Predicting Mortgage Loan Default with Machine Learning Methods

Ali Bagherpour*
University of California, Riverside.

Abstract

This paper applies machine learning algorithms to construct non-parametric, nonlinear predictions of mortgage loan default. I compile a large dataset with over 20 million loan observations from Fannie Mae and Freddie Mac, for the period 2001-2016 at the quarterly frequency. Different machine learning algorithms are applied to predict in sample (training sample), and to forecast out-of-sample (testing data). We find that the forecast performance of nonlinear and non-parametric algorithms are substantially better than the traditional logit model. Additionally, machine learning algorithms allow identification of the predictive power of specific variables. The results indicate that loan age is the most important predictor of loan default before and after the 2008 financial crisis. However, we find that market loan-to-value is the most effective predictor of mortgage loan default during the recent financial crisis. Finally, we use machine learning to formulate risk-based capital stress tests for Fannie Mae and Freddie Mac under different scenarios. We forecast their mortgage credit losses and associated capital needs during the financial crisis. The results obtained are more accurate than those from the Federal Housing Enterprise Oversights (OFHEO), and other existing stress test studies.

Keywords: Mortgage Loan, Machine Learning, Stress test, Risk Managements, Government-Sponsored Enterprises

JEL Classification: C53, C55, C80, G17, G28

*Department of Economics, 3126 Sproul Hall, University of California, Riverside, CA 92507, USA, E-mail: abagh005@ucr.edu.

"The root cause of the troubles that mortgage giants Freddie Mac and Fannie Mae experienced during the financial crisis can be found in the conflicting mandates of their business models"(Daniel H.Mudd, former CEO Fannie Mae)

1 Introduction

The recent financial crisis that hit the U.S. in 2008 and spread to other countries highlighted the importance of risk measurement and management at large financial institutions. Policymakers have since been conducting annual stress tests of banks as one of their supervision and regulation tools. These tests involve evaluating financial stability under adverse economic scenarios. Although these tests may help stabilize financial institutions against severe recessions or crises, they are also subject to shortfalls. In particular, the tests require model forecasts of financial institutions' losses, which involve "model risk" arising from problems in model validation, stress scenarios, and data[1]. That is, the model used may be misspecified, the scenarios may not accurately reflect all possible outcomes, and the data might not contain enough information, which can all lead to unreliable stress test results. Thus, model specification and data choices can be crucial to their success.

This paper proposes the use of machine learning techniques to model and forecast bank losses, which can be used in stress tests. In particular, we study the dynamics of mortgage loan losses in the U.S. and provide forecasts. We use a set of parametric and non-parametric algorithms specifically designed to tackle computationally intensive recognition of patterns in very large datasets. The methods are applied to a large data set of over 20 million observations ("big data"). The methods use all possible information available regardless on prior conceptions about their usefulness, and take into account interaction among all variables. Thus, they may unveil underlying relationships that have not yet been discovered. We compare the forecast performance of several different specifications. Since the methods are not constrained to a limited dataset, but use instead full available information in a big data set, the methods reduce model risk in terms of specification and data[2].

Two main financial service corporations that perform an important role in the

[1]See e.g. Frame, Gerardi, and Willen (2015) and Jacobs, Sensenbrenner, and Karagozoglu (2017), Kandani, Kim, and Lo (2010)

[2]There are very few studies that apply machine learning in Economics (there are many more in Finance and a large literature on science fields). Recently, Chakraborty and Joseph 2017 introduced practical aspects of machine learning and associated methodologies in the context of central banking and policy analysis. Other studies with applications of machine learning in Economics are Varian (2014) and Einav and Levin (2013).

U.S. housing finance system are Fannie Mae and Freddie Mac. In the past decades, these agencies led to increases in home ownership of low- and middle-income families by improving availability of mortgage credit under a range of economic conditions. The Office of Federal Housing Enterprise Oversight (OFHEO) has developed and conducted capital regulation "stress tests" on these agencies starting in 2002. However, these tests proved to be a failure in forecasting the housing crisis and mortgage loan default risks of the late 2000s. The large losses of these agencies led them to be insolvent and the Federal Housing Finance Agency (FHFA) put them into conservatorship in 2008[3].

Frame, Gerardi, and Willen (2015) study the sources of failure of these OFHEO risk-based capital stress tests on 30-year fixed-rate mortgage performance. They challenged the OFHEO stress test in terms of model specification, sample period, scenarios, and assumptions. In particular, these tests were estimated using mortgages data from 1979 and 1997, and then applied to forecast mortgage performance between 2002 and 2008. That is, the tests were implemented with no changes in variables, and the models and forecasts of mortgage default were not reestimated as new data become available[4]. Frame, Gerardi, and Willen (2015) forecast quarterly default forecasts over the period 2000-2010 using the Lender Processing Services (LPS) dataset for this period. They consider loan performance data and relevant acquisition features, such as borrower credit score, loan documentation and local unemployment rates. The OFHEO stress test, on the other hand, used only loan performance data without regarding information on borrowers' risk. Frame, Gerardi, and Willen (2015) show that their updated model improves forecasts of loan default and default rate even from 2005 on, when the housing market risk increased substantially. However, the paper use data from 2000 to 2010 to provide predictions for the same period. That is, the study does not conduct out-of- sample real time forecasting tests. Also, both OFHEO and Frame, Gerardi, and Willen (2015) used a logistic regression to forecast loan default, which is one of the most widely used techniques for classification purpose.

This paper uses machine learning algorithms to model and forecast loan performances using a new large database of over 20 million observations from Fannie Mae and Freddie Mac single family dataset, which was released in the beginning of 2013.

[3]Concurrently, the Treasury entered into senior preferred stock purchase agreements with each institution. Under these agreements, U.S. taxpayers ultimately injected $187.5 billion into Fannie Mae and Freddie Mac.

[4]One of the reasons, as pointed by Frame, Gerardi, and Willen (2015), was that all details of OFHEO stress tests had to be made public by law, which made it difficult to implement constant updates and changes.

The quarterly data are at loan level ranging from 2001Q1 to 2016Q4, and include loan performances and acquisitions. In order to evaluate the model performance and provide forecasts that could be used out of sample in real time, we divide the data into three samples: before the financial crisis (2000-2006), during the crisis (2007-2011), and after the crisis (2012-2015). In each period, the data is split into a training dataset (70%) for model estimation and a testing dataset (30%) used for out of sample forecasts.

We apply four machine learning models to forecast mortgage loan default: K-Nearest Neighbors (KNN), the tree-based classifier Random Forest (RF), Support-Vector Machines (SVM), and Factorization Machines (FM). These models are ideally suited for loan level analysis because of the large sample sizes and complexity of the possible relationships among features. We compare our results with those obtained from logistic regression as estimated in the OFHEO stress tests and in Frame, Gerardi, and Willen (2015). The logistic regression is the most widely used techniques for classification purpose[5].

The machine learning methods used have many advantages and some disadvantages over logistic regressions. In particular, K-nearest neighbors are non-parametric, simple, and suited for forecasting, particularly when the training data contain missing observations and outliers, and are unbalanced. However, since the method does not offer a description of the learned concepts, the results are not directly interpretable. Random Forest applies the general technique of bootstrap aggregating to tree learners. This algorithm is not only extremely useful for forecasting but the methods also allow classification on whether a variable is more or less important when constructing the (bootstrapped) decision tree. The Support-Vector Machines algorithm seeks to find a hyperplane that separates the data as well as possible. Additionally, we can use the kernel function to unveil non-linear connection between input variables with output variables. Finally, we propose the application of Factorization Machines introduced by Rendle (2010) to forecast mortgage loan defaults. Factorization Machine models consider all interactions among variables implicitly, which may lead to large improvements in forecast accuracy.

The results indicate that each one of the four machine learning methods considered display higher forecasting accuracy relative to logistic models. We evaluate the forecast performance using the metric Area Under the Curve (AUC) using the ROC Curve, which plots the true positive rates against false positive rates[6]. The AUC

[5]logistic regression are parametric algorithms that consider linear connections between predictors and classes. Moreover, these regressions measure the effect of changes in a predictor on the response, which is independent of the values of the other predictors.

[6]Given that we balance the data before model fitting and forecasting, other accuracy measures

is a metric for binary classification that measures the accuracy of forecasts, ranging from 50% to 100% [7].

The machine learning methods provide substantially more accurate forecasts of mortgage loan defaults than the traditional logit methods for all periods including before, during, and after the finance crisis. The AUC values for Factorization Machines forecasts are the highest compared to all other methods with values between 88% and 91%. This is followed by the K-Nearest Neighbors and Random Forest forecasts, whose AUC is 88% on average, while the forecast accuracy of the nonlinear Support-Vector Machine with radial kernel is around 87%. On the other hand, the AUC for logit models are just around 85%.

Additionally, some of the machine learning methods (such as Random Forest) allow classification of the importance of variables in forecast accuracy in each period considered. We find that negative equity is the most important variable in loan default during the financial crisis. However, the age of the loan has the most weight in forecasts of loan default during normal periods.

We construct out-of-sample forecasts with four-year rolling windows starting from 2001 to 2016. The results show that all machine learning methods provide substantially more accurate forecasts of mortgage loan defaults compared to logistic regressions.

The remainder of the paper is structured as follows: section 2 describes the machine learning models used, which include K-Nearest Neighbors, Random Forests, Support Vector Machines, and Factorization Machines. Section 3 explain the data collection, data treatment and procedure, and provides statistical analysis of Fannie Mae and Freddie Mac loan level data. Section 4 reports the empirical results for several forecasting procedures in-sample and out-of-sample. Section 5 concludes.

2 Machine learning algorithms

In this section, we describe the machine learning techniques that we use to forecast mortgage loan default in a supervised learning framework. In the supervised learning problem, a learner is presented with input/output pairs, where the input data to be used to determine the output value (Khandani et al.,2010)[8]. In this study the supervised learning problem represent as classification problem since the output is a

such as Recall, Precision, Sensitive, and F1 score are all encompassed in the AUC metric used in this paper.

[7]Generally, the AUC value between 0.9-1 is considered excellent forecast accuracy, 0.8-0.9 good, 0.7-0.8 fair, 0.6-0.7 poor, 0.5-0.6 fail.

[8]In this study, we call input variables either "features" or "exogenous variables"

discreet binary variables. In fact, we are looking for a classifier function that correctly maps such input vectors to the output values. We focus on commonly used machine learning techniques such as KNN, RF, SVM, and FM and compare them with the logistic regression.

2.1 Logistic Regression

This study uses a logistic regression as a benchmark model to forecast mortgage loan defaults to compare logistic regression forecast performance with other classifier methods. Let Y be a binary random variable that gets value one if the loan is defaulted and zero otherwise. Suppose $X = (x_1....x_p)$ are exogenous variables. Then the conditional probability of a defaulted loan is:

$$p(X) = E(Y|x_1, x_2, ..., x_p) = \frac{exp(\beta_0 + \sum_{i=1}^{p} X_i\beta_i)}{1 + exp(\beta_0 + \sum_{i=1}^{p} X_i\beta_i))} \tag{1}$$

Note that feature space$X$ contains to the borrower characteristic variables, loan performance variables ,and macroeconomic variables such as the unemployment rate, slope of yield curve, and housing price index that do not vary for loans but change across geographic areas during the life of the mortgage loan. Applying the transformation and taking logarithm from equation 1:

$$log(y) = log(y/(1-y)) = \beta_0 + \sum_{i=1}^{p} X_i\beta_i \tag{2}$$

Equation 2 is a linear model in the parameters and one can use "maximum Likelihood" methods to estimate parameters $\beta = (\beta_1, ..., \beta_p)$. These parameters have been applied to forecast testing observations. For example, the default probability of a new observation from testing data x^* can be written as :

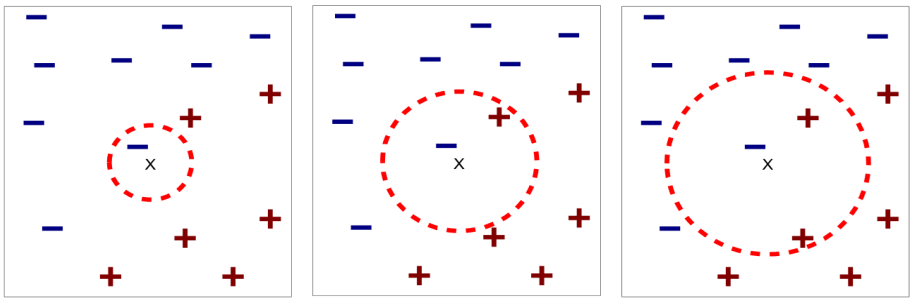
$$p(x^*) = \frac{exp(\beta_0 + \sum_{i=1}^{p} X_i^*\beta_i)}{1 + exp(\beta_0 + \sum_{i=1}^{p} X_i^*\beta_i))} \tag{3}$$

If the probability is more than a certain threshold, then observation x^* takes value one and zero otherwise.

2.2 K-Nearest Neighbors (KNN)

In a forecasting problem with a classification setting we are interested to know what is the appropriate class for a new test observation. However, we don't know the exact

Figure 1: Definition of K Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

distribution of the data. One of the simple, non-parametric methods that doesn't consider a distribution for the data is K-Nearest Neighbors (KNN). KNN highlights K points close to the test observation [9] x^*, and fits a conditional probability for class $i = \{1, 0\}$ as a portion of K points that belong to class$i$ in the training dataset:

$$Pr(Y = i|X = x_0) = 1/K \sum I(y_i - i) \tag{4}$$

One of the challenging parts of KNN is choosing K. In fact, K sets the trade off between variance and bias in the estimation. A small number of K returns low bias and high variance for the prediction.This causes excellent prediction performances during training observation but weak performances during the testing dataset. If $K = 1$ the forecast error rate is zero for the training observation but unreliable for the test datasets. On the other hand, a large K may decrease the variance of the predication but cause high bias that returns an inaccurate forecast. Figure 1 shows an example for KNN algorithm with k =1,2, and 3. There are two classes (+,-), and x is a new observation. When k =1 , the new observation belongs to class -, however, KNN classifier assigns + to the new observation when k=3. This study uses across validation methods to find the optimal K.

In comparison to logistic regressions that present as a linear decision boundary, KNN is non-parametric methods and doesn't assume decision boundary. Therefore, one can expect the KNN approach to return higher performances than a logistic regression when the decision boundary is not linear. On the other hand, KNN is not able to find important predictors, and doesn't deliver the table of coefficients.

[9] One can use euclidean distance $d(p, q) = \sqrt{\sum_i (p_i - q_i)^2}$ to measure how close two points are.

2.3 Tree Based Learning Algorithms

Decision trees are useful algorithms for interpretation since they are simple but generally have a lower prediction performance in comparison to other machine learning algorithms. In the classification tree, the training observations are partitioned in to m regions. A class is assigned to each region based on the most commonly occurring classes. The tree classifier finds the cut off points by the minimizing the classifier error rate in each region. Figure 2 shows a tree classifier alghorithm. Suppose there are two classes, loan default and not default, and two features, loan to value(LTV) and debt to income (DTI) . The tree classifiers find cut off points, L_1, L_2 and L_3 such as if LOTV $> L_1$ and DTI $> L_2$, default is assigned; if LOTV $< L_1$ and DTI $> L_2$, Non-default is assigned ; and so on.

There are a few measures for classifier error rates; the simplest one is defined as the portion of the observations that don't belong to the most common classes. We can show it mathematically as:

$$\text{error} = 1 - \max P_{i,k} \tag{5}$$

Where $P_{i,k}$ is the portion of observations that belong to class k in region i. Since the simple classifier error rate is not sensitive enough to grow a tree (James et al., 2015), many studies considered Gini index:

$$\text{G} = \sum_{k=1}^{K} P_{i,k}(1 - P_{i,k}) \tag{6}$$

Similar to other classifiers, one can add a tuning parameter to the above Gini index in order to control the tradeoff between bias and variance, and to avoid overfitting. Specifically, a complicated tree may have a very low bias, and a high variance of fitting. In this case, there is an overfitting issue for training observations. It means, prediction performance is very good during training observations, but very weak during the test set. By making the tree simpler, the variance may decrease significantly with the cost of increasing bias a little. Suppose T_0 is the biggest possible tree. Then for each tuning parameter λ there is a subtree T. The tuning parameter λ is determine by cross validation. Therefore, the Gini index can written as :

$$\text{G} = \sum_{k=1}^{K} P_{i,k}(1 - P_{i,k}) + \lambda \|T\| \tag{7}$$

where $\|T\|$ is the number of terminal nodes of tree T.
There are some advantages to using tree classifiers rather than linear logistic regressions. Besides easy and understandable visualizations, tree classifiers consider the

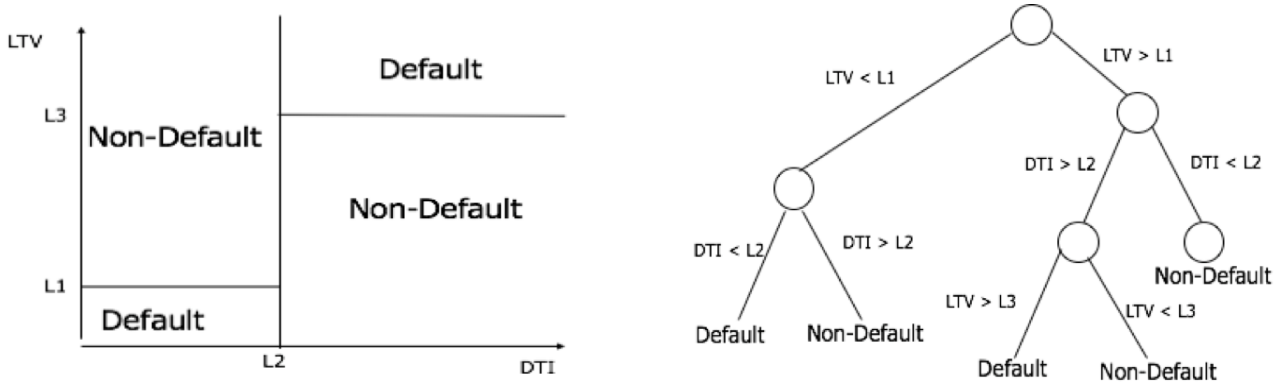


Figure 2: A tree example with binary dependent variables default and non-default, and two independent variables {LTV,DTI}

.

non-linear connection between outputs and inputs.But, one problem with the tree classifier is low accuracy of prediction in relation to other classifiers. In fact, tree classifiers suffer from high variances : a slightly different sample can yield entirely different splits. Furthermore, the forecast performance is good during train observations but weak during the test. To avoid large variance of the tree classifiers, one can bootstrap by taking repeated samples from a training observation and find classifier prediction for each sample. The average of predictions gives a new classifier called bagging:

$$\hat{f(x)} = 1/N \sum_{n=1}^{N} \hat{f}_n(x) \tag{8}$$

Where N is number of bootstraps. Bagging is a proper algorithm to reduce variance of tree classifiers and improve the forecast accuracy by splitting the boosted training observations. However, bagging considers the same features in each split. It means the important features are considered in the top of each bagged tree in each bootstrap. Therefore, highly correlated trees may not decrease the variance of the prediction significantly. To achieve un-corolated trees, one can use Random Forest (RF) algorithms. RF classifiers not only split training observations but get random samples from features. There are different features in each tree and less important features are considered and may get a top position in a specific tree. While the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated.

2.4 Support vector machines (SVM)

Support Vector Machines (SVM) are one of the best learning algorithms for classification and regressions. The SVM finds a hyperplane that separates training observations to maximize the margin(smallest vertical distance between observations and the hyperplane). Intuitively, there are many hyperplanes that can separate the classes and each of them has a certain margin. The distance between observations and the decision boundary explains how sure about prediction. If one observation is in longer distance with hyperplane, more probably it belongs to the correct classes. Therefore, an optimal hyperplane maximizes the margin. This optimal hyperplane is determined based on observations within the margin which are called support vectors. Therefore, the observations outside of support vectors don't influence the hyperplane. Considering a training set $S = \{(x^i, y^i), i = 1, ..., m\}$ where m is the number of observations. Define (w, b) as the smallest margin on training observations S where w contains parameters of the hyperplane and b is the hyperplane intercept [10] :

$$M = min\, M_i \qquad i = 1, ..., m \tag{9}$$

Assume the positive and negative classes can be separated by a linear hyperplane then one can write the SVM optimization problem as :

$$Max_{w,b}\, M \tag{10}$$

$$st \qquad y_i(w^T x_i + b) \geq M, \qquad i = 1, ..., m \tag{11}$$

$$\|w\| = 1 \tag{12}$$

Where M is the margin. Y_i is the class of the training observation i,x_i is the feature spaces i in the training dataset. w are the parameters of the hyperplane, and b is the intercept. Constraint 11 guarantee all of the observations are on the correct side of the hyperplane and constraint 12 ensures that functional and geometric margins are the same. In fact, both constraints together guarantee that each observation is on the correct side of the hyperplane and at least a distance M from the hyperplane (James et al., 2013). Since the constraint $\|w\| = 1$ is non-convex, we can plug in to the objection function and maximize $M/\|w\|$. However, objection function $M/\|w\|$ is still non-convex. By scaling margin M to one the optimization problem can be written as:

$$\begin{aligned} &Min_{w,b} \qquad \frac{1}{2}\|w\|^2 \\ &st \qquad y_i(w^T x_i + b) \geq 1, \qquad i = 1, ..., m \end{aligned} \tag{13}$$

[10]Note that a functional margin define as $\hat{M}_i = y_i(w^T x + b)$ and a geometric margins define as $M_i = y_i((w/\|w\|)^T x_i + b/\|w\|)$

The optimization problem can be solved efficiently and returns parameters w and b. Note that above optimization problem included an inequality constraint. Thus, one can consider a Lagrange duality to solve the problem. We set the Lagrangian problem as :

$$L(w, b, \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \lambda_i [y_i(w^T x_i + b) - 1] \tag{14}$$

In the first step, we should minimize the above Lagrangian with respect to w and b:

$$\bigtriangledown_w L(w, b, \lambda) = w - \sum_{i=1}^{m} \lambda_i y_i x_i = 0 \tag{15}$$

which implies optimal w as:

$$w = \sum_{i=1}^{m} \lambda_i y_i x_i \tag{16}$$

and after taking the derivative with respect to b :

$$\sum_{i=1}^{m} \lambda_i y_i = 0 \tag{17}$$

In the second step, we should plug in the optimal w and constrain 17 in Lagrangian equation and maximize it with respect to λ:

$$\begin{aligned} & max_\lambda \qquad \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \lambda_i \lambda_j < x_i, x_j > \\ & s.t \qquad \lambda_i \geq 0, \qquad i = 1, ..., m \\ & \sum_{i=1}^{m} \lambda_i y_i = 0 \end{aligned} \tag{18}$$

Constraint $\lambda_i \geq 0$ implies that constraint 13 holds with equality. and constraint $\sum_{i=1}^{m} \lambda_i y_i = 0$ coming from equation 17. Also $< x_i, x_j >$ is inner product of two vector x_i and x_j.

Suppose there is a new observation from testing data x^* and we wish to predict if it belongs to class one or zero. We fit the model and find the optimal w as a function of λ, and then calculate $w^T x^* + b$. If this quantity is bigger than zero y takes a value of one and zero otherwise. By substitute 16 in the margin, this quantity can written

as:

$$w^T x^* + b = (\sum_{i=1}^{m} \lambda_i y_i x_i)^T x + b$$
$$\sum_{i=1}^{m} \lambda_i y_i < x_i, x^* > + b \tag{19}$$

In fact, the prediction depends on the inner product of the new observation x^* and the training observations x_i. However, λ_i are all zero except for support vectors; thus, many of the quantities will be zero, and we just need to find the inner products between x^* and the support vectors. Intuitively, the prediction depends on how far the new observation is from support vectors. If it is too far the quantity is higher and we can assign classes with more confidence. Similar to other classifiers, SVM includes parameters that implies tradeoff between bias and variance. Specifically, a hyperplane which separates all observations in exactly two classes may not be very reliable because changing only one observation, can create new hyperplane. Therefore, the SVM algorithm allow some observation to violate the correct classification and stand on the wrong side of the margin, or even the hyperplane. The measure of misclassifications depends on the tradeoff between bias-variance. Specifically when the width of the margin is narrow, the classifier fits the training data well but with low bias and high variance. Further, with a wider margin, more observations lie on the wrong side of the hyperplane and margin and classifier fits the data with lower variance and a higher bias. The measure of misclassification, known as the tuning parameter, is determined by cross validation. Therefore, optimization problem 13 can be adjuststed as:

$$\begin{aligned} & Min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \epsilon_i \\ & s.t \qquad y_i(w^T x_i + b \geq 1 - \epsilon_i \\ & \epsilon_i \geq 0 \quad and \quad \sum_{i=1}^{n} \epsilon_i \leq C \end{aligned} \tag{20}$$

Where C is a non negative tuning parameter and $\epsilon_1, ..., \epsilon_m$ are slack variables (James et al., 2013) that allow training observations to be on the wrong side of the margin.

We consider linear SVM models so far. But, in equation 19 one can substitute the kernel function $K(x_i, x_j)$ instead of the inner product $< x_i, x_j >$ where:

$$K(x_i, x_j) = \sum_{j=1}^{p} x_{ik} x_{jk} \tag{21}$$

Where p is number of observations. Kernel function $K(x_i, x_j)$ is called a linear kernel since the hyperplane is linear in features and it calculates how similar two observations are. However, we may be interested to learn using some features such as x^2, x^3, x_1x_2, and... . In this case we can use non-linear kernels. For example, suppose $\psi(x) = [x, x^2, x^3]^T$. Then one can replace $\psi(x)$ instead of the original x in the above equation. This means we can replace $< \psi(x_i), \psi(x_j) >$in equation 21 and corresponding kernel can be defined as :

$$K(x_i, x_j) = \psi(x_i)^T \psi(x_j) \tag{22}$$

Then the SVM algorithm uses the feature $\psi(x_i)$. One of the advantages of using a kernel instead of the inner product of two features is that computing $\psi(x_i)$ can be very complicated since it is a high dimensional vector(requires $O(n2)$ time),however, calculating a kernel is more simple and efficient (requires $O(n)time$). The intuition of using a SVM algorithm with a kernel is exactly like before: $K(x_i, x_j) = \psi(x_i)^T \psi(x_j)$ shows how far $\psi(x_i)$ and $\psi(x_j)$ are from each other. There are different kernel functions which can reasonably measure how similar $x_i$and x_j are. One of the most well known non linear kernels use in this study is known as the radial kernel function (RBF):

$$k(x_i, x_{i'}) = exp(-c \sum_{j=1}^{P} (x_{ij} - x_{i'j})^2) \tag{23}$$

Which is one when $x_i$and x_j are close and approach zero when they are far from each other. Using the kernel SVM for forecast mortgage loan defaults is different from the logistic regression in at least two ways: First, kernel SVM are non-parametric methods which use kernel function to enlarge the feature space and contain non-linear class boundary. But a logistic regression is a parametric method that estimates parameters based on a linear decision boundary. Second, In SVM only support vectors play a role in the classifier, and other observations do not affect the decision boundary, but in a logistic regression, all observations are involved.

2.5 Factorization Machines

This study proposes the application of Factorization Machines (FM) introduced by (Steffen Rendle, 2010) to forecast mortgage loan defaults. In this section we briefly describe FM in relation to other learning methods. In all methods, we are interested in including interactions between features. For example consider a degree 2 (d=2)

Table 1: The summary of categorical data

Borrower	Loan Size		Unemployment	Loan to Value		Credit Score
one or more	[0, 50]	[50, 100]	5%	[0 , 70%]	[70% , 99%]	600
0	0	1	0	1	0	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
0	0	1	0	0	1	1
0	1	0	0	1	0	0

polynomial classifier :

$$y(x) = w_0 + \sum_{i=0}^{n} w_i x_i + \sum_{i=0}^{n} \sum_{j=i+1}^{n} w_{i,j} x_i x_j \tag{24}$$

Where $w_0 \in R$ is global bias, $w \in R^n$ are the weights for feature vector x_i, and $W \in R^{n \times n}$ is the weight matrix for the feature vector combination $x_i x_j$. This polynomial classifier is an improvement over linear models since they can capture interactions between variables at least for two variables at a time. However, there is a $O(n^2)$ complexity, and training the model requires more time and memory. A more important problem is that the polynomial regression doesn't consider many interactions in a categorical dataset, which is a feature we are interested in. For example, consider table 1 which is a summary of our categorical dataset. The table displays five features which are categorical. Column one takes a value of one If the number of borrowers is more than 1, and zero otherwise ; column 2 is loan size split in to two categories : $[0, 50]$and$[50, 100]$. Column three displays the change in the unemployment as categorical: if unemployment changes more than 5% this variable takes a value of one, and zero otherwise. LTV is split in to two categories : $[0, 70]$and$[70, 99]$, and the last column shows the credit score as categorical: if it is more than 600 takes a value of one and zero otherwise. Given this dataset, the polynomial in equation 24 won't consider the joint effects of a large loan size (e.g. , 70) when the number of borrowers is more than one. However, They can be highly related. This issue arises due to a lack of co-occurrence of variables in a specific row of the design matrix (Perros and Sun ,2015). Factorization machines (FM) on the other hand can ensure that all interactions between pairs of features are modeled using factorized interaction parameters. The FM model of order d=2 is define as follows :

$$\hat{y(x)} = w_0 + \sum_{i=0}^{n} w_i x_i + \sum_{i=0}^{n} \sum_{j=i+1}^{n} x_i x_j \sum_{f=1}^{k} v_{if} v_{jf} \tag{25}$$

Where $w_0 \in R$ is global bias, $w \in R^n$ are the weights for feature vector x_i, and $V \in R^{n \times k}$ is the weight matrix for feature vector combination $v_i v_j$. In fact, the feature combination weights $w_{ij} \in R^{n \times n}$ in an FM model are replaced with factorized interaction parameters between pairs such that, $w_{ij} =< v_i, v_j >= \sum_{f=1}^{k} v_{if} v_{jf}$ where $V \in R^{n \times k}$. One can write equation 25 as [11]:

$$\hat{y(x)} = w_0 + \sum_{i=0}^{n} w_i x_i + \frac{1}{2} \sum_{f=1}^{k} ((\sum_{i=1}^{n} v_{if} x_i)^2 - \sum_{i=1}^{n} v_{i,f}^2 x_i^2) \tag{26}$$

The computation complexity in equation 26 is $O(kn)$. Moreover, this equation shows $< v_i, v_j >$and$< v_i, v_l >$ are related, because parameter (v_i) is shared in both. Thus, variable interaction parameters can be estimated even thought there are no observations about the pairs of interest, since data for one interaction can facilitate the parameter estimation of related interactions (Perros and Sun ,2015)

3 Data

There are several mortgage loan level datasets that can apply for the prediction of loan defaults. The department of Housing and Urban Development (HUD) released single family loan level data on mortgages acquired by Fannie Mae and Freddie Mac. Also, Frame, Gerardi, and Willen (2015) use Lender Processing Services(LPS) mortgage data to estimate loan performances and capital stress tests. Since neither of these two datasets are available to me, I used Fannie Mae single family loan performances datasets released at the beginning of 2013. The public dataset includes around 22 million loan level observations of Fannie Mae's 30-year fixed-rate, amortizing loans that Fannie Mae owned or guaranteed on or after January 1, 2000.[12]We used the data from 2000Q1 to 2015Q4.

Fannie Mae loan performance data divides into two parts : acquisition and performance. Acquisition includes static data at the time of a mortgage loan?s origination and delivery to Fannie Mae and contains three types of information. First, it includes some information about the borrower such as, debt to income ratio (DTI), borrower credit score (C-Score), number of borrowers, and first time home buyer indicators. Second, some property characterizes data such as property type and location of property (State and zip code), and number of units. Finally, fixed loan characteristics

[11]Please see Steffen Rendle, 2010 for the proof

[12]In July of 2015 Fannie Mao published additional credit performances data.

such as original loan to value (OLTV), original date of loan, and original interest rate

The performance part contains the monthly performance data for each mortgage loan from the time of Fannie Mae's acquisition up until the mortgage loan has been liquidated (e.g., paid-off, repurchased, short sale, etc.). This part covers some information about unpaid principle balance (UBP), Interest rate, and delinquency status in each month. [13].

The size and dimension of the dataset is a significant challenge, even for computing basis summary statistics. For this reason, we included variables that either have been considered by FGW(2015) and OFHEO, or intuitively may influence probability of loan defaults. We can categorize theses variables into three parts. First, raw variables which catch directly from either performance or acquisition parts of the Fannie Mae dataset such as DTI, OLTV, loan size, loan age, loan purpose, and credit scores (FICO). Second, the variables that are derived by combining loan information with external economic data, such as mortgage premium (spread), market loan to value (MLTV), and loan size.[14] . Finally, macroeconomic variables such as slope of the yield curve, local unemployment rate, and housing price index. These variables are constant over loans but change across geographic areas during the life of the mortgage loan. List of the variables with more detail are presented in the appendix 1. Moreover, the target variable is a binary outcome that indicates whether a loan is defaulted or not. We follow Fannie Mae's definition of a default: a mortgage loan is defaulted if a property ends through either transfer of the property to third party, sale of real estate less than amount of the debts secured by liens against the property (short sale), owned property by lender after unsuccessful sale at foreclosure auction (REO), or note sale. This definition is consistent with the OFHEO definition of a default when a mortgage is terminated with a loss.

Also, we provide some illustrative examples of the relationship between certain variables and subsequent loan defaults to develop intuition for both the data and the machine-learning algorithms. [15] . OLTV ratio is an index of the borrower?s ability to pay off their loan. Figure 3a displays that loans with higher LTVs are more likely to default and figure 3b shows this connection is stronger during financial crisis. However, OLTV can change over time when the local housing price goes up or down.

[13]This dataset does not include data on adjustable-rate mortgage loans, balloon mortgage loans, interest-only mortgage loans, mortgage loans with prepayment penalties, government-insured mortgage loans, Home Affordable Refinance Program (HARP) mortgage loans, or non-standard mortgage loans.(http://www.fanniemae.com)

[14]The complete name of the variables and their calculations are listed in appendix 1

[15]Fannie Mae illustrate the connection between mortgage loss and certain variables over time (http://www.fanniemae.com/resources/file/fundmarket/pdf/webinar-102.pdf)

Sometime, a significant decline in housing prices lead to borrower negative equity where the value of house is less than borrower mortgage debt. OFHEO and FGW capture this by adding a measure of the probability that a borrower is currently in a position of negative equity. This process requires dynamic loan information since it is created until terminated. An easier way is updating loan balance and home price inputs to create a new LTV (MLTV) that reflects amortization and home price change to-date[16]. For this exercise, we used FHFA?s recently published 3-Digit-Zip Home Price Index (HPI). Figure 4 shows a significant higher MLTV for the loans that have been defaulted; specially during financial crisis. FGW (2015) considers borrower credit score (FICO) as an expansionary variable for loan default. Fig 5 displays that a lower credit score makes a loan default more likely. Also, we consider DTI as an expansionary variable for loan defaults. The DTI can be an indicator of the borrower?s income, and loans with higher DTI are more likely to default. Although, figure 6 shows a positive relationship between DTI and default but, this connection is not as strong as other variables. Gyourko and Tracy (2013) show there is strong correlation between household-level unemployment rates and defaults, however this relationship is weak for aggregate unemployment. We use county-level unemployment rates from the Bureau of Labor Statistics to add local unemployment change as an expansionary variable. Figure 7 shows change in unemployment over time in connection with defaults. Finally, slope of the yield curve (Slp_Yld), which is calculated as the difference between the ten-year Constant Maturity Treasury yield to the one-year Constant Maturity Treasury yield can influence expectations of the future levels of interest rates. Figure 8 shows the connection between slope of yield curve and default during time.

3.1 Using Smote function to balance data

One issue with the Fannie Mae loan level dataset is highly unbalanced distribution of the two classes default and non-default. 1644452 out of 1689199 observations (97%) [17] defined as non-default while just 3% of the data is assigned to class 1(defaulted). In this case the classifiers won't be able to recognize minor classes and are influenced by major classes. For example, In a logistic regression the conditional probability of minor classes are underestimated (King and Zeng, 2001) and Tree based classifiers, and KNN yield high recall but low sensitivity when the data set is extremely unbalanced (Cieslak and Chawla, 2008). Before fitting the model over the training dataset

[16]The calculation of MLTV is in appendix one

[17]Total of observations are around twenty millions loan level data. After cleaning the data and 10% sampling for each year, we work on 1689199 loan level observations.

and forecast classes over the testing dataset, we should balance the data. There are different methods to balance the data such as oversampling , under-sampling , and Synthetic Minority Oversampling Technique (SMOTE) proposed by Chawla et al. (2002). Oversampling methods replicate the observations from the minority class to balance the data. However, adding the same observation to the original data causes overfitting, where the training accuracy is high but forecast accuracy over testing data is low. Conversely, the under-sampling methods remove the majority of classes to balance data. Obviously, removing observations causes the training data to lose useful information pertaining to the majority class. SMOTE finds random points within nearest neighbors of each minor observation and by boosting methods generates new minor observations. Since the new data are not the same as the existing data the overfitting problem won't be an issue anymore, and we won't loose the information as much as with the under-sampling methods. For these reasons, this study considers the SMOTE function to balance the data.

3.2 Measure of Forecast Accuracy

To compare different classification methods, we should find an appropriate measure for forecast accuracy. In a classification problem, appropriate performance measures may be obtained from a confusion matrix [18] which displays predicted classes versus true classes (Table 2). A common performance measure are test error rates. The test error rate is the referring loans that do not default to the default class or defaulted loans that are incorrectly assigned to non-defaulted loans. Base on confusion matrix 2 error rate can be defined as $(FP+FN)/(TN+TP+FP+FN)$, however, error rates are highly dependent on changes in data and return ambiguous results. Moreover, the performance of different models can be compared by sensitivity and specificity of classifiers over the test data set. [19]. Note that class-specific performances vary with a change in the threshold ratio, and there is a tradeoff between specificity and sensitivity when the threshold ratio goes up or goes down. If $Pr(y_i = 1)$ is relatively large there is a high probability that loan i belongs to default classes ,but we need to determine the thresholds. For this reason, this study considers one of the widely used measures of forecast accuracy known as the Receiver Operating Characteristics (ROC) curve. The true positive rate (sensitivity of the classifier) is then plotted versus the false positive rate (1 - specificity of the classifier) for each classification

[18] Some literatures use term "classification table" instead Confusion matrix

[19] the sensitivity define as percentage of true defaulters and specifying specify as a percentage of non-defaulters that recognize correctly. Therefore, error type 1 is (1- specifying) and error type 2 is (1-sensitivity)

threshold. If the ROC curve is closer to the top left, the classifier performs better and the area under roc curve (AUC value) is larger. A completely random guess yields AUC 50% which is a point a_ long of the diagonal line from the bottom left to the top right corners of ROC curve while a perfect classifier would yield a point in the upper left corner of the ROC space, representing 100 % sensitivity (all true positives are found) and 100% specificity (no false positives are found). This study applies AUC to compare the forecast accuracy of different classifiers.

Table 2: Confusion Matrix

	True Default	
Predicted Default	True Negative False Negative	False Positive True Positive

4 Results

We now turn to present the results of model evaluation. First, we focus on the data that we split into three parts: before financial crisis (2002-2006),during financial crisis (2007-2011), and after financial crisis (2012-2017). In each period, we split the sample to the training and testing datasets randomly, balance both training and testing observations, and finally normalize the data to have mean zero and variance one[20]. We use training observations to fit certain model and adopt fitted parameters to forecast testing observation.

Table 3 displays forecast accuracy measures for the different models during different time horizons. The first row of table 3 is AUC for the logit model. Forecast accuracy of the logistic regression is around 85% all the time. However, AUC value for KNN classifiers is higher than logistic regression all the time but it is decreasing : 90% before financial crisis, 89% during financial crisis, and 87% after financial crisis. Note k inside the parenthesis is the optimal k which is earned by cross validation. The AUC value for SVM and RF models are almost similar to each other, higher than logistic regressions, and lower than KNN models before and during financial crisis. Note, SVM classifiers adopts a radial basis kernel function (RBF). We try other kernel functions which obtain almost the same results. . [21]. Accuracy forecast in FM models dominates logistic regressions and other machine learning classifiers:

[20]Note that the order of this operations are important. For example, we won't get an accurate results if the data was balanced then split to training and testing observations.

[21]We don't report AUC value for the linear SVM models since it is similar to logistic regressions

91% AUC value before financial crisis, 89% during financial crisis, and 88% after financial crisis. [22]. FM models include the interaction of the all variables and this can be one reason for high forecast accuracy of this models.

Figure 9 presents ROC curve for the classifiers where three graphs are for three different time horizons. [23] In a ROC curve, the horizontal axes is the true positive rate (Sensitivity) and the vertical axes is the false positive rate (1-Specificity) for different threshold points of a parameters. Thus, If the curve is closer to the top left then the accuracy of the forecast is higher. According to the ROC curve, FM models have the highest accuracy and the logistic models have the lowest one in all three different time horizons. One of the important results of this study is to determine the important features. Figure 10 ,11, and 12 show important variables in each period of the time. The loan age is the most important variable before and after financial crisis, however, MLTV is the most effective variable in failing a mortgage loan during financial crisis. Generally, the loan age, MLTV, credit score (CSCORE), and the ratio of loan to value(OLTV) are the top four important features to predict loan default.

Moreover, we construct an out of sample forecast with three year windows and predict loan default one year ahead. The result is displayed in figure 13 where the horizontal axes is the time and the vertical axes is the AUC value. FM forecast accuracy dominates other classifiers and logistic regressions over time. machine learning methods (RF, SVM, and KNN) perform better than logistic regressions but their forecast accuracy are lower than FM models.

5 Conclusion

After financial crisis (in 2009) supervisory authorities have consider stress testing a central part of their supervisory regimes to insure health of big and complex financial institutions. However, in designing stress tests, authorities need to explore models, stress scenarios, and data. If forecast models do not accurately reflect all possible outcomes, it can lead to a failed stress test.

One of the supervisory authorities, Office of Federal Housing Enterprise Oversight (OFHEO), was charged with ensuring the capital adequacy, financial safety, and soundness of Fannie Mae and Freddie Mac. However, the Federal Housing Finance

[22]Note, that we can also present other accuracy measures such as Recall, Precision, and F1 score. But the value of these measures are almost same as AUC value. The reason is that we balance the data before fit and forecast the models

[23]Remind that the AUC value is the area under ROC curve.

Agency (FHFA) put Fannie Mae and Freddie Mac into conservatorship in the summer of 2008.

This studies? focus is to challenge the models that are used by OFHEO to forecast mortgage loan defaults. More specificity, OFHEO and some other studies use logistic regression to forecast mortgage loan defaults, and prepayments. In this study, we use machine learning methods to forecast mortgage loan defaults. we discuss about the most commonly used supervised learning methods such as KNN,SVM,RF, and FM and apply them on Fannie Mae single family loan performances dataset from 2001 to 2016 to predict mortgage loan defaults.

First, we divide the data to three parts: before financial crisis (2002-2006), during financial crisis (2007-2011), and after financial crisis (2012-2017). In each period the sample is split to training observation to fit the models and testing observations to forecast new observations. The results confirm that applying machine learning methods yields better forecast accuracy than traditional classifier methods such as logistic regression. AUC value using logistic regression is 85% on average, however, this value is 88% on average for the KNN,SVM, RF, and it is around 90% for the FM model. Moreover, we construct an out of sample forecast with rolling windows three years ahead and forecast loans default one year ahead.

The results, verify that machine learning classifiers yield higher forecast accuracy than logistic regressions. Finally, we can recognize the importance of the variables using machine learning algorithms. The loan age is the most important variable before and after financial crisis, however, MLTV is the most effective variable in failing a mortgage loan during financial crisis. Generally, the loan age, MLTV, credit score (CSCORE), and LTV are the top four important features to predict loan defaults.

References

[1] Amir E. Khandani, Adlar J. Kim, and Andrew W. Lo , 2010*Consumer Credit Risk Models via Machine-Learning Algorithms*. Journal of Banking & Finance 34 : 2767-2787.

[2] Chiranjit Chakraborty and Andreas Joseph, 2017 *Machine learning at central banks*. Bank of England, Staff Working Paper No. 674

[3] Diego Alejandro Salazar, Jorge Ivn Vlez, Juan Carlos Salazar, 2012*Comparison between SVM and Logistic Regression: Which One is Better to Discriminate?*. Revista Colombiana de Estadstica, Junio , volumen 35, no. 2, pp. 223 a 237

[4] David A.Cieslak, and Nitesh V.Chawla. 2008. *Learning Decision Trees for Unbalanced Data* ECML PKDD '08 Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases

[5] Gareth James , Daniela Witten , Trevor Hastie, and Robert Tibshirani 2013 *An Introduction to Statistical Learning*. Springer New York Heidelberg Dordrecht London, ISSN 1431-875X

[6] Gyourko, Joseph, and Joseph Tracy. 2013. *Unemployment and Unobserved Credit Risk in the FHA Single Family Mortgage Insurance Fund.* Technical Report. National Bureau of Economic Research 18880.

[7] Einav, L. and Levin, J. D., 2013 *The data revolution and economic analysis* . Working Paper 19035, National Bureau of Economic Research.

[8] King, G. and Zeng, L. 2001. *Logistic regression in rare events data* Political Analysis, Vol. 9, pp.137163.

[9] Ioakeim Perros and Jimeng Sun, 2015 *Factorization Machines as a Tool for Healthcare*. TGeorgia Institute of Technology, Atlanta,

[10] Margaret Ryznar,Frank Sensenbrenner, and Michael Jacobs, Jr. ,2013 *Implementing Dodd-Frank Act Stress Testing* . Depaul University Business and commercial

[11] Michael Jacobs Jr.,Ahmet K. Karagozoglu and Frank J. Sensenbrenner, 2017 *Stress testing and model validation: application of the Bayesian approach to a credit risk portfolio.*Journal of Risk Model Validation 9

[12] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer 2002. *SMOTE: Synthetic Minority Over-sampling Technique* Journal of Artificial Intelligence Research 16 , 321357

[13] Powers, D. 2011. *Evaluation: From precision, recall and f-measure to ROC., informedness, markedness & correlation.* Journal of Machine Learning Technologies, 2(1):37?63.

[14] Pedregosa, F. e. a. 2011. *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, 12:28252830.

[15] Steffen Rendle, 2010 *Factorization Machines*. Proceedings of the 2010 IEEE International Conference on Data Mining. pp. 995-1000. ICDM '10 (2010)

[16] Steffen Rendle, 2012 *Factorization Machines with libFM*. ACM Transactions on Intelligent Systems and Technology (TIST) ,Volume 3 Issue 3.

[17] Smola, A. J. and Scholkopf, B. 2004. *A tutorial on support vector regression.* Statistics and Computing, 14(3):199?222.

[18] Varian, H. ,2014 *Big data: New tricks for econometrics* . Journal of Economic Perspectives, 28(2):328.

[19] W. Scott Frame, Kristopher Gerardi, and Paul S. Willen, 2015 *The Failure of Supervisory Stress Testing: Fannie Mae, Freddie Mac, and OFHEO*. Federal Reserve Bank of Atlanta, Working Paper Series.

[20] 2002 Annual Housing Activities Report (AHAR)

6 Appendix

6.1 Data

External Economic Data	Raw Variables	Macroeconomics Variables
loan age	MLTV	Unemployment Rate
loan purpose	Spread	Slope of Yield Curve
FICO	Loan Size	
DTI		
OLTV		
Number of Borrower		

Mortgage Premium (Spread) = (Current Interest Rate - Market Mortgage Rate)/Current Interest Rate
Loan Size = Original un-payment balance (UPB) / Average UPB in the same state and same date
Market Loan to Value (MLTV) = (Current UPB/HPI Factor) * Original Housing Value
Where HPI Factor = Current HPI / Original HPI

Table 3: AUC for Classifires

	Time		
	01-06	07-11	12-16
MODEL			
LOGIT	0.85	0.85	0.85
KNN	0.90 (k=30)	0.89(k=12)	0.87(k=14)
RF	0.88	0.87	0.87
SVM	0.88	0.87	0.86
FM	0.91	0.89	0.88

Figure 3: The relationship between OLTV and loan default

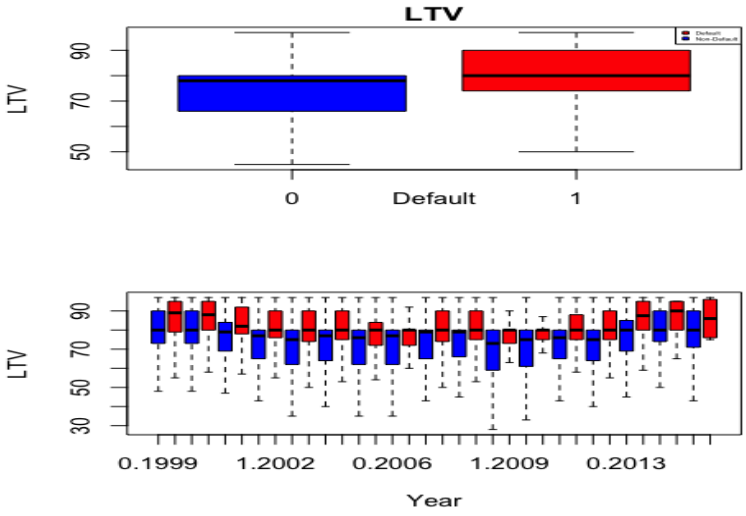


Figure 4: The relationship between MLTV and loan default

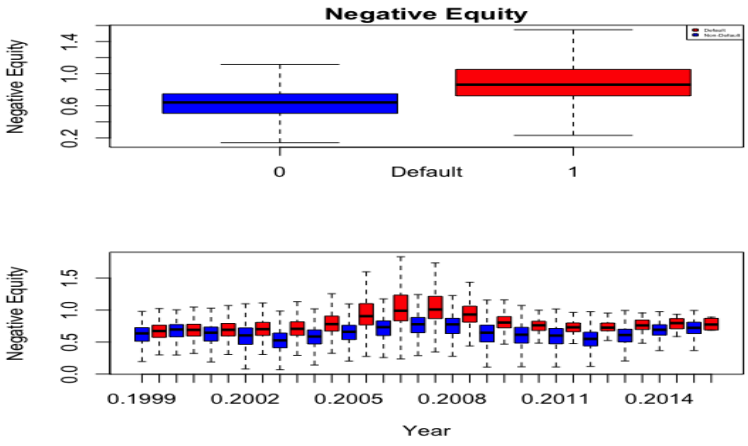


Figure 5: The relationship between DTI and loan default

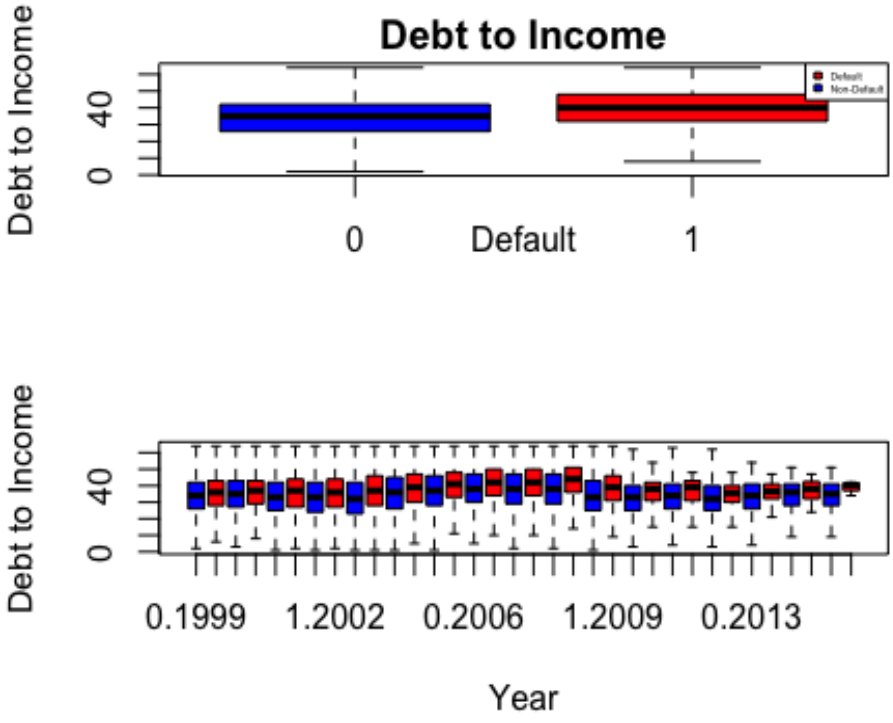


Figure 6: The relationship between MLTV and loan default

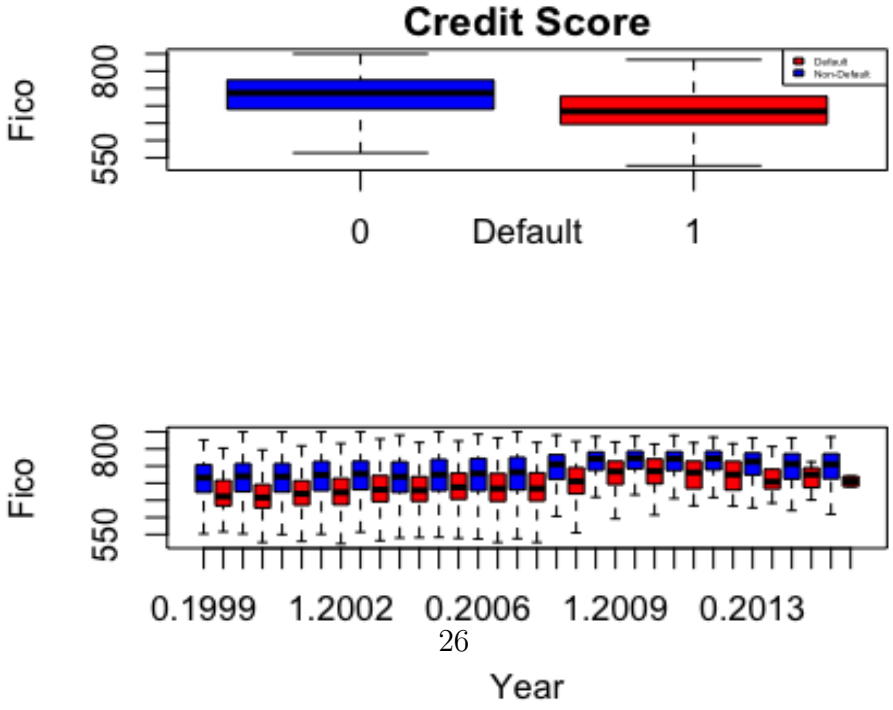


Figure 7: The relationship between change in unemployment and loan default

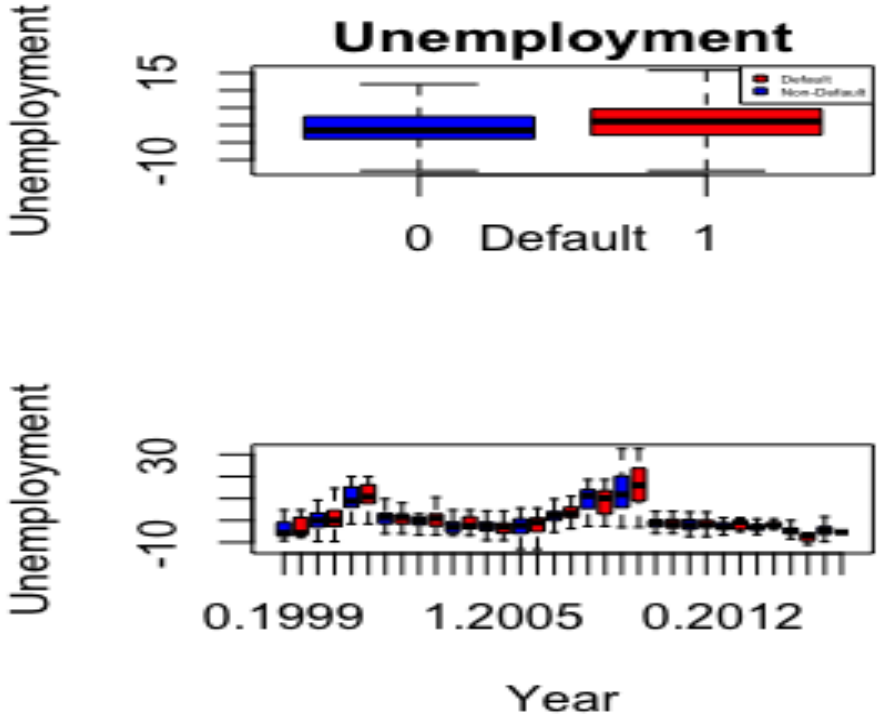


Figure 8: The relationship between slope of yileld and loan default

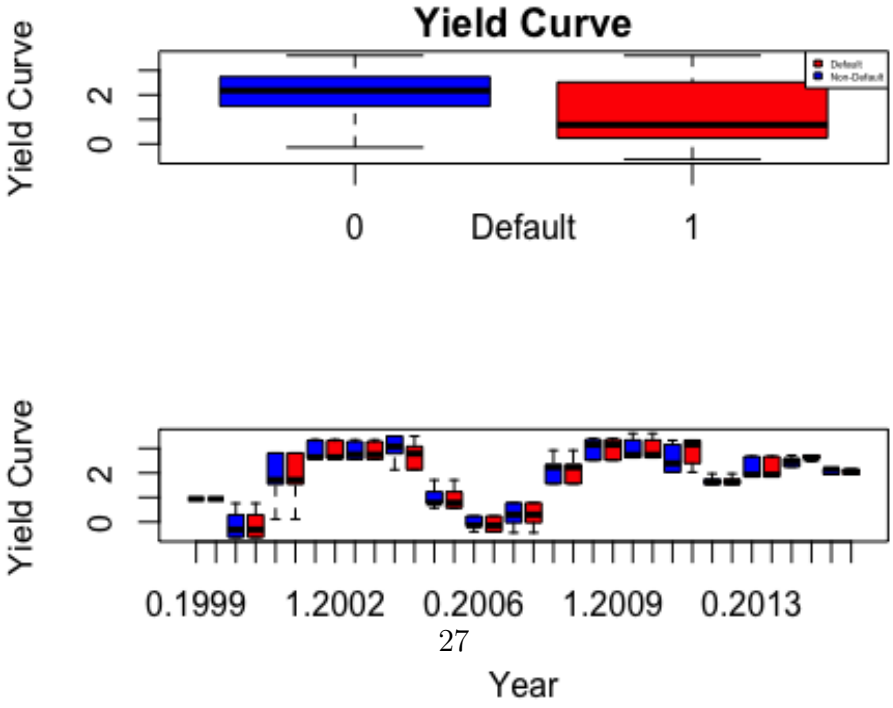


Figure 9: ROC Curve

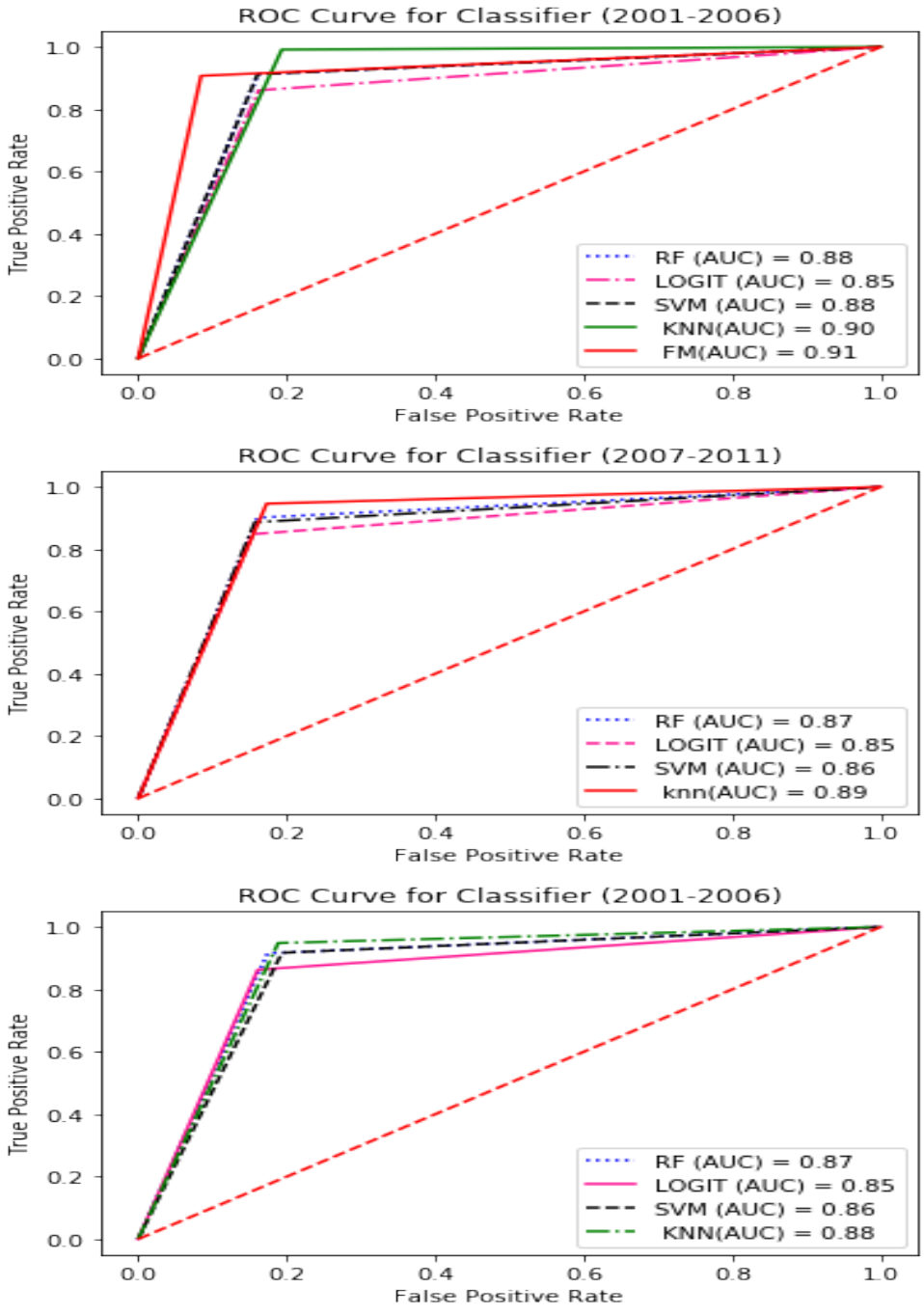


Figure 10: Importance of variables (2001-2006)

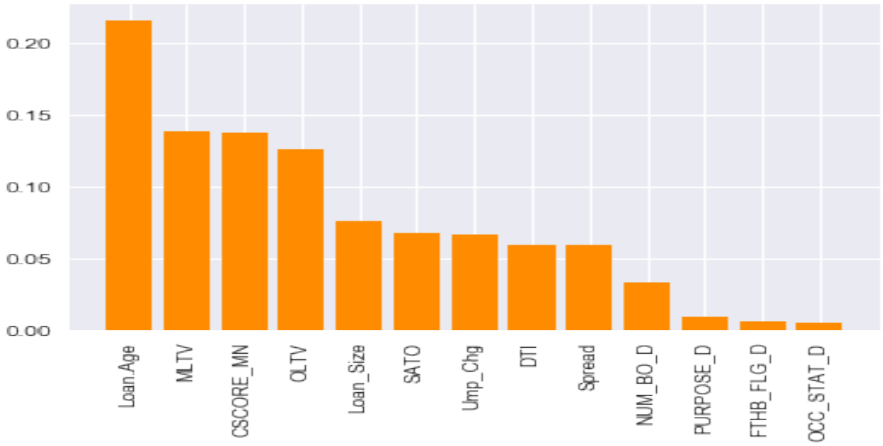


Figure 11: Importance of variables (2006-2011)

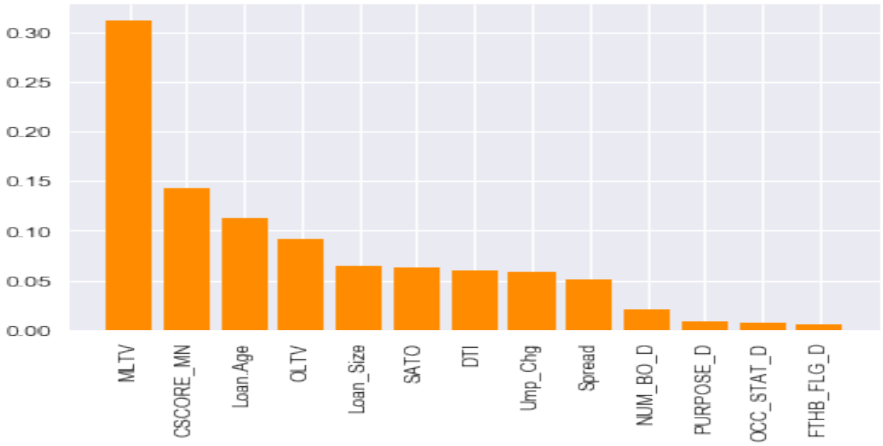


Figure 12: Importance of variables (2012-2016)

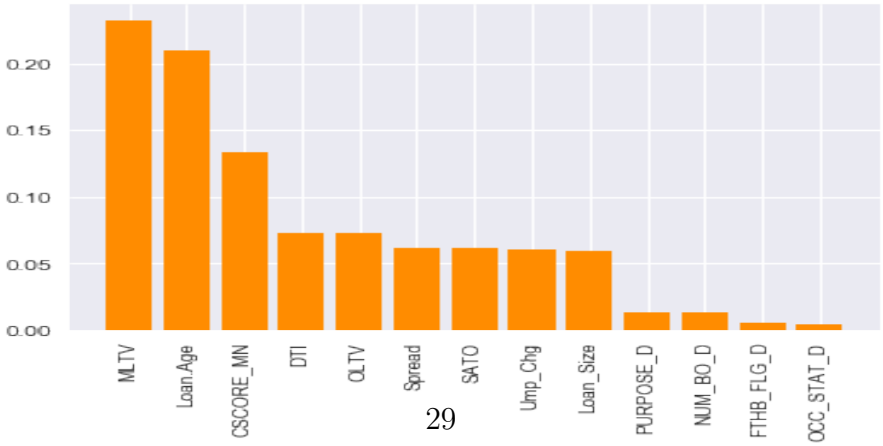


Figure 13: AUC value for classifiers over time