A Study of the Factors Influencing Mortgage Serious Delinquency:

with machine learning methods using mortgage data from Fannie Mae

Wujian Xue

September 2024

Commentary in November 2025:

I wrote this article in September 2024. At that time, lots of new techniques in AI such as Large Language Model (LLM), Retrieval Augmented Generation (RAG), or Agent were not popular or even exist.

Standing at November 2025, I would say there are 4 stages of AI application during my nearly 2 decades of careers in finance. The first one is traditional statistics, such as linear regression and the beta in Capital Asset Pricing Model(CAPM). The second stage is machine learning models, such as decision tree and random forest. They are discussed intensively in this article. The third stage is deep learning. Recurrent Neural Networks(RNNs) are ideal for sequential data like text or time series, whereas Convolutional Neural Networks(CNNs) are best for processing spatial data like images. The 4th stage is Generative AI. This is the most popular and advanced one for financial institutions. Not only ChatGPT and Copilot are widely used in banks, but also many domain specific AI tools have been building such as BloombergGPT or JPMorgan's Contract Intelligence (COiN).

In the past 20 years, finance doesn't change that much, but technologies change very much. In the following articles, I will write more advanced AI applications in finance.

The other thing I want to point out is the tools. Although I used R/RStudio in this paper, there are some changes as time goes by. It is true that R/RStudio is very popular for statisticians and data scientists. I have to say that Python is dominate in AI world. Therefore, in my next article, I changed to the tool from R to Python. In addition, there is also a competition between R/Python and SAS. Since R/Python are open-source software, we can run linear regressions by using several packages and hence have slight differences in parameters. These differences are acceptable in school. But when it comes the exact dollar amounts in finance, especially for mortgage with large balances, then differences may not be always acceptable.

1.Introduction

When financial institutions lend to borrowers, they know some borrowers will default. Therefore, the credit analysis in financial institutions focus on the Days Past Due (DPD). Usually, DPD can be divided to 1-30 days, 31-60 days, 61-90 days, 91-120 days. After 120 DPD, then the loan is very likely to default.

In this project, I will discuss the drivers for Serious Delinquency (SDQ), which means a mortgage loan is 90 days or more past due. At this time, financial institutions consider the mortgage in danger of default due to nonpayment. As a result, some loss mitigation strategies such as forbearance and loan modification can be applied to help borrowers.

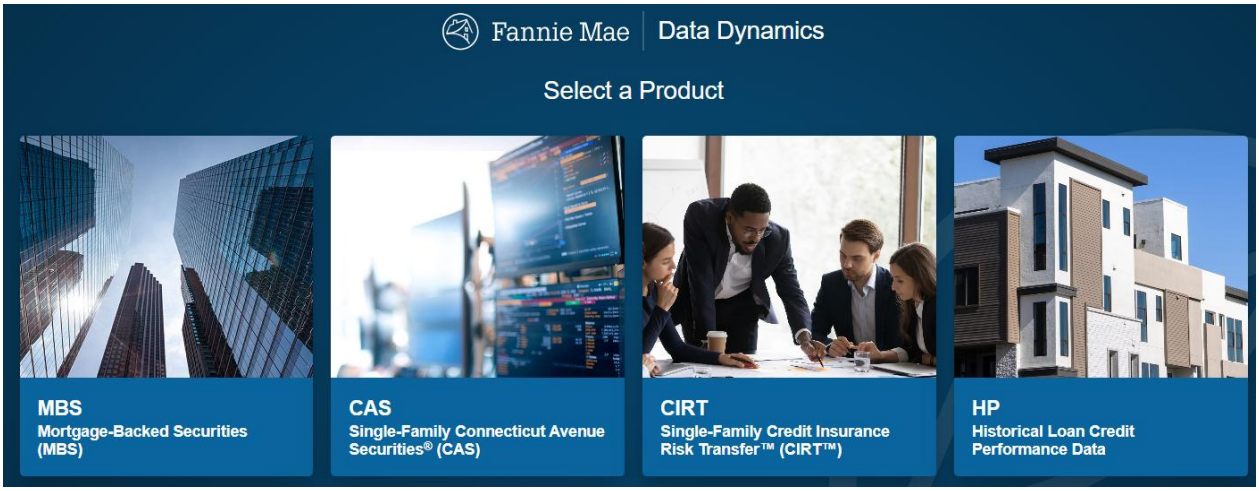
The data for this analysis comes from Fannie Mae Data Dynamics. And I will apply machine learning classification methods, such as logistic regression, decision tree and random forest to mortgage data. R/RStudio is the major analytical tool for this analysis.

2.Mortgage Data and Analytical Tools

2.1 Data Source

For this project, I find and download multifamily mortgage data from Fannie Mae Data Dynamics. Data Dynamics is a free data analytics platform in Fannie Mae to evaluate and analyze vast amount of mortgage data.

Basically, it contains loan level data from 2000 to 2021. This is a huge dataset, over 2GB, in csv format. This dataset includes 4,628,626 rows and 62 variables. I will explain more about the variables in the next section.



2.2 Variables

I list all 62 variables in the dataset below. These variables have several types of variables in the dataset, such as date, continuous and categorical.

[1] "Loan.Number"
[2] "Acquisition.Date"
[3] "Note.Date"
[4] "Maturity.Date.at.Acquisition"
[5] "Loan.Acquisition.UPB"
[6] "Amortization.Type"
[7] "Interest.Type"
[8] "Loan.Product.Type"
[9] "Original.UPB"
[10] "Amortization.Term"
[11] "Original.Interest.Rate"
[12] "Lien.Position"
[13] "Transaction.ID."
[14] "Issue.Date"
[15] "Loan.Acquisition.LTV"
[16] "Underwritten.DSCR"
[17] "Underwritten.DSCR.Type"
[18] "Original.Term"
[19] "Original.I.O.Term"
[20] "I.O.End.Date"
[21] "Loan.Ever.60..Days.Delinquent"
[22] "Loss.Sharing.Type"
[23] "Modified.Loss.Sharing.Percentage"
[24] "Number.of.Properties.at.Acquisition"
[25] "Property.Acquisition.Total.Unit.Count"
[26] "Specific.Property.Type"
[27] "Year.Built"
[28] "Property.City"
[29] "Property.State"
[30] "Property.Zip.Code"
[31] "Metropolitan.Statistical.Area"
[32] "Physical.Occupancy.."
[33] "Liquidation.Prepayment.Code"
[34] "Liquidation.Prepayment.Date"
[35] "Foreclosure.Date"
[36] "Credit.Event.Date"
[37] "Foreclosure.Value"
[38] "Lifetime.Net.Credit.Loss.Amount"
[39] "Sale.Price"
[40] "Default.Amount"
[41] "Credit.Event.Type"
[42] "Reporting.Period.Date"
[43] "Loan.Active.Property.Count"
[44] "Note.Rate"
[45] "Maturity.Date...Current"
[46] "UPB...Current"
[47] "Delinquency.UPB"
[48] "Loan.Payment.Status"
[49] "SDQ.Indicator"
[50] "Most.Recent.Modification.Date"
[51] "Modification.Indicator"
[52] "Defeasance.Date"
[53] "Prepayment.Provision"
[54] "Prepayment.Provision.End.Date"
[55] "Affordable.Housing.Type"
[56] "MCIRT.Deal.ID"
[57] "MCAS.Deal.ID"
[58] "DUS.Prepayment.Outcomes"
[59] "DUS.Prepayment.Segments"
[60] "Loan.Age"
[61] "Green.Bond.Indicator"
[62] "Social.Bond.Indicator"

Examples for the date variables include Acquisition.Date, Note.Date, Maturity.Date.at.Acquisition. And Loan.Acquisition.UPB (Unpaid Balance), Original.UPB are illustration of continuous variables. For categorical variables, Interest.Type, Amortization.Term, SDQ(Serious delinquent) are examples. In the table below, I demonstrate one example of each type of variable.

Variable Type	Variable Name	Example
Date	Acquisition.Date,	2000-01-01
Continuous	Loan.Acquisition.UPB	$82,501.71
Categorical	SDQ	Y/N

In the project, I'm particularly interested in the reasons which cause seriously delinquent. Therefore, I will use variable SDQ as the independent variable. There are lots of choices for independent variables, such as UPB, interest rate type, term, lien position, etc.

- Dependent variable: SDQ - seriously delinquent
- Independent variable: for simplicity, I will just use 5 variables as independent variable in this analysis, Namely,
    - loan amount(Loan.Acquisition.UPB), continuous variable.
    - interest rate(Original.Interest.Rate), continuous variable.
    - loan term(Original.Term), continuous variable.
    - amortization type (Amortization.Type), categorical variable.
    - lien (Lien.Position), categorical variable.

2.3 Analytical tools

The major tool for this project is R/RStudio. I upgrade my R to version 4.4.1. Following chart is an illustration of the dataset in RStudio.

	Loan.Number	Acquisition.Date	Note.Date	Maturity.Date.at.Acquisition	Loan.Acquisition.UPB	Amortization.Type	Interest.Type	Loan.Product.Type	Original.UPB
1	140296	2000-10-31	1985-07-16	2001-08-10	$82,501.71		ARM	DUS	$82,501.71
2	140296	2000-10-31	1985-07-16	2001-08-10	$82,501.71		ARM	DUS	$82,501.71
3	140296	2000-10-31	1985-07-16	2001-08-10	$82,501.71		ARM	DUS	$82,501.71
4	140296	2000-10-31	1985-07-16	2001-08-10	$82,501.71		ARM	DUS	$82,501.71
5	140297	2000-10-31	1985-07-18	2001-08-10	$548,872.98		ARM	DUS	$548,872.98
6	140297	2000-10-31	1985-07-18	2001-08-10	$548,872.98		ARM	DUS	$548,872.98
7	140297	2000-10-31	1985-07-18	2001-08-10	$548,872.98		ARM	DUS	$548,872.98
8	140297	2000-10-31	1985-07-18	2001-08-10	$548,872.98		ARM	DUS	$548,872.98
9	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
10	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
11	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
12	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
13	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
14	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
15	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
16	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
17	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
18	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
19	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00
20	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS	$14,736,000.00

R is an open-source software. Whenever using R in data analysis, I need to install a few libraries. For this project, I install and use following packages in this project.

- readxl: import data
- dplyr: data manipulation
- tree: decision tree
- randomForest: random forest
- gbm: boosting

2.4 Data preparation

The first step in data preparation is to remove partial duplicates. As you can see from the chart in section 2.3, one loan number has several rows. After removing partial duplicates, each loan number only has one row. There are 39,390 rows total.

Filter | Cols: « < 1 - 50 > »

	Loan.Number	Acquisition.Date	Note.Date	Maturity.Date.at.Acquisition	Loan.Acquisition.UPB	Amortization.Type	Interest.Type	Loan.Product.Type
1	140296	2000-10-31	1985-07-16	2001-08-10	$82,501.71		ARM	DUS
2	140297	2000-10-31	1985-07-18	2001-08-10	$548,872.98		ARM	DUS
3	1673867584	2000-01-01	1999-11-01	2009-12-01	$14,725,385.83	Amortizing Balloon	Fixed	DUS
4	1673867585	2000-01-01	1999-11-01	2009-12-01	$7,248,775.03	Amortizing Balloon	Fixed	DUS
5	1673887828	2000-01-01	1999-12-01	2009-12-01	$6,196,905.91	Amortizing Balloon	Fixed	DUS
6	1673888928	2000-01-01	1999-11-30	2009-12-01	$4,061,163.35	Amortizing Balloon	Fixed	DUS
7	1673893027	2000-01-01	1999-11-19	2009-09-01	$7,050,000.00	Interest Only/Balloon	Fixed	DUS
8	1673893029	2000-01-01	1999-12-01	2010-01-01	$4,396,000.00	Amortizing Balloon	Fixed	DUS
9	1673893030	2000-01-01	1999-12-01	2010-01-01	$3,368,000.00	Amortizing Balloon	Fixed	DUS
10	1673901049	2000-01-01	1999-12-01	2010-01-01	$5,800,000.00	Amortizing Balloon	Fixed	DUS
11	1673901050	2000-01-01	1999-12-16	2010-01-01	$46,553,280.00	Amortizing Balloon	Fixed	DUS
12	1673904357	2000-01-01	1999-12-02	2009-12-01	$15,359,252.01	Amortizing Balloon	Fixed	DUS
13	1673904358	2000-01-01	1999-11-23	2017-12-01	$938,401.22	Amortizing Balloon	Fixed	DUS
14	1673904359	2000-01-01	1999-12-06	2007-01-01	$3,626,700.00	Amortizing Balloon	ARM	DUS
15	1673904360	2000-01-01	1999-12-13	2010-01-01	$1,270,000.00	Amortizing Balloon	Fixed	DUS
16	1673904361	2000-01-01	1999-12-09	2010-01-01	$19,045,000.00	Amortizing Balloon	Fixed	DUS
17	1673908755	2000-01-01	1999-12-01	2010-01-01	$4,000,000.00	Amortizing Balloon	Fixed	DUS
18	1673908756	2000-01-01	1999-12-01	2010-01-01	$5,800,000.00	Amortizing Balloon	Fixed	DUS
19	1673908757	2000-01-01	1999-12-01	2018-01-01	$3,085,000.00	Amortizing Balloon	Fixed	DUS

Next, I check the distribution of independent variables and dependent variable. As we can see from following table, the max Unpaid Principal Balance (UPB) is $607.5 million and the average UPB is $9.2 million. The original interest rate ranges from 1.222% to 9.5%. The original terms are less than 30 years.

	Loan.Acquisition.UPB	Original.Interest.Rate	Loan.Acquisition.LTV	Original.Term
Min.	11,440	1.222	0	11
1st	2,115,000	4.37	60	108
Median	4,600,000	5.33	69.9	120
Mean	9,208,161	5.222	65.98	119
3rd	10,400,000	5.99	75.7	120
Max.	607,500,000	9.5	276.2	360

Among, 39390 records, only 987 loans are SDQ. I change SDQ data type to factor in R, because R requires this for future analysis.

	Amortization.Type	Lien.Position	SDQ.Indicator	SDQ.factor
Length	39390	39390	39390	N:38403
Class	character	character	character	Y: 987
Mode	character	character	character	

Then, I examine the two categorical independent variables. For Amortization.Type, most of them are Amortizing Balloon, followed by Interest Only/Amortizing/Balloon and Interest Only/Balloon. For Lien.Position, most them are first lien. Only about 10% loans are second lien. Third or more liens are very small amount.

Amortization.Type	Count
Amortizing Balloon	25463
Fully Amortizing	1414
Interest Only/Amortizing/Balloon	9197
Interest Only/Balloon	3301
Interest Only/Fully Amortizing	13
blank	2
Total	39390

Lien.Position	Count
First	35001
Second	4061
Third	283
Fourth or More Subordinate	44
Blank	1
Total	39390

3. Machine Learning Methods

In this section, I will use 5 different machine learning or artificial intelligence methods to explore the classification problem. I start with logistic regression and then I try ensemble methods including decision tree, bagging, random forest and boosting.

3.1 Logistic regression

For the logistic regression, the dependent variable is SDQ and the 5 Independent variables are

- loan amount(Loan.Acquisition.UPB), continuous variable.
- interest rate(Original.Interest.Rate), continuous variable.
- loan term(Original.Term), continuous variable.
- amortization type (Amortization.Type), categorical variable.
- lien (Lien.Position), categorical variable.

After fitting the logistic regression in R using glm() function, the estimated Coefficients and other parameters are listed below.

Coefficients:	Estimate	Std.Error	z-value	Pr(>|z|)	
(Intercept)	2.06E+00	6.56E+02	0.00300	0.99749	
Loan.Acquisition.UPB3	-3.41E-08	5.41E-09	-6.31200	0.00000	***
Original.Interest.Rate	5.15E-01	3.04E-02	16.93600	<2e-16	***
Original.Term	-2.74E-03	1.05E-03	-2.62000	0.00881	**
Amortization.TypeAmortizing.Balloon	-1.79E+01	3.79E+02	-0.04700	0.96239	
Amortization.TypeFully.Amortizing	-1.83E+01	3.79E+02	-0.04800	0.96149	
Amortization.TypeInterest.Only/Amortizing/Balloon	-1.72E+01	3.79E+02	-0.04500	0.96371	
Amortization.TypeInterest.Only/Balloon	-1.87E+01	3.79E+02	-0.04900	0.96067	
Amortization.TypeInterest.Only/Fully.Amortizing	-2.72E+01	4.06E+02	-0.06700	0.94668	
Lien.PositionFirst	9.82E+00	5.35E+02	0.01800	0.98536	
Lien.PositionFourth.or.More.Subordinate	8.91E+00	5.35E+02	0.01700	0.98673	
Lien.PositionSecond	8.74E+00	5.35E+02	0.01600	0.98697	
Lien.PositionThird	9.29E+00	5.35E+02	0.01700	0.98615	

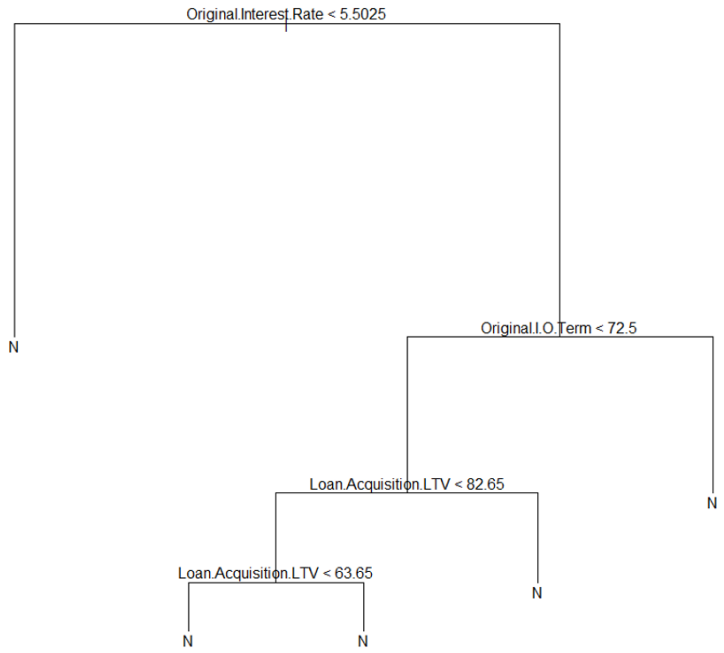
As we can see from the table, p-values for UPB, original interest rate and original term are almost 0. Thus, there is clear evidence of a real association between them and SDQ. However, since the p-values for amortization type and lien position are large, there is no clear evidence of a real association between them and SDQ.

3.2 Decision tree

There are two types of decision tree: regression tree and classification tree. A classification tree can be applied to this project, because SDQ is a categorical variable.

The decision tree is easy to explain and understand. In fact, it's even easier than to explain linear/logistic regression, because decision tree is more similar to how human make decisions. In addition, trees can be displayed graphically. Therefore, it's easily to interpreted even to non-technical people.

I plot the decision tree for this project. As we can see from the chart below. The first node is original interest rate. The cut off of interest rate is 5.5025%. The lower interest rate, the less likely to be SDQ.



The second cut off is original IO term. If the original IO term is greater than 72.5, then the mortgage is not likely to be SDQ. Otherwise, it's possible to be SDQ.

The decision tree also lists the variables that are used as internal nodes in the tree, the number of terminal nodes, and the error rate. In the tree I built, the misclassification error rate is 2.27%.

```
Classification tree:
tree(formula = SDQ.factor ~ . - SDQ.factor - SDQ, data = MFLP3)
Variables actually used in tree construction:
[1] "Original.Interest.Rate" "Original.I.O.Term"      "Loan.Acquisition.LTV"
Number of terminal nodes:  5
Residual mean deviance:  0.1852 = 2307 / 12460
Misclassification error rate: 0.02271 = 283 / 12464
```

3.3 Bagging

From here, I will try some ensemble method, which is an approach that combines many simple models to obtain a single and potentially very powerful model. I will start with bagging, then try random forest and boosting.

The decision trees suffer from high variance. This means that if we split the training data into two parts at random, and ft a decision tree to both halves, the results that we get could be quite different. Bagging is a general-purpose procedure for reducing the variance of a machine learning method.

From following table, we can see that this is a classification random forest, which includes 500 trees. The out-of-bag(OOB) error rate is 2.48%, which is a little bit higher than the misclassification error rate of 2.271% for a single tree. Thus, the bagging doesn't improve the accuracy over the single tree. The confusion matrix has very low error for N.

```
Call:
randomForest(formula = SDQ.factor ~ Loan.Acquisition.UPB3 + Original.Interest.Rate +      Original.Term +
Amortization.Type + Lien.Position, data = MFLP3,  importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2
        OOB estimate of error rate: 2.48%
Confusion matrix:
          N   Y   class.error
N 38377  26  0.0006770304
Y      950  37  0.9625126646
```

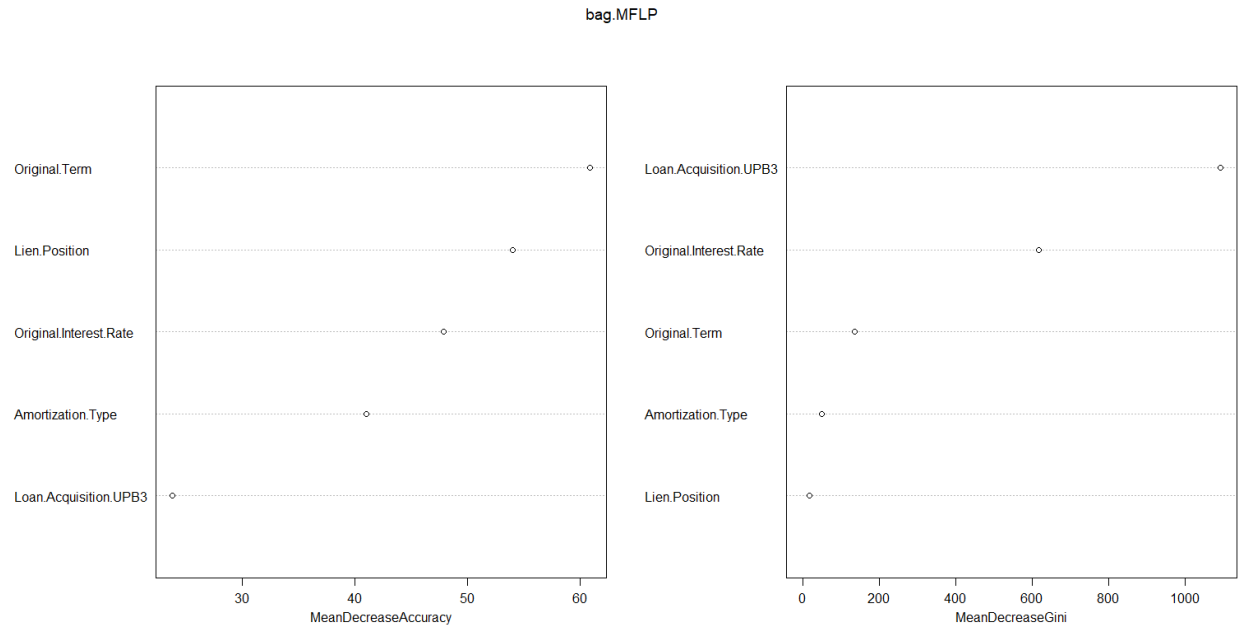
Next, I create the table for mean decrease accuracy and mean decrease Gini. This is a fundamental outcome and it shows for each variable how important it is in classifying the data. The higher the value of mean decrease accuracy or mean decrease Gini score, the higher the importance of the variable.

The Mean Decrease Accuracy expresses how much accuracy the model losses by excluding each variable. The more the accuracy suffers, the more important the variable is for the successful classification. In this project, the top 3 variables are original.term, Lien.Position and Original.Interest.Rate.

The mean decrease in Gini coefficient is a measure of how each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest. In this project, the top 3 variables are Loan.Acquisition.UPB3, Original.Interest.Rate, Original.Term

	N	Y	MeanDecreaseAccuracy	MeanDecreaseGini
Loan.Acquisition.UPB3	16.77846	34.895032	23.77978	1092.74187
Original.Interest.Rate	32.5116	90.066972	47.82772	616.63661
Original.Term	53.58865	41.162471	60.85125	136.31815
Amortization.Type	27.83412	65.853935	40.9981	49.48313
Lien.Position	53.12982	6.750389	53.96096	17.47154

The figure below visualizes the above results.



3.4 Random Forests

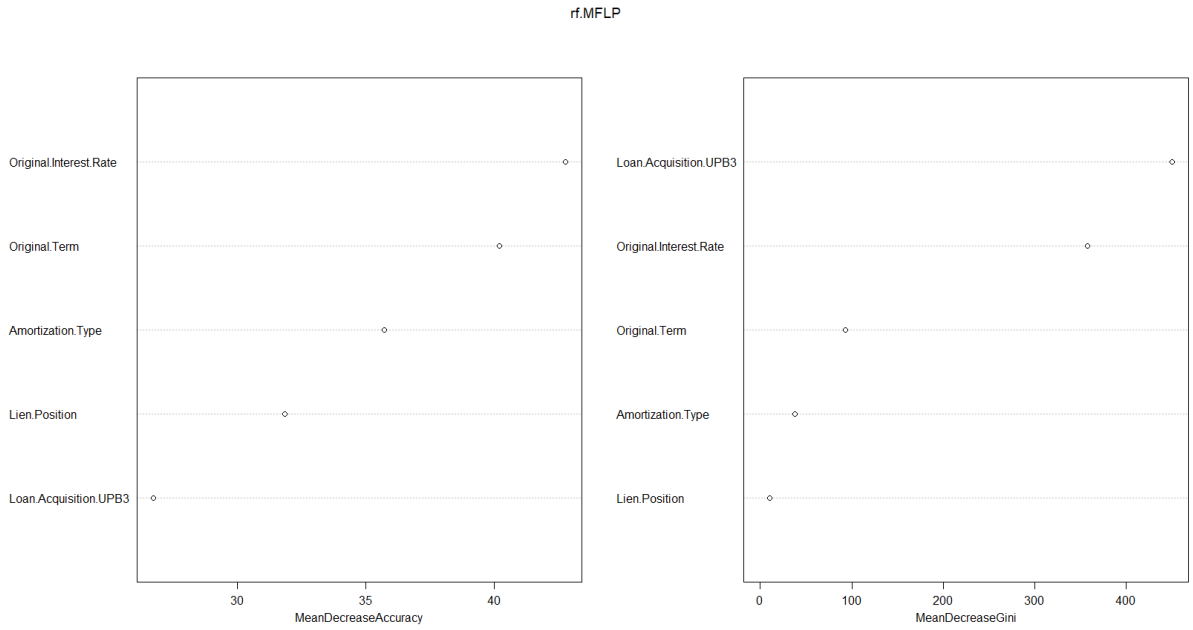
Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees. Bagging builds a number of decision trees on bootstrapped training samples. In section 3.3, I choose 2 as the number of variables tried at each split. Given the similarity of bagging and random forests, I will choose 3 here.

The results from random forests are close to those from bagging. OOB estimate error rate increase a little to 2.51%. This is still higher than the misclassification error rate 2.27% from single decision tree.

```
Call:
 randomForest(formula = SDQ.factor ~ Loan.Acquisition.UPB3 + Original.Interest.Rate +      Original.Term +
Amortization.Type + Lien.Position, data = MFLP3,   mtry = 3, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 2.51%
Confusion matrix:
          N    Y   class.error
N 38349 54  0.00140614
Y   934    53  0.94630193
```

Figure below is the variable importance from random forest. The most importance variable from mean decrease accuracy is Original.Interest.Rate, followed by Original.Term and Amortization.Type.



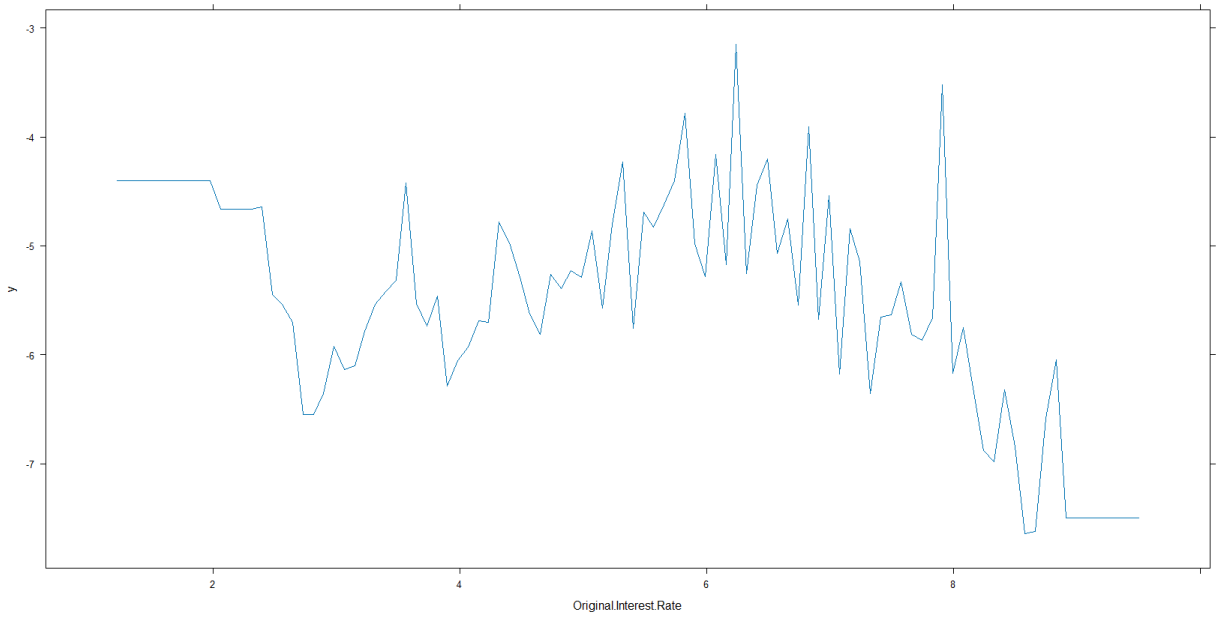
3.5 Boosting

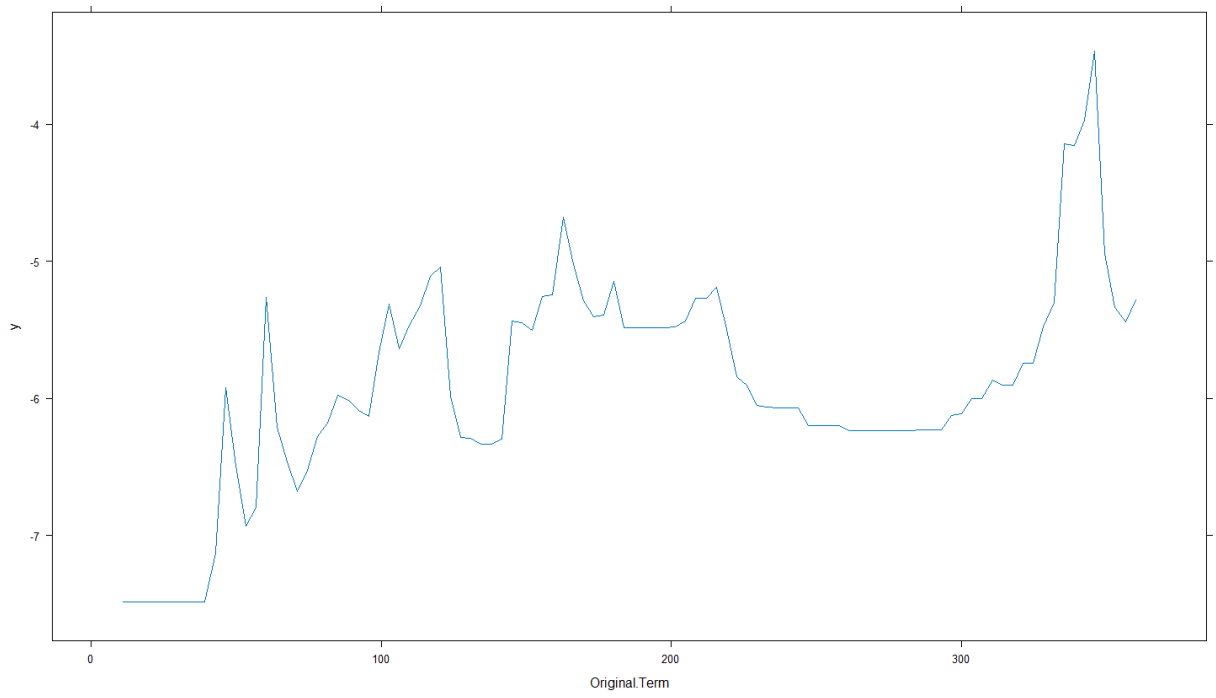
Boosting is another way to improve the results from decision tree. It also can be applied to regression and classification problems. In bagging, each tree is built on a bootstrap data set, independent of the other trees. However, in boosting, each tree is grown sequentially. That is to say, each tree is grown using information from previously grown trees.

Table below produces the outputs the relative influence statistics from boosting. As we can see, Original.Interest.Rate and Loan.Acquisition.UPB3 are the most important variables.

var	rel.inf
Original.Interest.Rate	34.412509
Loan.Acquisition.UPB3	32.098218
Loan.Acquisition.LTV	26.034847
Original.Term	7.454426

I also produce partial dependence plots for Original.Interest.Rate and Original.Term. They illustrate the marginal effect of the selected variables on the response after integrating out the other variables.





4.Conclusion

In this project, I try to analyze the factors that impact mortgage seriously delinquency by using AI and ML methods. I first download Fannie Mae Multi-family loan performance data and then clean the data. For the analysis, I start with the classical method for classification problem, logistic regression. And then, I apply decision tree to this problem. Finally, I try ensemble methods such as bagging, random forests and boosting in the analysis. In conclusion, I still believe logistic regression and decision tree are the best methods.

Reference

- Fannie Mae Data Dynamics, https://datadynamics.fannie mae.com/data-dynamics/#/reportMenu;category=HP
- LenderLetter, https://singlefamily.fanniemae.com/media/33711/display
- R code: https://capitalmarkets.fanniemae.com/credit-risk-transfer/single-family-credit-risk-transfer/fannie-mae-single-family-loan-performance-data
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2023). An introduction to statistical learning with applications in R. Springer.
- Package 'gbm', https://cran.r-project.org/web/packages/gbm/gbm.pdf
- Package 'h2o', https://cran.r-project.org/web/packages/h2o/h2o.pdf
- Variable importance plot (mean decrease accuracy and mean decrease Gini). https://plos.figshare.com/articles/figure/Variable_importance_plot_mean_decrease_accuracy_and_mean_decrease_Gini_/12060105?file=22168122

Appendix: GitHub

More resource for this article such as data and code can be found in my Github at https://github.com/wx2123/SDQ-mortgage/tree/main

Appendix: R code

```
# Wujian Xue
# 2024/6/17

# import data
#install.packages("readxl")
#install.packages("lubridate")

library(readxl)
MFLP <- read.csv("C:/Users/xuewu/Downloads/FNMA_MF_Loan_Performance_Data_202312.csv")
```

```
head(MFLP)
tail(data2)
summary(MFLP)
names(MFLP)

library(dplyr)
MFLP2 <- MFLP %>%
  filter(Liquidation.Prepayment.Code != "")

# remove duplicates
MFLP2 %>% distinct(Loan.Number, .keep_all = TRUE)
summary(MFLP2)

my_string <- gsub('$', '', Loan.Acquisition.UPB)

MFLP_new <- MFLP2 %>%
  mutate(Loan.Acquisition.UPB2 = gsub('[\\$,]', '', Loan.Acquisition.UPB),
         Loan.Acquisition.UPB3 = as.numeric(Loan.Acquisition.UPB2),
         SDQ        = ifelse(SDQ.Indicator =='Y',1,0),
         SDQ.factor = factor(SDQ.Indicator)
         )
summary(MFLP_new)

# select variables
MFLP3 <- MFLP_new %>%
  select(Loan.Acquisition.UPB3,
         #Amortization.Term,
         Original.Interest.Rate,
         Loan.Acquisition.LTV,
         Original.Term,
         Original.I.O.Term,
         Amortization.Type,
         Lien.Position,
         SDQ,
         SDQ.Indicator,
         SDQ.factor
         )

summary(MFLP3)

table(MFLP3$SDQ)
table(MFLP3$Amortization.Type)
table(MFLP3$Lien.Position)

# Logistic Regression
log_model <- glm(SDQ ~ Loan.Acquisition.UPB3 + Original.Interest.Rate + Original.Term +
                   Amortization.Type     +  Lien.Position,
                 data = MFLP3, family = "binomial")

summary(log_model)

#install.packages("writexl")
library("writexl")
write_xlsx(out,"C:\\1910_UoNA\\out.xlsx")

table(MFLP$SDQ.Indicator)

library(tree)
library(ISLR2)

# Decision Tree
tree.MFLP3 <- tree(SDQ.factor ~ Loan.Acquisition.UPB3 + Original.Interest.Rate + Original.Term +
                             Amortization.Type     +  Lien.Position,          MFLP3)
tree.MFLP3 <- tree(SDQ.factor ~ . - SDQ.factor - SDQ,MFLP3)

summary(tree.MFLP3)

plot(tree.MFLP3)
text(tree.MFLP3, pretty = 0)
```

```
# Bag and Random Forest
library(randomForest)
set.seed(1)
bag.MFLP <- randomForest(SDQ.factor ~ ., data = MFLP3,
                         mtry = 12, importance = TRUE)

bag.MFLP

library(randomForest)
set.seed(1)
bag.MFLP <- randomForest(SDQ.factor ~ Loan.Acquisition.UPB3 + Original.Interest.Rate + Original.Term +
                           Amortization.Type     +  Lien.Position, data = MFLP3,
                           mtry = 12,
                         importance = TRUE)

bag.MFLP

importance(bag.MFLP)
varImpPlot(bag.MFLP)

yhat.bag <- predict(bag.MFLP, newdata = MFLP3 )
plot(yhat.bag, MFLP3)
abline(0, 1)
mean((yhat.bag - MFLP3)^2)

set.seed(1)
rf.MFLP <- randomForest(SDQ.factor ~ Loan.Acquisition.UPB3 + Original.Interest.Rate + Original.Term +
                          Amortization.Type     +  Lien.Position, data = MFLP3,
                          mtry = 2, importance = TRUE)
rf.MFLP
importance(rf.MFLP )
varImpPlot(rf.MFLP )

# Boosting
#install.packages("gbm")
library(gbm)
set.seed(1)
boost.MFLP<- gbm(SDQ ~ Loan.Acquisition.UPB3 + Original.Interest.Rate + Original.Term + Loan.Acquisition.LTV,
                 #+ Amortization.Type     +  Lien.Position,
               data = MFLP3,
                   distribution = "bernoulli", n.trees = 5000,
                   interaction.depth = 4)
summary(boost.MFLP)

plot(boost.MFLP, i = "Original.Interest.Rate")
plot(boost.MFLP, i = "Loan.Acquisition.UPB3")
plot(boost.MFLP, i = "Original.Term")
```