ACKNOWLEDGEMENT

Above all, I wish to extend my heartfelt appreciation to the University of Economics and Law, particularly the Faculty of Information Systems, for granting me the chance to pursue my academic endeavors and carry out this research.

I am profoundly grateful to my supervisor, Le Hoanh Su, for his invaluable mentorship, support, and motivation during this research endeavor. His insightful feedback and recommendations have significantly influenced the trajectory of this study and enhanced its overall caliber. Without his steadfast guidance, I would not have been able to complete my graduate thesis.

COMMITMENT

I ensure that I have autonomously undertaken and completed the project titled "CLASSIFYING CREDIT SCORES WITH K-MEANS: INSIGHTS FROM CHIT FUND DATA IN INDIA." All content in the report mirrors my committed research efforts, supplemented by properly cited sources, and the data utilized has been collected and processed with honesty. I accept full responsibility for this assurance before the Board.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

Abbreviation	Describe
Chit Fund	A financial scheme where members contribute money into a pool at regular intervals, with a predetermined payout to one member through a bidding process.
K-Means	A clustering algorithm used to partition data into groups based on similarities in features, often employed in data analysis and machine learning.
Bid	An offer made by a member in a chit fund auction to claim the accumulated pool of funds at a particular interval.
Members	Individuals or entities who participate in a chit fund by contributing funds regularly and may bid to receive payouts.
Auction	A process within a chit fund where members bid for the accumulated pool of funds, typically held at regular intervals.
Inst	Abbreviation for "Installment," referring to the periodic contribution made by members to the chit fund pool.

ABSTRACT

This research article explores the concept of Chit Funds in India, delving into the intricacies of this financial scheme. It focuses on analyzing transactional behaviors and member demographic information within a dataset to construct a classification model. This model aims to assess the creditworthiness of Chit Fund participants, providing insights into their financial behaviors and aiding in risk assessment and decision-making processes within the Chit Fund system.

CHAPTER 1. INTRODUCTION

1.1 Reasons for choosing the topic

This paper explores the potential of Chit Funds, a rotational savings and credit scheme, as a platform for credit score evaluation. By analyzing an individual's participation and behavior within a Chit Fund, we aim to develop a new method for assessing creditworthiness.

Traditional credit scoring methods rely heavily on factors like credit history and debt-to-income ratio. However, these methods may not accurately reflect the creditworthiness of individuals who lack a formal credit history or operate outside traditional financial systems.

Chit Funds offer a unique opportunity to evaluate creditworthiness based on real-world financial behavior. By examining an individual's contribution history, bidding patterns, and overall participation within the Chit Fund, we can potentially build a credit score that reflects their: Financial Discipline, Creditworthiness, Trustworthiness.

This research proposes a novel approach to credit scoring by leveraging the rich data available within Chit Funds. By analyzing play behavior, we hope to create a more inclusive and accurate credit assessment system, particularly for those who are underserved by traditional methods.

1.2 Subject goal

The overarching goal of this research is to develop a robust credit score classification model utilizing data extracted from Chit Funds. This model will leverage insights gleaned from individuals' participation behaviors within these financial collectives. By analyzing factors such as contribution history, bidding patterns, and overall adherence to Chit Fund rules, we aim to establish a novel and more inclusive credit scoring system. This system has the potential to broaden access to financial services, particularly for those who lack a formal credit history or operate outside traditional financial institutions.

By achieving these objectives, this research hopes to pave the way for a more comprehensive and inclusive credit scoring system that leverages alternative data sources. This will

ultimately facilitate greater financial inclusion and empower individuals who have previously been underserved by traditional credit assessment methods.

1.3 Object and scope

Data on Chit Fund is provided by Harvard Dataverse

1.4 Research Methods

The research methodology encompasses two main approaches: theoretical research and experimental research. The theoretical aspect involves studying foundational concepts such as the definition, theory, and methodologies for evaluating and classifying credit scores, alongside gaining insights into the workings of chit funds. Additionally, understanding prediction methods for unsupervised machine learning models is crucial in this phase.

On the other hand, the experimental research involves practical implementation. It begins with data collection, followed by preprocessing and comprehensive data exploration to uncover patterns and insights within the dataset. Subsequently, a credit score classification model is proposed and developed for each individual member of the chit fund, leveraging the insights gained from the theoretical research and the analysis of the dataset. This experimental approach allows for the application and validation of theoretical knowledge in real-world scenarios, enhancing the understanding and effectiveness of the credit evaluation process within chit funds.

1.5 The structure of the thesis

The summary report on the research topic is structured into distinct chapters, each serving a specific purpose. Chapter 1 provides a comprehensive introduction to the research paper, encompassing the reasons behind the topic selection, an overview of the research landscape, the scope and objectives of the study, the involved subjects, research methodologies employed, and the overall significance of the topic. In Chapter 2, the theoretical framework and related research are presented, offering foundational knowledge essential for understanding the topic under scrutiny.

Chapter 3 delves into the heart of the research by analyzing Chit Fund credit data and developing a credit score model classification. This section elaborates on the process of data

analysis, encompassing data preprocessing, attribute selection, attribute construction, and the methodology utilized for building a credit score classification model. Chapter 4 then presents the experimental results, where the performance and outcomes of the model developed in Chapter 3 are rigorously evaluated using relevant classification metrics.

Finally, Chapter 5 encapsulates the conclusions drawn from the research findings and outlines potential future directions for further exploration. It synthesizes the key insights gleaned from the experiments and suggests avenues for future research and development in the field. Together, these chapters provide a structured and comprehensive overview of the research journey, from its inception to its conclusions and potential future implications.

CHAPTER 2. THEORETICAL BASE

2.1 Overview Credit Score

Credit Score definition

In short, a credit score is a numerical representation (between 300 and 850) used to assess your creditworthiness. The higher the score, the better your chances of getting loans and favorable interest rates. It is determined by your credit history, which includes details like the number of accounts you have, your debt levels, and your repayment history. Lenders use credit scores to assess your likelihood of repaying loans on time[1].

Credit Scores mechanism

Having a good credit score is a major advantage when it comes to borrowing money. Lenders are much more likely to approve your loan application and offer you lower interest rates if your score is high. This translates to significant savings over time, making a good credit score a valuable tool for managing your finances. On the other hand, a low credit score can make it harder to get loans approved and can result in higher interest rates, potentially hindering your financial goals.

While specific requirements vary between lenders, a score of 700 or above is typically seen favorably and could mean a lower interest rate. Scores exceeding 800 are considered top-notch. The following provides a general breakdown of how credit scores are categorized: Excellent (800–850), Very Good (740–799), Good (670–739), Fair (580–669), Poor (300–579).

Credit Score calculation

Five key components determine your credit score (can be differences in the information collected):

- Payment history (weighted 35%) - This is the most important factor and reflects your track record of making timely payments.
- Amounts owed (weighted 30%) - This considers how much credit you're using compared to your total credit limit.
- Length of credit history (weighted 15%) - A longer credit history with responsible management can improve your score.

- Credit mix (weighted 10%) - Having different types of credit accounts, like installment loans and credit cards, can be beneficial.
- New credit inquiries (weighted 10%) - Applying for too much new credit in a short period can lower your score.

2.2 Overview of Chit Funds

Chit Funds mechanism

Chit funds function with a start date and a duration that matches the number of participants. Members contribute regular installments to a common pool. Each month, a unique auction is held where members compete with "reverse bids" to access a lump sum, the "Prize Money." The member offering to forfeit the greatest portion of their future contributions in exchange for the immediate payout wins. The remaining pot, minus a small management fee, is then distributed as a dividend among all members. This cycle repeats monthly, giving everyone the chance to access a larger sum upfront. In essence, chit funds allow you to borrow against your future savings[2].

Key roles in a Chit fund

- The winning bidder**:** The member who wins the auction with the lowest bid receives the lump sum "Prize Money," essentially taking a short-term loan from the pool.
- The members**:** All members, including the borrower, contribute monthly installments. They receive a portion of the pot as a "dividend" each month, effectively reducing their following month's contribution. This functions as a savings plan.
- The organizer**:** This individual manages the chit fund, overseeing auctions and distributions. They earn a commission (usually 5%) for their service.

2.3 Elbow

The elbow method is a simple yet helpful tool to choose the ideal number of clusters (K) in K-means clustering. It works by visualizing the trade-off between increasing the number of clusters and the improvement in explaining the data's variance.[3]

The elbow method gets its name from the desired shape of the resulting graph. Ideally, the graph should have a sharp bend (elbow) where the WCSS starts to decrease slowly with each

additional cluster. This "elbow" suggests that adding more clusters beyond this point provides diminishing returns in terms of explaining the data's structure. Therefore, the number of clusters corresponding to the elbow is considered the optimal K for your data.

In simpler terms, the elbow method helps you find the "sweet spot" between creating too few or too many clusters in your K-means analysis.

2.4 Silhouette

Silhouette Score Concepts

The silhouette score helps assess how well clustering algorithms (like K-Means) group similar data points together. It's calculated for each data point within a cluster.
Here's the gist:

- We consider two distances for each data point:
    - Average distance to other points in its own cluster (a)
    - Average distance to the closest points in the next nearest cluster (b)
- A good silhouette score indicates a data point is well-placed within its cluster (low **a**) and far from points in other clusters (high **b**).
- The formula calculates the silhouette score (S) using these distances: [latex]S = \frac {(b – a)} {max (a, b)} [/latex]

In simpler terms, the silhouette score reflects how well-separated clusters are and how well individual points fit within their assigned clusters.

Python-explained Silhouette Score

Scikit-learn offers functions to evaluate clustering with silhouette scores:

- silhouette_score: Calculates average score (how well your clustering separated the data)
- silhouette_samples: Scores each data point (how well specific points fit within their assigned clusters)

We'll use these to:

- Calculate K-Means Silhouette Score: See how well data is clustered for a specific number of clusters (K).

- Find Best K: Compare silhouette scores for different K values to identify the optimal number of clusters for your data.

2.5 K-Means

K-Means clustering is a popular unsupervised learning algorithm and might be the first you encounter in this field. It works by grouping data points into clusters in a way that minimizes the distance between points within each cluster. Think of it as sorting objects based on how similar they are to each other. K-Means achieves this by strategically placing "centroids" (imaginary cluster centers) and assigning data points to the closest centroid. This process continues until the overall distance within each cluster is minimized. The simplicity and effectiveness of K-Means make it a widely used technique for data exploration and analysis.

Strength and drawbacks of K-Means

It is simple to implement, high-performance, easy to interpret, and suitable for big sets of data. On the other hand, K-Means is sensitive to scale, difficult to incorporate categorical variables, sensitive to outliers.

K-Means Variation

- K-Medians: Uses medians (middle values) instead of means for centroids (useful for outliers).
- K-Medoids: Uses real data points as centers (medoids).
- Fuzzy C-Means: Allows points to belong to multiple clusters (fuzzy boundaries).
- K-Means++: Smart initialization for K-Means (often default in scikit-learn).

CHAPTER 3. ANALYZING CHIT FUND CREDIT DATA AND DEVELOP A CREDIT SCORE MODEL CLASSIFICATION.

This chapter focuses on the process of collecting, analyzing, and processing data to develop a credit score evaluation model. It begins with data collection and description to comprehend the intricacies of the problem at hand. Subsequently, essential variables are selected for analysis to extract valuable insights. Finally, the chapter delves into data preprocessing, during which the requisite columns are chosen for inclusion in the data evaluation classification model.

3.1 Define Analysis Objectives

The objective of this study is to analyze members of chit funds. By examining their demographic data, financial history, and transaction records within the chit fund, I aim to extract insights and categorize their credit scores into distinct groups.

3.2 Data Collection and Description

The dataset collected is Primary and Secondary data from chit-fund companies for the Credit Scoring project in India (2012-12-01), The data set includes 5 tables, but because of the analysis problem, I only took 3 tables including cf1_delhi_collateral, cf1_delhi_surety, cf1_delhi_transaction_data. Table cf1_delhi_collateral contains information indicating members' collateral assets, each member can have more than 1 collateral asset. Table cf1_delhi_surety shows detailed information about each member's guarantor, including information such as age, gender, occupation, land assets, house assets, etc. Table cf1_delhi_transaction_data contains information about transactions of each member in the chit fund, including information about transaction date, chit value, payment method, payment nature (early, late, partial), and participation auction. The following table is a detailed description of each column in each table

Variable	Data type	Description
Table: cf1_delhi_collateral		

chit_id	categorical	Identifier for individual members of each chit fund.
p_recno	categorical	The receipt number of the payment a member contributed to the chit fund.
collateral	categorical	The type of collateral(s) the members has (e.g., CER, CHT, etc.).
chit_value	real-valued multiplicative	The total value of a chit fund is obtained by (duration x monthly_contribution)
duration	count	The duration of operation for a chit fund, must be equal to the number of members involved.
monthly_contribution	real-valued multiplicative	The amount of money each member must contribute to the chit fund every month
year	categorical	The inception year of a chit fund.

Table: cf1_delhi_surety

chit	categorical	Identifier for each chit fund.
chit_id	*Referenced from table cf1_delhi_collateral*	
p_recno	*Referenced from table cf1_delhi_collateral*	
winning_aucn	real-valued multiplicative	The highest discounted price that the member to win the bid
duration	*Referenced from table cf1_delhi_collateral*	
fman	binary	If a foreman manages the chit fund, with '0' for no and '1' for yes.
surety_p_recno	categorical	Identifier for a guarantor associated with each chit fund member's contributions.
n_surety	count	The number of guarantors associated with each chit fund member's contributions.
age	count	Guarantor’s age.
salary	real-valued multiplicative	The amount of money the guarantors receives as salary on a monthly basis.

sex	binary	Guarantor's gender ("M": "male", "F": "female").
occupation	categorical	Guarantor's occupation or job title ("B": "Business"; "GS": "Graduate Student"; "HW": "Housework"; "P": "Professor"; "PS": "Public Service"; "R": "Researcher"; "SE": "Software Engineer".
other_chits	binary	If a guarantor associated with two or more chit in a same year ("Yes": "1"; "No": "0").
surety_others	binary	If a guarantor associated with two or more members in a same year ("Yes": "1"; "No": "0").
years_of_service	real-valued multiplicative	The number of years a Guarantor has been working or serving in a occupation.
house_owner	binary	If a guarantor owns the house ("Yes": "1"; "No": "0").
land_owner	binary	If a guarantor owns the land ("Yes": "1"; "No": "0").
income_tax	binary	If a guarantor is required to pay income tax or not. ("Yes": "1"; "No": "0").
insurance_policy	binary	If a guarantor has an insurance policy or not ("Yes": "1"; "No": "0").
insurance_amount	real-valued multiplicative	The amount of money that guarantor will receive in case of an insurance event such as accidents, illnesses, death, etc.
res_pin	categorical	Reserve Participant Identification Number of a members
off_pin	categorical	Official Participant Identification Number of a members
chit_value	*Referenced from table cf1_delhi_collateral*	

monthly_contribution	*Referenced from table cf1_delhi_collateral*	
year	*Referenced from table cf1_delhi_collateral*	
Table: cf1_delhi_transaction_data		
chit	*Referenced from table cf1_delhi_surety*	
chit_id	*Referenced from table cf1_delhi_collateral*	
p_recno	*Referenced from table cf1_delhi_collateral*	
winning_aucn	*Referenced from table cf1_delhi_surety*	
aucn_no	categorical	Auction Number is an identifier assigned to each monthly auction event within the chit fund.
aucn_date	categorical	Auction Date would represent the date the auction takes place within the chit fund.
inst_due	real-valued multiplicative	The installment that is currently due which members making regular contributions monthly.
inst_paid	real-valued multiplicative	The installment amount that the member has paid during the month
inst_spread	real-valued multiplicative	The spread between the installment due and the outstanding installment (The outstanding installment is calculated by: total_inst_due - total_inst_paid).
total_inst_due	real-valued multiplicative	The cumulative amounts of installments that are currently due from each of the members in the chit fund.
total_inst_paid	real-valued multiplicative	The cumulative amounts of installments that the member has paid in the chit fund.
div_due	real-valued multiplicative	The dividend that is currently due is received by members in each auction (The dividend is calculated by: monthly_contribution - inst_due).

div_paid	real-valued multiplicative	The dividend is based on what the member has paid during the month
total_div_due	real-valued multiplicative	The cumulative amounts of dividend due that are currently due from each of the members in the chit fund.
total_div_paid	real-valued multiplicative	The cumulative amounts of dividend paid that are currently due from each of the members in the chit fund.
participation	binary	If the member participated in the auction during the month (“Participate”: “1”; “Not Participate”: “0”).
all_bids	real-valued multiplicative	All Bids (Bid is discount price offered) would encompass the individual bids submitted by members participants during an auction.
win_loss	binary	If the member participants win the bid (“Yes”: “1”; “No”: “0”).
win_bid_amt	real-valued multiplicative	The amount of discount price the member bid to win the Bids
prized_amt	real-valued multiplicative	The amount of prize to the members who win the Bids (Prized Amount is calculated by: chit_value - prized_amt).
chit_value	*Referenced from table cf1_delhi_collateral*	
start_date	categorical	The date of starting paying a monthly installment
monthly_contribution	*Referenced from table cf1_delhi_collateral*	
duration	*Referenced from table cf1_delhi_collateral*	
month	categorical	Month of operation of chit fund
tot_memb	count	The total number of members of a chit-fund

fman_tkt	binomial	The management fee that the foreman receives in tickets during the month is approved by the total number of members in the fund
bylaw_no	categorical	The Law is applied by Chit fund
penalty	real-valued multiplicative	Penalty refers to a financial charge imposed on members for non-compliance with the agreed-upon terms and conditions.
postage_cost	real-valued multiplicative	The cost of postage may include the charges for sending communication, documents, or notices to the members via postal services.
nj_stamp_cost	real-valued multiplicative	The cost of stamp.
other_cost	real-valued multiplicative	The cost other may include: related to legal compliance, documentation, or regulatory requirements, cost specific software for its operations.
by_chq	binary	If members make installment payments by cheque (“Yes”: “1”; “No”: “0”).
by_cash	binary	If members make installment payments by cash (“Yes”: “1”; “No”: “0”).
by_other	binary	If members make installment payments by others (“Yes”: “1”; “No”: “0”).
bounced_chq	binary	If the member is bounced the cheque (“Yes”: “1”; “No”: “0”).
last_trans_date	categorical	The date of the last transaction made by the member in the month
last_payment_date	categorical	The date of last paying a monthly installment

missed_inst	binary	If the member does not make any transactions during the month (“Yes”: “1”; “No”: “0”).
missed_div	binary	If the member does not receive profit during the month (because no members bids) (“Yes”: “1”; “No”: “0”).
diff_inst	real-valued additive	Installment difference between total installment due and total installment paid (Installment difference is calculated by: total_inst_due - total_inst_paid).
no_trans	count	Number of transactions made by members during the month.
total_trans	count	The cumulative amounts of transactions
multi_payment	binary	If the member makes payments using multiple methods (“Yes”: “1”; “No”: “0”).
early_payment	binary	If members make payment transactions before the last date (“Yes”: “1”; “No”: “0”).
part_payment	binary	If members make payment transactions or not which the cumulative amounts of installments due are larger than the cumulative amounts of installments paid (total_inst_due > total_inst_paid) is considered a partial payment (“Yes”: “1”; “No”: “0”).
irr_payment	binary	If the members make payment transactions irregular (“Yes”: “1”; “No”: “0”).
late_payment	binary	If members make payment transactions after the last date (“Yes”: “1”; “No”: “0”).
default	binary	If a member default to fulfill their financial obligations as per the terms and conditions

		outlined in the chit agreement during the month (“Yes”: “1”; “No”: “0”).
monthly_income	real-valued multiplicative	The amount of income a member receives monthly
sex	binary	Member’s gender (“M”: “male”, “F”: “female”).
age	count	Member’s age.
occupation	categorical	Member’s occupation
lottery	binary	If an auction must be decided by lottery to see which member wins the Bids (“Yes”: “1”; “No”: “0”).
bid_type	categorical	Determine the bid type by counting how many people are participating in the auction. • “0”: “There are no members participating in the auction” • “1”: “There is one member participating in the auction” • “2”: “There is more than one member participating in the auction”.
before_after	binary	• The transaction is made **before** the auction is successful (win_loss = 1): “0”. • The transaction is made **after** the auction is successful (win_loss = 1): “1”.
all_trans	categorical	Count all transactions
default_90	binary	If the member defaults continuously for 3 months (“Yes”: “1”; “No”: “0”).

Table 3. 1: Describe all the variables of the 3 collected tables

To swiftly visualize the available data in tabular form, here is the data loaded in Excel

chit_id	p_recno	collateral	chit_value	duration	monthly_contribution	year
AB-002-01	8381		900000	300	30000	19990
AB-002-02	8382		900000	300	30000	19990
AB-002-03	8382		900000	300	30000	19990
AB-002-04	8382		900000	300	30000	19990
AB-002-05	7363		900000	300	30000	19990
AB-002-06	8383		900000	300	30000	19990
AB-002-07	8383		900000	300	30000	19990
AB-002-08	4164		900000	300	30000	19990
AB-002-09	7364		900000	300	30000	19990
AB-002-10	7365		900000	300	30000	19990
AB-002-11	7371		900000	300	30000	19990
AB-002-12	8384		900000	300	30000	19990
AB-002-13	8521		900000	300	30000	19990
AB-002-14	8385		900000	300	30000	19990
AB-002-15	8386		900000	300	30000	19990
AB-002-16	8387		900000	300	30000	19990
AB-002-17	3805		900000	300	30000	19990
AB-002-18	310		900000	300	30000	19990
AB-002-19	310		900000	300	30000	19990
AB-002-20	8388		900000	300	30000	19990
AB-002-21	1455		900000	300	30000	19990
AB-002-22	8389		900000	300	30000	19990
AB-002-23	7419		900000	300	30000	19990
AB-002-24	8390		900000	300	30000	19990
AB-002-25	4134		900000	300	30000	19990
AB-002-26	160		900000	300	30000	19990
AB-002-27	399		900000	300	30000	19990
AB-002-28	8288		900000	300	30000	19990

Figure 3. 1: Table cf1_delhi_collateral

chit	chit_id	p_recno	winning_aucn	duration	fman	surety_p_recno	n_surety	age	salary	sex	occupation	other_chits	surety_others	years_of_service
AB-002	AB-002-01	8381	190	300	0									
AB-002	AB-002-02	8382	80	300	0									
AB-002	AB-002-03	8382	180	300	0									
AB-002	AB-002-04	8382	300	300	0									
AB-002	AB-002-05	7363	70	300	0									
AB-002	AB-002-06	8383	50	300	0									
AB-002	AB-002-07	8383	60	300	0									
AB-002	AB-002-08	4164	10	300	0									
AB-002	AB-002-09	7364	240	300	0									
AB-002	AB-002-10	7365	20	300	0									
AB-002	AB-002-11	7371	260	300	0									
AB-002	AB-002-12	8384	160	300	0									
AB-002	AB-002-13	8521	230	300	0									
AB-002	AB-002-14	8385	210	300	0									
AB-002	AB-002-15	8386	200	300	0									
AB-002	AB-002-16	8387	130	300	0									
AB-002	AB-002-17	3805	40	300	0									
AB-002	AB-002-18	310	280	300	0									
AB-002	AB-002-19	310	290	300	0									
AB-002	AB-002-20	8388	220	300	0									
AB-002	AB-002-21	1455	30	300	0									
AB-002	AB-002-22	8389	150	300	0									
AB-002	AB-002-23	7419	270	300	0									
AB-002	AB-002-24	8390	110	300	0	88140	50					0	0	
AB-002	AB-002-24	8390	110	300	0	79820	50	370	60000	M	PS	10	10	
AB-002	AB-002-24	8390	110	300	0	88150	50					0	0	
AB-002	AB-002-24	8390	110	300	0	88130	50					0	0	
AB-002	AB-002-24	8390	110	300	0	88160	50					0	0	

Figure 3. 2: Table cf1_delhi_surety 1

house_owner	land_owner	income_tax	insurance_policy	insurance_amount	res_pin	off_pin	chit_value	monthly_contribution	year
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990
							900000	30000	19990

Figure 3. 3: Table cf1_delhi_surety 2

chit	chit_id	p_recno	aucn_no	aucn_date	inst_due	inst_paid	inst_spread	total_inst_due	total_inst_paid	div_due	div_paid	total_div_due	total_div_paid	participation	all_bids	win_loss
AB-002	AB-002-01	8381	1	01/03/1999	24000	40000	24000	24000	40000	6000	6000	6000	6000	0	0	0
AB-002	AB-002-01	8381	2	16/03/1999	24000	26000	24000	48000	66000	6000	6000	12000	12000	0	0	0
AB-002	AB-002-01	8381	3	01/04/1999	24000	10000	24000	72000	76000	6000	6000	18000	18000	0	0	0
AB-002	AB-002-01	8381	4	16/04/1999	24000	10000	14000	96000	86000	6000	6000	24000	24000	0	0	0
AB-002	AB-002-01	8381	5	01/05/1999	24000	30000	30000	120000	116000	6000	6000	30000	30000	0	0	0
AB-002	AB-002-01	8381	6	17/05/1999	24000	20000	20000	144000	136000	6000	6000	36000	36000	0	0	0
AB-002	AB-002-01	8381	7	01/06/1999	25000	36000	33000	169000	172000	5000	5000	41000	41000	0	0	0
AB-002	AB-002-01	8381	8	16/06/1999	25000	28000	25000	194000	200000	5000	5000	46000	46000	0	0	0
AB-002	AB-002-01	8381	9	01/07/1999	25000	26000	25000	219000	226000	5000	5000	51000	51000	0	0	0
AB-002	AB-002-01	8381	10	16/07/1999	25000	26000	25000	244000	252000	5000	5000	56000	56000	0	0	0
AB-002	AB-002-01	8381	11	02/08/1999	26000	81560	26000	270000	333560	4000	4000	60000	60000	0	0	0
AB-002	AB-002-01	8381	12	16/08/1999	26560	6000	26560	296560	339560	3440	3440	63440	63440	0	0	0
AB-002	AB-002-01	8381	13	01/09/1999	27000	10000	27000	323560	349560	3000	3000	66440	66440	0	0	0
AB-002	AB-002-01	8381	14	16/09/1999	26990	50290	26990	350550	399850	3010	0	69450	66440	0	0	0
AB-002	AB-002-01	8381	15	01/10/1999	27300	0	27300	377850	399850	2700	0	72150	66440	0	0	0
AB-002	AB-002-01	8381	16	16/10/1999	27230	0	22000	405080	399850	2770	0	74920	66440	0	0	0
AB-002	AB-002-01	8381	17	01/11/1999	27400	0	0	432480	399850	2600	0	77520	66440	0	0	0
AB-002	AB-002-01	8381	18	16/11/1999	27600	0	0	460080	399850	2400	13480	79920	79920	0	0	0
AB-002	AB-002-01	8381	19	01/12/1999	27930	82230	82230	488010	482080	2070	2070	81990	81990	10	107100	1
AB-002	AB-002-01	8381	20	16/12/1999	28430	28000	28000	516440	510080	1570	0	83560	81990	0	0	0
AB-002	AB-002-01	8381	21	03/01/2000	28930	0	0	545370	510080	1070	2640	84630	84630	0	0	0
AB-002	AB-002-01	8381	22	17/01/2000	29160	56000	56000	574530	566080	840	0	85470	84630	0	0	0
AB-002	AB-002-01	8381	23	01/02/2000	29160	0	0	603690	566080	840	1680	86310	86310	0	0	0
AB-002	AB-002-01	8381	24	16/02/2000	29300	40000	40000	632990	606080	700	700	87010	87010	0	0	0
AB-002	AB-002-01	8381	25	01/03/2000	29600	64000	56510	662590	670080	400	400	87410	87410	0	0	0
AB-002	AB-002-01	8381	26	16/03/2000	29800	14000	21490	692390	684080	200	200	87610	87610	0	0	0
AB-002	AB-002-01	8381	27	01/04/2000	29900	38000	38000	722290	722080	100	100	87710	87710	0	0	0
AB-002	AB-002-01	8381	28	17/04/2000	30000	56000	30210	752290	778080	0	0	87710	87710	0	0	0

Figure 3. 4: Table cf1_delhi_transaction_data 1

win_bid_amt	prized_amt	chit_value	start_date	monthly_contribution	duration	month	tot_memb	fman_tkt	bylaw_no	penalty	postage_cost	nj_stamp_cost	other_cost	by_chq
0	0	900000	27/02/1999	30000	300	10	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	20	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	30	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	40	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	50	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	60	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	70	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	80	30	0	F2/17/2104/98-99	0	0	0	0	1
0	0	900000	27/02/1999	30000	300	90	30	0	F2/17/2104/98-99	0	0	0	0	1
0	0	900000	27/02/1999	30000	300	100	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	110	30	0	F2/17/2104/98-99	0	0	0	0	1
0	0	900000	27/02/1999	30000	300	120	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	130	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	140	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	150	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	160	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	170	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	180	30	0	F2/17/2104/98-99	0	0	0	0	0
10710	792900	900000	27/02/1999	30000	300	190	30	0	F2/17/2104/98-99	0	0	0	10	0
0	0	900000	27/02/1999	30000	300	200	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	210	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	220	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	230	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	240	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	250	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	260	30	0	F2/17/2104/98-99	0	0	0	0	0
0	0	900000	27/02/1999	30000	300	270	30	0	F2/17/2104/98-99	0	0	0	0	1
0	0	900000	27/02/1999	30000	300	280	30	0	F2/17/2104/98-99	0	0	0	0	0

Figure 3. 5: Table cf1_delhi_transaction_data 2

by_other	bounced_chq	last_trans_date	last_payment_date	missed_inst	missed_div	diff_inst	no_trans	total_trans	multi_payment	early_payment	part_payment	irr_payment	late_payment	default
0	0	11/03/1999	15/03/1999	0	0	-16000	2	2	1	1	0	1	0	0
0	0	26/03/1999	31/03/1999	0	0	-18000	2	4	1	1	0	1	0	0
0	0	13/04/1999	15/04/1999	0	0	-4000	1	5	0	1	0	0	0	0
0	0	28/04/1999	30/04/1999	0	0	10000	2	7	0	0	1	1	1	0
0	0	07/06/1999	16/05/1999	0	0	4000	2	9	0	0	1	1	1	0
0	0	08/06/1999	31/05/1999	0	0	8000	1	10	0	0	1	1	1	0
0	0	09/07/1999	15/06/1999	0	0	-3000	4	14	1	1	0	1	0	0
0	0	05/08/1999	30/06/1999	0	0	-6000	3	17	1	1	0	1	0	0
0	0	10/08/1999	15/07/1999	0	0	-7000	1	18	0	1	0	0	0	0
0	0	09/09/1999	01/08/1999	0	0	-8000	2	20	1	1	0	1	0	0
0	0	25/10/1999	15/08/1999	0	0	-63560	3	23	1	1	0	1	0	0
0	0	26/10/1999	31/08/1999	0	0	-43000	1	24	0	1	0	0	0	0
0	0	16/11/1999	15/09/1999	0	0	-26000	1	25	0	1	0	0	0	0
0	0	27/11/1999	30/09/1999	0	1	-49300	1	26	0	1	0	0	0	0
1	0		15/10/1999	1	1	-22000	0	26	0	1	0	0	0	0
1	0		31/10/1999	1	1	5230	0	26	0	0	1	1	1	0
1	0		15/11/1999	1	1	32630	0	26	0	0	0	0	1	0
1	0		30/11/1999	1	0	60230	0	26	0	0	0	0	0	1
0	0	15/12/1999	15/12/1999	0	0	5930	2	28	0	0	1	1	1	0
0	0	21/02/2000	02/01/2000	0	1	6360	6	34	0	0	1	1	1	0
1	0		16/01/2000	1	0	35290	0	34	0	0	0	0	1	0
0	0	28/03/2000	31/01/2000	0	1	8450	2	36	0	0	1	1	1	0
1	0		15/02/2000	1	0	37610	0	36	0	0	0	0	1	0
0	0	31/03/2000	29/02/2000	0	0	26910	1	37	0	0	1	1	1	0
0	0	06/04/2000	15/03/2000	0	0	-7490	3	40	1	1	0	1	0	0
0	0	18/04/2000	31/03/2000	0	0	8310	3	43	0	0	1	1	1	0
0	0	05/05/2000	16/04/2000	0	0	210	5	48	0	0	1	1	1	0
0	0	23/05/2000	30/04/2000	0	0	-25790	2	50	1	1	0	1	0	0

Figure 3. 6: Table cf1_delhi_transaction_data 3

monthly_income	sex	age	occupation	lottery	bid_type	before_after	all_trans	default_90
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B			0	10	0
	M	350	B		20	10	10	0
	M	350	B			10	10	0
	M	350	B			10	10	0
	M	350	B			10	10	0
	M	350	B			10	10	0
	M	350	B			10	10	0
	M	350	B			10	10	0
	M	350	B			10	10	0
	M	350	B			10	10	0
	M	350	B			10	10	0

Figure 3. 7: Table cf1_delhi_transaction_data 4

3.3 Data cleaning

Table cf1_delhi_collateral

The table provides details on the types of collateral held by members. However, my focus is solely on determining the amount of collateral each member possesses. To accomplish this, I will compute the total amount of collateral for each member. In the "collateral" column, each entry corresponds to a member's collateral. The total amount of collateral for each member, along with the method for creating the column, is illustrated in a new column named "n_collateral" as depicted in the following figure.

```
1. df_cf1_delhi_collateral['n_collateral'] = df_cf1_delhi_collateral.groupby('chit_id')['c
   ollateral'].transform('count')
```

Table 3. 2: Code create column n_collateral

	chit_id	p_recno	n_collateral
0	AB-002-01	8381	0
1	AB-002-02	8382	0
2	AB-002-03	8382	0
3	AB-002-04	8382	0
4	AB-002-05	7363	0
...	...	...	...
8125	Z-014-26	1684	1
8126	Z-014-27	5493	1
8127	Z-014-28	19199	1
8128	Z-014-29	19199	0
8129	Z-014-30	1	0

7990 rows × 3 columns

Figure 3. 8: Table cf1_delhi_collateral after transformed

Table cf1_delhi_surety

The table contains information regarding the types of surety held by members. However, my focus lies solely on determining the amount of surety each member possesses. To achieve this, I will compute the total amount of surety for each member. In the "collateral" column, each entry corresponds to a member's surety. The total amount of surety for each member will be displayed in a new column named "n_surety".

	chit	chit_id	p_recno	n_surety
0	AB-002	AB-002-01	8381	0.0
1	AB-002	AB-002-02	8382	0.0
2	AB-002	AB-002-03	8382	0.0
3	AB-002	AB-002-04	8382	0.0
4	AB-002	AB-002-05	7363	0.0
...	...	...	...	...
19783	Z-014	Z-014-26	1684	6.0
19789	Z-014	Z-014-27	5493	4.0
19793	Z-014	Z-014-28	19199	5.0
19798	Z-014	Z-014-29	19199	1.0
19799	Z-014	Z-014-30	1	0.0

7990 rows × 4 columns

Figure 3. 9: Table cf1_delhi_surety group after transformed

Table cf1_delhi_transaction_data

The table "cf1_delhi_transaction_data" contains transaction records of members in the chit fund, aiding in the assessment of their credit capacity. During initial analysis, it was noted that the "bid_type" column contained missing data due to data entry issues. This column denotes the number of members participating in an auction, which can be computed by grouping the data according to the "chit" column and filtering based on the condition "participation" = 1, then counting the resulting rows.

Considering the provided data description, I have adjusted the "bid_type" values greater than or equal to 2 to 2, as specified.

```
1. cf1_delhi_transaction_data['bid_type'] = cf1_delhi_transaction_data[(cf1_delhi_transact
   ion_data['participation'] == 1) & (cf1_delhi_transaction_data['all_bids'] != 0)].groupb
   y(['chit', 'month'])['participation'].transform('count')

2. cf1_delhi_transaction_data['bid_type'] = cf1_delhi_transaction_data['bid_type'].apply(l
   ambda x: 2 if x > 2 else x)
```

Table 3. 3: Regenerate data for the "bid_type" column

	chit	chit_id	p_recno	aucn_date	inst_due	inst_paid	inst_spread	total_inst_due	total_inst_paid	div_due	...	late_payment	default	monthly_income	sex
0	AB-002	AB-002-01	8381	1999-03-01	2400.0	4000.0	2400.0	2400.0	4000.0	600.0	...	0	0	NaN	M
1	AB-002	AB-002-01	8381	1999-03-16	2400.0	2600.0	2400.0	4800.0	6600.0	600.0	...	0	0	NaN	M
2	AB-002	AB-002-01	8381	1999-04-01	2400.0	1000.0	2400.0	7200.0	7600.0	600.0	...	0	0	NaN	M
3	AB-002	AB-002-01	8381	1999-04-16	2400.0	1000.0	1400.0	9600.0	8600.0	600.0	...	1	0	NaN	M
4	AB-002	AB-002-01	8381	1999-05-01	2400.0	3000.0	3000.0	12000.0	11600.0	600.0	...	1	0	NaN	M
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
294895	Z-014	Z-014-30	1	2008-03-24	9920.0	9900.0	9920.0	237820.0	237900.0	80.0	...	0	0	NaN	M
294896	Z-014	Z-014-30	1	2008-04-26	9970.0	9920.0	9970.0	247790.0	247820.0	30.0	...	0	0	NaN	M
294897	Z-014	Z-014-30	1	2008-05-24	10000.0	9970.0	10000.0	257790.0	257790.0	0.0	...	0	0	NaN	M
294898	Z-014	Z-014-30	1	2008-06-28	10000.0	10000.0	10000.0	267790.0	267790.0	0.0	...	0	0	NaN	M
294899	Z-014	Z-014-30	1	2008-07-26	10000.0	10000.0	10000.0	277790.0	277790.0	0.0	...	0	0	NaN	M

294900 rows × 48 columns

Figure 3. 10: Table cf1_delhi_surety after transformed

Merge 3 tables into one final table

Currently there are 3 separate data tables. To facilitate exploratory analysis and input into machine learning models, I will merge these 3 tables into a single table, first we need merge table cf1_delhi_collateral with table cf1_delhi_surety,

```
merge_collateral_surety = pd.merge(cf1_delhi_surety_final, df_cf1_delhi_collateral_fina
l, on=['chit_id', 'p_recno'], how='outer')
```

Table 3. 4: The code merge 2 tables table cf1_delhi collateral, table cf1_delhi surety

	chit	chit_id	n_surety	n_collateral
0	AB-002	AB-002-01	0.0	0
1	AB-002	AB-002-02	0.0	0
2	AB-002	AB-002-03	0.0	0
3	AB-002	AB-002-04	0.0	0
4	AB-002	AB-002-05	0.0	0
...	...	...	...	...
7985	Z-014	Z-014-26	6.0	1
7986	Z-014	Z-014-27	4.0	1
7987	Z-014	Z-014-28	5.0	1
7988	Z-014	Z-014-29	1.0	0
7989	Z-014	Z-014-30	0.0	0

7990 rows × 4 columns

Figure 3. 11: Merge 2 tables table cf1_delhi collateral, table cf1_delhi surety

Seconds, merge table cf1_delhi_transaction_data with merge_collateral_surety table into one final dataframe

```
df = pd.merge(cf1_delhi_transaction_data_final, merge_collateral_surety_final,
on=['chit', 'chit_id'], how='left')
```

Table 3. 5: The code merge 3 tables table cf1_delhi collateral, table cf1_delhi surety, cf1_delhi_transaction_data

	chit	chit_id	p_recno	aucn_date	inst_due	inst_paid	inst_spread	total_inst_due	total_inst_paid	div_due	...	monthly_income	sex	age	occupation
0	AB-002	AB-002-01	8381	1999-03-01	2400.0	4000.0	2400.0	2400.0	4000.0	600.0	...	NaN	M	35.0	B
1	AB-002	AB-002-01	8381	1999-03-16	2400.0	2600.0	2400.0	4800.0	6600.0	600.0	...	NaN	M	35.0	B
2	AB-002	AB-002-01	8381	1999-04-01	2400.0	1000.0	2400.0	7200.0	7600.0	600.0	...	NaN	M	35.0	B
3	AB-002	AB-002-01	8381	1999-04-16	2400.0	1000.0	1400.0	9600.0	8600.0	600.0	...	NaN	M	35.0	B
4	AB-002	AB-002-01	8381	1999-05-01	2400.0	3000.0	3000.0	12000.0	11600.0	600.0	...	NaN	M	35.0	B
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
294895	Z-014	Z-014-30	1	2008-03-24	9920.0	9900.0	9920.0	237820.0	237900.0	80.0	...	NaN	M	NaN	NaN
294896	Z-014	Z-014-30	1	2008-04-26	9970.0	9920.0	9970.0	247790.0	247820.0	30.0	...	NaN	M	NaN	NaN
294897	Z-014	Z-014-30	1	2008-05-24	10000.0	9970.0	10000.0	257790.0	257790.0	0.0	...	NaN	M	NaN	NaN
294898	Z-014	Z-014-30	1	2008-06-28	10000.0	10000.0	10000.0	267790.0	267790.0	0.0	...	NaN	M	NaN	NaN
294899	Z-014	Z-014-30	1	2008-07-26	10000.0	10000.0	10000.0	277790.0	277790.0	0.0	...	NaN	M	NaN	NaN

294900 rows × 50 columns

Figure 3. 12: Merge 3 tables table cf1_delhi collateral, table cf1_delhi surety, cf1_delhi_transaction_data

3.3.1 Handling missing data

After merging the three tables into a single table, the next step is to check for missing values. Utilizing the isnull().sum() function in Python, I found that there are seven columns with missing values. These columns are: all_bids, monthly_income, sex, age, occupation, and lottery.

Column all_bids

Upon inspecting the all_bids column, I observed a missing value. Further investigation is necessary to determine whether this is an error that I can rectify on my own.

	chit	chit_id	month	all_bids	participation	win_loss	win_bid_amt	prized_amt	lottery
245730	W-063	W-063-24	11.0	NaN	1.0	0	0	0.0	0.0
247610	W-064	W-064-31	11.0	NaN	1.0	0	0	0.0	0.0

Figure 3. 13: Missing values in the "all_bids" column

Upon deeper analysis of the auctions where members participated and the missing data in the "all_bids" column, it is evident that the participating members in these auctions won through the lottery ("lottery" = 1). Therefore, the value of "all_bids" is equivalent to the winning bid value.

Column monthly_income

I conducted a count of monthly income by chit_id to verify if any member possesses more than one monthly income value. The outcome indicates that no member possesses more than one monthly income value. Consequently, since the missing values cannot be derived from existing data, I will replace the missing values with the symbol 'NA'.

Column sex

Similar to the approach with the monthly_income column, I conducted a count to ascertain the number of genders each member has. The results affirm that each member possesses only one "sex" value. Therefore, I will replace the missing values with the symbol 'NA'.

Column age

Following the same methodology applied to the sex column, I conducted a count to determine the number of ages each member has. The findings reveal that each member possesses only one "age" value. Consequently, I will replace the missing value with the symbol 'NA'

Column occupation

Similar to the process employed for the age column, I performed a count to ascertain the number of occupations each member holds. The findings confirm that each member possesses only one "occupations" value. Hence, I will replace the missing value with the symbol 'NA'.

Column lottery

The values in the lottery column can be determined through inference from other columns. Thus, I will fill in the missing values by recalculating the entire lottery column. As described in the data, the lottery column signifies whether members have won bids through a lottery system or not. Therefore, I will compute the values for the lottery column by grouping the data according to the 'chit' and 'month' columns (as 'month' also represents auctions), filtering the data where 'participation' equals 1, and calculating the highest auction values (all_bids). If an auction has more than one highest value, I will designate 'lottery' as 1; conversely, it will be set as 0.

3.3.2 Create new columns

Column std_inst_paid

According to the data description provided for the inst_paid column, it represents the actual payments made by each member towards their monthly installments. However, for the purpose of credit classification based on individual members, it's necessary to transform the inst_paid values to be unique for each member. To achieve this, I will calculate the standard deviation for monthly installments. This will result in each member having only one standard deviation value for their monthly installments. A higher standard deviation indicates uneven payment patterns, suggesting that the member often underpays.

To create the std_inst_paid column, I will group the data by chit_id, access the "inst_paid" column, and then apply the transform function using the "std" method. The implementation code is provided in the following table:

```
df2['std_inst_paid'] = (df2.groupby('chit_id')['inst_paid']
                           .transform('std'))
```

Table 3. 6: The code create column 'std_inst_paid

Column prized_amt/chit_value'

Each member possesses a prized_amt value, but this may introduce bias since the prized_amt value tends to increase with the chit value. To mitigate this bias, I will transform the win_bids value into a prized_amt ratio relative to the chit_value. This new column will provide a fairer representation, indicating the proportion of bids each member made relative to their chit value.

To create this column, I will transform the two columns, prize_amt and chit_value, into another dataframe. First, I will group by chit_id and then use the agg function to transform the prize_amt column, obtaining the largest value. For the chit_value column, I will extract the first value. Subsequently, from the newly created dataframe containing these two transformed columns, I will calculate the prized_amt/chit_value column by dividing the values from the prize_amt column by the values from the chit_value column. Finally, I will join the values of the prized_amt/chit_value column from the new dataframe to the old dataframe. The implementation code is provided in the following table:

```
df2_prized = df2.groupby('chit_id').agg({'prized_amt': 'max', 'chit_value': 'first'}).reset_index()
df2_prized['prized_amt/chit_value'] = df2_prized['prized_amt'] / df2_prized['chit_value']
```

```
# Merge back into the original dataframe
df2 = pd.merge(df2, df2_prized[['chit_id', 'prized_amt/chit_value']], on='chit_id', how='left')
```

Table 3. 7: The code create column 'prized_amt/chit_value'

Column ratio_missed_inst

The missed_inst column indicates whether a member received a profit on the paid installment in a given month. Aligning with the objective of classifying each member's credit, I aim to compute a value of missed_inst unique to each member. To achieve this, I will convert the missed_inst value into the missed_inst rate. This represents the percentage of members who miss out on paid profits relative to the total chit participation time.

To create a new column named ratio_missed_inst, I will group the data by chit_id, access the missed_inst column, and count the occurrences where missed_inst equals 1. Then, I will divide this count by the total time of participation in the chit fund. This will give us the ratio of missed installments for each chit. The implementation code is provided in the following table:

```
df2['ratio_missed_inst'] = (df2.groupby('chit_id')['missed_inst']
                            .transform(lambda x: (sum(x == 1) / len(x))))
```

Table 3. 8: The code create column 'ratio_missed_inst'

Column std_diff_inst

Similar to the std_inst_paid column, I aim to ensure that each member represents a unique value. The diff_inst column is calculated by subtracting the cumulative total paid from the cumulative total payable. This indicates whether each member is in debt and the amount owed. Therefore, I will calculate the standard deviation of diff_inst, which represents the average difference in monthly installment debts of the member. A higher standard deviation of diff_inst indicates that the debts of members in each installment period are larger.

The code to create a new column is similar to the std_inst_paid column and is shown in the following table:

```
df2['std_diff_inst'] = (df2.groupby('chit_id')['diff_inst']
                            .transform('std'))
```

Table 3. 9: The code create column 'std_diff_inst'

Column ratio_early_payment

Similar to the ratio_missed_inst column, I aim to determine the unique early_payment value that represents each member. The early_payment column indicates whether the member's payment date precedes the last allowed payment date. I will calculate the early payment rate for each member by dividing the number of months of early payment by the total number of months of the member's operation in the chit fund. The early payment rate showcases the percentage of members who have made early payments out of the total number of payments.

The code to create a new column is similar to the ratio_missed_inst column and is shown in the following table:

```
1.  df2['ratio_early_payment'] = (df2.groupby('chit_id')['early_payment']
2.                               .transform(lambda x: (sum(x == 1) / len(x))))
```

Table 3. 10: The code create column 'ratio_early_payment'

Column ratio_part_payment

Similar to the ratio_early_payment column, I aim to determine the unique part_payment value that represents each member. The part_payment column indicates whether the member has paid the monthly installment due but only made a partial payment. I will calculate the part payment rate for each member by dividing the number of months of part payment by the total number of months of the member's operation in the chit fund. The part payment rate showcases the percentage of members who have made part payments out of the total number of payments.

The code to create a new column is similar to the ratio_part_payment column and is shown in the following table:

```
1.  df2['ratio_part_payment'] = (df2.groupby('chit_id')['part_payment']
2.                               .transform(lambda x: (sum(x == 1) / len(x))))
```

Table 3. 11: The code create column 'ratio_part_payment'

Column ratio_late_payment

Similar to the ratio_late_payment column, I aim to determine the unique late_payment value that represents each member. The late_payment column indicates whether the member's payment date after the last allowed payment date. I will calculate the late payment rate for each member by dividing the number of months of late payment by the total number of months

of the member's operation in the chit fund. The late payment rate showcases the percentage of members who have made late payments out of the total number of payments.

The code to create a new column is similar to the ratio_missed_inst column and is shown in the following table:

```
df2['ratio_late_payment'] = (df2.groupby('chit_id')['late_payment']
                             .transform(lambda x: (sum(x == 1) / len(x))))
```

Table 3. 12: The code create column 'ratio_late_payment'

Column ratio_default

Similar to the ratio_late_payment column, I aim to determine the unique default value that represents each member. The ratio_default column indicates whether a member violates the chit fund rules during the month. I will calculate the default rate for each member by dividing the number of months that the member violates by the total number of months of the member's operation in the chit fund. The default rate showcases the percentage of members who have violated the rules out of the total number of months of operation in the chit fund.

The code to create a new column is similar to the ratio_late_payment column and is shown in the following table:

```
df2['ratio_default'] = (df2.groupby('chit_id')['default']
                        .transform(lambda x: (sum(x == 1) / len(x))))
```

Table 3. 13: The code create column 'ratio_default'

Column ratio_default_90

Similar to the ratio_default column, I aim to determine the unique default_90 value that represents each member. The ratio_default_90 column indicates whether the member has continuously violated the chit fund rules within the last 2 months. I will calculate the default rate_90 for each member by dividing the number of months that the member violates continuously within the last 2 months by the total number of months of the member's operation in the chit fund. The default rate_90 showcases the percentage of members who have continuously violated the rules within the last 2 months out of the total number of months of operation in the chit fund.

The code to create a new column is similar to the ratio_default column and is shown in the following table:

```
df2['ratio_default_90'] = (df2.groupby('chit_id')['default_90']
                           .transform(lambda x: (sum(x == 1) / len(x))))
```

Table 3. 14: The code create column 'ratio_default_90'

3.4 Exploratory Data Analysis

After processing the data, I will conduct exploratory analysis to comprehend the data and uncover insights within the dataset.

Age distribution in the dataset

First, let's examine the age distribution of chit fund participants in the dataset. We observe a broad age range among chit fund participants in India, spanning from 14 to 86 years old, as depicted in the chart. Notably, the age group from 25 to 54 constitutes the majority and exhibits a steady increase, indicating that this demographic has substantial financial needs such as home purchases, car acquisitions, etc. However, beyond age 54, the number of members participating in chit funds gradually declines, suggesting a trend wherein as individuals age, their demand for loans and financial savings diminishes.

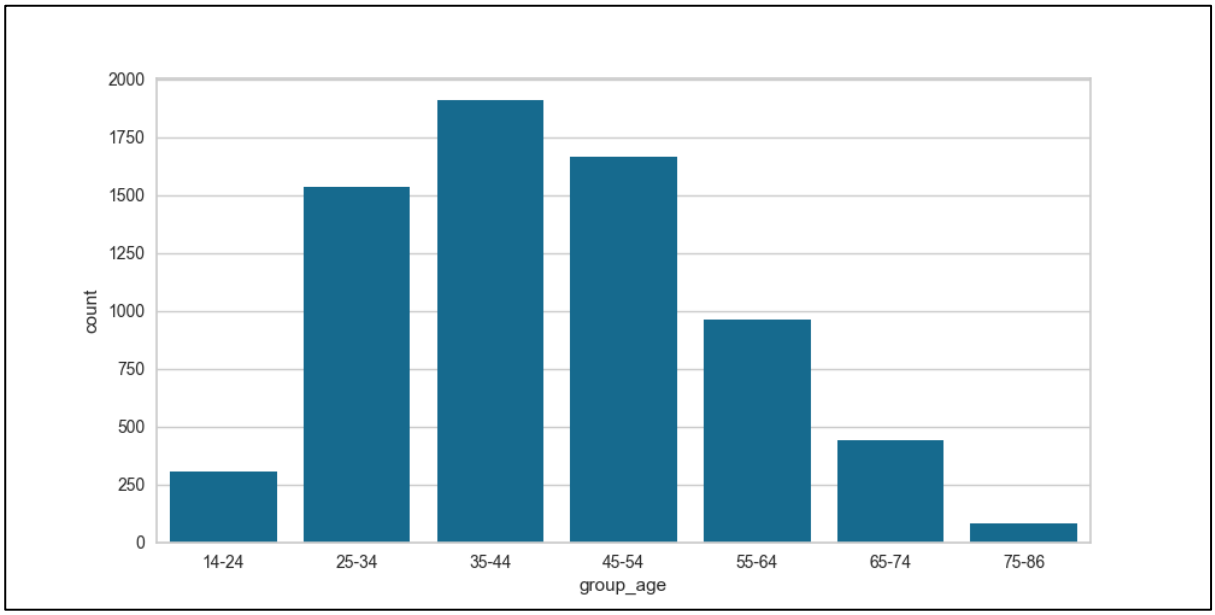

Figure 3. 14: Age distribution in the dataset

Which Group Age tends to have higher chit value?

From the observed age groups, I calculated the average chit value for each age group's participation. It is evident that the middle-aged group (ranging from 45 to 74 years old) tends to engage in chit funds with larger chit values compared to the younger age group (ranging

from 14 to 44 years old). This observation is reasonable because middle-aged members have had more time to accumulate assets, possess higher income to afford larger installments, or have greater financial needs such as purchasing a house or car.

In contrast, the senior age group (ranging from 75 to 86 years old) tends to participate in chit funds with lower chit values. This trend may be attributed to fewer individuals in this age group participating in chit funds, and they may also have fewer financial needs.

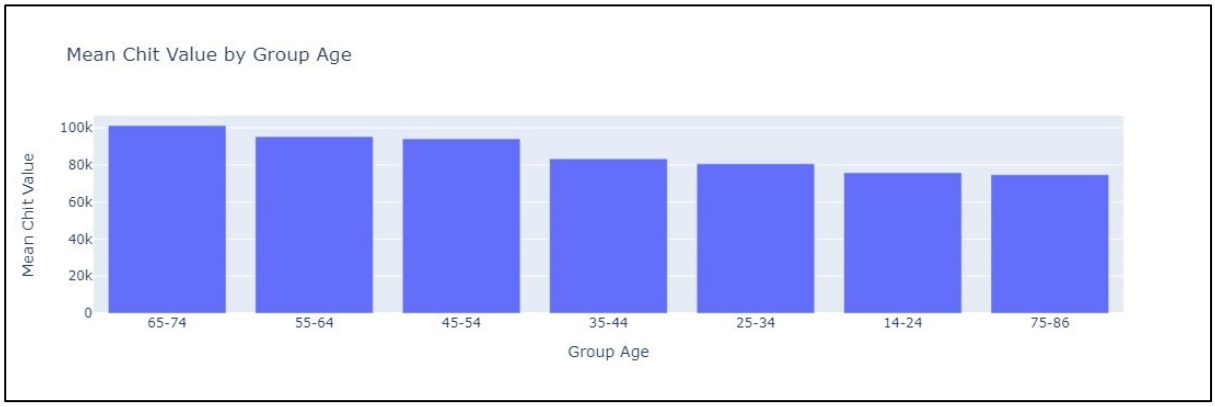

Figure 3. 15: Mean Chit Value by Group Age

Which age group is more likely to make early payments, and conversely?

I also examined the early payment rates and late payment rates for each age group. From the two charts below, we observe that as age increases, the late payment rate decreases, while the early payment rate increases. This trend suggests that older members are less likely to make late payments and more likely to make early payments.

One possible explanation for this trend is that as members grow older, they are more likely to achieve financial stability, possibly due to career success leading to higher and more stable incomes. Additionally, older members may have accumulated more financial experience and are more conscientious about managing their finances responsibly.

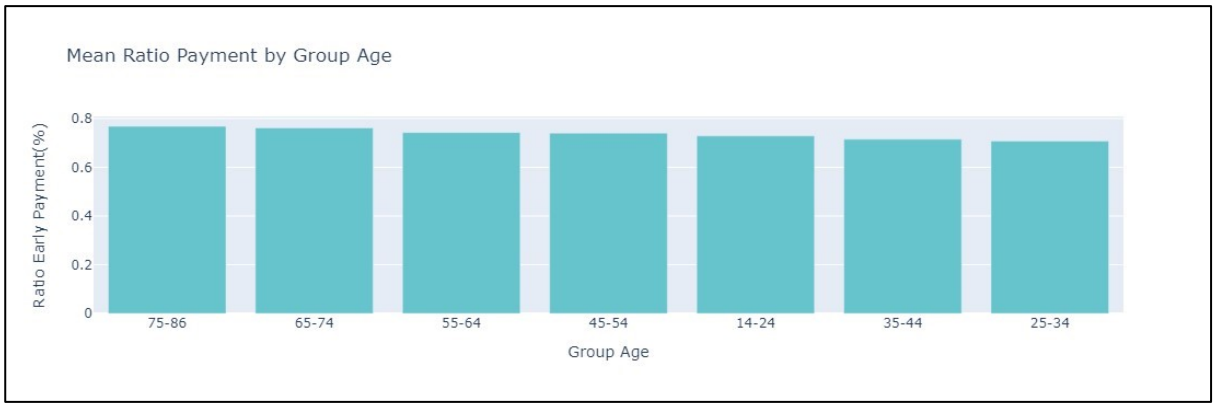

Figure 3. 16: Mean Ratio Payment by Group Age

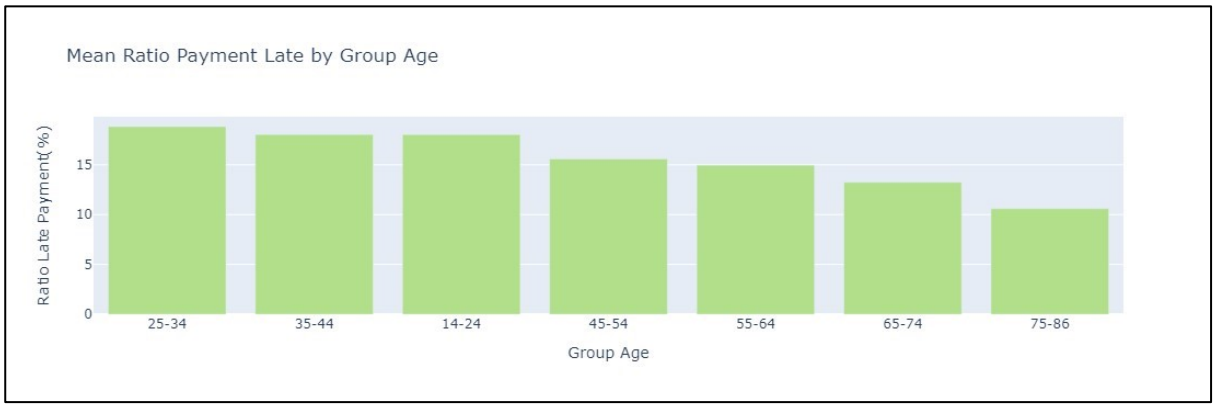

Figure 3. 17: Mean Ratio Payment Late by Group Age

Number of Chit Fund by Year

The number of established chit funds also shows a trend of annual increase, reaching its peak in 2001. Subsequently, it maintained at 80% compared to the peak year. However, in 2007, the number of newly established chit funds declined to nearly 90% compared to 2001, and dropped by about 70% compared to the previous year. This decline can be attributed to the global economic crisis, which resulted in financial hardships for people, thereby reducing their inclination to establish new chit funds.

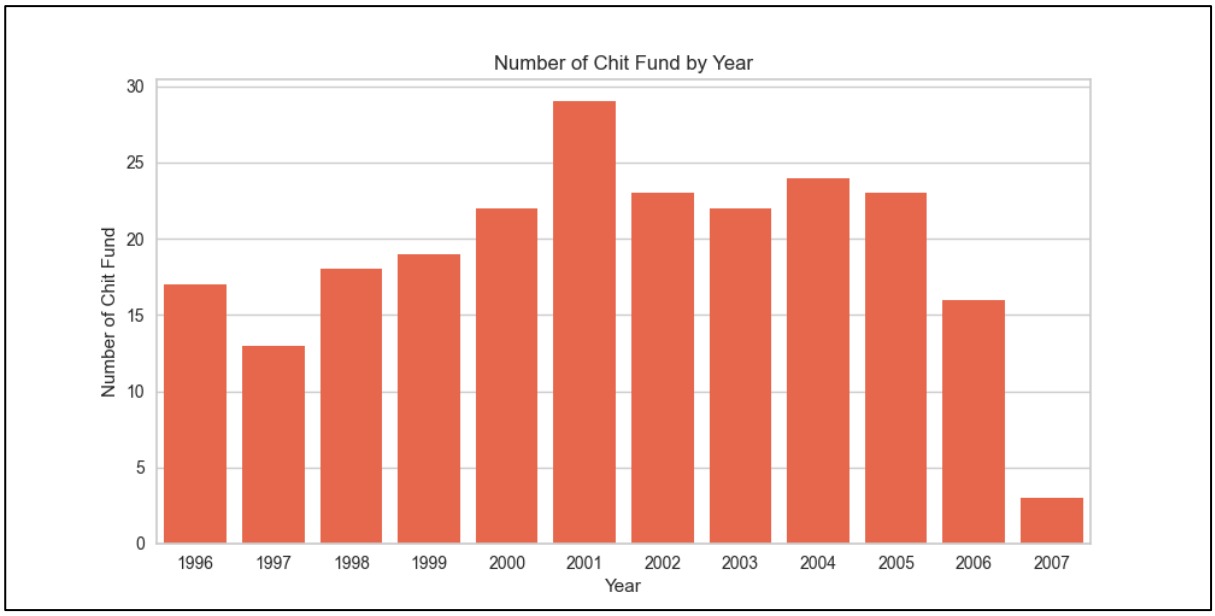

Figure 3. 18: Number of Chit Fund by Year

3.5 Data Preprocessing

During the data preprocessing step, i need to select the appropriate columns to include in the model, normalize the data, and adjust the dataframe according to the goal of the problem. The dataframe used for exploratory data analysis (EDA) is grouped by chit, representing each member's transactions throughout their participation in the Chit Fund. However, the research problem focuses on classifying the credit score of each member in the Chit Fund. Therefore, to align with the problem, i will group the data by the "Chit_id" column and select the necessary columns for the model. The final dataframe to be included in the model is prepared using the following code:

```python
# Select the necessary columns to include in the classification model
df_model = df2[[
    'chit_id',
    # 'std_inst_paid',
    # 'total_inst_paid', 'total_div_paid',
    # 'win_bid_amt',
    'prized_amt/chit_value',
    # 'chit_value',
    # 'monthly_contribution', 'duration',
    # 'ratio_missed_inst',
    'std_diff_inst',
    'ratio_early_payment',
    # 'ratio_part_payment',
    # 'ratio_late_payment',
    # 'ratio_default',
    # 'ratio_default_90',
    # 'n_surety',
    'n_collateral'
]]

# Group by dataframe by chit id, creating a data frame used to train the final model
df_model_final = df_model.groupby('chit_id').max().reset_index()

df_model_final.drop(columns = ['chit_id'], inplace = True)
df_model_final
```

Figure 3. 19: The code processes the dataframe to include in the model

The final dataset selected includes numerous columns. However, after conducting multiple testing processes and utilizing methods to evaluate the clustering performance of the data, only four columns were deemed essential for building a classification model. These columns are: prized_amt/chit_value, std_diff_inst, ratio_early_payment, and n_collateral. The final DataFrame to be used for machine learning modeling.

	prized_amt/chit_value	std_diff_inst	ratio_early_payment	n_collateral
0	0.881000	2586.594508	0.466667	0
1	0.783333	1151.831733	0.800000	0
2	0.870000	3338.041029	0.500000	0
3	0.950000	1421.110129	0.900000	0
4	0.783333	2174.012054	0.966667	0
...	...	...	...	...
7985	0.775000	2757.051961	0.800000	1
7986	0.750000	4118.798860	0.733333	1
7987	0.750000	3434.411681	0.666667	1
7988	0.836500	3437.982188	0.633333	0
7989	1.000000	672.447836	0.866667	0

7990 rows × 4 columns

Figure 3. 20: The dataframe includes the columns finally selected to be included in the model

Finally, apply the Standard Scaler method to standardize the data. This process ensures that the resulting data will have a mean of 0 and a standard deviation of 1. Standardization helps to eliminate the influence of measurement units and speeds up the learning process.

```
array([[ 0.31723324,  0.20535528, -1.23421436, -0.51878573],
       [-0.70489387, -0.15091565,  0.37274393, -0.51878573],
       [ 0.20211312,  0.39194957, -1.07351853, -0.51878573],
       ...,
       [-1.05374271,  0.4158797 , -0.27003939,  1.60038503],
       [-0.14847996,  0.41676631, -0.43073521, -0.51878573],
       [ 1.56262361, -0.26995313,  0.69413559, -0.51878573]])
```

Figure 3. 21: Data after being standardized

3.6 Data Modeling

Choosing the appropriate number of clusters is crucial when applying clustering algorithms to a dataset, such as k-means clustering, which necessitates specifying the number of clusters, denoted as k, to be produced. This process is somewhat arbitrary and represents one of the most challenging aspects of conducting cluster analysis.

To identify the optimal number of clusters, the study employs two methods: the Elbow method and the Silhouette method.

Elbow method

The Elbow method serves as a technique to assist in determining the suitable number of clusters by analyzing a visualization graph, focusing on the diminishing rate of the distortion function and identifying the elbow point. The elbow point represents the point at which the rate of decline of the distortion function undergoes the most significant change. Beyond this point, increasing the number of clusters does not substantially decrease the distortion function. By dividing according to the number of clusters at this position, the algorithm achieves a balance between capturing general clustering properties and avoiding overfitting. In the provided figure, it is observed that the primary elbow point occurs at k = 3, indicating that increasing the number of clusters beyond k = 3 does not significantly reduce the rate of decline of the distortion function.

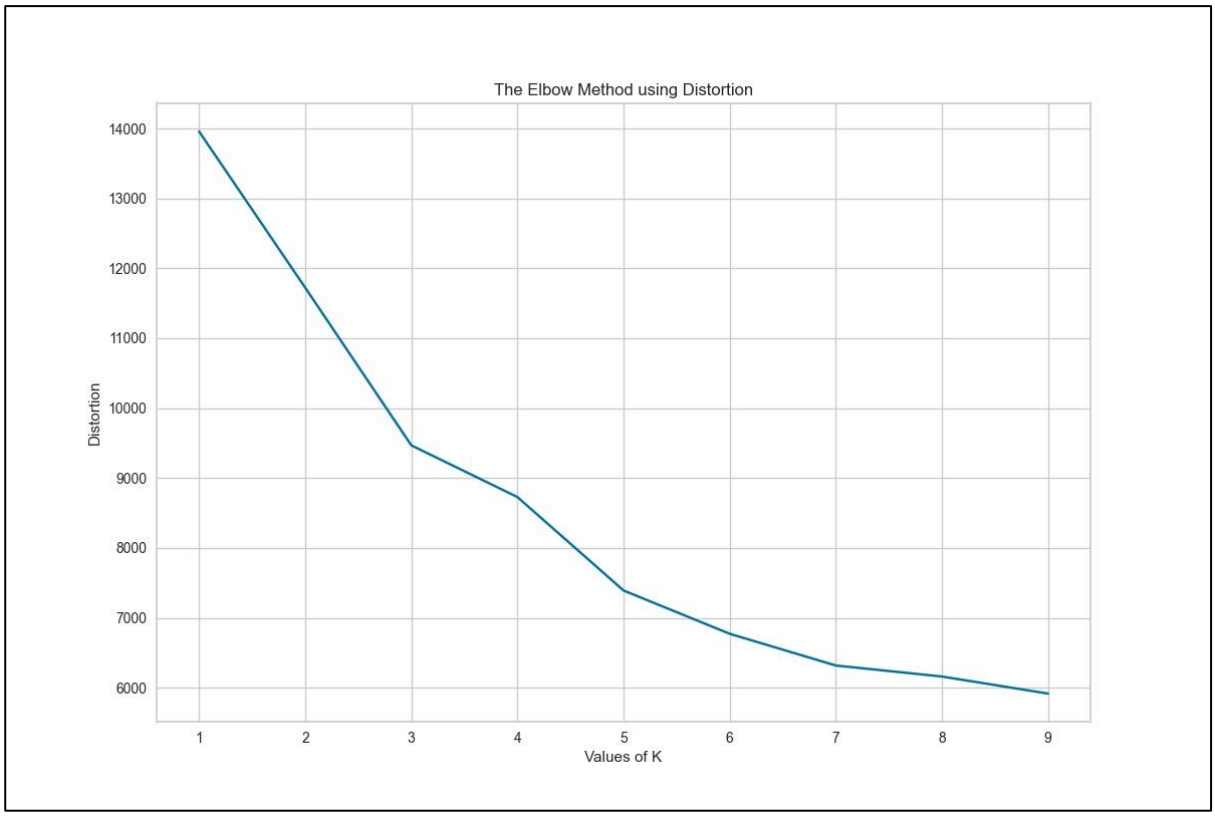

Figure 3. 22: The Elbow Method using Distortion

Silhouette method

Silhouette scores assess the clustering quality of algorithms like K-Means by measuring how well samples are grouped with similar ones. Each sample is assigned a Silhouette score, calculated by comparing its distances to samples in all clusters. Scores range from -1 to 1: a score of 1 indicates a dense and distinct cluster, while values near 0 imply overlapping clusters with samples near decision boundaries. Negative scores [-1, 0] suggest potential misassignments of samples.

The Silhouette analysis depicted in the above graphs aims to determine the best value for n_clusters.

- Values of 4, 5, 6, and 7 for n_clusters are deemed suboptimal for the given data due to clusters with silhouette scores below the average and wide variations in silhouette cell sizes.
- Values of 2 and 3 for n_clusters appear to be optimal, as each cluster's silhouette score exceeds the average, and size variations are similar. Additionally, the uniform thickness of silhouette charts aids in decision-making. For n_clusters = 3, the thickness

of silhouette cells is more consistent compared to n_clusters = 2, where one cluster's thickness significantly outweighs the other. Therefore, the optimal number of clusters can be selected as **3**.

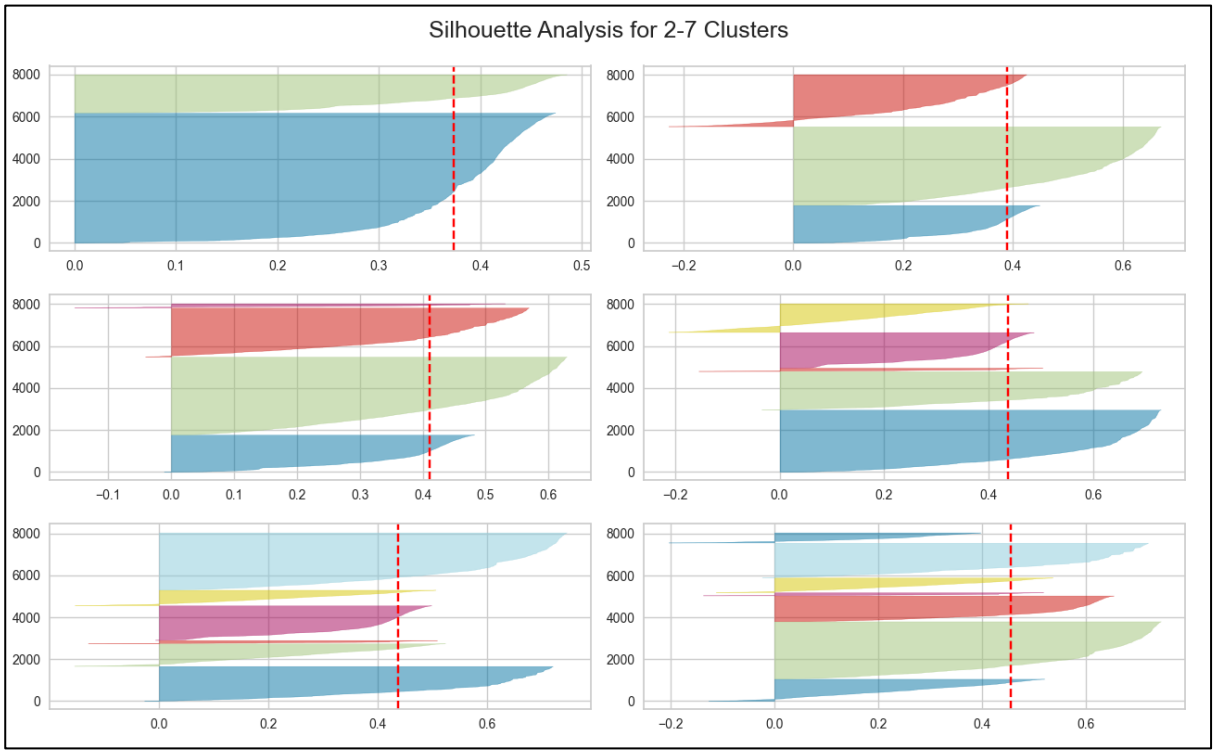

Figure 3. 23: Sihouetle Analysis

Traing and predict model

After selecting the optimal number of clusters as 3, I will proceed to construct a model and then generate predictions using the variable y_kmeans.

```
1.  kmeans = KMeans(n_clusters = 3, init = 'k-means++', random_state = 42)
2.  y_kmeans = kmeans.fit_predict(df_scale)
```

Table 3. 15: The code buid Data Modeling

CHAPTER 4. EXPERIMENTAL RESULTS

4.1 Research results

After the model categorizes the data into three groups, I will perform a statistical analysis of all columns in tabular format as follows. Generally, it is evident that cluster 1 exhibits superior average financial metrics compared to clusters 0 and 2. For instance, the average std_inst_paid is notably lower in cluster 1 than in the other two clusters, and the prized_amt/chit_value ratio is also higher. Regarding financial behavior, cluster 1 demonstrates a higher average early payment rate and lower average default rate (ratio_default) and 3-month default rate (ratio_default_90) compared to the other two groups.

Cluster	0	1	2
std_inst_paid	2054,641	1210,302	2349,367
total_inst_paid	122233,6	85188,98	88728,37
total_div_paid	12573,55	8898,64	10358,3
win_bid_amt	19259,61	7729,559	23767,61
prized_amt/chit_value	0,864855	0,916719	0,740613
chit_value	134807,2	94087,62	99086,67
monthly_contribution	3989,869	2799,002	2783,009
duration	36,2493	35,60021	39,36412
ratio_missed_inst	0,132279	0,120674	0,203583
std_diff_inst	1959,882	1107,694	2599,844
ratio_early_payment	0,748857	0,75919	0,64853
ratio_part_payment	0,111722	0,103327	0,165421
ratio_irr_payment	0,202825	0,188149	0,247061
ratio_late_payment	0,148278	0,136421	0,223495
ratio_default	0,026643	0,02388	0,063919
ratio_default_90	0,015558	0,014276	0,043865
n_surety	1,155394	1,140943	3,303767
n_collateral	1,078256	0	0,010936

Table 4. 1: Compare the tabular overview of the average values between the three clusters

Firstly, there is an overview chart displaying the count of each cluster in the classified models within the dataset. It indicates that the number of members in cluster 1 is the highest, followed by cluster 2, and finally cluster 0.

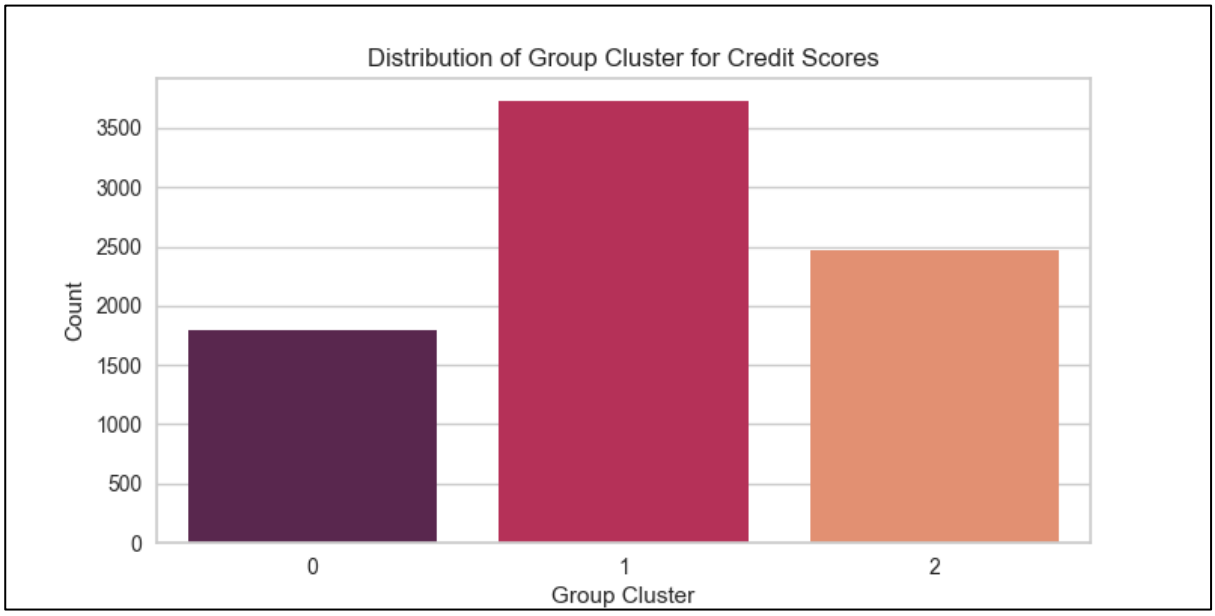

Figure 4. 1: Distribution of Group Cluster

Next, the chart illustrates the distribution of collateral and surety. Collateral amounts predominantly belong to members of cluster 0, while the surety amount is relatively equal between members in cluster 1 and cluster 2.

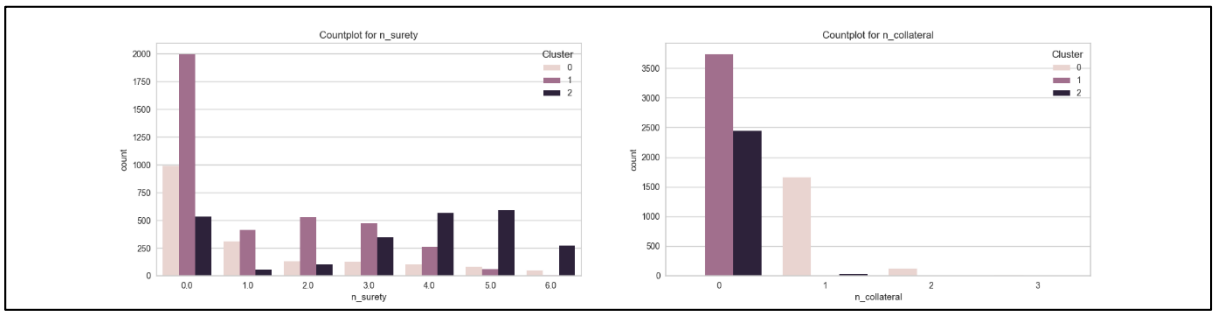

Figure 4. 2: Compare n_surety and n_collateral by cluster

To compare the dispersion of financial transaction indicators, box plot charts are utilized, grouped according to cluster values. Similar to the comparison table above, it's apparent that members of cluster 1 exhibit better financial indicators, particularly in the prized_amt/chit_value column. Additionally, other indicators show relatively minor differences across clusters.

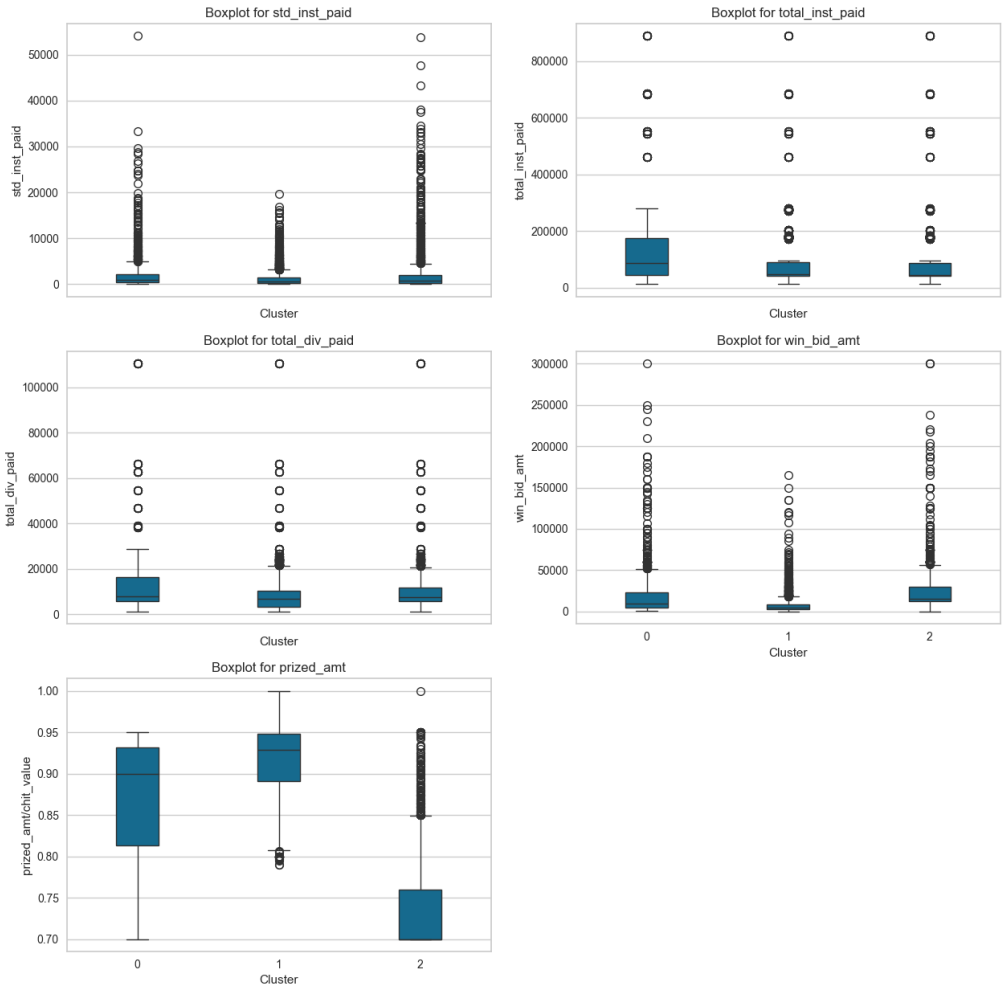

Figure 4. 3: Compare values by cluster using box plot

Finally, the scatter plot depicts the correlation between the prized_amt/chit_value and std_diff_inst columns. It's evident that the majority of members in cluster 1 are situated at the bottom right, indicating a high prized_amt/chit_value and low std_diff_inst. Meanwhile, members of cluster 2 predominantly occupy the lower and middle left portions of the plot, signifying a lower prized_amt/chit_value and higher std_diff_inst compared to cluster 1. Members of cluster 0 exhibit similar characteristics to those of cluster 1, although some points have higher std_diff_inst values and prized_amt/chit_value lower than cluster 1 but higher than cluster 2.

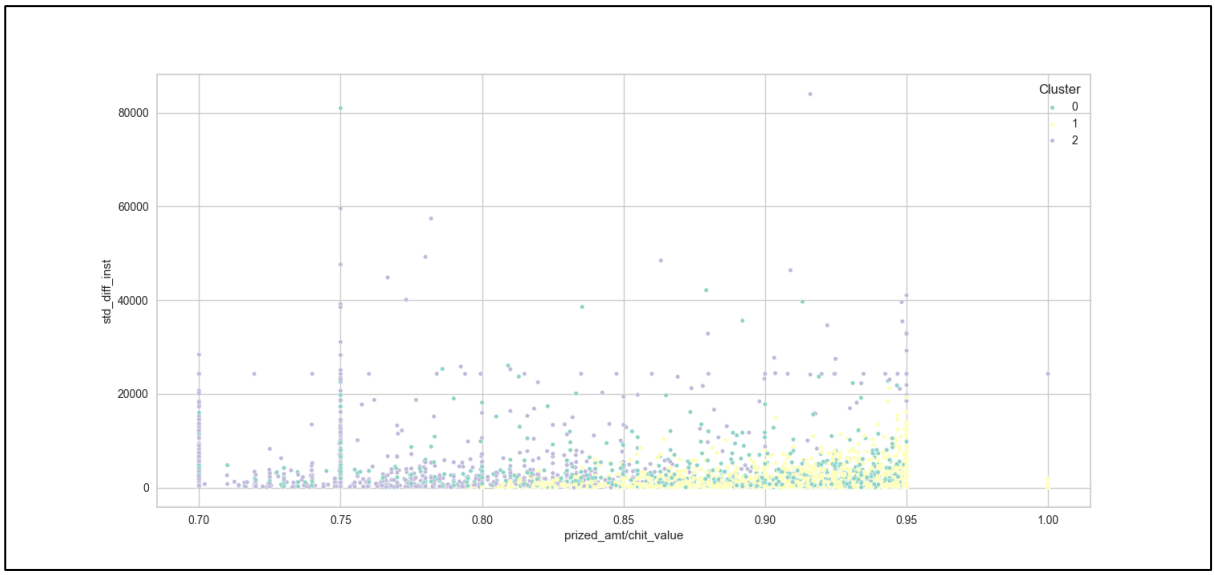

Figure 4. 4: Scatter plot between std_diff_inst and prized_amt/chit_value by cluster

4.2 Discussion

Summarize the 3 clusters:

Group 0:

- chit value (~ 134807 average)
- prized_amt/chit_value (~ 86% average)
- std_diff_inst (~ 1959 average)
- ratio_late_payment (~ 14%)
- ratio_default (~ 2.6%) và ratio_default_90 (~ 1.5%)
- From the above data, this is the member that should be evaluated in the credit score group: "**Good**"

Group 1:

- chit value (~ 94087 average)
- prized_amt/chit_value (~ 91% average)
- std_diff_inst (~ 1107 average)
- ratio_late_payment (~ 13%)
- ratio_default (~ 2.3%) và ratio_default_90 (~ 1.4%)
- From the above data, this is the member that should be evaluated in the credit score group: "**Very Good**"

Group 2:

- chit value (~ 99086 average)
- Tỷ lệ prized_amt/chit_value (~ 74% average)
- std_diff_inst (~ 2599 average)
- ratio_late_payment (~ 22%)
- ratio_default (~ 6.3%) và ratio_default_90 (~ 4.3%)
- From the above data, this is the member that should be evaluated in the credit score group: "**Fair**"

CHAPTER 5: CONCLUSION AND FUTURE DEVELOPMENT

5.1 Conclusion

The project "Classifying Credit Scores with K-Means: Insights from Chit Fund Data in India" focuses on constructing a classification model to assess the creditworthiness of participants within chit funds. This contribution is particularly significant amidst the rising popularity of various investment forms and financial mobilization efforts. By enabling capital mobilization organizations to evaluate the creditworthiness of registered members, the project addresses a crucial need in the financial landscape.

The proposed model relies on the K-means machine learning algorithm, applied to a dataset representative of real-world scenarios in terms of size and characteristics. Notably, the project has yielded several key outcomes:

Firstly, through thorough data analysis, the project elucidates the operational processes of chit funds, providing insights into member behavior based on factors such as chit value and early payment rates. This exploratory analysis serves as a foundation for subsequent model development.

Secondly, the project presents a classification model tailored to assess the credit scores of all chit fund participants, categorizing them into three distinct groups based on their creditworthiness.

This research underscores the importance of leveraging machine learning techniques to enhance the efficiency and accuracy of credit evaluation processes within chit funds, ultimately contributing to more informed decision-making and risk management within the financial sector.

5.2 Limitation

During the project implementation, certain limitations arose due to constraints in time and expertise:

Lack of Practical Application Design: The system lacks practical application capabilities, indicating a gap between the developed model and its real-world deployment. This limitation hinders the seamless integration of the model into practical scenarios.

Low Silhouette Clustering Score: The Silhouette clustering score remains low at 0.4, suggesting suboptimal cluster separation. Additionally, the uneven size of clusters indicates overlap, diminishing the effectiveness of clustering outcomes. This undermines the precision and reliability of the clustering results.

Underutilized Exploratory Data Analysis: The exploratory data analysis phase revealed numerous factors within the dataset that could potentially yield valuable insights. However, the analysis was not exhaustive, indicating missed opportunities to extract comprehensive insights from the dataset. This limitation highlights the need for more thorough exploration to fully understand the dataset's nuances and potential.

Addressing these limitations is imperative to enhance the effectiveness and practical applicability of the project outcomes, ensuring more robust clustering results and maximizing the insights gleaned from the dataset.

5.3 Future Development

Given the widespread popularity of Chit Funds as a form of financial mobilization, not only in India but also in Vietnam, there exists significant potential for further development and application of the research topic. The research team proposes several directions for future exploration:

Optimization of the Model: To enhance the model's effectiveness, the research team suggests optimizing it by prioritizing columns with substantial impacts on the classification of members' credit scores. This approach aims to boost Silhouette scores and achieve more uniform cluster sizes, thereby improving the accuracy and reliability of the classification outcomes.

Development of a Website System: To facilitate practical implementation, the research team recommends developing a website system capable of integrating the classification model.

This system would enable the classification of data based on various attribute values, offering flexibility and accessibility for users seeking to evaluate credit scores within Chit Funds.

These proposed directions for future research aim to leverage advancements in model optimization and technological integration, ultimately enhancing the utility and applicability of the research findings within the context of Chit Fund operations in both India and Vietnam.

REFERENCES

[1] "What Is a Credit Score? Definition, Factors, and Ways to Raise It." Accessed: Mar. 20, 2024. [Online]. Available: https://www.investopedia.com/terms/c/credit_score.asp

[2] "The Chit Process – How it Works - Muthoot Chits." Accessed: Mar. 20, 2024. [Online]. Available: https://muthootchits.com/chits/process-about-chits/

[3] "Elbow Method — Yellowbrick v1.5 documentation." Accessed: Mar. 20, 2024. [Online]. Available: https://www.scikit-yb.org/en/latest/api/cluster/elbow.html