MATP-4910 Final Project Notebook

DeFi

Jaimin Vyas

Due 15 December 2021

Contents

Final Project: Github Info 1

Imports 2

Overview & Problems Tackled 2

Data Description 3

Results 3

Problem 1 3

Problem 2 6

Summary and Recommendations 13

Final Project: Github Info

- github repository: https://github.rpi.edu/DataINCITE/IDEA-Blockchain
- Your github ID: vyasj2
- Final notebook: dar_final_vyasj2_3nov2021.Rmd
- Summary of github contributions including github issues addressed.
  - https://github.rpi.edu/DataINCITE/IDEA-Blockchain/blob/master/DefiResearch/StudentNotebooks/Assignment03/vyasj2_assignment03.pdf
  - https://github.rpi.edu/DataINCITE/IDEA-Blockchain/blob/master/DefiResearch/StudentNotebooks/Assignment04/vyasj2_assignment04.pdf'
  - https://github.rpi.edu/DataINCITE/IDEA-Blockchain/blob/master/DefiResearch/StudentNotebooks/Assignment05/vyasj2_assignment05.pdf
  - https://github.rpi.edu/DataINCITE/IDEA-Blockchain/blob/master/DefiResearch/StudentNotebooks/Assignment06/vyasj2_assignment06.pdf
  - https://github.rpi.edu/DataINCITE/IDEA-Blockchain/blob/master/DefiResearch/StudentNotebooks/analysis.Rmd
  - https://github.rpi.edu/DataINCITE/IDEA-Blockchain/blob/master/DefiResearch/StudentNotebooks/analysisv2.Rmd
  - Issues #82 and #98

Imports

```
## Loading required package: readr

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: ggplot2

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: lubridate

## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

## Loading required package: survival

## Loading required package: survminer

## Loading required package: ggpubr

##
## Attaching package: 'survminer'

## The following object is masked from 'package:survival':
##
##     myeloma
```

Overview & Problems Tackled

AAVE is a decentralized lending protocol built on top of the Ethereum network. It supports many different cryptocurrencies, and its native internal currency is ETH, or Ether. In simple terms, it is a place for users to transact with one another. The key transactions are: * Borrow: Users borrow money from one another or from the lending pool, a large public "pool" of money that other users contribute to. When users borrow money, they have to overcollateralize their loan, meaning the amount of collateral they put in is worth more money than the initial loan. This is because prices of cryptocurrencies fluctuate very heavily, so this overcollateralization protects the lender. If the price of the total collateral drops below a certain percent

threshold of the amount borrowed, the loan can be liquidated, meaning a third party user comes and repays part of the original loan to the lender and takes an equal part of the collateral plus an extra fraction for free, with the fraction depending on what currency the collateral was paid in. When users borrow, there is also an interest rate that accrues on the principal amount borrowed. This is often referred to as the borrow rate. Provided they don't get liquidated, users can repay their loan in full plus interest, reclaim their collateral, and all is well. * Deposit: Users can also deposit money into the lending pool, and receive passive income for providing liquidity, which is a measure of how much of each coin exists in the lending pool. This is known as yield farming. After an amount of time, the user can redeem however much money they deposited into the lending pool plus whatever interest has accrued in that time.

There are two main types of cryptocurrencies: stablecoins and non-stablecoins. A "stablecoin" is a cryptocurrency whose value is directly pegged to that of a real-world currency. The stablecoins that we analyze are ones whose value is equal to that of the US Dollar. In other words, 1 coin = 1 USD. The value is not always exact, but the difference is mostly negligible (~0.005 USD error). A non-stablecoin is simply a cryptocurrency that is not a stablecoin.

In this paper, I walk through two types of analyses; borrow rate and survival rate analysis. Borrow rate analysis was performed using rolling average calculations for borrow rates in AAVE, and visualizing how the rates shifted over time. Survival rate modeling was performed using Kaplan-Meier curves, along with some median survival rate calculations.

Data Description

The data was queried from TheGraph API, a website that hosts decentralized protocol data and regularly updates their databases. The original transaction data had 745,612 observations with 34 features. There were 51,421 unique users, with 6 stablecoins and 47 non-stablecoins. The transaction dates ranged from November 30th, 2020, to October 17th, 2021. The entire dataset consisted of 109,770 unique borrows, as well as 6731 unique liquidations.

To prepare the data for survival analysis, I had to split the original dataset into all of the "borrow" transactions, and all of the "liquidation" transactions. Then, I merged the data on user id, so that we would get a dataset that contained all of the borrow data and all of the liquidation data for each user. Then, I just took the time difference between the borrow date and the liquidation date, and that was the "age" of the loan. After performing survival analysis on this, and then using different strata, there were some interesting results.

Results

From the survival analysis, I found that for all borrows through the entire dataset, the probability of a loan surviving for more than 200 days is about 35%. For specific cryptocurrencies, the behavior is a bit different. For the borrow rate analysis, I found that after China's "crypto ban" in April of 2020, stable and variable borrow rates went from extremely unpredictable and variable to a screeching halt, becoming more or less constant, or fluctuating around a solid average as opposed to oscillating between very high and very low.

Problem 1

We will start with the borrow rate analysis. As explained earlier, borrow rate refers to the percent interest that a person taking out a loan has to pay on the principal amount to the lender of the loan. These rates fluctuate based mainly on the utilization rate of the coin the user would like to borrow, which is a fraction determined by the total borrows of that coin divided by the total liquidity (amount available in the lending pool). The problem that I was aiming to solve here was to understand how the borrow rates fluctuated aside from amount available in the lending pool, and what other kinds of factors aside from utilization rate could be used as predictors.

Methods

To analyze how borrow rate fluctuates aside from utilization rate, I plotted a 21-day rolling average of the percent borrow rate on the y-axis over time, to visualize the trends. To get a representative sample of the data while also cutting out the majority of noise, I only plotted the borrow rates of stablecoins, because users borrowing stablecoins made up more than 80% of the borrow data. I filtered by stablecoins because non-stablecoins did not have any sort of noticable change similar to how stablecoins did.

Results

We see the fluctuation in the figures below:

```
## Warning: Removed 20 row(s) containing missing values (geom_path).
```

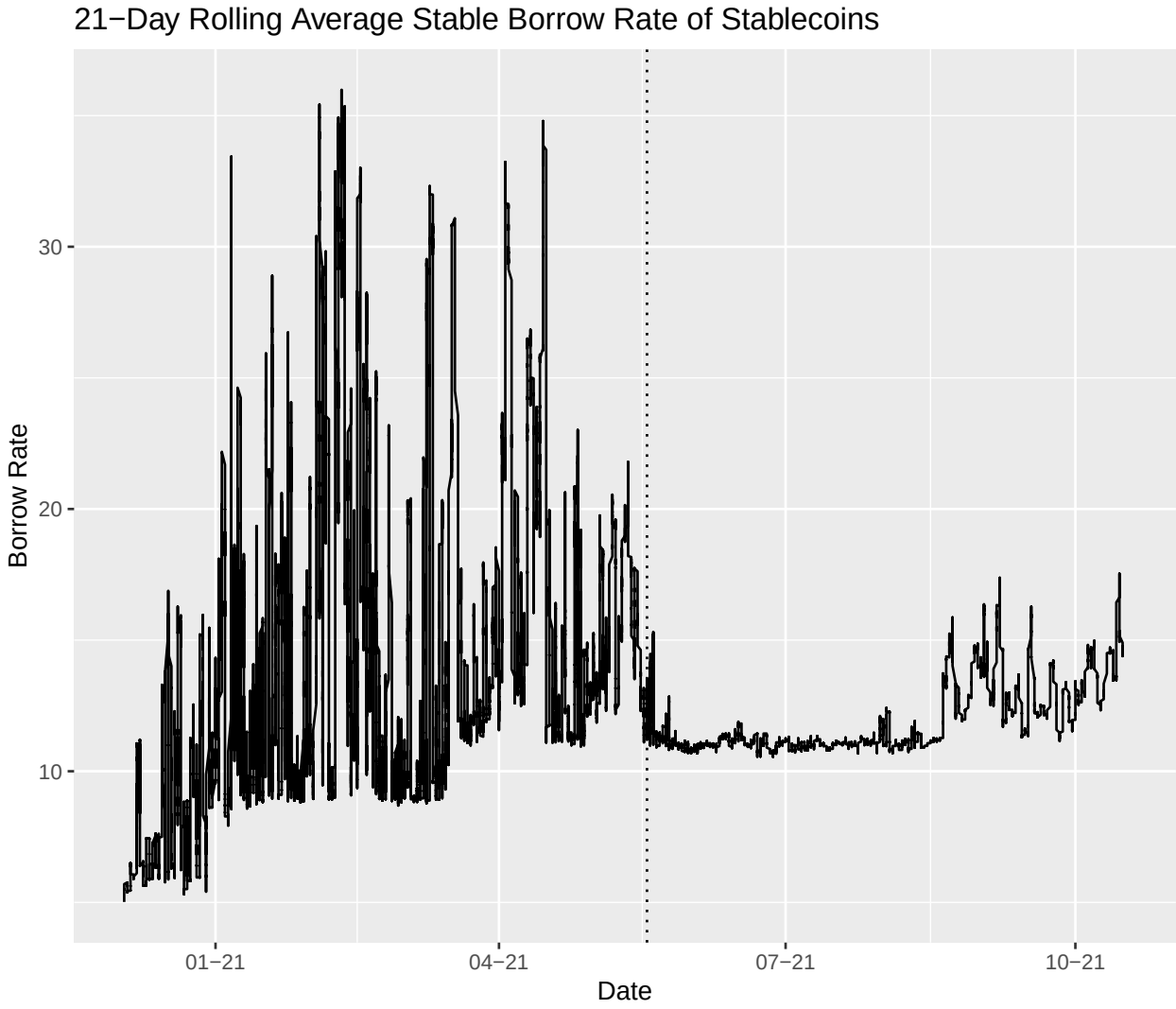

Figure 1: Rolling Average of Stable Borrow Rate

Figure 1 shows the 21-day rolling average of the stable borrow rate of USD-pegged cryptocurrencies. We see that this average has a few hitches, one around the beginning of January 2021, where the floor jumped from about 6% to 9% on average. Then again in late March of 2021, where the minimum again jumped from about 9% to 11%, where it then fluctuated wildly until around mid-May of 2021. After this date, for a solid three months, the 21-day rolling average of the stable borrow rate was almost consistently 11%. It started to move again in late August of the same year, but nowhere near as wildly as it had before.

After doing some digging, it appeared that China, arguably the biggest player in the cryptospace, announced they would be banning all cryptocurrency transactions sometime during that date of May 2021. This date is the dotted vertical line in both Figure 1 and Figure 2.

```
## Warning: Removed 20 row(s) containing missing values (geom_path).
```

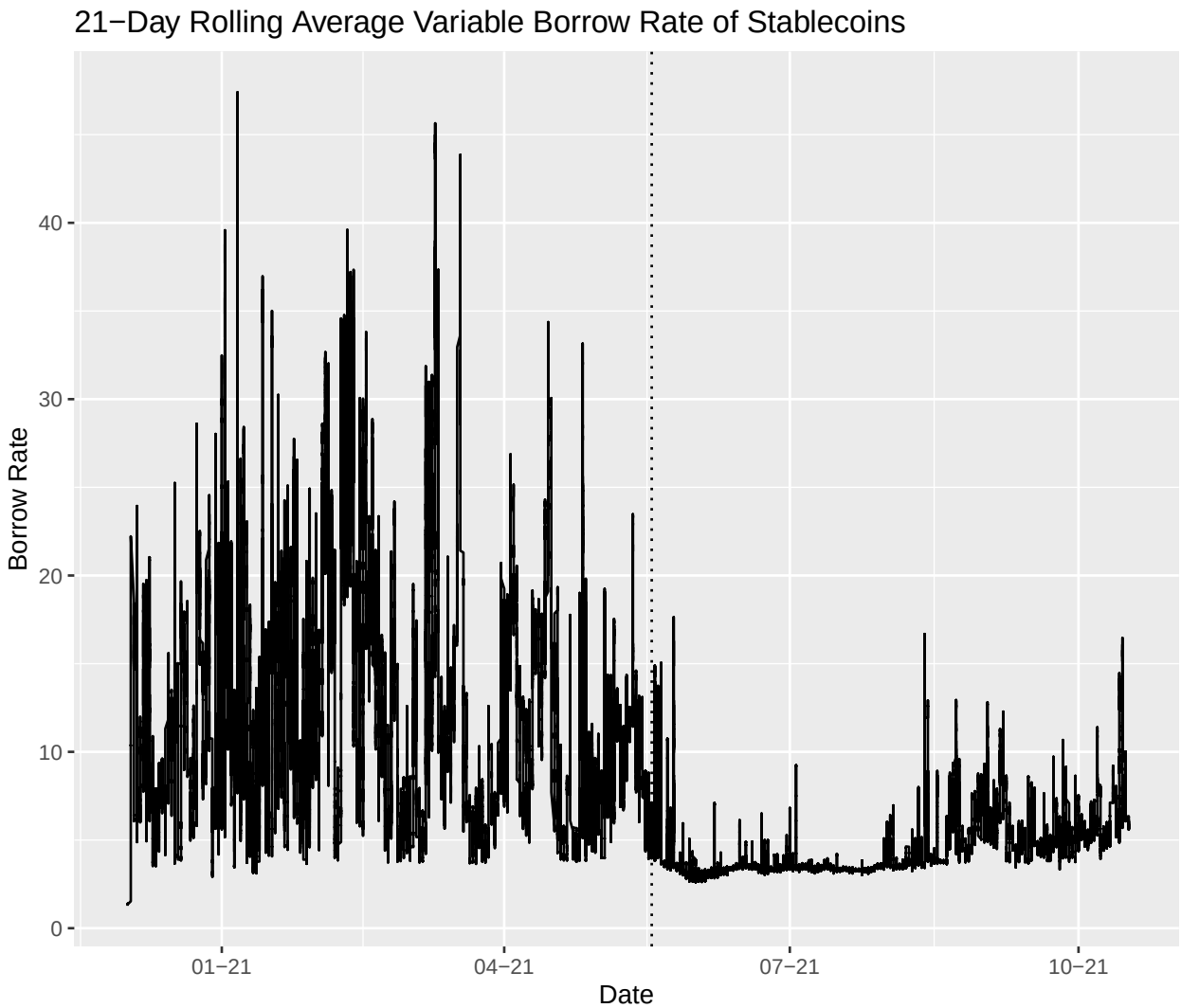

Figure 2: ROlling Average of Variable Borrow Rate

In Figure 2, a similar trend can be seen with variable borrow rates of USD-pegged cryptocurrencies. This time, there was no jump in the floor of the borrow rate, but around a similar time of mid-May of 2021, the 21-day rolling average variable borrow rate settled to about 4% for a similar period of time as the 21-day rolling average stable borrow rate.

Discussion

In both figures above, the dotted line represents the China crypto ban. As seen in Figure 1, the stable borrow rate for stablecoins settled around 11% almost immediately after the China crypto ban, and in Figure 2, the variable borrow rate settled around 4%, with some minor jumps every now and then. They both were revitalized after about 2 months, and went back to regularly fluctuating, though nowhere near as wildly as they were before the ban.

The findings of this analysis depict how the borrow rate for stablecoins is very heavily impacted by world events. This tells us that the utilization of stablecoins drops during these periods after "shock events", but not for non-stablecoins. The impact of China's crypto ban was not noticeable at all for cryptocurrencies whose value is not pegged to a fiat currency.

From this we can deduce that many users stopped borrowing USD-pegged coins, because utilization rate is determined by the fraction $U = \frac{TotalBorrows}{TotalLiquidity}$. Thus a low utilization rate implies a low total amount borrowed and a high total liquidity. Perhaps USD-pegged coins are not used in foreign markets as much, and thus when China announced the ban, many users dropped their usage of USD-pegged coins and focused on non-stablecoins to get all of their assets out before the prices crashed.

Problem 2

Survival modeling is a statistical method of analyzing the time it takes for an event to occur from a given starting event. A probability curve is then fitted on the data to represent the model's prediction on what the chances of the second event occurring are at a given time. For the transaction data that was given, I performed these kinds of analyses using the Kaplan-Meier model on the time it takes for a user to borrow, and then liquidate on that borrow.

Liquidations are the cryptocurrency version of defaulting on a loan. Essentially, when someone lends you money, you would provide the lender some amount of collateral. However, because cryptocurrency prices are so volatile, you need to overcollateralize your loan, meaning you have to put in more ETH worth of collateral than you actually initially borrowed. When the collateral's total value drops below a certain fraction of the principal amount borrowed, your loan gets liquidated.

Methods

For this problem, to initially sort the data and extract the correct features, I first performed a left join on the borrow data and liquidation data. This yielded a dataframe that contained all borrows that either did or did not liquidate. These records were joined on user ID, so that each user's borrows and liquidations were joined into the same dataset. To perform survival analysis, the two main features that are needed is "age", which is the time from the first event to the second event (in this case, time from borrow to liquidation), and "status", which is just a binary value representing whether or not that specific user liquidated, 1 if yes, 0 if no liquidation was on record. Because the timestamps given in the dataset were given in Epoch time, which is measured in seconds, taking the difference and then dividing by 86,400 gives us the number of days that the loan lasted before liquidation. The "age" for this analysis was calculated by each user as the following: $\frac{\min(LiquidationTimestamp) - \min(BorrowTimestamp)}{86400}$. This approach would give the time difference for the first borrow to liquidation for each user, which was the only way to calculate the "age" variable, since transaction ID was not given.

Results

The results were interesting enough, because sorting by borrow rate and collateral type yielded some different graphs that provided more insight on how the conditions of the borrow (borrow rate mode, principal reserve, collateral reserve) can impact the survival of a loan. Because there are many different reserves to stratify survival by for both collateral and principal of the loan, we first make a barplot that shows the counts of each reserve used for collateral and principal to determine the most used reserves, to get an accurate depiction of the data.

We see the barplots below:

In Figure 3, the barplot shows that the reserves LINK, WBTC, and WETH all have more than 15,000 records of a user using one of those coins as collateral for a borrow, so we will use those strata when plotting survival on collateral reserve.

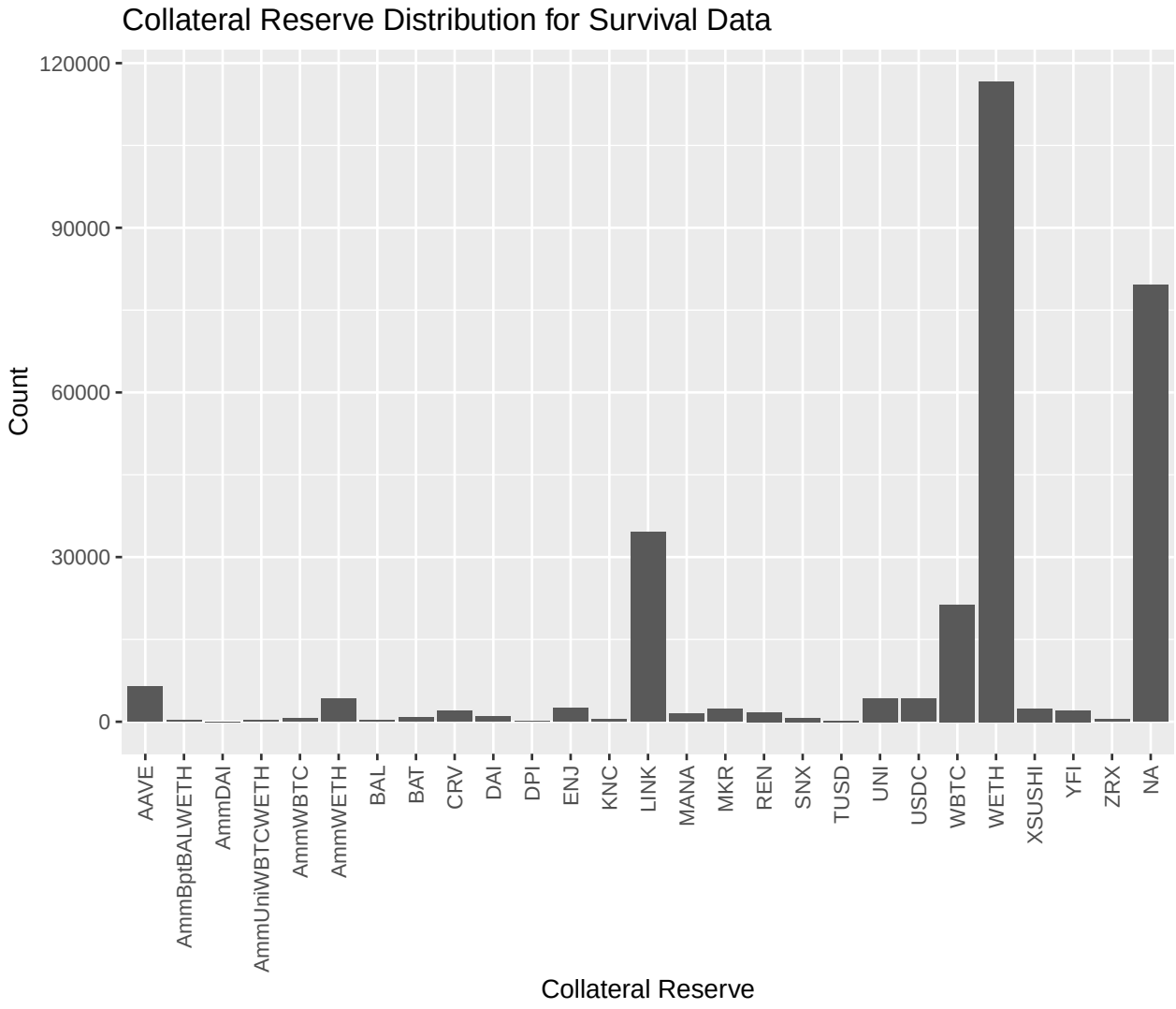

Figure 3: Count Distribution for Collateral Reserve

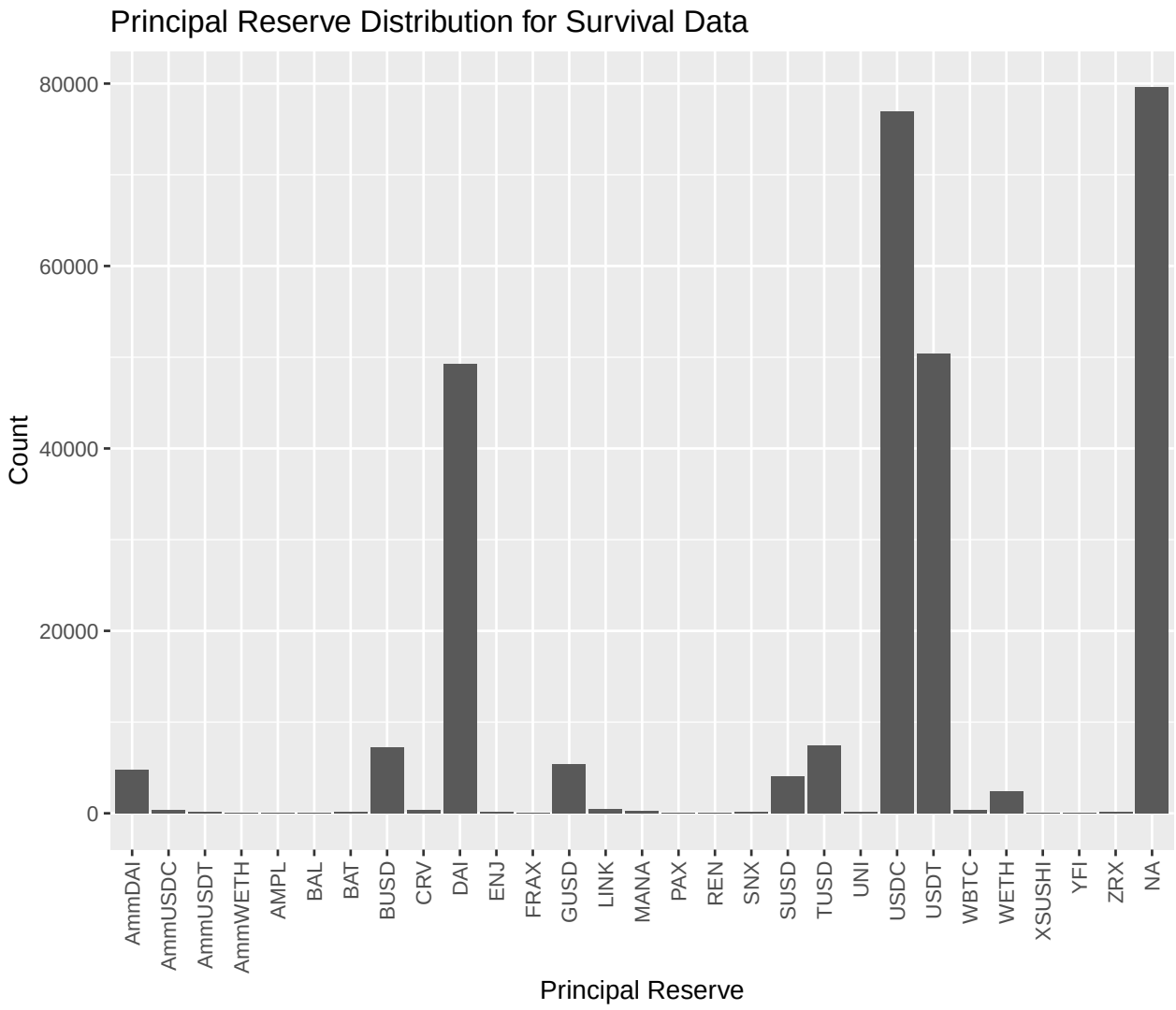

Figure 4: Count Distribution for Principal Reserve

The barplot in Figure 4 shows that DAI, USDC, and USDT are by far the most used for principal reserve, meaning that the vast majority of the time, users borrow stablecoins. We will use these three reserves as strata when plotting survival on principal reserve.

We now move on to the plotting of the survival curves.

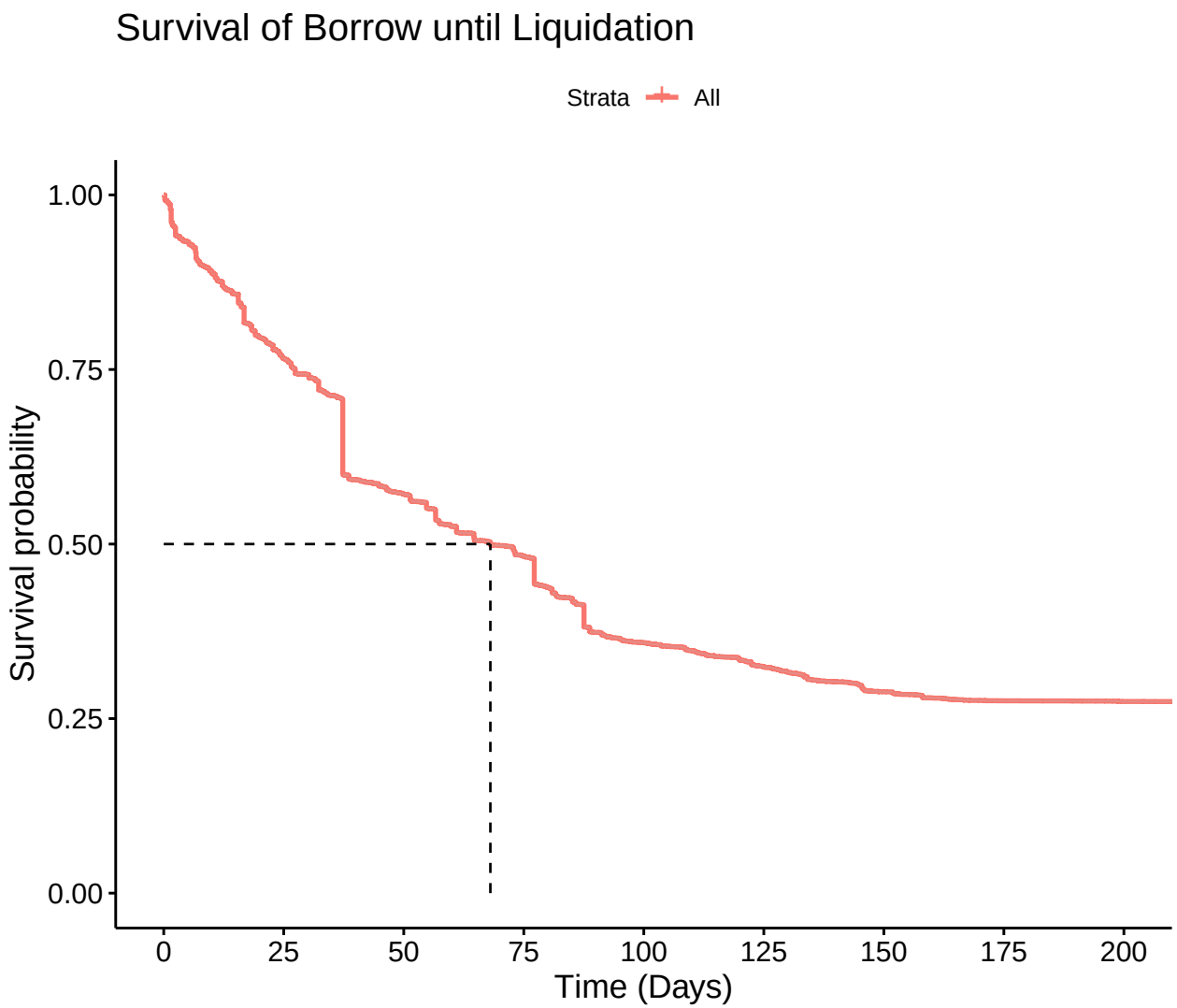

Figure 5: General Survival Model

Figure 5 above is a general survival model of the survival data that was calculated above, we see from the dotted line that after about 70 days, the probability of a loan not liquidating drops below 50%. This tells us that keeping a loan for more than 70 days is more of a risk than repaying the loan quickly. We also see that there is a straight vertical line at about 35 days, this could be the mass liquidation event that was triggered by the China crypto ban. A flat vertical section of the survival curve is a visualization of a large number of records ending, or in this case, a large number of liquidations.

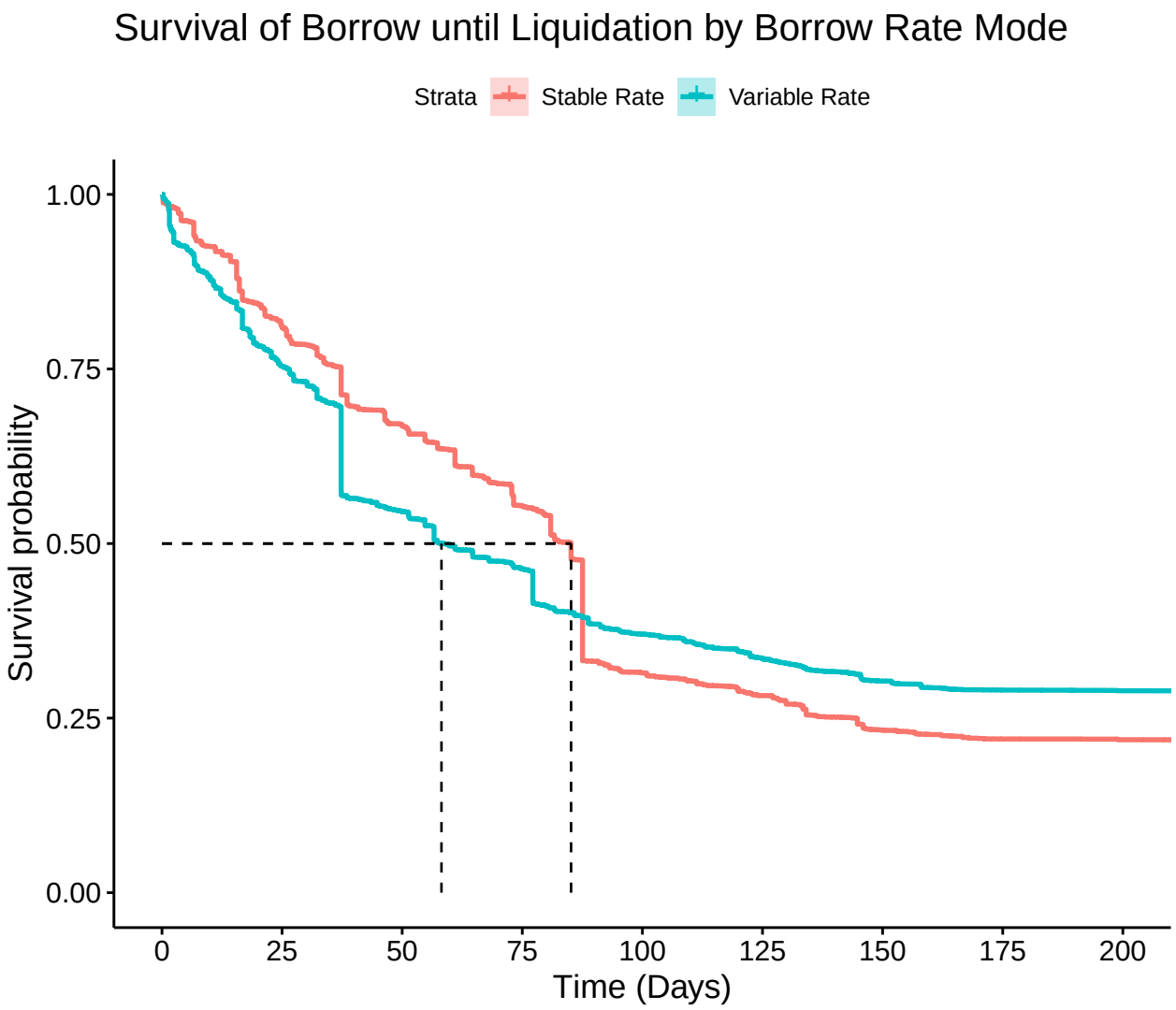

Figure 6: Survival Model by Borrow Rate Mode

Figure 6 is a survival curve where we split the data into two subsections: stable borrow rate and variable borrow rate. This graph displays that after almost 90 days, the survival rate of stable rate borrows drops below the survival rate of variable rate borrows. This follows the standard line of thinking that in the long run, a stable borrow rate is worse than a variable borrow rate, because in the long run, a stable borrow rate will yield a higher amount interest due.

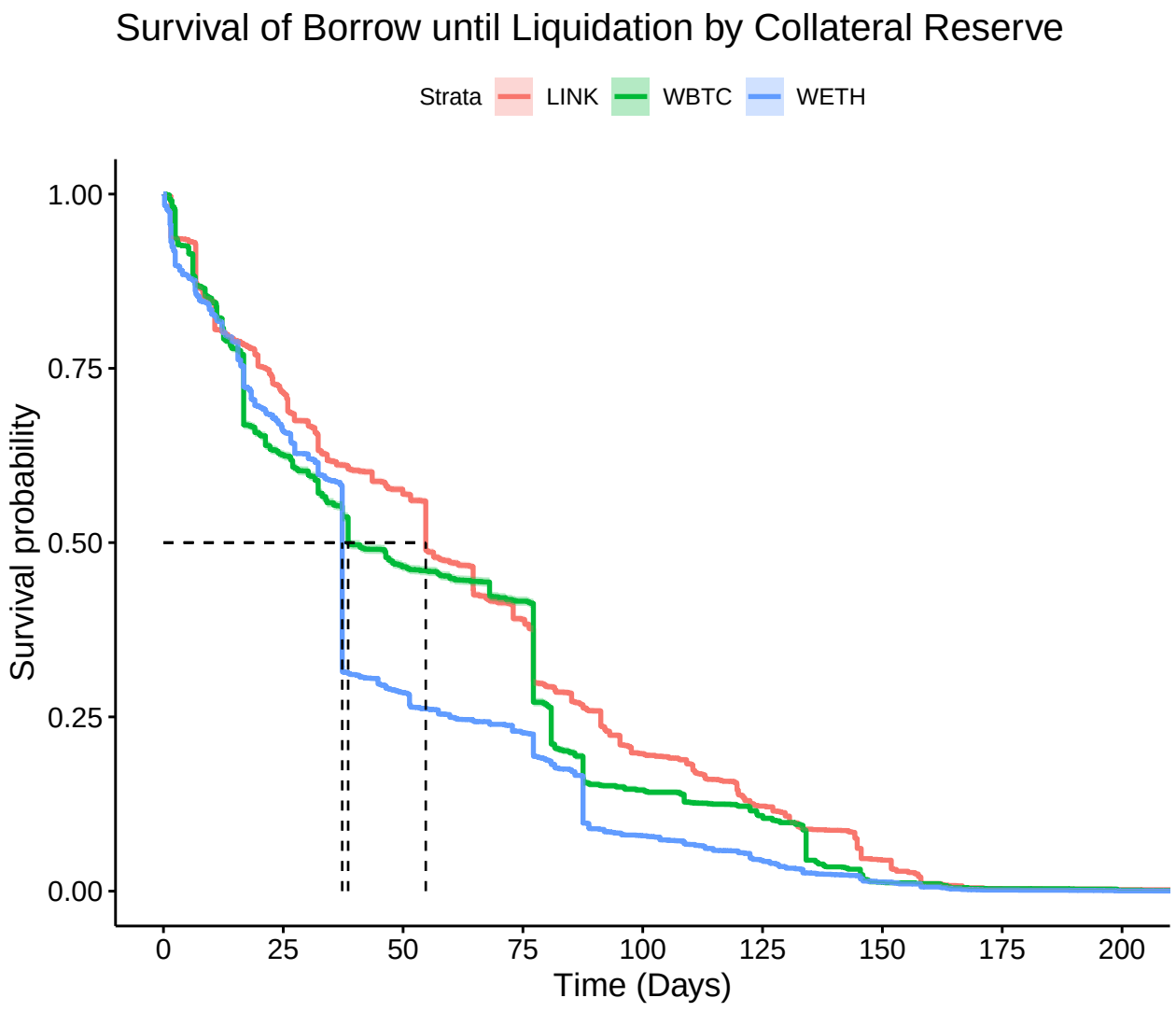

Figure 7: Survival Model by Collateral Reserve

Figure 7 shows survival curves on collateral reserve, using the three reserves that were found to be most commonly used as collateral from Figure 3. The plot shows that LINK is the best reserve to use as collateral, as the probability of the loan not liquidating becomes less than 50% after a longer period of time as opposed to the other two reserves. We also see that the curves all merge at the end, where the probability is 0. This means that all records in the data liquidate after 200 days, so if you were to borrow using one of these reserves as collateral, don't forget about your loan!

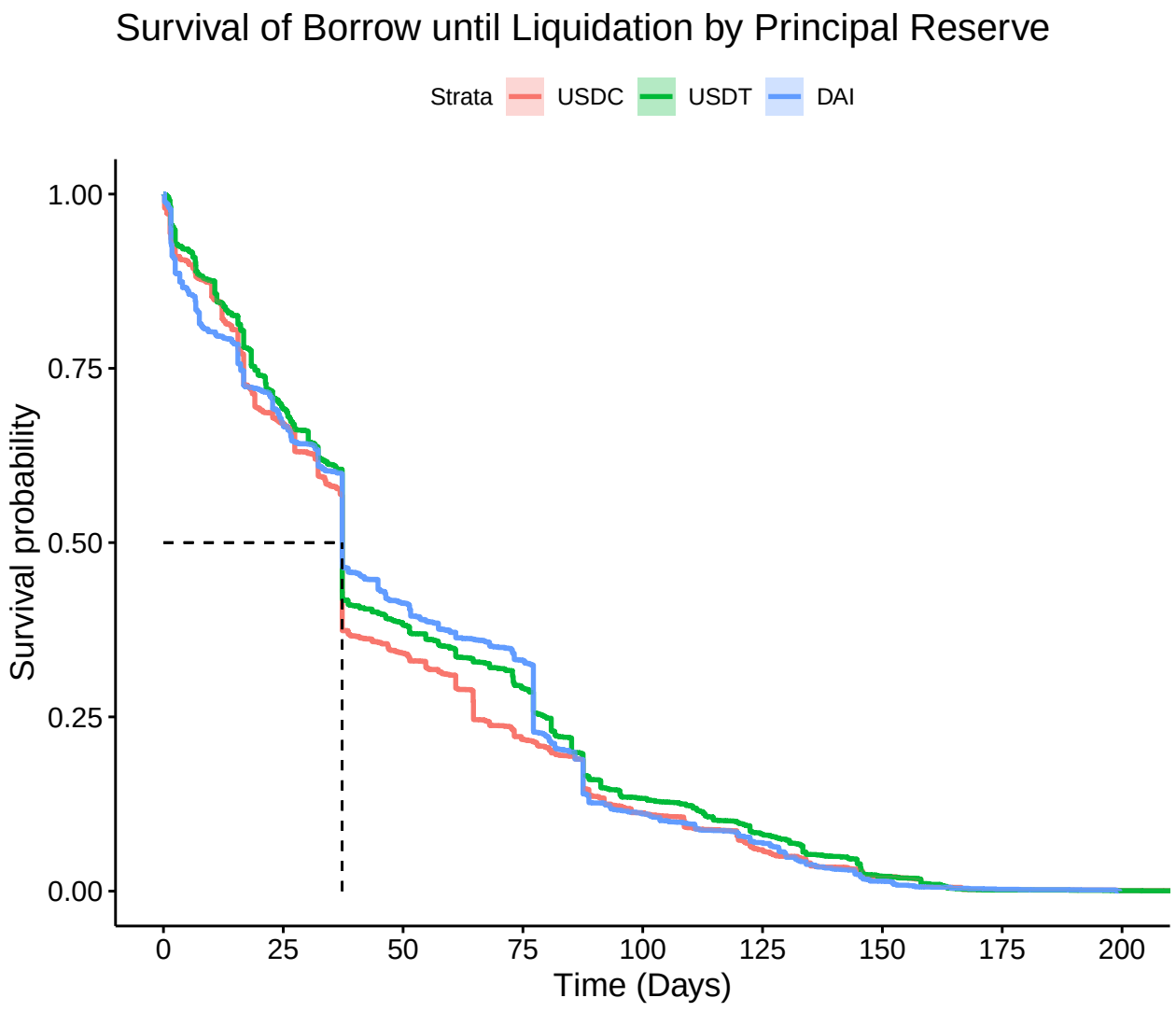

Figure 8: Survival Model by Principal Reserve

Figure 8 above shows survival curves on principal reserve, meaning the coin that was actually borrowed. This graph tells us that the stablecoins actually behave almost identically when it comes to borrows, and each of the three coins have a median survival at almost 40 days. It is interesting to see the solid vertical line at the same x-point of about 40 days, because that most likely is a representation of the China crypto ban causing the mass liquidation event, as seen earlier in Figure 5. The borrow rate analysis was also done only on stablecoins, so this is very likely to be the case.

Discussion

I chose to do survival analysis on the coins that I did because as can be seen from the bar graphs, they are the most used coins for collateral reserve and principal reserve. From Figure 5, we can see that almost 25% of loans survived after 200 days.

However, when we split this on borrow rate modes in Figure 6, we see that variable borrow rate loans tend to survive more in the long run than stable borrow rate loans. In the short run however, we see the opposite behavior. This leads to the conclusion that after about 2-3 months, variable borrow rate loans have a higher chance of not liquidating than stable borrow rate loans. This is interesting, it could be because of user behavior, or the fact that the coins that users borrow using stable rates are more volatile than those they borrow using variable rates. Further analysis involving time from borrow to repay could also be of interest

here, because stratifying on borrow rate mode would tell us which borrow rate type, stable or variable, causes users to repay earlier than the other.

In Figures 7 and 8, we see some familiar behavior in the survival graphs stratified on the most common reserves used for collateral (Figure 7), and principal (Figure 8). From this we see that the principal reserve does not seem to have an impact on liquidation, but the best reserve to use as collateral is LINK, as loans with LINK as collateral seem to survive the longest.

Summary and Recommendations

For future studies, I think analysis on borrow rates and different factors impacting liquidation would prove very useful. Predicting borrow rates using factors like sentiment, usage, and news to track market events would prove to be a very good approach,

I think a more in-depth version of the analysis seen in this paper could be included in a formal paper or an expanded app, as I believe the analysis would need to be more specific, since it might only apply to a very general portion of the population using AAVE's lending protocol. For example, in an app, users who want to borrow a certain cryptocurrency, they could input the coin they're borrowing and the amount, as well as the coin they deposited as collateral along with that amount, and the app could tell them their loan's predicted survival rate for the next week based on previous data, similar to what was done in this paper, as well as using price data for the collateral coin to predict what the price would be over the next week, to give an even better estimate of survival probability.

Decentralized finance is a fascinating up and coming field, and there are a plethora of research opportunities, many of which could have huge impacts on user behavior and could create a convention for how users perform transactions. With blockchain technology most likely taking over in the forseeable future, it would be very valuable to have a deep understanding of how these computer-driven decentralized networks facilitate user interactions.