Mortgage Loan Project Report

Baiting Gai, Jiatian Xie, Ziyi Zhang

Content

1 Background

The data we used comes from a credit risk-sharing program launched by Fannie Mae. Fannie Mae released an extensive dataset beginning in 2013 that provides insight into the credit performance of a portion of Fannie Mae's single-family book of business. It consists of nearly 22 million records and the dataset provides monthly loan-level detail and is offered to help investors gain a better understanding of the credit performance of a portion of single-family loans owned or guaranteed by Fannie Mae. The public dataset includes a subset of Fannie Mae's 30-year, fixed-rate, fully documented, single-family amortizing loans that the company owned or guaranteed on or after January 1, 2000.

Fannie Mae provides the Fixed-Rate Mortgage (primary) dataset that contains a subset of fully amortizing, full-documentation, single-family, conventional fixed-rate mortgages. The data includes two files, 'Acquisition' and 'Performance' file. The 'Acquisition' file includes static mortgage loan data at the time of the mortgage loan origination and delivery to Fannie Mae. The 'Performance' file provides monthly performance data for each loan, from acquisition up until its current status as of the previous quarter. The dates of the data range from 2006 to 2016, and the loans usually maturate within 15 years to 30 years.

In our project, we aim to cluster the loans according to the borrower and loan characteristics and analyze the differences between the groups. We assume that the borrowers' credit scores will be different in each cluster.

2 Exploratory Data Analysis

In this part, we did some exploratory data analytics to better know our data and use this information to do feature engineering and feature selection.

We first checked the geographical distribution of loans (Figure1). The map shows that California has the most loans, which account for 14.48% of orders. Since there are too many states and the distribution is not even, we choose not to use this feature when we build models.

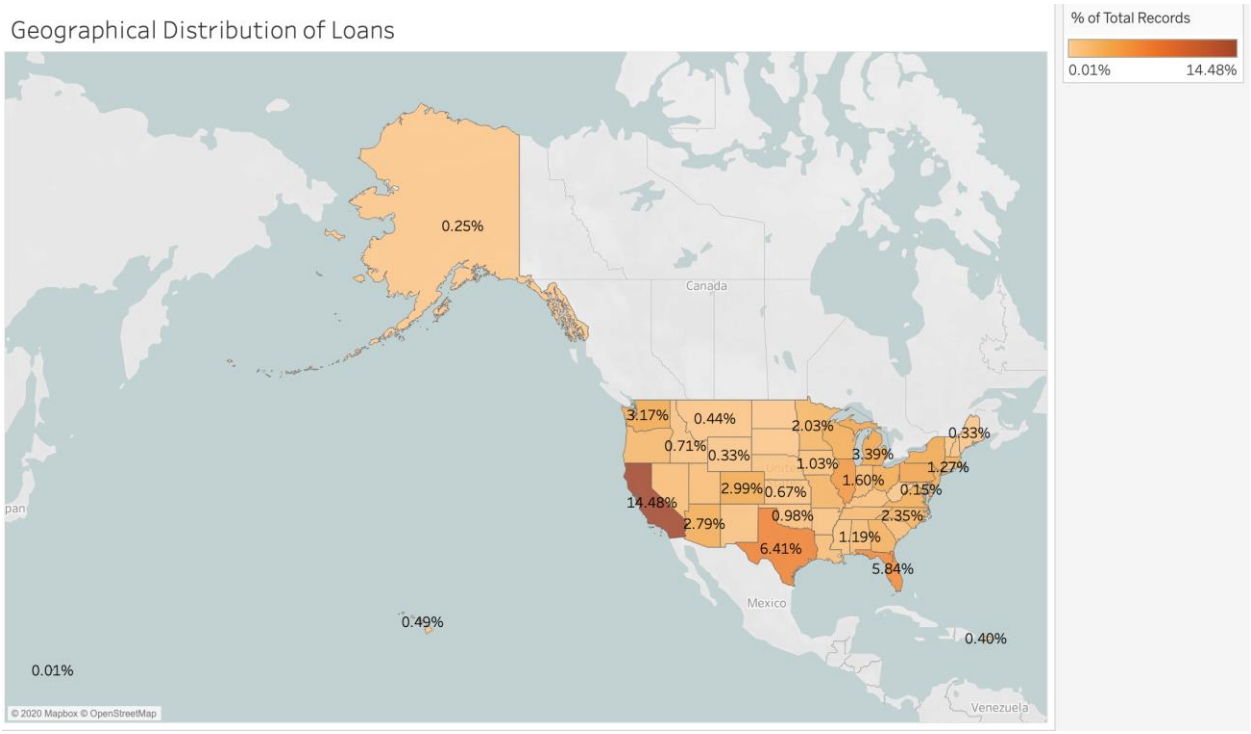

Figure 1: Geographical Distribution of Loans

Secondly, we found some features are imbalanced, which means that too many values in these features are the same. Figure 2 shows that all the values in the Modification Flag are all N. Figure 3 shows that over 90% of the loans come from the 'other' category. Figure 4 shows that over 90% of the loans come from the '0' category, which means that current or less than 30 days past due. Figure 5 shows that most transfer flag categories are 'N' or 'None'.

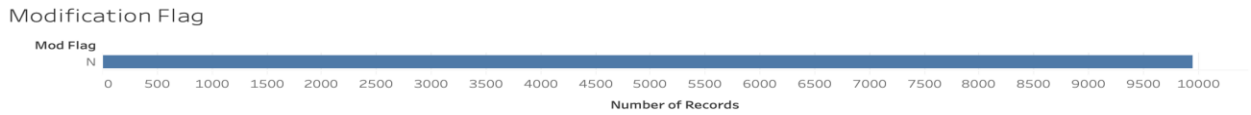

Figure 2: Modification Flag

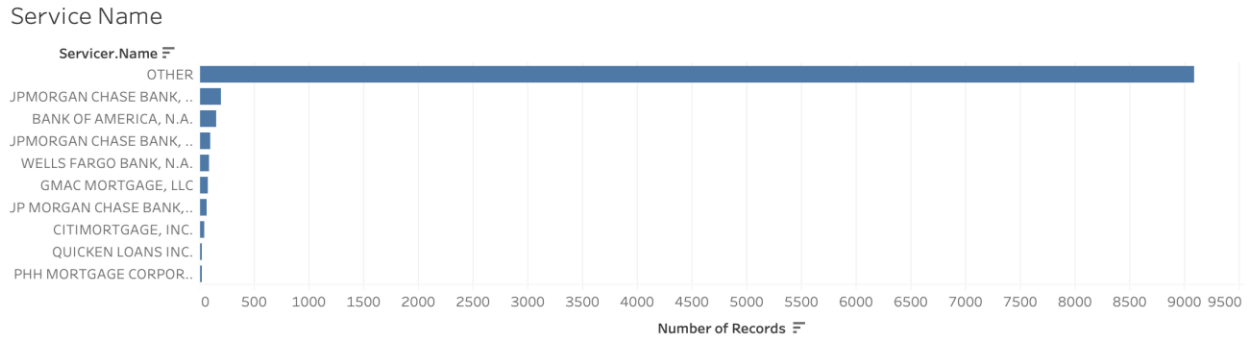

Figure 3: Service Name

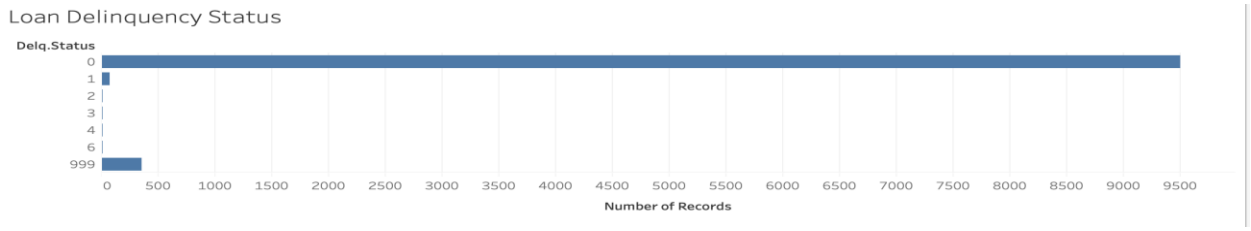

Figure 4: Loan Delinquency Status

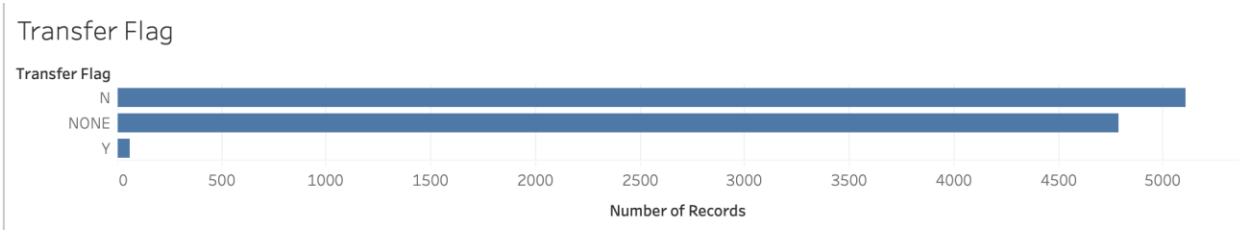

Figure 5: Transfer Flag

Thirdly, we found there are too many features related to money and we want to see the differences among them. From Figure5, we can see that almost all the features related to UPB show the same distribution and it is close to a normal distribution. There exists multicollinearity. Thus, we will just keep one variable, the 'original amount', which means the original amount of the mortgage loan as indicated by the mortgage documents.

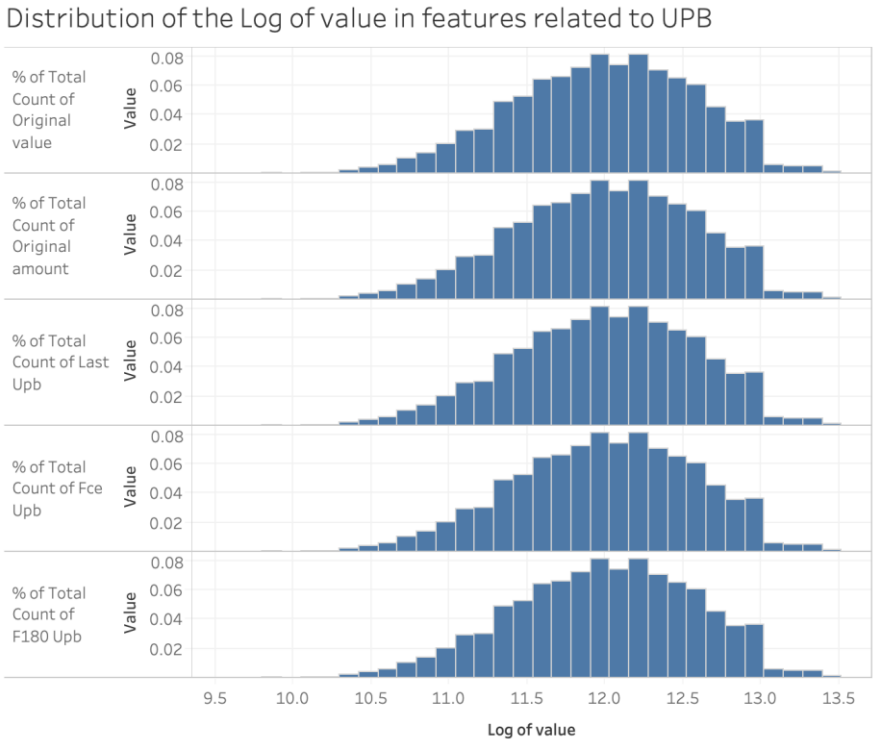

Figure 6: Distribution of the log of value in features related to UPB

Thirdly, we found there are too many features related to money and we want to see the differences among them. From Figure 5, we can see that almost all the features related to UPB show the same distribution and it is close to a normal distribution. There exists multicollinearity. Thus, we will just keep one variable, the 'original amount', which means the original amount of the mortgage loan as indicated by the mortgage documents. Next, we checked out the distribution of DTI, which is a ratio calculated at origination derived by dividing the borrower's total monthly obligations (including housing expense) by his or her stable monthly income. This calculation is used to determine the mortgage amount for which a borrower qualifies. From the histogram, we can see that most DTI values are in the interval between 40% and 45%.

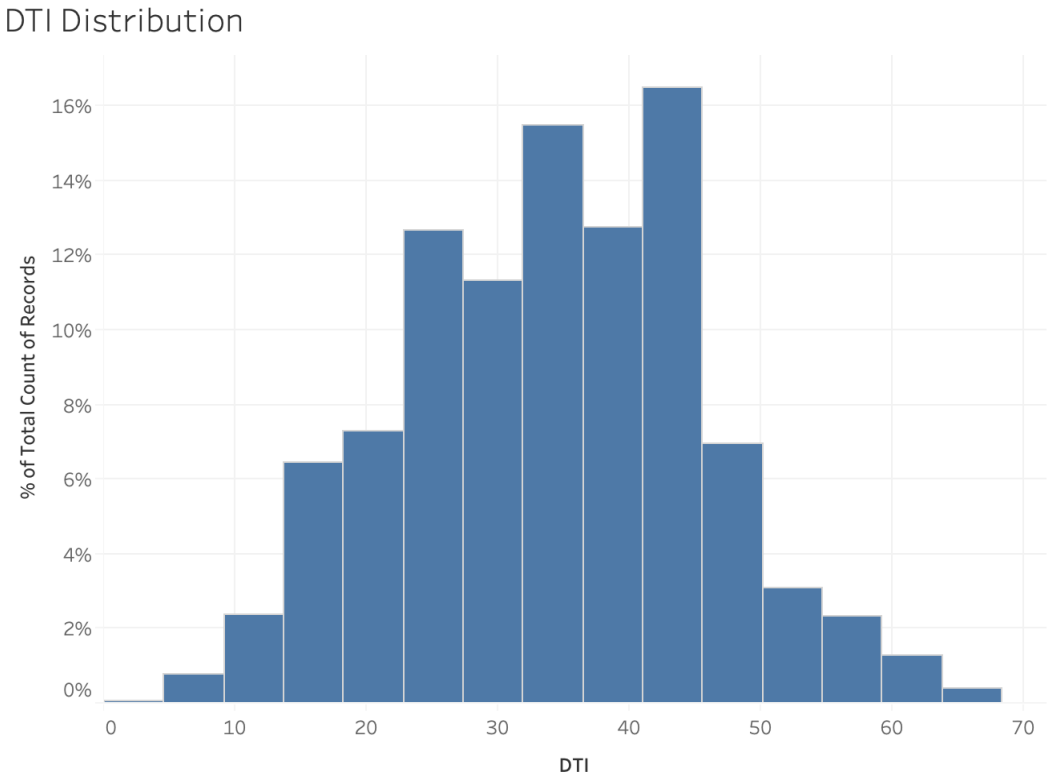

Figure 7: DTI Distribution

There are two interest rates in our dataset, which are the original interest rate and last interest rate. From Figure8, we can see that the distribution of the interest rate is the same, which means that there is almost no change between the interest rate.

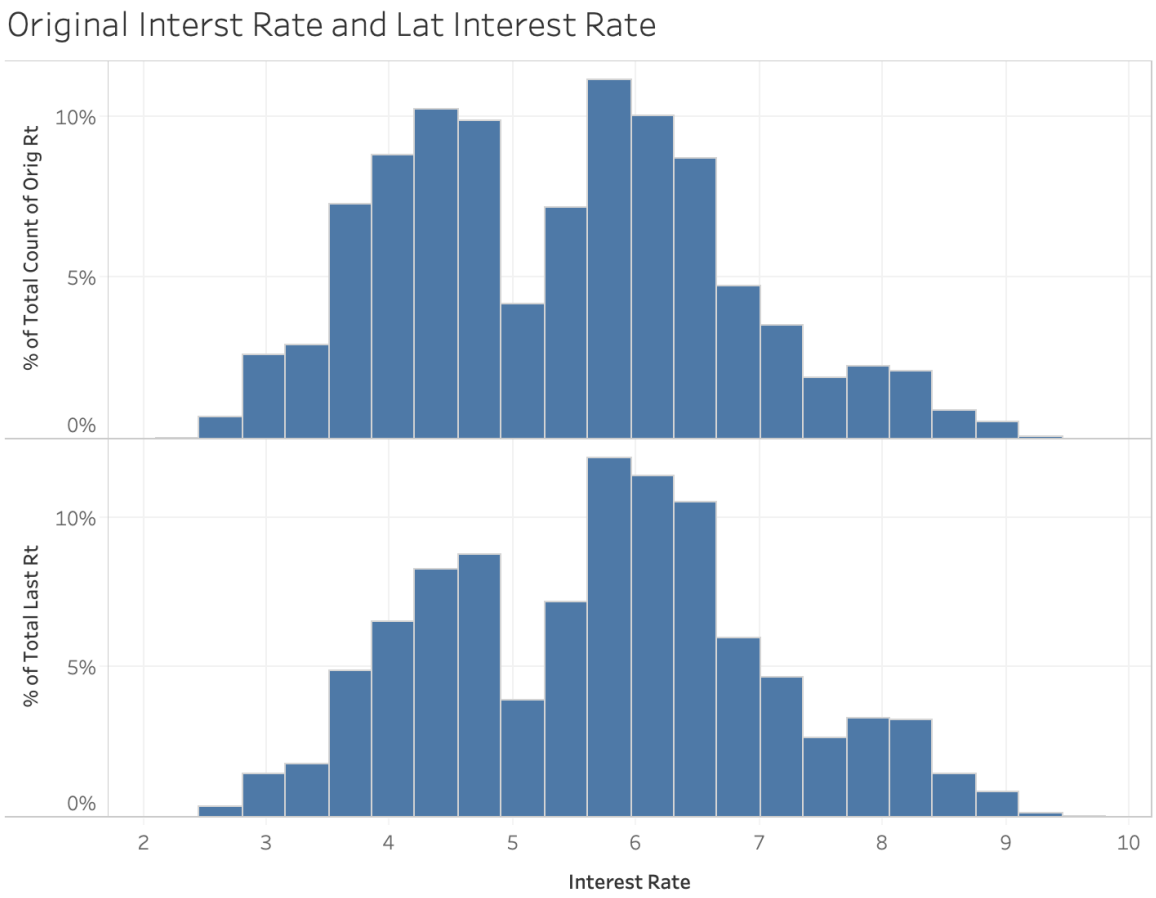

Figure 8: Distribution of Original Interest Rate and Last Interest Rate

OLTV and OCLTV are also two important features in our dataset. OLTV is a ratio calculated at origination derived by dividing the borrower's total monthly obligations (including housing expense) by his or her stable monthly income. This calculation is used to determine the mortgage amount for which a borrower qualifies. OCLTV is a ratio calculated at the time of origination for a mortgage loan. The CLTV reflects the loan-to-value ratio inclusive of all loans secured by a mortgaged property on the origination date of the underlying mortgage loan. Figure9 shows the same distribution between OLTV AND OCLTV. We can also see that most values are in the interval between 80% and 85%, which means that most people choose to borrow 80%-85% of the home value.

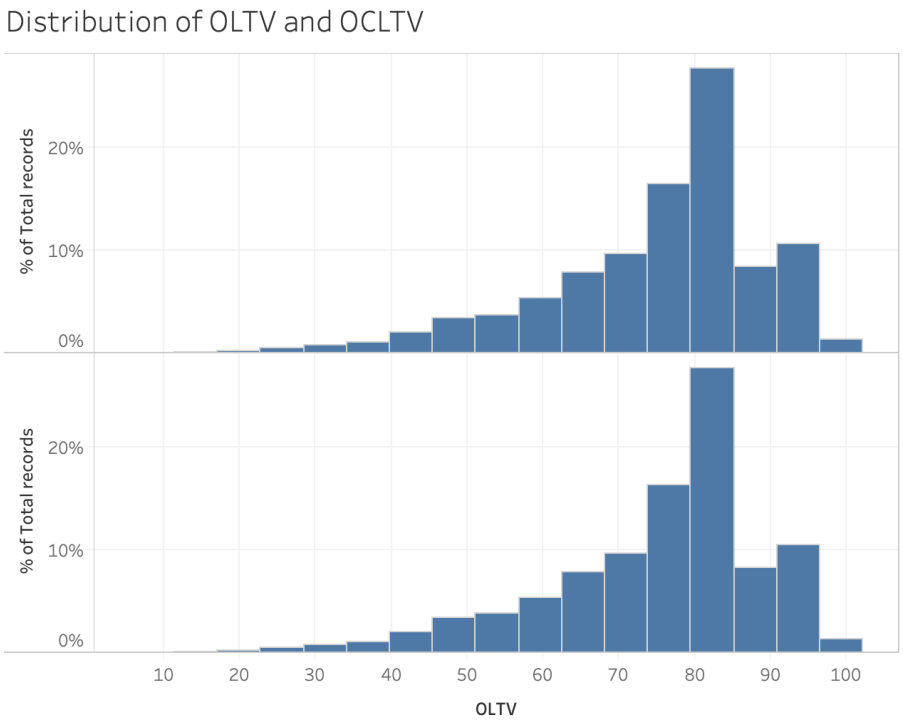

Figure 9: Distribution of OLTV and OCLTV

Loan Age is the number of calendar months since the mortgage loan's origination date. For purposes of calculating this data element, origination means the date on which the first full month of interest begins to accrue. We can see that the distribution of Loan Age is almost uniformly distributed in the interval of 1-12.

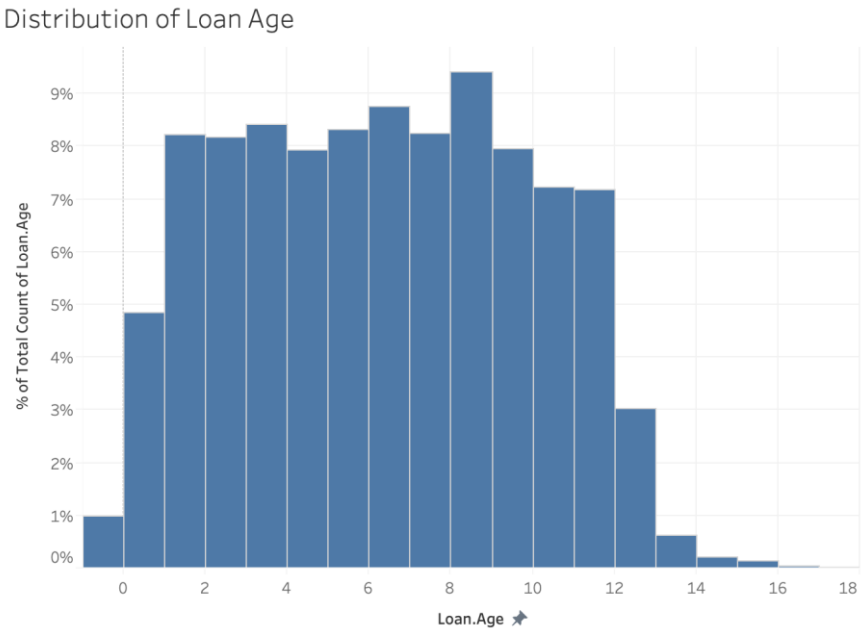

Figure 10: Distribution of Loan Age

A credit score is a very important feature when a bank decides whether to lend money to the borrowers. We can see that the distribution of the credit score of the first borrower and the minimum credit score of all borrowers are the same (Figure11). We also check the distribution of the difference of the values and over 85% of the values are in the interval between 0-10, which means the differences are too small.

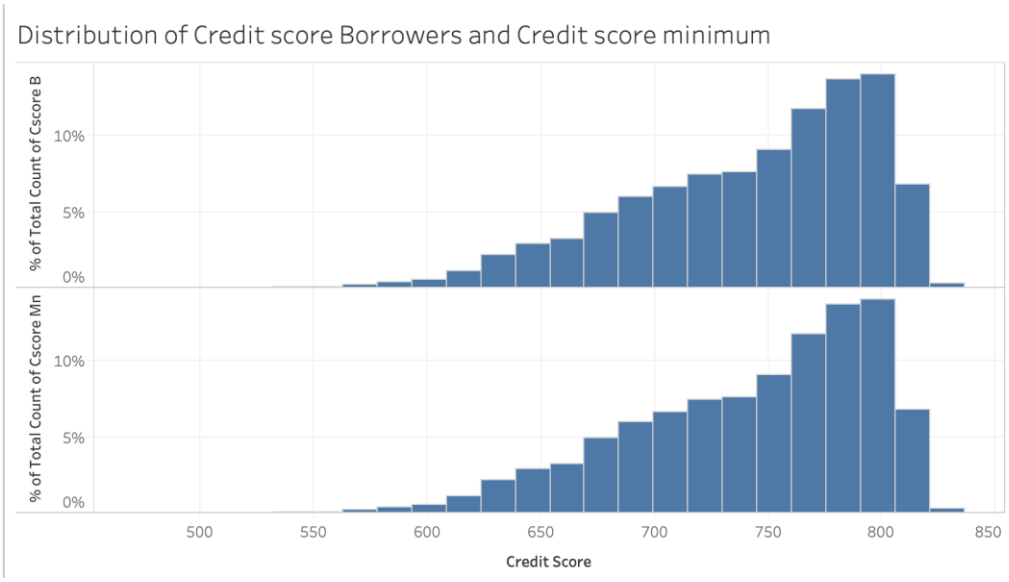

Figure 11: Distribution of Credit Score of Borrowers and Credit Score Minimum

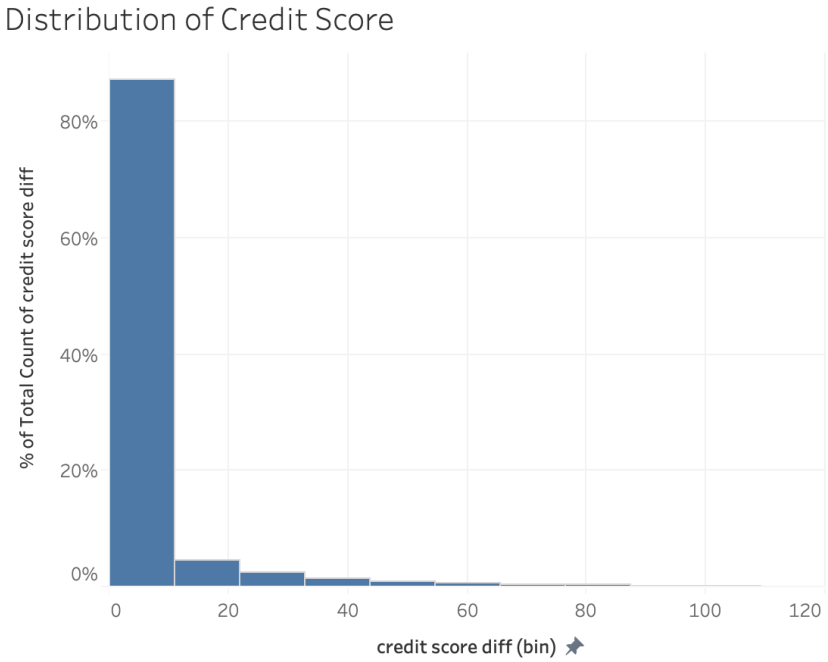

Figure 12: Distribution of Difference between Credit Score of Borrowers and Credit Score Minimum

3 Feature Engineering and Feature Selection

Before we built models, we did some feature engineering and feature selection.

Firstly, we subsample 10,000 rows from the whole dataset to save the computation time. When we use data to calculate the distance and similarity, the computation is slow and if we use the whole data, the system will crash.

Secondly, we deleted unrelated features. The features include "ZB_DTE", "Zero.Bal.Code", "DispYr", "MODIR_COST", "MODFB_COST", "MODTOT_COST", etc. Our project objective is to cluster the loan according to the borrower and loan characteristics. However, these features are useful for predicting for default rate but not useful for clustering.

Thirdly, we deleted the numeric features with too many '0' and categorical features which are imbalanced. These features will affect the clustering result because the same value or too many '0' both do not tell the differences between the clusters. These features include "INT_COST", "total_expense", "total_proceeds", "NET_LOSS", "NET_SEV", "Total_Cost", "Seller.Name", "Servicer.Name", "STATE", "MOD_FLAG", "Product.Type", "Delq.Status", etc.

Next, we dealt with features related to 'money'. We log-transformed the features and kept one variable, 'original amount' because of the same distribution of the features related to 'money' we found in the EDA part.

We also dealt with missing values. Two numeric features were needed to be dealt with, the DTI and the number of borrowers. We used the median value and mode value to replace the null values. We used 45% to replace the null values in DTI and use 1 to replace the null values in the number of borrowers.

There is some date data and we dealt with them. We calculated the date difference and generated two new features, which are 'loan_month' and 'first_month'. The 'loan_month' is the day difference between the maturity date and the original date and the 'first_month' is the day difference between the first payment date and the original date. In addition, there are some other features related to 'time'. To unify the unit of these features, we use the 'month' unit.

To calculate the distance in the clustering, we must change the categorical variables into binary variables. These features are "ORIG_CHN", "FTHB_FLG", "PURPOSE","PROP_TYP", "OCC_STAT".

Finally, we get 31 variables. Before the next step, we standardized the data. The scale of the data in different columns are different, so we need to scale the data.

4 Dimension Reduction

Since there are too many features and we don't know whether some of the data really represents one thing. So, we used PCA to reduce the dimension. Figure7 shows the PCA result. We can see that when we choose 10 components, it can explain almost 100% of the data and when we choose 7 components, it can explain 81% of the data. Although there is a big drop in variance when the dimension changes from 3 to 5, we finally chose the 7 dimensions to reduce our data through the clustering result.

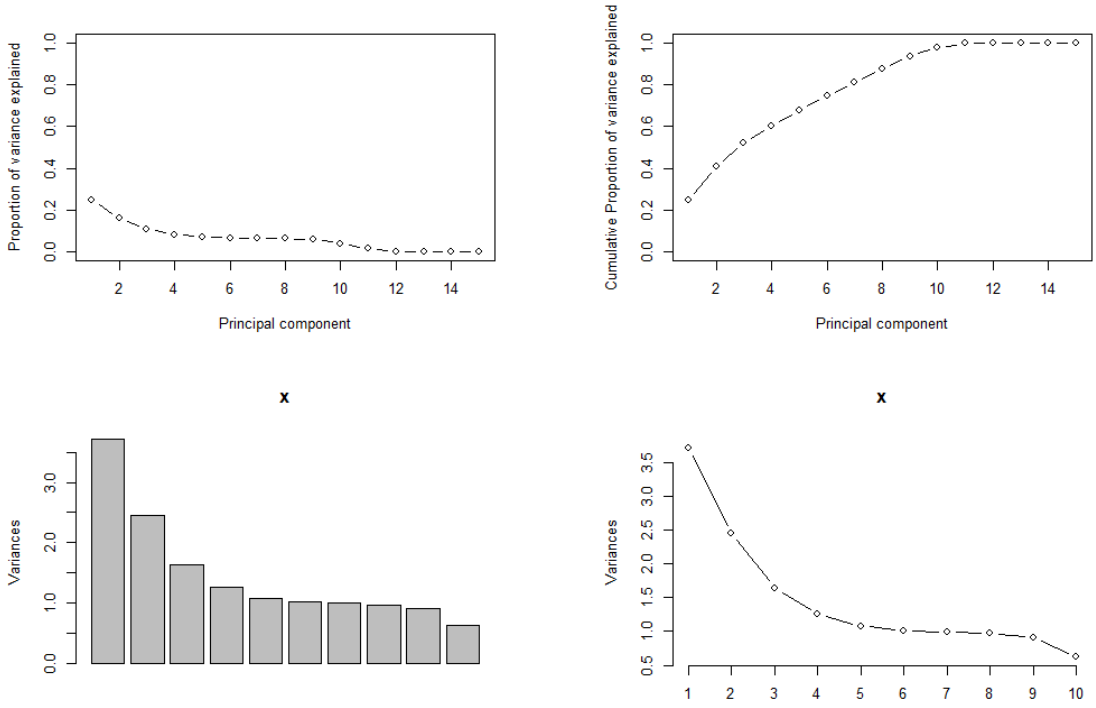

Figure 13: PCA result

According to Figure 8, we can see that features in the same dimension are in the same direction and we can find each component of what actually means. Component 1 represents Loan Term, which is the number of months in which regularly scheduled borrower payments are due under the terms of the related mortgage documents (180 months and 360 months in this dataset). Component 2 is the Loan Interest Rate, which is the original interest rate of a loan. Component 3 is OLTV, which is an original loan to value, the ratio of a loan to the value of a property. Component 4 is the minimum credit score of the borrowers. Component 5 is the number of borrowers. Component 6 is the number of property units. Component 7 is the first_month, which is the months between the loan date and the first payment date.

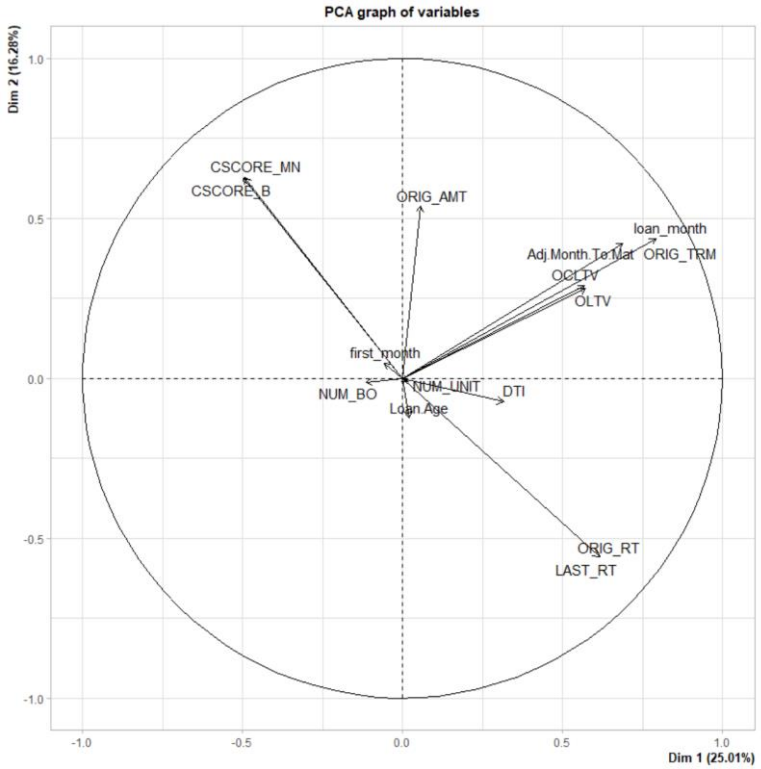

Figure 14: PCA component visualization

5 Clustering Models

5.1 K-Means

K–Means clustering one of the most commonly used unsupervised machine learning algorithms. The objective of K-Means is to group similar data points together and discover patterns. Every point belongs to one cluster and each cluster indicates a collection of points that gathered together because of certain similarities (Garbade, 2018). In order to define K, which refers to the number of centroids that we need, we drew a plot to see the group sum of squares by the number of clusters. As you can see, the slope of 1-5 clusters is higher than the others (Figure 15). Therefore we decided to divide the whole dataset into 5 clusters and got the characteristic of each cluster according to the different components (Figure 16).

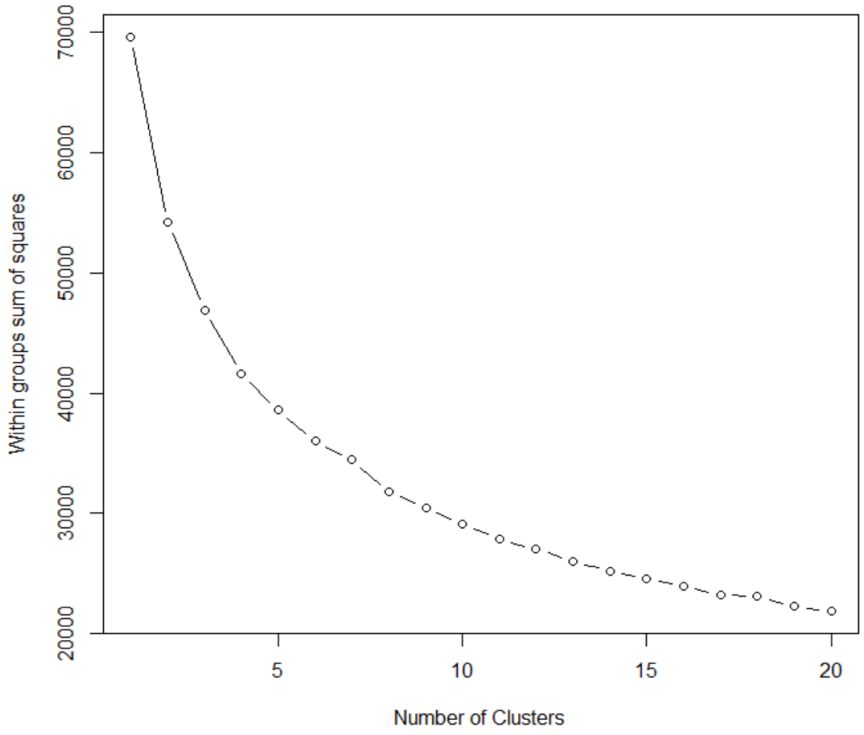

Figure 15: Sum of squares by cluster

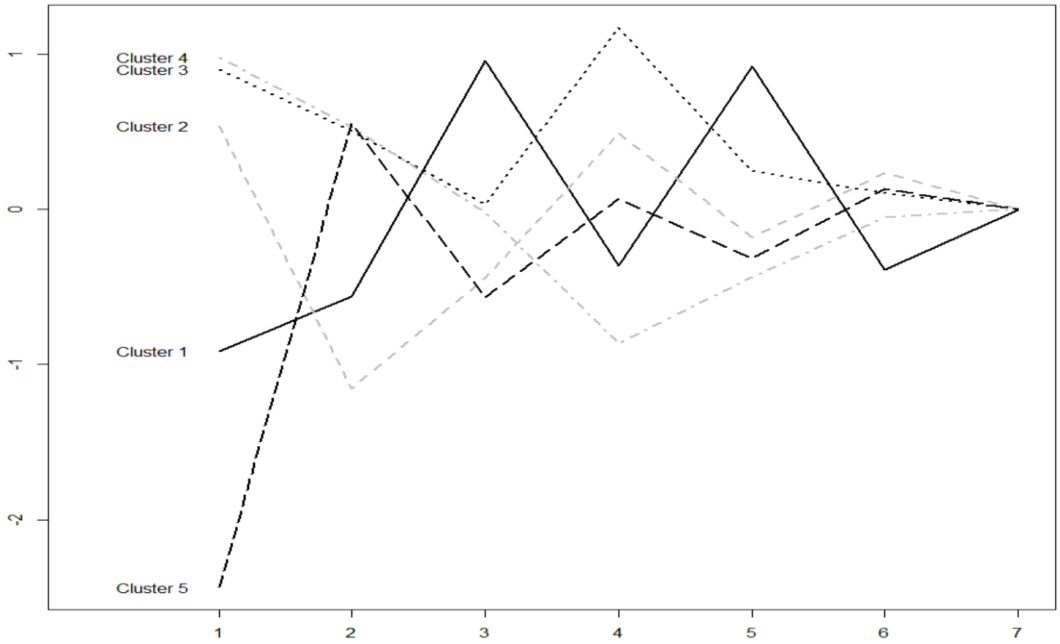

Figure 16: Characteristic by each cluster

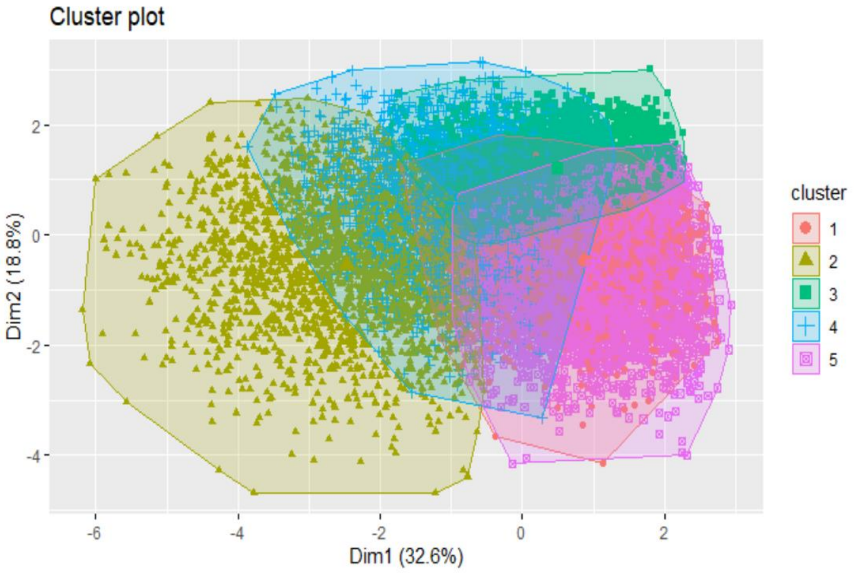

Figure 17: K means 2D plot

After that, we visualized the result of the K-Means clustering method by creating plots to see if the method that we chose made sense. From the 2D plot, though it looks like some of the dots from different clusters are aggregated together, it can be clearly seen from the 3D plots that these 5 clusters are separated well (Figure 18 & 19).

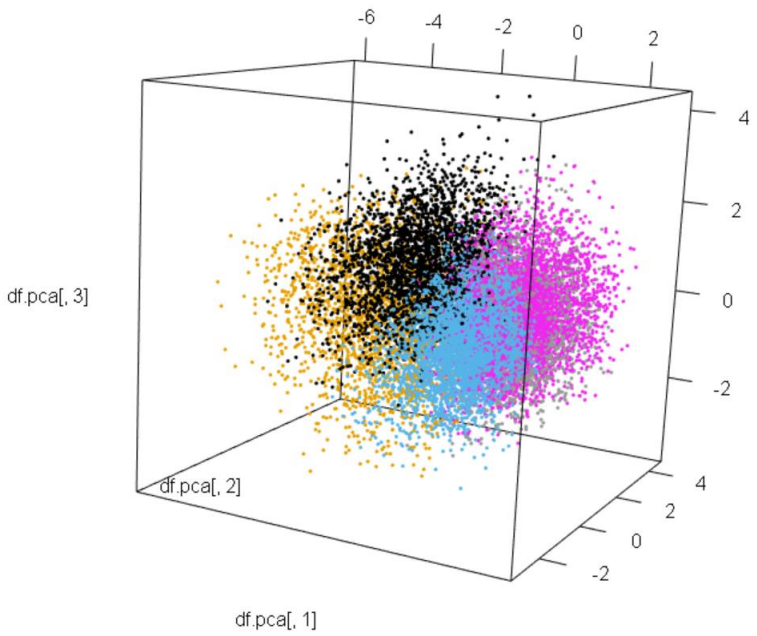

Figure 18: K means 3D plot

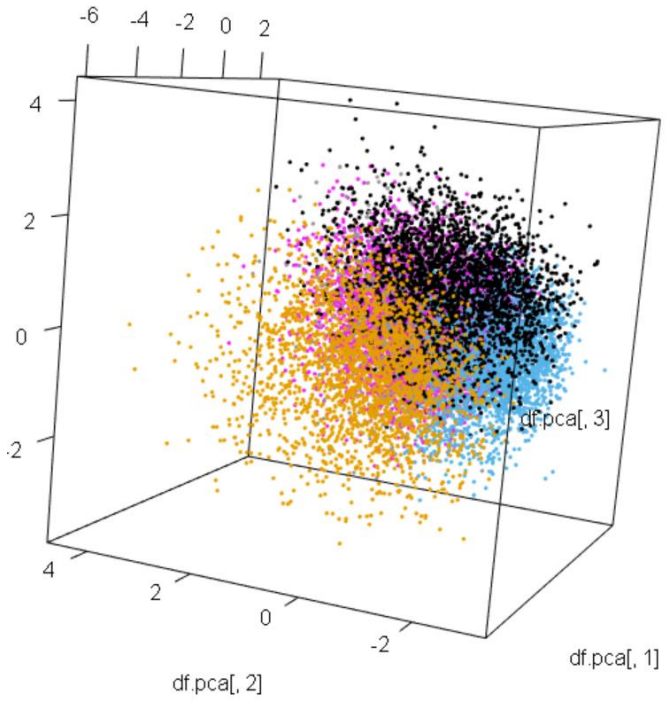

Figure 19: K means 3D plot

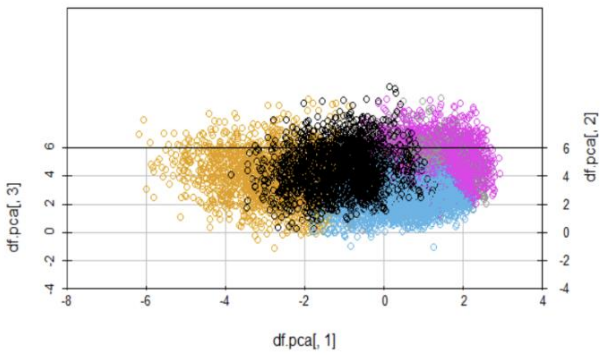

Figure 20: K means 3D plot Angle = 90

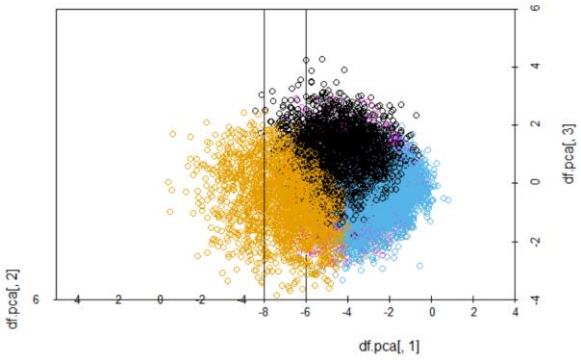

Figure 21: K means 3D plot Angle = 180

Take a further look at the 3D plots with different angles, clusters in yellow, blue and pink are clearly divided at the first dimension of PCA, clusters in blue and black are clearly divided at the third dimension of PCA (Figure 20 & 21).

5.2 K-Medoids

Like K-Means, K-Medoids is also a partitioned clustering method that aims to break the dataset into groups. What's different from the K-Means method is that K-Medoids attempts to minimize the sum of dissimilarities between points assigned to be in a cluster and other objects that are labeled to be other clusters. In addition, K-Medoids can be applied to deal with mixed data types in both numerical and categorical. To estimate the optimal number of clusters, we generated the silhouette width by using a different number of clusters. Silhouette widths, which measures the quality of a clustering. The high silhouette width refers to a good clustering. From the plot below, 4 and 7 clusters have relatively better performance (Figure 22). Since 7 clusters are too much for our dataset, so decided to use 4 clusters.

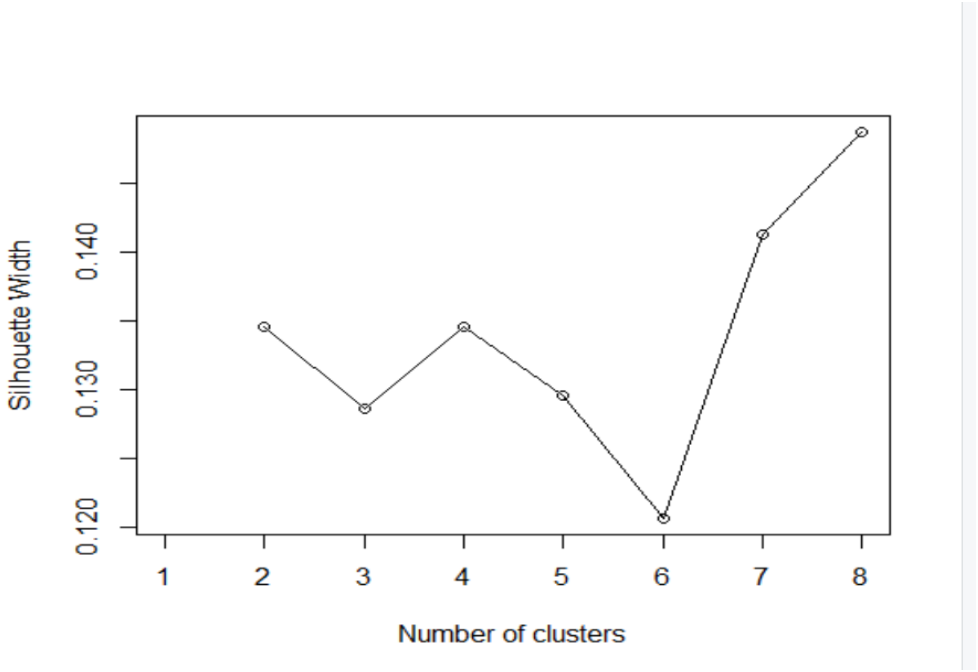

Figure 22: Silhouette Width by cluster

Again, to evaluate the performance of the clustering result. We visualized into 4 and 7 clusters. The objects were not separated well. Because some of the clusters are mixed, and the dots within the same cluster are spread around the plot (Figure 23 & 24).

Figure 23: K-Medoids 2D plot

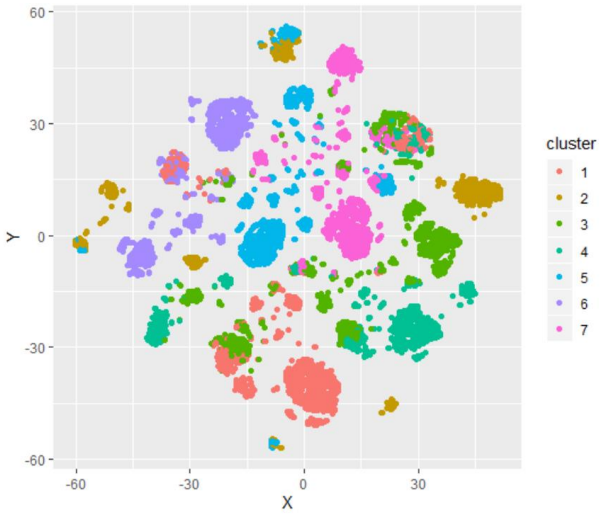

Figure 24: K-Medoids 2D plot

5.3 Mean-Shift

Mean-Shift assigned the data points to the clusters iteratively by shifting points toward the mode. This algorithm iteratively assigns each data point toward the closest cluster centroid and direction to the closest cluster centroid is determined by where most of the points nearby are at (Ankurtripathi, 2019). Because this model will determine the number of clusters by itself, so we got 526 clusters at the very beginning, and then increased iteration and specified the bandwidth to get 80 clusters in the end. But still, it's hard for us to visualize.

5.4 DBSCAN

After that, we used the DBSCAN method, which means the density-based spatial clustering of application with noise (Prado, 2019). DBSCAN groups together point that they are close to each other based on a distance measurement and a minimum number of points. We started with the determination of the optimal eps value by computing the k-nearest neighbor distances in a matrix of points, it can be seen that the optimal eps value is around a distance of 1.1 (Figure 25). DBSCAN is great at separating high-density clusters from low-density clusters.

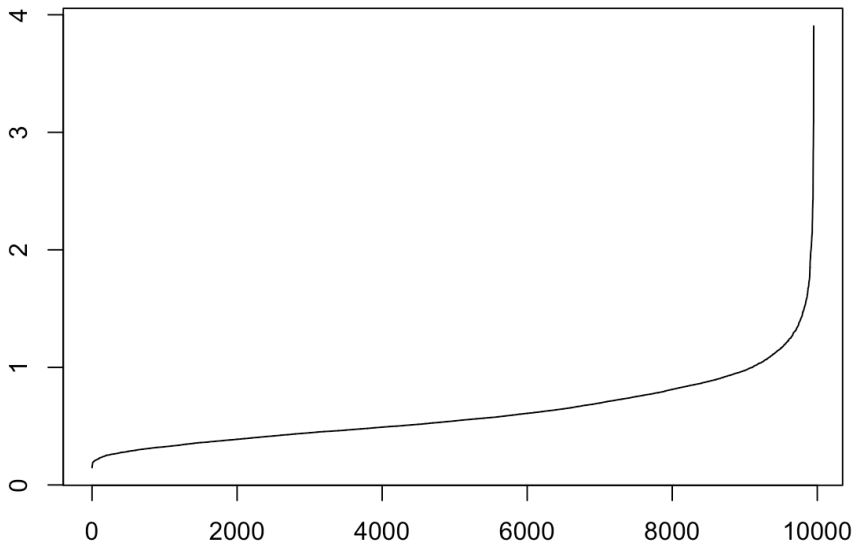

Figure 25: KNN Distribution Plot

As you can see from the plots, the dots are mixed with equal density, which means the clusters are not separated well. From the 2D plot shown below, Cluster 1 and 2 are mixed, and there are only a small portion of the objects labeled as cluster 3 (Figure 26). The 3D plots also show that all the points in grey, yellow, and green are mixed so that we couldn't find patterns or similarities for each cluster (Figure 27). A reason to explain that is DBSCAN is not good at dealing with clusters of similar density.

Figure 26: DBSCAN 2D Plot

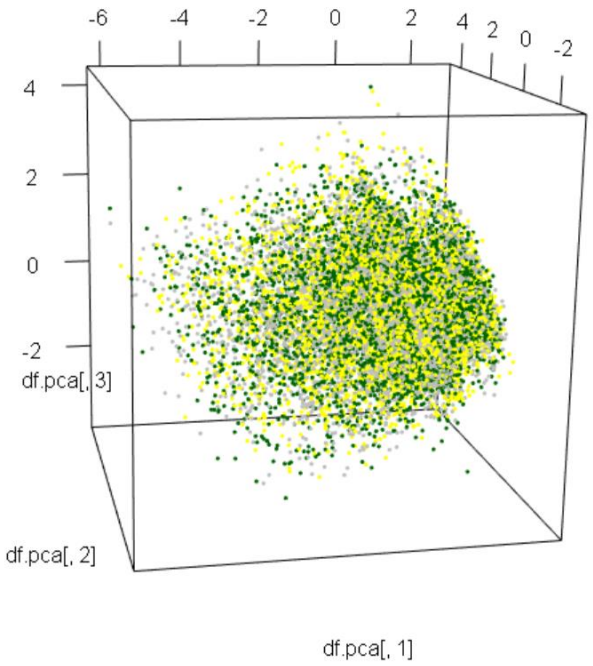

Figure 27: DBSCAN 3D Plot

5.5 GMM

GMM, also known as the Gaussian Mixture Model, uses a function that is composed of several Gaussians, each identified by k ∈ {1,…, K}, where K is the number of clusters. Based on the value of K, GMM can generate Gaussian distributions that can explain the data contained in each cluster. For the parameters, GMM also takes a mean μ that defines its center and a covariance Σ that defines the width. The covariance is also known as the dimension when the data is multivariate. There is also a mixing probability π that defines how big or small the Gaussian function will be. Putting together, the Gaussian Mixture model function is as the following:

$$\mathcal{N}(\mathbf{x}|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

Where x is the data points and D is the dimension. Taking the log of this equation is also helpful for finding the optimized parameters (Carrasco, 2019).

To determine the number of clusters for the mortgage loan dataset, we want to know the highest silhouette score. The silhouette score is a measure of how a data point is to its cluster compared to other clusters. It ranges from -1 to 1, and the higher the silhouette score is, the better the match of a data point. Figure 28 shows the silhouette scores of GMM:

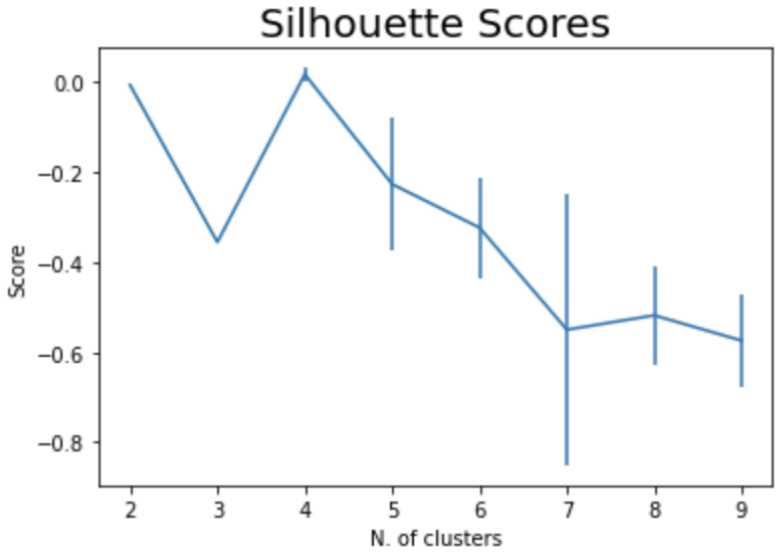

Figure 28: The Silhouette Scores of GMM

The highest silhouette score is around 0 when there are four clusters. Figure 29 is the density plot when k = 4 (4 clusters). The plot shows a lot of overlaps among all the clusters, especially the yellow cluster, which has overlapped with all the remaining clusters. The red cluster and the green cluster, however, are well separated.

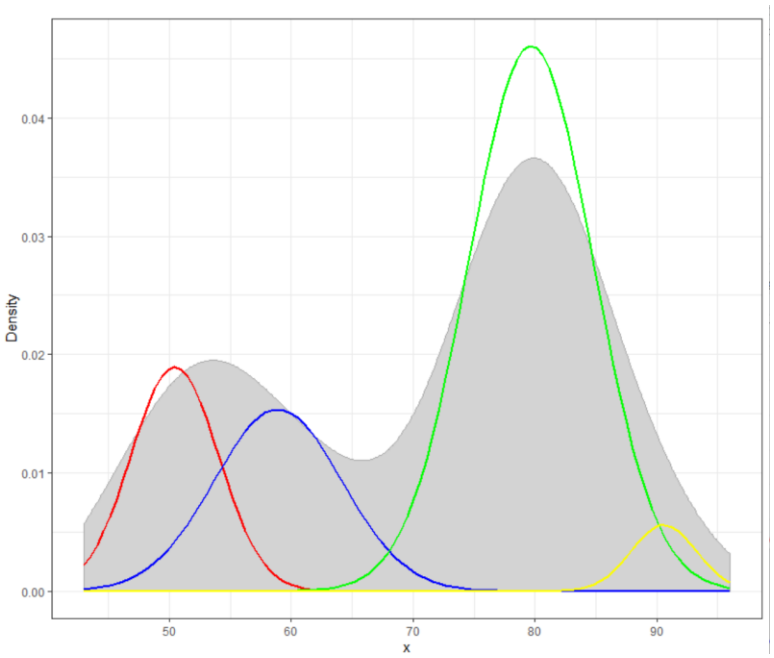

Figure 29: Gaussian Distribution When k = 4

Figure 30 and figure 31 shows the visualization of GMM clustering in 3d plots for k = 4. Even though there are four clusters, two of them can be barely seen in the 3d plots. The remaining two clusters also have overlapped. Therefore, GMM did not perform well in separating the cluster. One reason is that the silhouette score is still low at k = 4, so the model did not separate all the four clusters well enough.

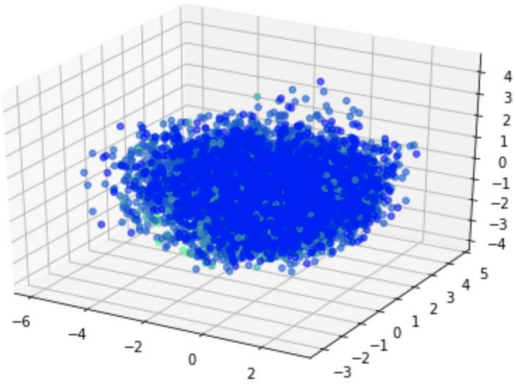

Figure 30: GMM 3D Plot

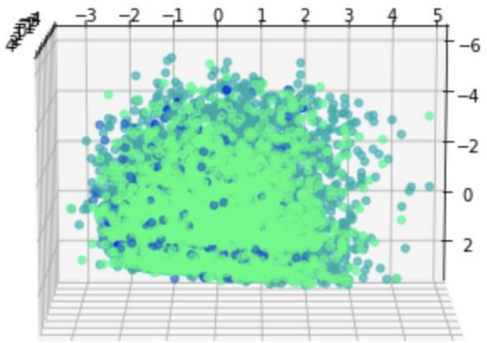

Figure 31: GMM 3D Plot

6 Clustering Results

After analyzing all the models and evaluating all the visualizations, we found out that k-means clustering has the best performances among all the clustering models we have applied because the visualizations of k-means show more separated clusters. Besides, the k-means model is also the fastest, and it is easier to intercept. Moreover, since all the density-based models do not perform well, the dataset may be more suitable for distance-based models, another reason that k-means clustering has the best performance. Table 1 below shows the centroids of the 7 important elements in the 5 clusters. The centroids show differentiations among the 5 clusters and therefore are helpful for the cluster analysis.

	cluster1	cluster2	cluster3	cluster4	cluster5
Loan Term	-0.9130976	0.53797253	0.90185342	0.98125136	-2.4352853
loan Interest Rate	-0.5584709	-1.1528588	0.50635696	0.52931658	0.55032291
OLTV	0.95628355	-0.440004	0.03364429	-0.0179827	-0.5640573
Credit Score	-0.3652234	0.49140301	1.17265512	-0.8636415	0.07109005
Number of Borrowers	0.91956767	-0.1810568	0.24734508	-0.4374156	-0.3166827
Number of Property units	-0.3877515	0.23616098	0.10545262	-0.0511661	0.12924848
First Loan Payment Month	-0.0027501	-0.0012586	0.00068377	0.00079847	0.00253412

Table 1: K-Means centroids

6.1 Two-way ANOVA

To see if the k-means clustering results are usable, we have applied two-way ANOVA to check the p-values of the clusters, the features, and their interactions. Table 2 shows the results of two-way ANOVA. The p-values for the clusters, the features and the interaction are all less than 0.05. Hence, with 95% confidence, we can conclude that there are differences between the 5 clusters, there are differences between the features, and the clusters and the features are dependent on each other. The small p-values also prove that the results from k-means clustering are significant and valid.

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
cluster	185761.3	4	46440.32	110.6	4.63E-94	2.372095
variable	3.52E+09	6	5.87E+08	1397284	0	2.098763
Interaction	3274156	24	136423.2	324.8987	0	1.517494
Within	22882151	54495	419.8945			
Total	3.55E+09	54529				

Table 2: Two-way ANOVA

6.2 Cluster Analysis

The 5 clusters formed by the k-means model all have their characteristics. Cluster 1 has 1,810 loans borrowed. The original term for loans in cluster one is all 15 years only, whereas loans borrowed in other clusters are 30 years. The original loan to value ratio median in cluster 1 is 80%, which is relatively high. The number of units and first months (the number of months between the borrow date and the first loan payment date) cannot differentiate the cluster from other clusters. Therefore, borrowers in cluster 1 have short-term mortgage loans with relatively low interest rates and smaller portions of the original values.

Cluster 2 has 1,954 loans borrowed. The credit score median of cluster 2 is 760, which is good compared to the score in other clusters. This cluster has the lowest interest rate, which median is 4.5%. The original loan to value ratio in this cluster is 77%, so borrowers in cluster 2 also borrowed a larger portion of loans. They have good credits, and the good credits could potentially lead to lower interest rates.

Cluster 3 has 1,759 loans borrowed. The credit score median in cluster 3 is 714, which is the lowest. The original loan to value ratio is 80%, and the original interest rate is 6.5%, which is the highest. Borrowers in cluster 3 have bad credit, and they may have to pay higher interests because of their credit.

Cluster 4 has 2,863 loans borrowed. The credit score median in cluster 4 is 740, which is neither too high nor too low. The number of borrowers in cluster 4 are 1, while the number of borrowers in other clustering are 2. The original loan to value ratio is 80%, so borrowers in cluster 4 have relatively lower credit scores, borrowed large portions of mortgage loans, and are responsible for the loans independently.

Cluster 5 has 1,558 loans borrowed. The credit score median is 768, which is the highest. The original loan to value ratio in cluster 5 is 42%, which is the lowest. The original interest rate median is around 5.4%, so borrowers in cluster 5 have good credits and they borrowed the smallest portion of mortgage loans. They may be in better financial condition than borrowers in other clusters.

7 Conclusions

In summary, we prepared the data for modeling by dropping a lot of imbalanced and unrelated features, engineering the remaining features, and using PCA for dimension reduction. After applying the appropriate clustering models on the important components from PCA, we found out that k-means clustering has the best performance. Then, from the k-means clustering results, we conclude that cluster 1 borrowers have short-terms loans with low interest rate, cluster 2 borrowers have long-term loans and good credit, cluster 3 borrowers have bad credit and high-interest-rate loans, cluster 4 borrowers have mediocre credit and are independent, and cluster 5 borrowers have excellent credit with small amount of loans.

8 References

Ankurtripathi, (2019, May 16). ML: Mean-Shift Clustering. Retrieved from https://www.geeksforgeeks.org/ml-mean-shift-clustering/

Budiaji, W., & Leisch, F. (2019). Simple K-Medoids Partitioning Algorithm for Mixed Variable Data. Algorithms, 12(9), 177. doi: 10.3390/a12090177

Carrasco, Oscar. (2019). Gaussian Mixture Models Explained. *Toward Data Science.* Retrieved from https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

Fannie Mae Single-Family Loan Performance Data. (n.d.). *Fannie Mae*. Retrieved from https://www.fanniemae.com/portal/funding-the-market/data/loan-performance-data.html

Garbade, M. J. (2018, September 12). Understanding K-means Clustering in Machine Learning. Retrieved from https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

Prado, K. S. do. (2019, June 3). How DBSCAN works and why should we use it? Retrieved from https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80