Credit Risk Estimation Using Various Graphs

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Sumant Chopde

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

April, 2025

Supervisor: Sanket Tilekar

Certificate

This is to certify that this dissertation entitled 'Credit Risk Estimation Using Various Graphs' towards the partial fulfillment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Sumant Chopde at Axio Digital Pvt. Ltd under the supervision of Sanket Tilekar, Data Scientist, Axio Digital Pvt. Ltd , during the academic year 2024-2025.

Sanket Tilekar

27th March 2025

Committee:
Sanket Tilekar - Supervisor
Dr. Anindya Goswami - Expert

This thesis is dedicated to my family, without whom I would not be here.

Declaration

I hereby declare that the matter embodied in the report entitled 'Credit Risk Estimation Using Various Graphs' are the results of the work carried out by me as an intern at Axio Digital Pvt. Ltd as a student of Indian Institute of Science Education and Research, Pune, under the supervision of Sanket Tilekar, Axio Digital Pvt. Ltd and the same has not been submitted elsewhere for any other degree. Wherever others contribute, every effort is made to indicate this clearly, with due reference to the literature and acknowledgement of collaborative research and discussions.

27/03/2025

Sumant Chopde

Sanket Tilekar

27th March 2025

Acknowledgments

I am thankful to multiple people who have helped me in this year-long journey. This is a culmination of the five years of the BS-MS coursework, where I discovered myself and became the person a young Sumant would most certainly be impressed by.

First and foremost, I want to thank IISER Pune for this opportunity to work in an industrial setting. I am thankful to my supervisor, Sanket, who mentored me, helped me learn from my mistakes and gave me enough flexibility to explore multiple options throughout this work. I am thankful to Shyam for sharing his pearls of wisdom and helping me adjust to a new environment as I took my baby steps in a corporate setting. I am grateful to Bharat, Shoaib, Purva, Alok, and others at Axio for helping me with technical difficulties and making me feel accepted for my first internship at a company. I want to thank Dr Anindya Goswami, whose suggestions helped make this project more rigorous.

I want to thank the mighty internet, which has provided me with an ocean of knowledge as well as hours of entertainment and distraction. The memories made with my friends this year will be cherished all my life. Special mentions to Aditya, Akanksha, Akilan, Amogha, Gaurav, Parth, Shriyansh, Siddharth, Ujwal, and Yash made this stressful year enjoyable. Finally, no words are enough to thank my mother, father and sister, whose support throughout the year motivated me and kept me mentally healthy.

Abstract

For banks and financial companies lending to individuals, understanding the creditworthiness of new customers is essential. It is challenging for people with limited/no past credit data. Most credit risk models extract features about existing users and aggregate them to infer for new ones. There are not enough known features about such customers to be extracted satisfactorily.

Axio Digital Pvt Ltd is an NBFC lending to millions of Indians. This project uses anonymised datasets (please refer to Data Declaration) to develop various customer networks. Each node and edge has attributes reflecting various aspects of the customers and the various shared pieces of information among them, like mobile numbers, email IDs, address components, etc. These provide valuable insights into their credit behaviour.

Specifically, networks based on customer email ID and mobile numbers were developed in the first part of the project. Then, addresses from each pin code regions were clustered to derive address cluster-based features for an individual. Additionally, address unigram co-occurrence graphs were created. Using these, several features were calculated for each customer to predict their default/non-default status using binary classifiers.

The models achieved appreciable predictive power on both in-time and out-of-time test datasets. They perform comparably to standalone inference models built on the same dataset. They can provide an alternative way to determine creditworthiness, especially for customers with limited/no past credit history. Overall, this study enhances Axio's capability to make sound credit decisions, thereby fostering customer trust.

Key words: Credit risk assessment, graph features, network analysis, binary classification, unigram co-occurrence.

Contents

List of Tables

List of Figures

Data Declaration

This project uses an anonymised dataset. This dataset is representative of a particular subset of interest from Axio's customer base. Henceforth in this document, whenever a reference is made to any PII (Personally Identifiable Information) data, it implies, that PII data anonymised using Axio's proprietary algorithm.

Chapter 1

Introduction

Financial institutions, such as banks and NBFCs (Non-Banking Financial Corporations), assess the creditworthiness of new customers using past credit information linked to the PAN (Permanent Account Number). It is essential to evaluate their creditworthiness (i.e., the intention and ability to repay the borrowed credit) before approving a loan [8]. This task becomes challenging for 'thin' customers — those with limited or no past credit data. Existing models, both traditional and AI-based, used for making credit decisions, extract features about existing users and aggregate them to infer information for new customers. However, for thin customers, there are insufficient known features to be extracted effectively.

Axio (`axio.co.in`) is an NBFC that serves millions of Indians. It utilises credit bureau and banking data to underwrite customers. To enhance its credit risk prediction model for the dataset under consideration, I propose using a network based inference model in addition to a standalone one. These networks have nodes representing customers and edges representing different relationships between them, such as:

- Sharing the same telephone or mobile number,
- Sharing the same email ID,
- Sharing address components (such as the street, locality, and landmark), etc.

Each node and edge of these networks will have multiple attributes reflecting various available information. These networks can provide valuable insights into customer credit behaviour. See Figure (1.1). Underwriting thin customers using these network models would provide additional predictive power beyond the existing models by utilising the edges between the thin and thick customers. I completed the development and analysis of redacted phone

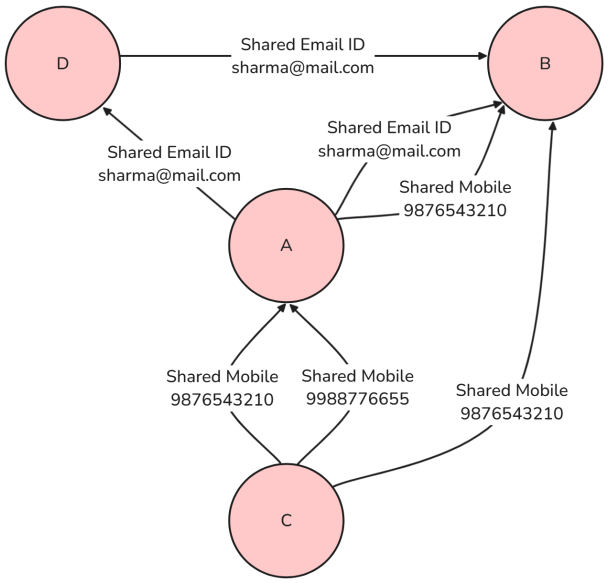


Figure 1.1: A representative customer network made using phone numbers and email IDs

number and email ID networks. These will be referred to as **Personal Identifier-based Networks (PI networks)**.

Figure 1.2: A representative clustering of customer addresses into 3 clusters for the pin code region 411008 (Screenshot taken from Google Maps)

The latter part of the project focuses on the use of customer addresses and pin codes to predict their loan performance. For this, I propose clustering all the addresses in a pin code region into three clusters, with the expectation that the different clusters correspond to different credit behaviours. If the clusters are formed successfully, one can then predict a new user's loan performance based on the cluster where they reside. Refer to Figure (1.2).

This can help us build an address cluster network.

Another proposal for addresses is to use the **address unigram co-occurrence graph**, abbreviated as **AUC graph** here onwards. Such a graph has address unigrams as its nodes, and edges are formed when two unigrams co-occur in any address. This is described in Figure (1.3). Tags such as 'house number', 'street', 'locality', 'landmark', etc., are identified for each unigram. Average default rates are attached as node and edge attributes, respectively, based on the occurrence of the unigrams and the bigrams in the defaulters' vs non-defaulters' addresses. Thus, I obtain graph-based, performance-based, and tag-based features, which have strong predictive power.

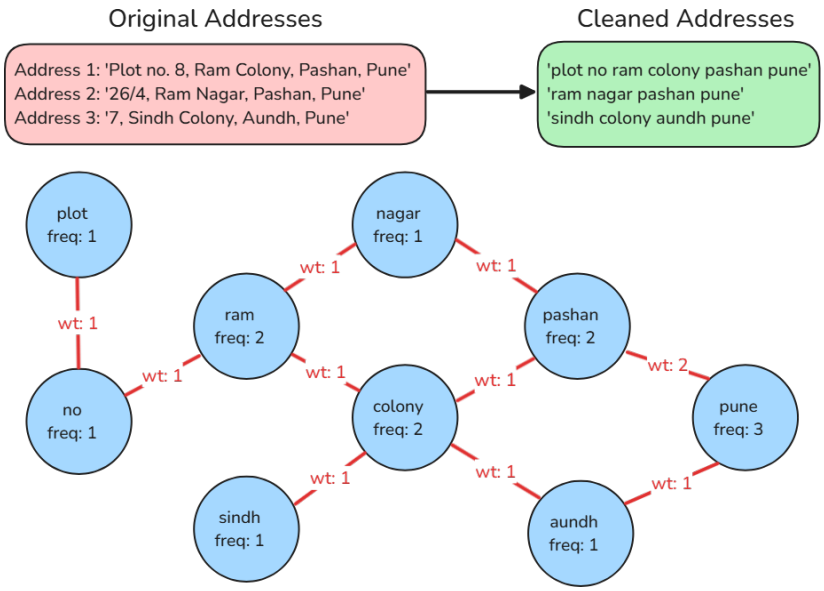


Figure 1.3: A representative address unigram co-occurrence graph for synthetic addresses

I utilised various datasets to understand the company's current customers better and make informed credit decisions for new customers. Overall, this project promises to significantly enhance Axio's capability to make sound credit decisions, thereby reducing risk and fostering customer trust.

1.1 Overview of Credit Risk Prediction Models

A credit risk model outputs a credit risk score, representing the likelihood that the borrower will default on their loan obligations. Typically ranging from 0 to 900, a higher score represents a better ability to return credit.

Majority of the credit scoring methods typically focus on payment history, credit utilisation and type, total credit accounts, the number of recent credit applications, the amount of available credit being used by the borrower, employment status, income, and so on. [8].

In India, four **credit information companies (CIC)** are licensed by Reserve Bank of India [8]:

1. Credit Information Bureau (India) Limited (CIBIL) (www.cibil.com/),
2. Experian (www.experian.in/),
3. Equifax (www.equifax.com/) and
4. CRIF High Mark (www.crifhighmark.com/)

1.2 Challenges in Credit Risk Assessment and Ethical Considerations

Credit Service Providers (CSPs) must overcome the following challenges and understand the following considerations in evaluating the credit default risk [9].

1.2.1 Bias and fairness

AI systems for credit risk prediction can perpetuate existing biases if the training data contains biased representations, leading to unfair credit decisions. Sources of bias include biased data, biased algorithms, and biased human decisions.

By using alternative data (i.e., apart from bank statements and past credit history) to assess creditworthiness, these systems have the potential to improve financial inclusion for the deserving population, which is generally not allotted credit using traditional methods. However, such algorithms may inadvertently incorporate legally and ethically restricted personal data, such as race or gender. Geolocation data can also be used to approximate ethnicity.

1.2.2 Interpretability

Some innovative algorithms are often complex and lack transparency, making it difficult to explain credit decisions to consumers and regulators. This opaqueness can increase systemic risk. Tree-based models are a suitable alternative, as they are powerful yet more explainable.

1.2.3 Accountability

Establishing measures to validate AI systems and prevent misuse is recommended to ensure accountability.

1.2.4 Data imbalance

Another significant challenge is the imbalance in datasets used for training credit risk models. The default rates of CSPs are often below 10%. Note that it is more detrimental for a financial institution to classify a defaulter customer as a non-defaulter than vice versa.

1.2.5 Feature engineering and data quality

Missing values, outliers, and irrelevant features can significantly degrade the performance of credit risk models. Preprocessing techniques, such as handling missing values and creating new features, can improve model accuracy.

1.2.6 Data privacy

There is a growing concern about how personal data is collected and used in credit decisions. Many digital credit processes are opaque, making it difficult for borrowers to understand data usage. Developing countries often lack adequate data privacy laws.

AI-based credit scoring, which relies on personal data, raises privacy concerns. Data sources such as family networks [33], social media information [11], email data [16], grocery transactions [27], etc., are utilised as well and must be monitored carefully.

1.2.7 International migration

Global operations are challenging for CSPs. Borrowers moving to new countries must rebuild their credit scores, and identity verification processes increase the risk of identity theft. The lack of international regulatory standards complicates innovation and regulation in cross-border financial activities.

1.3 An Introduction to Graph Databases and Network Analysis

1.3.1 Graph databases

Graph databases store data in the form of nodes, edges, and properties, with nodes representing entities and edges representing their relationships. They are designed to efficiently model and query complex relationships between data, which is a key advantage over traditional relational databases.

Graph databases excel in traversing and querying these interconnected data structures, allowing for quick analysis of relationships between data points. Many applications that work with interconnected data, such as social networks, knowledge graphs, recommendation systems, and biological networks, require high exploration speeds. Graph databases are dynamic and flexible, adapting to changing requirements [7].

Some popular graph database management systems include Neo4j, AWS Neptune, OrientDB, and TigerGraph.

I have used Amazon Neptune for my work, where I queried the data using the OpenCypher language. The Python library NetworkX [22] was also used.

1.3.2 Network analysis techniques

Network analysis techniques are used to study the structure and dynamics of complex networks, such as biological networks, social networks, and transportation networks. Some common network analysis techniques include node centrality measures, community detection, link prediction, network dynamics, etc.

The combination of graph databases and network analysis techniques can be particularly powerful for credit risk prediction, as it allows for the efficient storage and querying of complex, interconnected data, along with valuable insights into the structure and dynamics of the data [7].

1.3.3 Advantages of graph databases for credit risk prediction

Using graph databases and network analysis techniques for credit risk prediction offers several advantages over traditional machine learning-based methods:

- **Handling complex relationships:** Graph databases can efficiently model complex

relationships between borrowers, lenders, and credit transactions, which are often difficult to capture using traditional machine learning methods. This allows these relationships to be considered for accurate credit risk assessments.

- **Temporal and dynamic analysis:** They enable the analysis of evolving relationships and patterns over time.
- **Integration of multiple data sources:** Diverse data sources, including structured and unstructured data, can be integrated to create a comprehensive view of credit risk factors.
- **Scalability and flexibility:** Graph databases are designed to handle large-scale, complex networks, making them well-suited for the task of credit risk prediction.
- **Improved interpretability:** Graph databases provide a visual representation of the relationships between entities, enhancing transparency.
- **Enhanced semi-supervised learning:** Graph neural networks (GNNs) can leverage the structural information within graphs to perform semi-supervised learning, which is particularly useful when labelled data is scarce.
- **Multi-task learning:** Multi-task learning, which involves training the model on several related tasks at once, can be accomplished via graph neural networks.

1.4 Hindrances with Customer Addresses

Most addresses in India are unstandardised and do not follow any fixed pattern. The same address can be written in multiple different ways. There are variations due to misspellings, colloquial neighborhood names, local languages being transliterated into English, and other inconsistencies. Consider the following examples from [29] as an example:

1. 'XXX, AECS Layout, Geddalahalli, Sanjaynagar Main Road, Opp. Indian Oil Petrol Pump, Ramkrishna Layout, Bengaluru, Karnataka 560037'
2. 'XXX, B-Block, New Chandra CHS, Veera Desai Rd, Azad Nagar 2, Jeevan Nagar, Azad Nagar, Andheri West, Mumbai, Maharashtra 400102'
3. 'Gopalpur Gali XXX, Near Hanuman Temple, Vijayapura, Karnataka 586104'

4. ‘Sector 23, House Number XXX, Faridabad, Haryana 121004’

5. ‘H-XXX, Fortune Residency, Raj Nagar Extension, Ghaziabad, Uttar Pradesh 201003’

Relying solely on pin codes is also challenging. The area covered by a particular pin code varies widely, and a large fraction of written pin codes are incorrect or outdated [40]. Thus, developing a customer address network is not as straightforward as phone number or email ID networks.

Consider these two synthetic addresses:

```
‘a 14 shriram apartments jawaharlal nehru rd 1st flr maharashtra mulund west mumbai’
```

```
‘a 14 1st floor shriram apartments jawaharlal nehru rd mulund west mumbai maharashtra’
```

Although one can understand that these two addresses refer to the same place, their normalised Levenshtein distance is about 0.52. Thus, fuzzy matching is not optimal for dealing with addresses. Similarly, comparing common substrings or performing sequence alignment (like in bioinformatics) is inadequate. Sophisticated methods are required to extract the meaning and the location’s hierarchical structure.

1.4.1 Address Preprocessing

Cleaning customer-submitted addresses is crucial before performing any downstream tasks. Below are some common preprocessing steps:

1. **Setting the word case:** One can turn all words into lowercase or convert proper nouns to title case.

2. **Removing unnecessary elements:** Remove symbols, digits, excessively short or long words, phone numbers, extra spaces, pin codes, duplicate words, redundant terms, stop words, etc.

3. **Correcting misspelled words:** Correct spelling errors, e.g., ‘appartments’ → ‘apartments’.

4. **Splitting compound words:** Split compound words, e.g., ‘roadnear’ → ‘road near’.

5. **Joining mistakenly split words:** Merge words that were erroneously split, e.g., ‘bengal uru’ → ‘bengaluru’.

It is important to note that address cleaning involves some level of subjectivity. In the case of proper nouns, there is no universal standard for correctness, e.g., 'Ramnagar,' 'Raamnagar,' and 'Ramanagar.' These nuances must be considered beforehand.

1.4.2 Address Element Recognition

Recognizing elements such as house number, road, locality, landmark, etc., from Indian addresses is a more nuanced task than performing named entity recognition (NER) on standard text. The exact list of entities depends on the downstream task and whether the addresses originate from tier 1, 2, or 3 cities, or from villages. However, there is no large publicly available labeled dataset to train an NER model specifically for this task.

1.5 The Aim, Scope, and Significance of This Project

1.5.1 Aim

This study aims to develop a novel approach for assessing customer default risk using graph-based techniques. It focuses on constructing and analysing networks derived from customer addresses, mobile numbers, and email IDs. The study explores the predictive power of network-based features in predicting financial risk, without the need for traditional geospatial data.

1.5.2 Scope

1. **Phone number and email ID networks:** I constructed customer relationship networks where nodes represent a customer, and edges represent shared mobile numbers and email IDs, identified through past bureau records. This network helps in identifying the predictive power of the social circle of a person. The customer's addresses are not considered here. Also, I have not used sophisticated algorithms for graph feature representation like Node2Vec, Graph2Vec, etc., here.

2. **Clustering addresses in a particular pin code region:** I performed clustering of cleaned addresses for a particular pin code region into clusters based solely on the address text. This approximates the notion of different 'localities' in a pin code region. I hope that the different clusters exhibit distinctive loan performance with good predictive power. The goal of this exercise is to create an **address cluster network**,

where two customers have an edge if any of the addresses in their records are placed in the same cluster. I have not considered address entity recognition or geospatial data here.

3. **Address unigram co-occurrence networks:** I developed a graph based on the co-occurrence of address unigrams. The component tags, viz., house number, street, locality, landmark, post office, VTC (village/town/city), sub-district, and district, are identified. Average default rates are attached to the nodes and edges. Thus, I obtain graph-based, loan-performance-based, and tag-based features for risk prediction. Address pin codes and geospatial data are not used here. In addition, I have not used Node2Vec, Graph2Vec, etc., here.

The study does not incorporate traditional credit scoring parameters such as income or employment history. I also do not develop graph neural networks, since they are computationally expensive. Additionally, direct geospatial analysis using latitude/longitude is not included, as the focus is on the textual content of addresses.

1.5.3 Significance

This project integrates text-based address graphs and identity-based entity networks for financial risk modelling. It demonstrates the scalability and utility of using graphs for large-scale financial data. As more users are incorporated into graph formation, the number of nodes and edges increases, leading to a better understanding of customers.

Apart from default risk prediction, the AUC graph is also useful for other address-related tasks such as:

- Suggesting next words while an address is being entered,
- Suggesting the correct spelling if a person misspells a word while entering an address,
- If pin code data is also included, suggesting a pin code after an address is filled,
- Identifying points of interest in a city for on-site operations, etc.

On the practical side, the project provides a mechanism for identifying shared phone number and email ID connections among high-risk entities. The models developed here can work well for thin customers as well. It offers a scalable, cost-effective alternative to traditional credit risk assessment, especially for thin-file customers and regions with incomplete or unreliable geospatial data.

1.6 The Structure of This Document

This document is structured as follows. Chapter 2 discusses relevant background concepts and reviews the pertinent literature. Chapter 3 elaborates on the techniques used, and Chapter 4 presents the results obtained. Finally, Chapter 5 describes the conclusions of this project and suggests future research directions.

At the end, the reader can find the bibliography and the glossary of terms. The terms highlighted in blue are clickable and direct the reader to their corresponding definitions in the glossary.

Chapter 2

Background and Literature Review

2.1 Credit Risk Assessment

2.1.1 Traditional Credit Scoring Methods

The most prominent traditional techniques used to develop credit scores are:

- linear regression models,
- discriminant analysis,
- logit and probit models, and
- expert judgment-based models.

Edward Altman's 1968 model, as described in [1], remains a leading model for estimating the likelihood of a company going bankrupt within the next two years. It can also be modified to predict defaults.

A widely used expert judgment-based model, known as the Analytic Hierarchy Process (AHP), is effective in determining the creditworthiness of applicants [17].

2.1.2 AI-Based Credit Scoring Methods

Some of the supervised machine learning techniques include decision trees [3], support vector machines [3] [46], random forests [3] [46], extreme gradient boosting (XGBoost) [51], and neural networks. Various neural network (NN) architectures have been employed for this task, including MLPs [32], CNNs [14], LSTMs [42], Restricted Boltzmann Machines (RBMs)

[52], and RNNs [4]. Credit risk assessment has also been evaluated using attention-based deep learning methods and graph neural networks, as demonstrated in [48].

The unsupervised techniques for credit risk evaluation include K-means clustering [30] and hierarchical clustering [3]. Wang et al. [49] have also employed reinforcement learning along with variational autoencoders to assess consumer credit risk.

2.2 Graph Databases and Network Analysis in Credit Risk Evaluation

For credit risk evaluation, Giudici et al., in [20], have used similarity networks of borrowers to improve credit risk accuracy. Relational graph convolutional networks have been employed to classify the creditworthiness of Indian micro, small, and medium-sized enterprises (MSMEs) [31].

Graphs constructed from alternative data sources have also been used for this purpose, including family relationships, enterprise ownership, etc. [33]. Data from a user's Facebook friends [11] and supermarket grocery shopping patterns [27] have also been utilised for this purpose. Taking this further, Song et al., in [43], have used hypergraphs to address this problem. Similarly, Óskarsdóttir and Bravo have analysed multilayer networks using a modified PageRank algorithm [35].

2.3 Data Obtained from Credit Bureau Reports

Financial institutions such as banks and NBFCs are licensed by the Reserve Bank of India to obtain a person's past credit information through CICs like Experian. This process is known as a 'credit bureau pull'. The data includes the following types of information about an individual, stored alongside their PAN [38]:

- **Account details:** A comprehensive list of the consumer's loans and payment history, broken down by loan.
- **Enquiry details:** Information on previous enquiries made by the customer.
- **Personal information:** Name, date of birth, gender, addresses, telephone and mobile numbers, etc.

- **Credit score:** A score that indicates the likelihood of the individual missing payments on their loans.

This information helps financial institutions make informed credit decisions. For example, frequent changes in mobile numbers and addresses raise suspicions. I have used some of this data to build my networks.

2.4 Use of Alternative Data Sources for Credit Risk Assessment

Alternative data sources are particularly useful when a customer is considered 'thin'. For example, Yu [55] has used customer phone records to predict loan defaults. One's Facebook friendships [11] and email usage data [16] can also reveal insights into one's creditworthiness.

Tracking user transactions may raise privacy concerns, but when accessed ethically, they can provide valuable insights into a user's credit risk. For instance, credit card transactions have been analysed in [10]. Mobile money transactions are also beneficial for assessing credit risk, as demonstrated by Mhina and Labeau [30]. Debit card transactions were used to construct customer networks and predict loan defaults in [47]. Another study [56] applied multiple instance learning to analyse the comprehensive transaction history of applicants, including data from their credit cards, debit cards, passbooks, and other accounts.

2.5 Address Preprocessing

Previous work on address preprocessing has relied on simple regular expression-based techniques combined with machine learning. A large volume of data is utilised to calculate the empirical probability of unigrams and bigrams, which is then used to automatically identify misspellings (e.g., appartments'), incorrect splits (e.g., bengal uru'), and erroneous joins (e.g., 'roadnear').

The case with Indian addresses is distinct from that of developed countries due to the following reasons:

- Use of a multitude of local words instead of standard terminology
- Transliteration of regional languages into English along with code-mixing
- Incorrect grammar, punctuation, or formatting

- Use of landmarks that no longer exist or are located far from the target location (The average distance between the landmark and the doorstep is around 400 m) [40]

- Inclusion of irrelevant information, such as phone numbers, timings of availability, and directions to reach the location

A word may be spelt differently due to reasons such as convenience, abbreviations, typographical errors, and the user's limited literacy. The sub-regions of an Indian city or town can take highly irregular shapes depending on population density, road networks, and other factors [29].

Research on address preprocessing has been conducted by companies like Flipkart Pvt Ltd [5] and other e-commerce firms, as they possess large datasets of customer addresses. Such preprocessing techniques have been applied by companies such as Flipkart, Delhivery Ltd [25], Myntra Designs Pvt Ltd [29], and Swiggy Pvt Ltd [19].

Recognising address elements has been performed using various neural networks, such as MLP [41], one-dimensional CNN [13], bidirectional GRU NN [28], and the BERT model [57]. However, no publicly available research exists for Indian addresses, as such work is typically conducted by companies on private datasets that are not shared publicly.

2.6 Unigram Co-occurrence Graphs

Unigram co-occurrence networks have nodes representing unigrams and edges denoting the co-occurrence of words in a text document. They have been used for tasks such as single-label [54] and multi-label text classification [53], as well as identifying relationships between legal texts [45]. However, no work has been observed that uses unigrams from addresses to create a co-occurrence network.

2.7 Gradient Boosted Decision Trees as Binary Classifiers

Gradient Boosted Decision Trees (GBDTs) are a powerful class of ensemble learning algorithms that build a strong predictive model by sequentially training decision trees, where each tree corrects the errors of its predecessors. GBDTs leverage gradient-based optimisation techniques to minimise the loss function, making them highly effective for classification [18].

In this work, LightGBM (Light Gradient Boosting Machine), an optimised implementation of GBDT, was employed [26]. Multiple studies have shown that GBDTs outperform simpler classifiers, such as logistic regression, support vector machines, decision trees, and random forests, and yield similar evaluation results as neural networks.

2.8 Evaluation Metrics for Credit Risk Assessment

Most ML models for this task in the literature are binary classifiers. Various metrics have been used for their evaluation, like accuracy [3] [52], precision [46] [27], recall [27] [46], F1-score [46] [31] [27], area under the precision-recall curve [20], etc.

The most commonly used metrics are the ROC-AUC (area under the receiver operating characteristic curve) [51] [32] [42] [4] [48] [20] [31] [33] [11] [27] [43] [16] [47] [56] and the KS (Kolmogorov-Smirnov) statistic [42], [48], [33], [43]. I have used these for evaluating my models.

2.8.1 The ROC-AUC Score

This section has information referenced from the user guide of the `scikit-learn` library [37].

The ROC-AUC score is widely used to evaluate the discriminatory power of binary classifiers. It is defined as the area under the receiver operating characteristic (ROC) curve.

Consider a trained binary classifier that predicts test samples as positive or negative. The **sensitivity** is the ratio of correctly predicted positives to the total actual positives. Conversely, **specificity** is the ratio of correctly predicted negatives to the total actual negatives.

The ROC curve plots sensitivity against $1-$ specificity at various thresholds. A classifier that predicts randomly has an area under the curve (AUC) of 0.5, while a perfect classifier achieves an AUC of 1.0. Thus, a useful model has an ROC-AUC score greater than 0.5 [12].

I have used ROC-AUC as the primary metric for comparing classifiers due to the following advantages:

- Classifiers typically output class probabilities rather than labels, requiring a threshold to determine labels. Metrics like accuracy, precision, and F1-score depend on this threshold, whereas ROC-AUC does not.

- It facilitates comparison across multiple models trained during the project, and to the ones mentioned in the literature.

I computed the ROC-AUC scores using the `scikit-learn` library [37].

2.8.2 The KS Statistic

The Kolmogorov-Smirnov (KS) statistic is a non-parametric measure used to evaluate the discriminatory power of a binary classifier. It quantifies the maximum separation between the cumulative distribution functions (CDFs) of the positive (event) and negative (non-event) classes. [24]

Given a set of predicted probabilities, the CDFs of the positive and negative classes are defined as:

$$F_1(x) = \frac{1}{N_1} \sum_{i=1}^{N_1} \mathbf{1}\{p_i \leq x \mid y_i = 1\}$$

$$F_0(x) = \frac{1}{N_0} \sum_{i=1}^{N_0} \mathbf{1}\{p_i \leq x \mid y_i = 0\}$$

where:

- $F_1(x)$ is the CDF of the positive (event) class.
- $F_0(x)$ is the CDF of the negative (non-event) class.
- $\mathbf{1}$ is the indicator function.
- p_i is the predicted probability for the i-th observation.
- N_1 and N_0 are the total number of positive and negative observations, respectively.

The KS statistic is then computed as the maximum absolute difference between these two CDFs:

$$KS = \max_x |F_1(x) - F_0(x)| \tag{2.1}$$

The KS statistic ranges from 0 to 1, where a value close to 0 indicates poor discrimination between the classes. The value close to 1 suggests excellent model performance, with higher separation between positive and negative classes.

2.9 Weight of Evidence and Information Value in Feature Selection

Weight of Evidence (WoE) and Information Value (IV) are statistical tools used to evaluate the predictive power of independent variables, particularly in binary classification problems. I applied them in the context of credit default prediction.

2.9.1 Weight of Evidence (WoE)

WoE quantifies the strength of a feature in distinguishing between two classes (e.g., defaulters and non-defaulters) [2]. It is calculated for each bin of a feature as:

$$\text{WoE} = \ln\left(\frac{\frac{\text{Count of defaulters in bin}}{\text{Total defaulters}}}{\frac{\text{Count of non-defaulters in bin}}{\text{Total non-defaulters}}}\right)$$

A positive WoE suggests a higher proportion of defaulters, while a negative WoE indicates the opposite. WoE creates a linear relationship between the predictor and the log odds of the target, benefiting binary classification models.

2.9.2 Information Value (IV)

IV aggregates WoE across bins to quantify a feature's overall predictive power. It is computed as:

$$\text{IV} = \sum_i \left[\left(\frac{\text{Defaulters in bin } i}{\text{Total defaulters}} - \frac{\text{Non-defaulters in bin } i}{\text{Total non-defaulters}}\right) \times \text{WoE}_i\right]$$

Interpretation of IV Values

IV values guide feature selection based on predictive strength:

$$\begin{aligned}
\text{IV} < 0.02 &: \text{Not predictive} \\
0.02 \leq \text{IV} < 0.1 &: \text{Weak predictive power} \\
0.1 \leq \text{IV} < 0.3 &: \text{Medium predictive power} \\
0.3 \leq \text{IV} < 0.5 &: \text{Strong predictive power} \\
\text{IV} \geq 0.5 &: \text{Suspicious or overfitting}
\end{aligned}$$

These thresholds aid in determining which variables to retain for improving model performance.

2.10 Performing Out-of-Time Testing

Evaluating machine learning models requires assessing their generalizability and robustness under real-world conditions. **In-time testing** often fails to capture a model's ability to generalise to unseen future data. This limitation is especially critical in financial domains, where loan default rates are influenced by evolving factors such as economic conditions, regulatory policies, interest rates, and seasonal variations.

Out-of-time (OOT) testing mitigates this concern by evaluating model performance on data from a different, later time period than the training data. This approach simulates real-world scenarios and ensures that the model's predictive power remains stable over time. Given the temporal dependencies in credit risk modeling, OOT testing helps prevent reliance on patterns that may not persist in future periods.

To ensure reliable credit decisioning, I incorporated OOT testing in addition to in-time testing for all models [23].

Chapter 3

Methodology

3.1 The Target Variable

The objective of this project is to identify applicants who are most likely to default on a loan. The exact definition of `default` is variable, depending on the number of days after the EMI due date for which the debtor has not paid their EMI. This number will be referred to as **DPD** hereafter. For example, in fintech terminology, a person is said to have gone 'DPD84' if 84 days have passed after their due date without any payment of dues. Various levels of defaults are generally considered, such as DPD25, DPD54, DPD84, DPD180, etc., and different actions are taken against defaulters by the lender.

Definition 3.1.1. *The maximum DPD value of a person within 8 months of application is termed their* ***maximum bad rate****.*

Very low default rates in a population are challenging to predict. A window of 8 months provides the population sufficient time to produce an appreciable default rate. Using an indicator variable transforms this into a classification problem, referred to as the **performance bad rate**. The value of 25 was chosen as it is the smallest DPD value that makes practical sense on a monthly basis.

Definition 3.1.2. ***The performance bad rate:***

$$bad\_rate\_perf = \begin{cases} 1, & \text{if } maximum\_bad\_rate > 25 \\ 0, & \text{if } maximum\_bad\_rate \leq 25 \end{cases}$$

This is the target variable to be predicted by binary classifiers.

3.2 The Input Data

The organisation has provided anonymised datasets for analysis. This dataset has information about a particular subset of customers of interest to the organisation. It comprises the following components:

1. **UCID:** A string that serves as a unique identifier for each customer across different tables.

2. **Data from a Person's Bureau Records:**

   (a) **Phone Numbers:** Dummy entries were removed by ensuring that each phone number consists only of digits, is of length 10, and starts with either 6, 7, 8, or 9. Since the customers are from India, the country code +91 is ignored. These phone numbers are hashed to safeguard customer privacy.

   (b) **Email IDs:** Dummy entries were removed by ensuring that each email ID contains the symbol @. Entries such as *null@null.com*, *nomail@notavailable.com*, etc., were discarded. Similar to phone numbers, these are hashed to protect customer privacy.

   (c) **Customer Properties:** Attributes such as the customer's application date, bureau score, etc.

   (d) **Bureau Addresses:** These addresses are sourced from a person's bureau records. The digits and ordinals in the addresses are removed to ensure that customers are not personally identifiable. The pin codes are retained as they are essential for downstream tasks. These addresses were not hashed.

3. **KYC Addresses:** These addresses are sourced from the KYC (Know Your Customer) procedure. Similar to the bureau addresses, the digits and ordinals in the addresses are removed, but the pin codes are retained. Since the KYC addresses are sourced from identification documents, each customer has a unique KYC address. These addresses were not hashed.

4. **Performance Bad Rate (*bad_rate_perf*)**

3.3 Personal Identifier based Networks

The configuration of AWS Neptune allows for the creation of different types of edges and nodes, enabling the execution of specific queries. Therefore, if one intends to create nodes where the nodes represent UCIDs, and the edges represent shared phone numbers or email IDs, it is logical to load one type of node and different types of edges. This is precisely the approach I have adopted.

3.3.1 Phone number network

A directed network was developed in Neptune where the nodes are represented by UCIDs, as depicted in Figure (3.1). An edge exists between two nodes if a common mobile number is present in the bureau records of the two individuals. The direction is determined by the application dates — from the person who applied later to the person who applied earlier. Considering the application dates is essential to prevent the model from utilizing future data to predict past outcomes.

This graph is configured to exclude self-loops and nodes with a degree of zero. Multiple edges can exist between two nodes if multiple mobile numbers are shared between the two customers.

Even when the number of records is in the millions, the resulting graph has a relatively small number of nodes and edges due to the following reasons:

- To incorporate only those individuals who have been approved for credit by Axio and possess a non-null value of *bad_rate_perf*
- to incorporate only customers who have at least one neighbor in the graph

Node and edge attributes

The attributes of the nodes are as follows:

1. The application date.
2. The customer's bureau score at the time of application.
3. An indicator variable for whether the customer's maximum default rate on any of their loans from other lenders exceeds DPD 25 (*bad_rate_feat*).

4. The highest past credit limit (*max_credit_limit_ever*) allotted to the applicant by other lenders.

5. An indicator variable specifying whether the person currently uses or has ever used a credit card (*cc_flag*).

The only attribute on the edge is the shared phone number. These particular attributes were chosen because they are readily available for both thick and thin customers and remain stable over time.

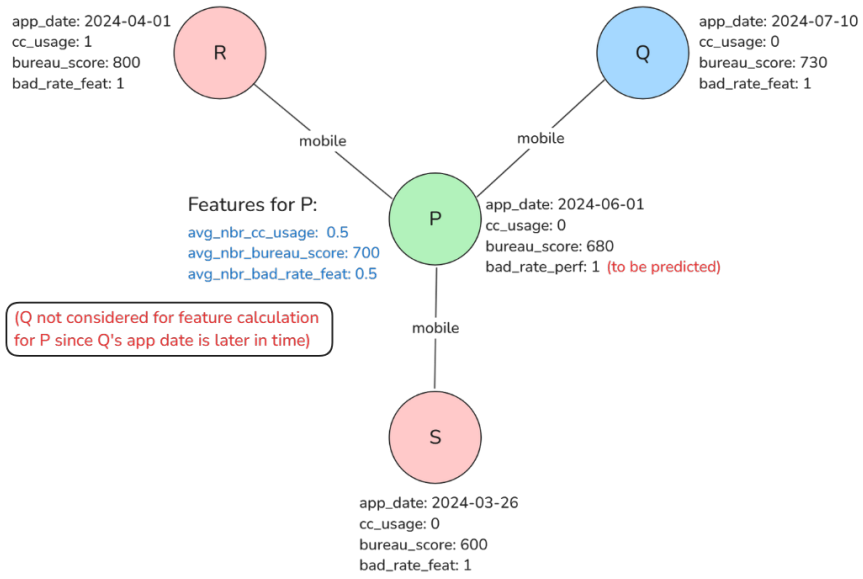


Figure 3.1: Calculating features for a person with three neighbours in mobile number network (the values are representative)

Features used to predict the *bad_rate_perf*

For each node n, I calculated the average of attributes 2 to 5 of its out-neighbours (neighbours connected with an outward-directed edge). A node's 'phone number degree' and 'email ID degree' are also used as features. See Figure (3.1). Although features involving neighbours of neighbours were devised, they were not utilised due to insufficient resources. These averaged features are used to predict *bad_rate_perf* using GBDT classifiers. Refer to figure (3.2).

3.3.2 Email ID network

Similar to the network described in Section (3.3.1), I developed a directed network where the nodes are the same as before (defined by UCID). An edge exists between two nodes if a common email ID is present in the bureau records of both individuals. The direction is again determined by the application dates. This graph is configured to exclude self-loops and nodes with a degree of zero. Multiple edges can exist between two nodes if multiple email IDs are shared between the two customers.

Node and edge attributes

The attributes of the nodes are identical to those used in the phone number network. The only attribute on the edge is the shared email ID.

Features used to predict the *bad_rate_perf*

The same features as in the phone number network are used to analyze this network.

3.3.3 Selecting features for training classifiers

After computation, each feature is divided into 5 roughly equal-sized bins. The Information Value (IV) is then calculated (see Section 2.9), and only the features with an IV between 0.02 and 1.00 are retained. Subsequently, features with a Spearman correlation [44] of less than 0.8 are selected for training the GBDT classifiers. These values were chosen from rules of thumb. Optimal model parameters are determined using randomised search cross-validation through the Scikit-learn library [37]. This process was performed three times:

1. Using only the phone number-based features.
2. Using only the email ID-based features.
3. Using both sets of features.

It is also important to note that classification using this method is possible only for customers who already have connections present in the network. Therefore, customer coverage is defined as follows:

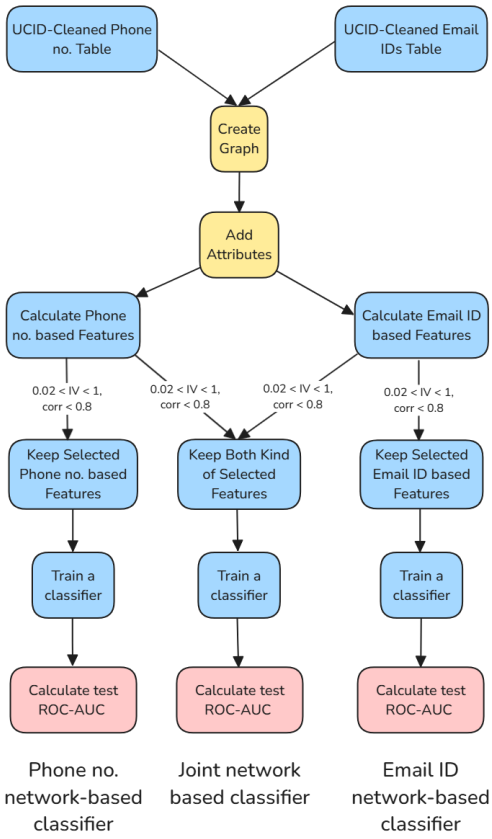


Figure 3.2: The pipeline for developing the PI network-based classifiers

Definition 3.3.1. *The **Customer Coverage** of the PI network model for a particular time period is defined as:*

$$Coverage = \frac{Number\ of\ customers\ for\ whom\ PI\text{-}network\ classifiers\ can\ generate\ results}{Total\ number\ of\ customers\ for\ whom\ bad\_rate\_perf\ is\ available}$$

Higher values of coverage and higher ROC-AUC scores for the classifiers are desirable.

3.3.4 Out-of-time testing

It is essential to perform **'out-of-time' testing**. See Figure (3.3). A classifier is considered robust if it performs well during out-of-time testing.

After conducting network analysis and experimenting with AWS Neptune, I opted to use NetworkX [22] for out-of-time testing to standardise the procedure for proper evaluation and

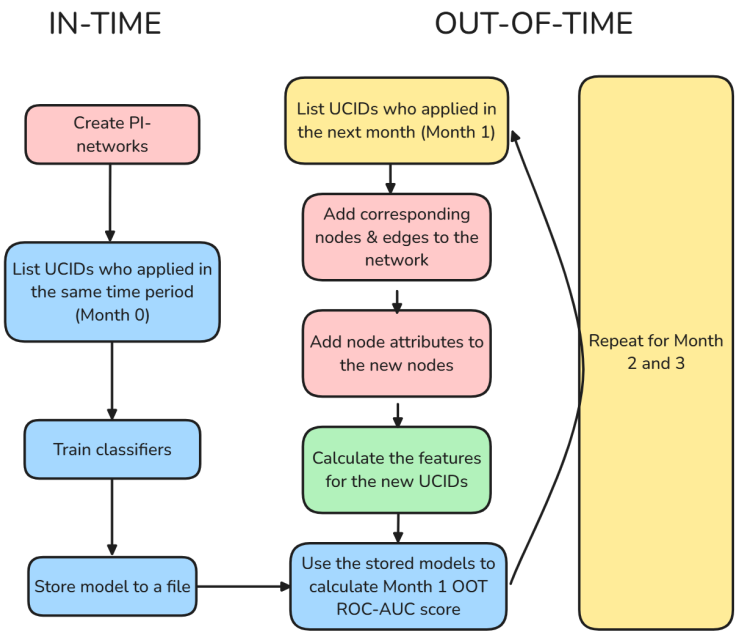


Figure 3.3: The out-of-time testing pipeline for PI networks

cost calculation.

Two different PI networks were created:

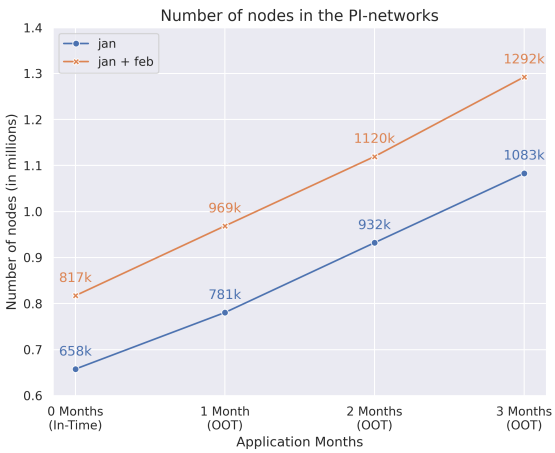
1. One constructed using January 2023 data, with connections to the 2020–2022 data. This will be referred to as the **jan-network**.

2. Another constructed using January and February 2023 data, with connections to the 2020–2022 data. This will be referred to as the **jan-feb-network**.

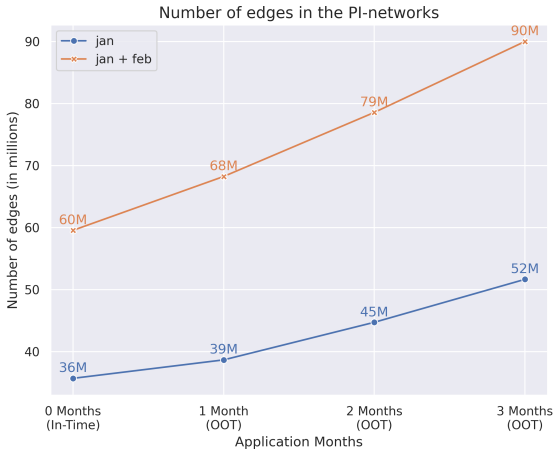
Both networks were subsequently updated three times, incorporating data from the following three months one by one for out-of-time testing. New nodes and edges were added each time a new month's dataset was included. As expected, the number of nodes and edges increased over time by approximately 17% and 13% per month, respectively. See Figure (3.4).

3.4 Pin code-wise Address Clustering

Within a particular pin code region, different localities may exhibit varying loan performance patterns. I aim to capture these variations using only the address text, without relying on geospatial data or satellite imagery. The goal of this exercise is to create an



(a) The number of nodes increases with time



(b) The number of edges increases with time

Figure 3.4: The network size increases with time

'address cluster' network, where two UCIDs have an edge if any of the addresses associated with the customer fall into the same cluster.

I used bureau reports to obtain multiple addresses for each customer, as these reports typically provide more than one address per individual. This approach offers a broader perspective on a person across their various addresses. In contrast, KYC records usually contain only one address per person. See Figure (3.5).

For each pin code under consideration, a separate clustering model is trained. Subsequently, for a test address, the cluster to which it belongs is predicted. Based on the identified cluster, features are generated for the address. These features are then aggregated for all addresses associated with a person and are ultimately utilised to train classifiers. Refer to Figure (3.6) for an overview of the entire pipeline.

3.4.1 Address Preprocessing and Embedding Generation

The following preprocessing steps were performed for each address:

1. Retain addresses containing more than three words.
2. Convert all words to lowercase.
3. Remove multiple spaces, phone numbers, PINs, and duplicate words.
4. Remove state and union territory names using normalised Levenshtein distance [6], as they are redundant for clustering.

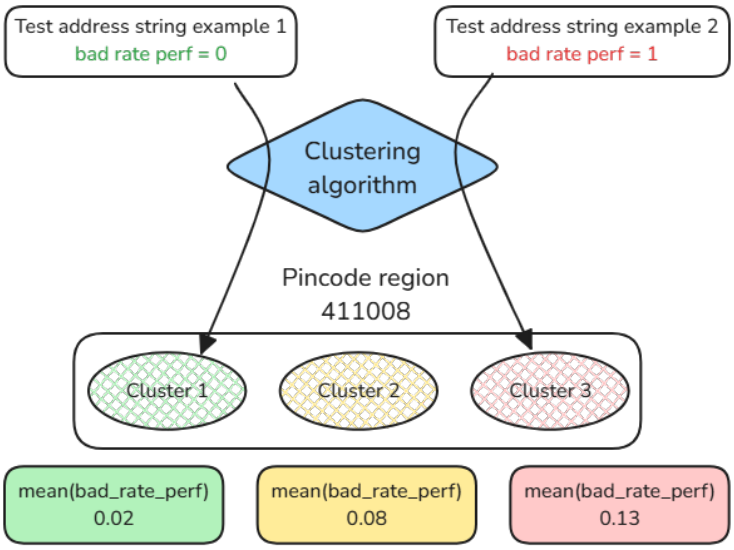


Figure 3.5: A representative example of clustering the addresses in a particular pin code region into clusters of different loan performances (the values are synthetic)

5. Retain valid pin codes by referring to the pin code dataset available on the Open Government Data platform, maintained by the Government of India [34].

6. For each UCID and pin code, retain the longest address string, as it is likely to contain the most information about the address location.

Only those pin codes that contain more than 50 distinct addresses are retained to ensure that there are sufficient addresses in each pin code for the clustering algorithm to learn effectively.

To enable the clustering algorithm to process the addresses, they are converted into embeddings. Using one of HuggingFace's (`huggingface.co`) multilingual LLM models [39], specifically `paraphrase-multilingual-MiniLM-L12-v2`, I have generated 384-dimensional embeddings. This model was chosen because it is a lightweight model capable of understanding English, Hindi, Gujarati, Marathi, Urdu, and other Indian languages, enabling a richer representation of the addresses.

3.4.2 Clustering Addresses

In addition to k-means clustering, I experimented with various clustering algorithms, including bisecting k-means clustering, DBSCAN, HDBSCAN, spectral clustering, and agglomerative clustering. Additionally, dimensionality reduction from 384 to 128 dimensions

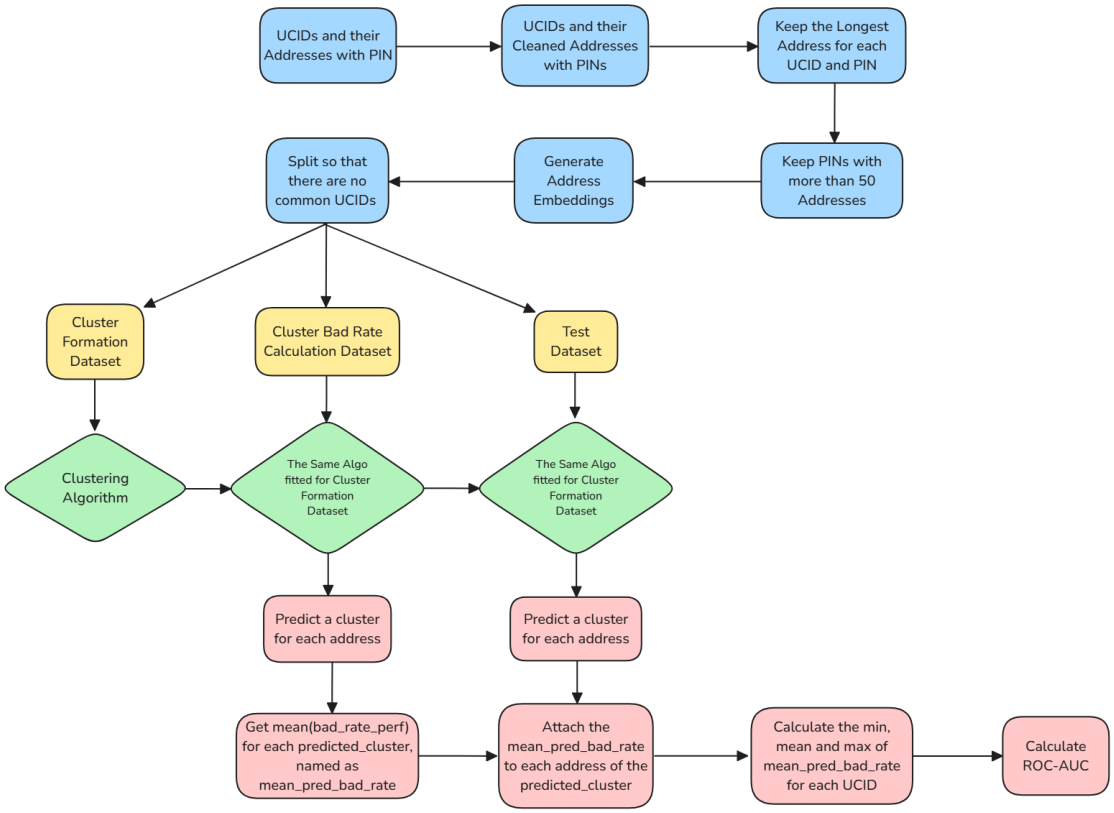


Figure 3.6: The address clustering pipeline

using PCA (Principal Component Analysis) was explored for each of these algorithms.

It was observed that HDBSCAN without compression performed the best in terms of silhouette scores. However, I chose to proceed with k-means clustering, with k set to 3, due to the following reasons:

- All hierarchical clustering algorithms require a large amount of computational resources.
- I aimed to avoid marking any address as noise, but several of these algorithms identified some addresses as noise.
- These algorithms generate a different number of clusters for different pin regions, which hinders comparisons across pin codes.

The addresses are then randomly divided into three sets. This split is performed carefully to ensure that there are no common UCIDs in the splits, while ensuring that the same pin codes are present in all three.

1. **Cluster Formation Dataset (50% of the total):** These addresses are used to train the clustering algorithm. A separate model is trained for each pin code region, and the parameters are stored in a file for subsequent steps.

2. **Dataset for Calculating Cluster Bad Rate (40% of the total):** For each address in this dataset, clustering is performed using the previously trained models. A *bad_rate_perf* value is available for each customer. The mean of these values is calculated for each cluster, referred to as ***mean_pred_bad_rate***.

3. **Test Dataset (10% of the total):** For a given address in this dataset, a cluster is predicted using the same fitted model, and the corresponding cluster's *mean_pred_bad_rate* is assigned to the address.

3.4.3 Calculating Features

There are multiple addresses for each UCID. For a given UCID, I calculate the following features for all clusters the UCID's addresses have been predicted in.

1. **The minimum** of the *mean_pred_bad_rate*

2. **The mean** of the *mean_pred_bad_rate*

3. **The maximum** of the *mean_pred_bad_rate*

These three features are used to predict the UCID's *bad_rate_perf*. Since only three features are used, IV calculation and feature selection are skipped. I calculate the ROC-AUC of these features with *bad_rate_perf*. If the results are satisfactory, classifiers are trained.

3.4.4 Evaluating if Clustering is Necessary at All

To evaluate whether clustering improves results over using only pin codes, the same process was conducted without clustering. The steps were:

1. The quantity ***mean_pin_bad_rate*** was calculated by taking the mean of the train addresses' *bad_rate_perf* for each pin code.

2. Similar to clustering, for a given UCID, I take the minimum, mean, and maximum of *mean_pin_bad_rate* for all pin codes associated with the UCID's addresses. These features are used to predict the UCID's *bad_rate_perf*.

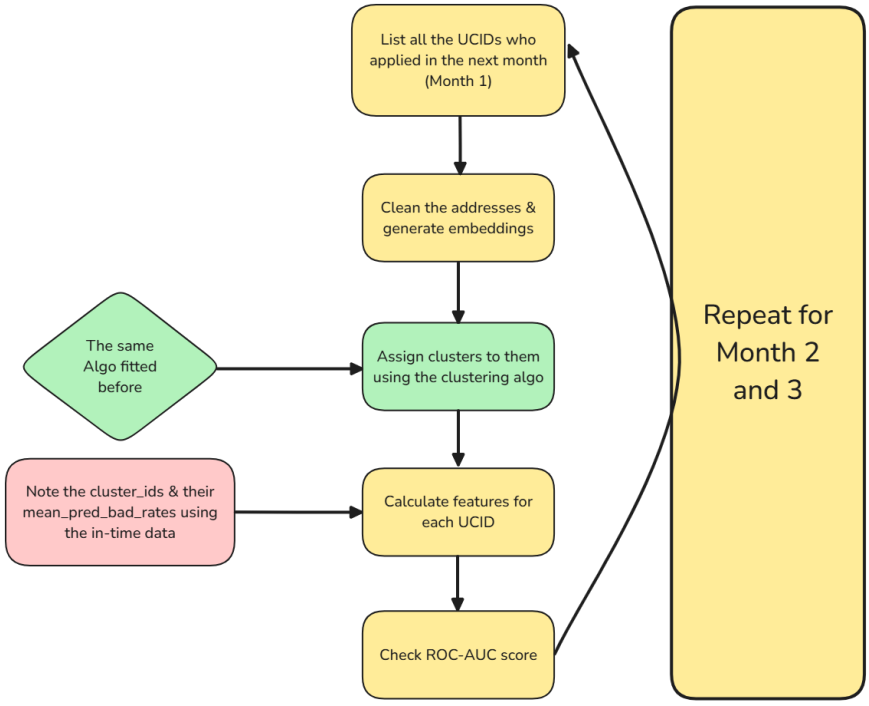


Figure 3.7: The out-of-time testing pipeline for address clustering

3. If these features' predictive power is worse than that of the clustering-based features, this implies that clustering enhances the prediction of *bad_rate_perf*.

3.4.5 Out-of-Time Testing

Refer to Figure (3.7).

3.5 Address Unigram Co-occurrence Graph for Risk Prediction

The intuition behind this graph is to score locations in a region based on the loan performance of people living nearby. For example, if the mean *bad_rate_perf* of customers living near a particular mall or school is higher than usual, this model can identify the pattern and predict a higher default probability for addresses mentioning that landmark. See Figure (3.8) for the model pipeline.

3.5.1 Address Preprocessing

With assistance from my colleagues at Axio, I performed the following preprocessing steps on the addresses:

1. Retain addresses longer than 3 words.

2. Extract pin codes from addresses if present and populate them in the pin code column if missing.

3. Validate pin codes using the Indian government's official pin code dataset in [34].

4. Convert all words to lowercase.

5. Remove special characters, pin codes, repeated words, multiple commas, extra spaces, and spaces around hyphens.

6. Remove state/union territory names using normalised Levenshtein distance [6], as they are not relevant for credit risk prediction.

7. Merge adjacent tokens if the empirical probability of the compound token is higher than that of individual tokens.

8. Split a token into two using the following method:

   (a) Form a list of unique bigrams occurring across all addresses in the corpus.

   (b) Identify the bigram with the highest normalised Levenshtein similarity to the token.

   (c) Compute the normalised similarity between the bigram and the token based on the longest common subsequence.

   (d) If either similarity exceeds pre-defined thresholds, split the token into the bigram.

Misspelling correction could not be performed as it requires the leader clustering algorithm described in [5], which is resource-intensive.

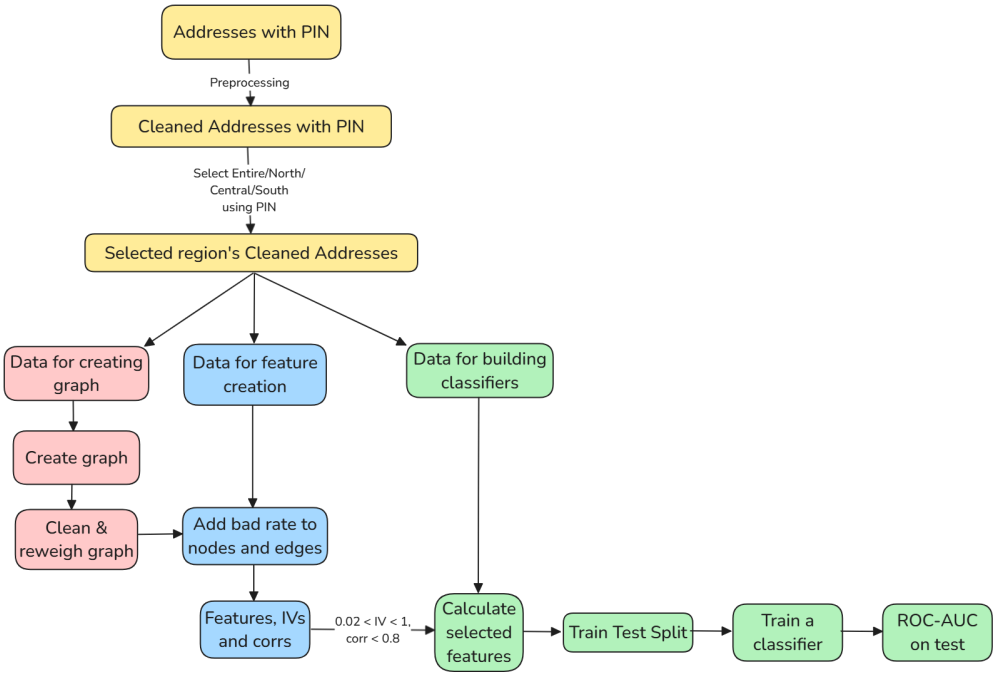


Figure 3.8: The address unigram co-occurrence graph pipeline

3.5.2 Entity Tags Assignment

The following component tags are identified for each unigram in the cleaned addresses, with assistance from my colleagues at Axio. This list was chosen because, during KYC at Axio, these tags are automatically assigned. However, many columns contain null and empty values.

1. **Care of:** Refers to the name of a resident of the house, often indicated by c/o (care of), s/o (son of), d/o (daughter of), w/o (wife of), etc.
2. **House number:** Includes unigrams related to the flat, wing, apartment, society, or colony name.
3. **Road/street name.**
4. **Locality:** Covers phases, sectors, nagars, and other subdivisions of a district.
5. **Landmark:** Indian addresses often mention landmarks such as banks, temples, schools, colleges, squares, and bridges.
6. **Post office:** More relevant for rural and suburban addresses.

7. **VTC (village/town/city).**

8. **Sub-district.**

9. **District.**

All words tagged as **Care of**, which do not contribute to credit risk prediction, were removed. Due to the high presence of null and empty values, an automatic tagging mechanism was required.

Initially, I trained a BERT [15] model on the cleaned addresses and the labelled dataset using Hugging Face [21] and PyTorch [36]. However, its performance was suboptimal (F1 score: 0.608, Accuracy: 0.755). Later, a colleague at Axio improved this model significantly, and it was used to tag addresses with missing labels.

Note that the same word may appear under different tags in different addresses, e.g., 'Gandhi Colony' and 'Gandhi Nagar.' To account for this, a dictionary was created for each unigram, assigning tags as keys and the frequency of occurrence under that tag across all addresses as values. This is referred to as the 'tag profile' of the unigram, with the values normalised to sum to 1. For example, the tag profile for 'bangalore' (the former name of Bengaluru) looks like:

$$\begin{aligned}
\text{house_number} &= 5.7422 \times 10^{-5} \\
\text{street} &= 0.0014 \\
\text{landmark} &= 0.0016 \\
\text{locality} &= 0.0044 \\
\text{vtc} &= 0.5169 \\
\text{post_office} &= 0.0014 \\
\text{sub_district} &= 0.1037 \\
\text{district} &= 0.3706
\end{aligned}$$

Ideally, the highest values in the tag profile should appear under *vtc*, *sub-district*, and *district*, with zero for other tags. However, small values are often observed for other tags due to both non-standard address formats and occasional misassignments by the NER model. Based on this dictionary, a 'most common tag' is also assigned to each unigram.

3.5.3 Creating the Graph

There are two approaches to create the edges of the graph:

1. **Using unigram co-occurrence in addresses:**
    - For every bigram in an address, if no edge exists between the nodes representing those two words, an edge is added with weight 1.
    - If an edge already exists, the weight is incremented by 1.
2. **Using a random walk on an address:**
    - The parameters *number_of_walks* and *walk_length* are set.
    - Given a cleaned address, a random walk of length *walk_length* is performed over the unigrams with a uniform probability distribution.
    - Edges are added between adjacent unigrams encountered in the random walk.
    - This process is repeated *number_of_walks* times.

For example, a random walk on the synthetic address: `'plot no 42 sindh society aundh pune city near ncl campus'` with *walk_length* = 10 can produce an output such as: `'aundh, plot, pune, sindh, pune, plot, campus, west, society, ncl'`.

The second method generates a much denser graph with higher edge weights. While this approach produces similar results, the resulting graph occupies more disk space and takes longer to traverse and generate features. However, this method is more useful for downstream tasks where the order of words in the address does not matter. Since I worked with KYC addresses, which have well-ordered elements, I opted for the first method. The resulting network is similar to the illustration in Figure (3.9).

3.5.4 Cleaning the Graph

After removing self-loops and edgeless nodes, a large number of high-weight edges remained, which skewed the distribution. To address this, three methods were employed:

1. **Triangle-based Cleaning:**
    (a) Nodes with a degree above a certain threshold were identified.
    (b) All triangles formed between these high-degree nodes were considered.

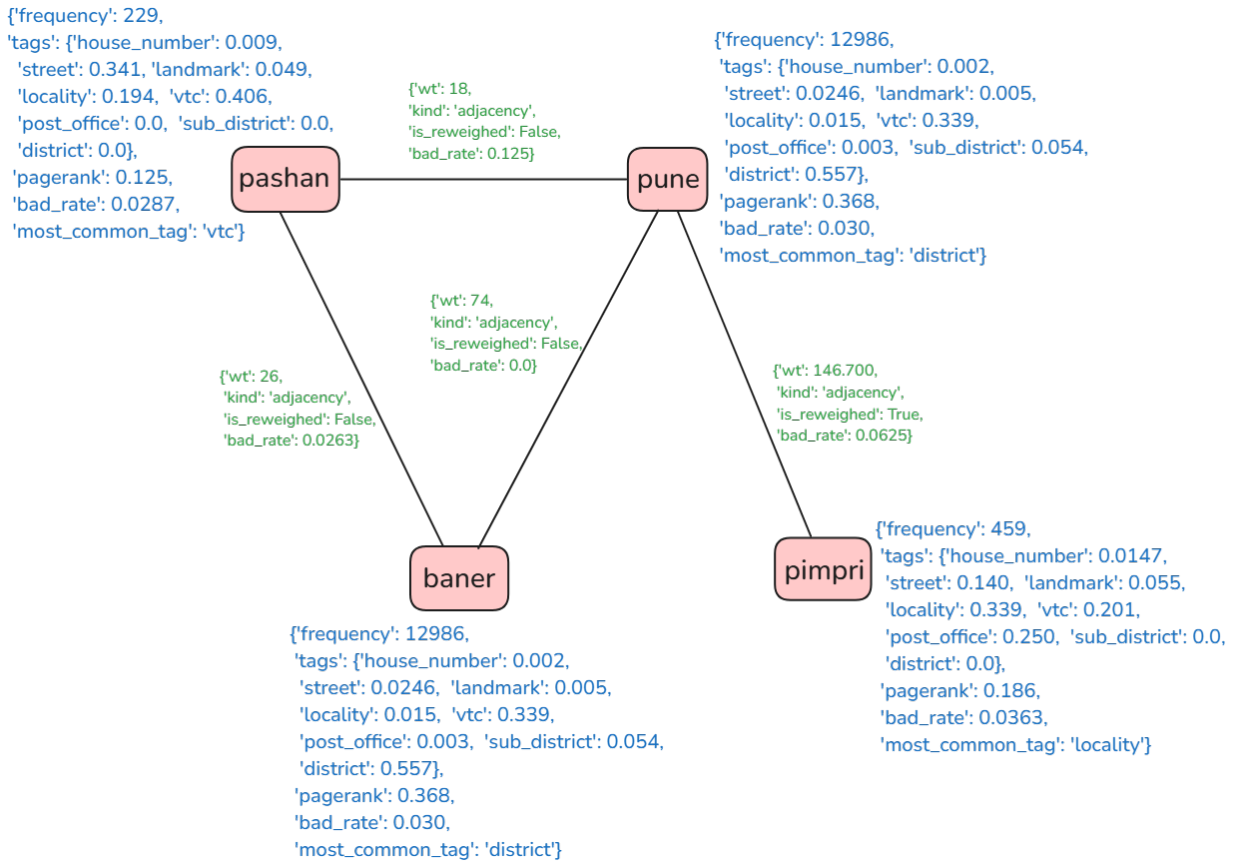


Figure 3.9: A representative example of the address unigram co-occurrence graph with node and edge attributes

(c) For each such triangle, the edge with the highest weight was removed.

2. **Density-based Cleaning:** This method is similar to the triangle-based approach.

   (a) The clustering coefficient was calculated for every node in the graph.

   (b) If the coefficient was above a certain threshold, edges incident to the node were removed if their weight exceeded a predefined limit.

3. **Edge Reweighing Using Node Frequencies:** This method was applied only when the edge weight exceeded a pre-defined threshold. The intuition behind this approach is that edges with high weights typically occur between high-frequency nodes, and their weights can be scaled down using these frequencies. To avoid drastic reweighing, the weight was adjusted using the average of the logarithm of the ratios of N and the unigram frequencies, where N is the maximum unigram frequency, scaled down by a

factor of .

$$w_{ij}^{\text{new}} = w_{ij} \times \min\left(1, \frac{\log\left(\frac{N}{f_i}\right) + \log\left(\frac{N}{f_j}\right)}{2}\right)$$

3.5.5 Generating Features

An average default rate was assigned to all nodes based on their occurrence in the addresses of defaulters and non-defaulters. Similarly, an average default rate was assigned to each edge based on the occurrence of the bigram formed by the edge's endpoints. Several additional features were considered, such as the graphlet degree vector and features involving neighbours of neighbours of a unigram. However, these features were not computed due to limited resources.

The following features were generated for each address.

1. The weighted mean of the **frequency/pagerank/node default rate** over the address unigrams

2. The weighted mean of the **edge default rate** over the address bigrams

3. The weighted mean of the **weighted degree centrality** or the **clustering coefficient** over all the unigrams

4. Bigram common neighbour default rate: (Figure 3.10)

   (a) For every address bigram and weight types $wt1$ and $wt2$, calculate the weighted mean of the node/edge default rate of their common neighbours using weights $wt1$.

   (b) Take the weighted mean over all the bigrams using $wt2$.

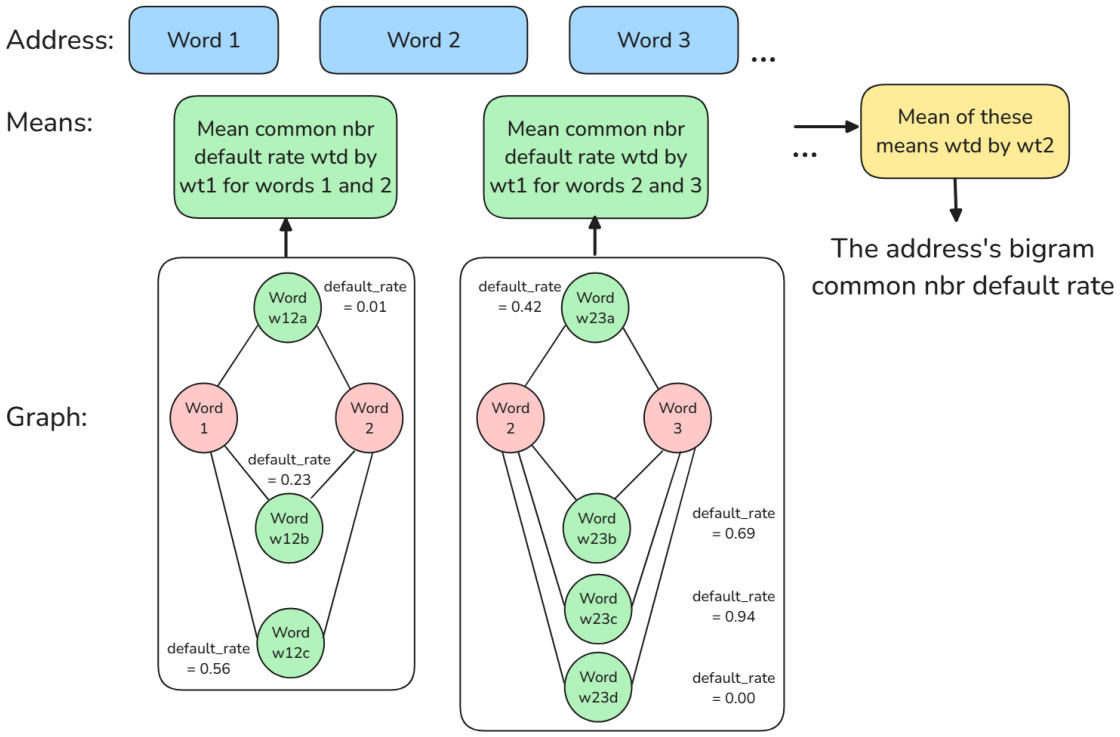


Figure 3.10: Calculating the bigram common neighbour default rate for each address

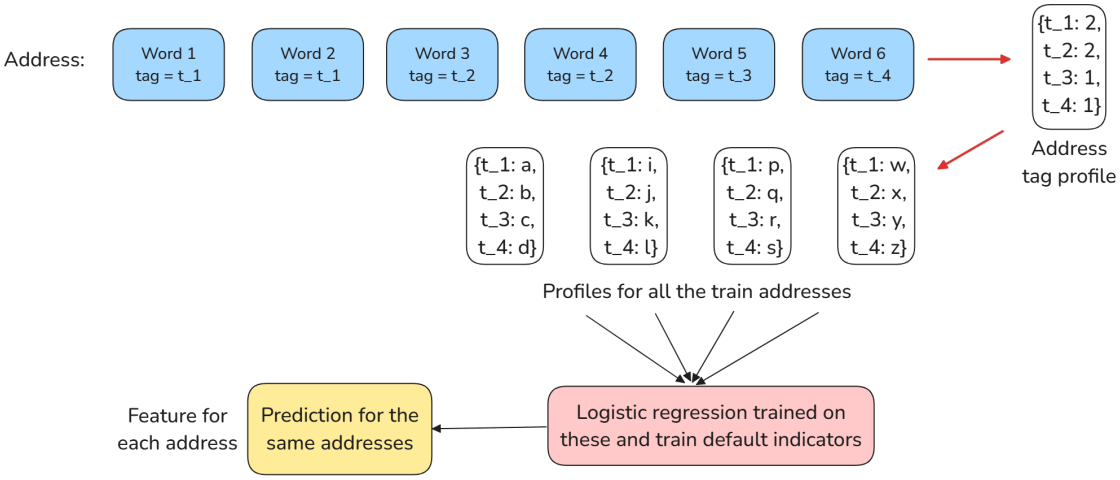


Figure 3.11: Calculating the address tag count profile prediction for each address

5. Address tag count profile prediction: (Figure 3.11)

   (a) Count the number of times each tag appears in the most common tag for each unigram. This creates an *'address tag count profile'* for each address, which is

stored as a dictionary.

(b) Then use the dictionary values to predict the default rate of that address using a simple logistic regression.

(c) The prediction probability is outputted as a feature.

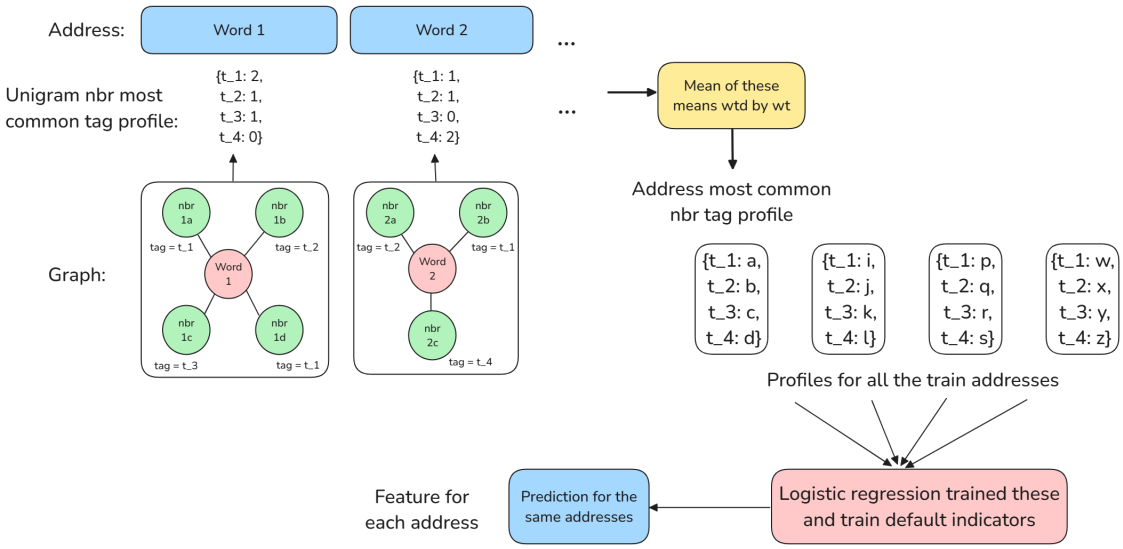


Figure 3.12: Calculating the most common tag profile prediction for each address

6. Most common tag profile prediction: (Figure 3.12)

   (a) For a unigram and a weight type, consider the most common tags of its neighbours.

   (b) Take a weighted sum over all unigrams to give an *'address most common neighbour tags dictionary'*.

   (c) Then use the dictionary values to predict the default rate of that address using a simple logistic regression.

   (d) The prediction probability is outputted as a feature.

7. Specific tags neighbour default rate: (Figure 3.13)

   (a) Given two entity tags $tag1$ and $tag2$ and two weight types $wt1$ and $wt2$, consider only those unigrams from the address having their most common tag as $tag1$.

   (b) Then for each of these unigrams, its neighbours are considered, having their most common tag as $tag2$.

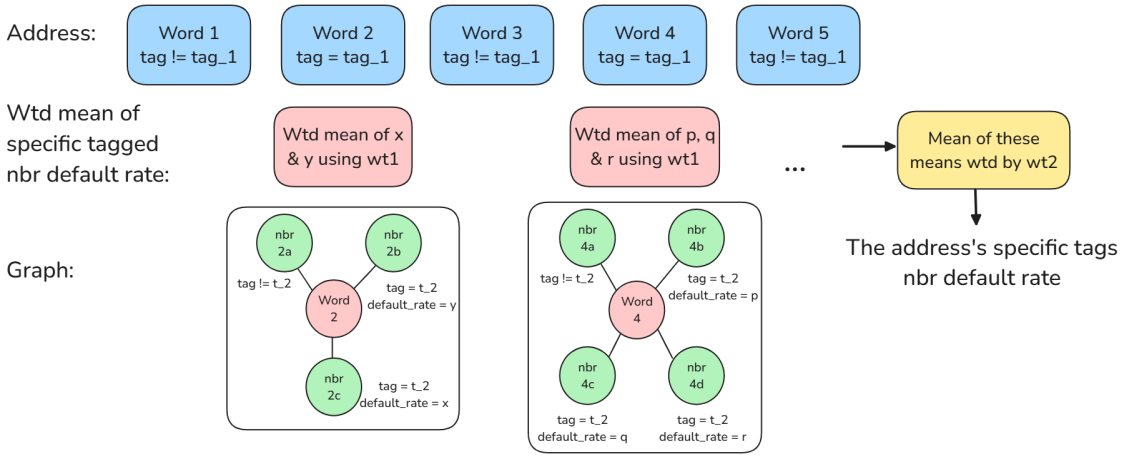


Figure 3.13: Calculating the specific tags neighbour default rate for each address

(c) Take the weighted mean of the default rates of these specific neighbours for each such unigram using $wt1$

(d) Consider the weighted mean of these means using $wt2$.

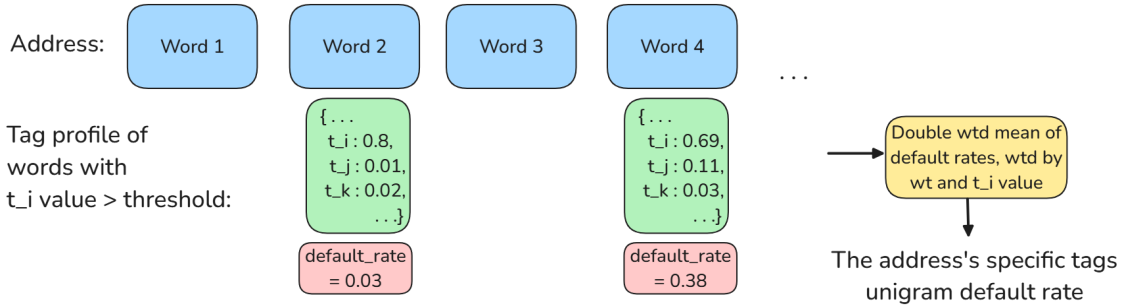


Figure 3.14: Calculating the specific tags unig default rate for each address

8. Specific tag unigram default rate: (Figure 3.14)

(a) For a given tag and a weight type wt, consider only those unigrams with their tag profile value above a pre-decided *tag_threshold*.

(b) For these unigrams, take the double weighted mean of the default rate, weighted using both the tag profile value and the weights.

Weights

In the descriptions above, I mean any of the following by weights, as suitable:

1. the node frequency
2. the node pagerank
3. n/(node frequency), where n is the maximum of the unigram frequencies
4. 1/(node pagerank)
5. the edge weight
6. 1/(edge weight)

Feature type	**Comments**	**Total**
Mean node bad rate	One for each weight	5
Mean edge bad rate	One for each weight	3
Frequency	One for each weight	5
Pagerank	One for each weight	5
Weighted Degree Centrality	One for each weight	5
Clustering Coefficient	One for each weight	5
Bigram common neighbour default rate	One for each of 3 edge weight types and each of 5 node weight types	15
Address tag count profile prediction	One for each weight	6
Most common tag profile prediction	One for each weight	5
Specific tags neighbour default rate	8 inner weights × 5 outer weights × 8 unigram tags × 8 neighbour tags	2560
Specific tags unigram default rate	5 weights × 8 tags	40
Total		**2654**

Table 3.1: Counting the total number of features that can be computed for each address

3.5.6 Feature Selection

As shown in Table (3.1), thousands of features can be computed for an address. However, some of these features require significant computational time, and others may have near-identical values across all addresses. As a result, the practically useful features are far fewer.

After computation, as done previously for the PI-networks, each feature is divided into 5 roughly equal-sized bins. The Information Value (IV) is then calculated, and only features

with IV between 0.02 and 1.00 are retained. Following this, features with a Spearman correlation [44] greater than 0.8 are removed. This drastically reduces the number of selected features.

Gradient boosted decision trees (GBDTs) are then built using these selected features. Hyperparameter tuning is performed using the randomized search cross-validation technique available in the Sci-kit Learn library [37].

3.5.7 Developing Separate Networks for North, Central, and South India

Customer behavior analysis has shown that dividing the country into North, Central, and South India reveals distinct address patterns in different regions. This division can be achieved using the first digits of the PIN code. However, to ensure comparability of results, the number of addresses considered for each region must be of the same order of magnitude. Around 887 thousand addresses were sampled to develop the entire India graph.

Region	**PIN's first digit**	**Sampled Addresses**	**States and Union Territories**
North India	1, 2	869 k	Himachal Pradesh, Jammu and Kashmir, Ladakh, Delhi, Haryana, Punjab, Chandigarh, Uttar Pradesh, Uttarakhand
Central India	3, 4, 7, 8, 9	948 k	Gujarat, The Dadra and Nagar Haveli and Daman and Diu, Rajasthan, Telangana, Chhattisgarh, Maharashtra, Madhya Pradesh, Goa, Assam, Meghalaya, Manipur, Mizoram, Nagaland, Tripura, Arunachal Pradesh, Odisha, West Bengal, Andaman and Nicobar Islands, Sikkim, Telangana, Jharkhand, Bihar
South India	5, 6	924 k	Andhra Pradesh, Telangana, Karnataka, Puducherry, Kerala, Tamil Nadu, Lakshadweep

Table 3.2: Using PIN's first digit to divide India into regions

Therefore, I also created separate networks for different regions of India, using the divisions shown in Table (3.2). The sizes of these networks are mentioned in Table (3.3). The same steps applied to the entire India graph were also performed on these regional networks.

Certain words appear more frequently in addresses from specific regions than in addresses across the country. For example, *'kovil'*, the transliteration of the Tamil word for temple, is common in addresses in South India but rare nationwide. Developing a country-wide network reduces the frequency and PageRank of such region-specific words compared to commonly used words like 'road' or 'near,' which occur uniformly across India. In addition to this advantage, region-wise networks also allow for faster feature computation due to fewer nodes and edges to traverse.

Region	**Normalised number of nodes**	**Normalised number of edges**
North India	0.284	1.381
Central India	0.461	2.109
South India	0.504	1.838
Entire India	0.575	2.254

Table 3.3: Sizes of the address unigram co-occurrence networks, normalised by the number of addresses used

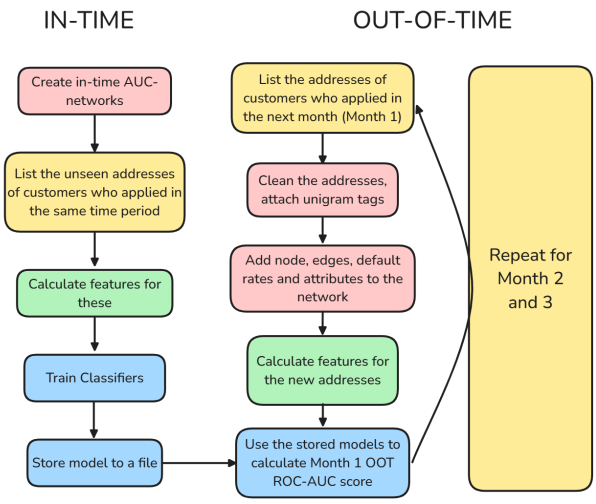


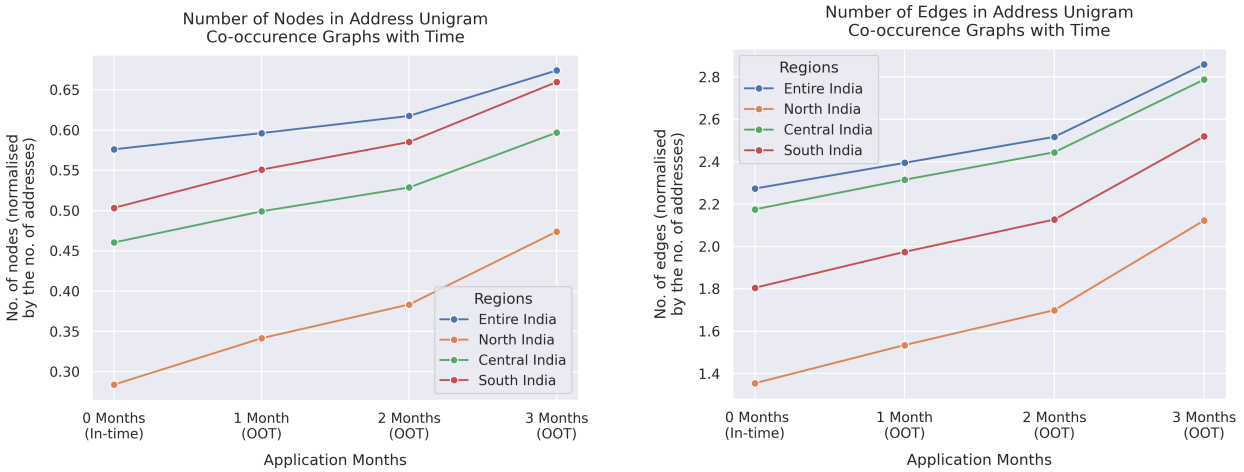
Figure 3.15: The out-of-time testing pipeline for AUC networks

3.5.8 Out-of-Time Testing

I performed out-of-time testing for three subsequent months to evaluate whether the model can extrapolate its learning to customer addresses that apply later in time. See

Figure (3.15). Similar to the PI-networks, the graph size updates with the addition of new addresses each month.

Figure (3.16) shows that the graph sizes for North India are smaller. One possible reason is the lower linguistic diversity in this region compared to the other two regions. Conversely, the graph sizes are larger for South India due to higher language diversity.



(a) Number of nodes increase with time

(b) Number of edges increase with time

Figure 3.16: The AUC graph size increases with time

3.6 Comparing with the Standalone Models

Among the models described earlier, the best-performing ones were compared with the standalone model developed on the same dataset to understand their true usefulness. If the results of my models are highly correlated with those of the standalone models, it implies that they are not adding much additional value.

In order to compare, a new model, similar to the organisation's existing model, was created with the available sample of the customer population. The features used were calculated from bureau reports at the time of the application. These are the traditional features and not the ones obtained from networks. The binary classification model was built with the same target variable with an out-of-time ROC-AUC score of 0.60.

Since the network-based classifiers are GBDT classifiers, they produce predicted default probabilities as floats between 0 and 1. The standalone model also outputs in the same format. To compare the models:

1. Both the *'network_scores'* and *'existing_scores'* were binned into 5 equal-sized bins based on the number of customers in each bin.

2. The population used for comparison consisted of customers who applied 1 to 3 months out-of-time.

3. Two pivot tables were created, where:

   - Each row corresponds to the existing score bins.
   - Each column corresponds to the network score bins.

4. The number of customers in each combination of bins was counted.

5. The mean of the *bad_rate_perf* for the customers falling in each combination of bins was calculated.

These tables provide insights into how differently the two models characterise the customers.

3.7 Predicting for Thin Customers

The best-performing models were also used to predict the default status of the thin customers. Their true default status was compared with the predictions. A high value of the ROC-AUC score indicates that the model performs well for the thin population as well — an area where the standalone models have shown limitations.

Chapter 4

Results

4.1 Results of PI network based classifiers

Recall that two PI-networks were created: one using data from January 2023, and the other using data from both January and February 2023. Subsequently, both networks were updated thrice by incorporating data from the next three months. These two networks will henceforth be referred to as the **jan-network** and the **jan-feb-network**, respectively.

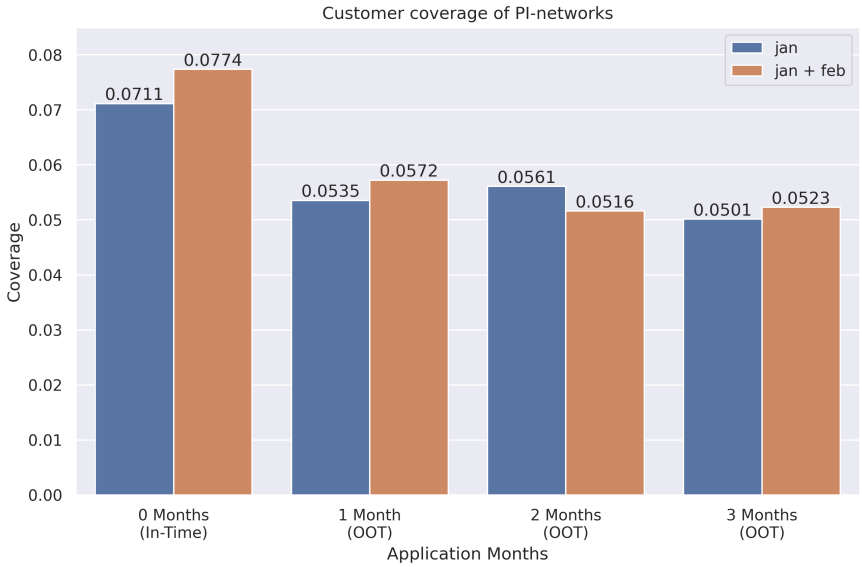


Figure 4.1: Customer Coverage of PI-based networks

4.1.1 Customer coverage

As defined in Definition (3.3.1), the classifiers built on in-time data exhibit a customer coverage of approximately 7%. When the network is updated to include out-of-time customers, the coverage decreases to around 5%. The jan-feb-network generally demonstrates higher coverage compared to the corresponding jan-network. See Figure (4.1).

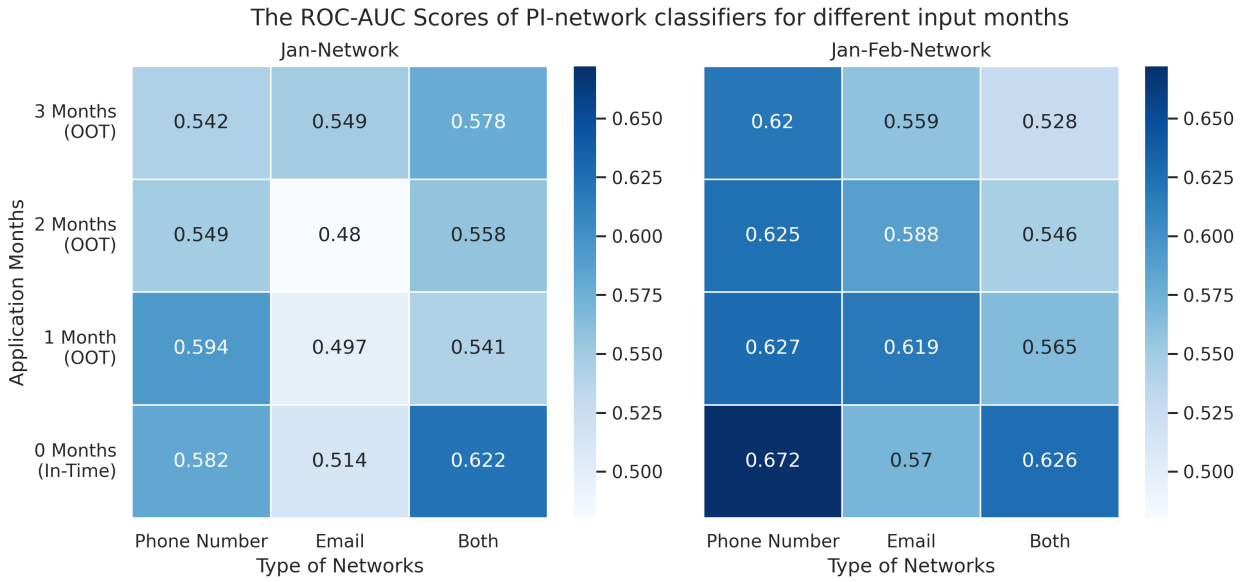


Figure 4.2: The ROC-AUC scores are satisfactory for phone number network based classifiers

4.1.2 PI-Network Classifiers' ROC-AUC Scores

Figure (4.2) presents the ROC-AUC scores of the GBDTs trained on the PI-network features. The scores for the jan-feb-network are generally higher than those for the jan-network, with the effect being most pronounced in the phone number networks. Thus, incorporating more customers into the networks proves to be beneficial.

Comparing the figures reveals that the GBDT models trained using phone number networks produce similar results for in-time and out-of-time test datasets. This, however, is not true for the other GBDT models. Thus, phone number-based GBDT models exhibit robustness to real-world variability in the input data. The ROC-AUC score of approximately 0.62 indicates that these models can be useful when used in conjunction with the standalone models at Axio.

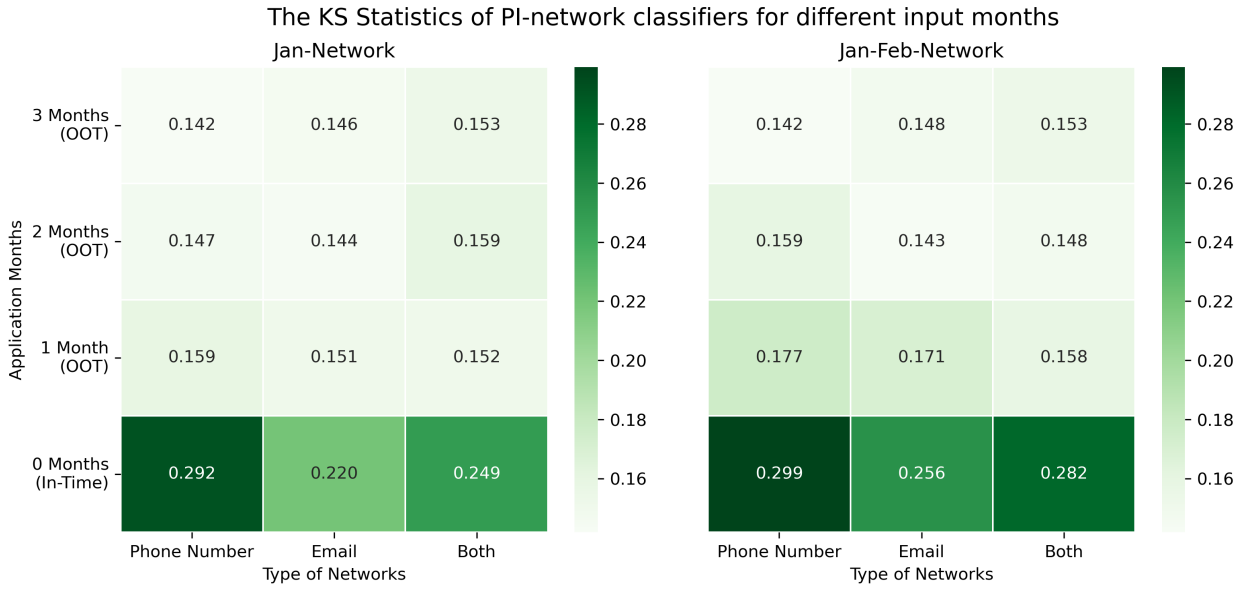


Figure 4.3: The KS statistics are the highest for phone number network based classifiers

4.1.3 PI-Network Classifiers' KS Statistics

Since the phone number network based classifiers show appreciable predictive power, their KS values were calculated for different input months. They were also compared with the other PI-classifiers from both jan-network and jan-feb-network.

Refer to Figure (4.3). It shows the KS statistics from the jan-feb-networks is, for most of the cases, higher than the corresponding values from the jan-network. This again shows that adding more data to the PI-networks increases the ability of the model to distinguish correctly between the two classes.

Another observation to note is that the values for the phone-number network are the highest among the three kinds of networks, both for jan-network and jan-feb-network. This trend is similar to the one observed from Figure (4.2) for ROC-AUC. However, unlike before, the values degrade over time.

The corresponding KS plots, 24 in number, have not been added for brevity. Thus, phone number-based GBDT models perform better than the other models in terms of the distinguishing ability between the defaulters and the non-defaulters.

4.1.4 Explaining the output for the Email ID Network classifiers

The primary reason for the disparity between the phone-number networks and email ID networks is that the email ID used by most people for their loan applications is not their own but that of their loan provider. Those who regularly use their personal email ID do provide it, but they are in the minority. Some of the most common email IDs on the network edges are organisation-level email IDs, which follow the format:

- *care@some_finance_company.in*,
- *asset.customer@some_bank.com*,
- *customer_care@some_bank.org*, etc.

As a result, customers who do not have a real-life acquaintance can also be neighbours in the email ID networks. In conclusion, making inferences about a person based on others who share the same email ID in past loan applications is difficult, which explains the poor performance of the email ID network.

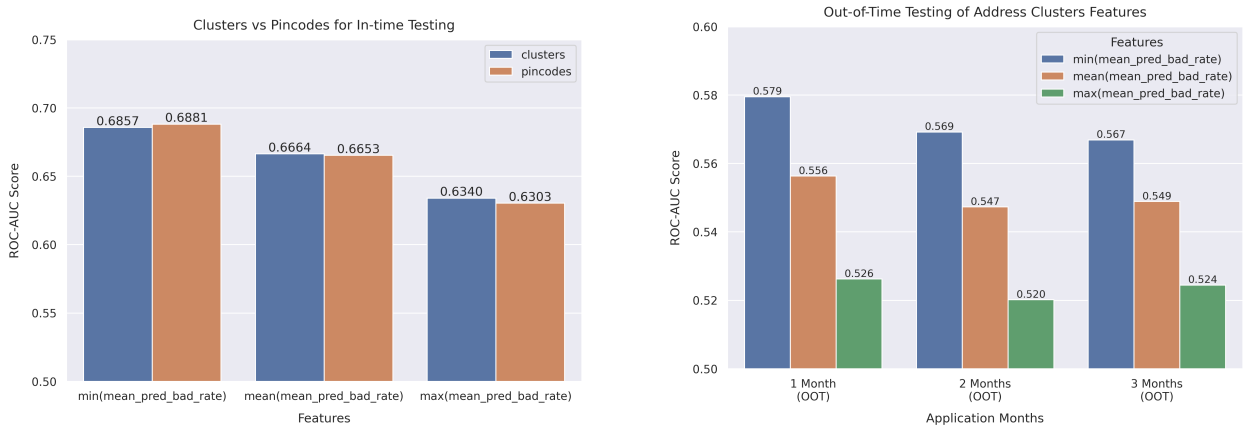
On the other hand, loan applicants typically provide phone numbers belonging to themselves, their family members, neighbours, flatmates or friends. Consequently, making inferences about a person based on others who share the same phone number in past loan applications is relatively easier, which accounts for the better performance demonstrated by the phone number network.

The features obtained from both networks, however, do not perform well. This is again attributed to the prevalence of organisation-level email IDs. In both the jan-network and the jan-feb-network, approximately 80% of the edges are of type *'email ID'*, while only 20% of the edges are of type *'phone number'*.

To conclude, phone number network-based classifiers are **advantageous** for Axio in identifying customers who are likely to default in the future.

4.2 Results of Pin code-wise Address Clustering

The clustering was performed on data from January 2022 to May 2023. The ROC-AUC score for the three features on the in-time test data is approximately **0.6620**. However, as shown in Figure 4.4a, the corresponding value for using pin codes without any clustering is



(a) Clusters vs pincodes for in-time testing

(b) Cluster performance in out-of-time testing

Figure 4.4: Was address clustering any useful?

0.6612. These two values are comparable, implying that the entire process of preprocessing the addresses, selecting the pin codes, embedding them, clustering them, and creating features for each user was **futile**.

As depicted in Figure 4.4b, the out-of-time testing performance deteriorates over time, with values approaching 0.5, which is the ROC-AUC score of a random classifier.

In conclusion, pin code-wise clustering of the addresses generates features that are acceptable for in-time testing but not remarkable. The technique, however, fails in out-of-time testing. Consequently, classifiers were not trained, and ROC-AUC scores and KS statistics were not calculated.

4.3 Results of Credit Risk Analysis Using AUC Graphs

Some features defined for the AUC graphs require a significant amount of time to be calculated, and hence, they were not computed for all the weights and tags. Among the calculated features, those with low Information Value (IV) and high correlation were removed.

Region	PIN's first digits	Addresses	Selected Features
Entire India	1 to 9	887 k	17
North India	1 and 2	869 k	12
Central India	3, 4, 7, 8 and 9	948 k	12
South India	5 and 6	924 k	16

Table 4.1: Sizes of the AUC graphs

Figure 4.5 shows that the in-time ROC-AUC scores are acceptable, if not remarkable. The scores for region-wise classifiers are better than those for the entire nation, as suggested by other models at Axio. Thus, using region-wise graphs proves to be more beneficial.

However, the out-of-time ROC-AUC scores are close to 0.5, which corresponds to the score of a random classifier. In conclusion, these models are **not useful** for real-time customer classification. So, the KS values were not calculated.

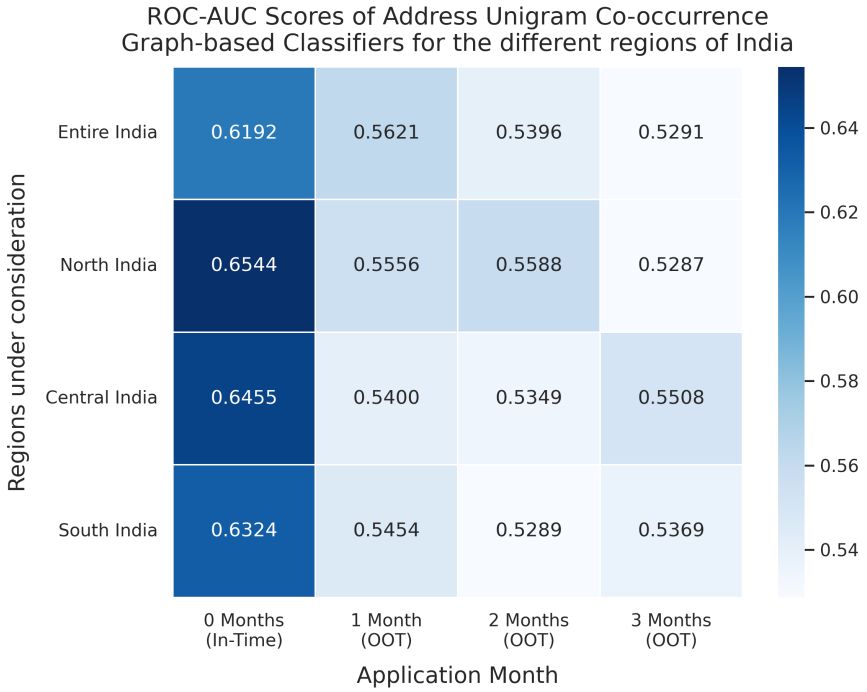


Figure 4.5: ROC-AUC scores of GBDTs trained on features from different Indian regions

4.4 Comparison with the Standalone Models

The phone number-based classifier has shown appreciable results in terms of ROC-AUC and KS. Therefore, its scores were compared with the standalone models at Axio to generate two pivot tables, as described in Section 3.6. Recall that the two scores were binned into five bins of equal size. Figure 4.6 presents their heatmaps, where both types of bins are labeled from 0 to 4, in increasing predicted probability of default.

As shown in Figure 4.6a, most of the customers fall along the main diagonal (top-left to bottom-right), indicating that the two scores classify customers in a similar manner. The

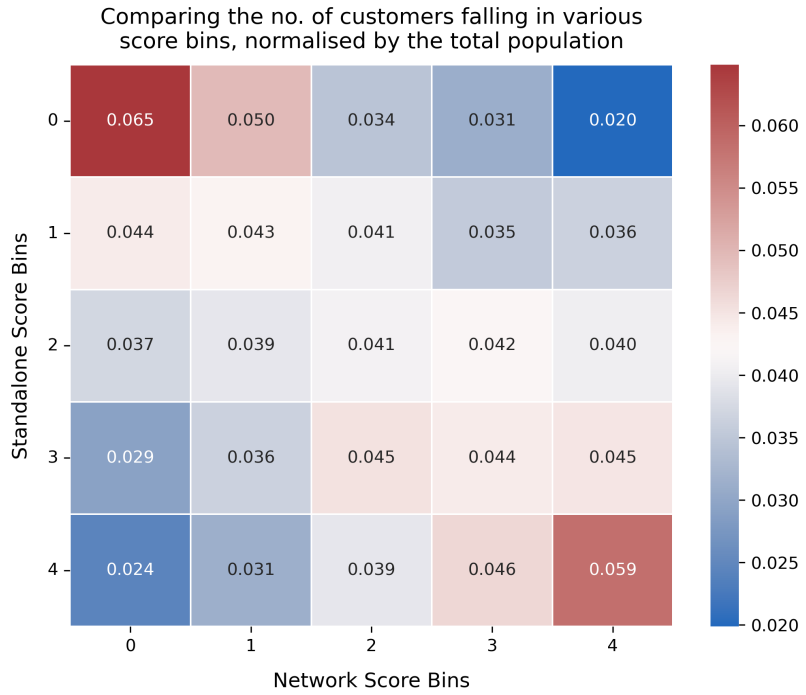
mean *bad_rate_perf* in Figure 4.6b increases from the top-left to the bottom-right. This suggests that the predictions of the two models are aligned with their true *bad_rate_perf*.

Another noteworthy detail is the bottom-left cell of the heatmap in Figure 4.6b. These customers are classified as high-risk by the standalone model but as low-risk by the PI-network model. Such customers would be denied credit if the standalone model were utilised. However, since their mean *bad_rate_perf* is significantly lower than that of the population, they can be safely granted credit. This insight was solely discovered through the use of the phone number network-based classifier. Therefore, it can be effectively used in conjunction with the standalone model to extend credit to a larger customer base.

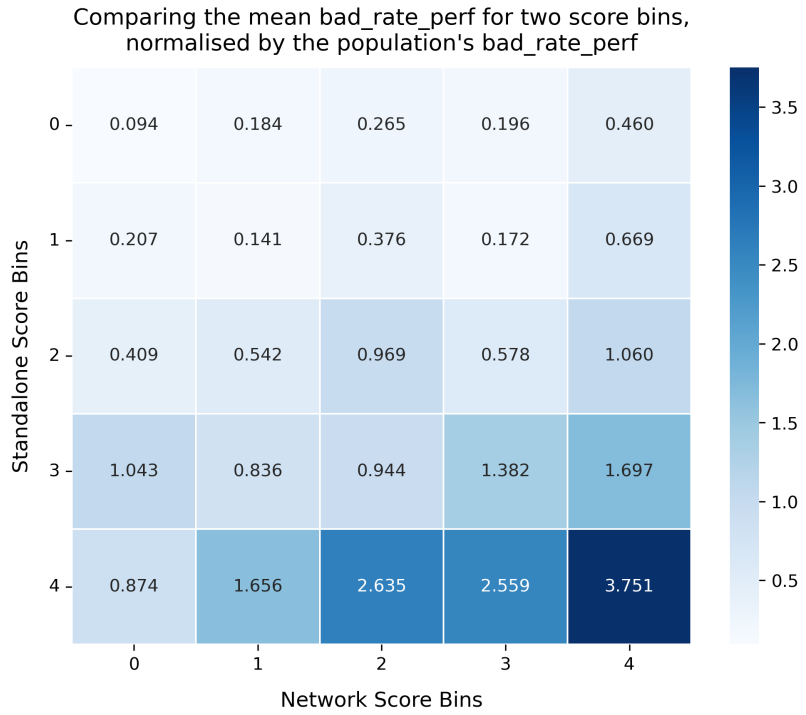
4.5 Prediction for Thin Customers

The proportion of thin customers is quite low. As mentioned in Subsection 4.1.1, the customer coverage of PI networks is also quite low. Consequently, most thin customers have no neighbours in the phone number networks. As a result, calculating the ROC-AUC score for such a small subset of customers does not provide meaningful insights into the predictive power of the model, and thus, it was not performed.

The current graph was constructed using 38 months of data. If a larger dataset were used, it would be possible to generate satisfactory predictions for thin customers. However, in such cases, one may encounter a shortage of computational resources.



(a) The number of customers in each bin combination



(b) The mean bad_rate_perf of customers in each bin combination

Figure 4.6: Comparing the PI-phone number network based classifier with the standalone classifier

Chapter 5

Conclusion and Future Directions

This project involved utilising the personal information of consumers to predict loan default. The use of shared mobile numbers has demonstrated good predictive power on both in-time and out-of-time test datasets. However, the proportion of customers for whom the classifier produces a prediction remains relatively low. The use of shared email IDs, however, is not as effective.

The phone number network-based classifiers can be used alongside the standalone classifier to extend credit to a larger customer base. Thus, this model is more inclusive than models that consider only traditional variables. However, the model is unable to predict outcomes for thin customers due to its limited customer coverage.

It is challenging to directly cluster addresses from a pin code region into different localities based solely on their text embeddings. The features generated for each individual, based on their inferred localities, predict loan performance effectively in in-time testing but not as well in out-of-time testing.

The development of address unigram co-occurrence graphs yields acceptable, though not remarkable, in-time testing results. However, these graphs fail in out-of-time testing. Better results could be achieved by allocating more resources for the calculation and selection of features.

Overall, this project demonstrates that graph-based features possess predictive power and can complement features derived from other sources to provide accurate predictions of loan performance for Axio's customers. However, further development is required to address the shortcomings of these models.

5.1 Future Directions

5.1.1 PI-Network Classifiers

- A natural language processing-based classifier can be developed to classify bureau emails into personal and organisational emails. Subsequently, email ID networks can be created using only personal emails.

- A larger dataset, spanning multiple years, should be used to construct the network to obtain richer features.

- Additional features, beyond those described here, can be devised.

- Neural network classifiers can be developed to utilise the features extracted from phone number and email ID networks.

5.1.2 Pin Code-wise Clustering of Addresses

- The techniques applied to preprocess KYC addresses, such as splitting, joining, and correcting misspellings, can also be applied to bureau addresses before performing pin code-wise clustering. Since bureau addresses are more irregular than KYC addresses, careful implementation would be essential.

- Geospatial data can be incorporated to complement the address text data. This approach may result in clusters that more closely align with actual localities.

- Additional features, apart from the three used here, can be formulated.

5.1.3 Address Unigram Co-occurrence Graph-based Classifiers

- Commonly occurring bigrams and trigrams can be included as nodes to expand the scope of the graphs, e.g., 'near railway station', 'co-operative housing society', 'road near', etc. This modification would require subsequent changes to the structure of the graph and the calculation of features.

- Separate AUC graphs can be built for each state and union territory of India, enhancing the relevance of words that are locally significant but not common across the country.

- For each node of the AUC graph, a list of all the pin codes where that unigram appears in addresses can be added. This addition could provide more features.
- With enhanced computational resources, all the features proposed in this work can be calculated, leading to a better understanding of patterns in the data by the classifier.

Bibliography

[1] Edward I. Altman. "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy". In: *The Journal of Finance* 23.4 (1968), pp. 589–609. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2978933 (visited on 07/23/2024).

[2] David Alvarez-Melis et al. *Weight of Evidence as a Basis for Human-Oriented Explanations*. 2019. arXiv: 1910.13503 [cs.LG]. URL: https://arxiv.org/abs/1910.13503.

[3] Micheal Olaolu Arowolo et al. "A Prediction Model for Bank Loans Using Agglomerative Hierarchical Clustering with Classification Approach". In: *Covenant Journal of Informatics and Communication Technology* (2022).

[4] Dmitrii Babaev et al. "E.T.-RNN: Applying Deep Learning to Credit Loan Applications". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 2183–2190. ISBN: 9781450362016. DOI: 10.1145/3292500.3330693. URL: https://doi.org/10.1145/3292500.3330693.

[5] T. Ravindra Babu et al. "Geographical address classification without using geolocation coordinates". In: *Proceedings of the 9th Workshop on Geographic Information Retrieval*. GIR '15: 9th Workshop on Geographic Information Retrieval. Paris France: ACM, Nov. 26, 2015, pp. 1–10. ISBN: 978-1-4503-3937-7. DOI: 10.1145/2837689.2837696. URL: https://dl.acm.org/doi/10.1145/2837689.2837696 (visited on 02/06/2025).

[6] Max Bachmann. *rapidfuzz/RapidFuzz: Release 3.8.1*. Version v3.8.1. Apr. 2024. DOI: 10.5281/zenodo.10938887. URL: https://doi.org/10.5281/zenodo.10938887.

[7] Dave Bechberger, Josh Perryman, and Ted Wilmes. *Graph databases in action: examples in Gremlin*. eng. Shelter Island, NY: Manning, 2020. ISBN: 978-1-61729-637-6. URL: https://www.manning.com/books/graph-databases-in-action.

[8] Ankan Bhowmik et al. “An Analytical Survey on Credit Scoring Mechanism in India”. In: *Cyber Intelligence and Information Retrieval*. Ed. by Soumi Dutta et al. Vol. 1025. Series Title: Lecture Notes in Networks and Systems. Singapore: Springer Nature Singapore, 2024, pp. 253–271. ISBN: 978-981-97-3593-8 978-981-97-3594-5. DOI: 10.1007/978-981-97-3594-5_21. URL: https://link.springer.com/10.1007/978-981-97-3594-5_21 (visited on 03/12/2025).

[9] Melissa L. Knutson. *CREDIT SCORING APPROACHES GUIDELINES-FINAL-WEB*. English. Apr. 2020. URL: https://thedocs.worldbank.org/en/doc/935891585869698451-0130022020/CREDIT-SCORING-APPROACHES-GUIDELINES-FINAL-WEB (visited on 07/04/2024).

[10] Jillian M. Clements et al. *Sequential Deep Learning for Credit Risk Monitoring with Tabular Financial Data*. Papers 2012.15330. arXiv.org, Dec. 2020. DOI: https://doi.org/10.48550/arXiv.2012.15330. URL: https://ideas.repec.org/p/arx/papers/2012.15330.html.

[11] Sofie de Cnudde et al. *Who cares about your Facebook friends? Credit scoring for microfinance*. Working Papers. University of Antwerp, Faculty of Business and Economics, 2015. URL: https://EconPapers.repec.org/RePEc:ant:wpaper:2015018.

[12] Şeref Kerem Çorbacıoğlu and Gökhan Aksel. “Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value”. In: *Turkish Journal of Emergency Medicine* 23.4 (Oct. 2023), pp. 195–198. ISSN: 2452-2473. DOI: 10.4103/tjem.tjem_182_23. URL: https://journals.lww.com/10.4103/tjem.tjem_182_23 (visited on 03/03/2025).

[13] Selman Delil et al. “Parsing Address Texts with Deep Learning Method”. In: *2020 28th Signal Processing and Communications Applications Conference (SIU)*. 2020 28th Signal Processing and Communications Applications Conference (SIU). Gaziantep, Turkey: IEEE, Oct. 5, 2020, pp. 1–4. ISBN: 978-1-7281-7206-4. DOI: 10.1109/SIU49456.2020.9302154. URL: https://ieeexplore.ieee.org/document/9302154/ (visited on 02/05/2025).

[14] Shizhe Deng et al. “CNN-based feature cross and classifier for loan default prediction”. In: *2020 International Conference on Image, Video Processing and Artificial Intelligence*. Third International Conference on Image, Video Processing and Artificial Intelligence. Ed. by Ruidan Su. Shanghai, China: SPIE, Nov. 10, 2020, p. 34. ISBN: 978-1-5106-3997-3 978-1-5106-3998-0. DOI: 10.1117/12.2579457. URL: https:

//www.spiedigitallibrary.org/conference-proceedings-of-spie/11584/2579457/CNN-based-feature-cross-and-classifier-for-loan-default-prediction/10.1117/12.2579457.full (visited on 07/24/2024).

[15] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[16] Viani B. Djeundje et al. "Enhancing credit scoring with alternative data". In: *Expert Systems with Applications* 163 (Jan. 2021), p. 113766. ISSN: 09574174. DOI: 10.1016/j.eswa.2020.113766. URL: https://linkinghub.elsevier.com/retrieve/pii/S095741742030590X (visited on 07/29/2024).

[17] Erick Fernando and Pandapotan Siagian. "Proposal to use the Analytic Hierarchy Process Method Evaluate Bank Credit Submissions". In: *Procedia Computer Science* 179 (2021). 5th International Conference on Computer Science and Computational Intelligence 2020, pp. 232–241. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2021.01.002. URL: https://www.sciencedirect.com/science/article/pii/S1877050921000028.

[18] Jerome H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232. ISSN: 00905364, 21688966. URL: http://www.jstor.org/stable/2699986 (visited on 03/07/2025).

[19] Abhinav Ganesan, Anubhav Gupta, and Jose Mathew. "Mining points of interest via address embeddings: an unsupervised approach". In: *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Location-based Recommendations, Geosocial Networks and Geoadvertising*. SIGSPATIAL '21: 29th International Conference on Advances in Geographic Information Systems. Beijing China: ACM, Nov. 2, 2021, pp. 1–10. ISBN: 978-1-4503-9100-9. DOI: 10.1145/3486183.3491002. URL: https://dl.acm.org/doi/10.1145/3486183.3491002 (visited on 02/04/2025).

[20] Paolo Giudici, Branka Hadji-Misheva, and Alessandro Spelta. "Network based credit risk models". In: *Quality Engineering* 32.2 (Apr. 2, 2020), pp. 199–211. ISSN: 0898-2112, 1532-4222. DOI: 10.1080/08982112.2019.1655159. URL: https://www.tandfonline.com/doi/full/10.1080/08982112.2019.1655159 (visited on 07/04/2024).

[21] *google-bert/bert-base-uncased · Hugging Face*. Mar. 11, 2024. URL: https://huggingface.co/google-bert/bert-base-uncased (visited on 02/06/2025).

[22] Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. "Exploring network structure, dynamics, and function using NetworkX". In: Los Alamos National Laboratory (LANL), Los Alamos, NM (United States). Jan. 2008. URL: https://www.osti.gov/biblio/960616.

[23] David J. Hand. "Mining the past to determine the future: Problems and possibilities". In: *International Journal of Forecasting* 25.3 (2009). Special Section: Time Series Monitoring, pp. 441–451. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2008.09.004. URL: https://www.sciencedirect.com/science/article/pii/S0169207008001088.

[24] J. L. Hodges. "The significance probability of the Smirnov two-sample test". In: *Arkiv för Matematik* 3.5 (Jan. 1958), pp. 469–486. ISSN: 0004-2080. DOI: 10.1007/BF02589501. URL: http://projecteuclid.org/euclid.afm/1485893310 (visited on 03/26/2025).

[25] *Intellectual Property India*. URL: https://iprsearch.ipindia.gov.in/PublicSearch/PublicationSearch/PatentDetails (visited on 02/06/2025).

[26] Guolin Ke et al. "LightGBM: a highly efficient gradient boosting decision tree". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 3149–3157. ISBN: 9781510860964.

[27] Jung Youn Lee, Joonhyuk Yang, and Eric T. Anderson. "Using Grocery Data for Credit Decisions". In: *Management Science* (July 15, 2024), mnsc.2022.02364. ISSN: 0025-1909, 1526-5501. DOI: 10.1287/mnsc.2022.02364. URL: https://pubsonline.informs.org/doi/10.1287/mnsc.2022.02364 (visited on 08/01/2024).

[28] Pengpeng Li et al. "Bidirectional Gated Recurrent Unit Neural Network for Chinese Address Element Segmentation". In: *ISPRS International Journal of Geo-Information* 9.11 (Oct. 26, 2020), p. 635. ISSN: 2220-9964. DOI: 10.3390/ijgi9110635. URL: https://www.mdpi.com/2220-9964/9/11/635 (visited on 02/05/2025).

[29] Shreyas Mangalgi, Lakshya Kumar, and Ravindra Babu Tallamraju. *Deep Contextual Embeddings for Address Classification in E-commerce*. Version Number: 1. 2020. DOI: 10.48550/ARXIV.2007.03020. URL: https://arxiv.org/abs/2007.03020 (visited on 02/04/2025).

[30] Monica Charles Mhina and Fabrice Labeau. "Using Machine Learning Algorithms to create a Credit Scoring Model for mobile money users". In: *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI). May 2021, pp. 000079–000084. DOI: 10.1109/SACI51354.2021.9465561. URL: https://ieeexplore.ieee.org/document/9465561 (visited on 07/30/2024).

[31] Rony Mitra et al. "Knowledge graph driven credit risk assessment for micro, small and medium-sized enterprises". In: *International Journal of Production Research* 62 (2023), pp. 4273–4289. URL: https://api.semanticscholar.org/CorpusID:261797402.

[32] Nasser Mohammadi and Maryam Zangeneh. "Customer Credit Risk Assessment using Artificial Neural Networks". In: *International Journal of Information Technology and Computer Science* 8 (2016), pp. 58–66. URL: https://api.semanticscholar.org/CorpusID:21139437.

[33] Ricardo Muñoz-Cancino et al. "On the dynamics of credit history and social interaction features, and their impact on creditworthiness assessment performance". In: *Expert Syst. Appl.* 218 (2022), p. 119599. URL: https://api.semanticscholar.org/CorpusID:248157542.

[34] *Open Government Data (OGD) Platform India*. Jan. 21, 2022. URL: https://data.gov.in (visited on 02/04/2025).

[35] María Óskarsdóttir and Cristián Bravo. "Multilayer network analysis for improved credit risk prediction". In: *Omega* 105 (2021), p. 102520. ISSN: 0305-0483. DOI: https://doi.org/10.1016/j.omega.2021.102520. URL: https://www.sciencedirect.com/science/article/pii/S0305048321001298.

[36] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[37] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[38] Aditya Puri. *Report of the Committee to Recommend Data Format for Furnishing of Credit Information to Credit Information Companies*. Mar. 2014. URL: https://rbi.org.in/scripts/PublicationReportDetails.aspx?UrlPage=&ID=763 (visited on 07/11/2024).

[39] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: http://arxiv.org/abs/1908.10084.

[40] Kabir Rustogi et al. *What is the right addressing scheme for India?* Version Number: 2. 2018. DOI: 10.48550/ARXIV.1801.06540. URL: https://arxiv.org/abs/1801.06540 (visited on 02/04/2025).

[41] Shikhar Sharma et al. "Automated Parsing of Geographical Addresses: A Multilayer Feedforward Neural Network Based Approach". In: *2018 IEEE 12th International Conference on Semantic Computing (ICSC)* (2018), pp. 123–130. URL: https://api.semanticscholar.org/CorpusID:4872803.

[42] Feng Shen et al. "A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique". In: *Applied Soft Computing* 98 (Jan. 2021), p. 106852. ISSN: 15684946. DOI: 10.1016/j.asoc.2020.106852. URL: https://linkinghub.elsevier.com/retrieve/pii/S1568494620307900 (visited on 07/24/2024).

[43] Lingyun Song et al. "Enhancing Enterprise Credit Risk Assessment with Cascaded Multi-level Graph Representation Learning". In: *Neural Networks* 169 (2024), pp. 475–484. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2023.10.050. URL: https://www.sciencedirect.com/science/article/pii/S0893608023006160.

[44] "Spearman Rank Correlation Coefficient". In: *The Concise Encyclopedia of Statistics*. New York, NY: Springer New York, 2008, pp. 502–505. ISBN: 978-0-387-32833-1. DOI: 10.1007/978-0-387-32833-1_379. URL: https://doi.org/10.1007/978-0-387-32833-1_379.

[45] Emilio Sulis et al. "Exploiting co-occurrence networks for classification of implicit inter-relationships in legal texts". In: *Information Systems* 106 (May 2022), p. 101821. ISSN: 03064379. DOI: 10.1016/j.is.2021.101821. URL: https://linkinghub.elsevier.com/retrieve/pii/S0306437921000648 (visited on 02/05/2025).

[46] Germanno Teles et al. "Comparative study of support vector machines and random forests machine learning algorithms on credit operation". In: *Software: Practice and Experience* 51.12 (2021), pp. 2492–2500. DOI: https://doi.org/10.1002/spe.2842. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.2842. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2842.

[47] Ellen Tobback and David Martens. "Retail Credit Scoring using Fine-Grained Payment Data". In: *Journal of the Royal Statistical Society Series A: Statistics in Society* 182.4 (Oct. 1, 2019), pp. 1227–1246. ISSN: 0964-1998, 1467-985X. DOI: 10.1111/rssa.12469. URL: https://academic.oup.com/jrsssa/article/182/4/1227/7068329 (visited on 07/30/2024).

[48] Daixin Wang et al. "Temporal-Aware Graph Neural Network for Credit Risk Prediction". In: *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. Proceedings. Society for Industrial and Applied Mathematics, Jan. 2021, pp. 702–710. DOI: 10.1137/1.9781611976700.79. URL: https://epubs.siam.org/doi/10.1137/1.9781611976700.79 (visited on 07/09/2024).

[49] Yadong Wang et al. "Balanced incremental deep reinforcement learning based on variational autoencoder data augmentation for customer credit scoring". In: *Engineering Applications of Artificial Intelligence* 122 (June 2023), p. 106056. ISSN: 09521976. DOI: 10.1016/j.engappai.2023.106056. URL: https://linkinghub.elsevier.com/retrieve/pii/S0952197623002403 (visited on 07/24/2024).

[50] Yu Wang et al. "Comparison of Different Generalizations of Clustering Coefficient and Local Efficiency for Weighted Undirected Graphs". In: *Neural Computation* 29.2 (Feb. 1, 2017), pp. 313–331. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/NECO_a_00914. URL: https://direct.mit.edu/neco/article/29/2/313/8232/Comparison-of-Different-Generalizations-of (visited on 02/12/2025).

[51] Huosong Xia, Wuyue An, and Zuopeng (Justin) Zhang. "Credit Risk Models for Financial Fraud Detection: A New Outlier Feature Analysis Method of XGBoost With SMOTE". In: *Journal of Database Management* 34.1 (Apr. 21, 2023), pp. 1–20. ISSN: 1063-8016, 1533-8010. DOI: 10.4018/JDM.321739. URL: https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/JDM.321739 (visited on 07/05/2024).

[52] Rongzhen Xu and Mengke He. "Application of Deep Learning Neural Network in Online Supply Chain Financial Credit Risk Assessment". In: *2020 International Conference on Computer Information and Big Data Applications (CIBDA)* (2020), pp. 224–232. URL: https://api.semanticscholar.org/CorpusID:220889443.

[53] Jiaqi Yao, Keren Wang, and Jikun Yan. "Incorporating Label Co-Occurrence Into Neural Network-Based Models for Multi-Label Text Classification". In: *IEEE Access* 7 (2019), pp. 183580–183588. DOI: 10.1109/ACCESS.2019.2960626.

[54] Liang Yao, Chengsheng Mao, and Yuan Luo. "Graph Convolutional Networks for Text Classification". In: *ArXiv* abs/1809.05679 (2018). DOI: 10.1609/aaai.v33i01.33017370.

[55] Xiaojiao Yu. *Machine learning application in online lending risk prediction*. Version Number: 1. 2017. DOI: 10.48550/ARXIV.1707.04831. URL: https://arxiv.org/abs/1707.04831 (visited on 07/30/2024).

[56] Tao Zhang et al. "Multiple instance learning for credit risk assessment with transaction data". In: *Knowledge-Based Systems* 161 (Dec. 2018), pp. 65–77. ISSN: 09507051. DOI: 10.1016/j.knosys.2018.07.030. URL: https://linkinghub.elsevier.com/retrieve/pii/S0950705118303800 (visited on 07/04/2024).

[57] Ruonan Zhao and Xiangwu Ding. "Enhancing BERT-Based Chinese Address Recognition Model with Tag Revision Module". In: *2023 8th International Conference on Information Systems Engineering (ICISE)* (2023), pp. 253–258. DOI: 10.1109/ICISE60366.2023.00060.

Appendix A

Glossary

1. **Agglomerative Clustering:** A 'bottom-up' approach of clustering observations into clusters. Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. This is a type of hierarchical clustering, a method of cluster analysis to build a hierarchy of clusters.

2. **AUC graph:** Address unigram co-occurrence graph, a graph with address unigrams as nodes and edges are formed when two unigrams co-occur in any address.

3. **Bisecting k-means Clustering:** A hierarchical variant of k-means clustering that recursively splits data into two clusters at each step. The algorithm begins with all data points in a single cluster and iteratively selects the largest cluster to bisect using standard K-Means. This process continues until the desired number of clusters is reached.

4. **Clustering Coefficient:** The clustering coefficient of a graph node quantifies how close its neighbours are to being a clique (a complete subgraph). It ranges from 0 to 1. See Figure (A.1).

   The value being 0 means the node's neighbours are not connected to each other at all, like a star-subgraph. On the other hand, it being 1 indicates that all of the node's neighbours are fully connected to each other, forming a clique. [50]

5. **CSP:** Credit Service Providers, such as banks or NBFCs

6. **CIC:** A Credit Information Company is an authorised organisation licensed by the RBI that regularly gathers, stores, and evaluates consumer and corporate credit data from

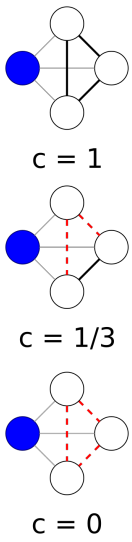


Figure A.1: Clustering coefficients of the blue node change with the presence and absence of edges (solid and dashed lines respectively) between its neighbours (Source: Wikipedia)

financial institutions. This data pertains to individuals, businesses, and companies nationwide. Examples in India include CIBIL, CRIF, Equifax and Experian.

7. **DBSCAN:** Density-Based Spatial Clustering of Applications with Noise is a density-based clustering algorithm that groups together points closely packed in high-density regions while marking points in low-density areas as noise. it does not require specifying the number of clusters and identifies arbitrarily shaped clusters based on two parameters: neighbourhood radius and minimum points required to form a dense region.

8. **Graphlet Degree Vectors:** A graph-based feature representation that captures the local structural role of a node by counting its participation in small, predefined subgraph patterns called graphlets. GDVs provide a compact numerical signature of a node's connectivity, useful for network analysis and machine learning tasks on graphs.

9. **HDBSCAN:** An extension of DBSCAN that performs hierarchical clustering based on varying density levels. It extracts clusters by building a hierarchy of density-based clusters and condensing it into a tree structure, allowing the algorithm to identify clusters of varying densities without requiring a fixed neighbourhood radius. It provides better cluster stability than DBSCAN.

10. **In-time testing:** Testing an ML model using the data obtained from the same time

frame as the training and validation data used to build the model, as opposed to out-of-time testing.

11. **Logistic Regression:** A statistical model used for binary classification that estimates the probability of an outcome using the logistic (sigmoid) function. It applies a linear combination of input features to predict class probabilities.

12. **NBFC:** Non-Banking Financial Corporation. It lends credits to individuals or organisations, but a person cannot open a savings account or avail other banking-related facilities in such a corporation.

13. **Out-of-time testing:** Testing an ML model using the data obtained from a time frame which is in future with respect to the training and validation data. This generally gives equal or worse prediction than in-time testing. A model which gives good results in this testing is robust enough to handle the changes in incoming data with respect to time.

14. **PAN:** Permanent Account Number, a ten-digit alphanumeric character issued to citizens through the Department of Income Tax, Government of India. A person's past credit history is stored linked with their PAN at a CIC.

15. **PCA:** Principal Component Analysis is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while preserving as much variance as possible. It computes new orthogonal features, called principal components, which are linear combinations of the original features. These components are ordered by the amount of variance they capture, allowing the most informative dimensions to be retained while reducing noise and redundancy.

16. **Pivot Table:** A data summarisation tool that automatically sorts, groups, and aggregates data from a larger dataset to provide a structured summary. It allows users to reorganise and analyse data by placing variables along rows and columns, making it easier to identify patterns, relationships, and trends within the data.

17. **Precision:** A performance metric for a binary classifier that measures the proportion of true positive predictions out of all positive predictions made by the model. It indicates how many of the predicted positive cases are actually correct. Precision is

calculated using the following formula.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

A high precision value means that the model makes fewer false positive errors.

18. **Randomised Search Cross-Validation:** A hyperparameter tuning technique that searches for the best model parameters by randomly sampling from a specified set of distribution instead of exhaustively evaluating all combinations. It balances efficiency and performance by testing a subset of hyperparameters while using cross-validation to assess model performance.

19. **Recall:** A performance metric for a binary classifier that measures the proportion of actual positive cases correctly identified by the model. It indicates how well the model captures all the positive instances. Recall is calculated using the following formula.

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

A high recall value means the model successfully identifies most of the actual positive cases.

20. **Silhouette Score:** Silhouette scoring is a method of interpretation and validation of consistency within clusters. It is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). It ranges from -1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.

21. **Spectral Clustering:** A graph-based clustering method that uses the eigenvalues of a similarity matrix to identify clusters. It transforms data into a lower-dimensional space via the graph Laplacian and applies a standard clustering algorithm like K-Means. This approach is effective for detecting non-convex and complex-shaped clusters.

22. **Thin customers:** A customer with a limited past credit history, as opposed to a thick customer. It is difficult to predict the creditworthiness for such individuals. For this work, I have defined thin customers as those with a bureau score less than 100 at the time of application to Axio.

23. **Thick customers:** As opposed to a thin customer, they have taken multiple loans before, and so their past credit history is available.

24. **Weighted Degree Centrality:** The concept of centralities is used to identify the importance of vertices within a graph. The degree centrality is the ratio of the degree of a node to the maximum possible degree of a node in the graph. [7] The weighted version of this treats an edge of weight n as n distinct edges, for $n \in \mathbb{Z}^+$.