1 Data understanding

Our dataset consists of a variety of customer data points describing the loan parameters (loan amount, interest rate, purpose, loan status etc.) collected by STF for previously issued loans. There are 53 attributes and 50,000 observations. Given the nature of clustering, I would need to eliminate attributes that do not contribute valuable information with respect to our objective.

2 Data preparation

Upon thorough analysis I have decided to exclude 26 attributes from the dataset. Table 1 below explains our reasons for exclusion.

Attribute	Reason
Id, member_id	The IDs are unique identifiers and are irrelevant
issue_d, last_pymnt_d, next_pymnt_d, last_credit_pull_d, earliest_cr_line	These columns represent dates which do not contribute to our cluster analysis
funded_amnt, funded_amnt_inv	Both values are similar to loan_amnt
emp_title, emp_length	The customer's job title and years of employment do not contribute to our objective
term	Only 2 values
grade	Attribute sub_grade provided is more detailed
desc	The content in this column is vague and unstructured
purpose, title	Data provided in columns cannot be ranked
mths_since_last_delinq, mths_since_last_record, mths_since_last_major_derog	These columns contain most NA values
last_pymnt_amnt, policy_code, tot_coll_amt, pymnt_plan, addr_state, zip_code, delinq_2yrs, inq_last_6mths	The data does not describe any useful information for our cluster analysis
total_rec_prncp, total_rec_int, total_rec_late_fee, recoveries, collection_recovery_fee	These columns provide a breakdown of the loan payments. This is not required for the analysis
total_pymnt_inv	This column data is available in column total_pymnt
collections_12_mths_ex_med, pub_rec	The data is highly imbalanced
loan_is_bad	A detailed view of this data is available in loan_status

Table 1: Reasoning for attribute exclusion

- Handling missing values

14,678 observations have been removed from the dataset due to occurrence of NA values.

- Factorization

All non-numeric values have been factorized.

- Encoding

Attribute	Encoding format
sub_grade	I assume that the grades A-G represent a descending order of quality of loan. Hence, A1=1, A2=2….G5=35

home_ownership	Descending order of financial liability. Own =1, Rent = 2, Mortgage = 3, Other = 4/ Remove NONE
verification_status	Decreasing order of reliability of profile. 1=verify, 2= source verify, 3 = not verify
loan_status	Ordered by logical progression from most positive to the most negative outcome on a loan repayment lifecycle.

Table 2: Encoding format for nominal attributes

- Sampling

Three datasets are used for our analysis. First, a random sample of 500 observations from the dataset was taken as a training dataset. Another random sample of 500 observations was taken to perform external validation of factor analysis. Lastly, 100 observations Ire sampled randomly from the training dataset to conduct an internal validation of the cluster analysis.

- Normalization

The numeric attributes have been normalized using scale() to support scalability and eliminate the influence of varying scales betIen each attribute.

- Outlier detection

By undertaking the Mahalanobi distance measure and its relative p-values of the training set yielded 24 outliers (p-value <0.01) (Refer Appendix 9.1 for results). These Ire removed from the data frame.

- Multicollinearity

From the correlation matrix (Figure 1), I can observe some highly correlated pairs of attributes with correlation score > 0.8. The KMO test was also conducted to evaluate the correlation among attributes (Figure 2). The value for the sample is 0.66 which indicates there is a high level of correlation within the variables.

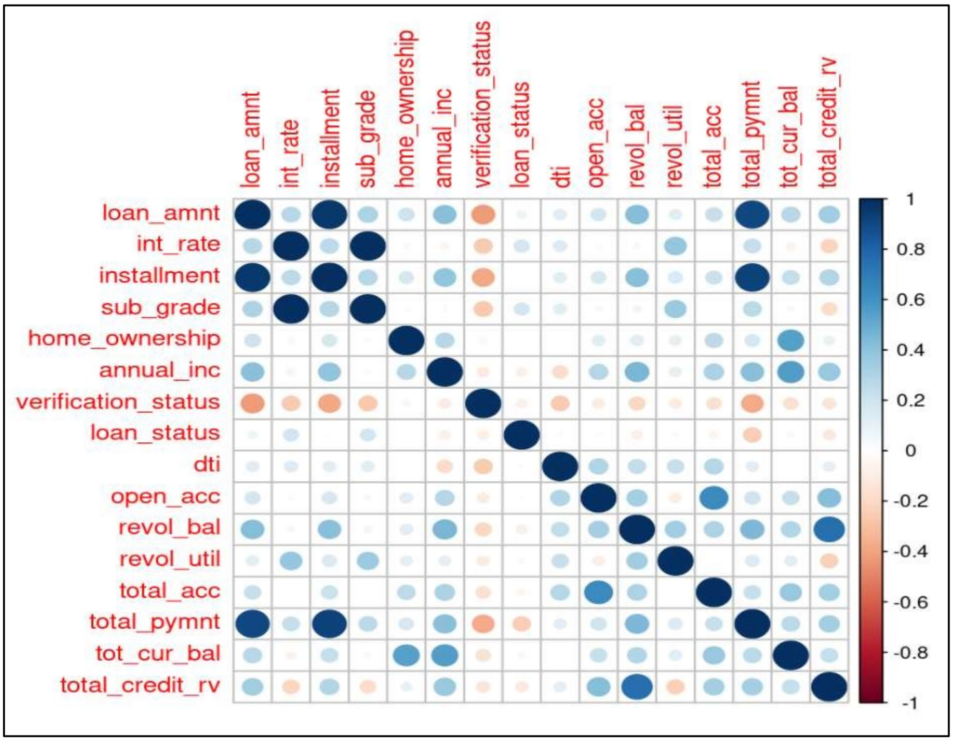


Figure 1:Multicollinearity heatmap of attributes

```
KMO(filtered_norm_samp_df_loan)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = filtered_norm_samp_df_loan)
## Overall MSA =  0.66
## MSA for each item =
##          loan_amnt            int_rate         installment           sub_grade
##               0.77                0.58                0.81                0.59
##     home_ownership          annual_inc verification_status         loan_status
##               0.70                0.77                0.89                0.17
##                dti            open_acc           revol_bal          revol_util
##               0.59                0.71                0.57                0.36
##          total_acc         total_pymnt         tot_cur_bal     total_credit_rv
##               0.77                0.73                0.73                0.54
```

Figure 2: Kaiser-Meyer-Olkin (KMO) test results

2.1 Principal Component Analysis (PCA)

PCA is conducted to reduce the dimensionality of the training sample. Theory states that I should keep information (cumulative variance) amounting to 60% or higher contained within the original variables. Our PCA output shows that this can be achieved with 4 principal components (PC)

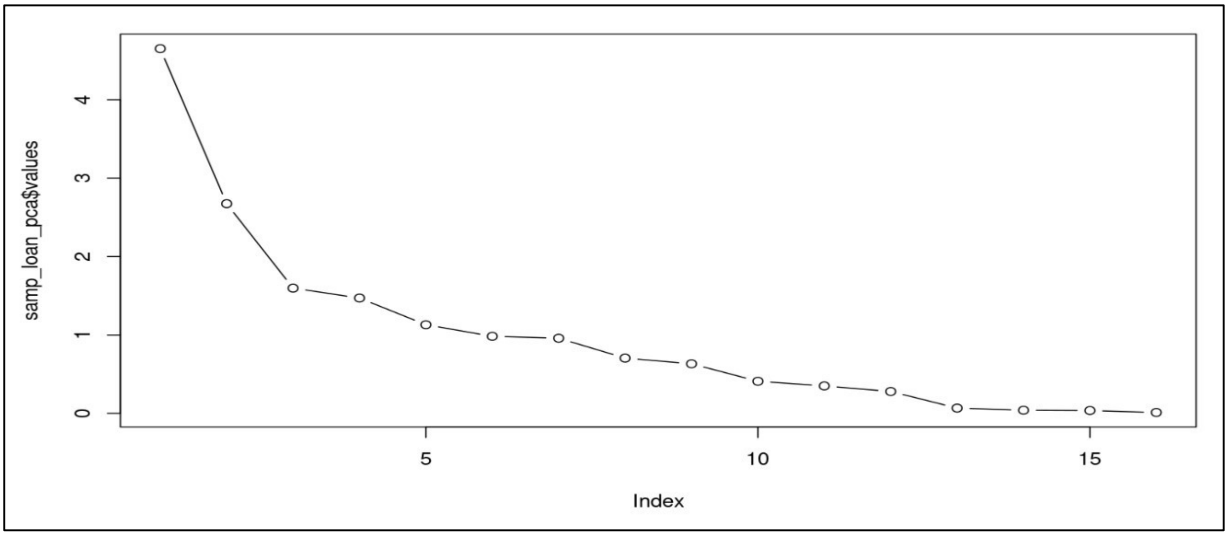


Figure 3: Scree plot of PCA

```
Principal Components Analysis
Call: principal(r = filtered_norm_samp_df_loan, nfactors = 16, rotate = "none",
    scores = TRUE, weights = TRUE)
Standardized loadings (pattern matrix) based upon correlation matrix

                       PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11 PC12 PC13 PC14
SS loadings           4.65 2.67 1.60 1.47 1.13 0.98 0.96 0.71 0.63 0.41 0.35 0.28 0.07 0.04
Proportion Var        0.29 0.17 0.10 0.09 0.07 0.06 0.06 0.04 0.04 0.03 0.02 0.02 0.00 0.00
Cumulative Var        0.29 0.46 0.56 0.65 0.72 0.78 0.84 0.89 0.93 0.95 0.97 0.99 0.99 1.00
Proportion Explained  0.29 0.17 0.10 0.09 0.07 0.06 0.06 0.04 0.04 0.03 0.02 0.02 0.00 0.00
Cumulative Proportion 0.29 0.46 0.56 0.65 0.72 0.78 0.84 0.89 0.93 0.95 0.97 0.99 0.99 1.00
                      PC15 PC16
SS loadings           0.04 0.01
Proportion Var        0.00 0.00
Cumulative Var        1.00 1.00
Proportion Explained  0.00 0.00
Cumulative Proportion 1.00 1.00

Mean item complexity =  3.7
Test of the hypothesis that 16 components are sufficient.

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

Figure 4: Cumulative variance scores highlighting 4 PCAs

2.2 **Factor Analysis (FA)**

FA is carried out to group highly correlated variables and find structure among variables. Initially, I conducted the analysis using 3 and 4 PCs and the results show that 4PCs are more appropriate. In addition, I observed there is high cross-loading for total_credit_rv and revol_bal, hence they Ire removed from the dataset (Refer Figure 5, Appendix 9.2 for more results).

```r
pcModel4q<-principal(filtered_norm_samp_df_loan, 4, rotate="quartimax")
print.psych(pcModel4q, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = filtered_norm_samp_df_loan, nfactors = 4, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   RC1   RC2   RC3   RC4   h2    u2 com
## total_pymnt           14  0.94                   0.90 0.099 1.0
## installment            3  0.93                   0.90 0.104 1.1
## loan_amnt              1  0.93                   0.90 0.099 1.1
## revol_bal             11  0.53         0.50      0.56 0.436 2.2
## verification_status    7 -0.45                   0.33 0.666 2.2
## int_rate               2        0.90             0.85 0.155 1.1
## sub_grade              4        0.88             0.83 0.167 1.1
## revol_util            12        0.58             0.41 0.586 1.4
## loan_status            8        0.34             0.13 0.866 1.3
## open_acc              10               0.79      0.67 0.335 1.1
## total_acc             13               0.73 0.33 0.65 0.346 1.5
## dti                    9               0.65      0.56 0.436 1.7
## total_credit_rv       16  0.47 -0.46  0.53       0.72 0.283 2.9
## tot_cur_bal           15                    0.82 0.74 0.256 1.2
## home_ownership         5                    0.76 0.59 0.405 1.0
## annual_inc             6  0.47              0.60 0.63 0.369 2.2
```

Figure 5: Cross loading observed on 4 factor orthogonal rotation

Following the removal of cross-loading variables, PC method was conducted again with orthogonal and oblique rotation for 4 factors. When choosing the most suitable factors, theory states that each variable must contain significant loadings towards a single factor. Based on our results, 4-factor oblique rotation satisfied this requirement while maintaining maximum difference in loading for variables present in more than one factor (Figures 6. and 7.). Furthermore, the external validation conducted to verify these results confirms the representation of each PC (Appendix 9.2.9)

```r
pcModel4o1<-principal(fa_loan, 4, rotate="oblimin")
print.psych(pcModel4o1, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = fa_loan, nfactors = 4, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   TC1   TC2   TC4   TC3   h2    u2 com
## total_pymnt           13  0.98                   0.94 0.064 1.0
## installment            3  0.95                   0.93 0.075 1.0
## loan_amnt              1  0.94                   0.92 0.078 1.0
## verification_status    7 -0.43                   0.36 0.639 2.2
## int_rate               2        0.92             0.88 0.120 1.0
## sub_grade              4        0.91             0.87 0.130 1.0
## revol_util            11        0.64             0.42 0.583 1.5
## loan_status            8        0.40             0.16 0.837 1.7
## tot_cur_bal           14              0.83       0.74 0.256 1.0
## home_ownership         5              0.77       0.57 0.433 1.1
## annual_inc             6  0.35        0.63       0.63 0.370 1.7
## open_acc              10                    0.81 0.70 0.301 1.1
## total_acc             12                    0.77 0.74 0.259 1.2
## dti                    9             -0.33  0.71 0.59 0.407 1.6
```

Figure 6: 4 factor oblique rotation

```
pcModel4q1<-principal(fa_loan, 4, rotate="quartimax")
print.psych(pcModel4q1, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = fa_loan, nfactors = 4, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   RC1   RC2   RC4   RC3   h2    u2 com
## total_pymnt           13  0.96                    0.94 0.064 1.0
## installment            3  0.95                    0.93 0.075 1.1
## loan_amnt              1  0.94                    0.92 0.078 1.1
## verification_status    7 -0.47                    0.36 0.639 2.2
## int_rate               2        0.92              0.88 0.120 1.1
## sub_grade              4        0.91              0.87 0.130 1.1
## revol_util            11        0.60              0.42 0.583 1.3
## loan_status            8        0.37              0.16 0.837 1.4
## tot_cur_bal           14              0.82        0.74 0.256 1.2
## home_ownership         5              0.74        0.57 0.433 1.1
## annual_inc             6  0.43        0.65        0.63 0.370 1.9
## open_acc              10                    0.81 0.70 0.301 1.2
## total_acc             12              0.33  0.78 0.74 0.259 1.4
## dti                    9                    0.68 0.59 0.407 1.6
```

Figure 7: 4 factor orthogonal rotation

3.3 Interpretation of factors

Interpretation of the factors is based on the grouping of attributes shown in Figure 8.

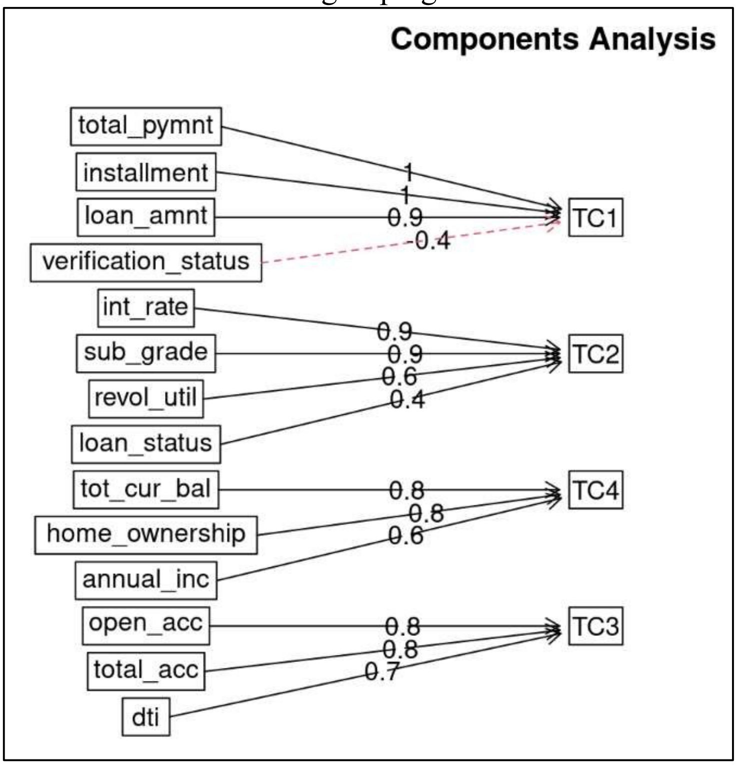


Figure 8: Grouping of attributes into 4 factors

- TC1 – Financial Commitment Profile

This factor is characterized by high loadings from total payment, loan payment, installment and verification status. It reflects the borroIr's financial engagement and obligations associated with the loan. This profile captures the monetary aspects of the loan, highlighting how these variables interact to define the borroIr's repayment structure and financial responsibility towards the loan.

- TC2 – BorroIr Credibility

These variables combined provide a comprehensive understanding of the risk associated with lending to a particular individual. They evaluate the likelihood of the borroIr repaying the debt. Due to our encoding method, the higher value in this group indicates more default risk associated with the borroIr.

- TC3 – Financial Management/Responsibility

This grouping of variables can be used by lenders to help gauge a borroIr's credit management skills, experience with handling credit and overall financial stability. It helps build a comprehensive picture of a borroIr's financial responsibility, stability and risk level. The higher value in this group indicates that borroIrs are likely to depend more on debt.

- TC4 – Net Worth

These variables describe the various segments that contribute to a borroIr's net worth.

4 Cluster Analysis

4.1 Linkage method

Figure 9. shows that ward's minimum variance method produces the highest agglomerative coefficient, hence I will be using this linkage method for further analysis.

```
##   average    single  complete      ward
## 0.8451020 0.6941339 0.9199687 0.9764011
```

Figure 9: List of agglomerative coefficients for different linkage methods

4.2 Number of clusters

Dendrogram plots can be used to determine the optimal number of clusters. I have generated dendrogram plots using 3 distance measures (Euclidean, Manhattan, Maximum). Based on our analysis, the stopping criteria of the largest increase in height indicates that 3 and 4 clusters are optimal (Refer Figures 10 & 11, Appendix 9.3).

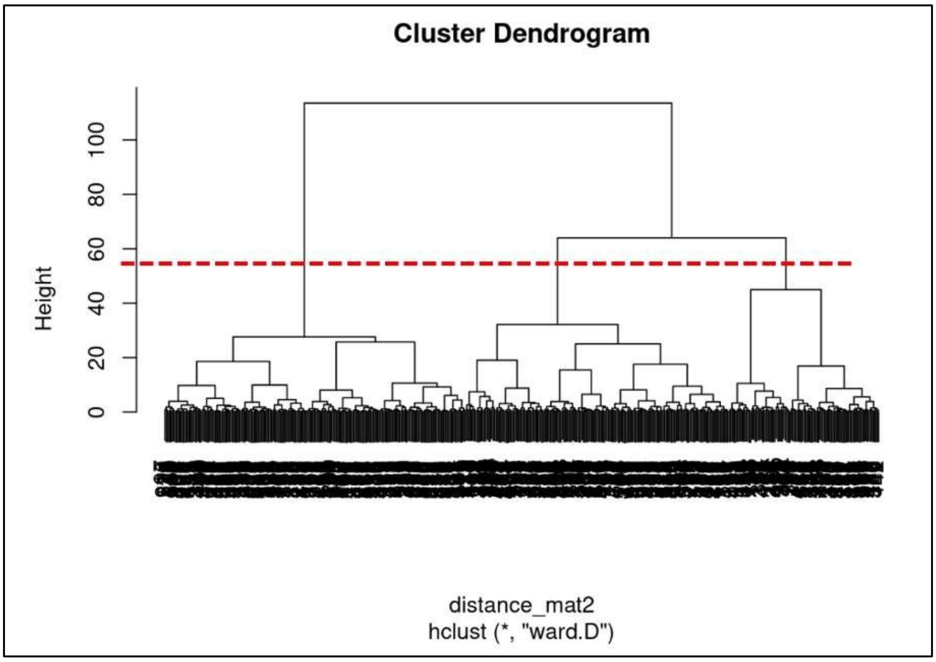


Figure 10: Dendrogram using maximum distance measure

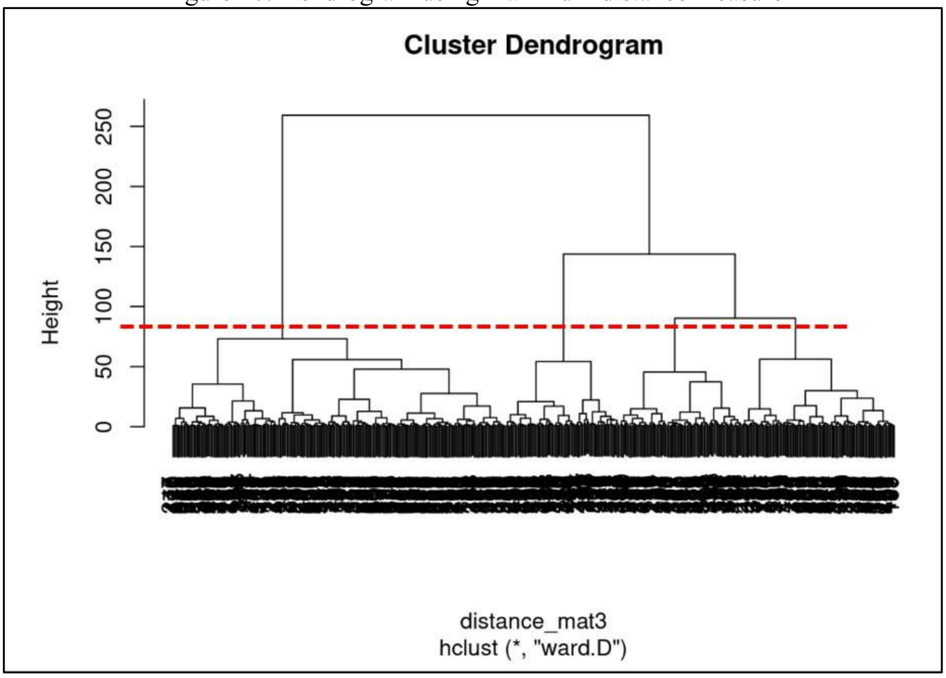


Figure 11: Dendrogram using manhattan distance measure

The plot of the clusters vs gap statistic is another way to determine the optimal number of clusters. The gap statistic is high at k=3 and k=4. Figure 12 plot uses hcut function (Refer Appendix 9.4 for more results).

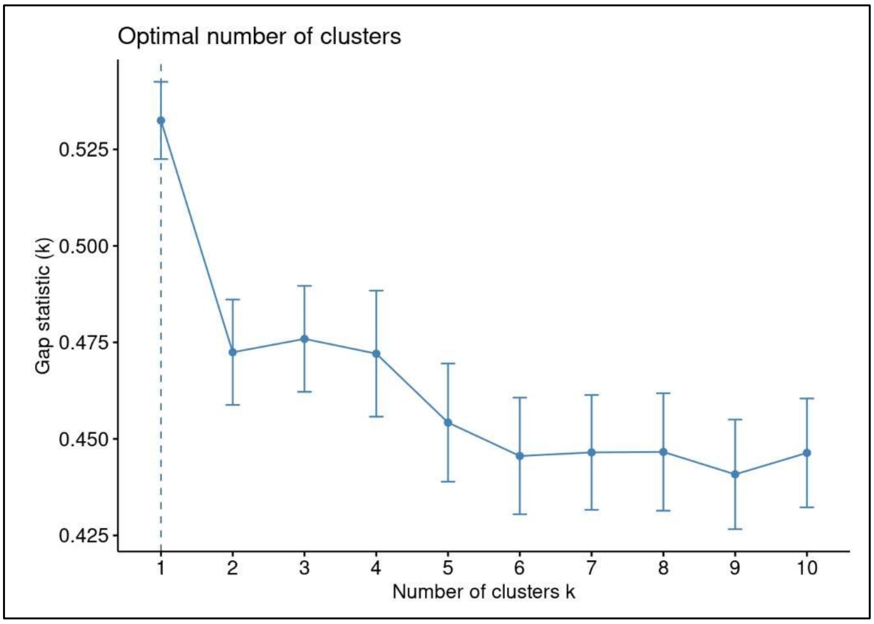


Figure 12: Plot of clusters vs gap statistic using hcut function

4.3 **Clustering**

- Hierarchical clustering

On cutting our dendrograms by 3 clusters, the results show a large difference in cluster sizes which is not ideal. Hence, I do not consider this method when forming our final clusters.

```
# Cutting tree by no. of clusters
fit_3_max <- cutree(Hierar_cl_max, k = 3 )

# Find number of observations in each cluster
table(fit_3_max)

## fit_3_max
##   1   2   3
## 175 200 100
```

```
# Cutting tree by no. of clusters
fit_3_man <- cutree(Hierar_cl_man, k = 3 )

# Find number of observations in each cluster
table(fit_3_man)

## fit_3_man
##   1   2   3
## 181 219  75
```

```
# Cutting tree by no. of clusters
fit_3_eu <- cutree(Hierar_cl_eu, k = 3 )

# Find number of observations in each cluster
table(fit_3_eu)

## fit_3_eu
##   1   2   3
##  99 194 182
```

Figure 13: Observation distribution among clusters for different distance measures

- K-means clustering

Through the implementation of k-means clustering, I observed a more balanced distribution in the cluster sizes. The figure below depicts the outcome of k-means clustering on a random sample of 500 observations. These results will be used to evaluate and infer recommendations. Furthermore, K-means clustering with 4 clusters wasn't considered because its respective cluster plot shoId major overlapping, which is not ideal. (Refer to Appendix 9.5 for cluster plots)

```
# Kmeans clustering

set.seed(99)
k_cl3 <- kmeans(fscores,3,nstart=25)
k_cl3

K-means clustering with 3 clusters of sizes 143, 131, 201

Cluster means:
          TC1         TC2         TC4        TC3
1 -0.08338865 -0.72972613  0.77409412  0.5783135
2  1.02158691  0.94978049  0.07928446  0.3662194
3 -0.60648412 -0.09985277 -0.60239663 -0.6501173
```

Figure 14: observation distribution among clusters and cluster mean scores using k-means

5 Validation

Two methods Ire implemented to internally validate our clusters:

- cl_predict() – predicting the cluster assignment of our 100 observations subset using the existing clusters resulted in an accuracy score of 89%

```
df_kmean_valid_3 <- data.frame(k_cl3["cluster"])
df_kmean_valid_3 <- df_kmean_valid_3[validate_indices, ]

K_validate_3 <- cl_predict(k_cl3,newdata = validate_fscores)

df_kfit_validate_3 <- data.frame(Difference= df_kmean_valid_3 - K_validate_3 )


# Count the number of 0s in 'Column1'_zero
num_zeros_k_3 <- nrow(filter(df_kfit_validate_3,Difference == 0))

# Calculate the total number of rows in the dataframe
total_rows_k_3 <- nrow(df_kfit_validate_3)

# Calculate the proportion of zeros
proportion_zeros_k_3 <- num_zeros_k_3 / total_rows_k_3

# Print the proportion of the correctly assign cluster
print(proportion_zeros_k_3)

## [1] 0.89
```

Figure 15: Accuracy score using cl_predict()

- Reclustering – through this method, I create a new set of clusters using the 100 observations subset. On comparing the cluster assignment in both models, I see an accuracy of 79%.

```r
New_K_CLust <- df_kmean_valid_3 - data.frame(validate_k_cl3["cluster"])
New_K_CLust

# Count the number of 0s in 'Column1'_zero
num_zeros_k_3_re <- nrow(filter(New_K_CLust ,cluster == 0))

# Calculate the total number of rows in the dataframe
total_rows_k_3_re <- nrow(New_K_CLust)

# Calculate the proportion of zeros
proportion_zeros_k_3_re <- num_zeros_k_3_re/total_rows_k_3_re

# Print the proportion of the correctly assign cluster
print(proportion_zeros_k_3_re)

## [1] 0.79
```

Figure 16: Accuracy score using reclustering

The result of the internal validation of the cluster analysis, based on the set of 100 random observations, reveals similarities in the cluster assignment as that of the 500 observation sample (See Figure 17 & 18). This confirms the consistency and reliability of the analysis.

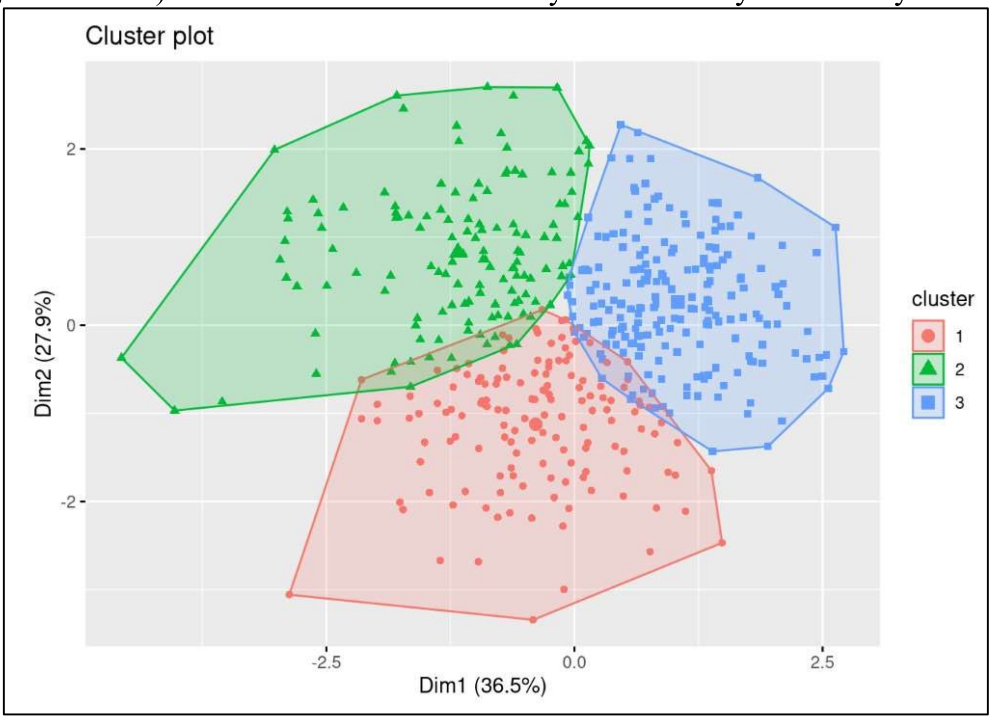


Figure 17: Scatter plot of 500 sample observations clustered into 3 groups

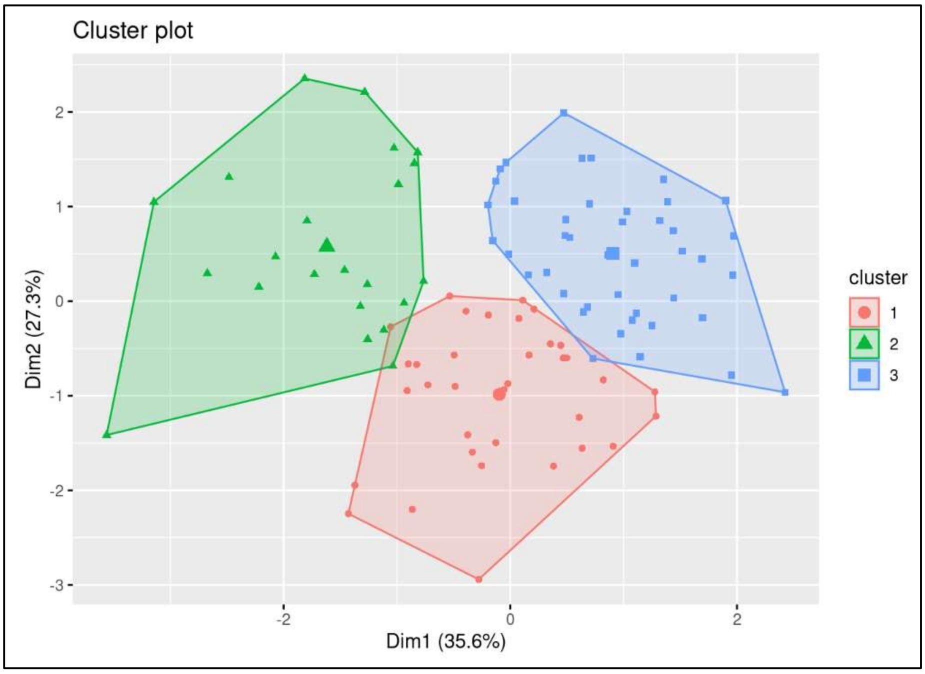


Figure 18: Scatter plot of 100 internal sample observations clustered into 3 groups

6 Evaluation

From the cluster means shown in Figure 14 and the attribute grouping in Figure 8 with its subsequent interpretation, I infer the characteristics of the 3 clusters (Refer to Table 3).

Cluster	Inferred characteristics	Borrower segment (BS)
1	Cluster 1 scores highly in TC4 and TC3, suggesting high net worth and competent financial management skills, which tend to depend less on debt. The negative TC2 score indicates high borroIr credibility (due to the format of variable encoding)	High Net Worth Individuals, Entrepreneurs, Selfemployed
2	Cluster 2 scores highly in TC1 and TC2, which infers that though they secure larger loans, they are deemed riskier owing to higher interest rates and negative loan statuses. Additionally, they have a significantly less net worth (low TC4 score). These characteristics provide a strong indication towards the middle-class income bracket.	Middle Income Individuals, Working class families and individuals
3	Cluster 3 scores negatively in 3 factors, i.e. TC1, TC4 and TC3. Overall, these individuals do not have significant net worth, hold subpar financial management skills/history, and tend to take out smaller loans. The characteristics demonstrated are that of a young adult or newly immigrated individuals/refugees.	Initial-phase earners, Fresh graduates, young adults, Foreigners/Refugees

Table 3: Cluster evaluation and borroIr segmentation

7 **Recommendation**

Based on the above evaluation, I provide the following structured recommendations for each of the cluster groups.

- BS 1

BorroIrs from this segment are a safe bet hence our focus should be on customer satisfaction. I aim to do this by providing high-value perks to borroIrs, like exclusive investment opportunities, sponsored excursions and complementary financial advisory services to incentivize higher loans. A dedicated relationship manager, along with priority banking, can be offered. Loans in business expansion, investments and luxury real estate should be marketed within this segment.

- BS 2

In addition to taking out more loans, I expect this segment to conduct more business with us. I aim to capitalize on this premise by providing everyday perks like membership, exclusive offers, and cashback. To correctly assess their financial profiles and avoid bad loans, the bank should introduce a 2-stage verification (internal and external) process to vet borroIrs. To improve customer satisfaction within this segment, I recommend introducing flexibility in the loan repayment structure. Car loans, home loans and personal loans should be marketed within this segment.

- BS 3

For borroIrs within this segment, I must take on stricter risk assessment measures, approve only verified profiles and essentially introduce policies to avoid defaulters. Additionally, I can start support programs to help these borroIrs enhance their financial profiles. Education loans and personal loans can be marketed for this segment.

In addition to these segment specific recommendations, I can also introduce a tier system for borroIrs where customers can take steps to move up the ladder to avail better benefits each time. This will help to provide more personalized loan products, initiate targeted marketing strategies and tier-specific customer support.

8 **Conclusion**

Undertaking this project, I Ire able to understand the data provided to us and identify attributes of the data that would most benefit our analysis. Further implementation of PCA and FA was to ensure multicollinearity and cross-loading Iren't present amongst variables. Cluster analysis was carried out on a sample of 500 observations from our final data set, where I achieved 3 optimal cluster groups. This was internally validated with a 100 observations sample yielding 79% accuracy. These clusters Ire evaluated based on their characteristics and assigned borroIr segments. I have given our recommendations to cater to each of these segments to improve our loan portfolio management.

9 **Appendices**

9.1 Mahalanobi distance results identifying outliers

```r
# Mahalanobi distance

Maha <- mahalanobis(norm_samp_df_loan ,colMeans(norm_samp_df_loan),cov(norm_samp_df_loan))
Maha_1 <- mahalanobis(norm_samp_df_loan_1 ,colMeans(norm_samp_df_loan_1),cov(norm_samp_df_loan_1))
```

```r
# The p value for each Mahalanobis distance

MahaPvalue <-pchisq(Maha,df=15,lower.tail = FALSE)
MahaPvalue_1 <-pchisq(Maha_1,df=15,lower.tail = FALSE)
```

```r
# Identify potential outlier (a p-value that is less than 0.001)

print(sum(MahaPvalue<0.001))
```

```
## [1] 25
```

```r
print(sum(MahaPvalue_1<0.001))
```

```
## [1] 24
```

9.2 Factor Analysis

9.2.1 PC extraction with Orthogonal rotation 3 PC

```r
pcModel3q<-principal(filtered_norm_samp_df_loan, 3, rotate="quartimax")
print.psych(pcModel3q, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = filtered_norm_samp_df_loan, nfactors = 3, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   RC1   RC2   RC3   h2   u2 com
## total_pymnt           14  0.92              0.86 0.14 1.0
## loan_amnt              1  0.92              0.87 0.13 1.1
## installment            3  0.91              0.86 0.14 1.1
## annual_inc             6  0.62 -0.30        0.53 0.47 1.7
## revol_bal             11  0.53        0.51 0.54 0.46 2.0
## tot_cur_bal           15  0.44        0.34 0.36 0.64 2.5
## verification_status    7 -0.41 -0.34        0.31 0.69 2.3
## home_ownership         5  0.30              0.17 0.83 2.4
## int_rate               2        0.89        0.83 0.17 1.1
## sub_grade              4        0.88        0.82 0.18 1.1
## revol_util            12        0.50        0.28 0.72 1.3
## loan_status            8        0.31        0.11 0.89 1.3
## open_acc              10              0.80 0.66 0.34 1.0
## total_acc             13              0.78 0.64 0.36 1.1
## dti                    9        0.39  0.57 0.49 0.51 1.8
## total_credit_rv       16  0.41 -0.39  0.52 0.59 0.41 2.8
##
##                        RC1  RC2  RC3
## SS loadings           3.97 2.58 2.37
## Proportion Var        0.25 0.16 0.15
## Cumulative Var        0.25 0.41 0.56
## Proportion Explained  0.45 0.29 0.27
## Cumulative Proportion 0.45 0.73 1.00
```

9.2.2 PC extraction with Orthogonal rotation 4 PC

```r
pcModel4q<-principal(filtered_norm_samp_df_loan, 4, rotate="quartimax")
print.psych(pcModel4q, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = filtered_norm_samp_df_loan, nfactors = 4, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   RC1   RC2   RC3   RC4   h2    u2 com
## total_pymnt           14  0.94                    0.90 0.099 1.0
## installment            3  0.93                    0.90 0.104 1.1
## loan_amnt              1  0.93                    0.90 0.099 1.1
## revol_bal             11  0.53        0.50        0.56 0.436 2.2
## verification_status    7 -0.45                    0.33 0.666 2.2
## int_rate               2        0.90              0.85 0.155 1.1
## sub_grade              4        0.88              0.83 0.167 1.1
## revol_util            12        0.58              0.41 0.586 1.4
## loan_status            8        0.34              0.13 0.866 1.3
## open_acc              10              0.79        0.67 0.335 1.1
## total_acc             13              0.73  0.33  0.65 0.346 1.5
## dti                    9              0.65        0.56 0.436 1.7
## total_credit_rv       16  0.47 -0.46  0.53        0.72 0.283 2.9
## tot_cur_bal           15                    0.82  0.74 0.256 1.2
## home_ownership         5                    0.76  0.59 0.405 1.0
## annual_inc             6  0.47              0.60  0.63 0.369 2.2
##
##                        RC1  RC2  RC3  RC4
## SS loadings           3.75 2.53 2.20 1.92
## Proportion Var        0.23 0.16 0.14 0.12
## Cumulative Var        0.23 0.39 0.53 0.65
## Proportion Explained  0.36 0.24 0.21 0.19
## Cumulative Proportion 0.36 0.60 0.81 1.00
```

9.2.3 PC extraction with Oblique rotation 3 PC

```r
pcModel3o<-principal(filtered_norm_samp_df_loan, 3, rotate="oblimin")
print.psych(pcModel3o, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = filtered_norm_samp_df_loan, nfactors = 3, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   TC1   TC2   TC3   h2   u2 com
## total_pymnt           14  0.92              0.86 0.14 1.0
## loan_amnt              1  0.91              0.87 0.13 1.0
## installment            3  0.91              0.86 0.14 1.0
## annual_inc             6  0.62 -0.34        0.53 0.47 1.7
## tot_cur_bal           15  0.40        0.30  0.36 0.64 2.6
## verification_status    7 -0.35 -0.33        0.31 0.69 2.4
## home_ownership         5                    0.17 0.83 2.5
## int_rate               2        0.89        0.83 0.17 1.0
## sub_grade              4        0.87        0.82 0.18 1.1
## revol_util            12        0.50        0.28 0.72 1.2
## loan_status            8        0.32        0.11 0.89 1.5
## open_acc              10              0.82  0.66 0.34 1.0
## total_acc             13              0.79  0.64 0.36 1.0
## dti                    9        0.43  0.63  0.49 0.51 2.0
## revol_bal             11  0.44        0.48  0.54 0.46 2.0
## total_credit_rv       16  0.35 -0.39  0.48  0.59 0.41 2.8
##
##                        TC1  TC2  TC3
## SS loadings           3.84 2.60 2.49
## Proportion Var        0.24 0.16 0.16
## Cumulative Var        0.24 0.40 0.56
## Proportion Explained  0.43 0.29 0.28
## Cumulative Proportion 0.43 0.72 1.00
```

9.2.4 PC extraction with Oblique rotation 4 PC

```
pcModel4o<-principal(filtered_norm_samp_df_loan, 4, rotate="oblimin")
print.psych(pcModel4o, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = filtered_norm_samp_df_loan, nfactors = 4, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   TC1   TC2   TC3   TC4   h2    u2 com
## total_pymnt           14  0.95                    0.90 0.099 1.0
## installment            3  0.94                    0.90 0.104 1.0
## loan_amnt              1  0.93                    0.90 0.099 1.0
## verification_status    7 -0.41                    0.33 0.666 2.4
## int_rate               2        0.89              0.85 0.155 1.0
## sub_grade              4        0.88              0.83 0.167 1.1
## revol_util            12        0.61              0.41 0.586 1.4
## total_credit_rv       16  0.43 -0.50  0.49        0.72 0.283 3.0
## loan_status            8        0.35              0.13 0.866 1.6
## open_acc              10              0.80        0.67 0.335 1.0
## total_acc             13              0.72  0.31 0.65 0.346 1.4
## dti                    9              0.71        0.56 0.436 1.7
## revol_bal             11  0.44        0.46        0.56 0.436 2.2
## tot_cur_bal           15                    0.83 0.74 0.256 1.0
## home_ownership         5                    0.79 0.59 0.405 1.0
## annual_inc             6  0.38              0.59 0.63 0.369 1.9
##
##                         TC1  TC2  TC3  TC4
## SS loadings            3.56 2.53 2.27 2.03
## Proportion Var         0.22 0.16 0.14 0.13
## Cumulative Var         0.22 0.38 0.52 0.65
## Proportion Explained   0.34 0.24 0.22 0.20
## Cumulative Proportion  0.34 0.59 0.80 1.00
```

9.2.5 PC extraction with Orthogonal rotation 4 PC After removing total_credit_rv and revol_bal

```
pcModel4q1<-principal(fa_loan, 4, rotate="quartimax")
print.psych(pcModel4q1, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = fa_loan, nfactors = 4, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   RC1   RC2   RC4   RC3   h2    u2 com
## total_pymnt           13  0.96                    0.94 0.064 1.0
## installment            3  0.95                    0.93 0.075 1.1
## loan_amnt              1  0.94                    0.92 0.078 1.1
## verification_status    7 -0.47                    0.36 0.639 2.2
## int_rate               2        0.92              0.88 0.120 1.1
## sub_grade              4        0.91              0.87 0.130 1.1
## revol_util            11        0.60              0.42 0.583 1.3
## loan_status            8        0.37              0.16 0.837 1.4
## tot_cur_bal           14              0.82        0.74 0.256 1.2
## home_ownership         5              0.74        0.57 0.433 1.1
## annual_inc             6  0.43        0.65        0.63 0.370 1.9
## open_acc              10                    0.81 0.70 0.301 1.2
## total_acc             12              0.33  0.78 0.74 0.259 1.4
## dti                    9                    0.68 0.59 0.407 1.6
##
##                         RC1  RC2  RC4  RC3
## SS loadings            3.32 2.36 1.94 1.83
## Proportion Var         0.24 0.17 0.14 0.13
## Cumulative Var         0.24 0.41 0.54 0.67
## Proportion Explained   0.35 0.25 0.21 0.19
## Cumulative Proportion  0.35 0.60 0.81 1.00
```

9.2.6

PC extraction with Oblique rotation 4 PC
After removing total_credit_rv and revol_bal

```
pcModel4o1<-principal(fa_loan, 4, rotate="oblimin")
print.psych(pcModel4o1, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = fa_loan, nfactors = 4, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   TC1   TC2   TC4   TC3   h2    u2 com
## total_pymnt           13  0.98                    0.94 0.064 1.0
## installment            3  0.95                    0.93 0.075 1.0
## loan_amnt              1  0.94                    0.92 0.078 1.0
## verification_status    7 -0.43                    0.36 0.639 2.2
## int_rate               2        0.92              0.88 0.120 1.0
## sub_grade              4        0.91              0.87 0.130 1.0
## revol_util            11        0.64              0.42 0.583 1.5
## loan_status            8        0.40              0.16 0.837 1.7
## tot_cur_bal           14              0.83        0.74 0.256 1.0
## home_ownership         5              0.77        0.57 0.433 1.1
## annual_inc             6  0.35        0.63        0.63 0.370 1.7
## open_acc              10                    0.81 0.70 0.301 1.1
## total_acc             12                    0.77 0.74 0.259 1.2
## dti                    9             -0.33  0.71 0.59 0.407 1.6
##
##                        TC1  TC2  TC4  TC3
## SS loadings           3.19 2.38 2.00 1.87
## Proportion Var        0.23 0.17 0.14 0.13
## Cumulative Var        0.23 0.40 0.54 0.67
## Proportion Explained  0.34 0.25 0.21 0.20
## Cumulative Proportion 0.34 0.59 0.80 1.00
```

9.2.7 Validation of Factor Analysis PC extraction with Orthogonal rotation 4 PC

```
pcModel4q1<-principal(fa_loan, 4, rotate="quartimax")
print.psych(pcModel4q1, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = fa_loan, nfactors = 4, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   RC1   RC2   RC4   RC3   h2    u2 com
## total_pymnt           13  0.96                    0.94 0.064 1.0
## installment            3  0.95                    0.93 0.075 1.1
## loan_amnt              1  0.94                    0.92 0.078 1.1
## verification_status    7 -0.47                    0.36 0.639 2.2
## int_rate               2        0.92              0.88 0.120 1.1
## sub_grade              4        0.91              0.87 0.130 1.1
## revol_util            11        0.60              0.42 0.583 1.3
## loan_status            8        0.37              0.16 0.837 1.4
## tot_cur_bal           14              0.82        0.74 0.256 1.2
## home_ownership         5              0.74        0.57 0.433 1.1
## annual_inc             6  0.43        0.65        0.63 0.370 1.9
## open_acc              10                    0.81 0.70 0.301 1.2
## total_acc             12              0.33  0.78 0.74 0.259 1.4
## dti                    9                    0.68 0.59 0.407 1.6
##
##                        RC1  RC2  RC4  RC3
## SS loadings           3.32 2.36 1.94 1.83
## Proportion Var        0.24 0.17 0.14 0.13
## Cumulative Var        0.24 0.41 0.54 0.67
## Proportion Explained  0.35 0.25 0.21 0.19
## Cumulative Proportion 0.35 0.60 0.81 1.00
```

9.2.8 PC extraction with Oblique rotation 4 PC

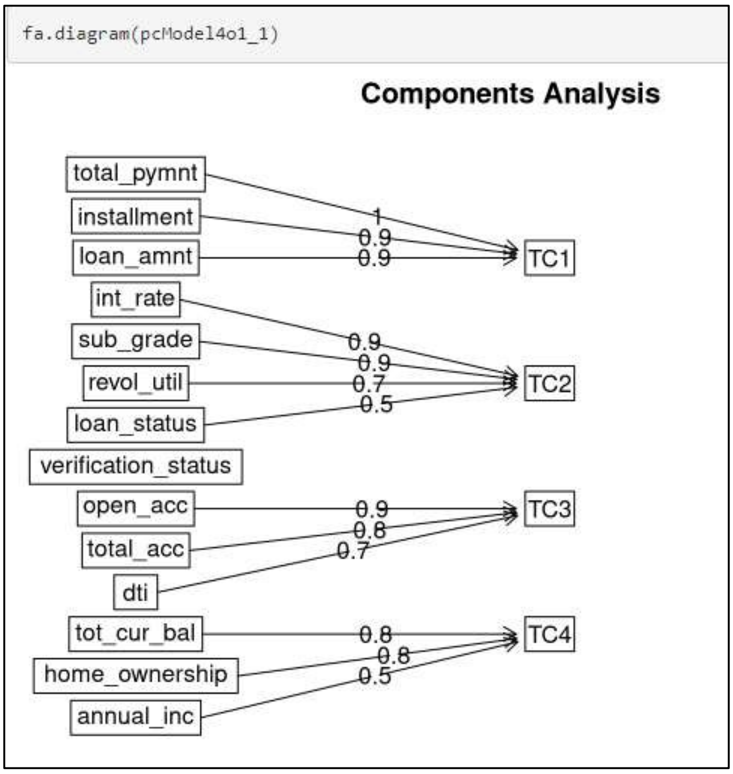
```
pcModel4o1_1<-principal(fa_loan_1, 4, rotate="oblimin")
print.psych(pcModel4o1_1, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = fa_loan_1, nfactors = 4, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item  TC1   TC2   TC3   TC4   h2    u2 com
## total_pymnt           13  0.98                  0.94 0.056 1.0
## installment            3  0.94                  0.92 0.084 1.0
## loan_amnt              1  0.93                  0.93 0.067 1.0
## int_rate               2        0.90            0.86 0.137 1.0
## sub_grade              4        0.89            0.86 0.143 1.1
## revol_util            11        0.68            0.44 0.557 1.3
## loan_status            8        0.48            0.25 0.748 1.9
## verification_status    7                        0.24 0.760 3.1
## open_acc              10              0.86      0.76 0.245 1.0
## total_acc             12              0.79      0.74 0.263 1.1
## dti                    9              0.68      0.55 0.453 1.6
## tot_cur_bal           14                   0.85 0.79 0.211 1.0
## home_ownership         5                   0.81 0.62 0.383 1.1
## annual_inc             6  0.41             0.53 0.63 0.367 2.5
##
##                        TC1  TC2  TC3  TC4
## SS loadings           3.13 2.57 1.92 1.90
## Proportion Var        0.22 0.18 0.14 0.14
## Cumulative Var        0.22 0.41 0.54 0.68
## Proportion Explained  0.33 0.27 0.20 0.20
## Cumulative Proportion 0.33 0.60 0.80 1.00
```

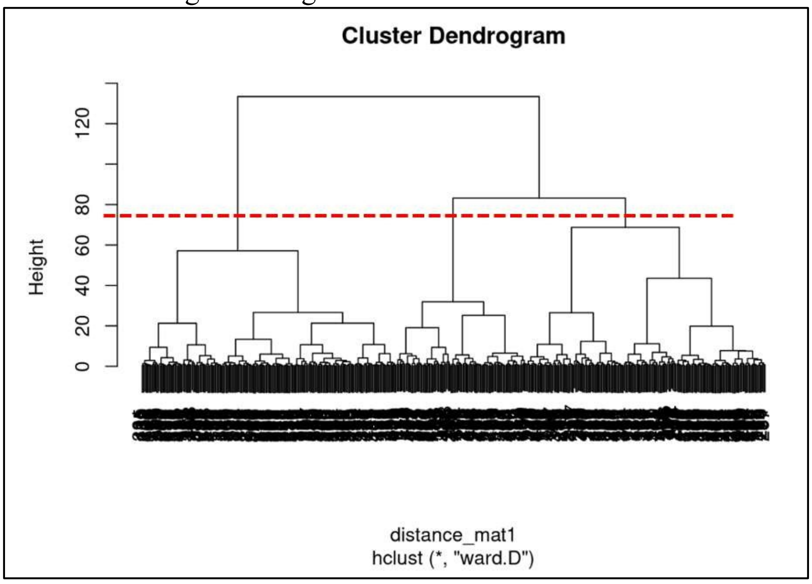
9.2.9 Factor Analysis Validation


```
pcModel4o1_1<-principal(fa_loan_1, 4, rotate="oblimin")
print.psych(pcModel4o1_1, cut=0.3, sort=TRUE)
```

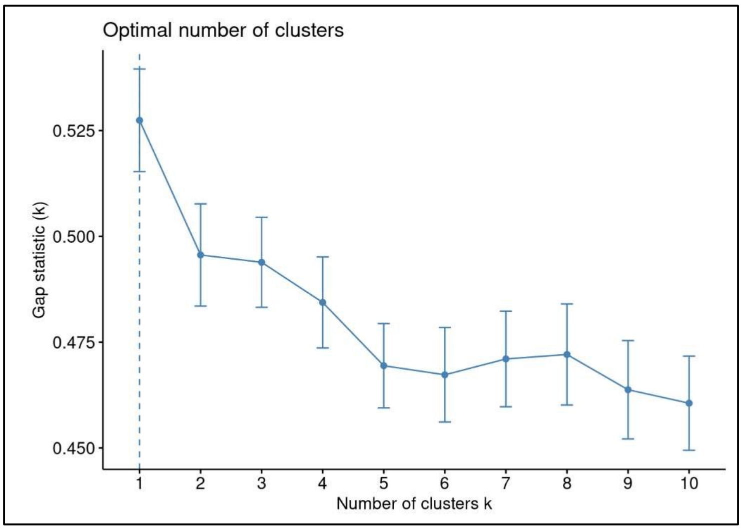
```
## Principal Components Analysis
## Call: principal(r = fa_loan_1, nfactors = 4, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item  TC1   TC2   TC3   TC4   h2    u2 com
## total_pymnt           13  0.98                  0.94 0.056 1.0
## installment            3  0.94                  0.92 0.084 1.0
## loan_amnt              1  0.93                  0.93 0.067 1.0
## int_rate               2        0.90            0.86 0.137 1.0
## sub_grade              4        0.89            0.86 0.143 1.1
## revol_util            11        0.68            0.44 0.557 1.3
## loan_status            8        0.48            0.25 0.748 1.9
## verification_status    7                        0.24 0.760 3.1
## open_acc              10              0.86      0.76 0.245 1.0
## total_acc             12              0.79      0.74 0.263 1.1
## dti                    9              0.68      0.55 0.453 1.6
## tot_cur_bal           14                   0.85 0.79 0.211 1.0
## home_ownership         5                   0.81 0.62 0.383 1.1
## annual_inc             6  0.41             0.53 0.63 0.367 2.5
##
##                        TC1  TC2  TC3  TC4
## SS loadings           3.13 2.57 1.92 1.90
## Proportion Var        0.22 0.18 0.14 0.14
## Cumulative Var        0.22 0.41 0.54 0.68
## Proportion Explained  0.33 0.27 0.20 0.20
## Cumulative Proportion 0.33 0.60 0.80 1.00
```



9.3 Dendrogram using euclidean distance measure



9.4 Plot of clusters vs gap statistic using k-means function



9.5 K - means clustering

9.5.1 4 Cluster analysis

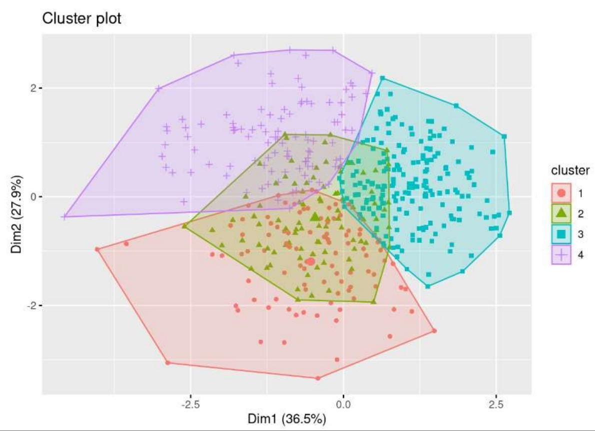
```
set.seed(99)
k_cl4 <- kmeans(fscores,4,nstart=25)
k_cl4
```

```
## K-means clustering with 4 clusters of sizes 97, 89, 187, 102
##
## Cluster means:
##          TC1        TC2         TC4         TC3
## 1  0.2805907 -0.6927721  1.25069777 -0.07478757
## 2 -0.2213021 -0.1918704 -0.04109324  1.34999015
## 3 -0.6377144 -0.1593500 -0.57200386 -0.72379701
## 4  1.0954038  1.1183707 -0.10485945  0.22015011
```

Plot of 4 clusters

Plot K-mean cluster

```
#Visualize clustering results for K = 4
plot_kcluster <- fviz_cluster(k_cl4, data = fscores, geom = "point")
plot_kcluster
```



9.5.2 Internal Validations

3 Cluster Validation

```
set.seed(85)
validate_k_cl3 <- kmeans(validate_fscores,3,nstart=25)
validate_k_cl3
```

```
## K-means clustering with 3 clusters of sizes 35, 22, 43
##
## Cluster means:
##           TC1         TC2        TC4          TC3
## 1 -0.04471555 -0.52270337  0.7659034 -0.457833359
## 2  1.20746783  0.86139451  0.4896403  0.747108361
## 3 -0.58137786 -0.01525724 -0.8739234 -0.009586427
```

4 Cluster Validation

```
set.seed(85)
validate_k_cl4 <- kmeans(validate_fscores,4,nstart=25)
validate_k_cl4
```

```
## K-means clustering with 4 clusters of sizes 32, 13, 15, 40
##
## Cluster means:
##           TC1         TC2        TC4        TC3
## 1 -0.02172999 -0.58066903  0.7569521 -0.5515239
## 2  0.06140022 -0.02665632  0.2288147  1.6252165
## 3  1.57874746  1.22011197  0.5730244  0.3312012
## 4 -0.59460138  0.01565654 -0.8948106 -0.2111767
```

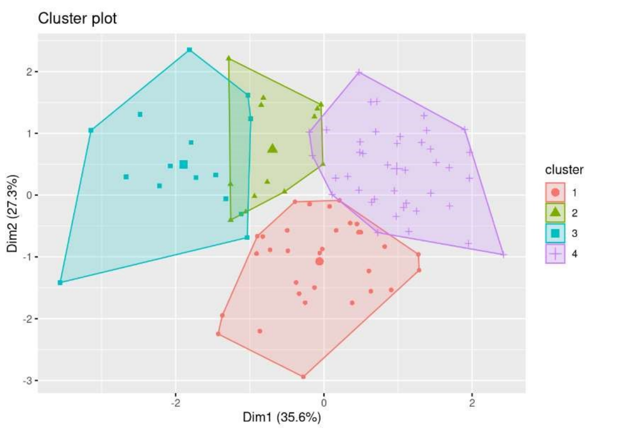
Accuracy Percentages

```
# Count the number of 0s in 'Column1'_zero
num_zeros_k_4_re <- nrow(filter(New_K_CLust ,cluster == 0))

# Calculate the total number of rows in the dataframe
total_rows_k_4_re <- nrow(New_K_CLust)

# Calculate the proportion of zeros
proportion_zeros_k_4_re <- num_zeros_k_4_re/total_rows_k_4_re

# Print the proportion of the correctly assign cluster
print(proportion_zeros_k_4_re)
```

```
## [1] 0.34
```

Cluster Plot

```
# Visualize clustering results for K = 4 - validate set
plot_kcluster_validate <- fviz_cluster(validate_k_cl4 , data = validate_fscores,geom="point")
plot_kcluster_validate
```



9.6 Meeting minutes

2 groups Ire created where 1 was assigned coding related tasks and other was assigned reporting related tasks.

Coding – 5562860, 5503555, 2216142, 5548256

Reporting – 5588137, 5584180

Meeting date	Action items	Contribution
19/02	Discuss the assignment question and queries. Initial elimination of attributes from the dataset. Allocating tasks to be carried out and project schedule.	Everyone
24/02	Import the dataset Check NA values Remove agreed upon variables Label encoding and factorize Visualize to detect the outliers Sampling Check multicollinearity and standardised the data Perform PCA	Coding team
	Start reporting on intro and data prep	Reporting team
29/02	Discuss PCA results	Everyone
	Perform Factor analysis	Coding team
	Report PCA results	Reporting team
7/03	Discuss FA results	Everyone
	Report FA results	Reporting team
10/03	Re-perform all steps carried out to verify results are consistent	Coding team
12/03	Cluster analysis	Coding team
14/03	Compare results and choose best solution Validate and profile cluster solution	Everyone

16/03	Reporting cluster analysis results, validation, recommendation and conclusion	Reporting team
17/03	Overall feedback and corrections	Everyone