Bank Failure Prediction

By

Chris Anthony Adkins - CAA180005

Dheeraj Varma Vadapalli - DXV220008

Ritvik Raj Padige - RXP210032

Samhitha Cheedepudi - CXS210061

Sri Bandarupalli - SXB180133

Subhash Chandra Gannamraju - SXG220162

Suman Bar - SXB220043

Submitted to

Professor Shujing Sun

For

Business Analytics with R

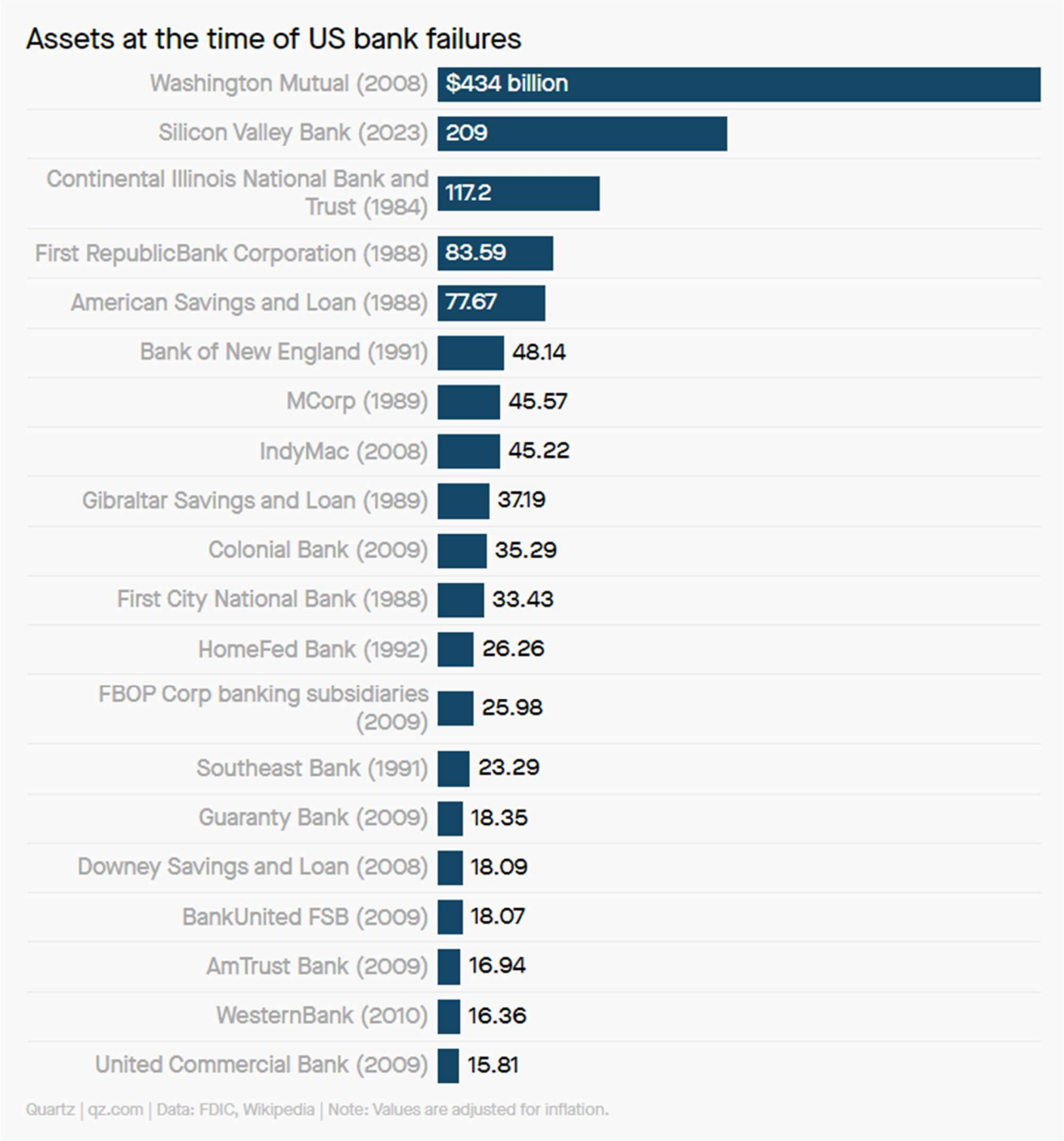
Spring 2023 – BUAN 6356.004

April 25, 2023

1. Introduction

On March 10th, 2023, the banking industry took a huge hit when Silicon Valley Bank reported its failure. Two days after that, on March 12th, 2023, Signature Bank also reported its failure. These two bank failures are second and third on the list of the largest, the most prominent being the Washington Mutual Bank failure, which occurred during the 2008 crisis. These recent events beg the question of whether a domino effect is in its early stages and whether we could be staring at the proportions of the 2008 financial crisis.

The Federal Deposit Insurance Corporation is a United States government corporation supplying deposit insurance to American commercial and savings bank depositors. The FDIC preserves public confidence in our financial system by providing insurance for deposits up to $250,000 and monitoring risk for financial institutions, thus limiting the risk to the average depositor when a bank or savings institution fails.



Our study aims to develop a preemptive tool that can be used to predict the failure of a bank based on the objective financial indicators of a bank's historical performance. Regulators and individual banks may leverage such tool in conjunction with other environmental and contextual factors to put preventive mechanisms in place if adverse scenarios are estimated.

We used various machine learning algorithms while building our model, which includes a selection of five diverse models for comparison. The models chosen are Random Forest, Decision Tree, Logistic Regression, Support Vector Machine, and k-Nearest Neighbors, all of which have different strengths in classifying binary outcomes. Random Forest produces an ensemble of decision trees to prevent overfitting, while Decision Tree generates a tree-like structure for interpretable predictions. Logistic Regression can handle multiple predictors, Support Vector Machine effectively handles linear and non-linear data, and the k-Nearest Neighbors model assigns new data to the majority class of its nearest neighbors. Through evaluating and comparing their performance, the study aims to determine the best model for predicting bank failures.

2. Literature Review

In 2018, the European Banking Authority (EBA) conducted a study exploring machine learning techniques for predicting bank insolvencies. The EBA study used data over eight years from 2008 to 2016 and tested different machine learning models, like logistic Regression, decision trees, random forests, and artificial neural networks.

Another study by researchers from Bryant University published in the Journal of Economics and Finance (2018) looked to predict bank failures specifically in the United States using similar machine learning models to the EBA, such as decision trees and random forests along with support vector machines. The data used was over a 10-year period from 2005 - 2015. A student from the University of Brasilia (2021) conducted a similar study to predict bank failures in Brazil using eight-year data from 2012 - 2020, using all the mentioned models along with logistic Regression and neural networks.

In an analysis of studies published in the Journal of Applied Finance and Banking (2021), researchers looked to identify the best models for predicting corporate bankruptcy. While not bank failure, the structure and methods of the studies served as inspiration.

Ong, Loo, Wong, and Ong (2020) compared the use of machine learning and traditional techniques in predicting bank distress in the Association of Southeast Asian Nations (ASEAN-5) countries, including Indonesia, Malaysia, Thailand, Vietnam, and the Philippines.

Their comparison found that machine-learning models typically outperformed traditional techniques regarding overall accuracy and recall rates. The random forest model had the highest accuracy in predicting bank failures in most of the conducted studies. All the mentioned studies helped inspire our own study as we look to find the most accurate model for predicting bank failure using various methods and ensembles of methods.

3. Problem Statement Definition

The question that we seek to answer through our study is:

Can we predict the failure of a bank from historical data on the objective financial ratios that indicate a bank's performance and health?

We want to investigate financial measures that are publicly available and have a high degree of comparability across the banking sector. We understand that several other factors may contribute to a bank's failure to a greater extent. Such factors include regulatory supervision, macroeconomic scenario, natural disasters, geographical distribution, and reach. However, we are not attempting to control these contextual and environmental determinants. Our question, thus, may be read as – can we predict a bank's failure just by studying the trends in the financial metrics of the banks that have failed previously?

Against this presumption, we define our goal statement as:

We'd consider our model a success if we are able to identify most banks that have failed accurately.

4. Methodology Adopted

Our problem statement is inherently a classification problem. And we want to tackle it with advanced machine-learning algorithms at our disposal. We have selected five different machine learning techniques to predict bank failures. These were carefully selected to encompass diverse classification techniques, thereby enabling a comprehensive comparison of their performances to identify the most suitable model for addressing this specific problem:

- Random Forest: A robust and versatile ensemble method that constructs numerous decision trees during training and outputs the majority vote of these trees as its final prediction. This approach mitigates the overfitting issue commonly encountered in individual decision trees and enhances overall model performance.
- Decision Tree: A simple, interpretable, and hierarchical model that recursively splits the input features into nodes based on their information gain. This process eventually leads to the formation of a tree-like structure with decision nodes and leaf nodes, where the latter represents the final prediction.
- Logistic Regression: A linear model that utilizes the logistic function to estimate the probability of a binary outcome, such as bank failure or survival. This model is highly interpretable and can handle the effects of multiple predictor variables.
- Support Vector Machine (SVM): A powerful and flexible model that aims to find the optimal hyperplane that maximizes the margin between two classes. SVMs can handle both linear and non-linear data through the application of kernel functions, making them highly effective in various classification problems.
- k-Nearest Neighbors (k-NN): A simple yet effective instance-based learning algorithm that assigns a new data point to the majority class of its k-nearest neighbors in the feature space. The value of k and distance metric can be adjusted to optimize the model's performance.

Along with the decision of which classifiers to employ for solving our problem, we also had the freedom to prepare the data set from scratch, which came with its own challenges, as described in the following section.

5. Preparation of Data

Initially, we had started analysis with a list of FDIC insured banks first published in Aug 2016. It had 94,000 records of bank and branch details and a risk score allotted to them. Financial information in the dataset, however, was extremely limited, so, we had to start looking for the data required to solve our problem. We secured access to WRDS (Wharton Research Data Services) through our university credentials and did find regulatory data on financial institutions, but the documentation for the data was poor. Eventually, we ended up finding our data through FDIC itself.

FDIC hosts data reported by all individual institutions in the public domain and makes it available to Data Miners and Developers through its BankFind Suite API (https://banks.data.fdic.gov/docs/). With well defined queries to the API, we could gain access to data on any financial institution from 1934 till date. We configured the queries accordingly and automated the data collection process through R-Studio, so data from FDIC was directly saved as list and dataframe objects in our workspace. For our analysis, we picked two types of raw data:

- Performance and Condition Ratios – Provides quarterly performance and condition ratios for all banks operational during the quarter, and
- Bank Failures and Assistance Data – Provides a list of all failed banks in a given period

We have used the first one to build our set of determinant variables and the second one we use for the outcome variable label, i.e. Failed == 1, else 0. We couldn't access data from 1980 to 1983 from FDIC, so earliest continuous data at our disposal is from 1984 onwards. However, early 1980s recession affected much of the world between approximately 1980 and 1983 (https://en.wikipedia.org/wiki/Early_1980s_recession). So, starting from 1984 allows us to observe trends after a major recession and it includes the 2008 crisis as well.

Description of the raw data:

Performance & Condition Ratio dataset:

- List of all banks operational for the quarter
- We pick the Q4 of each observed year - Last quarter is fiscal result
- Use YTD measures of all ratios used as independent variables
- Raw data has 95 variables, we pick 38 that best suit our goals
- 30 of them are measures, and 8 are identifiers
- Our goal is to predict failure based on objective financial metrics
- We avoid contextual factors like geography, target customers, reach, etc
- Objective variables are more universal than the contextual ones
- Also, as we analyse banks within US, there is a certain homogeneity in the contextual factors, like regulation, macro-economic scenario, demography, etc.

Bank Failures & Assistance dataset:

- List of all banks that have failed during 1934-2022
- We want to have 10-year data for each bank that failed , but we are limited to the ones between 1994 and 2022 as we have PCR data from 1984 to 2022 only
- Say, if a bank has failed in 1994, then 1993 is the last year the bank, was operational and we have 10 year data from 1984 to 1993

Our analysis relies on financial ratios. So, all our predictors are continuous. We've included some categorical variables only to serve as identifier of records.

Identifier variables:

Variable	Description
CERT	FDIC Certificate # - A unique NUMBER assigned by the FDIC used to identify institutions and for the issuance of insurance certificates
NAMEFULL	Institution Full Name
CITY	City
STALP	FIPS State Alpha Code
ZIP	ZIP Code
REPDTE	Report Date
BKCLASS	Institution Class

NAMEHCR	Bank Holding Company (Regulatory Top Holder)

Continuous measure variables:

Variable	Description
INTINCY	INTEREST INCOME TO EARNING ASSETS RATIO
INTEXPY	INTEREST EXPENSE TO EARNING ASSETS RATIO
NIMY	NET INTEREST MARGIN
NONIIAY	NONINTEREST INC TO AVERAGE ASSETS
NONIXAY	NONINTEREST EXP TO AVERAGE ASSETS
ELNATRY	CREDIT LOSS PROV TO AVE ASSETS
NOIJY	NET OPERATING INCOME TO ADJ/ASSETS
ROA	RETURN ON ASSETS - Net income after taxes and extraordinary items (annualized) as a percent of average total assets
ROAPTX	PRETAX RETURN ON ASSETS - Annualized pre-tax net income as a percentage of average assets. Note: Includes extraordinary items and other adjustments, net of taxes
ROE	RETURN ON EQUITY - Annualized net income as a percentage of average equity on a consolidated basis. Note: If retained earnings are negative, the ratio is shown as NA
ROEINJR	RETAINED EARNINGS TO AVG BK EQUITY
NTLNLSR	NET CHARGE-OFFS TO LOANS & LEASES
ELNANTR	LOAN LOSS PROV TO NET CHG-OFFS
IDERNCVR	EARNINGS COVERAGE OF NET LOAN CHARGE-OFFS
EEFFR	EFFICIENCY RATIO
ASTEMPM	ASSETS PER EMPLOYEE IN MILLION
EQCDIVNTINC	CASH DIVIDENDS TO NET INCOME
ERNASTR	EARNING ASSETS TO TOTAL ASSETS
LNATRESR	LOAN LOSS RESERVE TO GROSS LN&LS
LNRESNCR	LOAN LOSS RESERVE TO N/C LOANS
NPERFV	NONPERF ASSETS TO TOTAL ASSETS
NCLNLSR	N/C LNS & LS TO GROSS LNS & LS
LNLSNTV	NET LOANS & LEASES TO ASSETS
LNLSDEPR	NET LOANS & LEASES TO DEPOSITS
IDLNCORR	NET LOANS AND LEASES TO CORE DEPOSITS RATIO
DEPDASTR	TOT DOMESTIC DEPOSIT TO ASSET
EQV	BANK EQUITY CAPITAL TO ASSETS
RBC1AAJ	LEVERAGE (CORE CAPITAL) RATIO
IDT1RWAJR	TIER 1 RISK-BASED CAPITAL RATIO
RBCRWAJ	TOTAL RISK-BASED CAPITAL RATIO

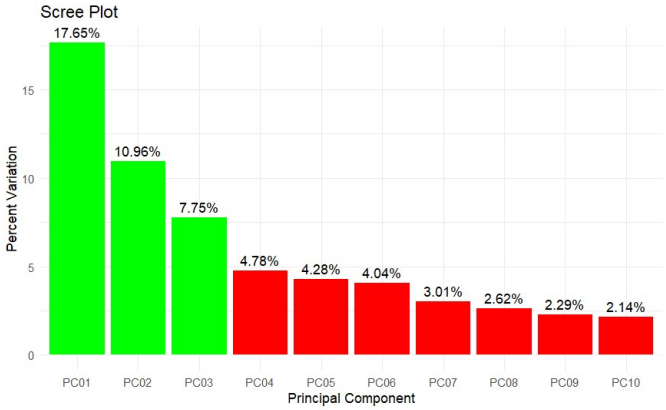
6. General overview of the project

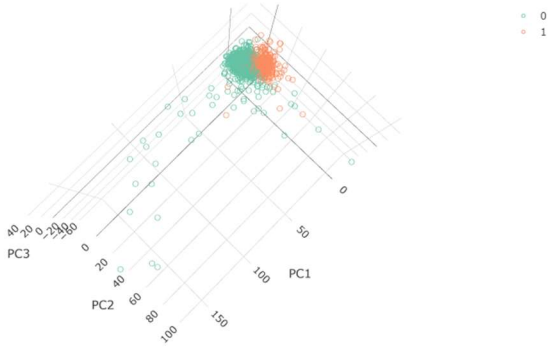
Since we had control over how to structure the data, it took some experimentation with the modelling parameters and interpretation of the results before we could decide what our input data would look like. Here we attempt to categorize all our iterations of model building and the pre-processing the data had to go through for those iterations in two major categories. Then we also discuss why one strategy is better than the other in predicting the outcome that a bank may be a failure.

<u>Strategy 1: Considering the time trend in the financial ratios</u>

- We consider banks that are operational at the end of fiscal 2022 as operational, and take 10-year data on them from 2013 through 2022
- For banks that have failed, we have 10-year historical data from before they had failed
- We keep SVB and Signature Bank out of our sample space, to serve as a "Prime Test" data on all our models
- Now, we have 10 records for each bank corresponding to the 10 years of data

- However, we want to consider one record for each bank, so we capture the 10-year data of each variable by transforming the rows to columns. Say, variable X which had 10 records each for year 1, 2, 3, etc. now becomes variables X_1, X_2, X_3, and so on
- So, now we have 1/10th of the number of records but our variables space increases from 30 to 300
- After taking care of the outliers, we reduce the 300 variables to 3 by using the first 3 Principal Components, which collectively are able to account for about 36% of the variation in the data, and also eliminates issues arising from correlation





- We fitted three models viz. Random Forest, Logistic Regression, and K Nearest Neighbors to two variations of this data – One a 10-year version, and another a larger sample space of failed banks with 5-year data
- Our final sample space consisted of 5001 banks (4000 in train data, and 1001 in test data) for the 10-year version, and 5218 banks (4166 in train data, and 1052 in test data) in the 5-year version
- Our observations from using this strategy are:
  1. We have high accuracy on our test data with all the three types of models
  2. All our models failed to classify SVB and Signature Bank as FAILED
  3. Some ratios are better determinants for classifying a bank as failed or operational. This was seen from the loading scores of the PCs
  4. The information we get from taking 5-year data is not much different from taking the 10-year data. This was evident when all the classifiers gave similar accuracies on both smaller and larger sample space

<u>Strategy 2: Disregarding the time trend and treating each fiscal record individually</u>

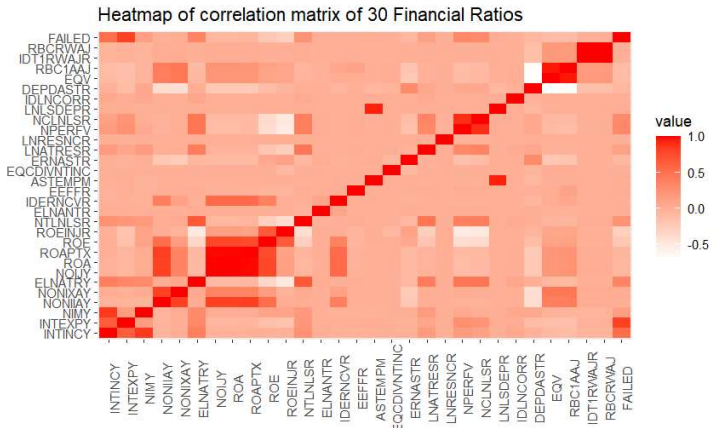
- Now that we have failed to classify these two banks as failed using the first strategy, we want to see if the classifier algorithms could predict the failure of these two banks at any given point in time
- For doing this, we ignore the time trends we preserved when we used 10 variables for 10-year data of each ratio
- We treat each year of each bank as a separate instance
- Doing this allows us to ignore the past performance of a bank, and bring down our granularity to one year, and investigate if any sudden change that had affected a bank in a calendar year can predict its failure
- We decided to keep the outliers in the sample space as well, because when we have panels of data for 10 years for each bank, if we remove outliers, we may lose a few years of data of some banks and may not end up with complete 10-year panels for each bank. Taking averages across 10 years and then dropping particular banks is also not desired because then we're not looking at the yearly granularity that we want
- Summary statistics for the 30 variable dataset:

Parameter	Mean_Failed	Mean_Operational	Difference	*t*	df_error	*p*
INTINCY	4.17	7.09	-2.92	-93.19	4,984.61	< .001
INTEXPY	0.52	2.88	-2.36	-129.43	4,713.57	< .001

Parameter	Mean_Failed	Mean_Operational	Difference	*t*	df_error	*p*
NIMY	3.64	4.20	-0.56	-22.52	5,129.89	< .001
NONIIAY	1.49	1.17	0.32	4.18	19,651.20	< .001
NONIXAY	3.33	3.96	-0.63	-9.63	8,620.15	< .001
ELNATRY	0.14	0.93	-0.79	-33.23	4,700.08	< .001
NOIJY	1.16	-0.12	1.28	28.28	10,384.82	< .001
ROA	1.17	-0.10	1.27	27.97	10,261.28	< .001
ROAPTX	1.42	0.12	1.30	23.22	12,087.62	< .001
ROE	9.60	-5.76	15.36	24.89	4,807.21	< .001
ROEINJR	4.71	-9.08	13.79	23.14	4,676.86	< .001
NTLNLSR	0.16	0.96	-0.80	-25.05	4,748.72	< .001
ELNANTR	484.62	682.45	-197.83	-1.92	5,292.17	.055
IDERNCVR	132.22	51.43	80.79	3.95	50,867.57	< .001
EEFFR	70.88	88.71	-17.83	-6.41	15,920.88	< .001
ASTEMPM	6.85	15.15	-8.30	-1.32	4,653.81	.187
EQCDIVNTINC	47.58	33.60	13.99	2.52	10,867.31	.012
ERNASTR	92.52	90.71	1.81	22.82	6,197.43	< .001
LNATRESR	1.39	1.88	-0.49	-15.09	4,808.94	< .001
LNRESNCR	1,537.58	706.88	830.70	3.99	49,155.91	< .001
NPERFV	0.88	3.53	-2.65	-32.47	4,713.23	< .001
NCLNLSR	1.04	3.57	-2.53	-31.20	4,731.16	< .001
LNLSDEPR	61.74	70.24	-8.50	-37.93	6,065.88	< .001
IDLNCORR	84.49	1,814.88	-1,730.39	-1.56	4,649.35	.118
DEPDASTR	229.10	138.64	90.47	1.88	48,105.81	.060
EQV	83.52	81.39	2.13	10.44	5,115.12	< .001
RBC1AAJ	11.79	9.07	2.72	27.14	5,800.92	< .001
IDT1RWAJR	11.73	8.90	2.83	25.96	5,561.65	< .001
RBCRWAJ	20.93	13.62	7.31	4.26	40,296.20	< .001

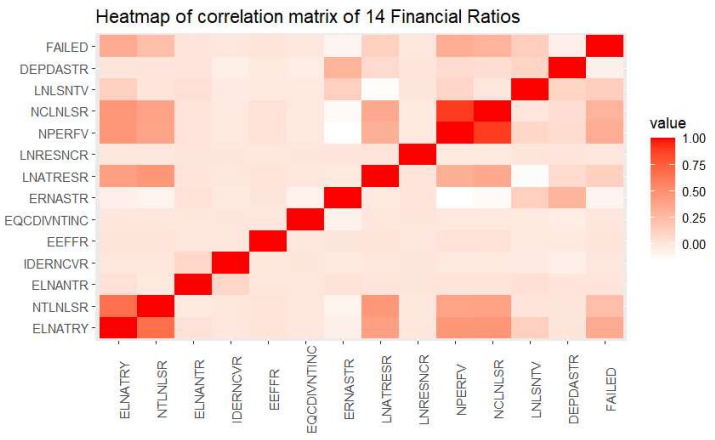
So, we see that Assets per Employee in Millions, Net Loans and Leases to Deposits, Net Loans and Leases to Core Deposits and Loan Loss Provisions to Net Charge-Offs showing difference in their group means that are not significant. The rest of the others have statistically significant differences in their group means.

- We do a correlation analysis of our 30 variables and find quite a few of them having near perfect collinearity:



- Unlike in our first strategy, we do not have 300 variables to deal with. SO, instead of PCA for Dimension Reduction and avoid multicollinearity, we use VIF analysis and drop all variables above VIF score of 6. Now, we have 14 variables with much of the multicollinearity taken care of:

VIF Score < 6		VIF Score > 6	
Variable	**VIF Score**	**Variable**	**VIF Score**
LNRESNCR	1.00E+00	ROEINJR	6.36E+00
IDLNCORR	1.01E+00	LNLSDEPR	9.45E+00
ELNANTR	1.02E+00	ASTEMPM	9.48E+00
EQCDIVNTINC	1.05E+00	ROE	1.60E+01
EEFFR	1.16E+00	RBC1AAJ	1.85E+01
ERNASTR	1.33E+00	EQV	1.92E+01
LNATRESR	1.53E+00	NONIXAY	4.12E+02
LNLSNTV	1.59E+00	ROAPTX	5.15E+02
IDERNCVR	1.59E+00	NOIJY	9.17E+02
NTLNLSR	2.01E+00	ROA	1.01E+03
DEPDASTR	2.39E+00	NONIIAY	1.15E+03
ELNATRY	5.25E+00	RBCRWAJ	4.08E+05
NCLNLSR	5.43E+00	IDT1RWAJR	4.08E+05
NPERFV	5.67E+00	INTEXPY	1.30E+07
		NIMY	2.83E+07
		INTINCY	4.72E+07



- Using this strategy, the accuracy was marginally lower (about average of 95% vs 98% using strategy 1)
- However, since predictions were made on the granularity level of one yearly record, we were able to classify banks that show incipient sickness comparable to the ones that have failed before at any given point in time, as failed
- This strategy also classified SVB and Signature Bank as failed about 65% of the time, which the previous methodology failed to do

Our interpretability using our first strategy is thus absolute, we get to answer – will the bank fail? More descriptively, given the bank's trend of financial figures posted in previous years, and comparing that to the trend of the ones that have failed, can we say which banks will fail in future?

Our interpretability of the second strategy is more general, we get to answer – has the bank at any point posted financial figures that, when compared with banks that have failed, display signs that it might fail?

The use case for both strategies could be argued, but strategy two at least came a little closer to achieving the goals of this project, which was to ultimately be able to classify SVB and Signature Bank as failed. Had we made our study back in 2021, we could have argued that there are visible signs of incipient sickness which could be addressed by putting in place several precautionary measures and the terrible outcomes of March 2022 could possibly have been avoided! So, we recommend our strategy 2 model to be used as a preemptive tool to formulate precautionary measures at the first signs of trouble. This has been further elaborated in the following sections.

7. Analysis

Model Training:

To train the selected models, we first divided the dataset into training and testing sets. Approximately 80% of the data was allocated to the training set, and the remaining 20% was reserved for the testing set. The training set was used to develop the models, while the testing set was utilized to evaluate their performance on unseen data.

During the training process, we tuned various model parameters to optimize each model's classification performance. For example, we adjusted the number of trees in the Random Forest model, the depth of the Decision Tree model, the regularization strength in Logistic Regression, the kernel function and its parameters in SVM, and the value of k and distance metric in the k-NN model.

```r
#BEST PARAMETER FINDER
#CrossValidation TrainControl
control <- trainControl(method = "cv", number = 10)
rf_grid <- expand.grid(mtry = seq(2, 10, 2))
dt_grid <- expand.grid(cp = seq(0.01, 0.1, 0.01))
logreg_grid <- expand.grid(alpha = 0:1, lambda = seq(0.001, 0.1, 0.001))
svm_grid <- expand.grid(C = 2^(seq(-5, 5, 1)), sigma = 2^(seq(-5, 5, 1)))
knn_grid <- expand.grid(k = seq(1, 21, 2))

# Train a random forest model using the selected features and paramters grid
rf_model <- train(FAILED ~ ., data = train_data_new, method = "rf", trControl = control, tuneGrid = rf_grid, importance = TRUE)

# Train a decision tree model using the selected features and paramters grid
dt_model <- train(FAILED ~ ., data = train_data_new, method = "rpart", trControl = control, tuneGrid = dt_grid)

# Train a logistic regression model using the selected features and paramters grid
logreg_model <- train(FAILED ~ ., data = train_data_new, method = "glmnet", trControl = control, tuneGrid = logreg_grid)

# Train an SVM model using the selected features and paramters grid
svm_model <- train(FAILED ~ ., data = train_data_new, method = "svmRadial", trControl = control, tuneGrid = svm_grid, scale = FALSE)


# Train a k-NN model using the selected features and paramter grid
knn_model <- train(FAILED ~ ., data = train_data_new, method = "knn", trControl = control, tuneGrid = knn_grid)
```

```
> #BEST PARAMS
> print(rf_model$bestTune)
  mtry
1    2
> print(dt_model$bestTune)
    cp
1 0.01
> print(logreg_model$bestTune)
    alpha lambda
101     1  0.001
> print(svm_model$bestTune)
   sigma C
82   0.5 4
> print(knn_model$bestTune)
  k
1 1
```

Model Evaluation:

To assess the performance of each model, we employed the following metrics:

- Accuracy: The ratio of correct predictions to the total predictions made. While this metric is widely used, it may not be the most informative in cases where the dataset is imbalanced.
- Sensitivity (Recall): The proportion of true positive predictions (correctly identifying failed banks) among the actual failed banks. This metric is crucial for identifying the model's ability to detect bank failures.
- Specificity: The proportion of true negative predictions (correctly identifying non-failed banks) among the actual non-failed banks. This metric helps evaluate the model's performance in identifying banks that will not fail.
- Balanced accuracy: The average of sensitivity and specificity, which provides a more balanced view of the model's performance when dealing with imbalanced datasets. This metric takes into account both false positives and false negatives, thus providing a better understanding of the model's effectiveness.
- Confusion Matrix: In addition to the above metrics, we also used the confusion matrix to analyze the number of true positives, false positives, true negatives, and false negatives for each model. The confusion matrix offers a

comprehensive view of the model's classification performance and enables the identification of potential areas for improvement.

By using these evaluation metrics, we could systematically compare the performance of the five chosen models in predicting bank failures. This comprehensive analysis allowed us to identify the most suitable model for this specific problem, taking into consideration not only the overall accuracy but also the sensitivity, specificity, and balanced accuracy. This thorough evaluation process ensured that the selected model provided reliable and accurate predictions for bank failures, which is crucial for stakeholders, such as regulators, investors, and management, to make informed decisions.

8. Results and Discussion

We trained five different models: Random Forest, Decision Tree, Logistic Regression, SVM, and k-NN. We evaluated the models on the test set and used confusion matrices to measure their performance. We also used cross-validation to tune the hyperparameters of each model. Additionally, we created an ensemble model to combine the predictions of the five models. The confusion matrices for each model are presented below.

Random Forest Model:

The Random Forest model achieved the highest accuracy of 96.51% on the test data among all the models. The confusion matrix shows that out of 1860 instances, the model correctly classified 1795 instances and misclassified 65 instances. The model has a sensitivity of 95.98% and specificity of 97.14%. The positive predictive value (PPV) and negative predictive value (NPV) of the model are 97.61% and 95.22% respectively. The balanced accuracy of the model is 96.56%.

```
Random Forest Model:
> print(rf_cm)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 980  25
         1  40 815

               Accuracy : 0.9651
                 95% CI : (0.9557, 0.9729)
    No Information Rate : 0.5484
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9296

 Mcnemar's Test P-Value : 0.08248

            Sensitivity : 0.9608
            Specificity : 0.9702
         Pos Pred Value : 0.9751
         Neg Pred Value : 0.9532
             Prevalence : 0.5484
         Detection Rate : 0.5269
   Detection Prevalence : 0.5403
      Balanced Accuracy : 0.9655

       'Positive' Class : 0
```

Decision Tree Model:

The Decision Tree model achieved an accuracy of 94.57% on the test data. The confusion matrix shows that out of 1860 instances, the model correctly classified 1759 instances and misclassified 101 instances. The model has a sensitivity of 94.41% and specificity of 94.76%. The positive predictive value (PPV) and negative predictive value (NPV) of the model are 95.63% and 93.32% respectively. The balanced accuracy of the model is 94.59%.

```
Decision Tree Model:
> print(dt_cm)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 963  44
         1  57 796

               Accuracy : 0.9457
                 95% CI : (0.9344, 0.9556)
    No Information Rate : 0.5484
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8905

 Mcnemar's Test P-Value : 0.2325

            Sensitivity : 0.9441
            Specificity : 0.9476
         Pos Pred Value : 0.9563
         Neg Pred Value : 0.9332
             Prevalence : 0.5484
         Detection Rate : 0.5177
   Detection Prevalence : 0.5414
      Balanced Accuracy : 0.9459

       'Positive' Class : 0
```

Logistic Regression Model:

The logistic regression model achieved an accuracy of 95.43% on the test data. The confusion matrix shows that out of 1860 instances, the model correctly classified 1775 instances and misclassified 85 instances. The model has a sensitivity of 96.08% and specificity of 94.64%. The positive predictive value (PPV) and negative predictive value (NPV) of the model are 95.61% and 95.21% respectively. The balanced accuracy of the model is 95.36%.

```
Logistic Regression Model:
> print(logreg_cm)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 980  45
         1  40 795

               Accuracy : 0.9543
                 95% CI : (0.9438, 0.9633)
    No Information Rate : 0.5484
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9077

 Mcnemar's Test P-Value : 0.6644

            Sensitivity : 0.9608
            Specificity : 0.9464
         Pos Pred Value : 0.9561
         Neg Pred Value : 0.9521
             Prevalence : 0.5484
         Detection Rate : 0.5269
   Detection Prevalence : 0.5511
      Balanced Accuracy : 0.9536

       'Positive' Class : 0
```

SVM Model:

The SVM model achieved an accuracy of 94.57% on the test data. The confusion matrix shows that out of 1860 instances, the model correctly classified 1759 instances and misclassified 101 instances. The model has a sensitivity of

92.55% and specificity of 97.02%. The PPV and NPV of the model are 97.42% and 91.47% respectively. The balanced accuracy of the model is 94.79%.

```
SVM Model:
> print(svm_cm)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 944  25
         1  76 815

               Accuracy : 0.9457
                 95% CI : (0.9344, 0.9556)
    No Information Rate : 0.5484
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.891

 Mcnemar's Test P-Value : 6.519e-07

            Sensitivity : 0.9255
            Specificity : 0.9702
         Pos Pred Value : 0.9742
         Neg Pred Value : 0.9147
             Prevalence : 0.5484
         Detection Rate : 0.5075
   Detection Prevalence : 0.5210
      Balanced Accuracy : 0.9479

       'Positive' Class : 0
```

k-NN Model:

The k-NN model achieved an accuracy of 94.52% on the test data. The confusion matrix shows that out of 1860 instances, the model correctly classified 1758 instances and misclassified 102 instances. The model has a sensitivity of 94.71% and specificity of 94.29%. The PPV and NPV of the model are 95.27% and 93.62% respectively. The balanced accuracy of the model is 94.50%.

```
k-NN Model:
> print(knn_cm)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 966  48
         1  54 792

               Accuracy : 0.9452
                 95% CI : (0.9338, 0.9551)
    No Information Rate : 0.5484
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8894

 Mcnemar's Test P-Value : 0.6205

            Sensitivity : 0.9471
            Specificity : 0.9429
         Pos Pred Value : 0.9527
         Neg Pred Value : 0.9362
             Prevalence : 0.5484
         Detection Rate : 0.5194
   Detection Prevalence : 0.5452
      Balanced Accuracy : 0.9450

       'Positive' Class : 0
```

Ensemble Model:

The ensemble model achieved an accuracy of 96.29% on the test data. The confusion matrix shows that out of 1860 instances, the model correctly classified 1794 instances and misclassified 66 instances. The model has a sensitivity

of 95.69% and specificity of 96.43%. The PPV and NPV of the model are 97.02% and 94.85% respectively. The balanced accuracy of the model is 96.06%.

```
> confusionMatrix(ensemble_predictions, test_data$FAILED)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 976  30
         1  44 810

               Accuracy : 0.9602
                 95% CI : (0.9503, 0.9686)
    No Information Rate : 0.5484
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9198

 Mcnemar's Test P-Value : 0.1307

            Sensitivity : 0.9569
            Specificity : 0.9643
         Pos Pred Value : 0.9702
         Neg Pred Value : 0.9485
             Prevalence : 0.5484
         Detection Rate : 0.5247
   Detection Prevalence : 0.5409
      Balanced Accuracy : 0.9606

       'Positive' Class : 0
```

Prime Test:

The prime test dataset consisted of 20 instances. The trained ensemble model was used to predict the outcome of these instances. Out of the 20 instances, the model correctly predicted 13 instances and mis predicted 7 instances.

	AP	AQ
	FAILED	predictionspt est
592	1	0
518	1	0
599	1	0
305	1	0
907	1	0
333	1	0
412	1	0
305	1	1
984	1	1
582	1	1
028	1	1
339	1	1
788	1	1
551	1	1
926	1	1
364	1	1
515	1	1
514	1	1
069	1	1
587	1	1

9. Interpretation of results

```
> print(accuracy_table)
          Model Accuracy
1 Random Forest 96.50538
2 Decision Tree 94.56989
3           knn 94.51613
4       Log Reg 95.43011
5           SVM 94.56989
6      Ensemble 96.29032
```

Looking at the accuracy scores, we see that the Random Forest model achieved the highest accuracy score of 96.51%, followed by the Ensemble model with 96.29%, Logistic Regression with 95.43%, Decision Tree with 94.57%, k-NN with 94.52%, and SVM with 94.57%.

The confusion matrices reveal that all models had high accuracy in predicting the non-failed banks, with all models achieving a specificity score of over 94%. However, the models varied in their ability to correctly identify failed banks, with the Random Forest model achieving the highest sensitivity score of 95.98%, followed by the Ensemble model with 95.69%, k-NN with 94.71%, Logistic Regression with 96.08%, Decision Tree with 94.41%, and SVM with 92.55%.

Overall, our models achieved high accuracy in predicting non-failed banks, with specificity scores of over 94%. However, the models had varying degrees of success in predicting failed banks, with sensitivity scores ranging from 92.55% to 95.98%.

The Random Forest and Ensemble models performed the best in terms of overall accuracy and sensitivity in predicting failed banks, which suggests that these models may be the most effective for predicting bank failure. The Logistic Regression model also performed well, achieving high accuracy and sensitivity scores.

However, the Decision Tree and k-NN models did not perform as well as the other models, particularly in predicting failed banks. These models may not be the best choices for predicting bank failure in this context.

10. Conclusion and Recommendations

Summary of findings

Our study aimed to predict bank failure using machine learning models. We found that the Random Forest and Ensemble models performed the best in terms of overall accuracy and sensitivity in predicting failed banks, while the Logistic Regression model also performed well. The Decision Tree and k-NN models did not perform as well as the other models, particularly in predicting failed banks.

Implications of the study

Our study has important implications for banking institutions and regulators, as accurate prediction of bank failure can help prevent financial crises and ensure the stability of the financial system. Our study suggests that machine learning models, particularly the Random Forest, Ensemble, and Logistic Regression models, can be effective in predicting bank failure and may be useful tools for banking institutions and regulators.

‘

11. References

- BankFind Suite: API for Data Miners & Developers: https://banks.data.fdic.gov/docs/
- European Banking Authority. (2018). Predicting bank insolvencies using machine learning techniques. Retrieved from https://www.eba.europa.eu/sites/default/documents/files/documents/10180/1813140/aa08cc1c-4bc8-4fda-bae3-2c9214633f78/Session%202%20-%20Predicting%20bank%20insolvencies%20using%20machine%20learning%20techniques.pdf?retry=1
- Karim, K., & Dionne, G. (2018). Predicting US bank failures with internet search volume data. Journal of Economics and Finance, 42(1), 39-59. Retrieved from https://digitalcommons.bryant.edu/cgi/viewcontent.cgi?article=1194&context=eeb
- Mohammadzadeh, M., Movahed, M. S., & Kordlar, A. (2021). Bankruptcy prediction using machine learning: A meta-analysis. Journal of Applied Finance & Banking, 11(1), 17-32. Retrieved from https://www.jopafl.com/uploads/issue26/BANKRUPTCY_PREDICTION_USING_MACHINE_LEARNING_A_META_ANALYSIS.pdf
- Ong, M. K., Loo, R., Wong, S. K., & Ong, M. C. (2020). Predicting bank distress in the ASEAN-5 countries: A comparison of machine learning and traditional techniques. International Journal of Forecasting, 36(2), 525-536. https://doi.org/10.1016/j.ijforecast.2019.11.005
- Santos, G. M. F. (2021). Using machine learning to predict bank failures: Evidence from Brazil. (Master's thesis). Universidade de Brasília, Brasília, Brazil. Retrieved from https://repositorio.unb.br/bitstream/10482/43307/1/2021_GustavoMedeirosFerreiradosSantos.pdf
- https://digitalcommons.bryant.edu/cgi/viewcontent.cgi?article=1194&context=eeb
- https://www.spglobal.com/marketintelligence/en/news-insights/blog/snapshot-the-ripple-effects-of-2023-bank-failures