Applying Advanced Modeling Techniques to Credit Risk Modeling

Alexander K. Bechler

Abstract

Statistical modeling is a technical practice that has quickly garnered attention as an effective way to manage risk at large financial institutions. With the advent of the financial crisis in 2008, there has been increased interest in more effective ways to leverage data to plan for expected credit losses and strengthen loss forecasts. Stronger loss forecasts increase a firm's ability to plan a correct distribution and disbursement of its capital. Regulatory scrutiny and the need for easily interpretable models has required banks to utilize more basic modeling techniques such as linear and logistic regression. With the rapid development of big data environments at many large companies, more advanced modeling techniques have quickly grown in both reach and use. This study will leverage a dataset containing 50,000 customers in a mortgage portfolio to compare the baseline analytical modeling frameworks with the advanced modeling approaches. The study will analyze the most effective combination of both advanced and basic modeling approaches in order to produce the most robust credit risk forecast on the provided dataset. The results will include the performance of the baseline and advanced modeling approaches, provide a recommendation on the models that perform the best on this application, and briefly comment on suggested extensions.

Introduction

Background Information:

Since the 2008 great financial crisis, credit risk modeling has become of increased interest to large financial institutions. The liquidity crisis that evolved throughout the years of 2006-2008 led to extremely large losses incurred by many of the nation's largest banks. Ineffective planning, obscene amounts of risk taking, and general moral hazard created one of the largest asset bubbles and subsequent crashes in the history of the world [1]. This crash led to the adoption of generally accepted credit risk modeling practices across the industry. The Federal Reserve system designed, oversees, and regulates these new guidelines and practices across the industry. Credit risk modeling is the process and practice that large financial institutions use to project the losses that they will incur on their financial instruments. These financial instruments are not only limited to stocks and bonds. Other financial instruments that present a potential for losses include credit cards, home loans, auto loans, and any other instrument that has a credit risk associated with it.

Institutions run their forecasts on a quarterly basis and are mandated to release these results to the public. These credit loss forecasts inform how much capital (money) that the bank needs to keep on its balance sheet to cover these expected losses. The affected capital *cannot be used for investment purposes, dividend purposes, or lending purposes.* It simply exists as a line item on the institution's balance sheet to cover any credit losses that occur over the forecast period, which is typically a forward looking forecast covering the following 2-years. Depending on the size of the institution, this capital could become billions of dollars. In 2023, Bank of America had about $20 billion held in reserves for credit losses [2].

Credit risk modeling is tightly regulated and controlled by the US Federal Reserve Bank and System. Comprehensive Capital Analysis and Review ('CCAR') and Current Expected Credit Loss ('CECL') are two of the many regulatory frameworks [3]. Risk model development takes place in three stages, as mandated by the federal reserve: model development, model validation, and model governance. Model development is the actual data mining and statistical piece of the credit risk modeling process. Development encompasses both the data preprocessing and cleaning part of model development as well as the development of appropriate statistical models. Model validation is the formal process where a team

that did not build the model examines the development team's process, decision, and final model and challenges it. Validation will build challenger models, test the model for weaknesses, and evaluate the practices the modeling team engaged in when they built it [4]. Model governance is the group that monitors the model once validation approves it. Governance ensures that the model continues to perform as expected on production-line data [4]. If the model's performance begins to deteriorate, governance will engage with the model development team to re-tune the model or to develop an entirely new model.

Business Problem and Analysis Motivations:

In crafting its regulations around credit risk modeling for loss forecasting, the Federal Reserve System has required banks to have *interpretable* and *explainable* results and models [5]. When a bank submits its loss forecast submission for CCAR and CECL, it must include both its output and the models used to generate those outputs. The Federal Reserve must be able to look into the modeling process and understand the relationships between predictors, why certain predictors were chosen, and other analytical questions [3]. This requirement has strictly limited banks to using modeling techniques that are easily interpretable, such as logistic regression and linear regression.

Since these regulations have been written, many more advanced statistical modeling techniques have come to the forefront of the data science community. These techniques include models such as random forests and xGBoosting. The requirements implemented by the Federal Reserve has made it so that many of these new modeling techniques cannot be used by model developers. These are ensemble modeling techniques, which generally include an average of many different models that form a final model. It can be difficult, or almost impossible, to provide readily interpretable results for these forms of models - violating the Federal Reserve's rule of interpretability and explainability [3,5]. Only recently has the data science community in financial services begun to explore utilizing these modeling techniques on real-world data to understand how they perform against legacy modeling frameworks.

This analysis will seek to explore if these advanced modeling/ensemble modeling techniques perform better on real-world data compared to the baseline regression models. If the accuracy of the advanced modeling techniques consistently prove to be more accurate and reliable than baseline modeling techniques, a gradual shift in analytical practice can change the dialogue in the industry. More accurate loss forecasting not only gives leaders a better picture of their portfolio performance, but it also provides an advantage when it comes to capital planning [5]. The better a firm's loss forecast, the more confident leadership can be that capital is correctly being allocated. This will reduce the risk of an overestimate of losses, which would involve a bank holding more in reserves than it needed to and therefore lowering its profitability by taking capital that could have been allocated elsewhere. It also reduces the risk of an underestimate of losses, which would require a bank to take an unexpected hit to its capital and equity position. This could cause significant implications for not only the firm's stock price, but also its reputation in the competitive marketplace as well. Should the more advanced modeling techniques be consistently superior to the baseline techniques, over time there will be consistent pressure not only from bank shareholders but also bank leadership and the analytical community as a whole to move towards implementing them.

Excess capital for purposes of this analysis will be defined as the amount in excess of actual losses that a model suggests a financial institution needs to hold on its balance sheet. For example, if the loss forecasting process suggests that a firm needs to hold $100M in capital on the balance sheet against credit losses, but the bank only experiences $90M in credit losses, then the amount of excess capital on the balance sheet is $10M. Excess capital is significant for a bank because money that is held on the

balance sheet cannot be allocated to other uses. This represents a significant opportunity cost for the bank. That $10M could be invested in risk-free assets, earning a 5% rate of return. It could also be allocated for lending purposes, earning between 5%-30% depending on the asset chosen[1]. Excess capital can also be negative, indicating that a financial institution does not have adequate capital on the balance sheet to cover against its losses. This would require either a loan from another financial institution, raising capital from investors, or intervention from the federal government. Of the two possibilities, negative excess capital represents a far worse reality than positive excess capital. This analysis will be focused on discording the combination of models that *minimize the amount of excess capital* while also providing accurate results.

Exploratory Data Analysis

The dataset used to test the business question associated with this analysis is a dataset containing a sample of mortgages from the *Credit Risk Analytics* textbook [6]. The dataset contains 50,000 observations of mortgages originated over the ten year period between 2004 and 2014. Variables included in the dataset are both *customer* level attributes and *economic* attributes. Of the 50,000 observations, there are about 13,000 defaults, equating to roughly a 26% default rate.

Customer level attributes could be features like the customer's credit score, the amount of the mortgage, and if the customer is an investor or not. These features will vary from customer to customer, as no two customers typically have exactly the same credit profile. Economic attributes are attributes that impact the entire macroeconomic environment. The economic attributes provide value here because certain customers with different profiles exhibit sensitivities to changes in different economic scenarios. For example, a customer with employment in an industry that is sensitive to housing price declines could lose their job in a recession that causes house prices to decline. This loss of income makes it a higher likelihood that this customer will be forced to miss a payment on their mortgage. Such an eventuality would not be true for a customer who is employed in a sector that might not be impacted by that economic scenario. That customer's ability to repay their mortgage would not be impacted by the change in housing price index, yet the values in the dataset would be the same for both.

Table 1 in the appendix provides a complete list of variables included in the dataset. There are sixteen variables for all 50,000 observations. The dataset contained no missing values, so no treatments needed to be applied to supplement any missing attributes. There are five variables that are categorical and serve as indicators, while the remaining variables are continuous. The dataset also contains two response variables. One is an indicator if the customer defaulted on the loan and the other is the dollar amount loss if the customer did default. There are two response variables indicating that two separate models will be necessary in the construction of a loss forecast, leaving the remaining 14 attributes to serve as predicting variables. The need for two separate models will be expanded on in the methodology section.

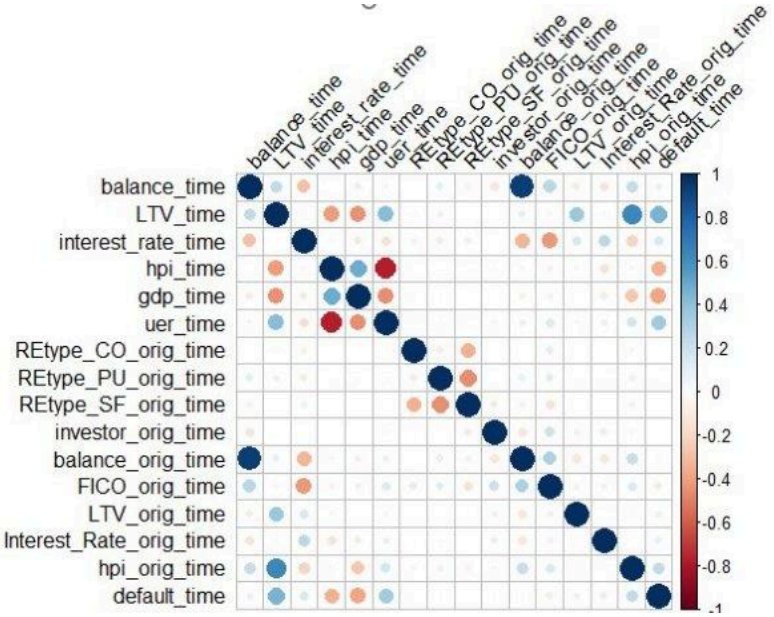
Correlation between variables:

Chart 1 contains a plot that shows the correlation of all variables in the dataset with each other. Correlation measures the *strength* and *direction* of the linear relationship between two data sources. If the value of the correlation coefficient is close to +1, then the variables are strongly linearly related and

[1] Yields on a fixed rate loan vary, but could be as high as 30% or more if the asset is credit card debt.

exhibit a direct (positive) relationship. If the value of the correlation coefficient is close to -1, then the variables are still strongly linearly related but exhibit an inverse (negative) relationship.

Chart 1: Variable Correlation Plot



In the mortgage dataset, the macroeconomic variables of unemployment rate, housing price index, and GDP all exhibit correlations with each other. Since all of these attributes are impacted by changes in the macroeconomic environment, it follows that there would be correlations between all of them. When GDP declines, the unemployment rate tends to increase and housing prices tend to decline. The response variable default, indicating if a customer defaulted on payment or not, appears correlated with the loan-to-value ratio (LTV_time), housing price index (HPI), gross domestic product (GDP), and unemployment rate (UER). Given these correlations, it is likely that these attributes will become important as models to predict customer defaults and loss are developed.

Another interesting observation is the moderately strong correlation between the predictor variables HPI, GDP, and UER. This potentially presents a problem of multicollinearity in the model development. Multicollinearity is when predictor variables in a regression model are highly correlated with other predictor variables. This indicates that these variables have a strong linear relationship with each other. A high correlation among predictor variables means it will be challenging for a linear regression model to determine the contribution of these predictors individually to explain the variance of the response variable. Problems multicollinearity contributes to include unstable coefficient estimates and inflated standard errors. Correlated predictors, by definition, behave similarly and exhibit similar signals for the model. The model is unable to properly determine the distinct effect of each variable on the response variable, which means that small changes in the data might cause large changes in the coefficients. Correlated predictor variables also inflate the standard error of the coefficients, potentially distorting the statistical significance of these coefficients. Several empirical methods exist that can test for the presence of multicollinearity. The model development section of the appendix contains a section evaluating multicollinearity for the models in this analysis

Customer variables:

Chart 2 in the appendix plots the distributions of some of the most correlated predictors against the binary response variable of default. The boxplots indicate that customers who default on their loan will typically have a lower credit score than customers who do not default on their loan. The inner-quartile range, as well as the median value, are both lower among customers that have defaulted on their loan. The plots also indicated that customers that default tend to have higher loan sizes and loan-to-value (LTV) ratio. Loan-to-value, which is the ratio of the amount of money the customer owes on the asset against the value of that asset, is typically an indicator of financial strength and liquidity of the

customer. A lower loan-to-value ratio presents less risk to the bank. The charts for loan size and LTV both indicate that customers that purchase larger homes or put less money down in the form of a down payment present a higher default risk.

Macroeconomic Variables:

As with the customer attributes in chart 2, chart 3 in the appendix plots the distributions of some of the more correlated predictors variables against the binary response variable of default. The boxplots indicate that customers who default on their loan tend to do so when the housing price index is lower and the unemployment rate is higher. This type of macroeconomic environment would indicate that a recession is taking place, so its intuition suggests that default rates would go up under those conditions. The third box plot plots interest rates on the mortgage against the customer's default behavior. At first glance, there is not much to glean from this chart. Both distributions appear relatively equal, aside from the significant number of outliers (represented by green dots) at the top of the non-default side of the plot. These loans are largely loans owned by investors, or entities with significant amounts of capital. These loans are typically priced differently, due to their complexity and duration, and usually default at much lower rates. When these outliers are omitted and only individual buyers are considered, the charts indicate that those loans with a higher interest rate tend to exhibit increased default volumes.

Loan-to-value:

Loan-to-value (LTV) appeared correlated with the response variable for customer defaults as well as several other variables in the correlation plot presented in chart 1. Chart 4 of the appendix presents LTV in a scatter plot against four other predictors: HPI, credit score, outstanding loan balance, and interest rate. The data points in each of the charts are colored by their default status. If a customer defaulted on the loan, the data point will be colored red. If the customer did not, then it will remain as black.

Immediately apparent is that most defaults occur when the LTV ratio is above 100, meaning that the amount that a customer owes is greater than the value of the home it is secured with. This is true across all four of the cuts provided in the diagram. On the LTV against score and LTV against balance charts, there appears to be no discernable pattern between LTV, defaults, and either balance size or credit score. This would indicate that regardless of credit score or outstanding balance, if a home loan reaches an LTV of 100 or above the likelihood of default is significantly higher. There do appear to be groupings on both the LTV against HPI and LTV against interest rate charts. Defaults appear to be clustered among instances with a lower HPI and a higher interest rate. When the housing price index falls, defaults among loans with an LTV above 100 will increase. Loans with interest rates above 8% also exhibit heightened default occurrences as well.

Methodology

Credit loss forecasting is typically conducted by first modeling the probability of default ('PD') and the loss at default ('LaD') for a financial instrument [5]. For each run of the model, each loan receives an estimate for both of these factors. To estimate the expected loss for each loan, a model development team multiplies the PD of that loan with the LaD of that loan. The total expected loss ('EL') of the portfolio then becomes the sum of the product of the PD and LaD for all of the financial instruments within that portfolio. Expected loss then can be written as the following equation:

$$\sum_{n=0}^{k} Probability\ of\ Default_i * Loss\ at\ Default_i \quad (1)$$

where k is the total number of loans in the portfolio, and i is each individual loan within the portfolio (k -> i). PD models predict a probability that is between 0 and 1 for each loan and are therefore styled as classification problems. LaD models estimate a continuous response variable, or an estimated financial loss for each loan, and are styled as classic regression problems.

Probability of Default Models:

PD models serve to estimate the probability that a customer is going to default on their loan. PD models produce a response variable between 0 and 1, which represents the probability that default will occur. Traditionally in credit loss forecasting, PD models are logistic regressions that took an additive form [5]. Much like regression problems, the PD for a loan could be estimated as an additive combination of several predictor variables that would be scaled by a regression coefficient. The functional form of a logistic regression includes a transformation via a 'link' function that would produce a probability rather than a continuous response[2].

Logistic regression also requires a selection of variables to use to evaluate a prediction. Oftentimes, utilizing all of the variables available will not produce the best results. Overfitting, multicollinearity, and a litany of other problems might cause the model that contains all of the variables available to not be the optimal selection. To decide which variables to use in a logistic PD model, several different optimization techniques could be used such as forward selection, best subset selection, and backward selection. Best subset selection typically always produces the best model, but can often be computationally expensive for datasets with tens of millions of observations. Developers dealing with significantly large datasets might opt for a more computationally efficient model development process such as forward/backward selection. Since this dataset has a manageable number of observations, best subset selection will be implemented for the logistic regression.

To draw a comparison to this 'baseline' method, the analysis will also explore two other types of advanced classification modeling techniques: k-nearest neighbors ('knn') and support vector machines ('svm'). KNN modeling functions off of a grouping type of logic. The algorithm will classify a data point based on how closely it resembles other data points that are nearby. The number of nearest data points to consider ('k') can be manually optimized by the developer based on selected performance measures. For classification problems like PD models, KNN will assign a value of default/no default based on the characteristics of other loans in its neighborhood. SVM modeling is another advanced modeling algorithm that can be used for both classification and regression. For PD models, SVM will find a boundary that best separates the 'defaults' and 'non-defaults'. Data points that exist on one side of the boundary will be

[2] This is a very simplistic definition of logistic regression, intended for a non-technical audience.

classified in one group and points on the other side will be classified in another group. SVM and KNN are both considered a supervised machine learning algorithm.

These two advanced model types were not selected at random. SVMs are known to perform well when the classification groups are unbalanced, meaning there is a larger presence of one group than another [8]. In the case of this data set less than 26% of the data belongs to the default class, indicating a significantly unbalanced group. SVMs contain inputs that can be modified ('tuned') to adjust for such an imbalance. SVMs also maximize the amount of space between the two class distinctions. This has the advantage of reducing the impact of outliers or noise [8]. Credit risk data typically contains variables that could be considered outliers, so having a model type that can accommodate this is advantageous. KNN holds similar advantages. Like SVM, it can be adapted to imbalance data. Simple adjust the value if *k* to address the imbalances in the data [9]. KNNs are also particularly good at capturing nonlinear patterns in the data. As seen in chart 4 of the exploratory data analysis, there are several nonlinear relationships that do exist in the dataset. Having a classification model that can accommodate the potential influence of these nonlinear relationships will be advantageous to include in the performance comparison.

Both the SVM and logistic regression also require optimization of a cutoff value. The cutoff value is the level of probability that separates each group. In the application for PD modeling, if the value for PD is above the cutoff value then the loan will be predicted to default. If it is below, then it will be predicted to be a non-defaulted loan. Cutoff values can be determined and optimized empirically[3].

Loss at Default Models:

LaD models estimate the financial loss that a bank will incur for each loan. LaD models produce a dollar ($) estimate for each loan, which typically also includes the amount that the bank expects to recover in any collection operations[4] [5]. LaD models are styled as linear regression problems where the financial loss of a loan can be predicted by an additive combination of several predictor variables scaled by a coefficient. Unlike PD models, there is no transformation via a 'link' function since regression problems produce a continuous response prediction as opposed to a binary output.

Much like logistic regression and PD modeling, linear regression models require a selection of variables to generate a prediction. As in PD modeling, including all possible variables in the model does not often produce the best output. Best subset selection and forward/backward selection are also common ways to determine the best model to utilize for a linear regression problem. Best subset selection will typically always produce the best possible model for the dataset and will be what is used in this analysis to generate a suitable LaD model for comparison.

As in the logistic regression example, two other advanced modeling techniques will be developed to be offered as challengers to the baseline linear regression model. Both random forest models and xGBoosting models are techniques that have become popular in the data science community. Both of these modeling types are ensemble models, meaning that they combine multiple models to form a best final model that is used in prediction. Random forests utilize decision trees, which is a modeling method that splits data into branches based on features and their predictive values. The resulting model framework appears like a tree. Random forests combine multiple decision trees to form a final model that will be used to generate predictions for a response variable. xGBoost also uses decision trees, but instead of averaging them together to form a final model it builds them sequentially. This allows each resulting tree in the model to learn from the errors of the prior tree, effectively correcting the errors of the model

[3] Further discussion on cutoff values for this modeling problem can found in the model development section of the appendix.
[4] For ease of example, recovery modeling was not carried out here and represents a possible extension of this analysis

before it. This allows xGBoost to capture complex patterns within data. Both of these modeling techniques require manual calibration/tuning of parameters. Random forests will need to determine the optimal number of features to randomly split within each decision tree and xGBoost requires tuning of seven parameters[5]. Once the parameters are optimized, the models can then be used to generate predictions.

Much like the logistic regression example, these two advanced model types were not selected at random. Both of them offer advantages over a linear regression based on how the model data and the form they take. Random forests are an average of many decision trees. This has advantages in that it allows non-linear and complex relationships to be more easily modeled [9], while linear regression requires an assumption of linearity between the variables and struggles to model non-linear relationships. xGBoost models have a similar advantage to random forests. xGBoost does not require linearity between the response variable and the predictors. xGBoost also has the advantage of being a sequential learning process, meaning that models can effectively learn from the errors of other models. No such correction process exists in a regression model.

Results

To develop the models and generate predictions, the 50,000 observation dataset was split into a training and testing dataset. Three-fourths of the dataset was set aside into the training set and would be used for model development. The remainder of the data would be set aside into the testing set and used for performance evaluation. The advantage of this framework is that the model can be evaluated for fairly on data that it has not been trained on. This allows the consumers of the model to better understand and approximate the models performance on real-world data. The six models were then optimized and tuned as necessary. For the models that require variable selection, best subset selection was used to build the most optimal model. For the ensemble models that require parameter tuning, tuning was carried out using an appropriate method. Once the models were developed, they were then evaluated on the testing dataset and evaluated for model performance comparisons. Further technical details on model development can be found in appendix II.

Probability of Default Model Performance:

Adequately comparing the performance of the three classification PD models requires selection of performance criteria that can be measured for all three models. Area-under-the-curve (AUC), Accuracy, and F1 score are all common classification model performance measures. AUC measures the ability of a binary classification model to separate each class from each other. A value of one indicates a model that perfectly classifies the data, a value of 0.5 indicates a model that is as effective as a random guess, and a value of 0 represents a model that is perfectly incorrect. While a model will likely never achieve an AUC of 1, it's the goal of every development team to land as close to 1 as possible. Accuracy measures how many correct predictions the model made out of the total predictions made. The closer to 100% indicates that the model is perfectly predicting the data - both for defaults and non-defaults. F1 score is another measure of accuracy that balances precision (the proportion of true default predictions out of all default predictions correct or incorrect) and recall (the proportion of true default predictions to true defaults and wrongly predicted non-defaults).

[5] Optimization of these parameters is detailed in the appendix

Training and testing error are also ways to compare and contrast model performance. Training error reflects the model's performance on the data that was used to develop it. Testing error represents the model's performance on the data that was set aside and not used to develop it. As mentioned previously, testing data is intended to more closely mimic the model's performance on real world data, as it is data that the model has not been developed on. The lower the testing error, the better the performance of the model. The formula for training/testing error is to simply sum the predicted value minus the actual value, square that difference, then divide the resulting sum by the total number of observations in the dataset.

Chart 5 displays the AUC, Accuracy, and F1 score for all three of the classification models. KNN performs better than logistic regression and SVM in two of the three performance measures. Logistic regression has a better AUC than KNN, while KNN has better accuracy and F1 score than the other two modeling types. The superior performance of KNN here can likely be attributed to the fact that the decision boundary between the two classes is highly nonlinear. Based on chart 3 and 4 provided in the appendix and given the complexity of the relationships in consumer credit modeling, many of the relationships in the dataset appear to be non-linear. Logistic regression and SVM perform better when there is a clear and linear decision boundary between the classes.

Chart 5: Probability of Default Model Performance Measures

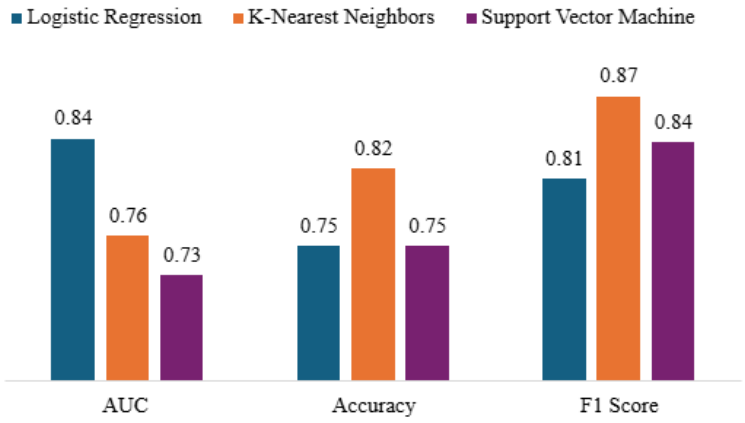


Table 2: PD Models Training/Testing Error

Model	Training Error	Testing Error
Logistic Regression	0.239	0.251
Support Vector Machine	0.245	0.252
K-nearest neighbors	0.176	0.192

Table 2 displays the training and testing error for all three classification models. The logistic regression and SVM have a similar testing error to each other, but the KNN model performs the best. This is additionally verified by the KNN having the highest accuracy of all of the models as well. Based on the measures evaluated here in conjunction with the models' training and testing errors, the KNN appears to be the best performing probability of default model on this dataset.

The appendix II section contains more detailed information about model development for the three probability of default models. After parameter tuning and variable selection, the best logistic regression model is a nine-variable model, the best KNN model is a model with *k* equal to 22, and the best support vector machine model is a model with a cost value equal to 1.25. Multicollinearty was mentioned as a potential concern for the logistic regression. Variance inflation factor (or 'VIF') is a tool to measure the presence and influence of multicollinearity [10]. A VIF value below 5 is considered normal and a VIF above 10 indicates multicollinearity. For the logistic regression, all nine variables were below 5. Multicollinearity is therefore not a concern. For the SVM and KNN models, multicollinearity is not a significant concern. These models do not rely on parameter estimates as in logistic regression. KNN classifies data points based on observations that are nearby to a data point and SVM makes classifications based on a support vector. Neither of these procedures involve the selection of coefficients, as in logistic regression.

Loss at Default Model Performance:

Similarly to the PD model performance measures, LaD model evaluation will also require selection of performance criteria that can be applied to all three models. To measure the performance of these three models, training error and testing error will be compared. The formula for training and testing error in this application will be similar to the formula used in the PD model development training/testing error: the square of the difference between the predicted value minus the actual value divided by the total number of observations in each dataset.

Table 3 displays the testing and training error for all three models. The strongest performing model is the random forest, followed by the xgBoost model, and finally the linear regression model. Notable is the significant performance disparity between the linear regression and the two ensemble methods. This is likely because of the presence of a number of nonlinear relationships in the dataset. The random forest and xGBoost models were able to accommodate those nonlinear relationships, while the linear regression model had a degree of difficulty accommodating those relationships. While normally protocol would demand exploration of nonlinear fits and/or transformations to the linear regression model, Federal Reserve guidelines are fairly strict on the hesitation to do this due transformations increasing interpretability complexity [5]. To conform with those guidelines spelled out in IFRS-09, this analysis will not explore any sort of non-linear fits or transformations to the data. Based on the performance of these three models, the random forest is the strongest performing model on the data.

Table 3: LaD Models Training/Testing Error

Model	Training Error	Testing Error
Linear Regression	48,242	50,792
Random Forest	7,402	9,490
xgboost	9,013	10,575

The appendix II section contains more detailed information on model development, including parameter tuning and variable selection. After parameter tuning and variable selection, the linear regression ended up being an 11 variable model and the random forest ended up with a mtry value of 11. The xGBoost model also had seven parameters that were tuned and listed in the table in the appendix. The previous concerns about multicollinearity were vetted using the VIF (variance inflation factor), which is a measure to detect multicollinearity in regression models. The results of this test are also present in the appendix. VIF values above 5 are moderately concerning and VIF values above 10 are significantly concerning. For all of the variables in the linear regression, VIF values remained below 5. Multicollinearity is therefore not a major concern for this regression model. An advantage of the two advanced models selected here is that multicollinearity typically is not a concern. The structure of ensemble models do not rely on individual feature selection and individual feature coefficients, as in linear regression models, thereby mitigating the concern over multicollinearity [8]. Ensemble methods are effective in handling redundant information that multicollinearity implies.

Expected Loss Forecast:

The analysis seeks to evaluate the optimal combination of PD and LaD modeling techniques in predicting an expected loss and minimizing excess capital. *Excess capital* can also be thought of as either the training or testing error of the datasets. By definition, excess capital represents the delta between the *actual* losses in the dataset and the *predicted* losses of the dataset, which is an identical definition to training/testing error. Table 3 contains all possible combinations of the different modeling types in this

analysis. Each PD model type would be paired with an LaD model type and a resulting expected loss and excess capital calculation is determined.

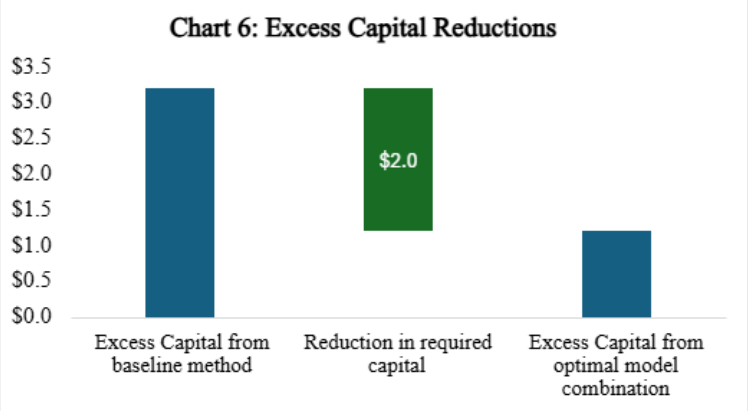
The training dataset contained $127M in actual losses, the test dataset contained $42.6M in actual losses. The baseline logistic regression/linear regression models had an error rate of 5.6% on the training dataset and 7.6% on the test dataset, resulting in an excess capital of $7.2M and $3.2M respectively. The best performing combination of PD and LaD models is the KNN and random forest model. This combination resulted in a training error of 2.2% ($2.8M excess capital) and a testing error of 2.8% ($1.2M in excess capital). The advanced model types were among the better performing in general, as the logistic/linear regression combination had nearly the highest error rate of all of the models. The only two combinations that were higher also contained baseline methods. The recommendation here is clear: advanced modeling techniques can improve credit loss forecasts and reduce excess capital on a financial institution's balance sheet.

Table 3: Expected Loss Estimates

Probability of Default Model	Loss at Default Model	Training Dataset		
		Loss Estimate	% Error	Excess Capital
Logistic Regression	Linear Regression	$134,920,669	5.6%	$7,154,884
Logistic Regression	Random Forest	$134,665,137	5.4%	$6,899,352
Logistic Regression	xGBoost	$134,281,840	5.1%	$6,516,055
K-Nearest Neighbors	Linear Regression	$139,392,471	9.1%	$11,626,686
K-Nearest Neighbors	Random Forest	$130,576,632	2.2%	$2,810,847
K-Nearest Neighbors	xGBoost	$136,964,922	7.2%	$9,199,137
Support Vector Machine	Linear Regression	$123,677,280	-3.2%	-$4,088,505
Support Vector Machine	Random Forest	$131,726,524	3.1%	$3,960,739
Support Vector Machine	xGBoost	$134,665,137	5.4%	$6,899,352
Probability of Default Model	Loss at Default Model	Testing Dataset		
		Loss Estimate	% Error	Excess Capital
Logistic Regression	Linear Regression	$45,839,575	7.6%	$3,250,980
Logistic Regression	Random Forest	$45,283,538	6.3%	$2,694,943
Logistic Regression	xGBoost	$45,064,573	5.8%	$2,475,978
K-Nearest Neighbors	Linear Regression	$46,940,201	10.2%	$4,351,606
K-Nearest Neighbors	Random Forest	$43,783,246	2.8%	$1,194,651
K-Nearest Neighbors	xGBoost	$45,987,601	8.0%	$3,399,006
Support Vector Machine	Linear Regression	$40,277,781	-5.4%	-$2,310,814
Support Vector Machine	Random Forest	$44,278,220	4.0%	$1,689,625
Support Vector Machine	xGBoost	$45,444,917	6.7%	$2,856,322

Recommendations:

The most important objective of this analysis is reducing the amount of excess capital (error) in the credit loss forecasts. A lower error in forecast results in a smaller amount of excess capital that must be held on a firm's balance sheet. Chart 6 shows a simplified comparison of the excess capital required for the baseline method compared to the optimal model combination. The combination of best performing models reduces excess capital by over 60% - from $3.2M to $1.2M. This represents a significant amount of capital freed up for investment. The results suggest that using more advanced



models for credit risk modeling can indeed form more accurate credit risk models that allow a bank to maintain accurate credit loss forecasts without holding too much in excess capital. The recommendation here would be to begin development of these more advanced models to run in parallel with the existing baseline method and monitor their performance over longer periods of time. If the models consistently perform better than the baseline methods as they have in this analysis, an ideal scenario would be for a financial institution to implement these forecasting methods. This would ensure that only excess capital that is required is kept on the balance sheet while also providing an accurate and reliable loss forecast.

Conclusion

Ever since the advent of the financial crisis in 2008, the Federal Reserve Bank has required financial institutions to forecast their expected credit losses through easily interpretable statistical models. This analysis sought to evaluate if loss forecasts can be improved through the use of advanced modeling techniques over the baseline methods required by the Federal Reserve. A dataset of 50,000 mortgages was used to develop three probability of default models and three loss at default models. For each model, two challenger advanced modeling techniques were evaluated against the baseline logistic/linear regression framework required by the federal reserve. The dataset was then split into training and testing datasets for use to build and test the models. Each of the six models were trained on the training dataset, with relevant tuning/variable selection procedures deployed. The models were then evaluated on the test dataset for predictive performance. The k-nearest neighbors model performed the best for the PD model and the random forest performed the best for the LaD model. Neither of these models are the baseline methods required by the Federal Reserve.

Next, every combination of PD and LaD model was evaluated to calculate expected loss. The baseline combination of linear/logistic regression had a testing error rate of 7.6%. The optimal combination of models (KNN and random forest) had an error rate of 2.8%, representing a significant improvement over the required methods by the Federal Reserve. The baseline combination would have required a bank to hold $45.8M in reserves, which works out to about $3.2M in excess capital over actual losses. The optimal combination required $2.0M less in excess capital, as it predicted losses of $43.8M - significantly closer to the actual credit losses in the test dataset of $42.6M. Utilizing the advanced models in the optimal combination would have saved >60% in excess capital that could have been deployed elsewhere. This ratio of savings represents a significant capital reduction at most large financial institutions as well as significant foregone investment opportunities should the baseline modeling framework be utilized.

A few important caveats exist before implementing the results of this analysis. First, this analysis was conducted on *home lending* data. Different lending portfolios have vastly different credit dynamics and customer bases, so what is applicable for this sample portfolio might not be applicable to a credit card or auto loan portfolio. Changes in the consumer payment hierarchy, income dynamics, and overall credit profiles might lead to different conclusions on these other portfolios. To evaluate the robustness of the conclusions presented here, a logical extension includes running a similar analysis on these other lending portfolios. Another important caveat comes to scalability and size. The portfolio of loans analyzed here is $8.8B in size. Put into context, that would make this portfolio #161 out of the list of large US banks [7]. Credit and portfolio dynamics change rapidly at different sizes of institutions. Additionally, what some

institutions might have in terms of data assets others might not have access to. Evaluating this analysis against different sizes of bank portfolios would also be a necessary next step. Testing whether or not advanced models outperform baseline methods and evaluating how much excess capital those advanced models save would be a next step to solidifying the robustness of the results presented..

Appendix 1: Tables and Charts

Table 1: Complete List of Variables in the Mortgage Dataset.

Variable Name	Type	Response Variable?	Binary?	Description
FICO_orig_time	Customer	N	N	FICO Score at Origination
balance_orig_time	Customer	N	N	Original Loan Size
balance_time	Customer	N	N	Remaining Loan Size
REtype_PU_orig_time	Customer	N	Y	Urban Development Indicator
REtype_CO_orig_time	Customer	N	Y	Condominium Indicator
hpi_orig_time	Economic	N	N	House Price Index at Origination
LTV_time	Customer	N	N	Loan-to-Value Ratio
uer_time	Economic	N	N	Unemployment Rate
investor_orig_time	Customer	N	Y	Indicator if the Owner is an Investor
Interest_Rate_orig_time	Customer	N	N	Interest Rate of the Loan (%)
LTV_orig_time	Customer	N	N	Loan-to-Value Ratio when loan was booked
REtype_SF_orig_time	Customer	N	Y	Indicator for Single Family Home Investor
gdp_time	Economic	N	N	Gross Domestic Product
hpi_time	Economic	N	N	House Price Index
default_time	Customer	Y	Y	Indicator if the Customer Defaulted
Loss	Customer	Y	N	Credit Losses if the Customer Defaulted

Chart 2: Boxplots of the default attribute against customer variables

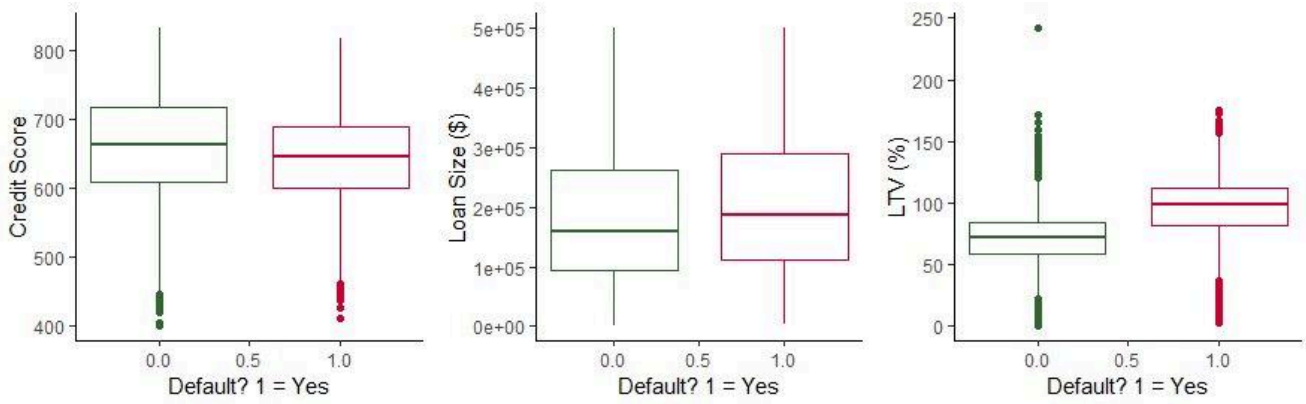


Chart 3: Boxplots of the default attribute against macroeconomic variables

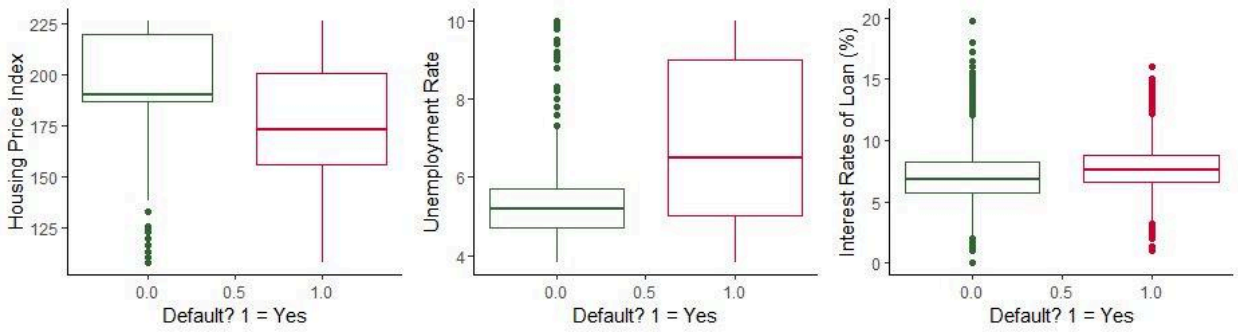
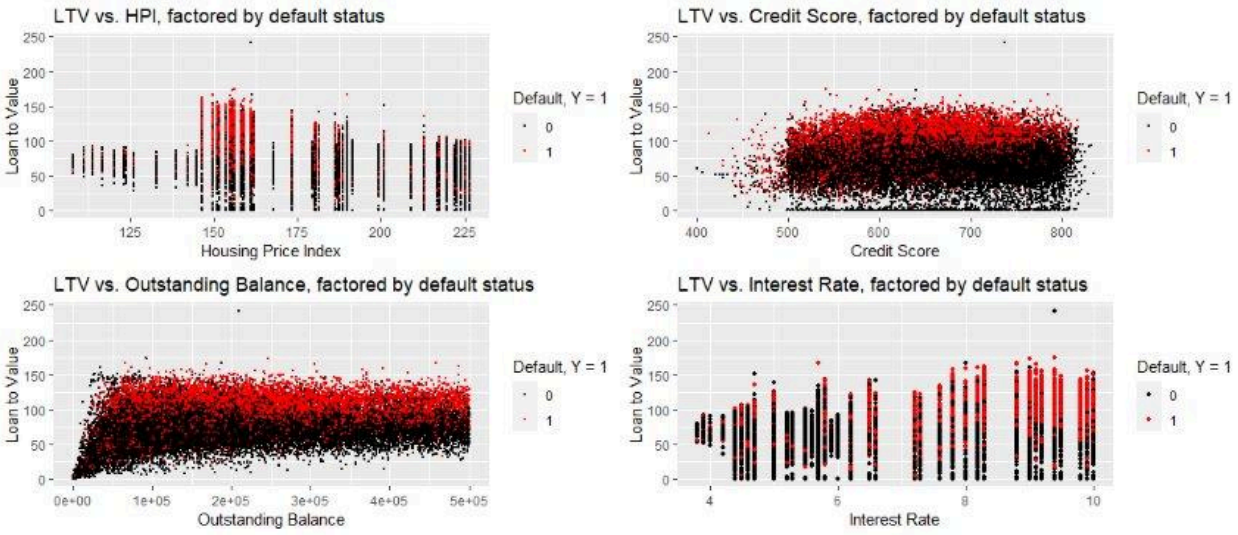


Chart 4: Matrix of LTV against macroeconomic attributes and customer attributes



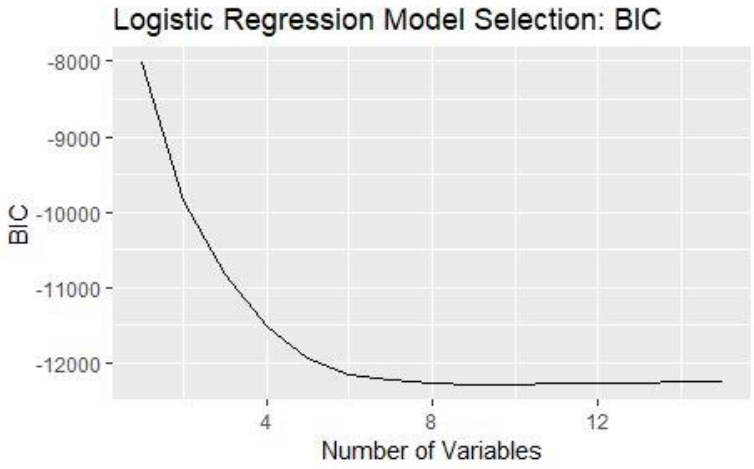
Appendix II: Model Development

Probability of Default Model Development

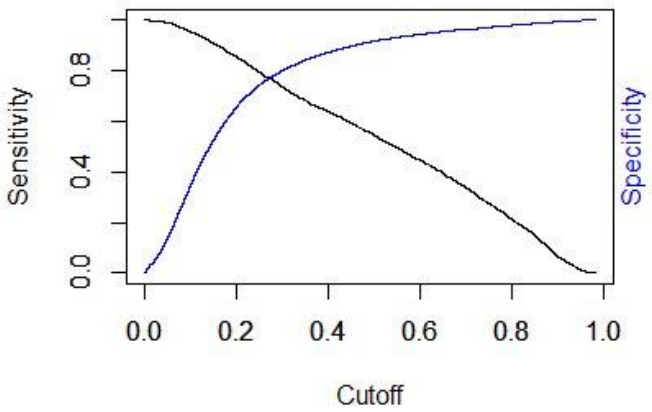
Three different classification models were used for the probability of default: logistic regression, k-nearest neighbors, and support vector machine. Each of these model types requires either variable selection or parameter tuning.

Logistic Regression (Baseline method):

Building a logistic regression model requires both optimization of the variables included in the model as well as selecting a cutoff value for the output that determines default/non-default (which class the predicted response belongs to). Best subset selection is the most optimal way to choose the strongest performing model. Certain model performance measures like AIC, BIC, or Mallows Cp can be used to determine the optimal model at each number of variables. BIC is preferred for models that are used for predictive purposes. For the logistic regression model, the best performing model in the Logistic Regression Model Selection: BIC chart is the model with 9 variables. It has the lowest value for BIC of all of the possible best models at each number of variables. The 9 variables are: LTV, interest rate, GDP, unemployment rate, investor indicator, balance, FICO score, HPI, and HPI_time.



Next, the optimal cutoff value needs to be determined. Each loan in this model will receive a probability of default value that ranges between 0 and 1, but typically will not be exactly 0 and exactly 1. For each classification problem, there exists some value of a cutoff that can serve as the best level to divide each class. The intersection of sensitivity and specificity is one way of selecting this cutoff value. The chart to the right plots sensitivity and specificity for all possible cutoff values in the model. The optimal cutoff value is 0.26. The selected logistic regression model will be a 9-variable model with a cutoff value of 0.25



Multicollinearity

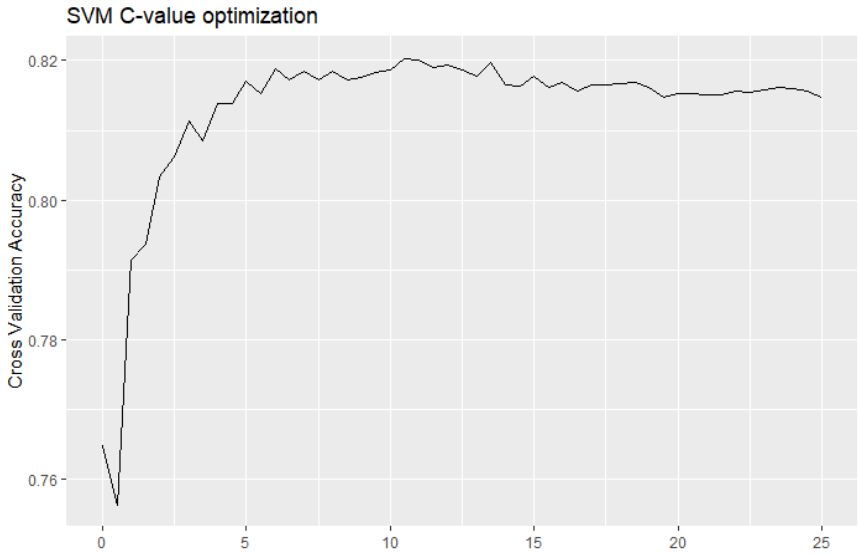
In the exploratory data analysis portion of this analysis, multicollinearity was suspected as a potential concern for both the logistic regression and the linear regression. To evaluate this concern, several tests exist with the simplest and easiest to understand test being variance inflation factor (or VIF). VIF quantifies how much the variance of a regression coefficient is affected by its correlation to other predictors. VIF will regress (model) each predictor variable against all other predictors. The R^2 of the resulting model is then used to calculate the VIF of that predictor. A VIF below 5 is generally considered to be unconcerning, a VIF above 10 suggests very high multicollinearity that will require correction [10]. The logistic regression VIF table contains the values of VIF for each of the 9 predictors in the developed logistic regression model. All of the predictors have a VIF below 5, indicating that multicollinearity is not a concern.

Logistic Regression VIF

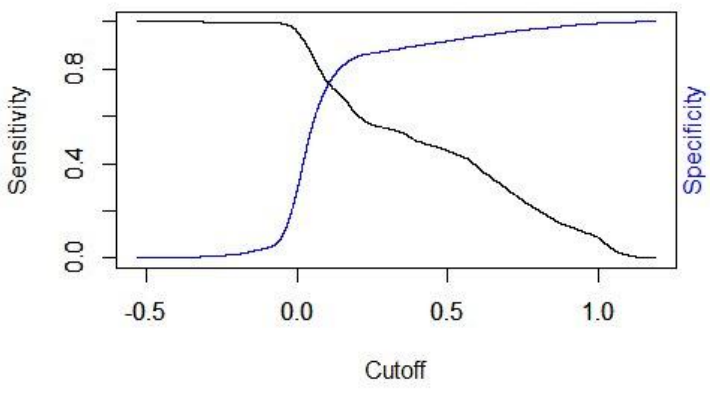
Variable	VIF Value
LTV_time	2.96
interest_rate_time	1.52
gdp_time	1.39
uer_time	2.08
investor_orig_time	1.12
balance_orig_time	1.26
FICO_orig_time	1.47
LTV_orig_time	2.10
hpi_orig_time	2.86

Support Vector Machine:

The support vector machine requires input of a cost parameter (*c*) as well as an optimization of a cutoff value like in logistic regression. To determine the best value for c, several values of c can be plotted against the cross validation accuracy of that value of *c*. Cross validation involves splitting the data into multiple folds to train a model and a single fold to test the model. This process is done *k* times and is often referred to as 'k-fold cross validation'. The error is averaged across all of the folds in order to produce a final result. The value of *c* with the smallest error is the optimal value of *c*. Values of C were tested between 0 and 25 in intervalues of 0.25. In the C-value optimization chart presented here, that optimal value is 1.25.



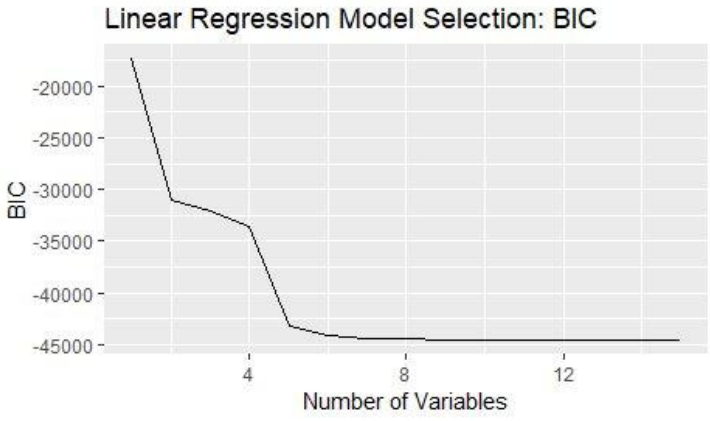
Similar to the logistic regression model, this application of an SVM model also requires optimization of a cutoff value. Normally the SVM defaults to 0.5, but for this project it's ideal that the output is as optimal as possible. Manually determining this level will ensure a more optimal solution. Like with logistic



regression models, the most optimal cutoff value is at the intersection of sensitivity and specificity. For the SVM model here with *c*-value of 1.25, the optimal cutoff value is 0.1.

K-nearest neighbors:

KNN requires only the optimization of a single parameter - the number of nearby points to consider. Similar to the SVM optimization process, several values of k between 1 and 200 were tested. The cross-validation error for each model at each level of k was then calculated and plotted on the chart at the right. The model with the lowest cross validation error would be the optimal model selected. In this case, a model with a value of $k = 22$ is the optimal number of nearby points to consider. This application of KNN does not require a cutoff value, as in SVM or logistic regression, as it does not assign a probability. It only groups the data points into a class based on data points near it.



Loss at Default Model Development:

Three different models were used for the loss at default: linear regression (which serves as the baseline), random forest, and xGBoost. Each of these model types requires either variable selection or parameter tuning.

Linear regression (Baseline method):

Similar to logistic regression, linear regression requires optimization of the number of variables and which variables to include. As in logistic regression, best subset selection is the approach to select the strongest performing model possible of all combinations of variables in the dataset. Comparing the BIC for each of the best models at each size will reveal the best performing model. In this case, the best performing model is an 11 variable model that includes LTV, interest rate, GDP, unemployment rate, investor indicator, balance, credit score, HPI, HPI_time, remaining loan balance, and LTV_time. Linear regression does not require the selection of a cutoff value.

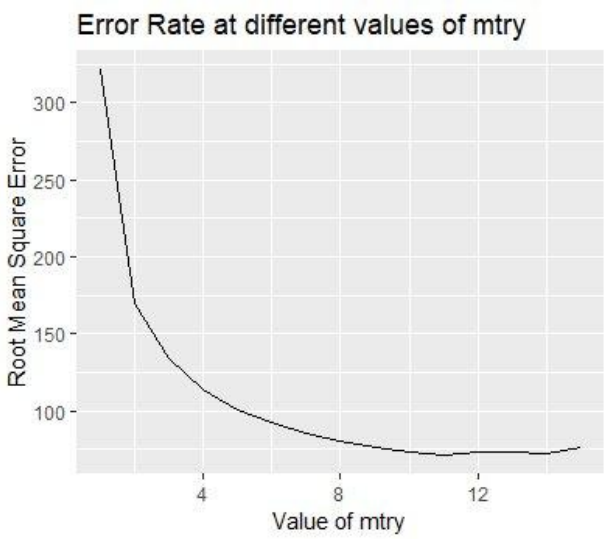
Multicollinearity:

As in the logistic regression, linear regressions can also be influenced by correlations between the predictor values. VIF can also be deployed to analyze the presence of excessive multicollinearity in a similar manner as the logistic regression. The same cutoff logic applies: a VIF below 5 is considered to be unconcerning while a VIF above 10 indicates a model that has a high degree of multicollinearity. As with the logistic regression, all of the VIF values for the linear regression are below 5. Multicollinearity is not a significant concern for this model.

Linear Regression VIF	
Variable	**VIF Value**
balance_time	4.86042
LTV_time	3.607921
interest_rate_time	1.576859
hpi_time	3.838347
gdp_time	1.589447
uer_time	2.780904
REtype_SF_orig_time	1.027162
balance_orig_time	3.880365
FICO_orig_time	1.295809
Interest_Rate_orig_time	1.11951
LTV_orig_time	2.096341

Random Forest:

The first 'advanced' model developed in this part of the exercise is a random forest. Random forest is an ensemble model that averages several hundred trees together into a single final model. If computational concerns were an issue, it would require a determination of the number of trees necessary. The dataset here is small enough to allow for a large number of trees without those concerns, so the number of trees was set to 500. The only parameter that will need to be tuned is the parameter 'mtry', which is the number of variables per split in the random forest. All of the possible variable sizes were tested in the chart at the right. The optimal value of mtry is the value with the lowest root mean squared error. For the random forest here, that value of mtry is 11.



xGBoost:

The second advanced model developed in this part of the exercise is an xGBoost model. xGBoost is an ensemble model that utilizes a sequential method. Each successor model in an xGBoost model learns from the errors of the model before it. There are seven parameters that need to be tuned for this model. To determine the best values for each of the model, a random grid search was conducted. The grid search included an array of randomly selected values for each of these parameters. This is typically a much for efficient process for determining the optimal values of these hyperparameters - especially when there are large ranges possible.

Rather than produce a panel of seven separate charts, the better performing results are tabulated in a table below. The combination of parameters that produces the lowest RMSE/MAE are the optimal parameters for the model. The optimal values are highlighted below:

eta	max_depth	gamma	colsample_bytree	min_child_weight	subsample	nrounds	RMSE	MAE
0.464	7	1.783	0.416	19	0.339	70	25488	12376
0.403	8	2.733	0.541	19	0.793	139	19417	8286
0.160	8	5.653	0.339	5	0.412	297	17800	7503
0.288	5	9.922	0.500	17	0.337	317	17322	8033
0.131	4	0.365	0.342	17	0.829	374	14894	6423
0.332	2	4.667	0.685	2	0.380	395	17545	9265
0.583	3	0.320	0.620	15	0.682	475	15102	6835
0.259	5	6.454	0.595	12	0.529	559	10,575	24,83
0.578	2	4.195	0.571	15	0.311	601	22766	12776
0.063	7	9.423	0.380	12	0.476	604	15754	5593
0.428	9	5.811	0.629	8	0.741	627	19607	7922

References

[1] Federal Reserve History. (n.d.). *The Great Recession and its aftermath.* Retrieved November 21, 2024, from **https://www.federalreservehistory.org/essays/great-recession-and-its-aftermath**

[2] **Bank of America.** (2024). *Regulatory and other filings: SEC filings - CCAR submission guidelines*. Retrieved from https://investor.bankofamerica.com/regulatory-and-other-filings/all-sec-filings/xbrl_doc_only/6549

[3] **Nasdaq.** (2024, August 12). *CCaR model guidelines*. Retrieved from https://www.nasdaq.com/CCaR_model_guidelines20240812

[4] **Federal Reserve Board.** (n.d.). *Comprehensive capital analysis and review: Questions and answers.* Retrieved November 21, 2024, from https://www.federalreserve.gov/publications/comprehensive-capital-analysis-and-review-questions-and-answers.htm

[5] **Board of Governors of the Federal Reserve System.** (2011). *Supervisory guidance on model risk management* (SR 11-7 Attachment). Retrieved from https://www.federalreserve.gov/supervisionreg/srletters/sr1107a1.pdf

[6] **Credit Risk Analytics.** (n.d.). *Credit Risk Analytics: Making Better Decisions through Data and Modeling.* Retrieved from http://www.creditriskanalytics.net/

[7] **Board of Governors of the Federal Reserve System.** (n.d.). *Release of the latest data on large bank reporting* [Federal Reserve statistical release]. Retrieved from https://www.federalreserve.gov/releases/lbr/current/

[8] Vishal, Sharma., et al. (2020). Support Vector Machine. In *Comprehensive Chemical Engineering Handbook* (3rd ed., Vol. 2, pp. 245-260). Elsevier. Retrieved from https://www.sciencedirect.com/topics/chemical-engineering/support-vector-machine

[9] Guo, Gongde. et. al. (2004) "KNN model-based approach in classification," *ResearchGate*. [Online]. Retrieved from https://www.researchgate.net/publication/2948052_KNN_Model-Based_Approach_in_Classification.

[10] **"Variance inflation factor (VIF),"** *Investopedia*. [Online]. Available: https://www.investopedia.com/terms/v/variance-inflation-factor.asp.