Model #101: Credit Card Default Model

Model Development Guide

Table of Contents

1. Introduction

At the start of the 21st century, consumer credit played a driving force behind the economies of most of the leading industrial companies. This drive force has made major impacts on the economy ranging from home ownership to consumer spending. With this growth in spending, an increasing importance is placed on lenders to quantify the risk customers pose when given credit. This importance has led to the continual development of credit scoring, which automatically estimates default risk for consumer credit. The idea behind credit and application scoring is to use the lender's data on past applications to rank the order of risk of defaulting (Thomas, 2009).

This growth on consumer spending leads to the increasing importance for risk prediction. The major purpose of risk prediction is to use various contextual information about a customer and their behavior predict an individual's credit risk in order to reduce the damage and uncertainty (Yeh & Lien, 2009). While predicting whether a customer will default is valuable, it is imperative to find a way how quantify how risky a customer is. Lenders can leverage their existing consumer data by creating predictive models to rank customers on how risky they are based on their probability of defaulting.

This model development guide provides a pipeline of components to generate a predictive solution to rank customers by their risk of defaulting, utilizing customer credit data from a bank in Taiwan. The approach for building this solution contains the following components:

Data Quality Check

When comparing the data with the data quality dictionary, there were a few variables represented inaccurately. These variables contained undocumented levels, and the unknown levels were mapped to distinct levels.

Exploratory Data Analysis and Feature Engineering

During the exploration of the data, it was identified that on average, customers with higher education level and age tend to have a higher given credit and yielded the lowest default rate. Additionally, as customers roll into further buckets of delinquency, the probability of default increases on average irrespective of age and education level.

After further analysis of the data, various features were engineered. The maximum delinquency, average utilization and average payment ratio of customers were identified as the most influential covariates.

Predictive Modeling and Comparison of Results

Using the engineered features, the following models were created: Logistic Regression, Random Forest, Catboost and K-Nearest Neighbors. The Catboost model yielded the best performance based on the chosen evaluation metric, followed by the Random Forest.

Given the regulation and governance around model building in this domain, predictive models must be locally interpretable and be bias free to comply with federal regulations such as the Fair Lending Act. Resulting from this, a Logistic Regression is chosen as the production model due to the inherent interpretability and the ease of deployment.

2. The Data

2.1. Data Description

The data set selected for analysis are customer credit data in Taiwan containing various characteristics about the customer, their payment behavior, and information about whether they defaulted on their payment at a given period of time. As shown in Table 1, there are three divisions in the data set: the training and testing for analysis and model building, and the validation for model validation and monitoring.

Data Set	Number of Observations
Training	15,180
Testing	7,323
Validation	7,497

Table 1: Number of observations in each data set

Table 2 provides a data dictionary including the name and the definition of the covariates given in the data set. The data dictionary is aimed to serve as a ground truth to describe the data. This data dictionary is a vital component of any analysis as it may lead to the discovery of data quality issues when comparing the values seen in the data with the observed data. The variable of interest is DEFAULT. Thorough analysis and modeling are performed in order to identify characteristics that may help describe and to predict whether a customer will default.

Variable	Definition
LIMIT_BAL	Individual and supplementary credit
SEX	Gender of customer
EDUCATION	Education level
MARRIAGE	Marital Status
AGE	Age of customer
PAY_0	Repayment status in September, 2005
PAY_2	Repayment status in August, 2005
PAY_3	Repayment status in July, 2005
PAY_4	Repayment status in June, 2005
PAY_5	Repayment status in May, 2005
PAY_6	Repayment status in April, 2005
BILL_AMT1	Amount of bill statement in September, 2005
BILL_AMT2	Amount of bill statement in August, 2005
BILL_AMT3	Amount of bill statement in July, 2005
BILL_AMT4	Amount of bill statement in June, 2005
BILL_AMT5	Amount of bill statement in May, 2005
BILL_AMT6	Amount of bill statement in April, 2005
PAY_AMT1	Amount paid in September, 2005
PAY_AMT2	Amount paid in August, 2005
PAY_AMT3	Amount paid in July, 2005
PAY_AMT4	Amount paid in June, 2005
PAY_AMT5	Amount paid in May, 2005
PAY_AMT6	Amount paid in April, 2005
DEFAULT	Default payment

Table 2: Data dictionary

2.2. Data Quality Check

A vital first step after defining the data dictionary would be a detailed data quality check. Real data is almost never perfect, and often contains issues requiring attention. It is imperative that this data quality check is performed, as it ensures accuracy and trustworthiness of the insights from analysis and modeling using the data. This data quality check first starts with identification of missing values. Across all sets, there are no missing values for any of the features.

Next, a comparison of the values and patterns observed in the data and the data dictionary is performed. While the data dictionary is intended to serve as a source of ground truth in the data, it is quite common for inconsistencies to exist between the data and the data dictionary. While the source for these inaccuracies is commonly tied to the reported data dictionary, it may also suggest issues with the data itself.

Holistically, the reported characteristics in the data dictionary and the observations are in alignment. Despite this alignment, there were a few variables that had some discrepancies. In Figure 1, a count for the unique values of EDUCATION is performed. According the data dictionary, the expected values are: College, Graduate, High School, and Others. When comparing this with the data, there are three unknown categories (marked with the prefix “Unknown”). This data quality issue was addressed by mapping the unknown levels to Others.

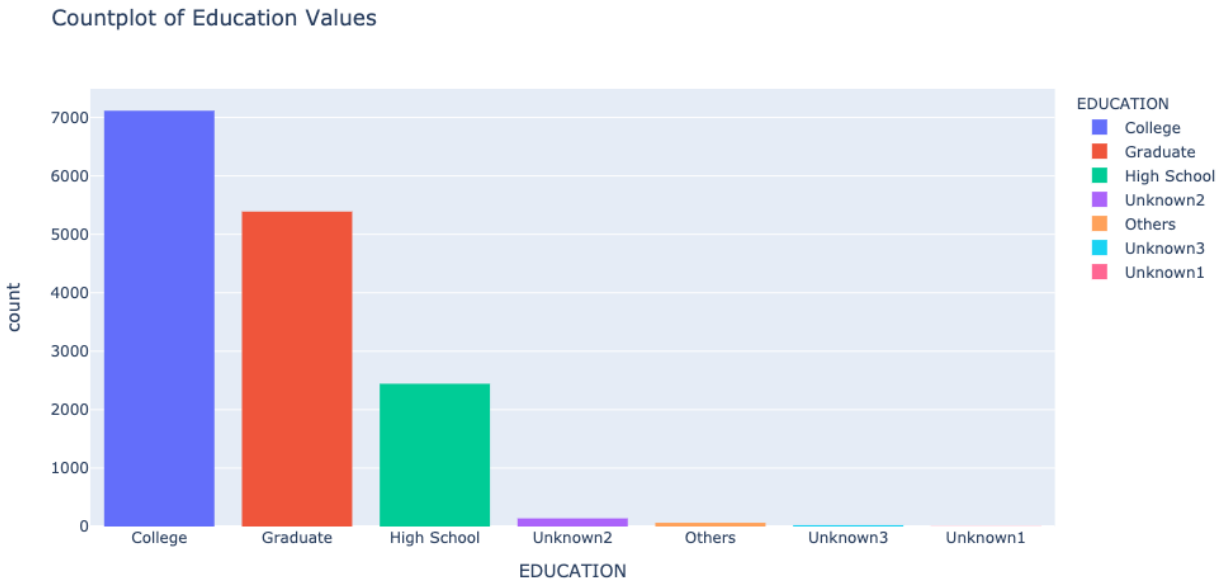

Figure 1: Count of Education Values

The next data quality issue was identified when comparing the MARRIAGE variable. According to the data dictionary, the expected values are: Married, Single, and Others. When comparing this with the data in Figure 2, there is one unknown category (identified as “Unknown”). This data quality issue was addressed by mapping the unknown category to Others.

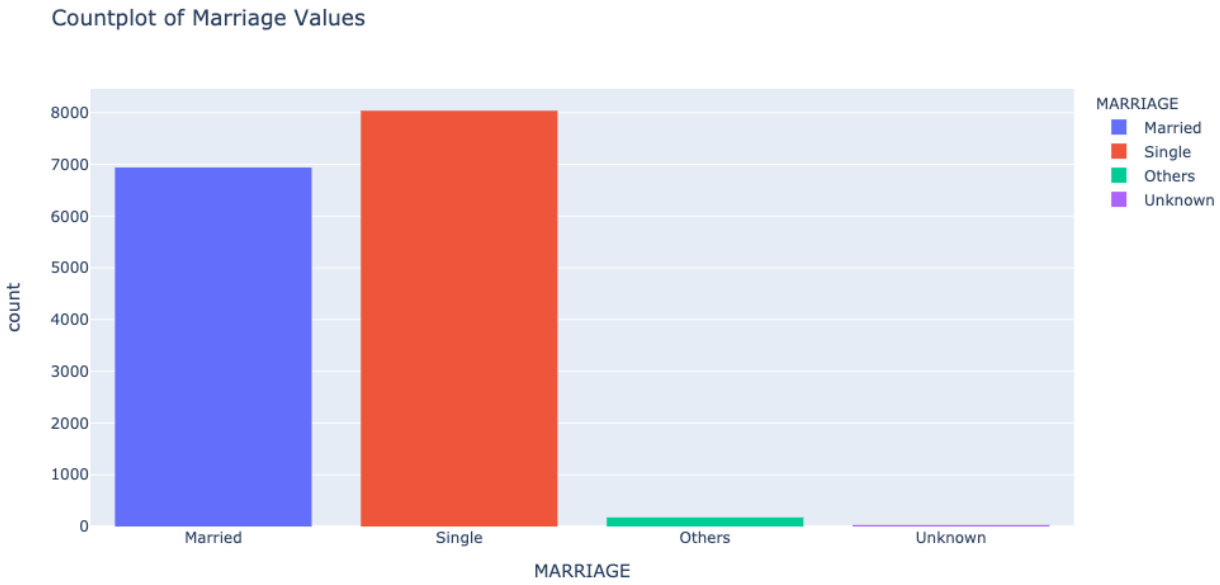

Figure 2: Count of Marriage Values

The final data quality issue was identified when comparing the Repayment status variables. According to the data dictionary, the expected values are: Pay Duly, and the variables indicating delay from 1- 6 months. Figure 3 indicates the observed repayment status in September 2005. The same values observed in Figure 3 is observed in all other repayment status variables.

In Figure 3, there are two unknown categories (identified with the prefix “Unknown”). This data quality issue was addressed by creating a new level Others and mapping the unknown categories to Others.

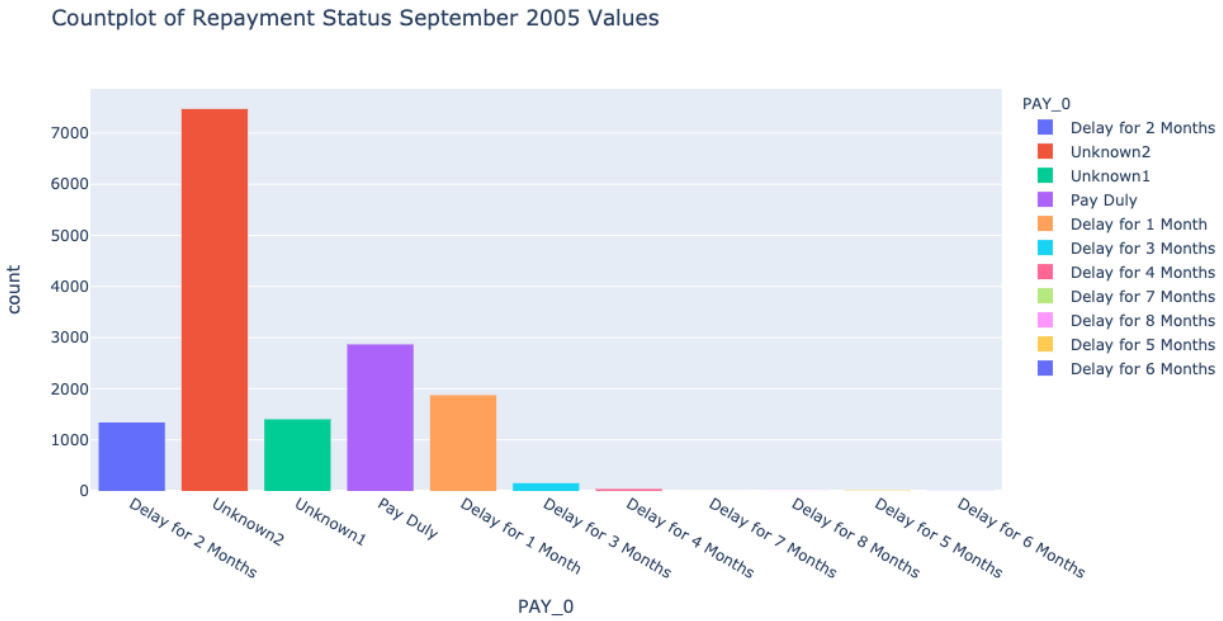

Figure 3: Count of Pay_0 Values

3. Feature Engineering

After conducting a data quality check and addressing the data quality issues, a sense of trust is established with the data. Given this sense of trust, the next process is to perform feature engineering. Feature engineering is the process of taking raw variables (as seen in the data dictionary) and creating descriptive, intelligent features. This process should always be guided with strong domain knowledge and understanding of the data.

Feature engineering and exploratory data analysis (EDA) are one of the most important activities in a machine learning life cycle. A focus on feature engineering and exploratory data analysis will make a large impact to the performance of a predictive model irrespective of the type of model used. Feature engineering, exploratory data analysis and model building are commonly a recurrent pipeline. Given the understanding of the data, some candidate features are engineered. The feaures

displayed in Table 4 below was created using the aforementioned recurrent pipeline. With this understanding, AGE was identified a variable that may play a significant role in whether a customer default or not. For this reason, AGE is first discretized into Age_Bin after performing EDA on this feature.

Following the engineering of Age_Bin, an additive smoothing algorithm (a variation of target encoding as seen in Figure 4) is utilized on Age_Bin to engineer an intelligent feature. Target encoding method of representing a categorical variable numerically. This procedure starts with taking each distinct element in categorical variable x and computing the average of the corresponding values in target variable y. Next, each xi (level in the variable) is replaced according to the mean. Additive smoothing is a modification to this approach to address issues of overfitting using a global mean. Additive smoothing "smooths" the average by including the average over all levels of a covariate:

$$\mu = \frac{n * \bar{x} + m * w}{n + m}$$

Formula 1: Additive Smoothing Formula

In Formula 1 above: μ is mean we are trying to compute, n is the number of values you have, $\bar{x}$ is your estimated mean, m is the assigned weight to the overall mean, and w is the overall mean (Halford, 2018).

Next, weight of evidence (WOE) is performed on the AGE variable. Weight of evidence is a staple technique in the credit scoring domain that helps explain the predictive power of an independent

variable in relation to the dependent variable. In the area of customer default, this can be interpreted as the separation of "good" (customers who have not defaulted) and "bad" (customers who have defaulted). WOE is calculated as:

$$WOE = \ln\left(\frac{Distribution\ of\ Good\ Customers}{Distribution\ of\ Bad\ Customers}\right)$$

Formula 2: Weight of Evidence Formula

In Formula 2 above, Distribution of Good Customers represents the percent of good customers in a particular group and Distribution of Bad Customers represents the percent of bad customers in a particular group. A natural log ln is applied on the quotient in Formula 2 (Bhalla, 2015).

Following the engineering of AGE based features, some distinct features are engineered to represent the largest observations of delinquency (Max_DLQ), largest bill amount (Max_Bill_Amt) and the largest bill payment (Max_Pay_Amt). Next, multiple averages are performed to represent customer behavior over the observed period. This is reflected by the features: Avg_Bill_Amt, Avg_Pmt_Amt, Avg_Pmt_Ratio, Avg_Util, Bal_Growth_6mo, Util_Growth_6mo.

Finally, after performing analysis on some of the prior engineered features, indicator variables are created to identify outliers in the data. Shown in the EDA portion, there were a handful of customers whose activity did not confirm to the average customer. These customers had a combination of an abnormal given credit, utilization, and average payment ratio.

Variable	Definition	Calculation
Age_Bin	Age of customer categorized	Group customer into three bins using AGE: Age_18_25, Age_26_40, and Age_41_100.
Age_WOE	Weight of Evidence of Age_Bin	Use weight of evidence encoding using the following bins AGE: 18-26, 26 to 29, 29 to 36, 36 to 46, and 46 and higher.
Age_Additive_Smoothed	Additive Smoothing target encoding of Age_Bin	Using function in Figure 4, use the following assignments: df = df, by='Age_Bin', on='DEFAULT', m = 20.
Avg_Bill_Amt	Average monthly bill over 6 months	Add all 6 BILL_AMT variables and divide by 6 for every customer.
Avg_Pmt_Amt	Average monthly payment over 6 months	Add all 6 PMT_AMT variables and divide by 6 for every customer.
Avg_Pmt_Ratio	Average payment ratio over 5 months	Average all monthly payment ratios and multiply by 100.
Avg_Util	Average utilization of credit over 6 months	Average all monthly utilization and multiply by 100.
Bal_Growth_6mo	Balance growth over 6 months	Sum all the changes from a preceding bill to the following bill.
Util_Growth_6mo	Utilization growth over 6 months	Sum all the changes from a month's utilization to the following months utilization.
Max_Bill_Amt	Largest bill over 6 months	Find max from all 6 BILL_AMT variables.
Max_Pay_Amt	Largest payment over 6 months	Find max from all 6 PAY_AMT variables.
Max_DLQ	Largest period of delinquency over 6 months	Take the max of the PAY_x variables.
Avg_Util_Anomaly	Indicator variable for presence of anomalies in Avg_Util	Indicator variable for observations that are greater than 3rd Quartile + 1.5*IQR and less than 1st Quartile -

		1.5*IQR for the Avg_Util variable.
Avg_Pmt_Ratio_Anomaly	Indicator variable for presence of anomalies in Avg_Pmt_Ratio	Indicator variable for observations that are greater than 3rd Quartile + 1.5*IQR and less than 1st Quartile - 1.5*IQR for the Avg_Pmt_Ratio variable.
Util_Growth_6mo_Anomaly	Indicator variable for presence of anomalies in Util_Growth_6mo	Indicator variable for observations that are greater than 3rd Quartile + 1.5*IQR and less than 1st Quartile - 1.5*IQR for the Util_Growth_6mo variable.
LIMIT_BAL_Anomaly	Indicator variable for presence of anomalies LIMIT_BAL	Indicator variable for observations that are greater than 3rd Quartile + 1.5*IQR and less than 1st Quartile - 1.5*IQR for the LIMIT_BAL variable.

Table 3: Engineered Features

```
def calc_smooth_mean(df, by, on, m):
    # Compute the global mean
    mean = df[on].mean()

    # Compute the number of values and the mean of each group
    agg = df.groupby(by)[on].agg(['count', 'mean'])
    counts = agg['count']
    means = agg['mean']

    # Compute the "smoothed" means
    smooth = (counts * means + m * mean) / (counts + m)

    # Replace each value by the according smoothed mean
    return df[by].map(smooth)
```

Figure 4: Additive Smoothing Target Encoding

4. Exploratory Data Analysis

4.1. Traditional EDA

An important step prior in a successful predictive model is exploration of the data. Exploratory data analysis (EDA) is a technique to investigate data sets to identify and summaries their main characteristics. While information may be understood viewing the results of statistical tests and summaries, some patterns may be difficult to identify and diagnose without viewing the main characteristics of different covariates.

Using the engineered features, the first check is to view the balance of classes of the target variable DEFAULT. Figure 5 below shows the imbalance of classes for this modeling exercise. It appears that 77.5% of the customers have not defaulted and 22.5% of the customers have defaulted. This behavior is expected given the nature of the problem and this may serve to be problematic given this imbalance. This can be addressed by: undersampling the majority class (those who have not defaulted), oversampling the minority class (adding multiple copies of customers who have defaulted records to the dataset), or use synthetic minority oversampling (creating synthetic examples based on customers who have defaulted to the dataset). Certain predictive models are very sensitive to class imbalance and other models are more resilient to this issue.

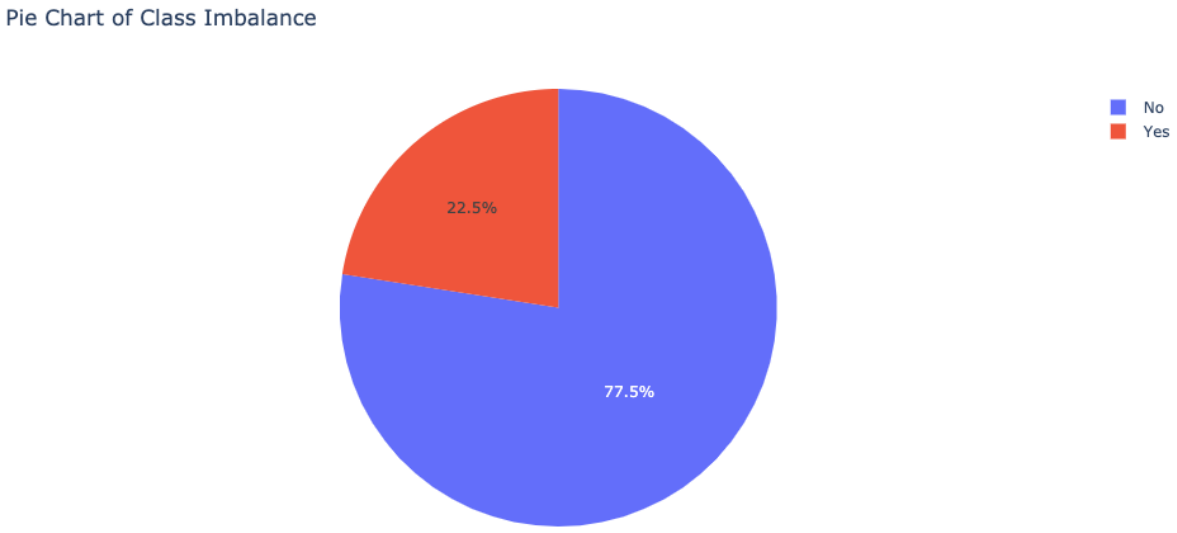

Figure 5: Pie chart of DEFAULT class imbalance

As mentioned during the feature engineering portion of the analysis, the AGE covariate was determined to be a variable of interest for which many other covariates were engineered. The first was a binned representation of the AGE variable. Figure 6 below shows the percent of customers that fall into each category along with those customers who defaulted. Table below shows the percentage of customer default based on their respective age bin. Holistically, we see that the vast majority of the customers were in the 26 to 40 age bin. With that being said, the proportion customers who defaulted is still the highest in the 18 to 25 bin. 26.88% of the customers in the 18 to 25 age bin defaulted, 20.84% in the 26 to 40 age bin defaulted, and 24.22% of the customers in the 41 to 100 age bin defaulted.

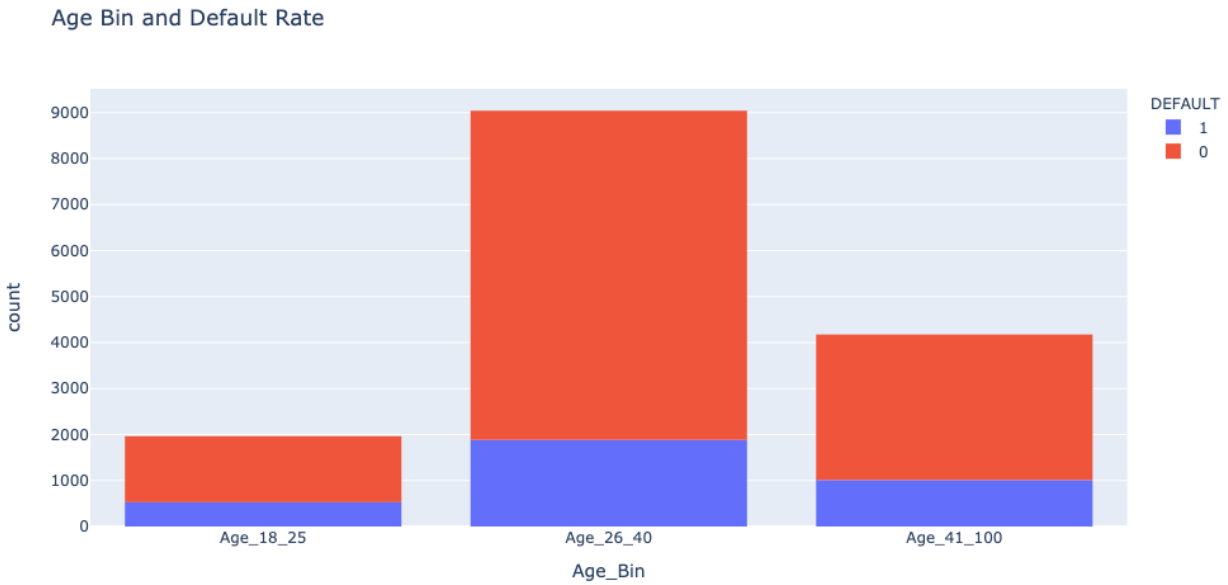

Figure 6: Age_Bin default breakdown

Age Bin	Percent of Customer Default
18 to 25	26.88 %
26 to 40	20.84 %
41 to 100	24.22 %

Table 4: Breakdown of Customer Age Bin and Percentage of Default

A common hypothesis is that those with a higher education tend to have a higher income, credit score and available credit. Based on the analysis of given credit and education in Figure 7, higher educated individuals do seem to have a higher given credit on average. Across the lower credit limits, we see high school to be very prominent with a higher number of customers. As the credit limit increases, the number of customers with High School education starts to decrease and we that the higher limits consist of predominately College and Graduate. In all brackets, we see that there

are still a few customers with Others as the education level. An important observation is that Others may consist of those who do not have a High School education at the minimum but may also include those who have a Post Graduate level of education.

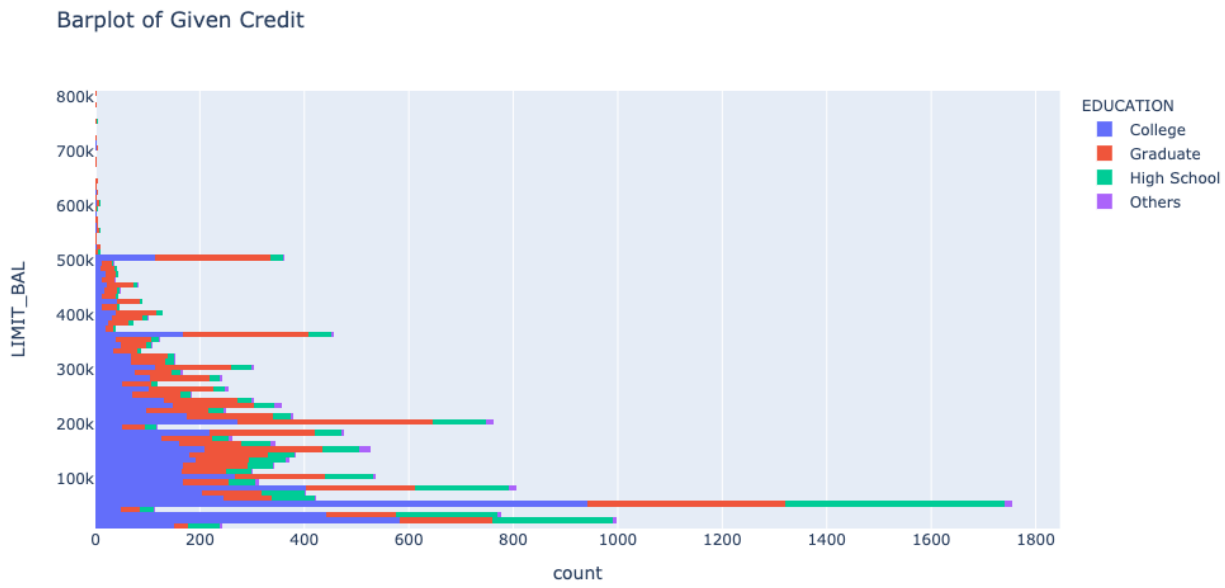

Figure 7: Given credit based on education level

Given that an individual's credit score increases over time (given a track record of good behavior), another popular hypothesis is that those who are older may have higher given credit. The findings on Figure 8 below shows that on average, customers that are younger do tend to have lower given credit. It is evident that the majority of customers between 18 to 25 have at most $200,000 worth of given credit. With that being said, there does appear to be point where this hypothesis may not always hold true. It is evident that even among the highest given credit, there are still customers in the 26 to 40 age bin and some of those in the 41 to 100 age bin have very low given credit.

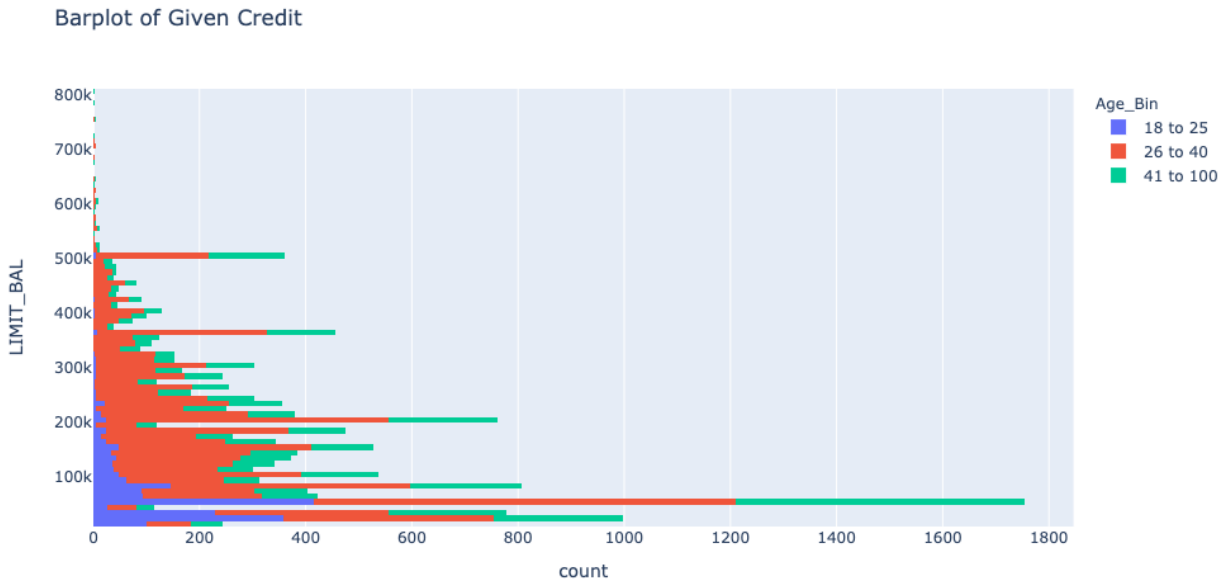

Figure 8: Given credit based on age bin

Although Figure 8 shows that on average customers who are younger have lower given credit, we still see that there are some customers in the 18 to 25 age bin that are still given a significant amount of credit, indicating anomalies in the data. Figure 9 tries to identify those customers in each age bin that are the most different from other customers in that age bin using box plots anomalies. We see that for every age bin, there are customers who have significantly higher given credit in comparison to other customers in the same bin. The customers who have this sort of unusual given credit could potentially be point anomalies. These point anomalies can often be categorized as influential points.

Influential points are records in a data set that greatly affect the way in which your model learns and makes predictions. In the case of generalized linear models such as linear regression or logistic regression, these influential points may significantly alter the estimated coefficients for a covariate,

which may lead to inaccurate learning and predictions. Resulting from this, we must be very careful in addressing these anomalies in order to prevent adding bias to the models learning and must use domain knowledge to guide this process. Given the understanding of the dataset, indicator variables are created to identify those customers who have values that are significantly different from other values. In Figure 10 below, indicator variables were created for given credit, average utilization and 6 month utilization growth.

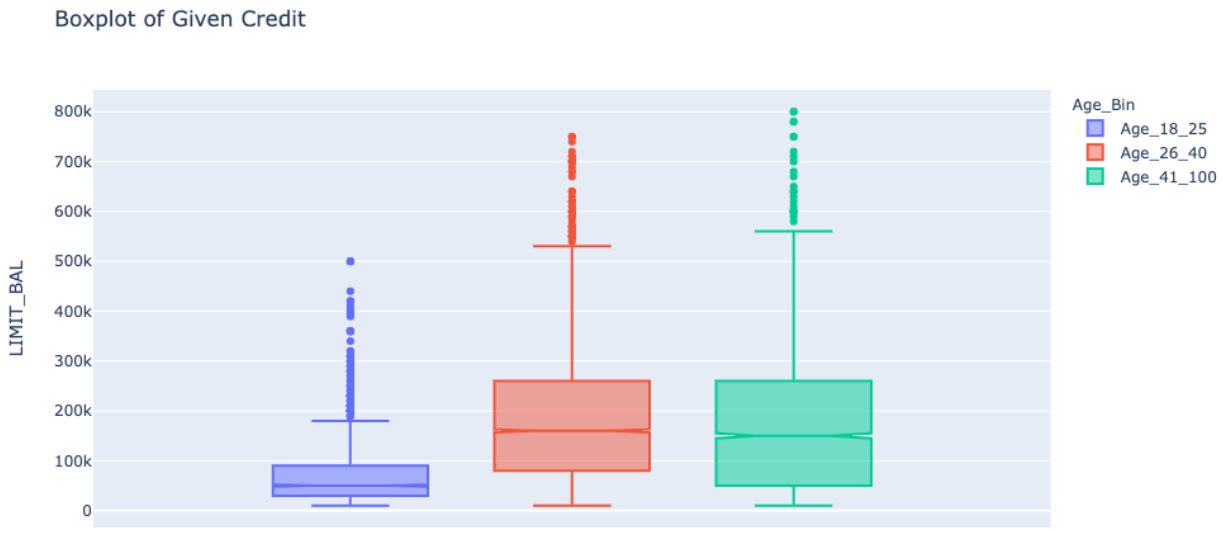

Figure 9: Point anomalies for given credit

Figure 10: Point anomalies using Box Plot Rules

Next, we analyze the maximum months delinquent and default percentages for customers in Figure 11. On average, we see that as customers roll into further buckets of delinquency, the percentage of those customers defaulting increases. Often lenders may have a particular time period or bucket where they write these delinquencies off as losses. With this in mind, customers who have gone

longer without making payments may be considered more likely to default on their payment and pose a higher risk.

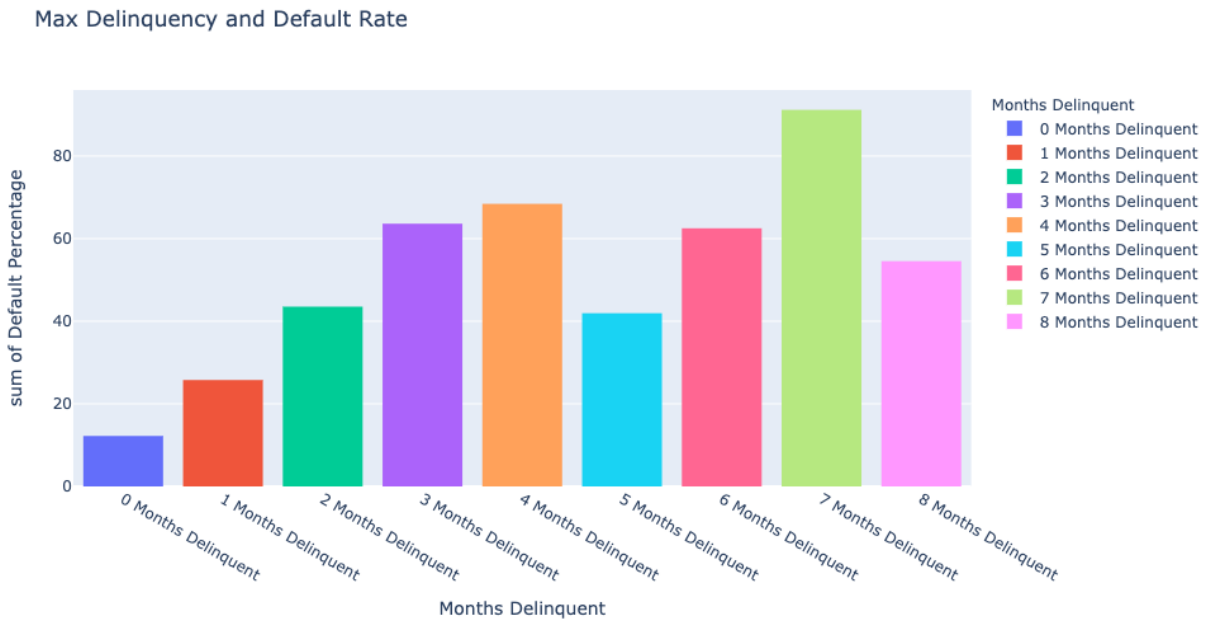

Figure 11: Maximum delinquency default breakdown

4.2. Model Based EDA

In the pipeline, another important type of EDA would be model based EDA. Model based EDA is our understanding of how the model is learning from examples given the covariates that we engineer. This can often help us spot unintended learnings and bias introduced and help us understand how features we engineered is impacting learning and predictions.

In Figure 12 below, a Decision Tree Classifier with a maximum depth of 10 is fit using the engineered features to predict whether a customer will default. All of the nodes highlighted in green are selected to split as they yield the lowest value of the Gini impurity. Building rules based off the splits of a decision tree can be very helpful in quantifying how much risk a customer poses

in defaulting. At the top, we see that the first feature used to split would be the maximum delinquency. It is splitting based on whether the customer has been at most 0 to 1 month delinquent on their payments historically. This validates the previous generalization that it can be common for good customers to roll into the first bucket of delinquency (usually 1 month) and make payments before rolling into the second bucket of delinquency. The next split would be with the maximum payment amount. It seems that there may be a clear division between customers who are within 1 month of delinquency and have paid greater than or equal to $1227.

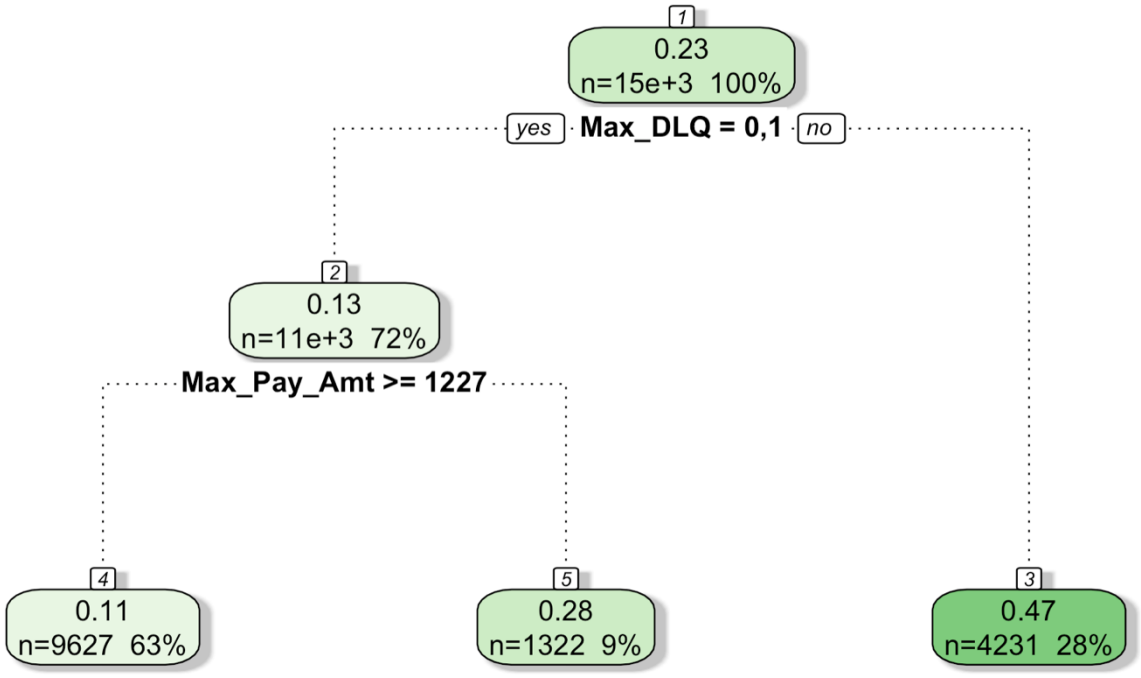

Figure 12: Decision Tree Plot

Next, a Random Forest Classifier is fit using 100 decision trees. Figure 13 shows how important each feature is in reducing the Gini impurity across all 100 trees on average. As seen in the individual decision tree, the maximum delinquency is also the most important feature for the

Random Forest Classifier as well on average. The next two important variables would be the average utilization and the average payment amount as they are quite close in value to the maximum delinquency variable on the plot.

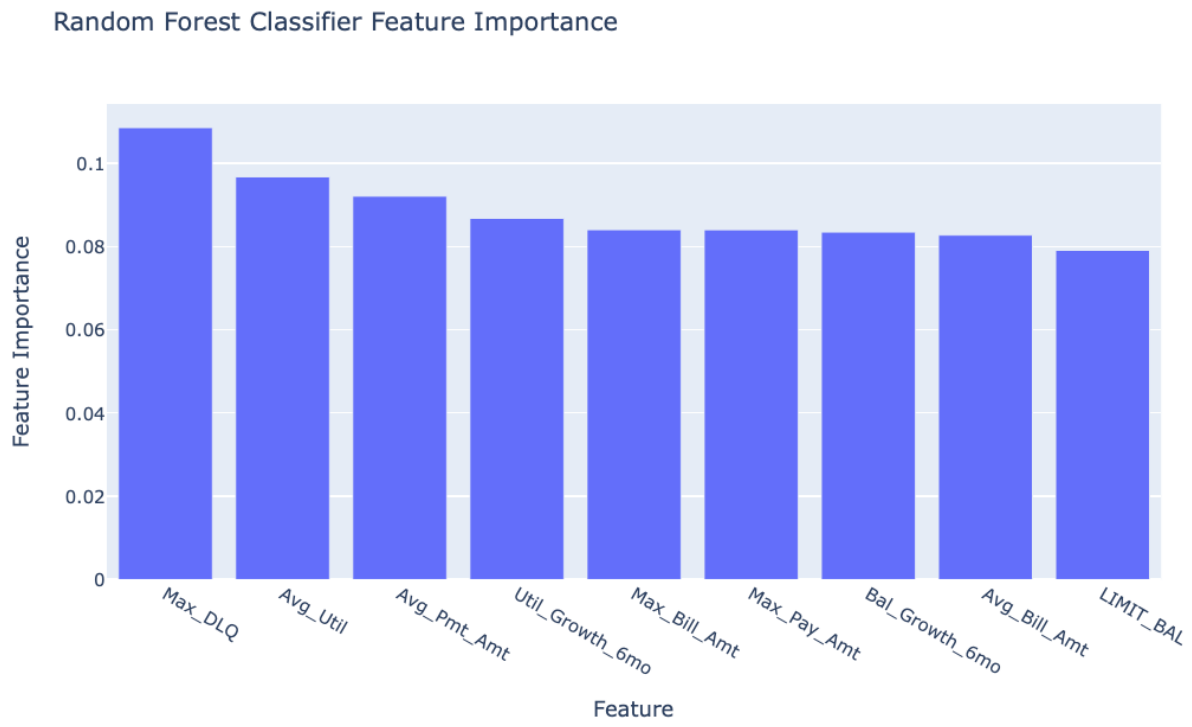

Figure 13: Random Forest Top 9 Feature Importance

An alternative way to calculate how influential a feature is in predicting whether a customer will default would be to analyze the contributions variables have in predictions. Shapely additive explanations are a game theoretic approach for calculating the contribution that a particular feature has in making predictions on average and provides local and global model explainability. The difference between this approach and other common feature importance or permutation importance is that this does not calculate the reduction in impurity or the increase in information gain, rather it quantifies how much the probability of a customer defaulting changes given a feature.

Figure 14 below confirms previous analysis that maximum delinquency has the largest contribution in Random Forest Classifier for predicting whether a customer will default or not. As seen in Figure 12 and 13, the average payment amount and the average utilization have among the highest contribution to these predictions along with the given credit for a customer. What is also evident is that few of the outlier indicator variables engineered seem to have no contribution to making predictions on average. Evaluation of the shapely additive explanations for a Catboost classifier yields identical rankings of the covariates with different contributions.

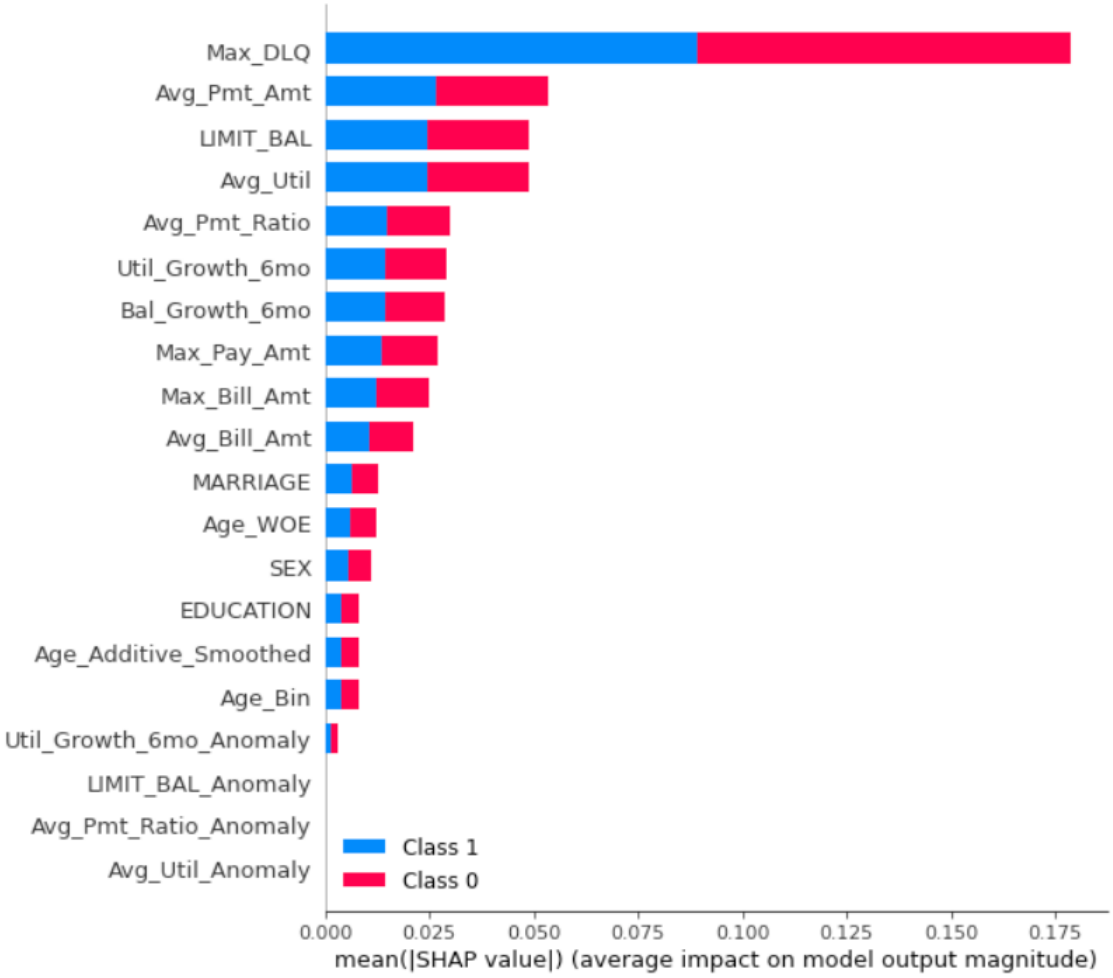

Figure 14: Random Forest Global Shapely Additive Explanations

5. Predictive Modeling: Methods and Results

5.1. Random Forest

Random forests are a collection of decorrelated decision trees that are generated from bootstrapped aggregated samples. The selection of random predictors for splitting for each of the trees in a forest addresses the variance and helps generate a predictive model that generalizes better to new examples in comparison to Decision Trees. The disadvantage for this method would be the loss of interpretability in comparison to other simpler models.

After getting a better understanding from both traditional and model based EDA, some basic feature selection is performed, and a Random Forest is created to predict customer default. Evaluation of the feature importance after feature selection, we see the same features are identified as the most important in comparison to Figure 13 (in different orders) with the exception of LIMIT_BAL. After feature selection, the model identifies Avg_Pmt_Ratio with a higher feature importance score than LIMIT_BAL.

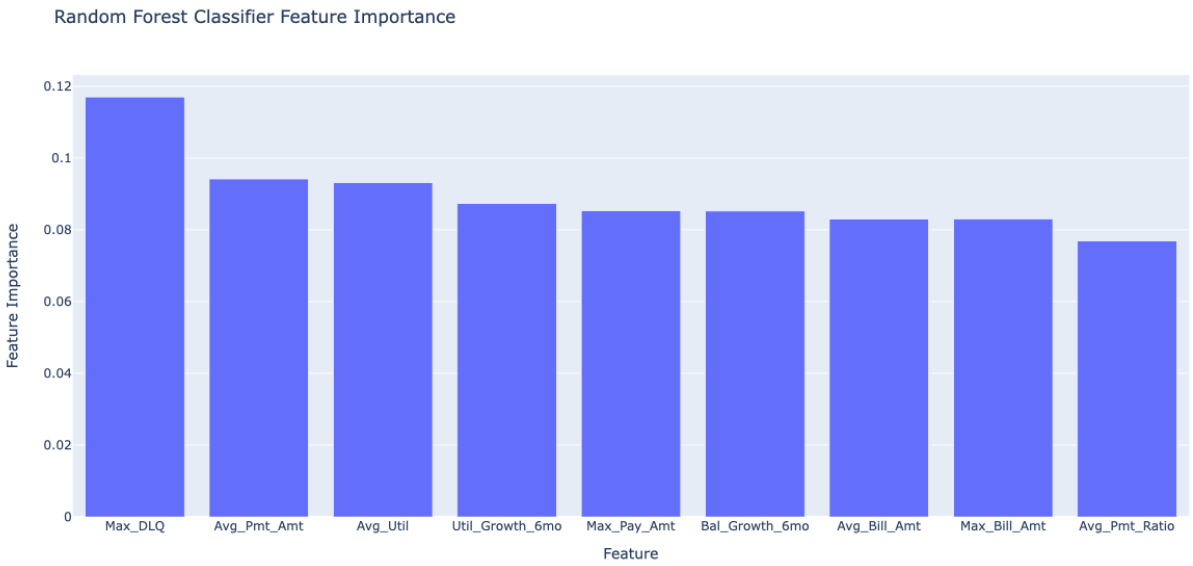

Figure 15: Random Forest Top 9 Feature Importance Feature Selection

Data Set	Accuracy	True Positive Rate	False Positive Rate
Training	0.735573	0.596845	0.224037
Testing	0.731667	0.596660	0.231877

Table 5: Random Forest Training Performance

5.2. Gradient Boosting

In comparison to bootstrap aggregation, boosted models grows trees sequentially. Each tree is grown using information from previously grown trees and the construction of each tree depends strongly on the trees that were previously grown. Rather than bootstrap sampling, each tree is fit on a modified version of the original data set. For every decision tree in the forest, a new decision tree is created using the residuals from the preceding model. This serves as method to "update" the residuals of the forest (Hastie et. Al., 2009).

Following the EDA, a Catboost classifier is created by removing the engineered features that had no contribution to the predictions and those that did not reduce the impurity. Catboost is a gradient boosting model that handles categorical variables by a combination of permutation, quantization and encoding.

When creating the Catboost model and checking for the contributions to predictions via the shapely additive explanation values, we find that the similar values are shown as having the most contribution but the ranking in contribution is differs. In Figure 16 below, we also find that the

maximum delinquency has the largest contribution in Catboost model for predicting whether a customer will default or not on average, followed by the average payment amount.

What is different after feature selection is that the average utilization has a larger contribution for predicting default over the customers given credit. Finally, it seems that the balance growth in 6 months does not have as significant of a contribution in comparison to the model that was fit without feature selection. After feature selection the maximum payment amount, average payment ratio, maximum bill amount, and average bill amount have a larger contribution for predicting customer default over the balance growth over 6 months.

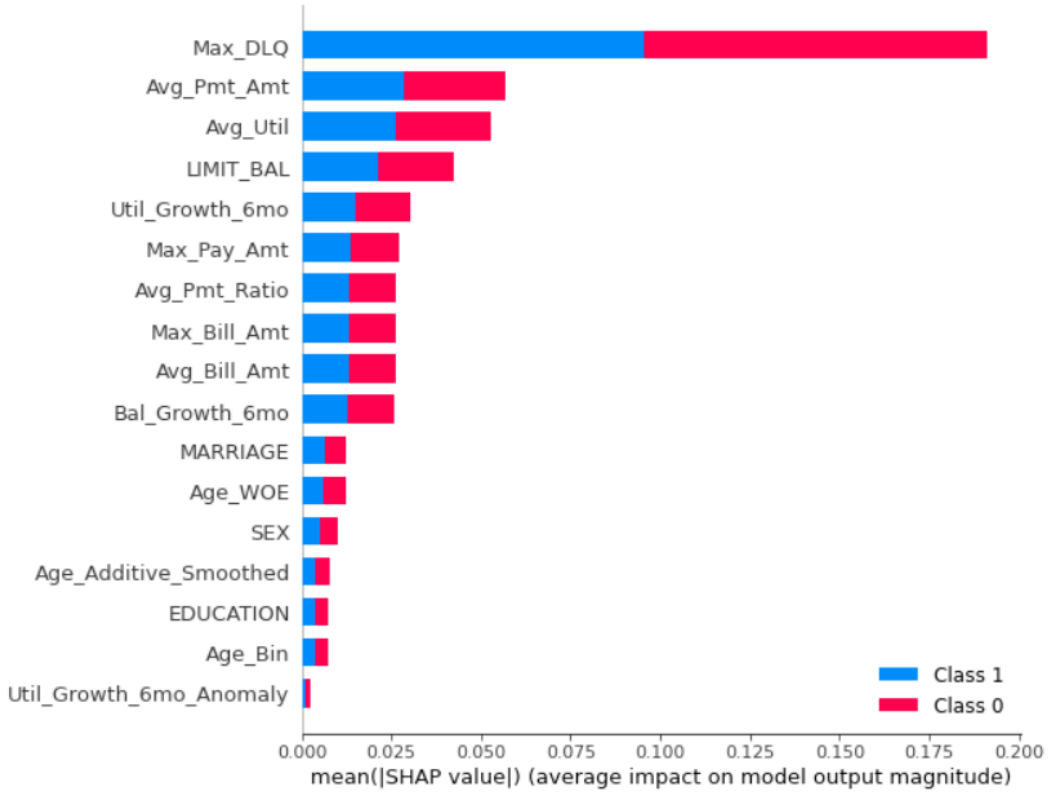

Figure 16: Catboost Global Shapely Additive Explanations Feature Selection

Data Set	Accuracy	True Positive Rate	False Positive Rate
Training	0.755599	0.582238	0.193927
Testing	0.761027	0.592807	0.193548

Table 6: Catboost Training Performance

5.3. Logistic Regression with Variable Selection

A logistic regression model is a Generalized Linear Model (GLM) that models the probability that a response variable belongs to a particular category. A logistic regression model gives outputs between 0 and 1 by using a logistic function and estimates the coefficients using maximum likelihood estimation. Coefficients in this model are represented as the log odds, and a positive coefficient is associated with an increase in the probability of X. Being a GLM, logistic regression models have the following assumptions that must be tested: data is independently and identically distributed (iid), no multicollinearity and a linear relationship between the quantitative covariates and the logit of the outcome. These assumptions must be validated to generate trustworthy predictions. Following the insights from the Random Forest and Gradient Boosting models, the following features were selected as the initial pool of candidate predictor variables as they yielded the highest contribution to predictions/reduction of impurity:

Initial Pool of Features
LIMIT_BAL
SEX
EDUCATION
MARRIAGE
Age_Bin

Avg_Bill_Amt
Avg_Pmt_Amt
Avg_Pmt_Ratio
Avg_Util
Bal_Growth_6mo
Util_Growth_6mo
Max_Bill_Amt
Max_Pay_Amt
Max_DLQ
Age_WOE
Age_Additive_Smoothed

Table 7: Logistic Regression Initial Pool of Features

First, we can confirm that the data is iid given there is no time order or dependence between records. When testing for multicollinearity, we find that Avg_Bill_Amt, Avg_Pmt_Amt, Max_Bill_Amt and Max_Pay_Amt were suffering from multicollinearity yielding VIF values of greater than 10. To satisfy this assumption, principal component analysis is performed, and the resulting variance explained is displayed in Figure 17. Resulting from this, three principal components are created to represent those features with high multicollinearity.

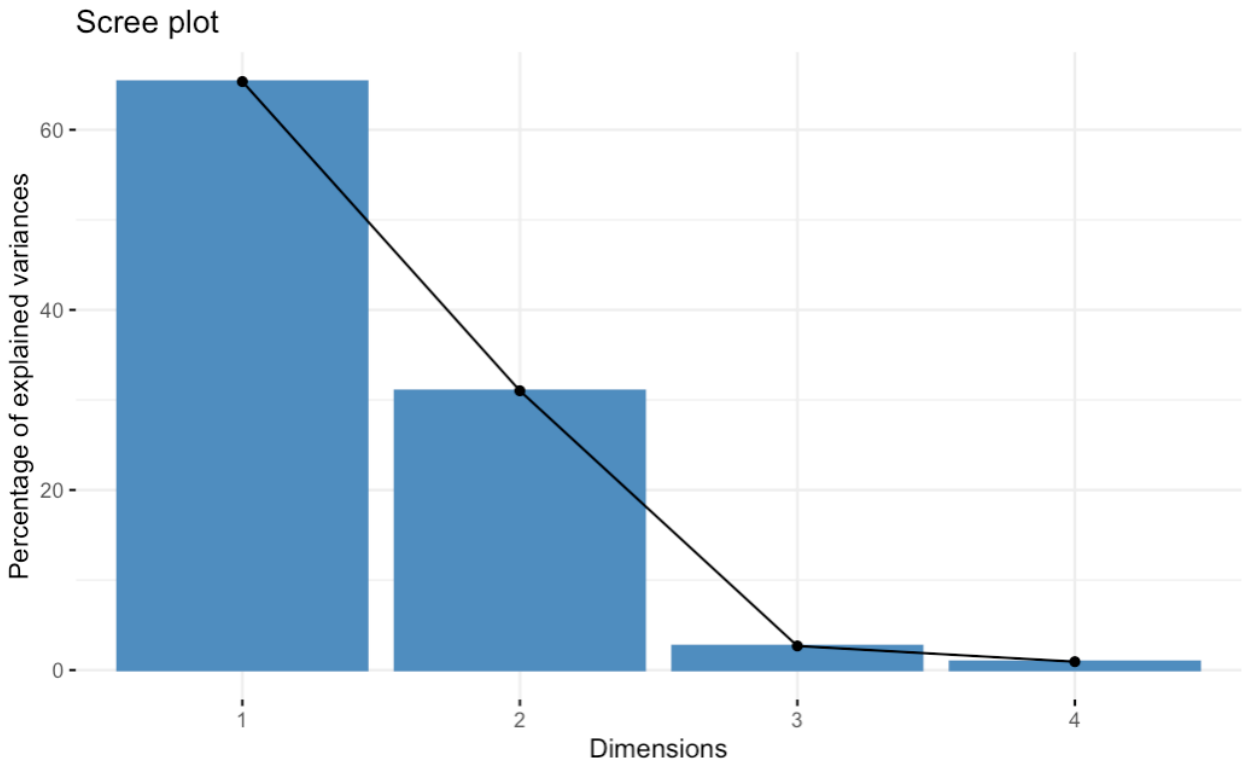

Figure 17: Scree Plot

Following the principal component analysis is the validation of the assumption for a linear relationship between the quantitative covariates and the logit of the outcome. After analyzing the relationships between all of the quantitative covariates and the logit of the outcome, all of plots appear to validate the assumption of linearity and no transformations appear to be required.

Next, we identify and analyze influential points in the model. Influential points are points that greatly affect the estimation of the coefficients of a GLM. It is imperative the identify and address how to treat these points. In Figure 18 below, cook's distance is utilized to find influential points and the displayed row numbers are the most influential points.

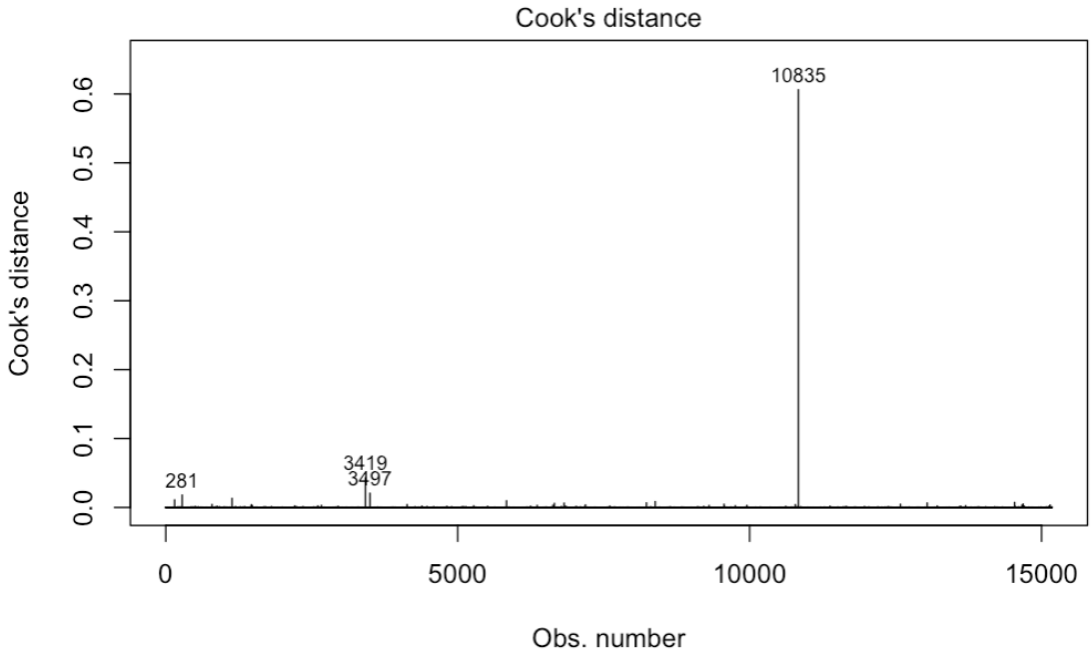

Figure 18: Cook's Distance for High Leverage Points

Finally, features selection is performed. Using this filtered list of features in Table 4, a model is estimated after performing Backward Stepwise AIC feature selection. This method starts with a fully saturated model and starts recursively eliminating features from the model that yields a decrease in the AIC value. The optimal model from this process will contain the optimal features for this model, and the coefficients are shown in Figure 19.

Logistic Regression Production Model

	Dependent variable:
	DEFAULT
Age_WOE	-10.221
	p = 0.111
EDUCATION	-0.053*
	p = 0.078
SEX	-0.140***
	p = 0.002
MARRIAGE	-0.211***
	p = 0.00000
PC1	-0.119***
	p = 0.00000
PC3	0.744***
	p = 0.00000
LIMIT_BAL	-0.00000***
	p = 0.000
PC2	0.268***
	p = 0.000
Max_DLQ	0.711***
	p = 0.000
Constant	-0.957***
	p = 0.000
Observations	15,180
Log Likelihood	-7,044.639
Akaike Inf. Crit.	14,109.280
Note:	*p**p***p<0.01

Figure 19: Logistic Regression Model Coefficients

Data Set	Accuracy	True Positive Rate	False Positive Rate
Training	0.579578	0.609115	0.429021
Testing	0.569166	0.626204	0.446237

Table 8: Logistic Regression Training Performance

5.4. K-Nearest Neighbors

The K-Nearest Neighbors (KNN) model is a non-parametric model that estimates the estimates the conditional distribution of the target Y given covariates X, and then classifies a given observation to the class with the highest estimated probability. The KNN model closely resembles an optimal Bayes classifier. When using this model for prediction for a customer z, the k closest points are identified. Given the true classes Y for the k points, the conditional probability of defaulting and not defaulting is calculated and z is classified to the class with the highest conditional probability (Hastie et. Al., 2009).

The KNN algorithm works as an optimization problem that clusters the points minimize the within cluster sums of squares (WCSS). This WCSS is measured using a distance metric. While the distance metric is configurable, this model was fit with a Euclidean distance, which it not compatible with ordinal or nominal variables. For that reason, only the quantitative features were selected for this model.

After selecting the features for this model, a key input into the KNN model to be identified is the number of neighbors k. This is a fundamental component to the model and an optimal k neighbors

must be selected. Figure 20 below, multiple models were fit to identify the optimal number of k neighbors and it appears that 8 neighbors is optimal as the misclassification error seems to increase after this.

Figure 20: Optimal Number of Neighbors

Data Set	Accuracy	True Positive Rate	False Positive Rate
Training	0.538801	0.620800	0.485073
Testing	0.533251	0.621708	0.490635

Table 9: K-Nearest Neighbors Performance

6. Comparison of Results

After thresholding the probabilities and selecting the optimal cut off points of probabilities, the Catboost model is yields the best performance based on the testing set accuracy and the false positive rate. While the Random Forest was close in terms of performance and yields a larger testing true positive rate, the model is more prone to false positives in comparison to the Catboost model. The Logistic Regression yields the largest testing true positive rate but also yields the largest testing false positive rate. The K-Nearest Neighbors algorithm yields comparable results to the Logistic Regression model but has a lower testing true positive rate and a higher testing false positive rate. The Logistic Regression was chosen as the final model for production given its interpretability and the largest testing true positive rate.

Additionally, representing the confusion matrix elements as profits and losses will greatly help quantify the best performing model. Determining the cumulative profit every model yields not only helps select the best performing model, but also helps quantify the value this predictive model adds through ties to a P&L statement.

Model	**Accuracy**	**True Positive Rate**	**False Positive Rate**
Random Forest	0.735573	0.596845	0.224037
Catboost	0.755599	0.582238	0.193927
Logistic Regression	0.579578	0.609115	0.429021
K-Nearest Neighbors	0.538801	0.620800	0.485073

Table 10: Training Set Performance

Model	Accuracy	True Positive Rate	False Positive Rate
Random Forest	0.731667	0.596660	0.231877
Catboost	0.761027	0.592807	0.193548
Logistic Regression	0.569166	0.626204	0.446237
K-Nearest Neighbors	0.533251	0.621708	0.490635

Table 11: Testing Set Performance

7. Conclusions

In conclusion, with the growth in data and online e-commerce, there is an increasing need for risk prediction. The major purpose of risk prediction is to use various contextual information about a customer and their behavior predict an individual's credit risk in order to reduce the damage and uncertainty (Yeh & Lien, 2009). While predicting whether a customer will default is valuable, it is imperative to find a way how quantify how risky a customer is. Lenders can leverage their existing consumer data by creating predictive models to rank customers on how risky they are based on their probability of defaulting. The major components of this solution are data quality checks, exploratory data analysis, feature engineering, predictive modeling and comparison of results.

When performing data quality checks, we see that there were a few variables represented inaccurately. These variables contained undocumented levels, and the unknown levels were mapped to distinct levels. During the exploration of the data, it was identified that on average, customers with higher education level and age tend to have a higher given credit and yielded the lowest default rate. Additionally, as customers roll into further buckets of delinquency, the probability of default increases on average irrespective of age and education level. The maximum

delinquency, average utilization and average payment ratio were identified as the most influential covariates in the model in both the feature importance and shapely additive explanations plots.

During the predictive modeling, 4 models were used: Logistic Regression, Random Forest, Catboost, and K-Nearest Neighbors. All the parametric models were created using the optimal probability thresholds based on the chosen metric for evaluation. While the Catboost model yielded the best performance on the metric for evaluation, the Logistic Regression was selected for the final production model.

The overall quality of the results is trustworthy. Using the solution implemented in the model development guide, lenders can now rank their customers based on the probability of defaulting. Given that the final model is a probabilistic model, lenders have the ability to rank customers depending on their preferred tolerance or risk by setting cutoff points for classification using the probability. The areas for improvement would include a heavier emphasis on feature engineering stemming from the utilization of more informative covariates describing the customers behavior.

8. Bibliography

Bhalla, D. (2015) Weight of Evidence (WOE) and Information Values (IV) Explained

Halford, M. (2018) Target encoding done the right way

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York: Springer.

Thomas, L. C. (2009). Consumer credit models: Pricing, profit, and portfolios. Oxford: Oxford University Press.

Yeh & Lien (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients.