeric(cc_train$DEFAULT)
predictions <- as.numeric(predict(boost, cc_train))
```

```
fun.auc(predictions, observations)

# Run the function with the example data
observations <- as.numeric(cc_test$DEFAULT)
predictions <- as.numeric(predict(boost, cc_test))
fun.auc(predictions, observations)

################################################################################ LOGISTIC REGRESSION

# run logistic regression
logis <- glm(DEFAULT ~ PAY_1 + max_bill_amt + max_DLQ + util + avg_pmt_amt
        + age_bin + education_by_age, data = cc_train, family = binomial())

# check logistic regression results of the full model
summary(logis)

# use variable selection method: stepwise
# Define the upper model as the FULL model
upper.lm <- glm(DEFAULT ~ PAY_1 + max_bill_amt + max_DLQ + util + avg_pmt_amt
          + age_bin + education_by_age, data = cc_train, family = binomial());

# check the full model results
summary(upper.lm)

# Define the lower model as the Intercept model
lower.lm <- glm(DEFAULT ~ 1,data = cc_train, family = binomial());

# check the summary of the lower model
summary(lower.lm)

# run stepwise model regression
forward.lm <- stepAIC(object=lower.lm,scope=list(upper=formula(upper.lm),lower=~1),
                direction=c('forward'), family = binomial());

# check results of forward
summary(forward.lm)

# create backward model
backward.lm <- stepAIC(object=upper.lm,direction=c('backward'), family = binomial());
summary(backward.lm)

# run models in both directions
stepwise.lm <- stepAIC(object=lower.lm,scope=list(upper=formula(upper.lm),lower=~1),
                 direction=c('both'), family = binomial());

# check model of both direction variable selection
summary(stepwise.lm)

AIC(forward.lm)
AIC(backward.lm)
AIC(stepwise.lm)
BIC(forward.lm)

# calculate theforward.lm expected values
cc_train$xdefault2 <- as.factor(round(predict.glm(stepwise.lm, cc_train, type = "response")))
cc_test$xdefault2 <- round(round(predict(stepwise.lm, cc_test, type = "response")))
cc_validate$xdefault2 <- round(round(predict(stepwise.lm, cc_validate, type = "response")))

# check the actual default
table(cc_train$DEFAULT)
table(cc_train$xdefault2)
table(cc_test$DEFAULT)
table(cc_test$xdefault2)
table(cc_validate$DEFAULT)
table(cc_validate$xdefault2)
```

```
############################################################################## SVM

# run an svm model
svm = svm(DEFAULT ~ PAY_1 + max_bill_amt + max_DLQ + util + avg_pmt_amt
      + age_bin + education_by_age
      , data = cc_train, kernel = "linear", cost = 10, scale = FALSE)

?svm
# show svm model
print(svm)

# plot the svm results
plot(svm, data = cc_train, education_by_age ~ max_DLQ)

# calculate the expected values
xdefault3_train = predict(svm, cc_train)
xdefault3_test = predict(svm, cc_test)
xdefault3_validate = predict(svm, cc_validate)

############################################################################## NAIVE BAYES

library(naivebayes)
nb <- naive_bayes(DEFAULT ~ LIMIT_BAL + SEX + EDUCATION + MARRIAGE
            + AGE + PAY_1 + PAY_2 + PAY_3 + PAY_4 + PAY_5 + PAY_6
            + BILL_AMT1 + BILL_AMT2 + BILL_AMT3 + BILL_AMT4 + BILL_AMT5
            + BILL_AMT6 + PAY_AMT1 + PAY_AMT2 + PAY_AMT3 + PAY_AMT4
            + PAY_AMT5 + PAY_AMT6 + age_bin + avg_bill_amt + pmt_ratio1
            + pmt_ratio2 + pmt_ratio3 + pmt_ratio4 + pmt_ratio5 + avg_pmt_ratio
            + max_bill_amt + util + util2 + util3 + util4 + util5 + util6 + avg_util
            + bal_growth_6mo + util_growth_6mo + max_DLQ + min_pmt_ratio + education_by_age
            ,data = cc_train)

# Compare the table to the table above;
nb

# What else do we get?
summary(nb)
names(nb)

nb$levels
nb$laplace
nb$data

# Plot Naive Bayes probabilities;
# Note that this is a degenerate plotting option since there is only one predictor;
plot(nb)

# Predict the class;
predicted.class <- predict(nb);
pct.accuracy <- mean(predicted.class==delay.df$gx_metric_ind);

# check the actual default
table(cc_train$DEFAULT)
table(xdefault3_train)
table(cc_test$DEFAULT)
table(xdefault3_test)
table(cc_validate$DEFAULT)
table(xdefault3_validate)

############################################################################## NAIVE BAYES TRY 2

# naive model
predictor.df <- cc_train[, (names(cc_train) %in% c("PAY_1", "MAX_BILL_AMT","MAX_DLQ","UTIL","AVG_PMT_AMT"
                                  ,"age_bin","education_by_age"))]
nb.2 <- naive_bayes(x=predictor.df,y=cc_train$DEFAULT)
```

```
# Look at output;
summary(nb.2)
plot(nb.2)

# Plot Naive Bayes probabilities;
plot(nb.2, which=c('PAY_1'))

# Open additional graphics window;
X11()
plot(nb.2, which=c("MAX_BILL_AMT"))


# Predict the class;
predicted.class <- predict(nb.2);
mean(predicted.class=="DEFAULT")
xdefault5_train = predict(nb.2, cc_train)
xdefault5_test = predict(nb.2, cc_test)
xdefault5_validate = predict(nb.2, cc_validate)
table(cc_train$DEFAULT)
table(xdefault5_train)
table(cc_test$DEFAULT)
table(xdefault5_test)
table(cc_validate$DEFAULT)
table(xdefault5_validate)
```