Table of Content

Executive Summary

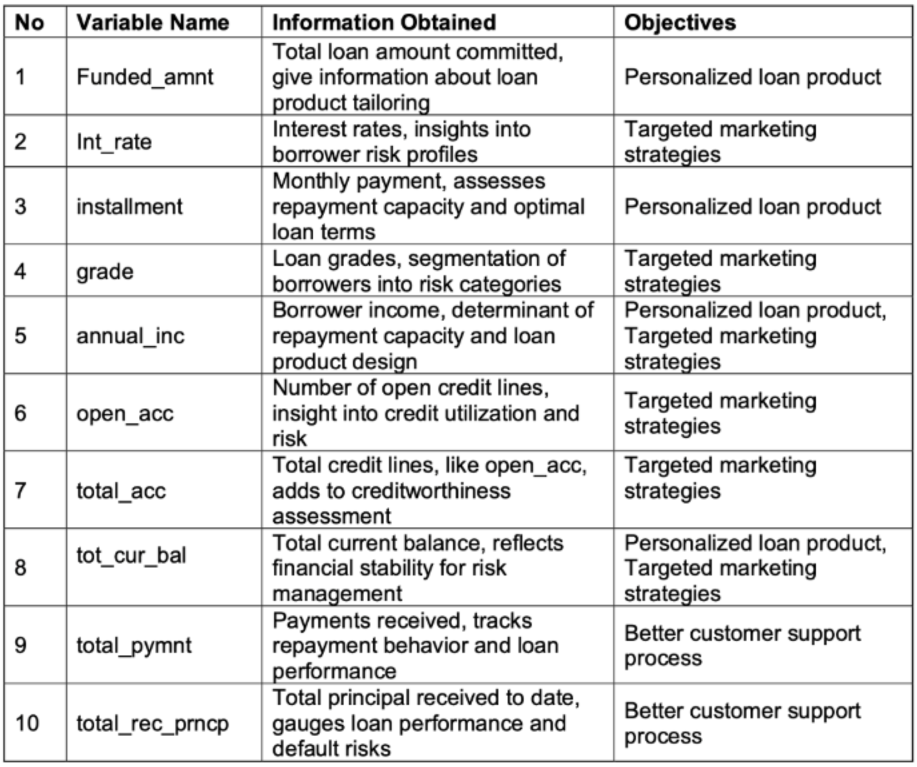
This report provides an analytical foundation for a peer-to-peer lending company named Lending Club to enhance loan portfolio management through data-driven strategies. We utilized cluster analysis to gain insights into customer behavior, customize loan products, refine marketing strategies, and enhance customer support. Key steps included careful variable selection, pre-analysis decisions, and assumption checks, followed by principal component analysis (PCA) and factor analysis (FA) to handle multicollinearity and dimensionality reduction. Cluster creation and analysis confirmed a three-cluster model as most suitable. Each cluster's profile was interpreted based on mean values, highlighting distinct borrower segments. Tailored recommendations for personalized loan products, targeted marketing, and customer support were provided for each cluster. In conclusion, this analysis offers valuable insights into borrower behavior, facilitating the implementation of targeted strategies to improve operational efficiency and customer satisfaction within the lending company.

1. Introduction

Effective loan portfolio management is crucial for lending organizations operating in a competitive industry. By leveraging analytics, Lending Club can make data-driven decisions to optimize loan performance and enhance customer satisfaction. This report presents our research within a peer-to-peer lending company's analytics department. Our goal is to use cluster analysis to gain insights into customer behavior, customize loan products, refine marketing strategies, and improve customer support.

2. Variable Selection

Variable selection is crucial part in cluster analysis. The chosen variables must align with the project's goals. The figure below explains why the 10 variables are included and how they relate to the project's objectives.



No	Variable Name	Information Obtained	Objectives
1	Funded_amnt	Total loan amount committed, give information about loan product tailoring	Personalized loan product
2	Int_rate	Interest rates, insights into borrower risk profiles	Targeted marketing strategies
3	installment	Monthly payment, assesses repayment capacity and optimal loan terms	Personalized loan product
4	grade	Loan grades, segmentation of borrowers into risk categories	Targeted marketing strategies
5	annual_inc	Borrower income, determinant of repayment capacity and loan product design	Personalized loan product, Targeted marketing strategies
6	open_acc	Number of open credit lines, insight into credit utilization and risk	Targeted marketing strategies
7	total_acc	Total credit lines, like open_acc, adds to creditworthiness assessment	Targeted marketing strategies
8	tot_cur_bal	Total current balance, reflects financial stability for risk management	Personalized loan product, Targeted marketing strategies
9	total_pymnt	Payments received, tracks repayment behavior and loan performance	Better customer support process
10	total_rec_prncp	Total principal received to date, gauges loan performance and default risks	Better customer support process

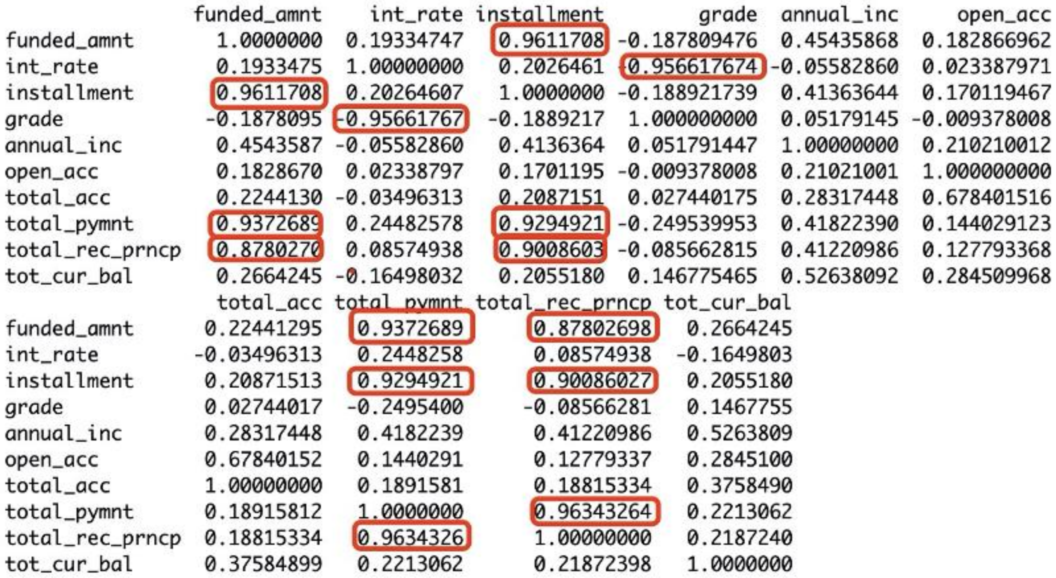
Figure 1. Variable Selection Table

3. Pre-analysis Decision and Assumption Check

Due to the large dataset, we randomly selected a representative sample of 500 observations for analysis. Mahalanobi’s D^2 method was used to detect and remove outliers with a p-value less than 0.001.

To ensure consistency, the dataset was standardized, adjusting each variable to lie within a mean of zero and a standard deviation of one. This prepares the data for subsequent analysis using Principal Component Analysis (PCA) and Factor Analysis (FA).

We then move on to check for substantial multicollinearity in the variables selected for clustering; with correlations between two variables above 0.8 indicating high correlation (Figure 2). Due to multiple multicollinearity pairs in the sample, we proceed to PCA to prevent certain highly correlated variables from dominating the cluster analysis results.



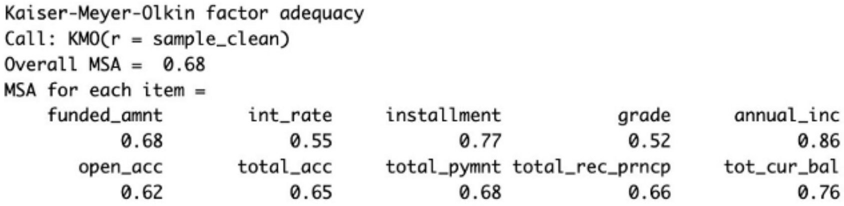
```
               funded_amnt    int_rate installment         grade  annual_inc     open_acc
funded_amnt      1.0000000  0.19334747   0.9611708 -0.187809476  0.45435868  0.182866962
int_rate         0.1933475  1.00000000   0.2026461 -0.956617674 -0.05582860  0.023387971
installment      0.9611708  0.20264607   1.0000000 -0.188921739  0.41363644  0.170119467
grade           -0.1878095 -0.95661767  -0.1889217  1.000000000  0.05179145 -0.009378008
annual_inc       0.4543587 -0.05582860   0.4136364  0.051791447  1.00000000  0.210210012
open_acc         0.1828670  0.02338797   0.1701195 -0.009378008  0.21021001  1.000000000
total_acc        0.2244130 -0.03496313   0.2087151  0.027440175  0.28317448  0.678401516
total_pymnt      0.9372689  0.24482578   0.9294921 -0.249539953  0.41822390  0.144029123
total_rec_prncp  0.8780270  0.08574938   0.9008603 -0.085662815  0.41220986  0.127793368
tot_cur_bal      0.2664245 -0.16498032   0.2055180  0.146775465  0.52638092  0.284509968
                 total_acc total_pymnt total_rec_prncp tot_cur_bal
funded_amnt     0.22441295   0.9372689      0.87802698   0.2664245
int_rate       -0.03496313   0.2448258      0.08574938  -0.1649803
installment     0.20871513   0.9294921      0.90086027   0.2055180
grade           0.02744017  -0.2495400     -0.08566281   0.1467755
annual_inc      0.28317448   0.4182239      0.41220986   0.5263809
open_acc        0.67840152   0.1440291      0.12779337   0.2845100
total_acc       1.00000000   0.1891581      0.18815334   0.3758490
total_pymnt     0.18915812   1.0000000      0.96343264   0.2213062
total_rec_prncp 0.18815334   0.9634326      1.00000000   0.2187240
tot_cur_bal     0.37584899   0.2213062      0.21872398   1.0000000
```

Figure 2. Correlation between two variables; we identified correlations > 0.8 as highly correlated variables. (red boxes)

4. Principal Component Analysis & Factor Analysis

Assumption Check for Principal Component Analysis & Factor Analysis

Both PCA and FA require the existence of sufficient intercorrelation between variables, and such intercorrelation should also make sense conceptually for FA to be conducted. We confirmed the presence of sufficient intercorrelation between variables, with numerous correlations exceeding 0.3. The KMO test yielded a score of 0.68 (Figure 3), indicating sufficient correlation. Additionally, the Bartlett test indicated suitability for Factor Analysis with a p-value below 0.05.



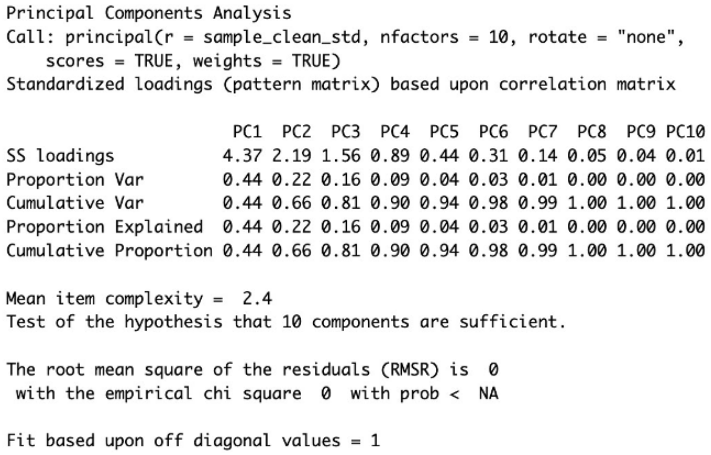
```
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = sample_clean)
Overall MSA =  0.68
MSA for each item = 
    funded_amnt        int_rate     installment           grade      annual_inc 
           0.68            0.55            0.77            0.52            0.86 
       open_acc       total_acc     total_pymnt total_rec_prncp     tot_cur_bal 
           0.62            0.65            0.68            0.66            0.76 
```

Figure 3. KMO test result

Principal Component Analysis (PCA)

PCA was applied and cumulative variance reached 1.00 by PC8 (Figure 4), suggesting the first 8 principal components captured all data information. However, since only the first 3 PCs had eigenvalues ≥ 1, and cumulative variance exceeded 0.6 after PC2, we considered 2 or 3 PCs suitable. On the other hand, Scree plots indicated the optimal number of PCs to be around 4 or

5 (Figure 5).



```
Principal Components Analysis
Call: principal(r = sample_clean_std, nfactors = 10, rotate = "none",
    scores = TRUE, weights = TRUE)
Standardized loadings (pattern matrix) based upon correlation matrix

                       PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10
SS loadings           4.37 2.19 1.56 0.89 0.44 0.31 0.14 0.05 0.04 0.01
Proportion Var        0.44 0.22 0.16 0.09 0.04 0.03 0.01 0.00 0.00 0.00
Cumulative Var        0.44 0.66 0.81 0.90 0.94 0.98 0.99 1.00 1.00 1.00
Proportion Explained  0.44 0.22 0.16 0.09 0.04 0.03 0.01 0.00 0.00 0.00
Cumulative Proportion 0.44 0.66 0.81 0.90 0.94 0.98 0.99 1.00 1.00 1.00

Mean item complexity =  2.4
Test of the hypothesis that 10 components are sufficient.

The root mean square of the residuals (RMSR) is  0
 with the empirical chi square  0  with prob <  NA

Fit based upon off diagonal values = 1
```

Figure 4. PCA results with 10 PCs.

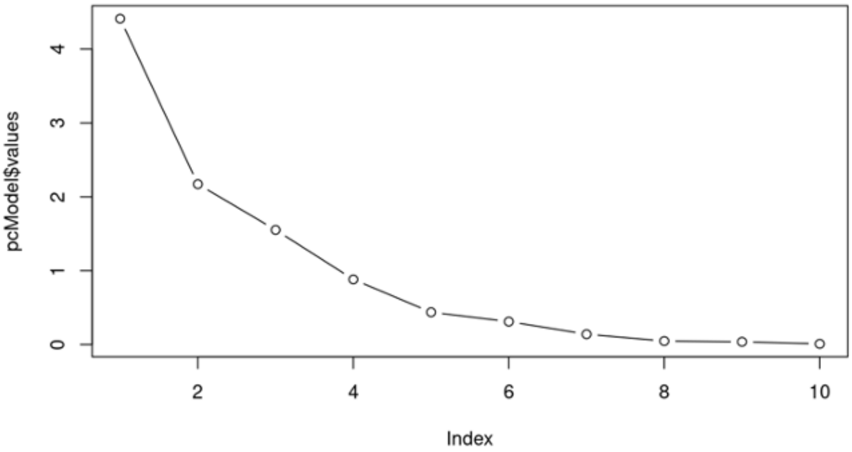
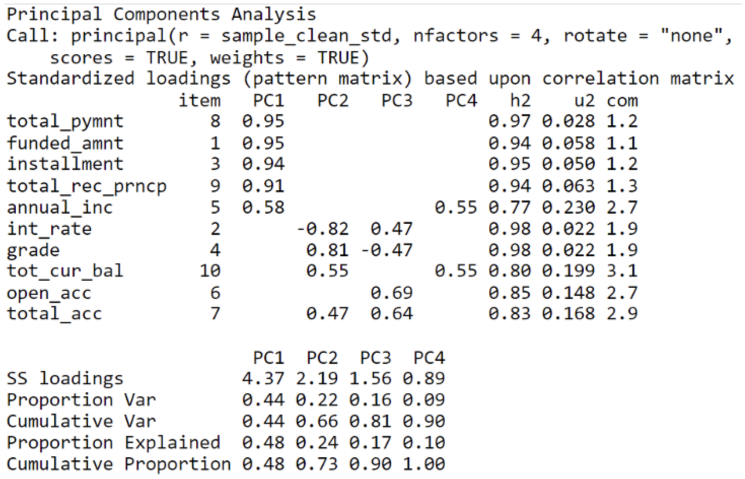


Figure 5. The scree plots. After PC5, the scree plot started becoming flattened. Thus, we can conclude that the optimal number of PCs might be around 4 or 5.

We conducted PCA with 2, 3, 4, and 5 PCs, setting a high correlation threshold of 0.3. However, due to numerous cross-loadings across all PCs, interpreting each PC became impractical. Thus, we opted for Factor Analysis as an alternative for dimensionality reduction and addressing multicollinearity.

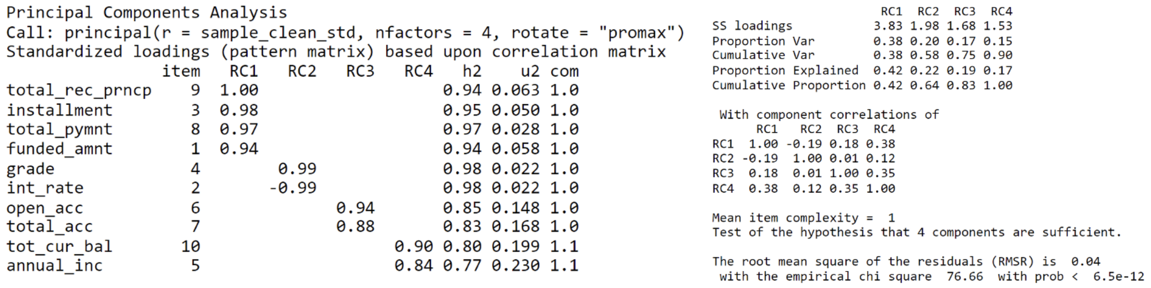


```
Principal Components Analysis
Call: principal(r = sample_clean_std, nfactors = 4, rotate = "none",
    scores = TRUE, weights = TRUE)
Standardized loadings (pattern matrix) based upon correlation matrix
                item  PC1   PC2   PC3   PC4   h2    u2  com
total_pymnt        8  0.95                   0.97 0.028 1.2
funded_amnt        1  0.95                   0.94 0.058 1.1
installment        3  0.94                   0.95 0.050 1.2
total_rec_prncp    9  0.91                   0.94 0.063 1.3
annual_inc         5  0.58              0.55 0.77 0.230 2.7
int_rate           2       -0.82  0.47       0.98 0.022 1.9
grade              4        0.81 -0.47       0.98 0.022 1.9
tot_cur_bal       10        0.55        0.55 0.80 0.199 3.1
open_acc           6              0.69       0.85 0.148 2.7
total_acc          7        0.47  0.64       0.83 0.168 2.9

                       PC1  PC2  PC3  PC4
SS loadings           4.37 2.19 1.56 0.89
Proportion Var        0.44 0.22 0.16 0.09
Cumulative Var        0.44 0.66 0.81 0.90
Proportion Explained  0.48 0.24 0.17 0.10
Cumulative Proportion 0.48 0.73 0.90 1.00
```

Figure 6. PCA results with 4 PCs and cut=0.3.

Factor Analysis (FA)

Factor Analysis was conducted with various rotation methods of Principal Component (PC) and Maximum Likelihood (ML) extraction and with various number of factors were performed. Ultimately, a 4-factor solution using Oblique rotation with the Promax method was chosen, effectively resolving cross-loading issues and capturing all variables. Cumulative variance exceeded 0.9, indicating satisfactory model performance. Additionally, eigenvalues across all factors are higher than 1.50. Please refer to Figure. 7 and 8 for the detailed outcome.



```
Principal Components Analysis
Call: principal(r = sample_clean_std, nfactors = 4, rotate = "promax")
Standardized loadings (pattern matrix) based upon correlation matrix
                item  RC1   RC2   RC3   RC4   h2    u2  com
total_rec_prncp    9  1.00                   0.94 0.063 1.0
installment        3  0.98                   0.95 0.050 1.0
total_pymnt        8  0.97                   0.97 0.028 1.0
funded_amnt        1  0.94                   0.94 0.058 1.0
grade              4        0.99             0.98 0.022 1.0
int_rate           2       -0.99             0.98 0.022 1.0
open_acc           6              0.94       0.85 0.148 1.0
total_acc          7              0.88       0.83 0.168 1.0
tot_cur_bal       10                    0.90 0.80 0.199 1.1
annual_inc         5                    0.84 0.77 0.230 1.1
```

```
                       RC1  RC2  RC3  RC4
SS loadings           3.83 1.98 1.68 1.53
Proportion Var        0.38 0.20 0.17 0.15
Cumulative Var        0.38 0.58 0.75 0.90
Proportion Explained  0.42 0.22 0.19 0.17
Cumulative Proportion 0.42 0.64 0.83 1.00

 With component correlations of
      RC1   RC2  RC3  RC4
RC1  1.00 -0.19 0.18 0.38
RC2 -0.19  1.00 0.01 0.12
RC3  0.18  0.01 1.00 0.35
RC4  0.38  0.12 0.35 1.00

Mean item complexity =  1
Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is  0.04
 with the empirical chi square  76.66  with prob <  6.5e-12
```

Figure 7. FA result (PC extraction) on 4 factors using Oblique rotation with the Promax method.

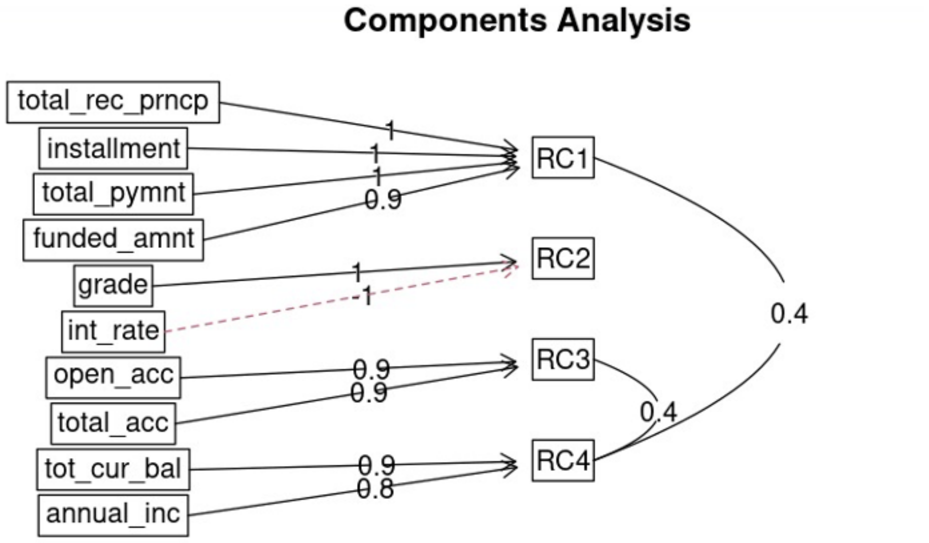


Figure 8. Diagram of FA results (PC extraction) on 4 factors using Oblique rotation with the Promax method.

Factor Interpretation

The interpretations for each factor are as follows:

RC1 – Loan Factor: Higher values of this RC indicate borrowers with larger loan amounts (funded_amnt); the positive relationship with values of installment (monthly payment), total_rec_prncp (principal received), and total_pymnt (payments received) suggests borrowers who are meeting their payment obligations and can possibly paying off their loans. This factor reflects borrowers' ability to manage and repay their loans effectively.

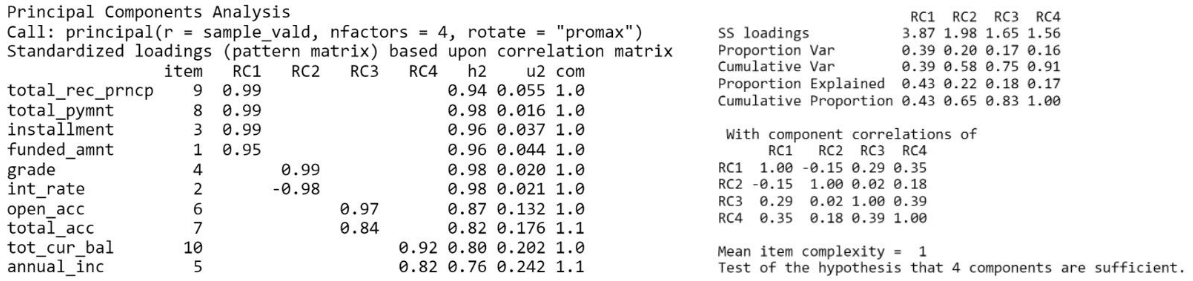
RC2 –Risk: Higher values of RC2 indicate borrowers with lower interest rates (int_rate), better grade assigned by the loan company (grade), indicating higher creditworthiness and lower risk of borrowers. Lenders and financial analysts can use RC2 as a metric to assess borrowers' credit risk profiles.

RC3 – Credit Utilization: The positive relationship with the total number of credit lines (total_acc) and the number of open credit lines (open_acc) indicates longer credit history and potentially more diversified credit portfolio. RC3 represents a dimension related to borrower's credit utilization and borrowing behavior.

RC4 – Financial Stability: RC4 is positively related to total current balance of all accounts (tot_cur_bal) and self-reported annual income (annual_inc), reflecting stronger financial standing and repayment capacity. RC4 represents the overall financial stability of borrowers encompassing factors such as asset accumulation, income level, and debt management.

Factor Validation

Internal validation was conducted using a sample of 100 instances from the original 500. Figures 9 demonstrate that while the factors retained similar relationships with variables, there were slight variations in correlations. This confirmed the stability and validity of the factor analysis.



```
Principal Components Analysis
Call: principal(r = sample_vald, nfactors = 4, rotate = "promax")
Standardized loadings (pattern matrix) based upon correlation matrix
               item  RC1   RC2   RC3   RC4   h2    u2 com
total_rec_prncp    9 0.99                   0.94 0.055 1.0
total_pymnt        8 0.99                   0.98 0.016 1.0
installment        3 0.99                   0.96 0.037 1.0
funded_amnt        1 0.95                   0.96 0.044 1.0
grade              4       0.99             0.98 0.020 1.0
int_rate           2      -0.98             0.98 0.021 1.0
open_acc           6             0.97       0.87 0.132 1.0
total_acc          7             0.84       0.82 0.176 1.1
tot_cur_bal       10                   0.92 0.80 0.202 1.0
annual_inc         5                   0.82 0.76 0.242 1.1
```

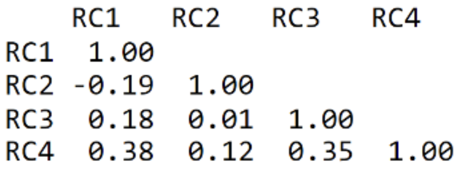
```
                       RC1  RC2  RC3  RC4
SS loadings           3.87 1.98 1.65 1.56
Proportion Var        0.39 0.20 0.17 0.16
Cumulative Var        0.39 0.58 0.75 0.91
Proportion Explained  0.43 0.22 0.18 0.17
Cumulative Proportion 0.43 0.65 0.83 1.00

 With component correlations of
      RC1   RC2  RC3  RC4
RC1  1.00 -0.15 0.29 0.35
RC2 -0.15  1.00 0.02 0.18
RC3  0.29  0.02 1.00 0.39
RC4  0.35  0.18 0.39 1.00

Mean item complexity =  1
Test of the hypothesis that 4 components are sufficient.
```

Figure 9. FA validation result (PC extraction) on 4 factors using Oblique rotation with the Promax method.

5. Cluster Creation and Analysis

Assumption Check for Cluster Analysis

We confirmed the absence of multicollinearity among the established factors (RC1, RC2, RC3, RC4) through correlation analysis (Figure 10), ensuring the validity of our cluster analysis.



```
      RC1   RC2   RC3   RC4
RC1  1.00                  
RC2 -0.19  1.00            
RC3  0.18  0.01  1.00      
RC4  0.38  0.12  0.35  1.00
```

Figure 10. Correlation Matrix between RCs.

Cluster Creation – Hierarchical

After comparing hierarchical clustering methods, Ward’s method was selected for its optimal agglomerative coefficient which minimizes within-cluster variance. The Gap statistic recommended 3 to 4 clusters; we opted for 4 clusters (Figure 11).

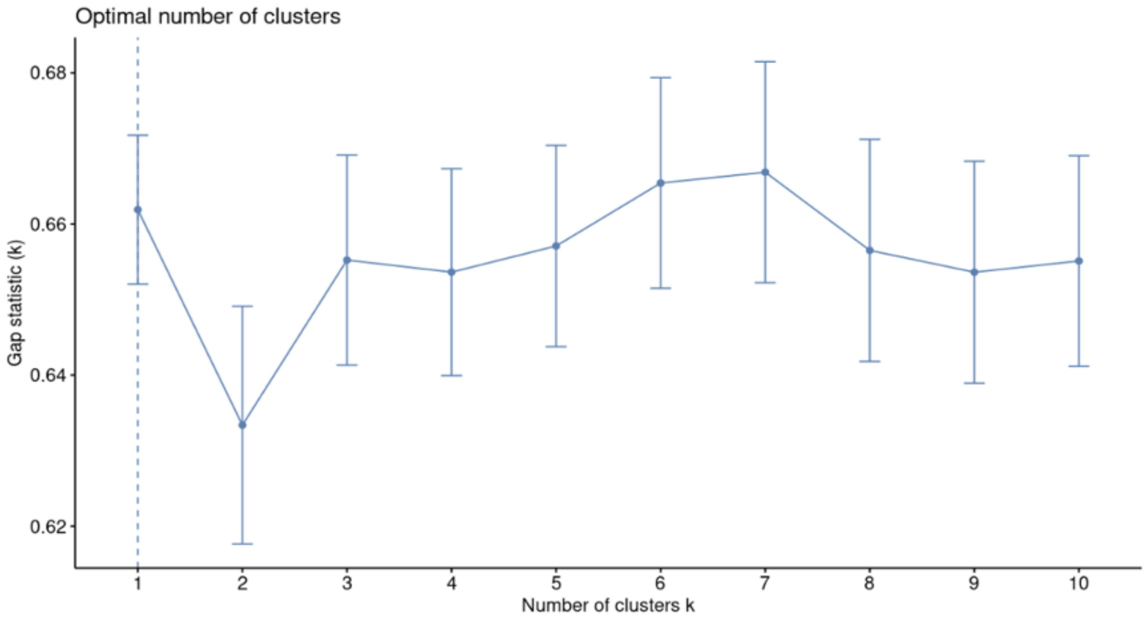


Figure 11: Gap Statistic to estimate the optimal number of clusters

After computing the distance matrix via Euclidean distance, we applied Ward's method to construct a dendrogram. The dendrogram further confirmed our earlier findings, suggesting that the optimal number of clusters is four. (Figure 12)

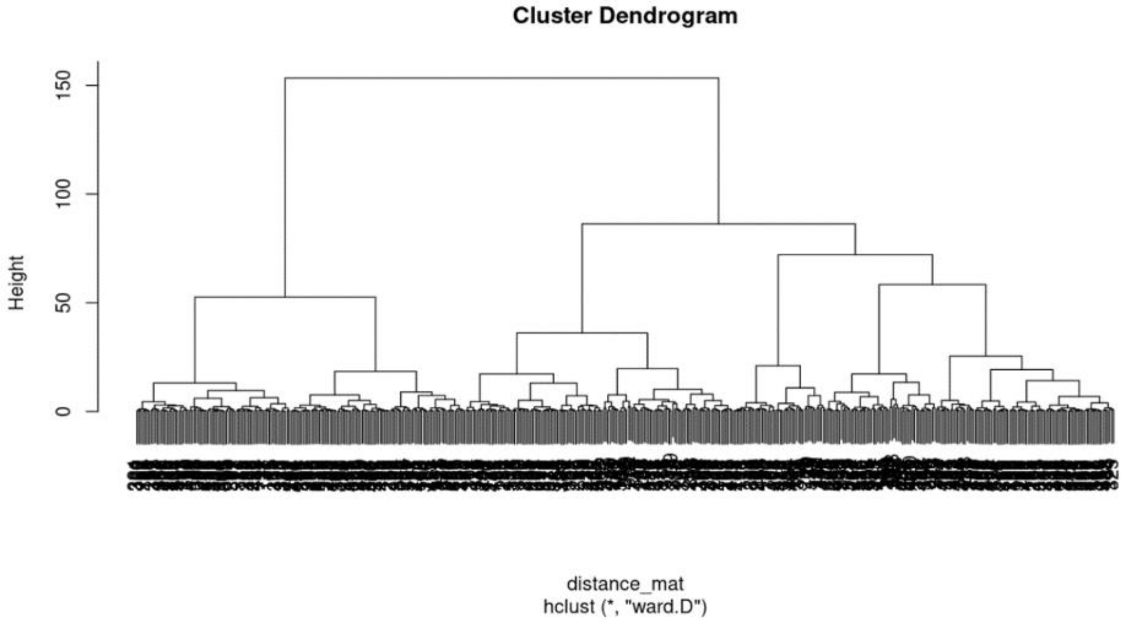


Figure 12: Dendrogram diagram suggesting optimal number of cluster is four.

If we combine hierarchical & non-hierarchical procedures, the number of clusters that hierarchical model suggests is 3 or 4. Thus, we can use suggested numbers to support following non-hierarchical clustering technique.

Cluster Creation – K-means

Building on the insights from hierarchical clustering, we employed the K-means algorithm for its straightforward approach to allocating observations to clusters. Figure 13 shows the results in testing different cluster numbers (k = 2 to 5)

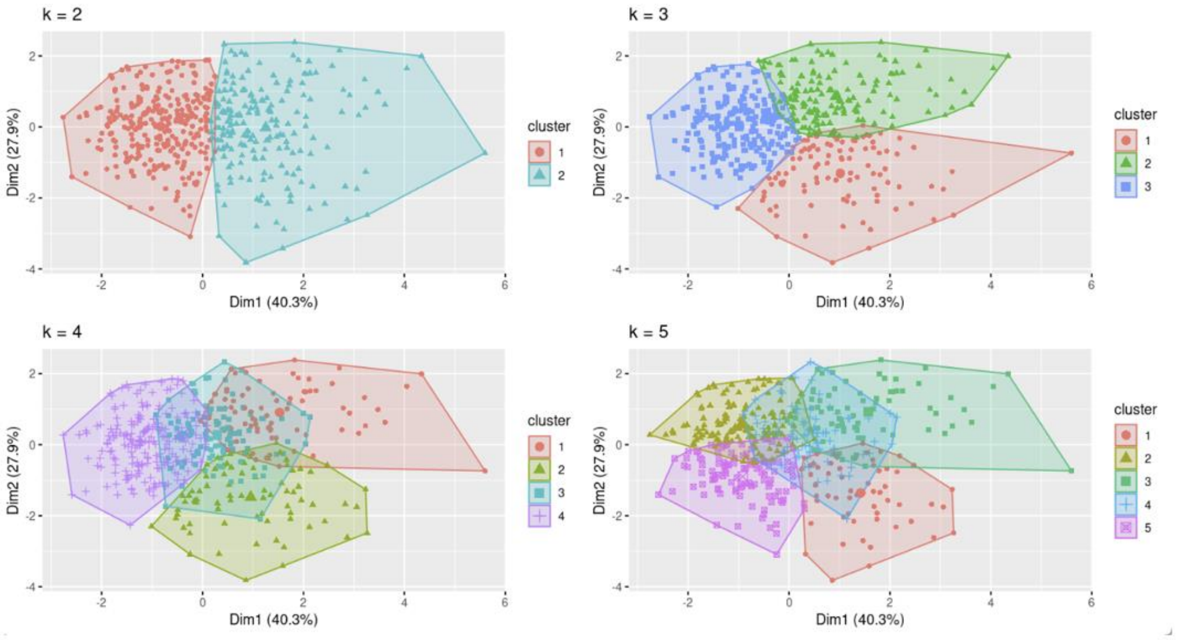


Figure 13: K-means Clustering Visualizations for k = 2 to 5

The analysis suggested that a three-cluster model (k = 3) best suited our dataset, displaying clear boundaries and distinct centroids, indicating robust clustering. While four or five clusters showed some differentiation, they risked over-segmentation.

Cluster Validation

Our cluster analysis internal validation confirmed model robustness. Random sampling of factor analysis f-scores across various subset sizes consistently identified k=3 as optimal. Peak accuracy reached 0.93 for a sample size of 200, slightly decreasing to 0.91 for 250 samples. Figure 14 illustrates model performance.

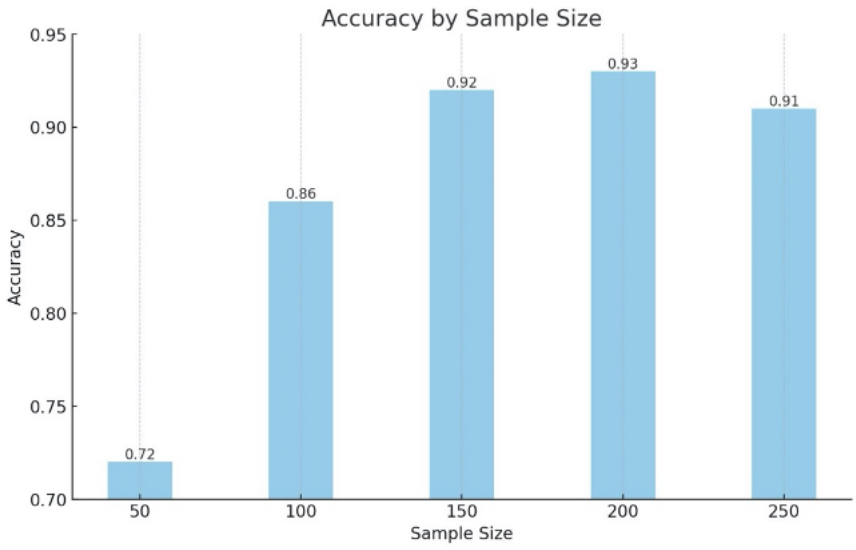
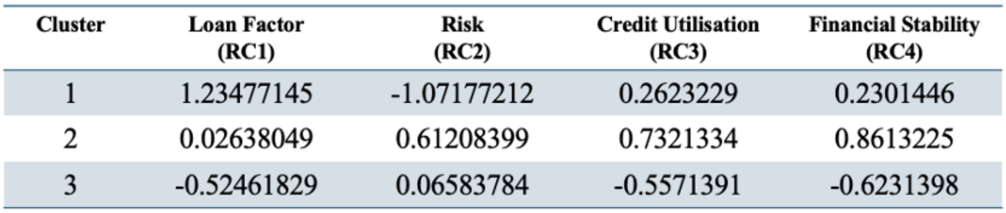


Figure 14: Accuracy vs Sample Size for k = 3

6. Cluster Interpretation

With the validated structure of our clustering model, the team evaluated the individual profiles of each cluster using the results from the cluster mean values (Figure 15)



Cluster	Loan Factor (RC1)	Risk (RC2)	Credit Utilisation (RC3)	Financial Stability (RC4)
1	1.23477145	-1.07177212	0.2623229	0.2301446
2	0.02638049	0.61208399	0.7321334	0.8613225
3	-0.52461829	0.06583784	-0.5571391	-0.6231398

Figure 15: Cluster mean values of the three clusters

Cluster 1 Profile:

Cluster 1 consists of borrowers with larger loans and moderate incomes, with a risk profile that may affect interest rates and credit ratings. Their moderate credit use and financial stability suggest an ability to handle substantial debts.

- Loan Factor: High positive mean value (1.235) indicates these borrowers have larger loan amounts.
- Risk: Negative mean value (-1.071) suggests that these borrowers have higher interest rates and potentially lower grades from the loan company, indicating lower creditworthiness or higher risk.
- Credit Utilization: Positive mean value (0.2623) indicates a reasonable amount of credit use but not as high as Cluster 2.
- Financial Stability: Positive mean value (0.2301) indicates these individuals have decent financial stability, but not as high as those in Cluster 2.

Cluster 2 Profile:

Cluster 2 includes borrowers with good financial health and credit management, likely resulting in favorable loan terms. Their stable financial status is reflected in higher credit lines and balances.

- Loan Factor: Slightly positive mean value (0.0264) indicates these borrowers have average-sized loans.
- Risk: Higher mean value (0.6121) indicates lower interest rates and better grades from the loan company, pointing to high creditworthiness and lower risk.
- Credit Utilization: Positive mean value (0.7321) indicates a higher number of credit lines, suggesting an established credit history and good management of credit.
- Financial Stability: High positive mean value (0.8613) suggests these individuals have higher total current balances and incomes, indicating strong financial stability.

Cluster 3 Profile:

This cluster likely represents conservative borrowers with smaller loans and modest financial stability, indicating a careful financial approach or less established credit. Their average credit scores suggest moderate risk.

- Loan Factor: Negative mean value (-0.5246) indicates this cluster consists of borrowers with smaller loan amounts.
- Risk: A slightly positive mean value (0.0658) indicates a mix of credit scores leaning towards average creditworthiness.
- Credit Utilization: Negative mean value (-0.5571) suggests these borrowers have fewer open credit lines, possibly indicating cautious use of credit or newer credit history.
- Financial Stability: Negative mean value (-0.6231) might indicate that these individuals have less financial stability compared to other clusters, which could mean lower savings or asset balances.

7. Final Solution and Recommendation

Based on the cluster interpretation above, here are some recommendations we can make in terms of personalized loan products, targeted marketing, and better customer support.

Personalized Loan Products (Figure 16)

- Cluster 1: Introduce loan products with higher amounts, averaging 23,000 USD, but with a slightly higher long-term interest rates (15% to 20.5%) because of the high-risk profile in this cluster. Offer structured repayment schedules to accommodate larger loan amounts and longer repayment periods.

- Cluster 2: Develop loan products with moderate amounts, averaging 15,000 USD, for various purposes like debt consolidation or home improvement. Offer tiered and negotiable interest rates-based ranging from 8% to 13%. Provide a variety of term lengths, including both intermediate and longer terms.

- Cluster 3: Offer short-term loans with lower borrowing limits, around 9,500 USD, with interest rates around 11% to 15%. Provide options for refinancing at better rates as creditworthiness improves.

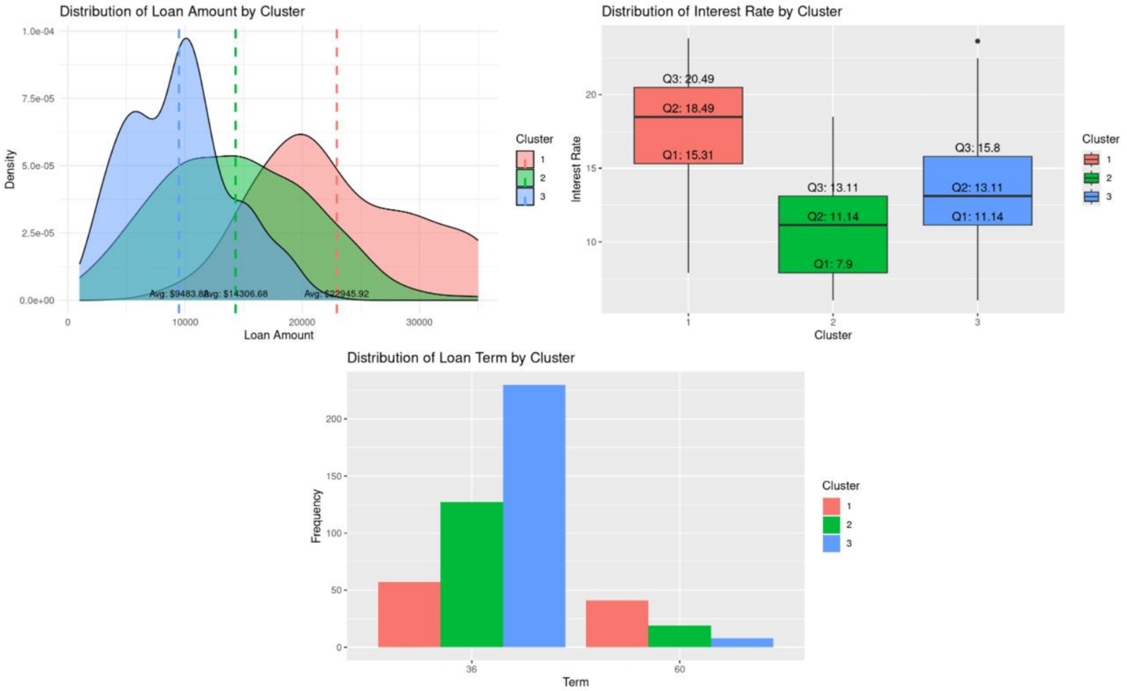


Figure 16. Graph for Personalized Loan Products (Loan Amount, Interest Rate, and Term)

Targeted Marketing (Figure 17)

- Cluster 1: Handling larger loans on moderate incomes, Cluster 1's strategy should focus on credit counselling services to aid those with C, D and E grades in managing their larger loans effectively. Additionally, introducing rental support programs can assist members, particularly renters, with financial education aimed at credit improvement and planning for future homeownership.

- Cluster 2: Loyalty programs could reward the financial health of those with top grades A and B. Offering educational events on maximizing the benefits of homeownership could provide them with valuable knowledge to make the most of their assets and stable financial status.

- Cluster 3: Target marketing towards financially cautious individuals should emphasize budget-friendly loan plans, and financial education aimed at improving credit for those with grade C and D. Additionally, resources should be provided to renters wishing for homeownership, fostering their financial growth and stability.

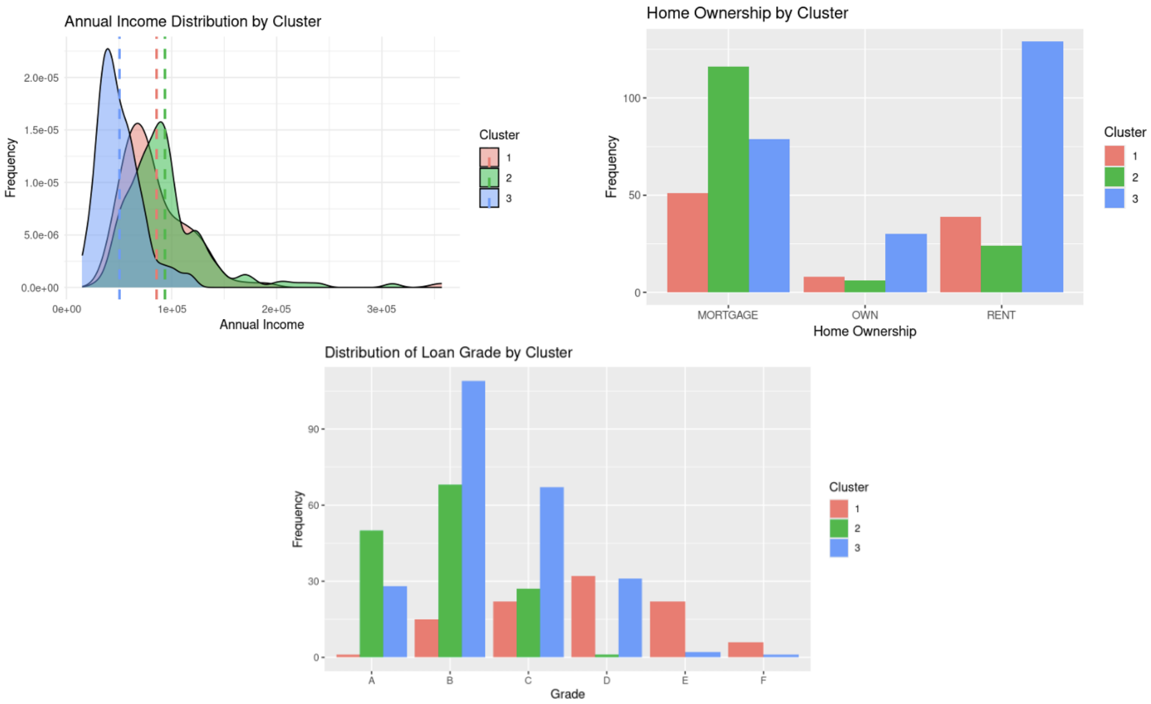


Figure 17: Graph for Targeted Marketing (Annual Income, Homeownership, Loan Grade)

Better Customer Satisfaction (Figure 18)

- Cluster 1: Provide personalized customer service with assigned financial advisers. Offer proactive advice and conduct financial evaluations to assist borrowers with larger loans and moderate incomes in maximizing their financial potential.

- Cluster 2: Offer responsive customer support with multiple communication channels and educational resources. Empower financially healthy borrowers with stable jobs to maintain their financial well-being.

- Cluster 3: Prioritize accessibility and affordability in customer support for financially vulnerable borrowers. Extend support hours and provide flexible repayment options, catering to cautious individuals with smaller loans and modest financial stability.

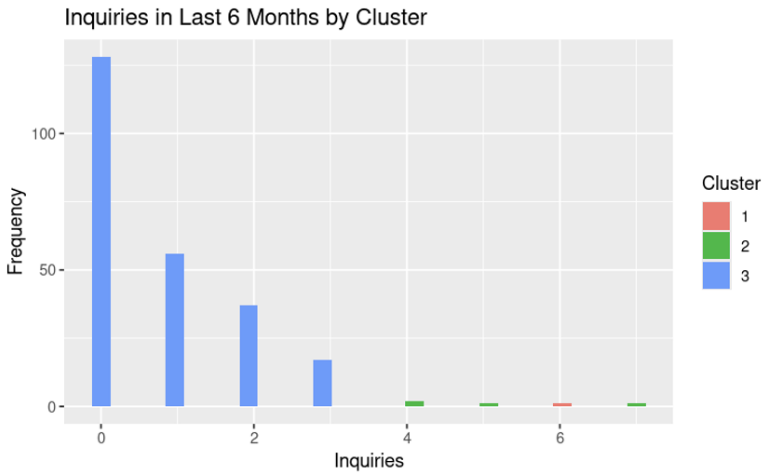


Figure 18: Graph for Customer Support (Inquiries)

8. Conclusion

The comprehensive analysis detailed in this report supports the creation of three distinct borrower clusters, offering insights into the financial behaviours and needs of customers. By validating the cluster analysis with internal methods, we have established a strong basis for implementing targeted business strategies. Our recommendations for personalized loan products, marketing strategies and customer support are tailored to each cluster's unique profile, thereby optimizing loan approval efficiency, risk assessments, and customer satisfaction.

9. Appendix

1. Records of the Meetings

Meeting Dates: Feb. 20
Meeting Goal: Meet and greet the group members and discuss the task allocation
Action Points for Each Group Member:
5563324: Data cleaning and feature selection
5574880: PCA and Factor Analysis
5534230: Data cleaning and feature selection
5572682: Introduction part in the report
5589303: PCA and Factor Analysis
5523411: PCA and Factor Analysis

Meeting Dates: Feb. 23
Meeting Goal: Review the data cleaning results together and finalize our feature selection.
Action Points for Each Group Member:
5563324: Sampling and remove outliers with different methods

5574880: Finish PCA and Factor Analysis
5534230: Sampling and remove outliers with different methods
5572682: Finish data cleaning and feature selection part in the report
5589303: Finish PCA and Factor Analysis
5523411: Finish PCA and Factor Analysis

Meeting Dates: Feb. 27
Meeting Goal: Review the PCA and FA results and decide which set of factors to use.
Action Points for Each Group Member:
5563324: Finish PCA part in the report
5574880: Start doing Cluster Analysis and interpret the factors
5534230: Finish FA part in the report
5572682: Finish FA part in the report
5589303: Interpret Clusters
5523411: Start doing Cluster Analysis and interpret the factors

---

Meeting Dates: Mar. 5
Meeting Goal: Review the Clustering results and try to interpret each cluster.
Action Points for Each Group Member:
5563324: Finalize FA part in the report, start writing the cluster part in the report
5574880: Finalize Cluster Analysis
5534230: Finish FA part in the report, start writing the cluster part in the report
5572682: Finish factor validation part in the report
5589303: Interpret Clusters and work on visualizations and analysis
5523411: Finalize Cluster Analysis

Meeting Dates: Mar. 12, 13, 14, 17
Meeting Goal: Finalize the interpretation of the clusters and finalize the report.
Action Points for Each Group Member:
5563324: Finalize the cluster validation part in the report, shorten and rephrase the report.
5574880: Finalize the interpretation of the clusters and validation result
5534230: Finalize the cluster part in the report, shorten and rephrase the report.
5572682: Adding comments to the finalized codes, shorten and rephrase the report
5589303: Finalize cluster visualizations, analysis, and recommendations
5523411: Finalize the interpretation of the clusters and validation result

2. Complete R Codes

Install packages

~~~
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

# Install and load necessary packages required for this analysis.

# install.packages('Hmisc')
# install.packages('emmeans')
# install.packages('tidyverse')
# install.packages('ggplot2')
# install.packages('lubridate')
# install.packages('grid')
# install.packages('gridExtra')
# install.packages('patchwork')
# install.packages('kableExtra')
# install.packages('cowplot')
# install.packages('knitr')
# install.packages('car')
# install.packages('readxl')
#install.packages("factoextra")
#install.packages(c("factoextra", "fpc", "NbClust"))
library(Hmisc)
library(emmeans)
library(tidyverse)
library(ggplot2)
library(lubridate)
library(grid)
library(gridExtra)
library(patchwork)
library(kableExtra)
library(cowplot)
library(car)
library(readxl)
library(caret)
library(mltools)
library(data.table)
~~~

~~~
library(psych)
library(psychTools)
library(GPArotation)
library(factoextra)
library(cluster)
library(dplyr)
library(factoextra)
library(dendextend)
options(width=100)
```

## Data Cleaning - Initial Preparation

Read the data in
```{r}
# read in the loan data given
data <- read_excel("loan_data_ADA_assignment.xlsx")
```

Check the structure and the summary of data
```{r}
str(data)
summary(data)
```

Describe data
```{r}
describe(data)
```

We found that there are many categorical or binary observation in the data set. We will deal with them one by one.


Check for duplicates
```{r}
#Check for duplicates on id column
duplicates_id <- duplicated(data$id)
~~~

~~~
any_duplicates_id <- any(duplicates_id) #FALSE; therefore no duplicates

#Check for duplicates on member_id column
duplicates_memberid <- duplicated(data$member_id)
any_duplicates_memberid <- any(duplicates_memberid) #FALSE; therefore no duplicates
```
~~~

We ensured that no duplicated record exists in the data. All ids are unique, representing each customer of the company

Since there are no duplicates, we moved on to remove 43 unnecessary columns, including nominal categorical columns (not suitable for encoding), binary variables (still categorical even after encoding), and those with unclear variable definition.

~~~
```{r}
#Delete unnecessary columns
data_filtered <- data %>% select( -acc_now_delinq,
-addr_state, -collection_recovery_fee, -collections_12_mths_ex_med, -delinq_2yrs, -desc, -dti,
-earliest_cr_line, -emp_title, -funded_amnt_inv, -issue_d, -last_credit_pull_d,
-last_pymnt_amnt, -last_pymnt_d, -loan_amnt, -member_id, -mths_since_last_major_derog,
-mths_since_last_delinq, -mths_since_last_record, -next_pymnt_d,
-pymnt_plan, -purpose, -policy_code, -revol_bal, -revol_util, -sub_grade,
-title, -tot_coll_amt, -total_credit_rv, -total_pymnt_inv, -total_rec_int, -emp_length,
-recoveries,
-total_rec_late_fee,
-zip_code, -loan_is_bad, -verification_status, -pub_rec,
-home_ownership, -loan_status, -term, -inq_last_6mths
)

```
~~~

Since grade is an ordinal variable, we encode it into numbers and assumed the gap=1.

~~~
```{r}
data_filtered$grade <- dplyr::recode(data_filtered$grade, "A" = 7, "B" = 6, "C" = 5, "D" = 4,
"E" = 3, "F" =2, "G" = 1)
```
~~~

We then removed all NA values in the filtered data.

```{r}
#removing all NA values
data_filtered_na_removed <- na.omit(data_filtered)
```

Random sampling to take 500 data for the following analysis.

```{r}
# random sample
set.seed (123)
sample <- data_filtered_na_removed [sample(nrow(data_filtered_na_removed), size = 500, replace = FALSE), ]
```

Since Cluster Analysis is sensitive to outliers, we check outliers using Mahalanobi distance.

```{r}
# calculate Mahalanobis distance
Maha <- mahalanobis(sample,colMeans(sample),cov(sample))
print(sample)
```

If the Maha p-value is smaller than .001, we considered it as an outlier and removed them.

```{r}
# calculate the Mahalanobis p-value
MahaPvalue <-pchisq(Maha,df=10,lower.tail = FALSE)
print(MahaPvalue)
# print those with p-value lower than 0.001
print(sum(MahaPvalue<0.001))
```

```{r}
# combine the p-value to the data set
sample_clean <-cbind(sample, Maha, MahaPvalue)
```

```{r}
# remove those with p-value smaller than .001
```

~~~
sample_clean <- sample_clean [sample_clean$MahaPvalue>= 0.001,]
```
~~~

~~~
```{r}
# remove the columns combined so that this will not affect the following analysis.
sample_clean$Maha <-NULL
sample_clean$MahaPvalue <- NULL
sample_clean_id <- sample_clean
sample_clean <- sample_clean %>% select(-id)
```
~~~

Standardizing the data

~~~
```{r}
#Standardise each variable with mean of 0 and sd of 1
sample_clean_std <-scale(sample_clean)
```
~~~

## Assumption Check for Cluster Analysis

Since multicollinearity is undesirable for cluster analysis, we need to check the pairwise correlations between variables.

~~~
```{r}
# correlation check
LoanMatrix<-cor(sample_clean_std)
print(LoanMatrix)
```
~~~

~~~
```{r}
# round the correlation to two decimals
round(LoanMatrix, 2)
```
~~~

~~~
```{r}
lowerCor(sample_clean_std)
```
~~~

As can be seen in the correlation matrix, there exist plenty of highly correlated variables, so we cannot carry out the Cluster Analysis right away. We use Principal Component and Factor Analysis to adress this issue and reduce dimensionality.

## PCA / FA for Addressing Multicollinearity

### Assumption Check for PCA / FA

Since PCA / FA requires sufficient correlation between variables, we used KMO statistic and Barlett's test to check if the data set contains enough correlation. Here we are looking for KMO score > 0.5 and Barlette p-value < .05.

```{r}
# use KMO measurement
KMO(sample_clean)


#KMO value is greater than 0.5, so we can conclude that the data set contain enough highly correlated variables.
```

```{r}
# Barlett's test
cortest.bartlett(sample_clean)


#Barlett's test p-value is smaller than .05, so we can say that the correlation matrix is different than an identity matrix.
```

### Principal Component Analysis (PCA)

Checking all variables to draw scree plot and decide on how many PCs should be decided.

```{r}
pcModel<-principal(sample_clean_std, 10, rotate="none", weights=TRUE, scores=TRUE)
print(pcModel)
```

When we look at the all variables, we see that cumulative variance reached 1.00 in PC8, which means first 8 PCs represents all information. Also, cumulative variance reached more than 0.6

after PC2; only the first 3 PCs have an Eigenvalues above 1.

```{r}
# use cut=0.3 to check cross-loading
print.psych(pcModel, cut=0.3, sort=TRUE)
```

There are so many cross-loading across PCs. It is hard to interpret the PCs and starting from PC7, no variables is correlated to the PCs. So we will try reduce the number of PC and do the PCA again.

Drawing scree plot

```{r}
plot(pcModel$values, type="b")
```

According to scree plot, We can conclude that ideal number of PCs might be 4 or 5, since the line started flattening out after these points.

We run PCA model again with 4 PCs to see final factor loading matrix.

```{r}
pcModel_4pc <- principal(sample_clean_std, 4, rotate="none", weights=TRUE, scores=TRUE)
print.psych(pcModel_4pc, cut=0.4, sort=TRUE)

#There are so many cross-loadings across PCs. Also, correlation between variables and PCs are significantly different. Because of number of PCs, cross-loadings and high correlations, it is very hard to interpret. We will conduct Factor Analysis to interpret variables and reduce dimension.
```

There is still a lot of cross-loading issue and it is impractical to interpret these PCs. Therefore, we move on to conduct factor analysis to try addressing multicollinearity issue.

# PC Analysis

Checking all variables to draw scree plot and decide on how many PCs should be decided.

```{r}
pcModel<-principal(sample_clean_std, 10, rotate="none", weights=TRUE, scores=TRUE)
print(pcModel)
```

~~~
# When we look at the all variables, we see that cumulative variance reached 1.00 in PC8, which means first 8 PCs represents all information.

```


```{r}
print.psych(pcModel, cut=0.3, sort=TRUE)
#When we check the Eigenvalues of PCs, first 3 PC have eigenvalues >= 1 for standardized data. And, cumulative variance is higher than 0.6 after PC2.
```



Drawing scree plot
```{r}
plot(pcModel$values, type="b")

#When we look at the scree plots, we can see that after PC5, the scree plot flattens out. We can conclude that the optimum number of PCs might be 5.

```

We can conclude that total number of PCs might be 4 or 5.

Lets run PCA model to see final factor loading matrix.

```{r}
pcModel_4pc <- principal(sample_clean_std, 4, rotate="none", weights=TRUE, scores=TRUE)
print.psych(pcModel_4pc, cut=0.4, sort=TRUE)

#There are so many cross-loadings across PCs. Also, correlation between variables and PCs are significantly different. Because of number of PCs, cross-loadings and high correlations, it is very hard to interpret. We will conduct Factor Analysis to interpret variables and reduce dimension.
```

# FACTOR ANALYSIS
We will do factor analysis to interpret the correlations among the variables. We will try different
~~~

rotations in PC and Maximum Likelihood extraction to get interpretable components.

## Factor Analysis with Oblique rotation

Four Factors Solution with Oblimin Rotation (pretty good, emp_length not include)

```{r}
fa4o<-(fa(sample_clean_std,4, n.obs=483, rotate="oblimin", fm="ml"))
print.psych(fa4o, cut=0.3,sort="TRUE")
fa.diagram(fa4o)
```

Four Factors Solution with Promax Rotation

```{r}
fa4p<-(fa(sample_clean_std,4, n.obs=483, rotate="promax", fm="ml"))
print.psych(fa4p, cut=0.3,sort="TRUE")
fa.diagram(fa4p)
```

## Factor Analysis with Orthogonal rotation

Five Factors Solution with Varimax Rotation (pretty good, emp_length not include)x

```{r}
fa4v<-(fa(sample_clean_std,4, n.obs=483, rotate="varimax", fm="ml"))
print.psych(fa4v, cut=0.3,sort="TRUE")
fa.diagram(fa4v)
```

Four Factors Solution with Quartimax Rotation

```{r}
fa4q<-(fa(sample_clean_std,4, n.obs=483, rotate="quartimax", fm="ml"))
print.psych(fa4q, cut=0.3,sort="TRUE")
fa.diagram(fa4q)
```

~~~
```

Four Factors Solution with Equimax Rotation

```{r}
fa4e<-(fa(sample_clean_std,4, n.obs=483, rotate="equimax", fm="ml"))
print.psych(fa4e, cut=0.3,sort="TRUE")
fa.diagram(fa4e)
```



# PC extraction with Oblique rotation
Four factors solution with oblimin (very very good).
```{r}
pcModel4o<-principal(sample_clean_std, 4, rotate="oblimin")
print.psych(pcModel4o, cut=0.3, sort=TRUE)
fa.diagram(pcModel4o)
```

Four factors solution with promax (very very good) >> our pick
```{r}
pcModel4p<-principal(sample_clean_std, 4, rotate="promax")
print.psych(pcModel4p, cut=0.3, sort=TRUE)
fa.diagram(pcModel4p)
```

Four factors solution with equimax (too many cross loadings)
```{r}
pcModel4e<-principal(sample_clean_std, 4, rotate="equimax")
print.psych(pcModel4e, cut=0.3, sort=TRUE)
```



#PC extraction with Orthogonal rotation
Four factors solution (one cross loadings)
```{r}
pcModel4q<-principal(sample_clean_std, 4, rotate="quartimax")
print.psych(pcModel4q, cut=0.3, sort=TRUE)
~~~

~~~
fa.diagram(pcModel4q)
```
~~~

Four factors solution with varimax (pretty good, slightly cross-loading)

~~~
```{r}
pcModel4v<-principal(sample_clean_std, 4, rotate="varimax")
print.psych(pcModel4v, cut=0.3, sort=TRUE)
fa.diagram(pcModel4v)
```
~~~

We can use the factor scores for further analysis, before doing that we need to add them into our data frame:

~~~
```{r}
fscores <- pcModel4p$scores
```
~~~

First, we describe the factor scores.

~~~
```{r}
describe(fscores)
headTail(fscores)

```
~~~

We check assumptions to see whether the data are suitable for Cluster Analysis:

~~~
```{r}
FscoresMatrix<-cor(fscores)
print(FscoresMatrix)
```
~~~

~~~
```{r}
round(FscoresMatrix, 2)
```
~~~

~~~
```{r}
lowerCor(fscores)
```
~~~

~~~~
```{r}
KMO(fscores)
```
## Select another sample to validate FA analysis

Random internal sampling
```{r}
set.seed (123)
sample_vald <- sample_clean_std [sample(nrow(sample_clean_std), size = 100, replace =
FALSE), ]
```



Four factors solution with oblimin --> not really verified the cluster
```{r}
pcModel4o_vald<-principal(sample_vald, 4, rotate="oblimin")
print.psych(pcModel4o_vald, cut=0.3, sort=TRUE)
fa.diagram(pcModel4o_vald)
```



Four factors solution with promax --> use this, no cross-loading and verified successfully
```{r}
pcModel4p_vald<-principal(sample_vald, 4, rotate="promax")
print.psych(pcModel4p_vald, cut=0.3, sort=TRUE)
fa.diagram(pcModel4p_vald)

```



```{r}
sample_clean_id <- cbind(sample_clean_id, fscores)
sample_clean_id <- sample_clean_id %>% select(id, RC1, RC2, RC3, RC4)
```
~~~~

#Clustering

Define linkage methods

~~~
```{r}
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")
```
~~~

Function to compute agglomerative coefficient

~~~
```{r}
ac <- function(x) {
  agnes(fscores, method = x)$ac
}
```
~~~

Calculate agglomerative coefficient for each clustering linkage method

~~~
```{r}
sapply(m, ac)
```
~~~

We can see that Ward’s minimum variance method produces the highest agglomerative coefficient, thus we’ll use that as the method for our final hierarchical clustering:

Determine the Optimal Number of Clusters.
To determine how many clusters the observations should be grouped in, we can use a metric known as the gap statistic, which compares the total intra-cluster variation for different values of k with their expected values for a distribution with no clustering.

Calculate gap statistic for each number of clusters (up to 10 clusters) for both hierarchical and non-hierarchical method

~~~
```{r}
gap_stat_k <- clusGap(fscores, FUN = hcut, nstart = 25, K.max = 10, B = 50)
```
~~~

produce plot of clusters vs. gap statistic

~~~
```{r}
fviz_gap_stat(gap_stat_k)
```
~~~

From the plot we can see that the gap statistic is high at k = 3 or 4   clusters. Thus, we'll choose to group our observations into 4 distinct clusters.

Finding distance matrix

~~~
```{r}
distance_mat <- dist(fscores, method = 'euclidean')
```
~~~

Fitting Hierarchical clustering Model to dataset

~~~
```{r}
set.seed(240)    # Setting seed
Hierar_cl <- hclust(distance_mat, method = "ward")
Hierar_cl
```
~~~

Plotting dendrogram

~~~
```{r}
plot(Hierar_cl)
```
~~~

# K-means

~~~
```{r}
set.seed(111)
k2 <- kmeans(fscores, 2, nstart = 25)
k3 <- kmeans(fscores, 3, nstart = 25)
k4 <- kmeans(fscores, 4, nstart = 25)
k5 <- kmeans(fscores, 5, nstart = 25)



# plots to compare
p1 <- fviz_cluster(k2, geom = "point", data = fscores) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",    data = fscores) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",    data = fscores) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",    data = fscores) + ggtitle("k = 5")
grid.arrange(p1, p2, p3, p4, nrow = 2)
print(k3)
```
~~~

# Cluster Validation

## Sampling the data (internal validation)

```{r}
set.seed(111)
fscores_indices <- sample(nrow(fscores), 250)
fscores_vald <- fscores[fscores_indices, ]

k2_vald <- kmeans(fscores_vald, 2, nstart = 25)
k3_vald <- kmeans(fscores_vald, 3, nstart = 25)
k4_vald <- kmeans(fscores_vald, 4, nstart = 25)
k5_vald <- kmeans(fscores_vald, 5, nstart = 25)



# plots to compare
p1 <- fviz_cluster(k2_vald, geom = "point", data = fscores_vald) + ggtitle("k = 2")
p2 <- fviz_cluster(k3_vald, geom = "point",   data = fscores_vald) + ggtitle("k = 3")
p3 <- fviz_cluster(k4_vald, geom = "point",   data = fscores_vald) + ggtitle("k = 4")
p4 <- fviz_cluster(k5_vald, geom = "point",   data = fscores_vald) + ggtitle("k = 5")
grid.arrange(p1, p2, p3, p4, nrow = 2)


print(k3_vald)
```

```{r}
# Assuming you already have clustering results from k3 and k3_vald
# fscores_indices are the indices of the samples used for clustering in k3_vald within the original dataset

# Extract the clustering results for the corresponding samples from the full dataset
```

~~~
full_dataset_assignments <- k3$cluster[fscores_indices]

# Extract the clustering results from the validation dataset
validation_dataset_assignments <- k3_vald$cluster

# Function to remap the cluster numbers of k3_vald to match the clusters of k3
# For instance, k3's group 1 now corresponds to k3_vald's group 2, etc.
relabel_vald_clusters <- function(cluster_number) {
  return(switch(as.character(cluster_number),
                '1' = 1,   # Remap group 1 to group 3
                '2' = 3,   # Remap group 2 to group 1
                '3' = 2)) # Remap group 3 to group 2
}

# Apply this mapping to the validation dataset's clustering results
relabelled_validation_assignments <- sapply(validation_dataset_assignments, relabel_vald_clusters)

# Calculate the consistency rate as an indicator of accuracy
consistency_rate <- mean(full_dataset_assignments == relabelled_validation_assignments)
print(paste("Consistency Rate:", consistency_rate))

# For a more detailed analysis, you can generate a confusion matrix to view the precise matching
library(caret)
conf_mat <- confusionMatrix(as.factor(full_dataset_assignments), as.factor(relabelled_validation_assignments))
print(conf_mat)


```



```{r}
kmeans_clusters <- k3$cluster

sample_clean_id <- cbind(sample_clean_id, kmeans_cluster = kmeans_clusters)
```
~~~

Merge data

~~~~
```{r}

complete_data <- merge(sample_clean_id, data, by = "id")

```


```{r}
# Personalized Loan Product
# Distribution of Interest Rate by Cluster
ggplot(complete_data, aes(x = int_rate, fill = factor(kmeans_cluster))) +
  geom_histogram(position = "identity", bins = 30) +
  labs(title = "Distribution of Interest Rate by Cluster",
       x = "Interest Rate", y = "Frequency", fill = "Cluster")
```


```{r}
# Calculate average interest rate for each cluster
avg_int_rate <- complete_data %>%
  group_by(kmeans_cluster) %>%
  summarise(avg_int_rate = mean(int_rate))

# Create density plot with average lines
ggplot(complete_data, aes(x = int_rate, fill = factor(kmeans_cluster))) +
  geom_density(alpha = 0.5) +
  geom_vline(data = avg_int_rate, aes(xintercept = avg_int_rate, color = factor(kmeans_cluster)), linetype = "dashed", size = 1) +
  labs(title = "Distribution of Interest Rate by Cluster",
       x = "Interest Rate", y = "Density", fill = "Cluster") +
  scale_color_discrete(name = "Cluster") +
  theme_minimal()

```


```{r}
ggplot(complete_data, aes(x = factor(kmeans_cluster), y = int_rate, fill = factor(kmeans_cluster))) +
~~~~

~~~
   geom_boxplot() +
   labs(title = "Distribution of Interest Rate by Cluster",
          x = "Cluster", y = "Interest Rate", fill = "Cluster")
```


```{r}
# Distribution of Loan Amount by Cluster
ggplot(complete_data, aes(x = loan_amnt, fill = factor(kmeans_cluster))) +
   geom_histogram(position = "identity", bins = 30) +
   labs(title = "Distribution of Loan Amount by Cluster",
          x = "Loan Amount", y = "Frequency", fill = "Cluster") +
   facet_wrap(~ kmeans_cluster)

```


```{r}
# Calculate average loan amount for each cluster
avg_loan_amount <- complete_data %>%
   group_by(kmeans_cluster) %>%
   summarise(avg_loan_amount = mean(loan_amnt))

# Create density plot with average lines
ggplot(complete_data, aes(x = loan_amnt, fill = factor(kmeans_cluster))) +
   geom_density(alpha = 0.5) +
   geom_vline(data = avg_loan_amount, aes(xintercept = avg_loan_amount, color =
factor(kmeans_cluster)),
                  linetype = "dashed", size = 1) +
   labs(title = "Distribution of Loan Amount by Cluster",
          x = "Loan Amount", y = "Density", fill = "Cluster") +
   scale_color_discrete(name = "Cluster") +
   theme_minimal()

```


```{r}
ggplot(complete_data, aes(x = factor(kmeans_cluster), y = loan_amnt, fill =
factor(kmeans_cluster))) +
   geom_boxplot() +
~~~

~~~
  labs(title = "Distribution of Loan Amount by Cluster",
         x = "Cluster", y = "Loan Amount", fill = "Cluster")


```



```{r}
# Distribution of Loan Term by Cluster
ggplot(complete_data, aes(x = factor(term), fill = factor(kmeans_cluster))) +
  geom_bar(position = "dodge") +
  labs(title = "Distribution of Loan Term by Cluster",
         x = "Term", y = "Frequency", fill = "Cluster")

```


```{r}
# Purpose of Loans by Cluster
ggplot(complete_data, aes(x = purpose, fill = factor(kmeans_cluster))) +
  geom_bar(position = "dodge") +
  labs(title = "Purpose of Loans by Cluster",
         x = "Purpose", y = "Frequency", fill = "Cluster") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```{r}
ggplot(complete_data, aes(x = purpose)) +
  geom_bar(aes(fill = purpose)) +
  labs(title = "Purpose of Loans by Cluster",
         x = "Purpose", y = "Frequency", fill = "Purpose") +
  facet_wrap(~ kmeans_cluster) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```{r}
# Targeted Marketing
~~~

~~~
# Annual Income Distribution by Cluster
ggplot(complete_data, aes(x = annual_inc, fill = factor(kmeans_cluster))) +
  geom_histogram(position = "identity", bins = 30) +
  labs(title = "Annual Income Distribution by Cluster",
       x = "Annual Income", y = "Frequency", fill = "Cluster")
```



```{r}
# Home Ownership by Cluster
ggplot(complete_data, aes(x = home_ownership, fill = factor(kmeans_cluster))) +
  geom_bar(position = "dodge") +
  labs(title = "Home Ownership by Cluster",
       x = "Home Ownership", y = "Frequency", fill = "Cluster")
```

```{r}
# Employment Length by Cluster
ggplot(complete_data, aes(x = emp_length, fill = factor(kmeans_cluster))) +
  geom_bar(position = "dodge") +
  labs(title = "Employment Length by Cluster",
       x = "Employment Length", y = "Frequency", fill = "Cluster") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```{r}
# Better Customer Satisfaction
# Loan Status by Cluster
ggplot(complete_data, aes(x = loan_status, fill = factor(kmeans_cluster))) +
  geom_bar(position = "dodge") +
  labs(title = "Loan Status by Cluster",
       x = "Loan Status", y = "Frequency", fill = "Cluster") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```{r}
# Inquiries in Last 6 Months by Cluster
~~~

~~~
ggplot(complete_data, aes(x = inq_last_6mths, fill = factor(kmeans_cluster))) +
  geom_histogram(position = "identity", bins = 30) +
  labs(title = "Inquiries in Last 6 Months by Cluster",
       x = "Inquiries", y = "Frequency", fill = "Cluster")
```

```{r}
# Earliest Credit Line by Cluster
complete_data$earliest_cr_line_year <- as.numeric(substr(complete_data$earliest_cr_line, 1,
4))
ggplot(complete_data, aes(x = earliest_cr_line_year, fill = factor(kmeans_cluster))) +
  geom_histogram(position = "identity", bins = 30) +
  labs(title = "Earliest Credit Line by Cluster",
       x = "Year", y = "Frequency", fill = "Cluster")

```

```{r}
ggplot(complete_data, aes(x = grade, fill = factor(kmeans_clusters))) +
  geom_bar() + facet_grid(kmeans_clusters)+
  labs(title = "Frequency of Grades", x = "Grade", y = "Frequency", fill = "Cluster")
```


```{r}
# Assuming complete_data is your dataset
summary_data <- complete_data %>%
  group_by(kmeans_cluster, loan_status, grade) %>%
  summarise(count = n())

ggplot(summary_data, aes(x = loan_status, y = grade, fill = count)) +
  geom_tile() +
  facet_grid(kmeans_cluster ~ .) +   # Facet based on kmeans_cluster
  labs(title = "Loan Status vs. Grade", x = "Loan Status", y = "Grade", fill = "Frequency") +
  geom_text(aes(label = count), vjust = 1) +
  scale_fill_gradient(low = "white", high = "red")
~~~