Financial Risk Management for SBA 504 Loans

Soumya Patro, Allen Wu, Hujia Yu, Michelle Zhang
Department of Management Science & Engineering
Stanford University
{sopatro, nalkpas, hujiay, mzhang8}@stanford.edu

Abstract

In this paper, we explore the SBA loan dataset and consider a number of different models for assessing the risks associated with small businesses. First, we predict default behavior by building classification models using various methods, such as logistic regression and neural networks. Our best logistic model had a test accuracy of 0.7913, an F1 score of 0.405864, and a test AUC of 0.7618. Then, we study time to default by using a hazard model. We fitted two different hazard models, one to the full data set and the other to just defaulted loans. We found that the most significant variables in both models are `[ApprovalFiscalYear, TermInMonths, SP500_Yearly_Return, Log_GrossApproval_Norm, Log_HPI_Norm, ThirdPartyDollars_Norm]`. We then use the hazard model and linear regression loss model to simulate the loss distribution for pools of loans. Lastly, we study the risk profiles of securitized tranches over one- and five-year time spans and estimate the cash flows for junior and senior tranches. We observed a mean loss estimate of approximately 22 million dollars and 54 million dollars in the shorter and longer time spans respectively. Furthermore, we saw that an investor in the junior tranche is significantly more likely to incur loss compared to one in the senior tranche.

1 Introduction

There are more than 25 million small businesses in the U.S., comprising of 99.7% of all businesses in the country. Unlike their larger cohorts, small businesses can also be quite volatile. On average, from 2005 to 2015, only 78.5% of new establishments survived one year [3]. Because small businesses are both vital and unpredictable, there's an urgent demand from lenders, investors, and supervising agencies to accurately enumerate and quantify the circumstances and probabilities of small businesses failing.

In this paper, we explore the SBA loan dataset and consider a number of different models for assessing the risks associated with small businesses. First, we consider risk of default using logistic regression, hazard rate, and neural network models. We primarily focus on the hazard rate and neural network models, using logistic regression as a baseline model. From this, we build a linear regression model to further estimate the loss at default of a pool of loans and study risk profiles of securitized tranches over one- and five-year time spans.

2 Dataset

We used data from the U.S. SBA 504 loan program, consisting of 150,000 loans issued between 1990 and 2014. We excluded 'CANCELLED' or 'EXEMPT' loans because they do not inform us about the risk profiles of the borrowing businesses.

2.1 Data Preprocessing

We first dealt with missing data by adding a `MISSING` level for categorical variables and using the column mean for numerical variables, so that missing values zero out after normalization. We also noticed that a number of the ZIP codes in the dataset were nonsensical, e.g. 99999 or out of the ZIP code range for the corresponding state, so we listed the zip code ranges for each U.S. state and the outlying U.S. territories in the data set, such as Guam and the Virgin Islands. We use that list to label incorrect zipcodes as missing.

Moreover, in order to manage the number of covariates and better capture the relationships between them, we derived the following variables:

```
[2DigitNaics, ProjectStateneqBorrowerState,
TermMultipleYear, RepeatBorrower,
BankStateneqBorrowerState]
```

`2DigitNaics` is the first two digits of the NAICS code for the business the loan was issued to, which represent which industry that business belongs to. `ProjectStateneqBorrowerState` is an indicator vari-

able of whether project state is not equal to borrower state. `TermMultipleYear` indicates whether the term of the loan is divisible by 12 months.

In addition to creating these higher-level variables, we also augmented the dataset with broad economic factors such as the housing and consumer price indexes, unemployment rate, and S&P 500 returns. We did this to consider the influence of national economic trends and systemic risks as well as the individual characteristics of the loans. We indexed these data from other data sets by each loan's state and approval year and fixed them for the term of the loan.

We also log-normalized most of our numerical variables so that they would be on an uniform scale. This helped our regression models converge more quickly and make more accurate predictions. This added the following variables to the data in place of the original columns,

```
[Log_GrossApproval_Norm, Log_HPI_Norm,
ThirdPartyDollars_Norm]
```

where `Log_..._Norm` indicates that the variable is first log-transformed and then normalized.

We removed the following variables to reduce number of parameters in the model, as we determined these variables to be either less informative than or already encapsulated in other variables:

```
[BorrName, BorrStreet, CDCName, CDCStreet,
ThirdPartyLenderName, NaicsDescription]
```

For example, the names of the borrowers or what streets their businesses are located at are unique for almost all loans and are unlikely to be meaningfully correlated to the economic health of the loans. We already considered geographic information in the state and ZIP code variables, and the NAICS description is simply a label for the NAICS code, arguably providing redundant information.

Finally, we split our dataset into training, test and validation sets using an 80-10-10 rule. We wanted to have as much data as possible in our training set, and 5,000 data points seemed sufficient for our test and validation sets.

We organized our data in two different ways before splitting to facilitate training the different models, either shuffling the loans or sorting the loans by time of issue. The randomly separated data is used to train our classification models, i.e. the logistic regression and neural network models, our hazard model, and the loss model in order to maintain consistency. The sorted data is used in simulating the loss distribution using a pool of loans, since we wanted to test our model on more recent, unseen data.

2.2 Data Exploration

Additionally, we explored the dataset for trends and patterns that could affect our models. We primarily plotted the loan default rate against several different features, both static and time-dependent.

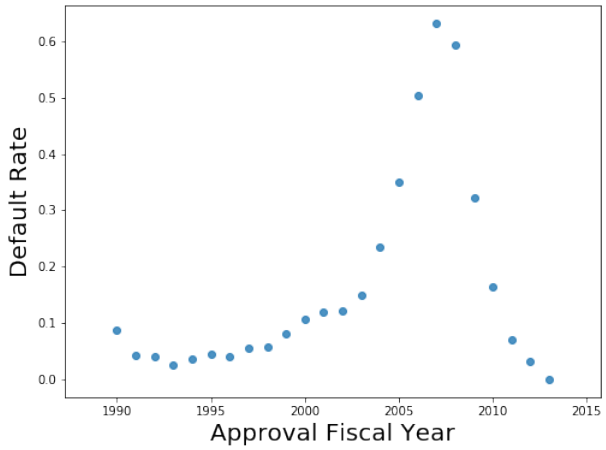


Figure 1: Default Rate By Approval Fiscal Year

Figure 1 shows the default rate by year of issue. The default rates peak in the time periods leading up to and of the financial crisis of 2007-2008.

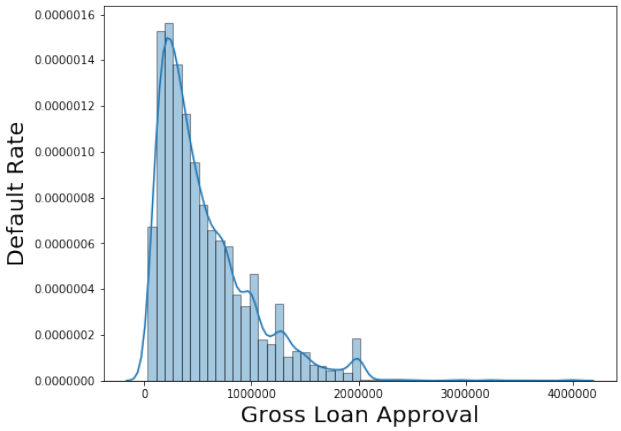


Figure 2: Default Rate By Gross Approved Loan Amount

Figure 2 shows default rate by loan size. It's worth noting that smaller loans are at greater risk of default. This seems to make some sense, as borrowers with poor credit are likely only approved for smaller loans.

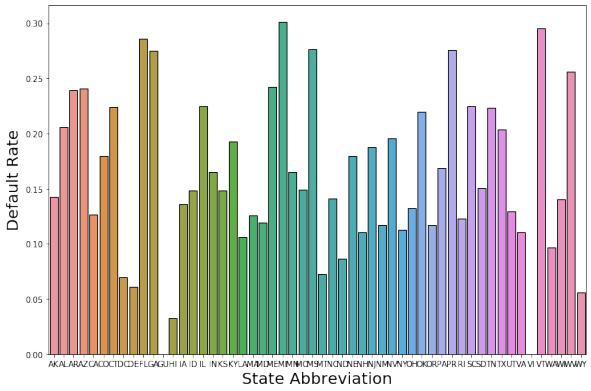


Figure 3: Default Rate By Borrower State 1990-2014

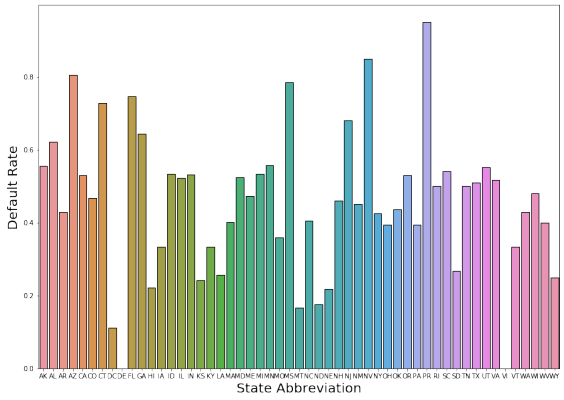


Figure 4: Default Rate By Borrower State 2007-2014

Figure 3 shows the default rate by borrower state. Michigan, Vermont and Florida seem to have the highest default rates. However the default rates for these states have been lower over the past 10 years. Figure 4 shows default rates by state since 2007.

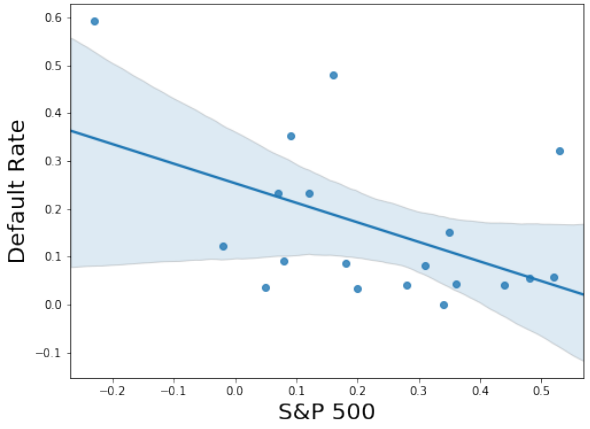


Figure 5: Default Rate vs S&P 500 Yearly Returns

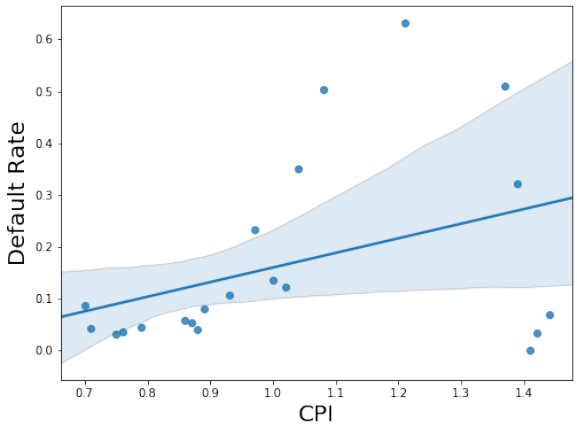


Figure 6: Default Rate vs Consumer Price Index

Figures 5 & 6 show the relationship between loan default rates and S&P 500 returns and the CPI, respectively. High S&P 500 returns correlate with lower default rates, since they indicate good market conditions. A high CPI means high inflation, and we see that leads to increased default rates.

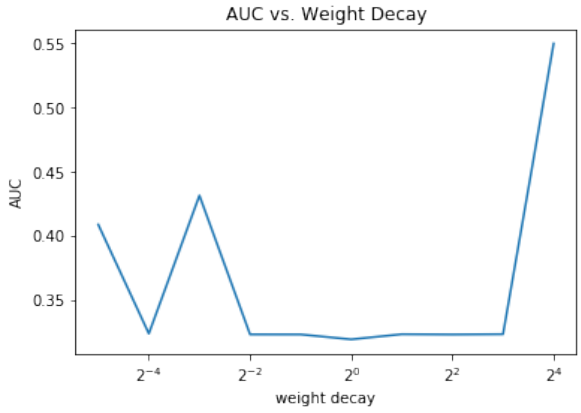
3 Model Selection

We implemented a number of different models and compared their performance.

3.1 Logistic Regression

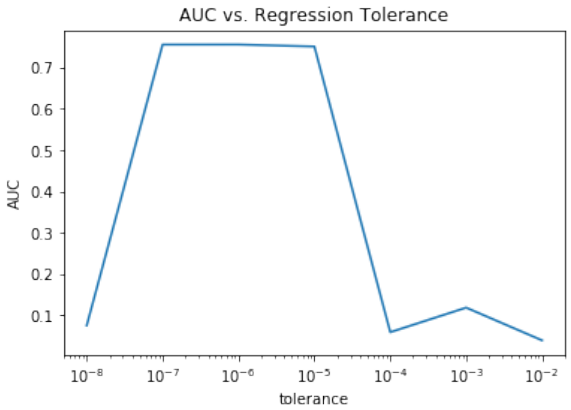
First, we decided on a logistic regression model to serve as a baseline. We used basic L2 regularization, also known as ridge regularization, and chose our hyperparameters to maximize the area under the ROC curve, or AUC. For our final parameterization, we chose the threshold that maximizes the harmonic mean of the sensitivity and specificity of model on the validation set.

We primarily considered two hyperparameters: the weight decay λ and the stopping threshold of gradient descent ϵ. This led to the graph of AUC as a function of $1/\lambda$ below:



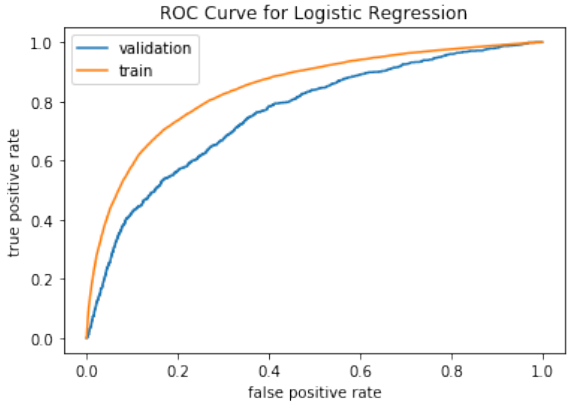
We see that there is no obvious correlation between λ and AUC but that some values work better for our data set.

We obtained the following graph for ϵ:



The relationship between the two variables appears somewhat more clear and influential. It seems important that ϵ be within 10^{-5} and 10^{-7} for the best results, so that we don't overfit or underfit.

After a pairwise search through the parameter space, we found that the optimal values of λ and ϵ are $1/32$ and 10^{-7}, giving us an AUC of 0.7578 for our validation set. Further, we had the following ROC curve:



This is about what we would hope for. The model fits the training set better than the validation set, but the gap between the two is reasonable. This indicates that we've fit the training set pretty well.

After trying every threshold up to four significant digits on our final model, we saw that $t = 0.2061$ gave us a maximum sensitivity of 0.7429 and specificity of 0.6479. To start, since we're primarily using the logistic model as a baseline, we chose to weigh type I and type II errors equally and aim for the most generally accurate model. This led us to predict that a proportion of 0.2745 of loans in our training set will default with an accuracy of 0.7983, and 0.5008 of loans in our test set will default with an accuracy of 0.5864 but an AUC of 0.7408. We also calculated that we have an F1 score of 0.3690 on our test set.

Instead of weighing type I and type II errors equally, we tried weighting sensitivity and specificity by the number of true negatives and positives in the data set, ie. by the proportion of defaults. So rather than maximizing $(1 - e_1) + (1 - e_2)$, where e_1 and e_2 are the type I and type II error rates, we maximize $(1 - p)(1 - e_1) + p(1 - e_2)$ for the proportion of defaults p. Essentially, we prioritize type I and type II errors here based on how many opportunities we have to make each kind of error. In our case, since there are fewer defaults than survivals, we hoped that this would lead to the prediction of fewer defaults and thus higher accuracy on the test set.

This weighting gave us an optimal threshold of 0.4118, which predicted 0.2116 of test loans to default, improving test set accuracy to 0.7913 with a sensitivity of 0.5094 and a specificity of 0.8429. This new threshold yielded fewer true positives but much fewer false positives as well. The F1 score also improved to 0.4059 and AUC to 0.7618.

We also performed a log-likelihood ratio test to find the most and least significant covariates for the logistic model. The following table contains the eight most critical variables:

Variable	Log-Likelihood Ratio
ApprovalFiscalYear	36920.5969
CPI	9523.6261
Log_HPI_Norm	5348.9835
TPL_State_MISSING	1202.9457
BorrState_CA	558.4069
ProjectState_CA	517.3812
2DigitNaics_72	318.9630
SP500_Yearly_Return	219.9957
Yearly_Unemployment_Rate	212.6620

Table 1: Most Significant Variables for the Logistic Model

It seems that geography and broad economic trends are the strongest predictors of default rate in this model, as one might expect. Very interestingly, NAICS code 72, comprising of accommodation and food services, is highly informative. This matches our intuition that restaurants are risky businesses and is at least a glimpse of real learning.

Below are the eight least critical variables:

Variable	Log-Likelihood Ratio
subpgmdesc_Premier	-0.02151
DeliveryMethod_PCLP	-0.02151
TPL_State_AK	-0.0001243
TPL_State_Ro	0.0
TPL_State_D_CA	0.0
CDC_State_AK	0.0002826
ProjectState_MISSING	0.0002863
2DigitNaics_99	0.001177
2DigitNaics_45	0.002361

Table 2: Least Significant Variables for the Logistic Model

While the negative log-likelihood ratios are concerning, the values are so small that the discrepancies are likely just computational rounding error. Here, we see that third party lender state, loan type, and some NAICS codes are particularly uninformative. This also matches our intuition. NAICS code 45 describes retail and code 99 describes public services. These categories are so broad that it is natural they would be uninformative.

Ultimately, while the performance of the logistic model is not perfect, it is impressively accurate for such a simple model and provides a number of interesting insights. Logistic regression represents a clear benchmark for our other models.

3.2 Neural Network

Unfortunately, we were unable to implement a functional neural network. We implemented a fully connected binary classification network with ReLU activations, using cross-entropy loss. However, we faced problems with exploding and vanishing gradients, such that our final network made fixed predictions rather than varying by the covariates of each loan. We tried a number of different remedies and ultimately managed to parameterize a network such that the parameters persisted even after long periods of training, but we were unable to further tune that network to make meaningful predictions. We present our process here as a foundation for future research.

We solved our problem with vanishing and exploding gradients via a number of techniques. We clipped the gradients at each step, added batch normalization, reduced the learning rate, reduced our mini-batch size, added dropout regularization, and switched from using stochastic gradient descent to Adam descent with the canonical parameterization [4].

We also increased the number of layers and neurons in our network. We parameterized the size of our network via the number of layers n and a scaling factor k, such that the first hidden layer of our network had kn_{in} neurons and the number of layers geometrically decreased to n_{out} neurons in the last layer.

Once we tested several numbers and parameters and checked the parameters of the network after 5-10 epochs of training, we ultimately found that $n = 7$, $k = 2$, a weight decay of 0.1, batch size of 20, learning rate of 0.0001, and dropout probability of 0.15 worked best. This model produced non-trivial predictions but still predicts most loans have around a 50% chance of defaulting. For example, this is a set of predictions on a batch of 10 loans:

Loan	P(survive)	P(default)
1	0.4999	0.5001
2	0.4732	0.5268
3	0.4999	0.5001
4	0.4999	0.5001
5	0.4732	0.5268
6	0.4999	0.5001
7	0.4732	0.5268
8	0.4999	0.5001
9	0.4999	0.5001
10	0.4732	0.5268

We also trained a smaller network of $n = 5$ and $k = 2$, but there was even less variation in the predictions of that model.

It is possible that if we trained this network for a very large number of epochs, it would eventually make good predictions. However, we suspect that our network still requires more refinement. Currently, the most promising path to further improvement seems to be more careful variable selection. We trained our neural network on a maximal set of parameters to give it as much data as possible. Because our covariates are mostly levels of categorical variables, however, the data we're feeding it is quite sparse. We believe future attempts should begin with more rigorous experimental selection of variables using simpler models.

3.3 Hazard Model

In addition to the classification models, we also modeled the time to default τ of each loan using hazard model. We used the methodology introduced by Kiefer (1988), where we describe the hazard rate as a function of explanatory variable X by defining the intensity

$$\lambda = \lambda(X)$$

for some non-negative function λ. The event time τ then is assumed to have an exponential distribution with parameter $\lambda(X)$ conditional on a realization of X such that

$$P(\tau \leq t|X) = 1 - \exp(-\lambda(X)t).$$

We modeled a linear relationship between our variables and event timing by using the proportional hazard model to capture the influence of borrower and other characteristics on event timing,

$$\lambda(x;\beta) = \exp(\beta^T x) = \exp(\beta_1 x_1 + ... + \beta_n x_n),$$

and we used our collected data to estimate the vector of coefficients β.

Given collected data and structure of the model, we estimated β by maximizing the following likelihood function

$$L(\beta|\{X_i\}) = \prod_{i=1}^{n} L_i(\beta),$$

where the likelihood $L_i(\beta)$ of a loan i originated at time t_i and maturing at time T_i without defaulting is

$$L_i(\beta) = \exp\{-\lambda(X_i;\beta)(T_i - t_i)\}$$

and that of a loan defaulting at time $\tau_i \leq T_i$ is

$$L_i(\beta) = \lambda(X_i;\beta)\exp\{-\lambda(X_i;\beta)(\tau_i - t_i)\}.$$

For the loans that were not paid in full and do not default in our observation frame, we considered the probability of no default by time S:

$$L_i(\beta) = \exp\{-\lambda(X_i;\beta)(S - t_i)\}$$

We estimated $\hat{\beta}$ by maximizing the overall likelihood function for our training set. As usual, we searched for optima via gradient descent.

After we fitted β, we predicted whether each given realization of X would default or not, and if so, when exactly the default was most likely to occur. To predict probability of survival of a loan x_i to time $T > S$, where S is the origination time of the loan, we calculated $P(\text{Survival}) = \exp\{-\lambda(X_i;\hat{\beta})(T - S)\}$.

Note this only gives us the survival probability of the loan X_i. In order to predict default, we needed to find an optimal threshold from an ROC curve.

We utilized the CoxPHFitter in the `lifelines` package [1] in Python to fit a cox hazard model to our data. Notably, CoxPHFitter goes to extra lengths to help gradient descent convergence by using decreasing step sizes. Moreover, CoxPHFitter includes a baseline hazard $\lambda_0(t)$ in the overall intensity. However, instead of having the time-varying covariates in both the baseline hazard and the feature variables, CoxPHFitter only fits time varying effects on the baseline hazard and keeps X independent of time:

$$\lambda(t|X) = \lambda_0(t)\exp(\beta^T X)$$

The idea behind the model is that the log-hazard of an individual is a linear function of its static covariates and a population-level baseline hazard that changes over time.

Notice that the only time component in this model is in the baseline hazard, λ_0. The exponential term is a scalar factor that increases or decreases the baseline hazard.

We fitted the hazard model two ways. One approach included all the data and set the duration of a loan to be its age in the case of default and its `TermInMonths` in the case of survival. The other approach fitted defaulted loans only and set the duration of a loan to be its age at default. The different perspectives of these approaches allowed us to better understand overall loan default behavior.

We included the following explanatory variables in our hazard model:

```
[ApprovalFiscalYear, DeliveryMethod, subpgmdesc,
TermInMonths, BusinessType, SP500_Yearly_Return,
CPI, Log_GrossApproval_Norm, Log_HPI_Norm,
ThirdPartyDollars_Norm, TermMultipleYear,
RepeatBorrower, BankStateneqBorrowerState,
ProjectStateneqBorrowerState, 2DigitNaics,
DaysToDefault]
```

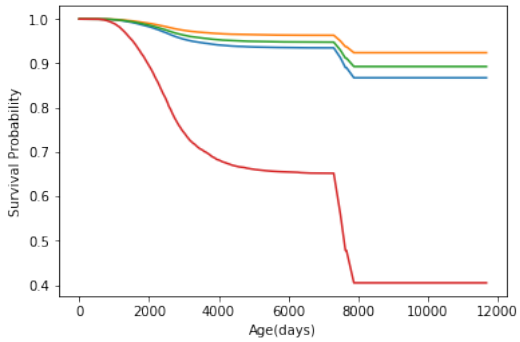


Figure 7: Hazard Distributions of 4 Random Loans for the Holistic Model

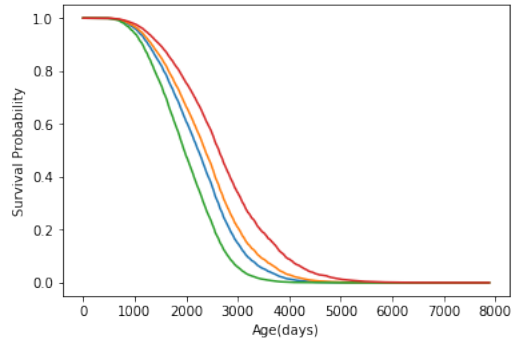


Figure 8: Hazard Distributions of 4 Random Loans for the Default Model

Figures 7 and 8 show some sample distributions of loans for the hazard models fit to all loans and only default loans respectively. In Figure 7, the survival distributions have kinks at around 7400 days, which corresponds to 240 months and is the most common `TermInMonths` of loans. Furthermore, they plateau at around 0.88, which is the percentage of defaulted loans in our dataset.

On the other hand, Figure 8 shows the survival distributions of default loans only, which decay to zero. We can observe from the four sampled distributions that the survival probabilities for different samples vary significantly from one another. For example, these four loans are expected to survive past 2000 days with probabilities of 0.48, 0.60, 0.66, and 0.76 respectively. We expect that the hazard model fit to default loans gives more accurate predictions of the specific time of default than hazard model fit to all loans, since the

holistic model has to take into account the likelihood of default in addition to time of default.

If we wanted an end-to-end hazard model to classify defaults as well as predict times to default, we would need to fit our model to the whole data set. However, to fully utilize the predictive power of hazard model for predicting time to default, we might prefer a two-stage model where we use first a specialized classification model to predict whether loans will default, such as a logistic regression model or neural network, and then feed loans we predict will default into a hazard model trained specifically to predict times of default.

The two-stage model, in contrast to the end-to-end model, allows us to tune each submodel to maximize its performance on its assigned task, rather than have a jack of all trades that is a master of none. The downside of this pipeline approach is that we risk overfitting, and the models cannot learn from one another. For example, if one model learns a feature that strongly predicts immediate default, the other model would have to learn that feature independently and may not learn it at all. We also have to train two models rather than just one.

Variable	Coef	p-value
ApprovalFiscalYear	0.2286	0.0000
TermInMonths	-0.0028	0.0000
SP500_Yearly_Return	-0.6363	0.000
Log_HPI_Norm	-0.1281	0.000
ThirdPartyDollars_Norm	0.0909	0.000
RepeatBorrower	-0.2186	0.0003
BankStateneqBorrowerState	0.3073	0.000
Log_GrossApproval_Norm	0.0309	0.0437

Table 3: Most Significant Variables for Hazard Model Fitted to All Loans

Variable	Coef	p-value
ApprovalFiscalYear	0.1387	0.0000
TermInMonths	-0.0016	0.0022
SP500_Yearly_Return	-0.2133	0.0533
Log_HPI_Norm	0.0670	0.0010
ThirdPartyDollars_Norm	0.0250	0.000
RepeatBorrower	0.0037	0.9498
BankStateneqBorrowerState	0.0441	0.4450
Log_GrossApproval_Norm	-0.0902	0.0000

Table 4: Comparison of Variable Significance for Hazard Model Fitted to Default Loans

In addition to the probability distributions, we also looked into the most significant variables and their coefficient values. Again, we present in Table 3 and Table 4 the respective coefficients of the holistic hazard model and the default-exclusive model.

Interestingly, we can note that `ApprovalFiscalYear` is positively correlated to survival probability, indicating that the later the loan is approved, the more likely the loan is to survive.

Next, we compare the significant variables between the two models. Most of the variables significant in the holistic hazard model are significant in the default-exclusive hazard model, with two notable exceptions: `RepeatBorrower` and `BankStateneqBorrowerState`. `BankStateneqBorrowerState` is an indicator for whether a loan's bank state not equal to borrower state. In Table 3, `RepeatBorrower` has a negative coefficient value and is highly significant, indicating an inverse relationship between `RepeatBorrower` and loan survival. This appears reasonable, as a person who repeatedly takes out loans is more likely to be in financial trouble.

On the other hand, the coefficient value of `RepeatBorrower` in Table 4 is close to zero and has a large p-value, indicating that it is not very significant. This also seems reasonable, as given that a loan has defaulted, whether the person is a repeat borrower is no longer relevant.

`BankStateneqBorrowerState` exhibits the same pattern between the two tables, likely for similar reasons. The positive coefficient of `BankStateneqBorrowerState` in Table 3 indicates that the survival probability is significantly higher if bank and the borrower are *not* from the same state. This significance again no longer holds for the default-exclusive model.

The end-to-end hazard model introduced in this section is later used to simulate loss distributions for tranches of loans.

4 Loss at Default

Supplementary to default rate, we also studied the losses lenders incur when loans default. Predicting loss is essential to quantifying the exposure investors and lenders face when backing loans. The natural next step a model that predicts loss given a loan has defaulted. For this model, we first define the loss-ratio at default as a target variable:

$$\text{Loss Ratio} = \frac{\text{Charge-off Amount}}{\text{Gross Approval Amount}}$$

Because our model examined loss at default, we restricted the dataset to only the loans that defaulted. We removed all variables related to gross approval and default, such as charge-off date, since these attributes are correlated to the loss ratio. Rather than using the approval date itself, we reused the loan duration attribute `DaysToDefault` from fitting our hazard models in the previous section as a relative time measure.

We estimated loss using a straightforward linear regression. It would make sense for loss to be a linear function of gross approval and loan term. Using the default least-squares model resulted in an R^2 value of 0.428 and a training MSE of 0.688. The test error was similar though slightly higher,

around 0.698. However, because the loss ratio is capped at 1.0, we expect the MSE to be small.

We attempted to perform random forest variable selection to further improve our model. However, we were unable to retrieve plausible results on our first several attempts and chose not to explore further.

5 Portfolio Selection

Loans are frequently securitized into *tranches*, certificates that represent the combined cash flow and risk of a large pool of loans. For investors, securitization reduces the variance of individual loans and promises more stable returns. This motivates understanding the risk and behavior of portfolios of loans in addition to individual loans.

To model pools of loans, we simulated several loan periods for a sample of loans and calculate the losses over those times. This process combines the models discussed in the previous sections, pipelining logistic classification, predicting time of default, and loss projection. We built a total loss distribution on a portfolio of 500 loans from the test set over a one- and five-year period using Monte Carlo simulation. Following that, we determined the value at risk (VaR) and conditional value at risk (CVaR) along with their confidence intervals. These numbers represent the amount of capital investors likely need to reserve to cover potential losses.

5.1 Selection of Loans

Following the methodology described above, we first drew a sample from a test set of the most recent 5,481 loans. Before simulating, we first ensured that the features and structure of the test set were consistent with the training sets for each of our models. We included all of the macroeconomic variables we added while again removing weaker categorical variables such as `CDC_Zip` and `ThirdPartyLender_City`.

We set the start date for both the one- and five-year periods to be the date of the most recent loan in the training set, February 1st, 2013, so that our sample periods end on February 1st, 2014 and February 1st, 2018 respectively.

When a loan defaulted, we checked that its default time fell within our time range and then considered whether it contributed any loss to our portfolio.

5.2 Monte Carlo Simulation

We used the hazard model to estimate both the probability of default for a loan and its approximate duration in days since loan origination. For each iteration of simulation, we followed the subsequent procedure, based on the inverse method:

1. Randomly select a set of 500 (unseen) loans from our test set.
2. For each loan L:
   (a) Generate a standard uniform variable U.
   (b) Use the hazard model to predict the probability density function of default probability for the loan, P_D.
   (c) Sample the default time as given by $P_D^{-1}(U)$.
   (d) If this time falls within the period, assume the loan has defaulted.
   (e) Estimate a loss for the loan if it defaulted using the loss model and zero otherwise.
3. Sum these loss values up to determine the total loss for the portfolio.

For purposes of estimating tranche loss later, we also kept track of the total loan amount from the 500 loans in each simulation.

5.3 Loss Distributions

We conducted $N = 1000$ simulations for both the one- and five-year periods and plotted histograms of the total losses, which can be found in Figures 9 and 10.

We expressed the total loss for each simulation in terms of tens of millions of dollars. We can see that both the one- and five-year periods follow approximately normal, with perhaps a slight right-skew.

A priori, we might expect a noticeable number of zero total losses in our simulation, at least over the one-year period. However, upon further examination of the test set, we noticed that the approval dates of the loans extend back as early as 2006. Given that, it seems even more unlikely that none of the loans in a sampled pool default.

Still, since that loan default is still an uncommon event, it is possible that our hazard model errs toward overestimating the number of defaults. However, this is just the reality of the data set. More loans do not default than do.

Period	Mean Loss Amount	Mean Loan Amount
1-year	2.2749	31.704
5-year	5.4217	31.633

Table 5: Loss distribution statistics.

We can observe from Table 5 that the average total loss over the one-year period is clustered close to 2.2, corresponding to 22 million dollars. The average total loss for the five-year period is slightly more than double that, around 55 million dollars. These losses are around 7% and 17% of the pool values, respectively. We estimate that the average gross approval for a sample of 500 loans from our test set is around $600,000.

5.4 Value at Risk

Regulatory authorities require financial institutions to reserve enough capital to cover for risks above a set risk threshold, usually 5%. The value at risk (VaR) of loan portfolios is important both for investors as well as financial institutions who securitize loans. Following our models above, we can calculate the value at risk and the average value at risk, also known as the expected shortfall. This allows us to more quantitatively measure the risk of pools of loans.

The VaR at some level κ for a position X is defined as

$$\text{VaR}_\kappa(X) = \inf\{m : P[m + X < 0] \leq \kappa\},$$

Essentially, the VaR is the minimum amount of liquid capital an institution needs so that the probability its position bankrupts it is at most κ.

A glaring disadvantage of VaR is that it does not appropriately account for rare events. For example, part of why so many financial institutions bankrupted when the housing bubble popped is that VaR doesn't appropriately account for such catastrophic events. This motivates also measuring the average VaR, or CVaR, which is a more conservative estimate. The CVaR is always at least the VaR.

We can calculate the VaR directly from our simulated loss distributions. Since the true average VaR requires taking the integral of the VaR over our discrete distributions, we estimated the average VaR by sampling VaR at several levels and averaging these VaR values.

Since we also want to determine the confidence intervals for our estimates of VaR and CVaR, this procedure is still insufficient. To calculate the confidence intervals, we used the following procedure:

1. Bootstrap $N = 1000$ total losses from the existing loss distribution.
2. For each bootstrap sample, determine the VaR and CVaR at the $\alpha = 95\%$ and $\alpha = 99\%$ levels.
3. Average the VaR and CVaR values to get the estimates for the risk measurements.
4. Find estimates for the standard error from the standard deviation of the VaR and CVaR values.
5. Compute the corresponding confidence bands.

Figures 11, 12, 13, and 14 display the VaR distribution at the 95% and 99% levels for both the one-year and five-year periods.

We obtained the following results:

Measure	α	$\hat{\mu}$	Lower	Upper
VaR	0.95	1.45436	1.38893	1.51978
VaR	0.99	1.24236	1.13431	1.35041
CVaR	0.95	1.31609	1.24879	1.38339
CVaR	0.99	1.15914	1.00058	1.31771

Table 6: 1-year VaR and CVaR estimates.

For the 95% level in Table 6, we see that the 1-year VaR is 1.45 with a confidence interval of [1.39, 1.52]. So we should reserve around this amount of capital to protect ourselves from losses stemming from loan defaults. Note that our estimates are in terms of tens of millions of dollars.

For the 99% level, we see that the 1-year VaR estimate is 1.24 with a confidence interval of [1.13, 1.35]. As we might expect, we have a wider margin of error but also a lower VaR value.

Our CVaR estimates at 95% and 99% are 1.32 and 1.16 with confidence intervals [1.25, 1.39] and [1.00, 1.32], respectively. The margin of error for CVaR at the higher α level is significantly greater.

Measure	α	$\hat{\mu}$	Lower	Upper
VaR	0.95	4.20169	4.062014	4.34137
VaR	0.99	3.78464	3.50761	4.06168
CVaR	0.95	3.94400	3.81717	4.07082
CVaR	0.99	3.64704	3.41189	3.88218

Table 7: 5-year VaR and CVaR estimates.

Now for the 95% level via Table 7, the 5-year VaR estimate is about 4.20 with a confidence interval of [4.06, 4.34]. For the 99% level, the 5-year VaR estimate is 3.78 with a confidence interval of [3.81, 4.07]. Finally, our CVaR estimates at 95% and 99% levels are about 3.94 and 3.65 with confidence intervals [3.82, 4.07] and [3.41, 3.88].

We observe from the histogram plots that our VaR distributions do not appear very normal. This means that our standard error estimate could be very inaccurate. We may need a larger number of simulations as well as a larger dataset to increase the accuracy of our VaR and CVaR intervals.

5.5 Securitization

Securitization, a practice that involves pooling together illiquid assets such as loans or mortgages, allows lenders, originators, and other entities to create securities in order to have a source of funding for said lenders and originators. These securities are backed by a cash flow pool, and both payment and loss are distributed via a waterfall structure, in which certain tranches are prioritized before others.

In particular, those in the senior tranche would receive payment from the cash flow first before the junior tranche and residual tranche, if applicable, and would also be penalized

last if losses occurred at some point, as in when a loan defaults for our case.

In order to assess the risk profiles of the investors for these tranches, we then explored the loss distributions we generated in more detail as regards to the two tranches: junior and senior.

5.6 Tranche Loss Distributions

We considered a [5%, 15%] junior tranche and a [15%, 100%] senior tranche and estimate the distributions for the one- and five-year losses of an investor who has bought into one of the two tranches.

From our 1000 simulations of total loss for each of the periods, we could also calculate the percentage of the loan amount lost to defaults for each pool of loans.

That is, we found the ratio

$$\text{Loss Percentage} = \frac{\text{Total Loss Amount}}{\text{Total Loan Amount}}$$

for every simulation.

If the calculated loss percentage was greater than or equal to the 5% quantile, this corresponded to a loss in the junior tranche. If the calculated loss percentage was greater than or equal to the 15% quantile, this corresponded to a loss in the senior tranche.

Moreover, based on the magnitude of the loss percentage, we were able to determine how much of the tranche was hit by the loss, i.e. a loss percentage of 20% implied the junior tranche had 100% of possible loss in its tranche, with the penalties spilling over to the senior tranche as well.

We plotted both the distribution of our tranche losses in addition to a cumulative distribution of the one-year and five-year tranche losses together to compare the junior and senior options. From Figures 15 and 18, there is a clear right skew for the 1-year loss in the junior tranche, biased toward a lower percent loss. The senior tranche, on the other hand, does not incur any penalty and has 0% loss for any of our simulations. This seems like reasonable behavior, as we would expect not too many loans should default in the one-year period to aggregate enough loss to affect the senior tranche.

Upon examining Figures 17 and 18 for the five-year period, however, we see an obvious shift in outcomes. Namely, the percent loss for the junior tranche has increased such that loss to some extent is practically guaranteed for the junior tranche and now skewed to the left toward 100% loss. The loss trickles further into the senior tranche where up to 10% loss may occur.

These distributions show that investing in senior tranches is much less risky than investing in junior tranches, as we'd expect. Figures 19 and 20 support this. We can observe that while the five-year period is more risky for both tranches, the senior tranche will not incur much loss even over the five-year period. The chance of any loss in one year is practically nonexistent based on our estimated distributions.

For the junior tranche, the risk not only increases in the five-year time frame, but the probability of total loss becomes nontrivial. Some amount of loss is likely even over a one-year period.

6 Conclusion

In this paper, we examined a number of different approaches to modeling loan default rates, loan default times, and loan risk. Understanding loans is vital for investors, lenders, and even borrowers, but so many factors influence the financial and institutional health of businesses that accurately predicting loan behavior is difficult. The mercurial nature of small businesses further complicates matters.

Our key conclusions are that careful attention to data is essential, even simple models can be quite powerful, and that it is more effective to combine several synergistic approaches than aim for a completely end-to-end model.

Despite being the most basic model for the scenario, our logistic regression model performed quite well after just minor tuning. Our best logistic model had a test accuracy of 0.7913, an F1 score of 0.405864, and a test AUC of 0.7618 —remarkable numbers for a simple model in such a complicated setting. Our primary innovation with the logistic model was weighting type I and type II errors by the proportions of true positives and negatives in the dataset when choosing the optimal threshold.

We then fit data to hazard model to predict time to default. We did this in two ways, using all the loans, including both default and non-default loans and using odefault loans only. We found that the most significant variables in both models to be `[ApprovalFiscalYear, TermInMonths, SP500_Yearly_Return, Log_GrossApproval_Norm, Log_HPI_Norm, ThirdPartyDollars_Norm]`.

Using a combination of Monte Carlo simulation and bootstrapping with our fitted hazard model and linear regression loss model, we were able to simulate a loss distribution for a randomly selected pool of 500 loans and then estimate the VaR and CVaR for a one- and five-year time horizon. The average loss in one year was around 22 million and more than doubled for five years. From the loan and loss amounts of the loan pool, we could observe the impact of losses in more detail on a hypothetical junior and senior tranche for the loans had they been securitized. From our observations, it was clear that there was much more risk for an investor to bear when in the junior tranche as opposed to the senior.

With our neural network, we learned that trying to incorporate too many covariates leads to a dysfunctional model. We recommend very carefully selecting inputs when training neural networks for messy, real-world data sets and making full use of the myriad optimizations developed in the past decade, such as Adam descent, dropout, and batch normalization. We also advise being careful of vanishing and exploding gradients. Our networks appeared to be training well until we paused and carefully examined their parameters and outputs, at which point we found that they really hadn't learned anything at all.

6.1 Future Work

For further research, we could have potentially better predictions for time of default by splitting our loans into multiple data points over several time periods and treating time-dependent macroeconomic variables dynamically rather than statically, from time of origination.

Due to concerns regarding underfitting, the large number of levels for some categorical variables such as zip code, and the desire to maintain consistency of data across the various models implemented, we had to omit certain data features such as `BorrZip`, `CDC_Zip`, etc., which could have added more granular, location-specific information to our models. Including this information would have required more computational power and a larger dataset.

We also considered modeling loan defaults for sample portfolios using an overall intensity [2] for each portfolio, representing the "arrival" of a default for any loan in the portfolio, rather than modeling the defaults of the loans in the dataset individually. We would then choose which loan defaulted via a weighted draw, weighting each loan by its relative intensity. This approach has the advantage of not being tied to a particular time frame, and is more in the intensity-based spirit of the hazard model. However, this approach is also more coarse and discounts time-dependent covariates like macroeconomic factors. Nonetheless, Duffie and Grleanu's approach is natural, mathematically satisfying, and very interesting. Had we had more time, we would have liked to implement both approaches and compare them. However, our existing hazard model implementation wasn't well-suited to this alternative.

We also could have further tuned and trained our neural network to try to finesse some real results from it. We could have tried more rigorous variable selection such as significance or random forest optimization with logistic or linear models, tuning our hyperparameters, or trying different kinds of regularization.

Ultimately, predicting loan performance is a protean and intricate task. We've explored a number of effective models and techniques in this paper, but there are an even greater number of approaches that we didn't take. We're humbled by the size of the task, and optimistic about the fruits of future research.

7 Code

The code for this project can be found in the following GitHub repository: `https://github.com/nalkpas/MSE246-2018-Project`.

References

[1] Davidson-Pilon, C. (2014). *Survival Regression*. Retrieved March 18, 2018, from Lifelines website: `http://lifelines.readthedocs.io/en/latest/Survival%20Regression.html`

[2] Duffie, D., & Grleanu, N. (2001, January). *Risk and Valuation of Collateralized Debt Obligations*.

[3] U.S. Small Business Administration. (2017, August). *Frequently Asked Questions about Small Business*. Retrieved March 15, 2018, from `https://www.sba.gov/advocacy/frequently-asked-questions-about-small-business`.

[4] Kingma, D., & Ba, J. (2014, December). *Adam: A Method for Stochastic Optimization*.

Appendix

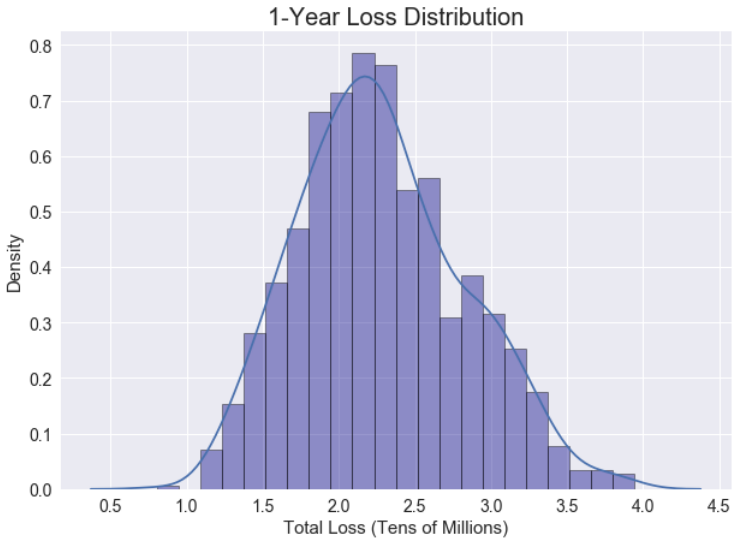


Figure 9: 1-year loss distribution.

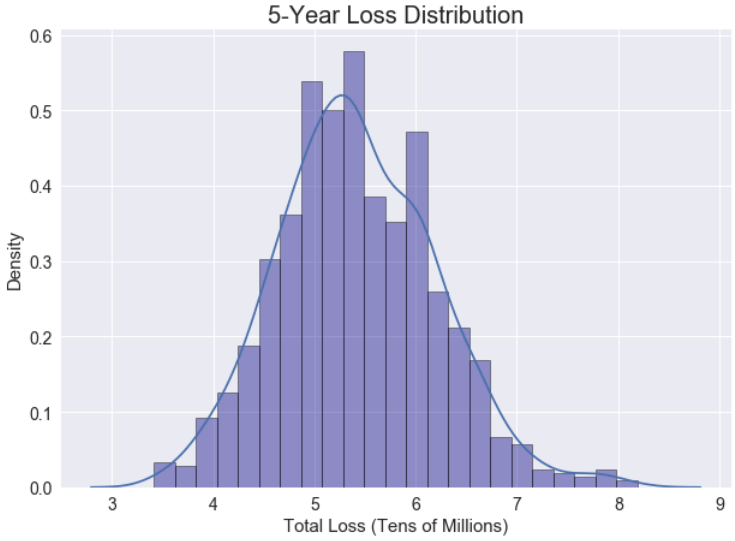


Figure 10: 5-year loss distribution.

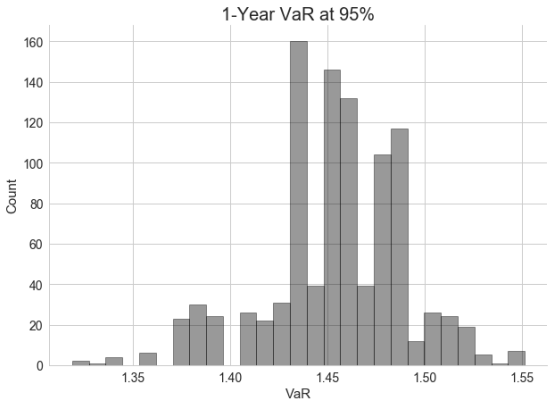


Figure 11: 1-year 95% VaR distribution.

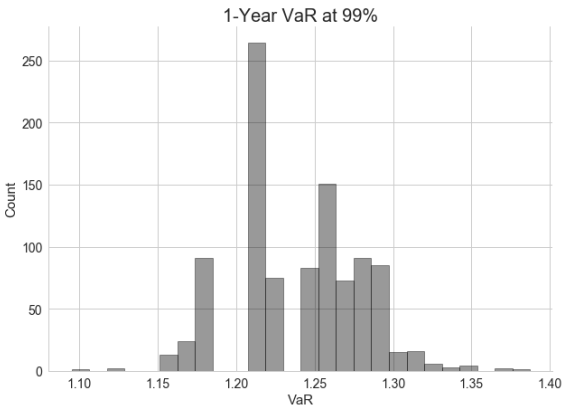


Figure 12: 1-year 99% VaR distribution.

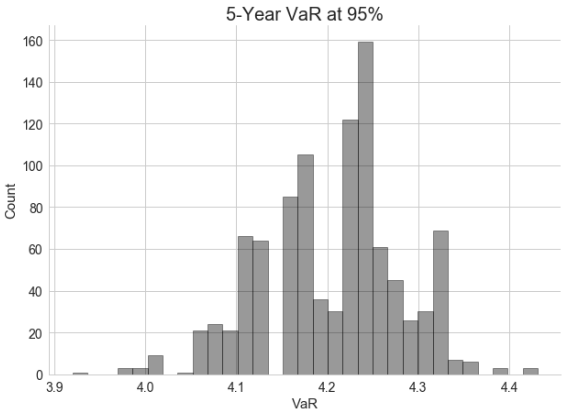


Figure 13: 1-year 95% VaR distribution.

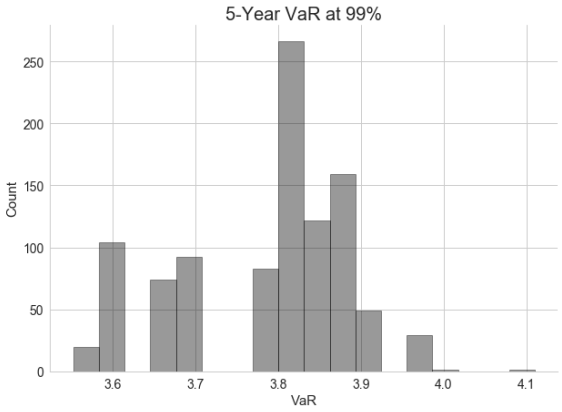


Figure 14: 5-year 99% VaR distribution.

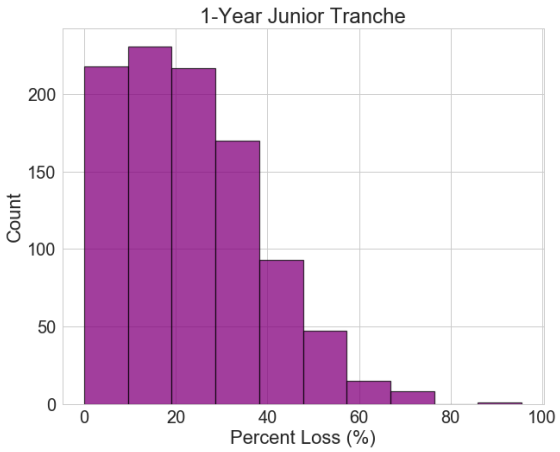


Figure 15: 1-year junior tranche loss.

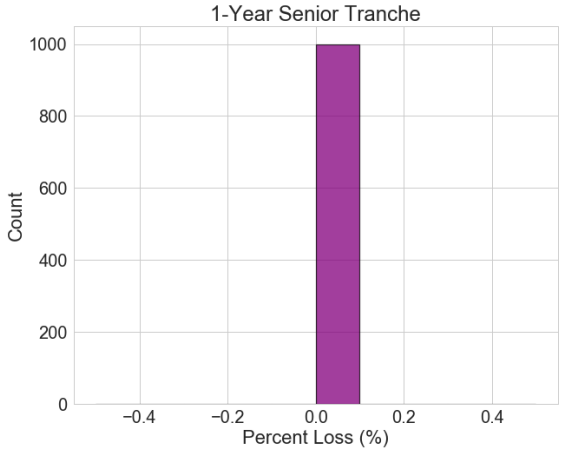


Figure 16: 1-year senior tranche loss.

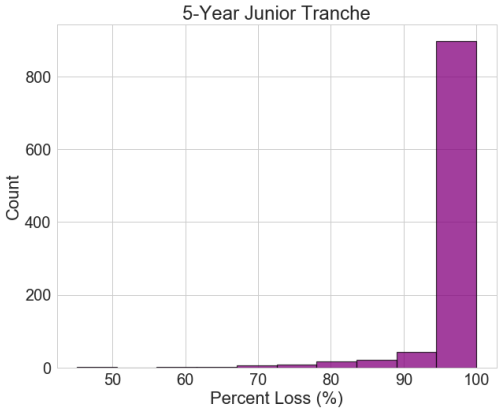


Figure 17: 5-year junior tranche loss.

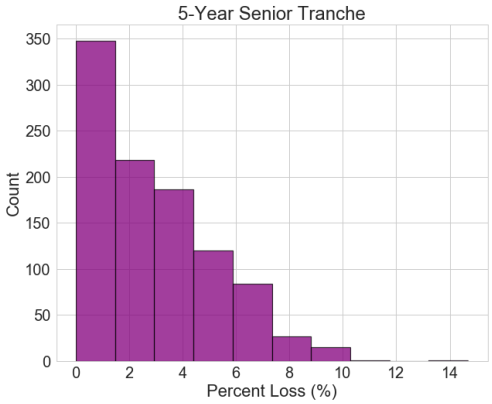


Figure 18: 5-year senior tranche loss.

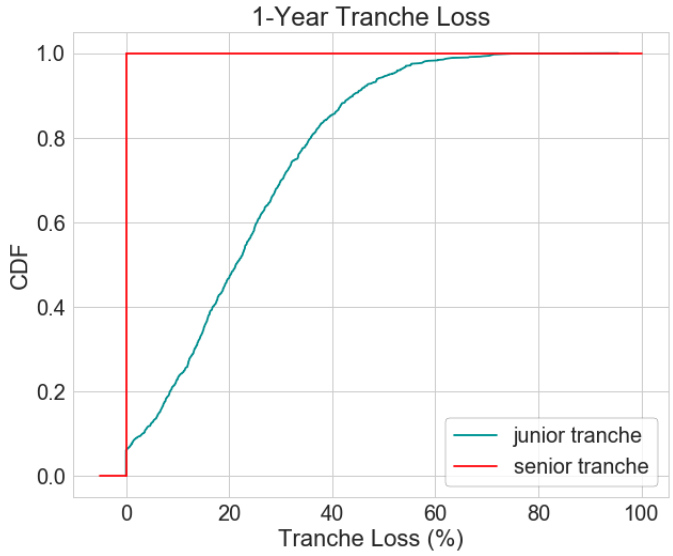


Figure 19: 1-year tranche cumulative probability of loss.

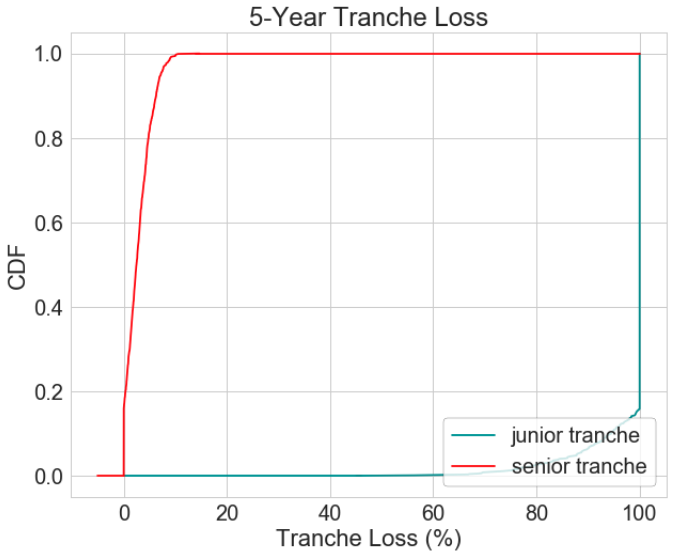


Figure 20: 5-year tranche cumulative probability of loss.