Financial Crisis Prediction: A Model Comparison

Daniel Fricke[a,b,c]

[a] *University College London, Financial Computing & Analytics*
[b] *London School of Economics & Political Science, Systemic Risk Centre*
[c] *University of Oxford, Saïd Business School, CABDyN Complexity Center*

November 29, 2017

Abstract

In this paper we compare different models for financial crisis prediction, focusing on methods from the field of Machine Learning (ML). These methods are particularly promising, since they were specifically designed for making predictions. In our application, we find that the performance on these methods depends on whether we look at in-sample or out-of-sample predictions. In the latter case, they do not always outperform more traditional approaches (such as Logistic regressions). Nevertheless, we find that these methods can be useful and should therefore become a standard element in the toolbox of empirical researchers.

Keywords: financial crisis; prediction; classification; machine learning.
JEL classification: C38; C53; G01

1. Introduction

Financial crises are costly and have a long-term impact on the real economy. For example, Laeven and Valencia (2007) find that cumulative output losses during crisis periods amount to 15-20% on average of annual GDP, and fiscal costs associated with crisis management average about 13% of annual GDP. Hence, having an accurate estimate of the probability of a financial crisis in real-time would allow taking precautionary action in advance and potentially mitigate its effects. As mentioned by Schularick and Taylor (2012), previous attempts of predicting financial crises were generally hampered by data availability. To overcome this issue, they gathered a long panel dataset and find that (1) financial crises appear to be predictable based on simple Logistic regressions, and (2) the most important determinant for a financial crisis is previous credit growth.

In this paper, we compare different Machine Learning (ML) models in order to explore to what extent we can improve upon the forecasting accuracy of the baseline specification of Schularick and Taylor (2012). ML methods are particularly relevant for our purposes as they are specifically designed to make accurate predictions and have shown impressive successes in different fields (see Hastie et al. (2011); Efron and Hastie (2016)). Given the growing availability of large economic datasets, a number of prominent economists have advocated

Email address: d.fricke@ucl.ac.uk (Daniel Fricke)

the use of ML methods (see, e.g., Varian (2014)). To date, however, these methods are still not necessarily part of the standard toolbox of empirical researchers.[1] As such, this paper illustrates the potential usefulness of several popular supervised ML methods in a specific application.

Anticipating our main results, we find that the performance of ML methods heavily depends on whether we look at in-sample or out-of-sample predictions. In the first case, several ML methods allow to make perfect in-sample predictions. In the second case, however, the results are much less impressive - in fact, in many cases the out-of-sample performance of traditional Logistic regressions is superior to that of various ML methods. Lastly, using a standard ML approach, we confirm that credit growth is the most important determinant of a financial crisis.

The remainder of this paper is organized as follows: section 2 briefly describes the underlying dataset and introduces the different ML models of interest. Section 3 presents the empirical analysis. Section 4 discusses the results.

2. Data and ML Methods

2.1. Data

Our panel dataset comes from Schularick and Taylor (2012), covering $T = 139$ years (1870–2008) and $N = 14$ developed countries.[2] We use exactly the same data filters as the original study (e.g., excluding data from the two world wars) and our main variable of interest is $y_{t,i}$, which takes a value of 1 if we observe a financial crisis in year t in country i (0 otherwise). We are interested in predicting the probability of a crisis using a set of lagged explanatory variables $x_{t,i}$. For the sake of comparability, our application focuses on the baseline specification of Schularick and Taylor (2012), which uses $M = 5$ lags of (real) credit growth only.[3] We should note that the dataset is highly imbalanced, since the vast majority of observations are $y_{t,i} = 0$ (no crisis). In total, we end up with 1,253 observations, of which there are 53 crisis periods (4% over the full sample). This infrequency of financial crises poses a challenge to all of the methods considered below.

2.2. ML Methods

In the jargon of machine learning we are using *supervised learning* models for binary classification: we know the observed outcome (i.e., $y_{t,i} = 0$ or 1) and want to assess how well we can predict these values using a variety of models. In the following, we briefly sketch the basics of 7 different models (details can be found in Hastie et al. (2011), and Efron and Hastie (2016)).

[1] An important reason is that asymptotic results are often missing for ML models, which is critical because economists are often more interested in the sign/significance levels of individual parameters, rather than predictive accuracy (Breiman (2001); Einav and Levin (2014)).

[2] The dataset is available on the AEA website (`https://www.aeaweb.org/aer/data/april2012/20091267_data.zip`). The countries included are: Australia, Canada, Denmark, France, Germany, Italy, Japan, Netherlands, Norway, Spain, Sweden, Switzerland, UK, and US.

[3] Schularick and Taylor (2012) also include country fixed effects in some specifications, but these are reported to be insignificant in most instances.

Logistic Regression. Given that the Logistic regression is a standard method in empirical economic research, we keep our presentation very short here. We are interested in the conditional probability of observing a financial crisis in country i in year t, which we denote as $p_{t,i} = \mathrm{Prob}(y_{t,i} = 1|X = x_{t,i})$, where $x_{t,i}$ is a vector of size M (lagged credit growth in our case). Logistic regression specifies this as

$$p_{t,i} = \frac{1}{1 + \exp(-x_{t,i}\beta)}, \tag{1}$$

where β is the corresponding parameter vector (estimated via maximum likelihood).

Classification Trees (C.Tree) and Forests (C.Forest). The basic idea of tree-based methods is to stratify (or segment) the predictor space into a small number of J distinct and non-overlapping rectangles or boxes $(R_1, R_2, \cdots, R_J)$. The prediction for a given observation, $\hat{y}_{t,i}$, is then simply the mean of those observations in the region to which it belongs (in our case, the class proportions for each region). The set of splitting rules used to segment the predictor space can then be easily visualized as a tree.

Given that finding the optimal tree is NP-complete, the standard approach is to use a greedy algorithm (*recursive binary splitting*): we start at the top of the tree (no branches), and then successively split the predictor space into branches, using the best possible split at each point. For this purpose, we need to define an objective function that quantifies the reduction in error at each step and a common choice is the Gini-Index,

$$Gini_j = 2\hat{y}_j(1 - \hat{y}_j), \tag{2}$$

where $\hat{y}_j$ is the predicted probability of a crisis for observations falling into region j. The algorithm stops when a certain convergence criterion is reached.

An important issue is that trees are prone to overfitting - they yield volatile predictions since they tend to look very different for different subsamples. Forests reduce the variance of individual trees by combining predictions of $B > 1$ (biased) trees.[4] For a given observation, $y_{t,i}$, we can then check the predicted value in each of the regression trees, $\hat{y}_{t,i}^{(b)}$, and get a weighted consensus estimate

$$\hat{y}_{t,i} = \sum_{b}^{B} \alpha^{(b)} \hat{y}_{t,i}^{(b)}, \tag{3}$$

which typically has lower variance compared to the individual predictions. Efficient algorithms for finding the optimal weights, α, exist. Here we use *AdaBoost1*, which is one of the most popular approaches.[5] In everything that follows, we set $B = 100$.

[4] The most basic approach ("Bagging") consists of a bootstrapping exercise. In this case, we generate a total number of B synthetic datasets from the original dataset (with replacement) and then fit a tree separately to each dataset. Random forests are a special case of Bagging (allowing at most $m < M$ for splitting at each step).

[5] A detailed description of the AdaBoost1-algorithm is beyond the scope of this paper. Intuitively, the algorithm emphasizes those models that are able to classify observations that are hard to predict.

K-Nearest Neighbors (KNN). KNN is a nonparametric method that can be used for both classification and regression. The input consists of the K closest observations in terms of the corresponding features, x, where 'closeness' is typically assessed via Euclidean distance (as we do here). In a classification setup, the output is class membership: each object is classified according to the most frequent value of y among its K neighbors. Note that, if $K = 1$, each observation is simply assigned to the class of its single nearest neighbor. For the sake of comparison, we use both $K = 1$ (KNN-1) and $K = 3$ (KNN-3) in our empirical application below.[6]

Neural Networks (NN). NNs are parametric models originally inspired by the physiology of nerve cells. For a single hidden-layer NN with a number of H hidden units, we can write the predicted outcome for a given observation as,

$$\hat{y}_{t,i} = g\left(\alpha_0 + \sum_{h=1}^{H} \alpha_h \cdot f\left(\beta_{0,h} + x_{t,i}\beta_h\right)\right), \tag{4}$$

where $f(\cdot)$ and $g(\cdot)$ are arbitrary activation functions. Note that each hidden unit h uses the same original activation function $f(\cdot)$, but possibly different weights β. The case of multilayer NNs is a straightforward extension of Eq. (4), but in this paper we restrict ourselves to single-layer NNs in order to keep the number of parameters reasonably small.[7] These can be estimated using nonlinear least squares. In our empirical application $f(\cdot) = g(\cdot)$ are sigmoids and we show results for both $H = 3$ and $H = 5$ (NN-3 and NN-5, respectively).[8]

Quadratic Discriminant Analysis (QDA). QDA is a popular classification approach that makes use of Bayes' theorem. It is related to the Logistic regression approach in that it also models the conditional probability $\text{Prob}(y_{t,i} = 1|X = x_{t,i})$ by modelling $\text{Prob}(x_{t,i}|Y = y_{t,i})$ separately in both classes, and then plugging these estimates into Bayes' theorem to get the conditional probability of interest. QDA assumes that the observations from each class are drawn from a multivariate Gaussian distribution, i.e., $x_{t,i}$ follows either $N(\mu_0, \Sigma_0)$ or $N(\mu_1, \Sigma_1)$. For the general case of c classes, we assign a given observation to the class for which it maximizes

$$\delta_c(x) = -\frac{1}{2}(x - \mu_c)^T \Sigma_c^{-1} (x - \mu_c) + \log \pi_c, \tag{5}$$

where π_k is the unconditional sample probability of class k (0 or 1).

Support Vector Machines (SVM). SVMs aim to find hyperplanes that separate the data into two classes. In $\mathcal{R}^M$ a hyperplane is a flat affine subspace in $\mathcal{R}^{M-1}$ defined by the equation

$$\beta_0 + x\beta = 0, \tag{6}$$

[6] In principle, we could find the best value of K using cross-validation (see below).

[7] This is not as restrictive as it might seem: even single-layer NNs are *universal approximators* since they are able to fit any nonlinear function to an arbitrary degree of accuracy for suitable H.

[8] As for the KNN approach, parameter H could be chosen via cross-validation.

and each x for which this holds is a point on the hyperplane. As such, hyperplanes can be for classification when the data are perfectly separable; however, in this case an infinite number of hyperplanes exists and the *Maximal Margin Classifier* (MMC) chooses the separating hyperplane with the largest margin. Technically, it solves

$$\max_{\beta_0,\beta} \mathrm{D} \quad \text{subject to} \tag{7}$$

$$\beta^T\beta = 1, \; y_{t,i}(\beta_0 + x_{t,i}\beta) \geq D \; \forall \; t, i,$$

which identifies the *support vectors* (the observations closest to the hyperplane) for each hyperplane, and then chooses the one for which this distance (D) is maximal.

SVMs are a useful practical extension of the MMC: they (1) do not require perfect separability (some observations are allowed to be on the wrong side of the hyperplane), and (2) allow for nonlinear decision boundaries. The latter is achieved by including functions of the M original predictors (such as quadratic or cubic terms) in the definition of the hyperplane. The most popular approach to enhance the feature space is the so-called radial kernel, which the one we use in this paper.

2.3. Model Accuracy

We use the *Area Under Curve* (AUC) of the standard *Receiver Operating Curve* (ROC) as performance measure. (The results are qualitatively similar when using Brior scores.) For a given set of predictions we calculate the corresponding false positive (FPR) and true positive rates (TPR) for different thresholds $\theta \in [0, 1]$. The ROC plots one against the other and will always cross the two points (1,1) and (0,0) for $\theta = 0$ and $\theta = 1$, respectively. The ROC of a completely random prediction will be a linear function between these two points, and useful models should display an ROC above this benchmark. The AUC is defined as the area under the ROC and a completely random model yields a value of 0.5.

3. Financial Crisis Prediction

In the following we perform three different exercises: first, we look at the performance for in-sample predictions. One might argue that, given the usage of lagged explanatory variables/features, this approach is not strictly in-sample in the usual sense. However, given that we do not separate the data into training and test sets in this case, the predictive accuracies from this exercise can be misleading. In our second exercise, we therefore compare the out-of-sample performance. Third, we generate a large number of potentially important explanatory variables and identify the most important ones using the C.Forest model.

3.1. In-Sample Performance

Table 1 shows the in-sample AUC for the different models and different sample periods. The results for the Logistic regression are in line with those shown in Schularick and Taylor (2012). The first column corresponds to the full sample, and the other two columns to the pre- and post-World War II data. The performance of the different models and their

In-Sample Performance (AUC)

	Sample		
Model	Full Sample	Pre-WW2	Post-WW2
Benchmarks			
Logistic	0.656	0.719	0.687
Random	0.500	0.500	0.500
ML			
C.Forest	0.817	0.948	0.928
C.Tree	0.982	0.986	0.992
KNN-1	1.000	1.000	1.000
KNN-3	0.957	0.950	0.964
NN-3	0.820	0.899	0.948
NN-5	0.866	0.977	0.955
QDA	0.701	0.767	0.826
SVM	0.949	0.963	0.991

Table 1: In-sample AUC of the different models and different sample periods. Details on the models can be found in the main text. The four best methods are highlighted as gray cells, with darker colors corresponding to higher scores.

rankings are largely consistent over the different samples: all prediction models are better than random (AUC> 0.5) and ML methods perform substantially better than the Logistic model. KNN-1 and C.Tree are generally the best models (with the former yielding an AUC of 1, i.e., *perfect* predictions), followed by SVM and KNN-3.[9] These results should be treated with caution, however, since KNN-1 and C.Tree are exactly those models that are prone to overfitting.[10]

3.2. Out-of-Sample Performance

We now focus on the out-of-sample performance. The typical approach is *K-fold cross-validation*: one splits the dataset into K equally sized blocks, and for each block trains a model using data from all other blocks only. For each test set, the performance of the trained model is evaluated by calculating the corresponding AUC. One obvious issue with this approach is that it ignores that we are dealing with time-series predictions here: we do not want to train a model for predicting a financial crisis in the 1960s using data from the 1990s. Therefore, we mainly focus on another validation approach, which again splits the data into K equally-sized blocks, but trains each model using information on *previous blocks only* (see Figure 1 for an illustration with $K = 5$).

Table 2 shows the average out-of-sample AUC (standard deviations in brackets) for the different models and different validation approaches. Columns 1-4 use the above-mentioned validation approach for different K, and the last column uses 5-fold cross-validation. As

[9] The results are largely consistent when including country fixed effects. The main difference is that NNs and SVM perform substantially better than before. Details available upon request from the author.

[10] In principle, we could also deal with the issue of overfitting using parameter regularization (i.e., Lasso, Ridge, or Elastic Nets). For the sake of brevity we leave this extension for future research.

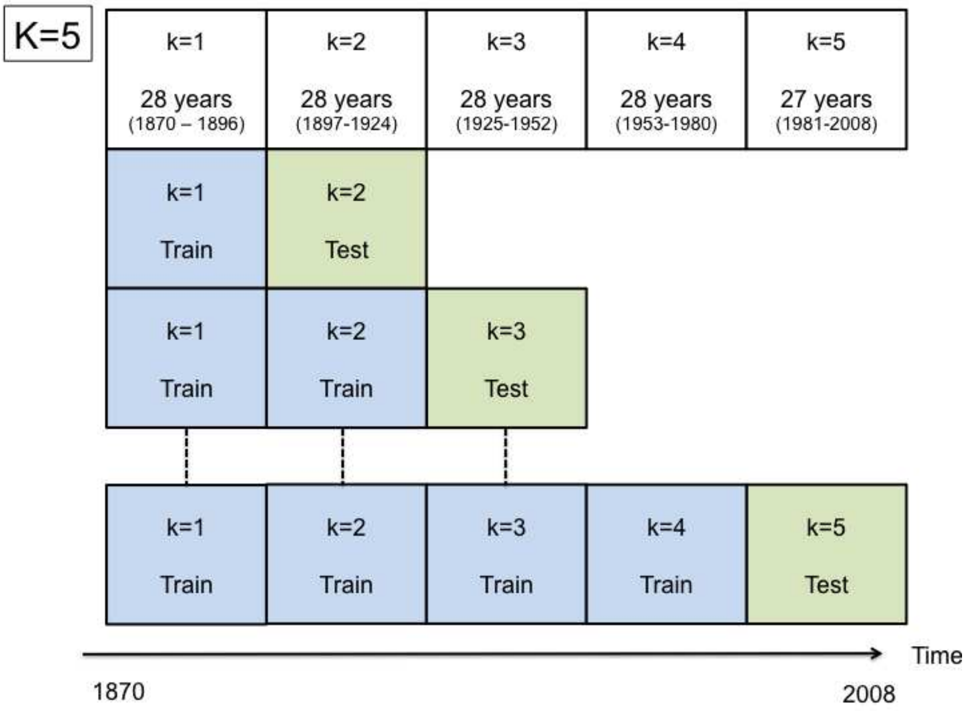

Figure 1: Illustration of the out-of-sample validation approach for the case $K = 5$.

expected, the AUCs of all prediction models are substantially lower compared to the in-sample values (in many cases even below 0.5). Moreover, models performing well in-sample tend to perform relatively poorly out-of-sample, suggesting that overfitting is indeed an issue. Interestingly, the relatively simplistic Logistic model is often among the best out-of-sample prediction models, so more complex models may not always improve the predictive accuracy.

Out-of-Sample Performance (AUC)

	Validation Approach				5-fold
Model	K = 2	K = 3	K = 4	K = 5	cross-val.
Benchmarks					
Logistic	0.657	0.645	0.670	0.668	0.627
	(-)	(0.047)	(0.089)	(0.082)	(0.089)
Random	0.500	0.500	0.500	0.500	0.500
ML					
C.Forest	0.608	0.580	0.703	0.655	0.532
	(-)	(0.049)	(0.216)	(0.109)	(0.094)
C.Tree	0.503	0.492	0.440	0.461	0.501
	(-)	(0.052)	(0.153)	(0.130)	(0.076)
KNN-1	0.475	0.483	0.479	0.539	0.514
	(-)	(0.004)	(0.032)	(0.105)	(0.041)
KNN-3	0.519	0.496	0.611	0.588	0.504
	(-)	(0.035)	(0.207)	(0.183)	(0.056)
NN-3	0.636	0.533	0.608	0.671	0.570
	(-)	(0.068)	(0.045)	(0.152)	(0.082)
NN-5	0.665	0.472	0.644	0.590	0.546
	(-)	(0.231)	(0.242)	(0.217)	(0.111)
QDA	0.577	0.573	0.639	0.610	0.603
	(-)	(0.000)	(0.129)	(0.167)	(0.089)
SVM	0.547	0.431	0.646	0.498	0.595
	(-)	(0.095)	(0.203)	(0.241)	(0.097)

Table 2: Average out-of-sample AUC (standard deviation in brackets) of the different models and different validation approaches. Details on the models and validation approaches can be found in the main text. The four best methods are highlighted as gray cells, with darker colors corresponding to higher scores.

3.3. Feature Selection and Variable Importance

Schularick and Taylor (2012) find that credit growth is the single most important determinant of a financial crisis. In order to explore to what extent a standard ML approach provides similar results, we generate a total of $M = 35$, possibly correlated, explanatory variables (see Table 3), and aim to assess their relative importance by re-applying the C.Forest model. We calculate our variable importance measure as follows: for each feature, we quantify how much it typically reduces the overall classification error when it is serves as a branch. In order to make these values comparable, we show the relative variable importance (most important variable = 1).

Variable Importance Exercise: Features of Interest

5 lags: BroadMoneyGrowth (real),	5 lags: ΔInterestRate (short-term),	5 lags: StockGrowth (real).
5 lags: CreditGrowth (real),	5 lags: Inflation,	
5 lags: Crisis,	5 lags: NarrowMoneyGrowth (real),	

Table 3: Set of $M = 35$ features used for variable importance exercise.

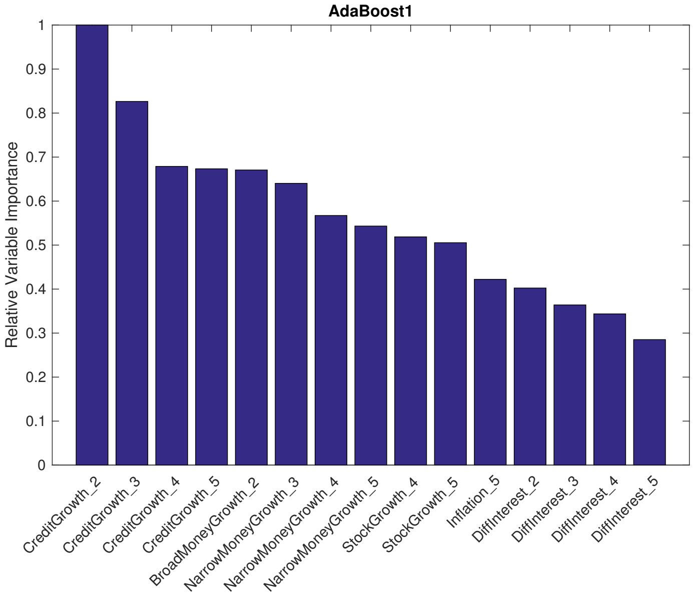

Figure 2: Relative variable importance. Results based on C.Forest model using the features in Table 3.

Figure 2 shows the results for the 15 most important features. In line with Schularick and Taylor (2012) we find that the second lag of credit growth is the single most important feature for financial crisis prediction. Interestingly, despite being insignificant in many specifications of the original paper, lags 3 to 5 appear to be relevant features as well in our application.

4. Discussion

Financial crisis prediction is of utmost importance for various stakeholders. Here we showed that ML methods can be useful for different prediction problems, but may not always outperform more traditional approaches (such as Logistic regressions). Overall, it is clear that these models should become a standard element in the toolbox of empirical researchers.

References

[1] Breiman, L., 2001. Statistical Modeling: The Two Cultures. Statistical Science, 16(3): 199–215.
[2] Efron, B., Hastie, F., 2016. Computer-Age Statistical Inference. Cambridge University Press.
[3] Einav, L., Levin, J., 2014. Economics in the Age of Big Data. Science, 346(6210): 1243089.
[4] Hastie, T., Tibshirani, R., Friedman, J., 2011. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, Springer.
[5] Laeven, L., Valencia, F., 2007. Systemic Banking Crises: A New Database. IMF Working Papers 08/224.
[6] Schularick, M., Taylor, A.M., 2012. Credit Booms Gone Bust: Monetary Policy, Leverage Cycles, and Financial Crises, 1870–2008. American Economic Review, 102(2): 1029–1061.
[7] Varian, H.R., 2014. Big Data: New Tricks for Econometrics. Journal of Economic Perspectives, 28(2): 3–28.