# 1 Top down (Macroeconomic) Approach

Residential mortgage-backed securities (MBS) are the largest and most liquid securitized asset class in the world. Banks, insurance companies, and money managers invest in MBS because they provide an attractive yield relative to U.S. Treasury securities with comparable credit risk. However, unlike most fixed income securities, which have specified contractual coupon and principal payments, the timing and amount of MBS cash flows is uncertain. Prepayment risk, which impacts the yield and interest rate risk of an MBS comes from five sources:

- Rate/term refinance, which occurs when borrowers lower interest payments or shorten the term of the current mortgage.
- Cash-out refinance, which involves extracting equity from a home.
- Involuntary buyouts, which in the case of agency MBS result in an early return of principal. However, the timing is contingent on the GSEs or GNMA loan services.
- Curtailments, which are partial prepayment of the full payoff before maturity.
- Turnover, which is caused by geographic migration and home upgrades. Turnover creates a baseline level of prepayments that are highly seasonal.

The COVID-19 crisis and the response by the Federal Reserve resulted in low-rate environment that elevated both levels of refinance and buyout activity. Since the initiation of quantitative tightening, sustained inflation, and rising interest rates, refinance activities have slowed considerably. At present, a major question affecting the source of prepayment is now turnover, curtailment, and buyouts. As a baseline level of prepayment activity, understanding turnover is important to evaluating the risk of mortgage-backed securities. The goal of this part of the project is to use macroeconomic data to quantify and predict turnover prepayment speed.

## 1.1 Data and Feature Engineering

In this part we describe the engineering of our top-down mathematical for turnover. The data and predictors that we use are macroeconomic features of the mortgage market and the US economy.

We define a proxy for turnover which we call *implied turnover*. We compute the implied turnover quarterly by dividing the average monthly home sales by the number of available home units in that period. For example, in 2008 Q4 there were 5.1 million home sales per month in the US and a total of 91.5 million available home units, providing an implied turnover of 5.6% for that quarter. A time series for implied turnover during 1999 - 2021 is provided below. We say that this quantity is a proxy for turnover because a home sale event generally implies a prepayment event of a mortgage associated to that home. Housing sales and a stock are provided by US Census data.

The two strongest predictors for implied turnover in our model are

- Home price appreciation
- Housing credit availability index

The *home price appreciation* (HPA) is measured as the year over year percentage change in the *house price index* (HPI). We computed HPA using purchase-only HPI calculated from the data provided by FHFA and collected from FRED's website. A timeseries for HPA plotted against a timeseries for implied turnover demonstrated the strong correlation between these quantities.

To strengthen the model's use of HPA as a predictor, we considered moving average of HPA and lagged HPA. An $n$-quarter moving average HPA is the average of HPA over the previous $n$ quarters, and is advantageous to use because this brings in a memory effect in to the model. We also experimented with the lagged version of HPA. An $n$-quarter lag of HPA is simply the value of HPA $n$ quarters ago, and it's advantageous to use if implied turnover takes some time to respond to the change in HPA then this can capture that effect. Of these newly feature engineered versions of HPA, we found that a 6-quarter moving average has the best correlation with implied turnover.

The *housing credit availability index* (HCAI) measures the difficulty of getting a mortgage in the United States by precisely quantifying lenders' tolerance for risk. We used the HCAI data provided by the Urban Institute. As with HPA we considered $n$-quarter moving average and $n$-quarter lags of

HCAI also and found that correlation between with implied turnover is the strongest when there is no lag or moving average attributed to HCAI.


Figure 1: HPI and HCAI plotted with implied turnover

## 1.2 Summary of the Model

The two plots above seem to suggest a strong correlation between HPA and HCAI with implied turnover. Also as we mentioned in the last section that after performing all the feature engineering with them we found that the 6-quarter moving average of HPA and HCAI work best for predicting the implied turnover. So we built a linear regression model for predicting implied turnover with those two as our predictors. Mathematically speaking our model is:

$$T(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

where $x_1$ = 6-quarter moving average of HPA, $x_2$ = HCAI, and $T(x_1, x_2)$ denotes the predicted implied turnover.

After training our model on our data sets we found:

- $\beta_0 = 4.475$ = intercept, with standard error 0.017,

- $\beta_1 = 0.125$ = sensitivity coefficient to 6-quarter moving average of HPA, with standard error 0.011,

- $\beta_2 = 0.124$ = sensitivity coefficient of HCAI, with standard error 0.013.

In the plot below we've plotted the implied turnover along with the prediction of the implied turnover along with our model. The purple, green, and red curves on the right denotes prediction of our model in the 95% confidence interval.

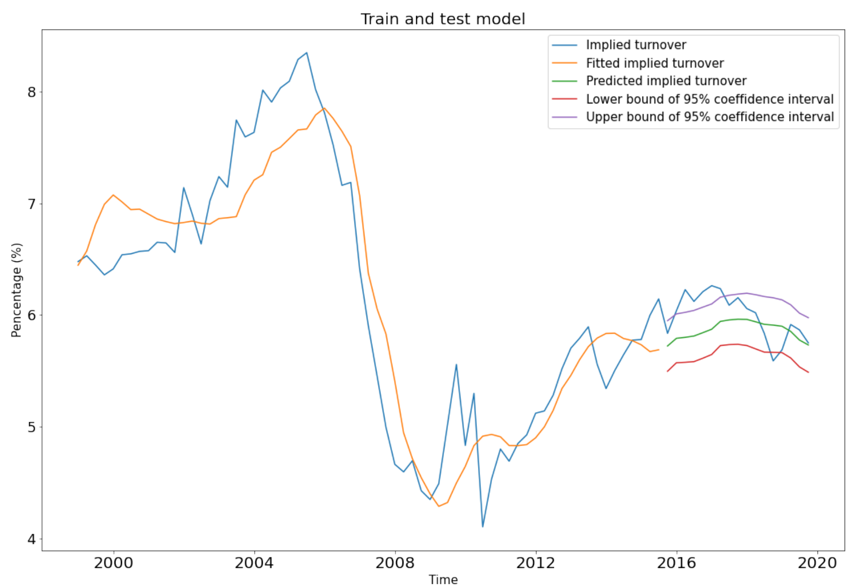



Figure 2: Performance of the Model

## 1.3 Future Work

A major goal of this project is to model and predict turnover, or the activity of people selling their houses. People sell their houses for a variety of reasons. We want to predict the speed at which turnover is happening in a certain period of time looking at some macroeconomic predictors. For instance our analysis in last two sections suggest that when home price appreciation increases the turnover increases, which intuitively makes sense because if home price appreciation goes up, value of the property goes up and so they get more incentive to sell the house. Similarly when housing credit availability index goes up it means it is easier for people to get credit and hence home selling/buying activity should increase. Our analysis above seem to agree with that too.

Intuitively, another macroeconomic factor that can affect turnover is *mobility*. The more people move from one place to another, the more home buying/selling activity should occur. So it is reasonable to include mobility as a predictor in the model. During this project, we modeled mobility by considering the annual change in the percentage of US homeowners, data provided by the US Census Bureau. We feature engineered this quantity for lags and moving averages, and found some positive correlation with implied turnover, but was not as strong as the correlation provided by HPA and HCAI, and so we did not include mobility as a predictor in our linear model. Additionally, we performed the same analysis by restricting to the change in percentage of homeowners above 65 years of age, and saw even weaker positive correlations with implied turnover.

It's possible that the proxy that we used for mobility is not robust enough to describe turnover, captured by the following example. Consider that the home sale of two homeowners to each other clearly would drive home sales, and thus, implied turnover, but would contribute nothing to the change to the percentage of US homeowners as we have defined mobility. With more time, we could construct a more sophisticated quantification of mobility to incorporate into our linear model to predict turnover.

# 2 Bottom Up (Loan Level Data) Approach

## 2.1 Cleaning the Noise

Since our goal is to study turnover, we targeted loans that have a negative incentive. In order to distinguish from cash-out, which can still occur when incentive is negative, we focused on very negative incentives. When incentive is very negative, we expect to see a larger presence of turnover.

One issue with studying loans with a very negative incentive is that there is not many of them. This can create noise and was the first issue we addressed.

When we aggregated the data and split into various cohorts based on credit scores and incentive, loans with a very negative incentive had low item counts. These low item counts can result in noisy data, that are often visualized as spikes in the time series depiction of the data. An example of this can be seen in Figure 3 at 2020 for the green line.

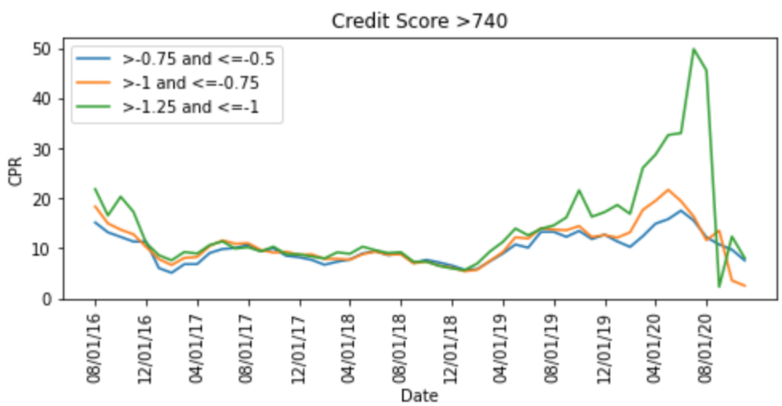



Figure 3: Time series of credit score > 740

Before we can address how to balance the good signals (not noisy) and the weak signals (noisy) to eliminate the noise, we first must identify the noisy locations. We attempted two approaches to finding noise. The first was to simply “eye-ball” them using time-series of the various credit score buckets and incentive buckets. However, since low item count is a good indicator of noise, we created algorithm 2 that flags the dates with an item count below a tolerance prescribed by the user. For each of the incentive buckets, we examined the mean and 25% quartiles to help determine what the prescribed tolerance should be for each of the incentive buckets.

After identifying the noisy data, we replace it with an approximation of a good signal. We chose to do this by analyzing the differences in CPR between different incentive buckets in the signals six months before and six months after the noise location. From these surrounding differences, we fit a polynomial of a degree three. From this fitted polynomial we approximated what the difference of the noisy incentive bucket and each of the other incentive buckets should be at the noise location. The new CPR was computed by adding the average of the approximated expected differences across all well-behaved buckets to the average value of the well-behaved buckets at the noisy location. This process was done by Algorithm 1. An explanation and code of Algorithms 1 and 2 are given in the next section.

## 2.2 Explanation and Code of Algorithms

The Noise Splicing Algorithm 1 has a goal of taking in dates that are considered noisey and outputting an estimate of the expected non-noisey value.

To accomplish this we first look at the data surrounding the noisey portion be an amount prescribed as an input by the user. In these surrounding areas, we collect the differences between the "well-behaved" buckets and the noisey bucket. After collecting the differences in the surrounding areas, we use this information to fit a polynomial across the whole interval, including the noisey portions. We can then use this polynomial to approximate the approximated expected difference of the noisey bucket from each of these well-behaved buckets.

The final approximation to replace the noisey data is made by adding the average of the approximated expected differences across all well-behaved buckets to the average value of the well-behaved buckets at the noisey location.

The Noise Splicing Algorithm given in 1 was designed with the thought of "eye-balling" the noisey data and inputting the dates collected manually. However, there is an additional algorithm, Low Item Count Finder 2, that will automatically output the dates associated with items that have low item counts. Where low is prescribed by the user in the form of a tolerance level. Since low item counts can be a large factor in noisey data, being able to easily locate the dates at which this occurs is important to removing noisey data.

## 2.3 Creating Cohorts

After the noise had been reduced in the data, the next step was to segment the data into the various cohorts. This was done in two ways. For the first

**Algorithm 1** Noise Splicing Algorithm

**Inputs:** Left and Right Interval Lengths, Noise Date Range, Dataframe with the noisey data and with the non-noisey data

**Output:** Approximation for a replacement of the noisey data

Given the noisey data date range, find the noise locations
**for** Each of the dates in the noisey data range **do**
  **for** Each of the non-noisey data **do**
    Find the difference of noisey and non-noisey data on the left interval
    Store location in $x$ and difference in $y$
    Find difference of noisey and non-noisey data on the right interval
    Store location in $x$ and difference in $y$
    **end for**
  **for** Each of the non-noisey datasets **do**
    Fit a polynomial, $P$, to the differences, using stored $x$ and $y$
    Evaluate $P$ at the noise location
    **end for**
  Calculate the average of the difference approximations found in the evaluation of $P$, call this the new noise difference
  Take the average of values the well-behaved data at the noise location and add to the the new noise difference, call this new noise
  noisey dataframe(noise location) = new noise
  **end for**
Return: Noisey Dataframe

**Algorithm 2** Low Item Count Finding Algorithm

**Inputs:** Dataframe, tolerance, buffer

**Output:** Dates associated with low item counts

Initialize: dates
Drop the first ‘buffer’ amount of rows and the last ‘buffer’ amount of rows
Noise Data = Dataframe( where ‘Item Count’ $<$ tolerance)
**for** i in the length of Noise Data **do**
  noise = Noise Data(’Date’)(i)
  append noise to dates
  **end for**
Return: dates

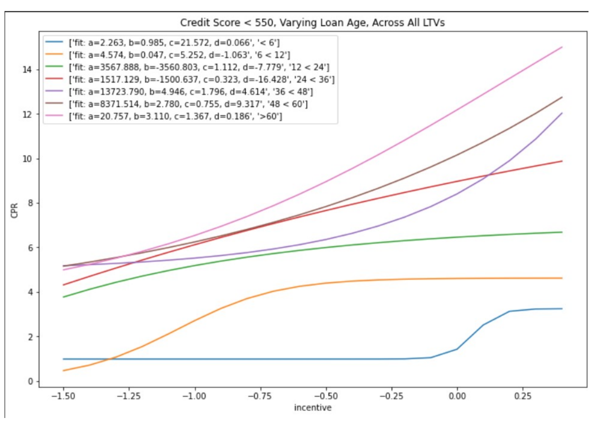
approach, we broke the data down into four credit score buckets, then further into seven loan Age buckets, and those further into seven buckets based on the Loan to Value Ratio (LTV). The second approach reversed the rolls of loan age and LTV. By breaking the data into such fine cohorts, the goal is to be able to more easily ascertain the motivation behind the prepayment for the borrowers when incentive is very negative.

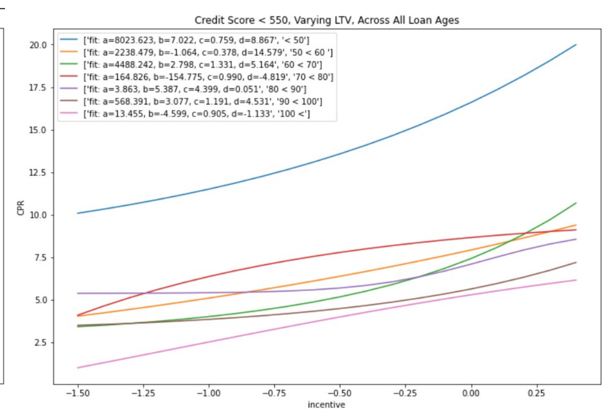
## 2.4 Fitting S-Curves

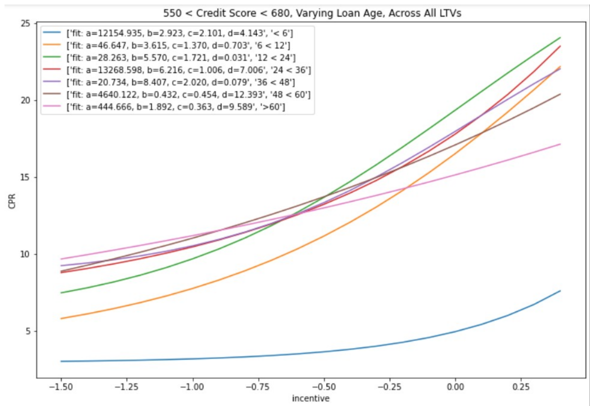
After creating the cohorts of data, we created scatter-plots of the cohorts which we fit logistic curves to. In order to fit to these curves, we optimized the coefficients to minimize the error from the points on the scatter-plot to the function

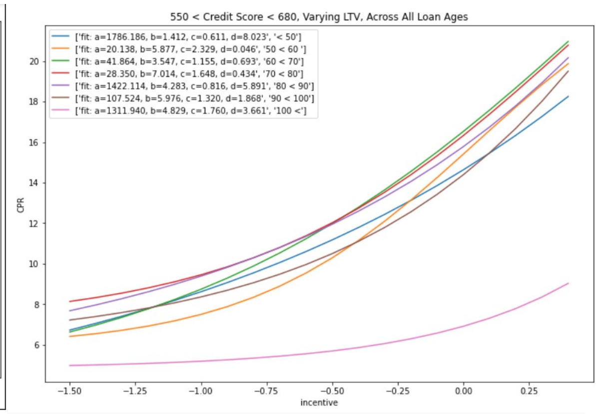
$$f(x) = \frac{a}{1 + e^{-c(x-d)}} + b.$$

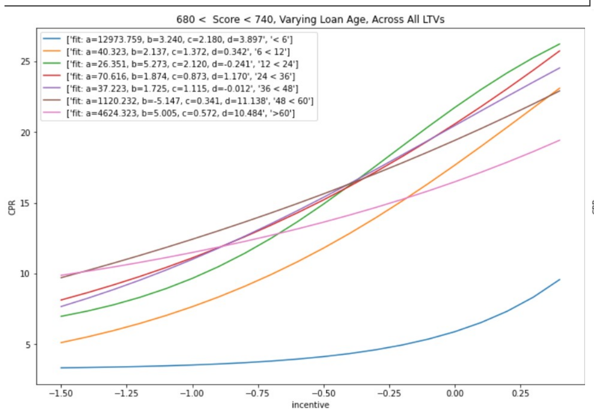
These logistic curves allow for a predictive CPR to be calculated from incentive for each of the cohorts. The scatter-plots are shown below with the scatter-plots from the first approach on the left and the second approach on the right.

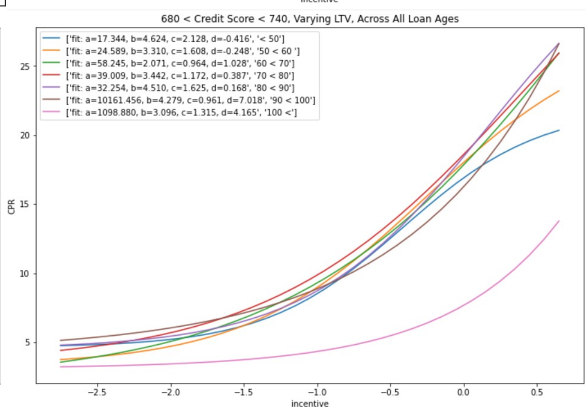


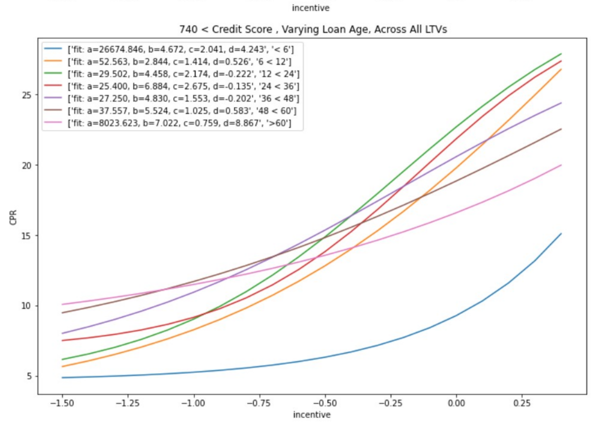



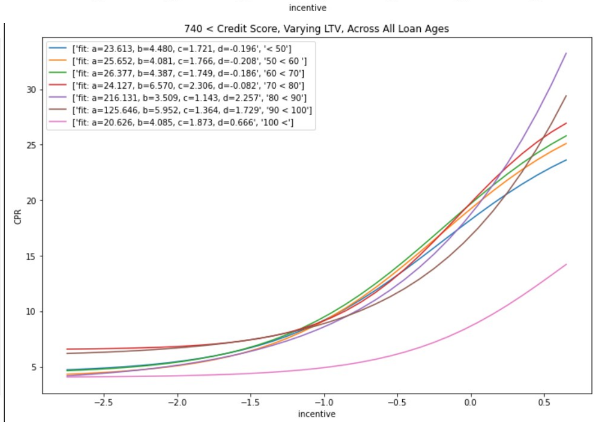



For each of these given fitted curves, we may calculate what we expect the CPR to be for given cohorts at various incentive rates. Examples of a few of these calculations can be seen below.

| Credit score Range | LTV Range | Loan Age | Predicted CPR @ Incentive -1.0 | Predicted CPR @ Incentive -0.5 |
|---|---|---|---|---|
| 680 < 740 | ——— | < 6 months | 3.540 | 4.132 |
| 680 < 740 | ——— | 6 - 12 months | 7.662 | 11.802 |
| 680 <740 | ——— | 24 - 36 months | 11.112 | 15.211 |
| 680 < 740 | ——— | 36 - 48 months | 11.007 | 15.392 |
| 680 < 740 | 60 - 70% | ——— | 8.960 | 13.144 |
| 680 < 740 | 70 - 80% | ——— | 9.291 | 12.930 |
| 680 < 740 | 80 - 90% | ——— | 9.856 | 13.630 |

# 3 Conclusion and Moving Forward

By cleaning the noise out of the data and separating the data into their various cohorts, we have been able to find models to describe the tail of the S-Curves that focus on the very negative incentives. By focusing on these negative incentives and modeling them, we have gained a better understanding of the behavior of prepayment when loans have a very negative incentive which have a larger turnover presence. From our models it appears as though loan age has a larger impact than LTV on the growth rate of the CPR compare to incentive. However this could be a consequence of how the data was aggregated and requires further investigation.