Mortgage-Backed Securities Prepayment Prediction With Deep Learning

Yu Dong Zhang
lz927@stanford.edu

Abstract

A mortgage loan gives the borrower the right to prepay the full or partial amount before its maturity. It is referred as mortgage prepayment. Mortgage-backed securities (MBS) are financial products structured based on cash flows of these mortgage loans. MBS is now among the largest financial sector in the United States. With this gigantic amount of MBS on banks' balance sheet, managing prepayment is vital. In this project will be exposed aspects of machine learning regression algorithms, and how they are applied in the setting of predicting Mortgage-Backed securities prepayment, specifically, the conditional prepayment rate. Models of lasso linear regression, as baseline model, and neural network are used along with hyperparameter tuning and training techniques to achieve the tasks. The results are that neural network outperforms baseline model and achieves an higher out-of-sample accuracy.

Keywords: MBS, Prepayment, Machine Learning, Neural Network

1 Introduction

Mortgage-backed securities (MBS), the "vicious" origin of 2008 financial crisis, which have severed as a crucial lever for government in managing monetary policies for nearly 40 years, are among the largest financial sector in United States. U.S. mortgage finance market had experienced structural changes after the crisis. Especially during recent couple of years, MBSs are mostly guaranteed by agencies, which lead to huge decrease in credit risk of such product, making them mostly default-free. The investing public had always been trying to solve the world that has sufficient uncertainties to generate excessive economical returns, instead of risk-free entities. This time is no exception. The elimination of credit uncertainties of MBSs has encouraged Wall Street quants to seek profits in the other key factor that drives the value of MBSs, and that is prepayment. Prepayment modelling is among the most complex and novel areas of financial modelling. Its demand has surged significantly only in recent 4 years, and the complexities involve enormous amount of data and factors, as well as the difficulties in model specification and estimation. In this project, I tend to tackle this problem with prepayment modelling. The goal is to model and predict the prepayment risk of agency MBSs, using regression learning algorithms of linear regression as baseline model and multi-layer neural networks. The input to our algorithms are many numerical and categorical variables that drive prepayments. We then use lasso linear regression and neural network to output a predicted constant prepayment rate (CPR), which is a percentage value between -1 and 1.

2 Related Work

The deep neural network has been applied and evaluated in prepayment modelling. In Zhang (2019)[4], neural network is designed and applied to predict conditional prepayment rate of 30-year mortgage and compared with an industry production model. The result is favorable towards neural network that it produces highly accurate results of not only prepayment rates but also nonlinear risk drivers. In a similar fashion, Amar (2020)[3], models prepayment rate on a loan-level instead of a portfolio-level and achieved wonderful results. However, there is no current research that studies prepayment modelling as both a classification problem and a regression problem. This project, on

a general level, not only tends to model prepayment rates as a regression problem, but also aim to predict if a loan will be paid prematurely as a classification problem. In a combination of two results, a higher degree of accuracy should be resulted.

3 Dataset and Features

The raw dataset that is used for this project is obtained from FreddieMac's official website. The dataset on the website is on a yearly basis from 1999 to 2021. It contains information regarding of all the loans that were issued during the period. It has separate file containing loan origination information such as FICO score, loan terms, etc, and in another file, the performance of the loans are tracked as time series data as well. Approximately speaking, there are around 50,000 loans issues each year and each of them survive from a few months to 264 months. Each monthly data of a single loan accounts for one data point. Thus, the raw dataset can be constructed into a dataset of hundreds of millions of data points. However, due to the computational resources that there is, only a sample of the dataset will be used. And additionally, economical factor data is also included, such as market mortgage rate, house price index, and unemployment rates. For this project, I will only be using FreddieMac's 2020 data which can be constructed into a dataset of over 700,000 data points.

3.1 Data Preprocessing

Since the data set is very large, significant work of data preprocessing needs to be done. First, for missing values, since these variables are time-varying, and current month of magnitude does dependent on that of previous month, linear interpolation is used to fill out missing values. Second, for outliers, since there are very few that can be spotted (after calculating their respective ranges), they are removed and they only occupy approximately 0.001% of the data. And also the raw dataset contains variables that are not going to be used which will be discard, and also contains some variables in undesired forms such as datetime and variables that have characters as their values.

3.2 Data Engineering

This section talks about the variables data that are not straight from the data and engineered based on paper in [2].

The raw dataset contains two file as mentioned earlier. One contains loan origination information which gives a single value for a single loan, i, at its origination. The other file contain the performance record of all the loans that are issued previously, in the year 2020, and not terminated. It is a time series data, which gives a single value for a single loan, i, at a single time stamp, t.

- Origination Features : Since origination feature are the same throughout the life of the loan, it is cast to each time series of a loan, for example, credit score stay unchanged for the life of the loan, so for a loan that survives for 12 month, there are 12 data points with the same credit score value. This type of features include from Credit Score to Loan Purpose Purchased Loans in the list in the next section.
- Categorical Features : There are features that are categorical and one-hot encoding is used to reformulate those features. Occupancy type takes three values, region takes 54 values and loan purpose takes 3 values. Notice here regions are divided into categories of South region, North-East region, West region and Mid-West region.
- Month as Feature : the month feature takes on value of 1-12 if we use the same method above with categorical features, there will be 12 new binary feature. Here we use a sine transformation and a cosine transformation of month value, m, to represent the feature.

The single dependent variable CPR is purely engineered. We want to model CPR as a percentage rate representing how much of the loan is prepaid. The CPR is calculated as follow:

$$CPR_t = \frac{PPA_t}{P_{t-1}} \tag{1}$$

where PRA is prepaid amount as follow and P_{t-1} is the previous period outstanding amount,

$$PPA_t = PPA_t^{act} - PPA_{t-1}^{exp} \tag{2}$$

and,

$$PPA_t^{act} = P_{t-1} - P_t \tag{3}$$

where P_t is current outstanding balance,

$$PPA_t^{exp} = \frac{r \cdot P_0}{1-(1+r)^n} - r \cdot P_{t-1} \tag{4}$$

where r is the current mortgage rate, n is the original term of the loan in months, and P_0 is original principal balance.

3.3 Features

The complete list of 21 features are shown below.

Current Actual Unpaid Balance	Numerical
Loan Age	Numerical
Current Interest Rate	Numerical
Estimated Loan to Value	Numerical
Credit Score	Numerical
Debt to Income Ratio	Numerical
Original Loan Size	Numerical
Original Interest	Numerical
Original Loan Term	Numerical
Occupancy Type Owner Occupied Loans	Binary
Occupancy Type Investor Loans	Binary
Occupancy Type Second Home Loans	Binary
South Region Loans	Binary
West Region Loans	Binary
Mid-West Region Loans	Binary
North-East Region Loans	Binary
Loan Purpose Renanced Loans	Binary
Loan Purpose Non Loans	Binary
Loan Purpose Purchased Loans	Binary
Sin Representation of Calendar Month	Numerical
Cos Representation of Calendar Month	Numerical

For feature selection, at this stage, only domain knowledge is used to select features. For example, refinancing rate is traditionally very relevant to prepayment rate where as issuers are not so relevant since average homeowners usually have no knowledge of what the issuers are. The most 21 relevant features are selected to proceed as to gain an insight about this data set. Generally, a neural network does not require complicated feature selection methodology due to the fact that it can form proper nonlinear combination of the attributes based on the intrinsic of the data features when there are enough layers and nodes to work with. But further more thorough feature selection methodology is discussed as in the Future Work section.

4 Methods

4.1 Lasso Linear Regression (Baseline)

I first choose to use the most simplistic regression model to apply to the data set to gain some further information about the relationship between x and y. Lasso linear regression is given below:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - \theta^T x^{(i)})^2 + \gamma||\theta||_1$$

Since we have 21 features at hand, Lasso linear regression is used. By adding this lasso penalty, or L1 norm of the parameters as penalty, it aims to sparse the parameters and prevent over-fitting. In

the case of multi-collinearity, as in the features may have correlations with each other, for example, mortgage rate and housing turnover, Lasso could help with this issue with some θ being zeros. And also, adding the regularization term is also a good practice in terms of finding the optimal bias and variance trade-off. Another reason that linear regression is used as a baseline model is that mortgage features are highly nonlinear and linear regression should significantly underperforms the neural network model.

4.2 Cross Validation for Hyperparameter

[1] K-fold cross validation is applied to determine lasso linear regression hyperparameter γ. $k = 10$ is selected as it is a typical value to use for machine learning practitioners. The data set is randomly divided into 10 subsets, 9 of which serves as training set and the other one serves as validation set. For each of the training set, there is one trained model with a set of associated parameters, θ. For evaluation purpose, the average of the training errors over entire 9 models is computed to serve as the estimated generalization error. The model's γ that produces the smallest generalization error is chosen to perform modelling on the entire training set with hypothesis shown in last section. The data set was further split into 80% for cross-validation aforementioned and the rest 20% is for model testing or error analysis.

4.3 Neural Network

The main method of this project is neural network. Lasso linear regression is only applied as a baseline regression model to gain insight about linearity. The neural network is an algorithm that has an architecture shown as follow:

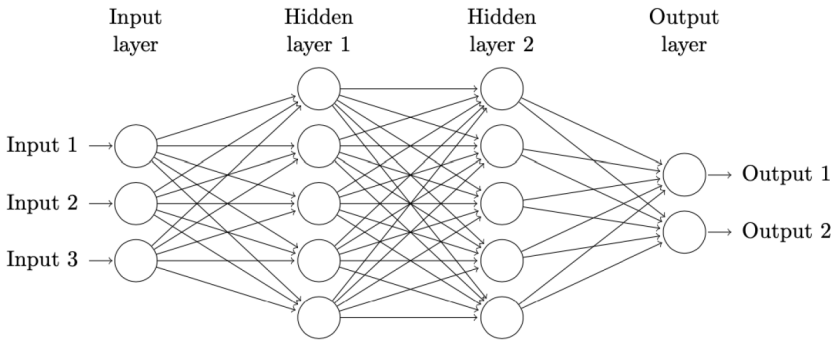

Figure 1: Neural Network Model Example

For a regression problem, the loss function is chosen as

$$C_n(x_n, t_n) = \frac{1}{2}||y_n - t_n||^2$$

The iterative method gradient descent as an optimization algorithms is used here to find the optimal solution. This iterative method requires a differentiable cost function and a starting point of the algorithm. Each iteration takes a step opposite to the direction of the gradient. In the context of feedforward neural networks, the learnable parameters are set to their initial values in the first iteration, and the input is pushed through the network to generate the output, which is then evaluated and compared with the target values by applying the cost function. Backpropagation is applied to calculate the gradient in order to change the weight in the direction of to reach the minimum point.

4.4 Neural Network Model Specification

A neural network which is the core model to this project will take the features selected from above as input and produce conditional prepayment rate as a single numerical output. Below are the specific design regarding the neural network:

- Structural Design. The neural network consists of 1 input layer, 1 output layer, and for hidden layers, the first hidden layer consists of 1024 nodes and the last has 2 nodes while in between the next hidden layer has half of the number of nodes than previous hidden layers. ReLU is used as activation function within hidden layers and sigmoid function is used on the output layer.
- Functional Design. In each hidden layer, 2 sublayers are added. The first sublayer is the dropout layer since there are two many nodes in the first layers and training time could be reduced significantly with dropout sublayer, and also overfitting can be treated. The second sublayer is batch normalization. This sublayer normalizes the output from activation function in order to fasten parameter convergence.
- Algorithm-related Design. ADAM optimizer is used to find the optimal set of weights. Grid search can be used to find the appropriate momentum rate, learning rate and learning rate decay.

With the aforementioned design, conditional prepayment rate can be predicted and also similar error analysis will also be performed and if time allows, model ensemble can also be adopted in the setting.

5 Results and Discussion

5.1 Lasso Linear Regression

R^2 is used as the evaluation metric to see how lasso linear regression performs. Below are the graph of hyperparameter selection based on R^2 scores, a graph of in-sample prediction vs true value, and a table of in-sample and out-of-sample mean absolute error (MAE) and mean squared error (MSE). Note that only in-sample lasso linear graph is plotted here because the graph of the rest testings look similar since the graph is not a good representation here for our case.[2]

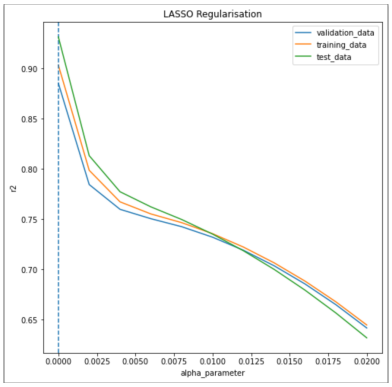

Figure 2: Scoring Based on different Hyperparameter

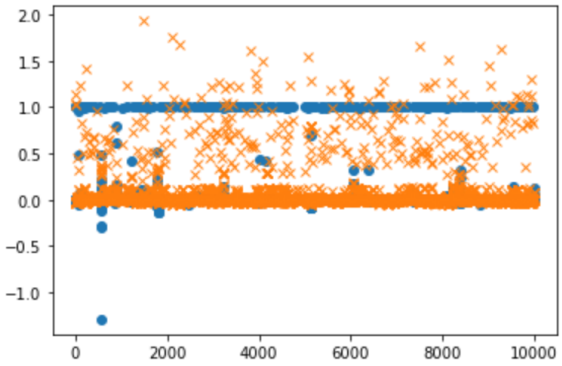

Figure 3: Scoring Based on different Hyperparameter

	in-sample	out-of-sample
MAE	0.0292	26.43
MSE	0.0067	Very Large

The result showing here is that the hyperparameter representing penalty for overfitting is 0. The reason may be that the input features are highly non-linear the objective function of linear regression does not really work in this case. And also, the reason why lasso linear regression performs such badly even out-of-sample, is that many of the risk factors that drive MBS prepayment is highly nonlinear and also interactive. For example, as many studies suggest, prepayment due to housing turnover and prepayment due to refinancing rate have very different dependency on same risk factors. More specifically, refinancing rate increases with loan size whereas housing turnover decreases with loan size, while they are all the factors that feed into lasso linear regression model.

5.2 Neural Network

After grid search for batch size and learning rate, the best hyperparameter that gives the smallest regression loss is that learning rate, 0.1 and best batch size is 256. Here we only use MAE as evaluation metric since MSE for linear regression gave unexpected results.

	in-sample	out-of-sample
MAE	0.00549	0.00648

IN comparison with linear lasso, the neural network model significantly improve the total accuracy for both in-sample and out-of-sample. Thus, we are confident that neural network model would be a very suitable model in the case of prepayment modelling. It further shows that overfitting problems are not presented here since out-of-sample MAE is also small for the predicted CPR.

6 Conclusion

In this project, we aim to model MBS prepayment with two machine learning models, lasso linear regression and neural network model. Lasso linear regression model as a baseline model selects the hyperparameter which is the penalty of overfitting as 0. It further shows that the 21 features are not correlated in any way, thus, the overfitting does not come from multicollinearity of features. With evaluation metrics of mean absolute error and mean squared error, it is observed that lasso linear regression does work for in-sample testing but it performance very poorly for out-of-sample testing. And in terms of magnitude of MAE and MSE, neural network performs much better overall than lasso linear regression. Further, it proofs that the nonlinearity of MBS features.

7 Future Works

As discussed in the previous section, lasso linear regression model is not a good choice for the data set. Thus, A more functionally rich neural network and a more thorough feature selection model are proposed in hope to achieve higher accuracy.

For our case, we have over 20 data features which is way too many, for the ease of computation and training time, feature selection will still be performed.

- Covariance Matrix. In order to tackle the problem with multi-collinearity, the factor covariance matrix is calculated explicitly to check if two factors are highly correlated. If so, only one will be included, and sign will be taken care of if there is negative correlation.
- Information Theory. The information value matrix is also calculated to select features. And only factors with information value greater than 0.02 are included.
- Decision Tree. A decision tree model is built to calculated mean square error and importance scores. The factors with high importance score are included.

With all four steps above, a final set of features can be selected and feed into the neural network.

References

[1] Hyperparameter tuning in lasso and ridge regressions.

[2] Mae, mse, rmse, coefficient of determination, adjusted r squared — which metric is better?

[3] Shlomo Amar. Modeling of mortgage loan prepayment risk with machine learning. 2020.

[4] J. "David" Zhang, X. "Jan" Zhao, J. Zhang, F. Teng, S. Lin, and H. "Henry" Li. Agency mbs prepayment model using neural networks. *The Journal of Structured Finance*, 24(4):17–33, 2019.