Predicting Mortgage Backed Securities Prepayment Using Machine Learning Methods

Authors:
Chong Guo (chguo93), John Boccio (jboccio), Christopher Cameron (cpcam)

1 Introduction

A residential mortgage backed security (MBS), is a fixed income security, and is one of the largest asset classes in the financial market. These securities are sold to investors. As the borrowers gradually pay off the underlying mortgage loans, the investors receive payments of interest and principal. A large risk factor in MBS lies in the possibility of prepayments. Prepayments are payments by borrowers, who pay back a part, or the full amount of the loan earlier than discussed in their residential mortgage contract. In this case, investors will stop receiving interest payments as originally scheduled resulting in decreased returns on MBS. Since prepayments can severely decrease the profitability of an MBS investment, it is one of the most important topics in the MBS world.

A typical way to study and predict mortgage prepayments in the industry is by empirical research and statistical modeling. Researchers use a number of variables to predict mortgage prepayment risk such as, borrower credit score, income, loan-to-value ratio, and loan age. Then a pre-specified form of model uses these factors to predict prepayment risk. Extensive data study and comprehensive industry knowledge are required to identify useful factors and suitable model frameworks. In this project, we aim to find an alternative method to predict prepayment risk of residential mortgage loans by exploring machine learning techniques.

2 Related Work

Applying machine learning (ML) techniques to loan related problems has been done before. Though they may not have necessarily been applied to loan prepayment specifically. In a work by Han [1], various ML techniques are applied to repayment for loans. In this work Han uses various methods such as: Random Forest, Support Vector Machines (SVM) and logistic regression to estimate the accuracy of each model. In the end, Han finds that logistic regression is the most optimal predictor for loan repayment with Fico Score and annual income being the greatest predictors for repayment. Works by Zhang [2] and Holm [3] focus primarily on logistic regression. Zhang looks at the probabilities and not just a classification of default or not. The results show that their use of "survival analysis" proves to be a formidable foe to logistic regression. In fact, when exploring probabilities, their survival analysis technique edges out logistic regression. In Holm's work, they found that the sum of assets and how long an individual has been with a bank to be driving factors as to whether an individual defaults or not. What's interesting here is the introduction of how long an individual has been with a bank. That may be a useful indicator for future work involving machine learning mortgage models.

There are some works that focus primarily on prepayment. They provide a closer framework for how we can approach our research problem. In a work by Amar [4] they use neural networks to consider prepayment or not. They conduct a sensitivity analysis to better understand the results, and break down prepayment into two types. This paper highlights the usefulness of neural networks when determining prepayment. The other example comes from our own backyard by Drembles[5]. His work projects not just prepayment, but also default on the loans. This paper uses a LSTM architecture and trains over a 20 month data period. They found that adding more layers results in more accuracy, but it can cause overfitting. This paper introduces the idea that prepayment can be predicted using deep neural networks.

3 Dataset and Features

Our primary dataset is Freddie Mac's Single Family Loan-Level Dataset. This dataset contains approximately 45.5 million mortgages originated between January 1, 1999 and September 30th, 2020. The data set mainly consists of fixed-rate single family mortgage loans with a maturity of thirty years. The dataset consists of two pieces: origination data and performance data. The origination data contains static information acquired at the time of loan origination, credit score, debt-to-income ratio, and loan-to-value ratio are a few examples. The performance data is reported monthly and includes dynamic information like current balance, loan age, and if the mortgage is prepaid in the current month. Each loan is tracked from origination until loan maturity or termination with a cut-off date of September 30th, 2020.

Merging the two pieces of data by loan id, we formed a dataset where each row contains a loan's origination information and performance in a certain month. In addition, we added several macroeconomic indicators to supplement the Freddie Mac dataset, including mortgage rate, housing price appreciation and unemployment rate. These factors are downloaded from Freddie Mac, Federal Housing Finance Agency and US Bureau of Labor Statistics. Joining all the datasets together by reporting data forms our final dataset. We then random sampled 100 loan series from each origination period and formed a final dataset of more than 2 million loan records.

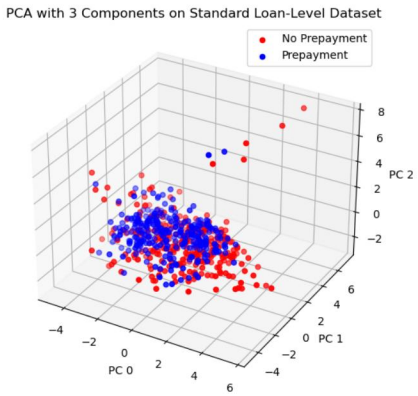

Figure 1: PCA result

Categorical variables are transformed with one-hot encoding and missing data is processed based on the underlying financial meaning. After data preprocessing, the dataset consists of 95 features with a 60%, 20%, 20%, train, validation, test data split. PCA is performed to visualize the dataset. As shown in Figure 1, prepaid observations are on top of the not-prepaid observations to some extent, indicating possible classifications with appropriate algorithms.

4 Methods

4.1 Logistic Regression

Logistic regression is one of the most commonly used machine learning models for classification problems. It uses sigmoid function to transform dependent variable and assumes linear relationship between independent variables and the transformed dependent variable. Thus it results in a linear classifier. Specifically,

$$P(y = 1; x, \theta) = h_x(x) = \frac{1}{1 + \exp^{-\theta^T x}}$$

In our case, $y = 1$ represents the observation prepaid. The loss function is

$$l(\theta) = \log L(\theta) = \sum_{i=1}^{n} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

In addition, we added regularization on top of the above formula to avoid overfitting. Both L1 and L2 regularization are attempted. With L1-regularization, the model tends to shrink coefficients towards 0 by penalizing the logistic regression model with the term $\lambda|\theta|$. This helps us to reduce dimension and identify important features. The L2-regularized model penalizes large coefficients and mitigates multicolinearity.

4.2 Support-Vector Machines

Support-vector machines operate by finding the hyper-plane that maximizes the separation between the classes. Typically this is a linear hyper-plane but lots of times the data is not linearly separable so a non-linear transformation must be performed, often into higher dimensions. The main issue that arises from this is that sometimes the dimensions increase so much (d » n) that it becomes computationally infeasible. This is where the "kernel trick" comes into play. With the kernel trick, we don't need to represent the underlying feature map explicitly but instead do so through a linear combination of the training examples and the kernel. This allows us to avoid storing estimators explicitly and significantly improves runtime efficiency at the cost of needing all the training data when performing inference. For the purposes of our project, the hyper-planes will separate the "No Prepayment" and "Prepayment" classes and we will do so by utilizing the kernel trick.

4.3 Gaussian Discriminant Analysis

Gaussian Discriminant Analysis works by creating Gaussian distributions for each class with the given training data. After these distributions have been created, a new data point is classified by evaluating the probability that it belongs to each of the classes based on it's respective Gaussian distribution. The distribution that we will predict that the new data point belongs to will be the one with highest probability of belonging to that Gaussian distribution. GDA has two main forms, linear and quadratic. When you assume that all the classes have the same covariance matrix, the decision boundary between the classes will be linear. A quadratic decision boundary between the classes will be created when each of the classes can have different covariance matrices. We employed both methods and compared results with logistic regression. When using GDA to predict prepayments, we will create two Gaussian distributions, one representing the distribution of the data for "No Prepayment" and one for "Prepayment".

4.4 Feed-Forward Neural Networks

Neural networks work by stacking and layering many neurons with activation functions together to create a highly non-linear function. Simply put, a single neuron will output the inner product between its inputs and its weights and add a bias to it, this output is then passed through an activation function and passed onto many other neurons. We train these neural networks very similarly to how other supervised methods are trained, we calculate the gradient of the loss function with respect to each of the parameters and then perform stochastic gradient decent. For the purposes of predicting if a prepayment occurs in a given month, we will use our dataset to train a neural network which will output the probability that a prepayment occurs.

5 Experiments and Results

5.1 Logistic Regression

We first built a logistic regression model with all the prepay relevant features, including original credit score, debt-to-equity ratio (DTI), and loan-to-value ratios (LTV). The baseline model predicted "prepayment" correctly 99.998% of the time and "no prepayment" correctly 95.98% of the time based on the test data. While the model gives reasonable accuracy on the test data, it uses all the 95 features. Next, we added a L1-regularization term to the baseline model to reduce the number of dimensions. This model yields similar accuracy and eliminated number of feature by 10%. With L2-regularization, we maxed out at an astounding 99.92% accuracy and were able to predict "No Prepayment" with 100% accuracy. Figure 2 lists the top 10 important features identified by L1-regularized logistic model. Importance is calculated as $abs(coefficient) * mean(feature)$. From this, you can see that by far the most important feature that logistic regression was looking at was "current_UPB". This refers to the current balance of the loan. The less balance a loan remains, the more likely the borrower will prepay. This finding is consistent with the work by Amar [4]. Figure 4, 5 and 6 show confusion matrix for each logistic model.

5.2 Support-Vector Machine

The SVM models that we used utilized the "kernel trick" with the RBF, sigmoid, and polynomial kernels. The main downside of using kernels for our application is that we have an abundant amount of data but we will not be able to utilize it all as our kernel matrix will grow exponentially for each extra data point we include. Due to this, we only used a very small subset of our dataset and this was reflected in the relatively poor results that these models achieved.

	coef	average	variance	importance
current_UPB	-0.0057	170552.8111	10924400289.6884	974.8498
months_to_maturity	-0.0084	320.4819	1540.1577	2.6982
orig_CLTV	-0.0194	76.0475	402.0275	1.4763
mtgrate	0.3112	4.6877	1.364	1.4588
orig_interest_rate	-0.2189	5.3984	1.4182	1.1818
occupancy_status_b'P'	1.1331	0.8895	0.0983	1.0079
current_interest_rate	0.1451	5.3607	1.4587	0.7776
loan_age	-0.017	42.0115	1426.5692	0.7158
loan_purpose_b'P'	1.7039	0.3849	0.2368	0.6559
credit_score	0.0007	746.846	137725.4752	0.5127

Figure 2: Top Features

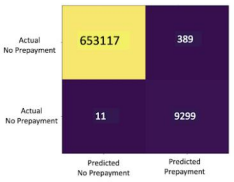

Figure 3: Base Logistic

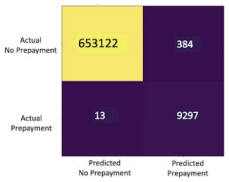

Figure 4: Regularized Logistic

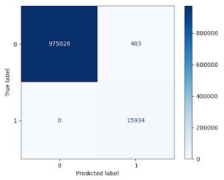

Figure 5: Final Logistic

5.3 Gaussian Discriminant Analysis

Gaussian Discriminant Analysis was our favorite algorithm to use since there were no hyper-parameters to tune, it ran quickly, and gave us the best overall accuracy. When creating a GDA model with a quadratic decision boundary (different covariance matrices per class), we achieved a 99.96% accuracy. Having the different covariance matrices per class was very important as we did not see this level of accuracy when performing GDA with a linear decision boundary (same covariance matrix for each class). Our 3D PCA graph did a good job of giving us a hint at using GDA since the data showed some natural clustering.

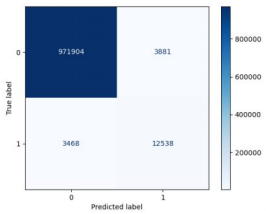

Figure 6: Linear GDA

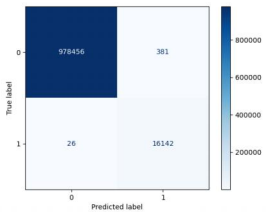

Figure 7: Quadratic GDA

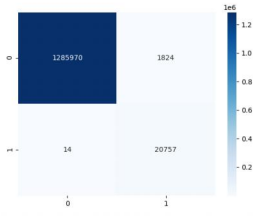

Figure 8: Neural Network

5.4 Feed-Forward Neural Network

Since we are only aiming to predict whether or not a prepayment occurs, our neural network needs to be able to perform binary classification. To achieve this, we created a neural network that only has 1 output neuron. The value of this neuron will represent the probability that a prepayment will occur. In order to restrict the value to a number between $[0, 1]$ so that it is a valid a probability, the final output of this neuron will be passed through a sigmoid activation function. With this output in

Model Performance on Test Set (1.3 million data points)					
Model	True No Prepayment	False No Prepayment	True Prepayment	False Prepayment	Overall Accuracy
SVM (Poly)	98.866%	1.134%	96.532%	3.468%	98.854%
SVM (RBF)	99.668%	0.332%	93.926%	6.074%	99.588%
SVM (Sigmoid)	99.448%	0.552%	75.208%	24.792%	99.098%
L2 Logistic	**100.0%**	**0.0%**	97.058%	2.942%	99.951%
Neural Network	99.999%	0.001%	91.922%	8.078%	99.859%
GDA (Quadratic)	99.961%	0.038%	**99.839%**	**0.160%**	**99.959%**
GDA (Linear)	99.644%	0.355%	76.362%	23.637%	99.259%

Table 1: Results From Confusion Matrices

place, we were then able to use the binary cross entropy loss function to evaluate and train our neural network against our dataset.

Unfortunately, there is not that much theory out there for creating the best neural network architecture for your specific problem, so we basically arrived at our architecture through trial and error. The final architecture that we arrived at consisted of 4 linear layers which had 95, 128, 256 and 1 neuron each. Between layers 1-2 and 2-3, we inserted batch normalization and a ReLU activation function. Between layers 3-4 was batch normalization and a sigmoid activation function and the final neuron's output in the 4th layer was passed through a sigmoid function to create the prepayment prediction. To train the neural network, we used stochastic gradient decent as our optimizer with a learning rate scheduler that started the learning rate at 0.1 and multiplied it by 0.91 after each epoch. These values were chosen so that we ended up at a value of about 0.005 after 30 epochs.

The last hyper-parameter we focused on was the positive weight value associated with the binary cross entropy loss function. This allowed us to create a larger penalty for getting false positives. This was a crucial hyper-parameter for our neural network since our dataset had many more examples of "no prepayment" than "prepayment". When we looked at the distribution of our dataset (about 50:1 ratio of no prepayment to prepayment), we thought that weighing positive examples by 50x would be optimal but this had poor results. This weight was too harsh and caused the neural network to begin to perform worse when no prepayment occurred. What we ended up finding was that a positive weight of 10.0 was optimal.

5.5 Final Results

The main metrics that we were concerned with when evaluating all these models was the overall accuracy, prepayment accuracy, and no prepayment accuracy. Some interesting results to note are that GDA with a quadratic decision boundary performed very well overall and did very well at predicting when prepayments would occur, which other models seemed to struggle more with. Also, logistic regression was perfect at predicting when no prepayment would occur. An interesting model to try out in the future would be an ensemble model which allows all the models we created to vote on what the final output should be. The conglomeration of all the results we had from our confusion matrices for our models can be seen in table 1.

6 Conclusion

Our work turned a prepayment in MBS into a classification problem at individual level. We asked: will the individual prepay or not? Many institutions use pool-level model to predict prepayment and a lot of loan-level information are missed in the process. Out work keeps loan information as much as possible and results in high accuracy. In addition, this project explores various ML techniques and answer the question: Which technique performs the best for the parameters we explored? We found that most of our models eclipsed the 99% mark. Amongst these top performers logistic regression was the second best, while GDA was the winner. We expected a neural network to be the top performer and maybe it can be. Future work would have to be done to explore more feature engineering for the attempted models and to compare results with pool-level models. This work was a great step in the right direction and future work would compare these models more rigorously with questions that aren't just limited to prepayments.

7 Contributions

Chong Guo: Worked on data generation, random sampling, data cleaning and transformation, logistic regression models and write-up of project for introduction, data description, logistic regression and conclusion sections.

John Boccio: Worked on efficiently parsing the data from the dataset which creates a large pandas dataframe to be passed into ML models, cleaned the data, created SVM models and write-up of SVM methods and results, created neural network model and write-up of neural network methods and results, created GDA models and write-up of GDA methods and results.

Chris Cameron: Worked on the project write-up, data post-processing, and final paper structure.

References

[1] Han et. al *"Loan Repayment Prediction Using Machine Learning Algorithms"*

[2] Zhang, *"Modeling The Probability Of Mortgage Default Via Logistic Regression And Survival Analysis"*

[3] Holm, *"Default Prediction of a Swedish Mortgage Portfolio using Logistic Regression"*

[4] Amar, *"Modeling of Mortgage Loan Prepayment Risk with Machine Learning"*

[5] Drembles, *"Time to Default & Time to Prepayment Estimation Using LSTM and Deep Learning Techniques"*