MORTGAGE LOAN VALUE PREDICTION WITH MACHINE LEARNING

by

Thomas R. Billman III

A paper submitted in partial fulfillment of the requirements to complete Honors in the Department of Mathematics and Statistics.

Approved By:

Examining Committee:

Dr. Yishi Wang
Faculty Supervisor

Dr. Zhuan Ye

Dr. Joseph Farinella

Dr. Zhuan Ye
Chair, Mathematics and Statistics

Honors Council Representative

Director of the Honors Scholars College

University of North Carolina Wilmington

Wilmington, North Carolina

April, 2017

TABLE OF CONTENTS

ABSTRACT

This project uses mortgages' Net Present Value as a continuous response variable representing their value. By using contemporary machine learning algorithms such as Random Forest and Random Generalized Linear Models we predict these loan values on a single loan level from data available at loan origination. This project shows that these machine learning algorithms outperform traditional regression and classification methods.

1 Introduction

Our research primarily focuses on using contemporary machine learning techniques to predict mortgage values more accurately than traditional linear methods. Previous research has focused on classifying loans at time of origination as prepaid, paid as planned, or default[4].Through literature review, we find related work, but nothing using the Net Present Value (NPV). While it is important to predict the status of a loan, banks may have more interest in profitability prediction. As a proxy for profit, we compute the NPV of each loan which adjusts all payments to the time the loan was issued. This is different way to compare the financial impact of loans compared to end state classification methods. Additionally, once we calculate NPVs we predict them using data available at time of loan origination.

2 Literature Review

A mortgage is a loan that is secured by real estate [8]. If a borrower stops making payments on your loan it goes into default, and the bank can foreclose on your property. Through the process of foreclosure, the bank that wrote the mortgage claims the property from the borrower. The bank then sells the property and uses the proceeds to recover the rest of their outstanding balance. If the property has dropped in value, it is possible that the bank can lose large sums of money by writing mortgages, so it is important to make sure banks select borrowers who are likely to make their payments. Due to the high financial stakes, the ability to determine which borrowers are mortgage-worthy is very important and the subject of much research.

One project used contemporary machine learning techniques to model whether loans will carry out as planned, end in default, or end prepaid [4]. This project considered methods such as Binary Logit, Multinomial Logit, K-Nearest Neighbors, K-fold

Cross Validation, and Random Forest. Of the models considered, the most accurate model was random forest (RF) classification. This model could classify loans with 93% accuracy. Given that loans ending in default can have large differences in financial impact, the NPV analysis helps with this issue. Due to the proven accuracy of RF classification, we decided to use random forest modeling in our analysis.

Another project involved an unprecedented dataset of 120 million prime and subprime mortgages from 1995 to 2014[5]. After adding local economic metrics to their dataset, neural networks were used to predict how many loans would be end as either prepaid or default within random portfolios of thousands of loans. Their research showed that neural networks considerably outperformed similar analysis using traditional logit techniques. This is particularly impactful for agencies that package and sell mortgage-backed securities as it can drastically improve the methods of choosing loans for their products. This is also a good indicator for our project, that machine learning algorithms will yield better predictions of NPV as compared to linear regression.

3 Our Dataset

Our dataset was obtained from the Federal Home Loan Mortgage Corporation, better known as Freddie Mac, which is a public government-sponsored enterprise. We used their Single Family Loan Level Dataset, which lists origination and performance data for loans based on financial quarter of origination [1]. The dataset is composed of two files where the first file lays out the details of each loan's origination. It contains 391,419 observations of 26 variables, which are:

Credit Score	A number summarizing the borrower's creditworthiness and prepared by third parties

[1] The dataset can be found at http://www.freddiemac.com/research/datasets/sf_loanlevel_dataset.html, and you must register to download it

First Payment Date	Date of the first scheduled payment
First Time Homebuyer Flag	Indicates if an individual is 1) Purchasing the mortgaged property, 2) will reside in the property as primary residence, 3) does not have ownership interests in other residential properties
Maturity Date	Date of the final scheduled payment
Metropolitan Statistical Area	Similar to Zip Codes, but for large metropolitan areas containing 2.5 million people or more. These are defined by the US Census
Mortgage Insurance Percentage	Percentage of loss coverage on the loan, to be paid to Freddie Mac in the event of a default
Number of Units	Number of properties covered by this mortgage
Occupancy Status	Denotes whether the home is owner occupied, a second home, or investment property
Original Combined-Loan-to-Value	Original mortgage loan amount plus possible second mortgage amount divided by initial property value
Original Debt-to-Income Ratio	Borrower monthly income divided by monthly mortgage payment
Original Unpaid Balance (UPB)	Initial amount loaned in the mortgage note
Original Loan-to-Value	Initial mortgage loan amount divided by initial property value
Original Interest Rate	Original rate indicated on mortgage note

Channel	What type of organization sold Freddie Mac this loan (Retail, Broker, etc.)
Prepayment Penalty Flag	Indicates if the borrower is penalized for prepayment
Product Type	All entries are Fixed Rate Mortgages
Property State	The U.S. State the property is located in
Property Type	Indicates property type (Single-Family Home, Condo, Co-op, etc)
Postal Code	First three numbers of the property's Zip Code
Loan Sequence Number	Unique identifier assigned to each loan
Loan Purpose	Indicates if the loan is to purchase the house, or refinance the property
Original Loan Term	Number of payments calculated from First Payment Date and Maturity Date
Number of Borrowers	Number of Borrowers obligated to repay the mortgage (1 or >1)
Seller Name	Entity who sold the loan to Freddie Mac
Servicer Name	Entity who is currently servicing the loan on Freddie Mac's behalf
Super Conforming Flag	Loans that exceed conforming loan limits
Pre-HARP Loan Sequence Number	Links a HARP loan to its pre-HARP origination data

This is the data we will be using to predict NPV, as it is all collected during the loan selection process.

The second file of the dataset contains monthly performance data for each loan where each loan has an entry for every month regarding the status of the loan. This file has 2,311,802 observations of 23 variables such as:

Loan Sequence Number	Same number found in origination file and used to link the two
Monthly Reporting Period	Current Month of entry
Current Actual UPB	Mortgage ending balance for the monthly reporting period. It includes scheduled and unscheduled principal reductions
Current Loan Delinquency Status	Continuous number of months since Due Date of Last Paid Installment (DDLPI)
Loan Age	Number of months since the origination of the loan
Remaining Months to Legal Maturity	The remaining number of months until the mortgage Maturity Date
Repurchase Flag	This indicates loans that have been repurchased or made whole
Modification Flag	This indicates that the loan has been modified
Zero Balance Code	A code indicating why the loan's balance was reduced to zero (1 = Prepaid/Matured Voluntarily, 3 = Foreclosure, etc.)
Zero Balance Effective Date	The month in which the event triggering the Zero Balance Code took place
Current Interest Rate	The current interest rate on the mortgage after any modifications

Current Deferred UPB	Current amount of non-interest bearing UPB (Only occurs in the event of some loan modifications)
Due Date of Last Paid Installment (DDLPI)	The date that the loan's scheduled interest and principal payments were paid through, regardless of when last payment was actually made
MI Recoveries	Proceeds received from mortgage insurance in the event of default
Net Sales Proceeds (NSP)	Amount received from sale of property less selling expenses
Non-MI Recoveries	Other proceeds such as tax, insurance, etc. paid to Freddie Mac
Expenses	Expenses Freddie Mac bears in the event of foreclosure. This is an aggregation of Legal Costs, Maintenance and Preservation Costs, Taxes and Insurance, and Miscellaneous Expenses
Legal Costs	Legal costs associated with sale of property (not included in NSP) in the event of foreclosure
Maintenance and Preservation Costs	Costs associated with maintaining property during foreclosure
Taxes and Insurance	Cost of taxes and insurance incurred with sale of property

Miscellaneous Expenses	Other expenses associated with sale of property
Actual Loss	Default UPB - NSP + Delinquent Accrued Interest - Expenses - Recoveries where Delinquent Accrued Interest is the interest owed on payments missed since DDLPI
Modification Cost	Costs associated with a rate modification event

The Actual Loss column is particularly relevant to our research, as it gives us a comprehensive overview of losses suffered by the bank holding the loan in the event of default. Between the loan origination and performance data, we can accurately asses how valuable each loan was for the bank at the time it was written and associate that with data collected at origination. It is also important to note that given the time restrictions of our research we only used the first quarter of 1999 as our dataset. We chose first quarter of 1999 because it was the oldest set. This gives it a larger proportion of loans that are already settled as compared to other sets which will have more loans which are still being paid off.

4 Net Present Value

The ability to spend money has value. A dollar received today is worth less than a dollar that will not be received for a year. The dollar received today possesses the option to be spent any time during the next year, which the delayed dollar does not. This implies that part of money's value is tied to the ability to spend it across time. This concept is called the time value of money and the driving force behind a Net Present Value(NPV). The NPV is a tool to compare assets that have cash inflows and outflows at different points in time. In general, to compute a NPV, take the sum of all the financial inflows and outflows associated with an asset and adjust them to the present time. To express this mathematically, we let R_t represent a cash flow at time t. If money is received R_t is positive, and if an amount is paid, R_t will be negative. $(1 + i)$ represents amount an investment should appreciate to compensate you for the time value of your money for one unit of time t. All these principals yield the following formula.

$$NPV_{total} = \sum_{j=1}^{n} \frac{R_t}{(1+i)^t}$$

Because you include all the cash flows associated with an asset, and bring them all to time 0, NPV_{total} represents what an asset is worth at time 0. It is also worth noting that we assume that if the bank did not invest in this loan, they would invest in a 30 year bond instead, as the most comparable financial asset. Because of this, we set i to be the monthly LIBOR rate from Q1 of 1999. Since the LIBOR rate was 2.93% yearly, our monthly rate came out to around .241%.

4.1 Our Analysis

To calculate the NPVs of these loans, we first need to determine whether or not they ended as paid off or with a foreclosure. We began by associating each loan with its corresponding performance entries. The code uses the final value of the Zero Balance Code for each loan to determine how it ended.These value codes and their meanings can be found in the data set user guide[2]. If the Zero Balance Code is not present or NA, it is assumed that the loan is still current and assigns the value "Current". If the Zero Balance Code is 1, that means that the loan is prepaid, and

[2]Data set guide: http://www.freddiemac.com/research/pdf/user_guide.pdf

it is marked “Prepaid”. Finally, if the Zero Balance Code is 3 or 9, the loan ended in foreclosure and it is marked “Default”. Once the loan has been classified, it gets put into one of two NPV calculation formulas.

4.2 Prepaid and Current Loans

If a loan is either current or prepaid the NPV is calculated with the same function. Due to the fact that many people do not make level payments on their loans, we use the performance data to compute each payment made separately. The theory behind this is that each payment has two parts; one part of the payment compensates the bank for the time value of holding the borrower's outstanding balance, and the other part pays down the outstanding balance. This can be referred to as the interest and principal portions of the payment, respectively. Additionally, since this is the only cash flow for the bank at this time, it is represented by R_t in our NPV calculations. So:

$$R_t = Payment_t = Interest_t + Principal_t$$

We define the outstanding principal, or Current Unpaid Balance, at time t as $CUPB_t$. Additionally, since the interest rate r is quoted as a yearly percent in our data set (3 for 3% instead of .03) we have to divide by 1200 to determine the monthly interest rate charged by the bank. It follows that the interest owed in a given monthly payment is the previous CUPB multiplied by the monthly interest rate. Additionally, by checking the difference in CUPB we find the amount the principal was paid down. Mathematically:

$$Interest_t = CUPB_{t-1} * (r/1200)$$

$$Principal_t = CUPB_{t-1} - CUPB_t$$

Therefore:

$$R_t = CUPB_{t-1} * (r/1200) + CUPB_{t-1} - CUPB_t$$

$$NPV_{payments} = \sum_{t=1}^{n} \frac{CUPB_{t-1} * (r/1200) + CUPB_{t-1} - CUPB_t}{(1+i)^t}$$

However, to get the total NPV of the loan we need to subtract the original amount lent or Original Unpaid Balance (OUPB). This does not need to be adjusted for time as it was lent at time of origination, yielding us a final:

$$NPVtotal = NPV_{payments} - OUPB$$

Our R code reflects this formula for NPV calculation.

4.3 Default

In the event that a loan ended in foreclosure, we needed a different NPV formula. To find the NPV here, we take a similar approach as in the previous case with one major difference. A defaulted loan has a remaining outstanding balance that was not paid off at the end of their loan. This balance has to be adjusted using all the expenses associated with foreclosing on a home and the net proceeds received by selling the home.

$$NPV_{total} = \sum_{t=1}^{n} \frac{CUPB_{t-1} * (r/1200) + CUPB_{t-1} - CUPB_t}{(1+i)^t} - OUPB + \frac{CUPB_T + AL}{(1+i)^T}$$

In this formula OUPB represents the original unpaid balance, or original loan amount. $CUPB_T$ is the current unpaid balance at time of account closure. AL is a figure called Actual Loss, which is given in the dataset. AL represents any amount of the unpaid balance Freddie Mac could not recuperate from foreclosure. Since it is listed as a negative number, the cash flow the bank receives at time of account closure would by CUPB + AL. Since this occurs many years into the mortgage, it is important to adjust it back to time of origination. In this case we assume account closure happens T months after origination. In our code we find this by taking the date of the last payment and adding the number of months it took until account closure. This is a simple task, as both of those terms are given in the performance file. This gives us the total number of months between origination and foreclosure, or T.

4.4 Application

Due to the fact that this data is comprised of two files, it was imperative to find a way to match the performance data for each loan to its respective origination data before we could compute each loan's NPV. This was challenging because the number of performance entries for each origination entry is variable. Additionally, due to the size of our data set we had to solve this problem in an efficient manner. Our first solution that worked utilized a for loop and took roughly 30 minutes to match performance data to 1000 origination entries. Once we switched to an sapply() method, the time was cut to around 10 minutes. Finally, by using matrix operations we could match 1000 origination entries to their performance counterparts in around 6 seconds. We determined that this was fast enough to precess the full dataset in a reasonable amount of time. The R code for this can be found in my Github Repository[3], but an outline of the process is as follows:

[3] GitHub Repository: https://github.com/tbillman/Wang499

Step 1: Read in the datasets	Read both with read.delim()
Step 2: Look for when the Sequence Number Changes	Subtract each sequence number from the previous entry
Step 3: Determine which have differences	Isolate nonzero entries
Step 4: Partition performance data into sets by origination file	Use lapply() with our list of sequence number changes

Once we had all the performance data for each origination entry, we can calculate the NPV and attach it to the origination entry for our data set. This is what will be used for analysis. Finally, once we had our loans classified and developed a formula to compute the NPV in either case, we computed them all. It is also worth noting that we did not compute NPVs of loans that only have one performance file or were marked as repurchased prior to property disposition. These were all minor cases, and not useful for prediction. We took the NPV and added them as another column to the origination file. This was the file used for our regression and classification.

5 Geographic Mapping

Given that our data had the first three numbers of each loan's zip code, we decided to look at how these NPVs look across the country. To do this, we used R packages such as ggplot2, evaluate, mapproj, fiftystater, zipcode, ggmap, and tidyverse. Our code followed this process:

Step 1: Load libraries	library("ggplot2"), etc.
Step 2: Read data	read_csv("File Location")
Step 3: Find representative Zip Code for all leading 3 digits of Zip Codes in our dataset	00200 → 00210, 00500 → 00501 ... 99800 → 99801,99900 → 99901
Step 4: Find representative states for all leading 3 digits of Zip Codes in our dataset	00200 → NH, 00500 → NY ... 99800 → AK, 99900 → AK
Step 5: Match each entry's Zip Code to it's respective state with our data frame	Entry 1 has Zip Code 19300, is in PA, and has NPV $-12,008.92
Step 6: Compute the mean NPV for each state	AK → $19,284.79, AL → $18,721.94 ...
Step 7: Compute the standard deviation of NPV for each state	AK → $13,673.22, AL → $15,694.98 ...
Step 8: Compute the ratio of mean of NPV and standard deviation of NPV for each state	AK → 1.4104, AL → 1.1927 ...
Step 9: Graph data with ggplot2()	Figures 1, 2 and 3

The three graphs we plotted were the average NPV (Figure 1), standard deviation of NPVs (Figure 2), and the ratio between the two (Figure 3). Figure 3 is useful for banks looking for the best risk adjusted loan opportunities.

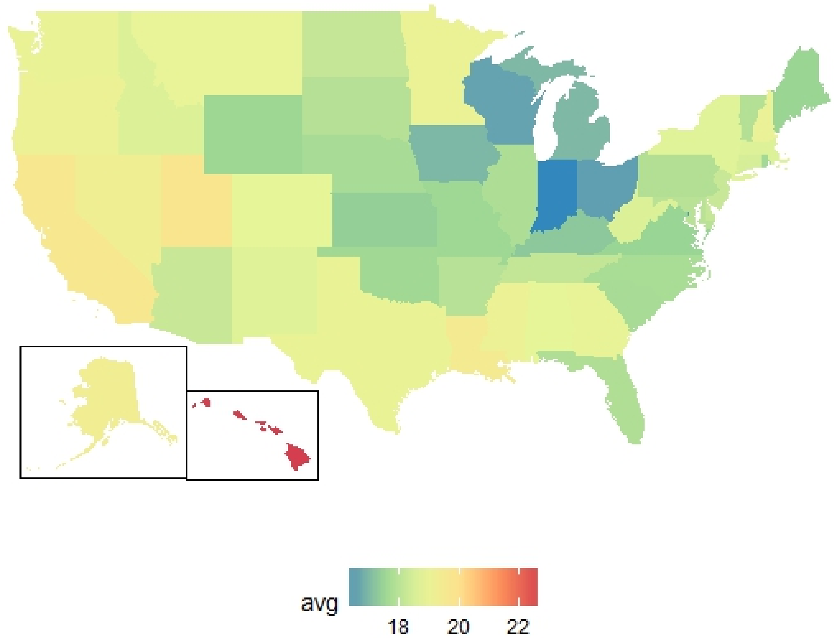


Figure 1: Average NPV by state

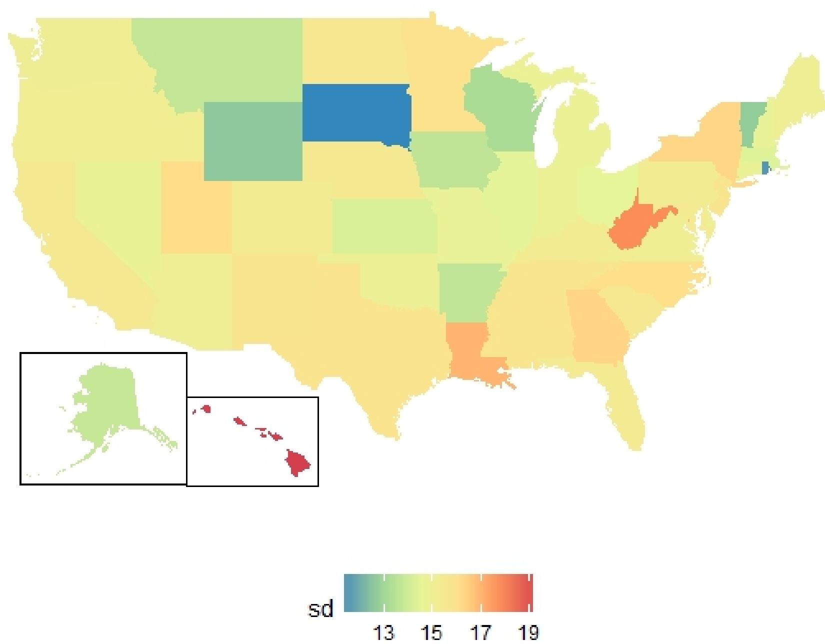


Figure 2: NPV standard deviation by state

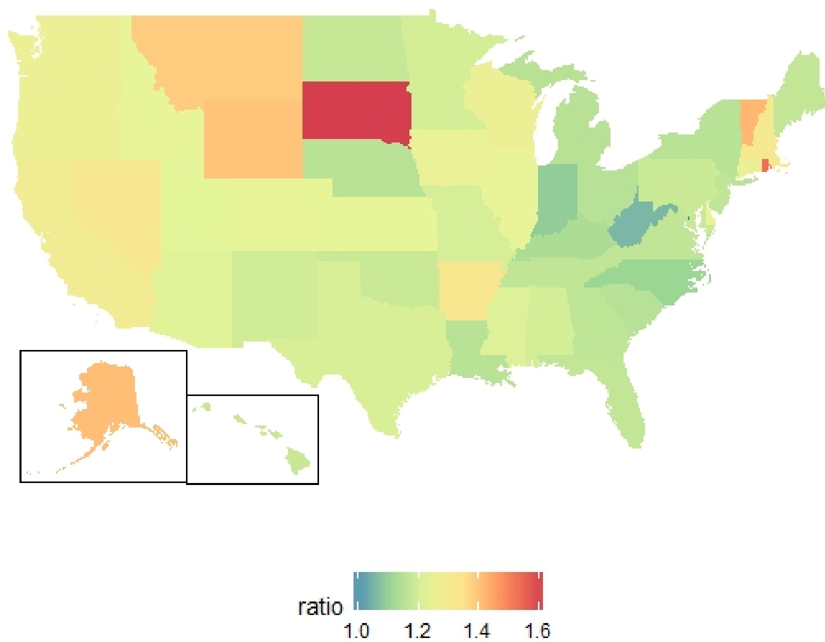


Figure 3: $\frac{Average NPV}{NPV standard deviation}$ by state

6 High Performance Computing

Due to the large size of this data set and the computational requirements of the regression and classification methods we are implementing, access to the Stampede2 supercomputer greatly sped up our ability to run this analysis. From the Stampede2 User Guide:

> Stampede2, generously funded by the National Science Foundation (NSF) through award ACI-1134872, is the flagship supercomputer at the Texas Advanced Computing Center (TACC), University of Texas at Austin. It entered full production in the Fall 2017 as an 18-petaflop national resource that builds on the successes of the original Stampede system it replaces. The first phase of the Stampede2 rollout featured the second generation of processors based on Intel's Many Integrated Core (MIC) architecture. Stampede2's 4,200 Knights Landing (KNL) nodes

represent a radical break with the first-generation Knights Corner (KNC) MIC coprocessor. Unlike the legacy KNC, a Stampede2 KNL is not a coprocessor: each 68-core KNL is a stand-alone, self-booting processor that is the sole processor in its node. Phase 2 added to Stampede2 a total of 1,736 Intel Xeon Skylake (SKX) nodes.[7]

Due to the high computing power and memory of the KNL nodes, we could run RF and RGLM analysis on our full data set in only three hours.

7 Linear Regression

7.1 Initial Data Cleaning

Once we had our NPVs calculated, we began with simple linear regression to see if there was much correlation between the origination data and NPVs. After removing trivial columns with only one unique value and only keeping rows that had information in all columns, we were left with a 178,058 x 25 matrix. Figure 4 shows the initial distribution of the NPVs. After running an initial Linear Regression on this data, we obtained an R_a^2 value of .008039, which is very low. Given the strong clustering in the middle, we believed that there were some outliers on both ends of the curve and that systematically removing these would boost the predictive power of our regression.

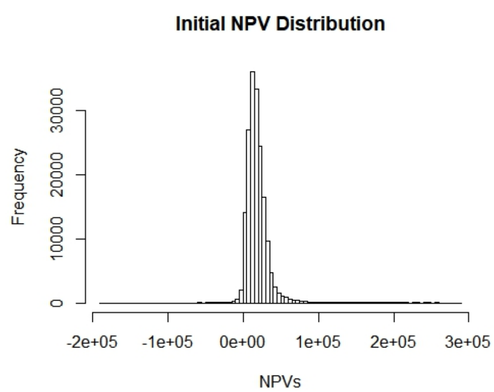


Figure 4: NPVs of complete cases

7.2 Cook's Distance

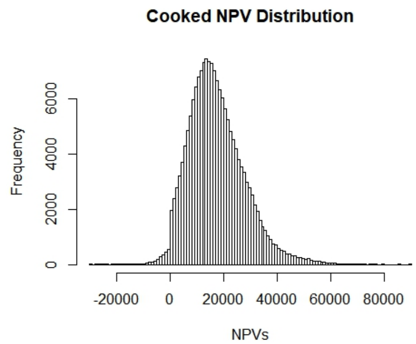


Figure 5: Entries without outlying covariates

In order to remedy our outlier problem, we calculated the Cook's Distance of each point. "Cook's distance, denoted by D_i is an aggregate influence measure, showing the effect off the ith case on all n fitted values":

$$D_i = \frac{\sum_{j=1}^{n}(\hat{Y_j} - \hat{Y_{j(i)}})^2}{pMSE}$$

Where $\hat{Y_j}$ denotes the predicted value of the jth observation and $\hat{Y_j(i)}$ is the predicted value of the jth observation when the ith observation is removed. Additionally, p represents the number of covariate predictors in our linear regression model, and MSE is the mean squared error of our model[2]. This is useful because our regression is aimed at predicting mortgage values of typical loans, and outlying loans can be considered on a case by case basis. A general rule of thumb is to discard points with distance greater than $\frac{4}{n}$, where n is the number of data points[3]. After points with outlying values were removed our distribution can be seen in Figure 5. It is also worth noting that in this distribution there is a large spike right around where NPV = 0. This is because there is a large number of defaulted mortgages with Actual Loss = 0. This is due to financial regulations where if the bank can recover more than their CUPB and foreclosure costs the remaining proceeds go to the borrower. This leaves many defaults that would have a positive NPV just above 0. After rerunning another linear regression, our R_a^2 value jumped to .01711. However, this is still very low, so we looked to other tactics to improve predictive power.

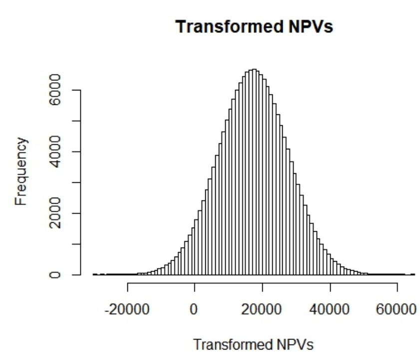
7.3 Multi-collinearity

We also checked for multi-collinearity between our different predictors. "A formal method of detecting the presence of multicollinearity that is widely accepted is the use of variance inflation factors [VIFs]. These factors measure how much the variances of the estimated regression coefficients are inflated as compared to when the predictor variables are not linearly related" [2]. This can be expressed quantitatively as follows:

$$(VIF)_k = (1 - R_k^2)^{-1}$$

Where R_k^2 is the R^2 value predicting the covariate k using the remaining covariates. Common tolerance limits for VIF are 10,100, and 1,000 ([2, p. 408-410]). A low VIF indicates low multicollinearity, whereas a high value indicates that certain variables are not important in our linear regression. When computing the VIF for all of our variables, all values were below 3 with the exception of Combined Loan to Value Ratio (CLTV) and Loan to Value Ratio (LTV). These variables had VIFs of over 2000. This is because CLTV is only different than LTV if someone refinances their mortgage, which is rare. Since LTV is more useful for loan origination, we kept that variable and removed CLTV from the regression. After removing CLTV, the VIF of LTV dropped below 3, but our R_a^2 value remained similar. This indicates that multicollinearity was not having a large effect on suppressing our R_a^2 value.

7.4 Data Transformation



The final tactic we tried was transforming the data into a normally distributed set. This was to test if the non-normality of our dataset was having an effect on suppressing our predic-

Figure 6: An example decision tree for income prediction

tive power. To do this, we found the percentile of each NPV value, and mapped it to a normal distribution with the same mean and standard deviation as our dataset. To do this we used the ecdf() function in R. However, even after this our R_a^2 value only rose to .01812. After this we concluded that simple linear regression would not have strong enough predictive power on this dataset to be useful.

8 Random Forest

Random forest is a contemporary machine learning technique developed by Dr. Leo Breiman at University of California at Berkeley in 2001[1]. Fundamentally, RF takes random subsets of the data and uses them to train decision trees. An example of a decision tree can be found in Figure 6. By training many of these trees on subsets of the data and taking an average of their predictions, we develop a more robust prediction model. This method can be used to predict either continuous or categorical variables. When deciding how to analyze this dataset with RF, we had a few options. Our first attempt was using the dataset to predict NPVs as a continuous

Step 1: Remove Degenerate Columns	Columns like Product Type, which only contain one value are not useful for analysis
Step 2: Remove Incomplete Rows	Rows with missing values are discarded, as the machine learning models do not deal with them well
Step 3: Remove Cook's Distance outliers	Any entries with Cook's Distance greater than 4/n are removed
Step 4: Remove Variables not useful for regression	Variables such as Loan Sequence Number and Borrower Number are removed
Step 5: Partition data into 5 sets	Randomly assign a number 1-5 to each entry form a uniform distribution
Step 6: Build model on each set of 4 indexes	This gives up 5 different models of either RF or RGLM for either regression or classification, each built on approximately 80% of the data
Step 7: Test each model on the remaining index	Use the model to try and predict NPV of test data with test origination data
Step 8: Verify and report accuracy	For regression, we report R_a^2, for classification we report proportion of observations that were correctly classified

Figure 7: Machine learning pseudo-code

variable via regression model. Another option was partitioning the NPV data into categories (Negative, Low, Medium, High) and predicting which NPV category a particular loan would fall into.

8.1 Regression

When running regression we recognized the importance of cross validation. This is a strategy of partitioning the data into a training set and testing set multiple times. For each partition you build a model using 80% of the data, then test it on the remaining 20% and check its accuracy. This is useful to prevent over-fitting in our model. We randomly assigned each entry to a number 1 through 5. Each entry consisted of the origination data of a loan as well as its NPV. We built five models, each with one index as our test data and the remaining four as our training data. This is a technique we used for all our machine learning analysis. Our R_a^2 values were all between .064 and .0661 with an average of .065. While this is significantly better than the linear regression, this still very poor predictive power. An outline of our code's process can be found in Figure 7.

8.2 Classification

Due to the low predictive power of regression, we also used RF to predict the NPV of a loan in a more general sense. To do this, we split the data into the following four categories:

- Below \$0
- $\$0 - \$10,000$
- $\$10,000 - \$30,000$
- Above $\$30,000$

Once we did this, we used RF to predict which NPV category a given loan would fall into, with only the origination data as predictors. If the model had no predictive power, we would still expect a correct guess 25% of the time by chance. After running our classification, our results had a minimum of .614, maximum of .625, and average of .620. This is significantly better than a blind guess, and a notable result.

9 Random Generalized Linear Model

Another technique we thought would be useful was using Random Generalized Linear Models. The way RGLM works is very similar to RF, however instead of training decision trees, RGLM trains generalized linear models. This is very nice because it takes the ensembling aspect of RF with a model that is easier to interpret.

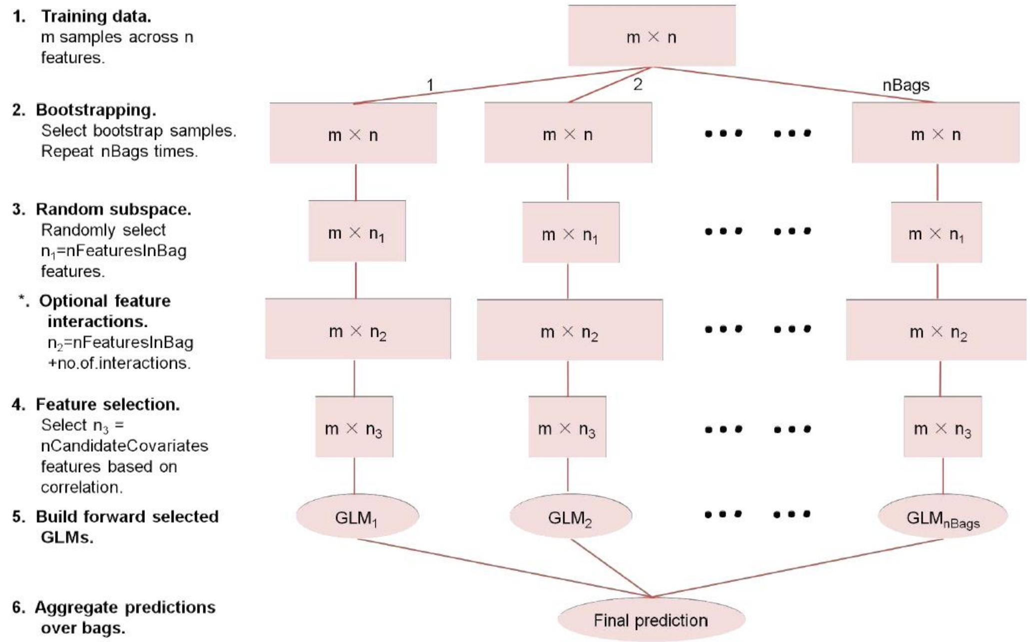


Figure 8: An overview of how RGLM works [6]

9.1 Regression

One issue we encountered with RGLM is that our code could not use factor variables for regression prediction. As such, the R_a^2 value for our RGLM regression suffered. After running our analysis our five values ranged from .0174 to .0194 with an average of .0187. This is around the same as our initial linear regressions, and does not have strong predictive power.

9.2 Classification

Due to the binary nature of GLM classification, to classify NPV with RGLM, we decided to opt for classification into positive and negative NPV. After running our analysis in a similar way to RF, our cross validated results had a minimum accuracy of , maximum accuracy of, and average of .

10 Conclusion

While the ability to predict mortgage loan NPV with the origination data was not as strong as we had initially thought, we did still prove that methods such as RF and RGLM outperform simple linear models. In the future, collating other quarters or data into the models as well as microeconomic and macroeconomic indicators may also help boost predictive power.

REFERENCES

[1] Breiman, Leo. ”Random forests.” *Machine learning* 45.1 (2001): 5-32.

[2] Kutner, Michael H. *Applied linear regression models. –4th ed.* Michael H. Kutner, Christopher J. Nachtsheim, John Neter.

[3] Bollen, Kenneth A.; Jackman, Robert W. (1990). Fox, John; Long, J. Scott, eds. *Regression Diagnostics: An Expository Treatment of Outliers and Influential Cases. Modern Methods of Data Analysis.* Newbury Park, CA: Sage. pp. 25791. ISBN 0-8039-3366-5.

[4] Deng, Grace. ”Analyzing the Risk of Mortgage Default” (2016)

[5] Sirignano, Justin. Sadhwani, Apaar. Giesecke, Kay. ”Deep Learning for Mortgage Risk” (2015)

[6] Song L, Langfelder P, Horvath S. (2013) Random generalized linear model: a highly accurate and interpretable ensemble predictor. BMC Bioinformatics 14:5 PMID: 23323760 DOI: 10.1186/1471-2105-14-5.

[7] Stampede 2 User Guide. (2018) https://portal.tacc.utexas.edu/user-guides/stampede2

[8] Mortgage Basics. https://www.knowyouroptions.com/buy/buying-process/qualify-for-a-mortgage/mortgage-basics

[9] A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22.

[10] Marvin N. Wright, Andreas Ziegler (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. Journal of Statistical Software, 77(1), 1-17. doi:10.18637/jss.v077.i01

[11] Hadley Wickham (2017). tidyverse: Easily Install and Load the 'Tidyverse'. R package version 1.2.1. https://CRAN.R-project.org/package=tidyverse

[12] Doug McIlroy. Packaged for R by Ray Brownrigg, Thomas P Minka and transition to Plan 9 codebase by Roger Bivand. (2017). mapproj: Map Projections. R package version 1.2-5. https://CRAN.R-project.org/package=mapproj

[13] William Murphy (2016). fiftystater: Map Data to Visualize the Fifty U.S. States with Alaska and Hawaii Insets. R package version 1.0.1. https://CRAN.R-project.org/package=fiftystater

[14] Jeffrey Breen (2012). zipcode: U.S. ZIP Code database for geocoding. R package version 1.0. https://CRAN.R-project.org/package=zipcode

[15] D. Kahle and H. Wickham. ggmap: Spatial Visualization with ggplot2. The R Journal, 5(1), 144-161. URL http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf

ACKNOWLEDGMENTS

I would like to thank Dr. Yishi Wang for advising me throughout this project. Whether it was our weekly meetings, emails, or online meetings at either 8:30 PM or 9:00 AM, Dr. Wang has been unbelievably helpful this whole process.

I'd like to thank Dr. Ann Stapleton for facilitating my learning of R, Github, UNIX, among so many other technical skills. Dr. Stapleton also help me learn other best practices for large software based projects. Without already having learned these skills working for Dr. Stapleton I would not have been able to widen the scope of this project to what it is.

I'd like to thank my family and friends for supporting me through this project. Without you guys, I don't know if I could have done it.

I'd like to thank the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper. URL: http://www.tacc.utexas.edu