International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-021-00243-7

APPLICATIONS

Using network features for credit scoring in microfinance

Paulo Paraíso[1] · Saulo Ruiz[2] · Pedro Gomes[2] · Luís Rodrigues[2] · João Gama[1]

Received: 5 July 2020 / Accepted: 11 January 2021

Abstract
The usage of non-traditional data in credit scoring, from microfinance institutions, is very useful when trying to address the problem, very common in emerging markets, of the lack of a verifiable customers' credit history. In this context, this paper relies on data acquired from smartphones in a loan classification problem. We conduct a set of experiments concerning feature selection, strategies to deal with imbalanced datasets and algorithm choice, to define a baseline model. This model is, then, compared to others adding network features to the original ones. For that comparison, we generate a network that links a given user to its phone book contacts which are users of a given mobile application, taking into account the ethics and privacy concerns involved, and use some feature extraction techniques, such as the introduction of centrality measures and the definition of node embeddings, in order to capture certain aspects of the network's topology. Several node embedding algorithms are tested, but only Node2Vec proves to be significantly better than the baseline model, applying Friedman's post hoc tests. This node embedding algorithm outperforms all the other, representing a relative improvement, in comparison with the baseline model, of 5.74% on the mean accuracy, 7.13% on the area under the Receiver Operating Characteristic curve and 30.83% on the Kolmogorov–Smirnov statistic scores. This method, therefore, proves to be very promising when trying to discriminate between "good" and "bad" customers, in credit scoring classification problems.

Keywords Credit scoring · Microfinance · Networks · Feature extraction · Node embeddings

1 Introduction

According to McKinsey Global Institute [1], in 2016, two billion individuals and 200 million businesses in emerging economies lacked access to financial services. The usage of digital technologies can boost financial inclusion and have a large economic impact. With it, they estimate a 6 percent increase, or a total of $3.7 trillion, of the Gross Domestic Product (GDP) and a potential creation of 95 million new jobs across all sectors of the economy, by 2025.

Despite the lack of access to financial services in emerging countries (only 55% had financial accounts), McKinsey Global Institute [1] estimated that, in 2014, 80% of adults in emerging economies owned a mobile phone. Therefore, the usage of mobile money could be vital in lowering the costs of providing financial services and enabling low-income customers to access this services which, otherwise, using traditional criteria, would be inaccessible, due to their lack of a verifiable credit history.

Over the last decade, several companies proposed products where customers can apply for loans, using mobile applications. For example, M-Shwari [2], a combination of savings and loans offerings which emerged as a collaboration between the Commercial Bank of Africa and the mobile network operator Safaricom, is used by two thirds of Kenyan adults and processes nearly $20 million in daily payment transactions. In these kinds of services, we can have access to non-traditional data, as data imputed by customers when signing up the mobile application, or data captured by their smartphones' logs, as contacts in their phone books, Short Message Service (SMS) logs or installed mobile applications.

This article is a result of the project Risk Assessment for Microfinance, supported by Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF).

✉ Paulo Paraíso
paulo.s.paraiso@inesctec.pt

João Gama
joao.gama@inesctec.pt ; jgama@fep.up.pt

[1] INESC TEC, University of Porto, Porto, Portugal

[2] Pelican Rhythms, Porto, Portugal

Published online: 15 February 2021

Ruiz et al. [3] used non-traditional data, acquired from smartphones, in credit scoring modeling, for loan classification purposes. They implemented the Weight of Evidence (WoE) coding of variables and applied feature selection using Information Value (IV). These techniques proved to add value in the process of identifying "good" borrowers. Our paper proposes extending that study with the extraction of features from a network, generated from phone books of customers of a given mobile application and added them to the previous information from the WoE and IV. We test different node embedding algorithms to verify if we can obtain significant improvements in the performance of the model. We also compare different approaches concerning feature selection using IV, resampling strategies to deal with imbalanced datasets and apply different classification algorithms to find the optimal choices for the baseline model of the given dataset.

The remainder of this paper is organized as follows. In the next section, we revise some of the related work concerning the usage of non-traditional data in credit scoring and network related features. Section 3 describes the used methodology in the problem resolution: feature selection methods are revised, two kind of networks' feature engineering techniques are presented, using centrality measures and node embeddings and implemented statistical tests are defined. Conducted experiments toward the baseline model definition are detailed in Sect. 4. In Sect. 5, we formulate the problem to resolve and describe the dataset used for that purpose. Section 6 compares different node embedding algorithms and proves the relevance of its usage in the improvement in the model's performance. Section 7 explains which performance metrics are used in this study and the results analysis is presented. Ethics and privacy concerns are addressed in Sect. 8, and conclusions and future work are stated in Sect. 9.

2 Related work

Mark Schreiner [4] was one of the precursors in the application of credit scoring models in microfinance. He argued that, as well as in rich countries, where credit scoring plays a relevant role in lending processes, the same could happen in the poorer. In his experiments in Bolivia and Colombia, he concluded that, although not as powerful as in rich countries, credit scoring models had some power in predicting risk. At the same time, he predicted many of the biggest microfinance lenders would add, over the upcoming decades, credit scoring as one of its primal decisions tools. Bumacov et al. [5] claimed that the usage of credit scoring by microfinance institutions (MFIs) has great potential, particularly in developing countries, as it can contribute to financial inclusion, the main mission of microfinance, diminishing opportunity costs and contributing to development.

In their study, Van Gool et al. [6] stated that best practices from other domains, such as the WoE coding or the area under the Receiver Operating Characteristic (ROC) curve, should be further introduced in microfinance credit scoring models. Using a dataset from a mid-sized Bosnian microlender, they extended the geographical reach of previous studies, mainly focused on Southern Africa and Latin America. The Kolmogorov–Smirnov (K-S) statistic is used as a discriminating tool between "good" and "bad" customers. They concluded that credit scoring should be a refinement tool, although not being able to replace the traditional credit process for microfinance. The WoE is also used in [3,7] to transform variables and IV to measure the strength of each variable, when selecting features to use in credit scoring models. These methods were shown to perform well, even with the lack of structured financial data.

The usage of mobile phone data in credit scoring has already been approached, addressing the problem of financial exclusion of customers without a verifiable credit history and being a complementary source of information in order to improve credit scores. In their studies [8,9], the authors used real data from telecommunications operators in Latin America.

Homophily says that individuals with same characteristics tend to be connected. The work of Wei et al. [10] has shown the capability of using network data in credit scoring, to improve its results. Recent work from Misheva et al. [11], with data collected, over 9 years, from modefinance, a FinTech registered as a Credit Rating Agency, proved that including Network centrality features, such as Degree or Closeness Centrality, in scoring algorithms, such as Logistic Regression or CART, can improve the accuracy of the results.

To the best of our knowledge, node embedding techniques have not been much used in credit scoring problems. Óskarsdóttir et al. [12] generated a pseudo-social network which links users by the number of common attributes (for example, installed mobile applications). Using this network, they extracted neighborhood features as good and bad degree or the number of triangles in the network, to get the most similar users and they extracted features from the network using node representation learning.

The contributions of our paper are the following:

1. We generate a network that links customers of a given mobile application to its phone book contacts and perform feature extraction based on centrality measures and node embeddings;
2. We compare different node embedding algorithms and conclude that Node2Vec is the only one that produces significantly better results comparing with the baseline model, generated from a set of experiments concerning

feature extraction, resampling techniques and algorithm choice;
3. We prove the relevance of the introduction of network based features and, in particular, node embeddings in credit scoring models,using a real world dataset.

3 Methodology

In this section, we describe methods used to transform the dataset, to select the most important features and to extract relevant features for the credit scoring model implementation. Statistical tests used to verify the existence of significant differences between the tested models are also introduced.

3.1 Weight of evidence and information value

We use the definitions proposed by Siddiqi [13] of the WoE and IV. These are techniques used to perform variable transformation and selection, respectively, that can be implemented in Credit Scoring to measure the degree of separation between "good" and "bad" customers. They have a huge connection to Logistic Regression modeling due to the logarithm transformation that is applied in WoE and, for that reason, IV is one of the preferred feature selection methods when using a Logistic Regression classifier.

The WoE of a given category X_i, with $i \in \{1, \ldots, n\}$, of a feature X, where n is the total number of categories of that feature is given by:

$$WoE(X_i) = \ln\left(\frac{\text{D. Goods}(X_i)}{\text{D. Bads}(X_i)}\right), \tag{1}$$

where D. Goods(X_i) (respectively, D. Bads(X_i)) represent the percentage of customers that have repaid the loan on time (respectively, defaulted on the repayment) for category i in feature X. Interpreting its value, the further away from 0 it is, the better is the category on distinguishing between "good" and "bad" customers. If its value is positive, the percentage of "good" customers is greater and the other way around, if its value is negative.

The IV of the feature X is the sum of the IV of each category i of X. It is given by:

$$\begin{aligned} \text{IV} &= \sum_i \text{IV}(X_i) \\ &= \sum_i (\text{D. Goods}(X_i) - \text{D. Bads}(X_i)) \times WoE(X_i). \end{aligned} \tag{2}$$

IV is a measure widely used in industry to select features to be utilized in the modeling phase, and the following rule of thumb, in Table 1, proposed by Siddiqi [13], can be applied to access the predictive power of a given feature:

Table 1 Predictive power of a feature using IV

Value	Predictive power
$IV < 0.02$	Useless for Prediction
$0.02 \leq IV < 0.1$	Week Predictor
$0.1 \leq IV < 0.3$	Medium Predictor
$0.3 \leq IV < 0.5$	Strong Predictor
$IV \geq 0.5$	Suspicious Predictor

Example 1 In Table 2, an example of the calculations of WoE and IV, from Eqs. (1) and (2), can be seen. Figure 1 (generated using scorecardpy [14] Python's package) shows us the bin (good/bad) count distribution for the age feature as well as the bad probability of each category. As a summary of Table 2, the IV for this feature is also presented.
Analyzing Table 2 and Fig. 1, we can conclude that the WoE of each category as ascending. This means that the extreme categories are better distinguishing between good and bad customers. The IV of the age feature, in this example, represents that it has a medium predictive power, according to the rule of thumb presented in Table 1.

In this paper, we use the WoE and IV to select features with value greater than a given threshold (see Sect. 4.1). The concepts of the WoE and IV evolved from the Logistic Regression technique, which has been a widely used technique in many domains. They have existed in the Credit Scoring domain for the last four or five decades and are used to perform the so called Initial Characteristic Analysis. This analysis consists of two main tasks. The first step is to access the strength of each characteristic as a predictor of performance to screen out weak characteristics. Then, the strongest characteristics are grouped because of the scorecard format.

Using the WoE has some advantages: it offers an easier way to deal with outliers, grouping them with adjacent classes and, instead of using the raw values, it would be used the WoE scores of each class; it can handle missing values as missing values can be binned separately; since the WoE transformation handles categorical variables, there is no need for dummy variables.

There are some empirical rules when using the WoE: each category/bin should have at least 5% of the observations; each category/bin should be non-zero for both "good" and "bad" observations; the WoE should be distinct for each category. Similar groups should be aggregated; the WoE for non-missing values should be monotonic, going from negative to positive (or from positive to negative) without any reversals; missing values are binned separately. Their weight gives of indication of from which categories/bins missing data comes from.

Table 2 Example of WoE and IV calculations

Category	# Good	# Bad	Total	Bad prob.	% Good	% Bad	WoE	IV
[0, 24)	145	393	538	0.7305	0.0415	0.1125	−0.9971	0.0708
[24, 32)	1097	1403	2500	0.5612	0.3140	0.4015	−0.2460	0.0215
[32, 40)	1308	1052	2360	0.4458	0.3744	0.3011	0.2178	0.0160
[40, 48)	678	465	1143	0.4068	0.1940	0.1331	0.3771	0.0230
[48, ∞)	266	181	447	0.4049	0.0761	0.0518	0.3850	0.0094
Total	3494	3494	6988		1	1		**0.1406**

Bold value refers to the Information Value of the given feature (age)

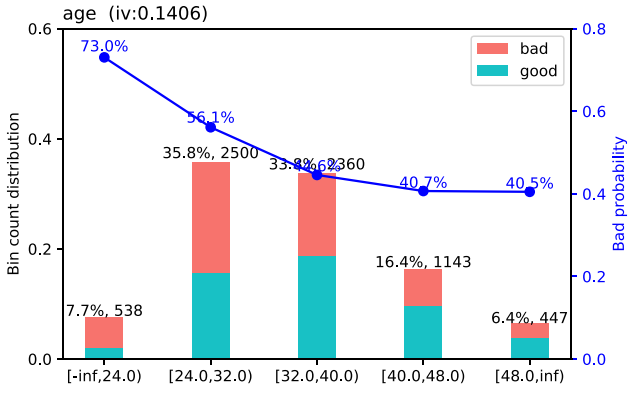


Fig. 1 Bin count distribution / Bad probability for age feature

3.2 Networks

A network [15] consists as two sets of information: a set of nodes V and a set of edges E, or links, between pairs of nodes. A directed network is a network in which each (directed) edge has a direction pointing from one node to another, that is, each edge is an ordered pair of nodes and a weighted network is a network where each edge has a weight attribute attached to it. Formally, a network G can be defined as $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$, with $m, n \in \mathbb{N}$. Given an undirected and unweighted network G, we can define A as the symmetric adjacency matrix of G, a $n \times n$ matrix, with entries a_{ij} such that:

$$a_{\text{ij}} = \begin{cases} 1 \text{ , if there is a edge between } v_i \text{ and } v_j \\ 0 \text{ , otherwise} \end{cases} \tag{3}$$

We can extend Eq. (3) to directed and/or weighted networks. If the network is directed, the adjacency matrix A isn't necessarily symmetric and if it is weighted its entries values aren't necessarily (1-0) binary.

In our study, we generate a directed network of users from the MFI's mobile application, created in the sense that two nodes of the network are related (there is an edge between them) if the second one (destination) is a phone book's contact of the first (origin). In order to do that, we extract the contacts from the users' phone books which were granted a loan in the previous two months, giving us almost 16 million observations. As we are only interested in contacts that are also users of the mobile application we filter our contacts based on that criterion and can generate a network with 209890 nodes and 356767 edges.

The generated network was used in order to try to capture default behavior through the usage of phone book contacts of a given user. In this sense, we wanted to verify if a user connected with defaulted users should be itself a defaulted user. If this network should add value to the model, there should be some sense on the saying "Tell me who your friends are, I'll tell you who you are."

3.2.1 Centrality measures

In order to capture particular characteristics of a network's topology, one can define certain measures of centrality. In a graph, the degree of a node v_i (k_{v_i}) is a measure of involvement of the node in the network and is defined as the (normalized) number of edges incident with it or, putting it in different words, the (normalized) number of nearest neighbors of the node,

$$k_{v_i} = \frac{\sum_{j=1}^{n} a_{ij}}{n-1} \tag{4}$$

On directed graphs, we can consider in-degree of a given node n_i ($k^+_{v_i}$) as the (normalized) number of edges of E ending on v_i, and out-degree of a given node v_i ($k^-_{v_i}$) as the number of edges of E beginning on v_i,

$$k^+_{v_i} = \frac{\sum_{j=1}^{n} a_{ji}}{n-1} \tag{5}$$

$$k^-_{v_i} = \frac{\sum_{j=1}^{n} a_{\text{ij}}}{n-1} \tag{6}$$

Closeness centrality is an approximate measure of the global position of a node in the network, and it is defined as the inverse of the mean distance of a node to other nodes by

$$C_{v_i} = \frac{n-1}{\sum_{v_j \in V \setminus \{v_i\}} d(v_i, v_j)} \tag{7}$$

where $d(v_i, v_j)$ is the length of the shortest path between v_i and v_j. It is a indicator of accessibility, measuring how quickly can a node reach the other nodes in a network.

Eigenvector centrality can be seen as an improved version of degree centrality which considers not only the quantity but also the quality of the relationships between nodes, defining the centrality of a given node v_i (x_{v_i}) with the centrality of the nodes to each they relate, and it is defined by

$$x_{v_i} = \frac{1}{\lambda} \sum_{j=1}^{n} a_{ij} x_{v_j}, \tag{8}$$

where $\lambda \in \mathbb{R}$ is the greatest eigenvalue of A.

Laplacian centrality relates the importance of a node to its ability to respond to the removal of that node from the network. We can define the Laplacian matrix of the network G as $L = D - A$, where D is a diagonal matrix with the nodes' degrees d_i. It can be quantified as the relative drop in Laplacian energy (E_L) of the network,

$$E_L(G) = \sum_{i=1}^{n} \lambda_i^2 \tag{9}$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of its Laplacian matrix. Therefore, the Laplacian centrality of a node v_i of the network G is given by

$$C_L(v_i, G) = E_L(G) - E_L(G_i), \tag{10}$$

where $E_L(G_i)$ is the Laplacian energy of network G on the removal of node i.

Example 2 Consider the marriage relationships data of 15 fifteenth-century Florence families, collected by John Padgett from historical documents, in Fig. 2. We can consider the network of those relationships, colored using different centrality measures: in Fig. 2a using degree centrality, in Fig. 2b, using closeness centrality, in Fig. 2c, using eigenvector centrality and in Fig. 2d, using Laplacian centrality.

Comparing the representations in Fig. 2, we can conclude that different types of centrality measures can capture different types of importance of the nodes, as we can see from their values.

In our work, we choose to calculate in-degree, out-degree, eigenvector, closeness and Laplacian centralities, in order to verify the importance of the nodes in the network or, in other words, which were the most important users of the mobile application. The features of this new model were given by adding the centrality measures features to the ones from the baseline model.

The application of centrality measures to our network was thought in terms of how we could measure how important is a customer in the mobile application community. This way we used in-degree to measure a customer as more important as being a contact or more customers. If we have a customer which is a contact of many customers, the possibility of him being able to default on a loan may have influence on its contacts. The same reasoning can be applied regarding out-degree, where the importance of a customer is measured on how many contacts he has.

The usage of closeness centrality was thought in a more global way, comparing to degree. This way, a customer is more important if he is closer to the other nodes. If we have a customer with high closeness centrality, the quicker he can reach other nodes and possibly pass the information about the default.

Eigenvector centrality is an upgrade to degree centrality and defines a customer as important as more important customers are in their contacts. This way, we can access communities of important nodes which can have an important role in spreading information about the default process.

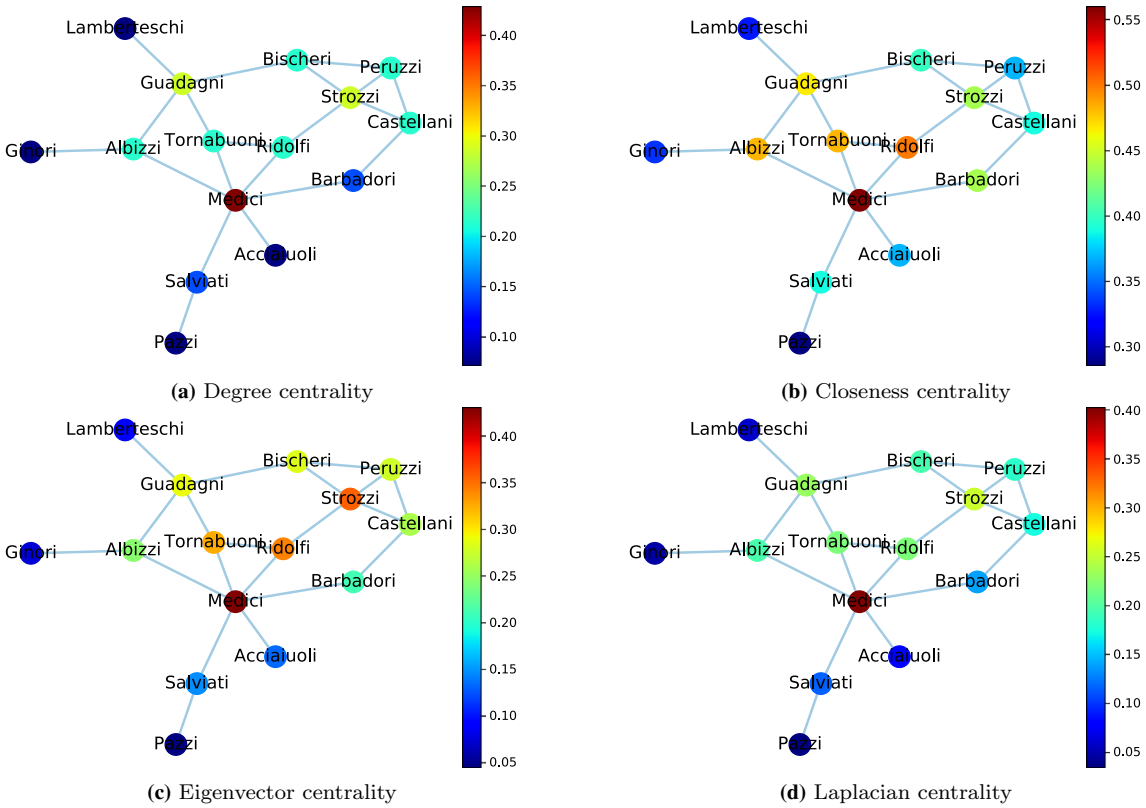
Finally, Laplacian centrality considers that a customer is more important if it has a high influence on its removal from the network, meaning that the information can no longer be passed properly on this new network.

3.2.2 Node embeddings

Recently, methods based on representing networks in a vector space, while preserving their properties, have become widely popular. In these, embeddings are used as features in a model, avoiding the usage of complex classification models applied directly on the network. Nevertheless, the task of obtaining a vector representation of each node in a network is difficult and should be able to preserve the structure of the network and the connections between nodes, be scalable enabling to process large networks and be optimized on the dimension of the embedding.

In random walk-based methods, the node embeddings are generated by constructing a matrix based on the nodes and edges of a given network. A network is represented as a set of random walk paths sampled from it. The methods in this category are applied to the sampled paths in order to embed a network, preserving the properties underneath it. The proximity between neighbors of a given node (second-order proximity) can be preserved by maximizing the probability of observing the neighborhood of a node. The differences between the methods lie on the used random walk method and the model underneath the embedding.

The idea in network representation learning algorithms, such as DeepWalk [16] or Node2Vec [17], is to define a encoder and similarity measures (in the network as well as in the embedding space) in order to, optimizing the parameters of the encoder, both similarities are approximate, meaning that if a node is similar to another in the network (according



(a) Degree centrality

(b) Closeness centrality

(c) Eigenvector centrality

(d) Laplacian centrality

Fig. 2 Centrality measures colored Florentine families marriage network

to the defined measure of similarity), the similarity between the corresponding embeddings should be also high. In this type of node embedding algorithms, cosine similarity is one of the measures of similarity used in the embed space, and two nodes are defined to be as similar, in the network, as higher is the probability of co-occurrence of the nodes in random walks over the network.

Specifically, given a network $G = (V, E)$, the encoder is the mapping function $f : V \rightarrow \mathbb{R}^d$ between nodes and its embeddings, where d is the specified embedding dimension. A network neighborhood of a given node u, $N_S(u) \subset V$, is defined according to some neighborhood sampling strategy S. In this optimization problem, given a network and an encoder function from nodes to its embeddings, the goal is to optimize the objective function,

$$\max_f \sum_{u \in V} \log P(N_S(u) \mid f(u)). \tag{11}$$

which maximizes the log-probability of observing a network neighborhood of a node in a sampling strategy, conditioned on its feature representation given by the encoder. In order to perform this optimization, we run short length random walks from each node on the graph using some sampling strategy, we then collect the neighborhood of each node and optimize the embeddings according to the predictions of the neighborhoods of a given node. In Eq. 11, it is assumed conditional independence meaning that the following equation holds:

$$P(N_S(u) \mid f(u)) = \prod_{n_i \in N_S(u)} P(n_i \mid f(u))$$

Softmax is used to parameterize $P(n_i \mid f(u))$, for $n_i \in N_S(u)$, the following way:

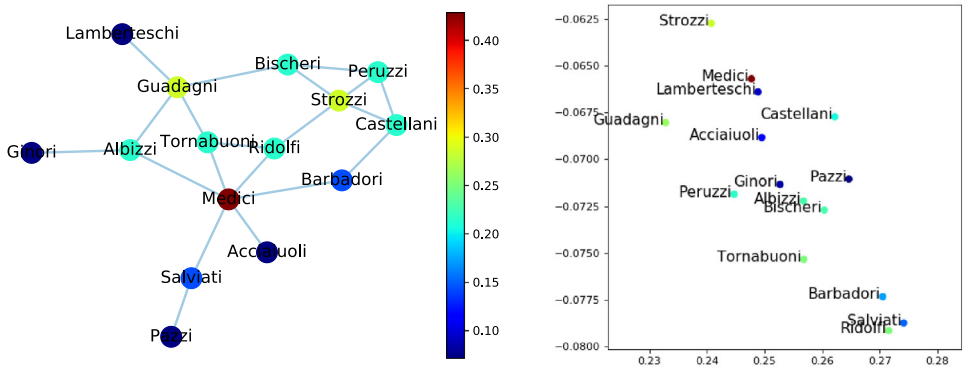
$$P(n_i \mid f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}$$

and negative sampling, sampled with probability proportional to the degree of the node, is used in order to reduce the computation complexity in

$$\sum_{v \in V} \exp(f(v) \cdot f(u)).$$

The optimization is done using stochastic gradient ascent. The formal derivation can be seen below:

$$\max_f \sum_{u \in V} \log(P(N_S(u) \mid f(u))) =$$
$$= \max_f \sum_{u \in V} \log(\prod_{n_i \in N_S(u)} P(n_i \mid f(u)))$$



(a) Degree centrality coloured Network

(b) 2d node embedding representation

Fig. 3 Representations of Florentine families marriage network

$$
\begin{aligned}
&= \max_f \sum_{u \in V} \sum_{n_i \in N_S(u)} \log(P(n_i \mid f(u))) \\
&= \max_f \sum_{u \in V} \sum_{n_i \in N_S(u)} \log\Big(\frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}\Big) \\
&= \max_f \sum_{u \in V} \Big[\sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \\
&\quad -\log\Big(\sum_{v \in V} \exp(f(v) \cdot f(u))\Big)\Big] \\
&\approx \max_f \sum_{u \in V} \Big[\sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \\
&\quad -\log\Big(\sum_{i=1}^{k} \exp(f(v) \cdot f(u))\Big)\Big]
\end{aligned}
$$

Several strategies can be used in order to perform the random walks. In DeepWalk [16], one can run fixed length unbiased random walks starting from each node of the network and, in Node2Vec [17], run biased (on defined hyperparameters p and q) 2nd order random walks (with one step memory) to generate neighborhoods of a given node. This enables to interpolate between classic search strategies Breadth-first Search (BFS) and Depth-first Search (DFS) which, respectively, tend to capture two kinds of similarities: homophily (local view of the network) and structural equivalence (global view of the network).

Example 3 Consider, again, the marriage relationships data among fifteenth-century Florence families, in Fig. 3. The network of those relationships is represented in Fig. 3a while, in Fig. 3b, we generate the node representation learning, using Node2Vec, and apply dimensionality reduction, using PCA, to obtain the following 2-dimensions representation, for visual purposes.

We can see that the 2-dimensional node representation using Node2Vec tend to represent closer nodes that have similar importance in the network. However, this is only a partial view, due to the loss of information in the dimensionality reduction process.

The usage of node embeddings and, in particular, random walk-based node embeddings can enable the preservation of the network structure outperforming centrality measures-based models. In the case of the specific network defined in this work, the usage of node embeddings preserves not only proximity between nodes (first-order proximity), but also between neighbors of a given node (second-order proximity), which is relevant on this application in the sense that if we know the behavior of a given neighbor as well as its contacts, we are closer to predict the behavior of a new customer that enters the network. The pros of using this kind of approaches are that representing a network into a vector tends to preserve to a higher standard its properties. The cons are the high increase on the model's features, due to the embedding's dimensions.

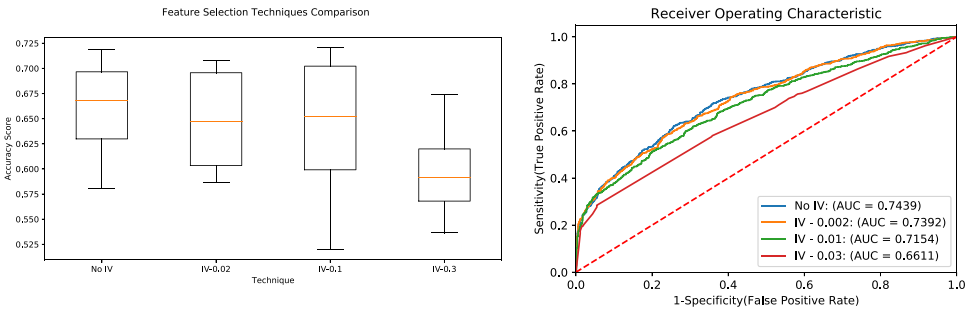
3.3 Friedman's test

When several models are to be compared, Demšar [18] suggests the usage of the Friedman's test [19]. This test is based on the comparison of performance ranks and the algorithms are ordered according with their performance (R_j). In case of a tie, mean position values are given to the tied algorithms. The null hypotheses states the algorithms are equivalent, meaning their performance ranks are similar. The calculated statistic is given by

$$F_F = \frac{(N-1)\chi_F^2}{N(A-1)-\chi_F^2} \tag{12}$$

where N represents the number of datasets, A the number of tested algorithms and

$$\chi_F^2 = \frac{12N}{A(A+1)}\left[\sum_j R_j^2 - \frac{A(A+1)^2}{4}\right] \tag{13}$$



(a) Accuracy scores

(b) AUROC scores

Fig. 4 Feature selection techniques scores

The null hypotheses is rejected if the calculated statistic is greater than the Fisher–Snedecor distribution with $A-1$ and $(A-1)(N-1)$ degrees of freedom, $F_{A-1,(A-1)(N-1)}$.

Friedman's test does not indicate which of the algorithms is better. In order to obtain that information, one should apply a post hoc test. In this, two algorithms are considered to be significantly different if their mean rankings difference is bigger than the so called Critical Difference (CD), where

$$\mathrm{CD} = q_\alpha \sqrt{\frac{A(A+1)}{6N}} \tag{14}$$

In Eq. (14), q_α is the statistic of the applied post hoc test, for a given significance level α.

4 Experiments

In this section, we describe the experiments and the conclusions obtained in order to define the baseline model. First, we compare performing feature selection using IV with different thresholds with not to perform it. Then, we test some resampling techniques in order to balance our dataset and compare the results with the imbalanced dataset. Lastly, we test the performance of some classification algorithms using different metrics.

4.1 Feature Selection using Information Value

There are different reasons why one should consider to perform feature selection on a given model. Reducing the number of parameters in the model makes it simpler and easier to interpret. It can help to produce faster and more effective predictors, as with less features we can decrease the training time of the model. Finally, we can avoid the curse of dimensionality and enhance generalization by reduce overfitting.

In this paper, we compare the results of performing feature selection with IV with not to do feature selection (corresponding to maintaining the 72 initial features of the dataset). In the case of performing feature selection, we test different thresholds for the IV, in order to select features with IV greater than the threshold. The selected values (0.02 - with 61 selected features, 0.1 - with 21 selected features and 0.3 - with 3 selected features) were chosen due to their relevance in classifying the predictive power of a given user, according to the rule of thumb proposed by Siddiqi [13]. Using accuracy, measured over a tenfold cross-validation, and the AUROC scores we obtain the results presented in Fig. 4.

From the Boxplots' analysis (Fig. 4a), we can observe some differences between the compared techniques, being the results from feature selection with a 0.3 threshold worse than the remaining. Comparing the AUROC of different techniques (Fig. 4b), we can conclude that not to perform feature selection with IV gives us the best results. Once again, doing feature selection with the threshold 0.3 performs worse in terms of the AUROC score.

Applying the Friedman's test, on the accuracy scores, we obtain a statistic of 12.59 and a p-value of 0.0056 meaning that we should reject, at a 5% significance level, the hypotheses that there are no differences between them. Observing the Friedman's post hoc tests, in Fig. 5 (graphical representations generated using Orange [20] Python's package), we can conclude that the model performing feature selection using IV with a 0.3 threshold is significantly worse than the other models, as shown in Fig. 5a, and the only one significantly different with not to perform feature selection, as observed in Fig. 5b.

In conclusion, comparing the tested models we can observe that there are no significant differences between not to perform feature selection with IV and doing with 0.02 and 0.1 thresholds. Therefore, for the baseline model, we should choose the most simple model, that is, to perform feature selection using a 0.1 threshold from the IV, maintaining 21 of the 72 total features (29%).

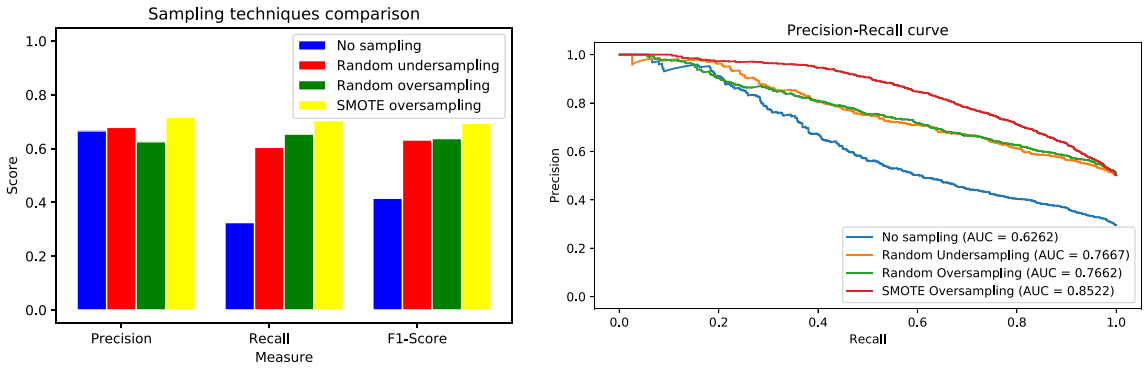
4.2 Resampling Techniques

When we have an imbalanced dataset, there are different ways we can balance it, for example, using undersampling or oversampling techniques. In this paper, we test different resampling techniques for balancing the dataset (random undersampling, random and SMOTE oversampling) and

(a) Nemenyi

(b) Bonferroni-Dunn

Fig. 5 Friedman's post hoc tests for the feature selection techniques



(a) Precision, recall and f1-scores

(b) Precison-recall curves

Fig. 6 Sampling techniques scores

compare them with the imbalanced dataset, using precision, recall and f1-score calculated on tenfold cross-validation and the area under the Precision–Recall curve, obtaining the results presented in Fig. 6.

We can conclude that, in all selected measures, SMOTE oversampling gives us the best results. We can highlight the poor recall score (and consequently f1-score) in the not resampled model, as shown in Fig. 6a. Once again, observing the Precision–Recall curves in Fig. 6b we can infer that not to balance the dataset produces the worse results and performing SMOTE oversampling produces the best results.

Performing a Friedman's test, using the f1-score, we obtain a 17.64 statistic and a 0.0005 p-value, meaning that there are significant differences between the models. The post hoc tests presented in Fig. 7 enable us to conclude there are significant differences between the model with the imbalanced dataset and all the other, being the model with SMOTE oversampling the one with the best f1-scores, however, with no significant differences with the other sampling techniques. Because we have a sufficient number of examples in our dataset, we choose to perform random undersampling, for the baseline model.

4.3 Classification algorithms comparison

To select the best-suited algorithm for the problem, we compare

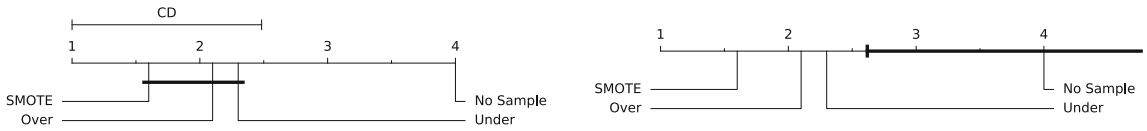
different classification algorithms: Logistic Regression, Decision Trees, k-Nearest Neighbors, Naive Bayes, Random Forest, Support Vector Machines and Extreme Gradient Boosting. We use, for that comparison, both accuracy and f1-score using a tenfold cross-validation and compare AUROC scores. The results are shown in Fig. 8.

We can conclude there are not, apart from Naive Bayes which has a very low f1-score, big differences between the tested algorithms. However, Logistic Regression has the best AUROC score.

Applying the Friedman's test, we obtain a 31.58 statistic and a 0.00002 p-value, meaning that there are significant differences between the tested models. The post hoc tests results are shown in Fig. 9. We can conclude that Naive Bayes gives us the worse accuracy scores and there are not significant differences between the other algorithms. However, Support Vector Machines, Random Forest, Logistic Regression, Extreme Gradient Boosting and k-Nearest Neighbors are the algorithms with the best results. We choose to apply the Logistic Regression algorithm, for the baseline model, for its relevance in credit scoring and because it is the one that gives us the best AUROC score.

5 Problem formulation and data description

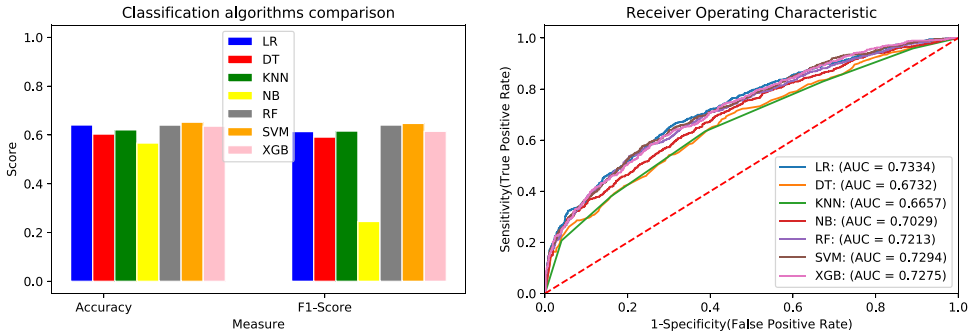
In this problem, we want to predict whether a customer of a MFI, based on the Sub-Saharan region, is to default on a loan. For that purpose, we use a dataset granted from the MFI, consisting of 11,486 loans, 7992 of which have been repaid on time (these were considered "good" customers), whether



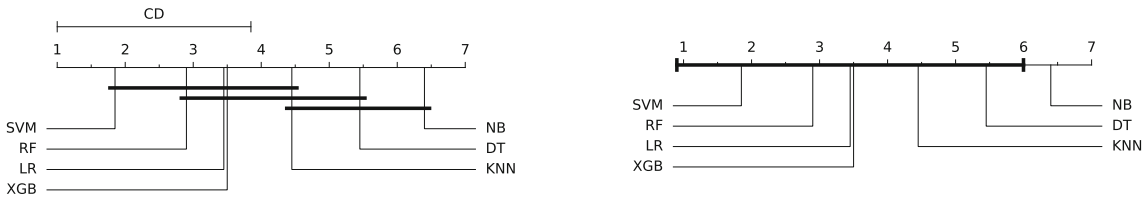
(a) Nemenyi **(b)** Bonferroni-Dunn

Fig. 7 Friedman's post hoc tests for the sampling techniques

Fig. 8 Classification algorithms scores



(a) Accuracy and f1-scores **(b)** AUROC scores



(a) Nemenyi **(b)** Bonferroni-Dunn

Fig. 9 Friedman's post hoc tests for the classification algorithms

the remaining 3,494 have failed on repaying the loan (considered "bad" customers). In this context, the criterion used for classifying a client as a "good" customer is the ability to pay its loan up to five days after the due date. Some empirical analysis show that most of the loans are repaid within five days after the due date, which is related to the payment methods available to the customers, can take up to 48 hours to be registered. Also weekends or official holidays can delay the registration of the loan repayment, appearing that the customer has overdue when it has not. There is no previous information on the customer, as we have only have considered its first application. The dataset has been balanced using undersampling on the majority class. (see Sect. 4.2)

Other than the target variable (binary variable which evaluates whether a customer is "good"/"bad," under the given criterion), there were 72 features present in the dataset, comprised of Socio-Demographic features such as gender, age or marital status; Loan-Related features such as amount, length or reason for applying to the loan and Mobile-Related features such as the device model, number of mobile applications installed or number of SMS. More specifically, socio-demographic data are gathered when the customer creates a profile from the app. Furthermore, mobile characteristics and information about SMS has been gathered. Finally, information about the loan itself has been included and some features have been engineered.

An IV feature selection method has been applied in order to select the most relevant features. In this sense, 21 features (considering $IV > 0.1$) have been selected, which are shown in Table 3, grouped by source (see Sect. 4.1).

As we can see, there were only 3 socio-demographic features selected, concerning the age, state and months of employment of the customer. This source of data is commonly used in credit scoring, as for example, in [4,7]. As for the loan-related features, we can observe the presence of monthly income, and the bank credit of the customer (total and in the last 30 days). There was also been performed feature engineering with the amount of the loan feature, namely

$$\text{rs ratio} = \frac{\text{amount of the loan}}{\text{monthly income of the customer}}, \tag{15}$$

$$\text{da ratio} = \frac{\text{amount of the loan}}{\text{length of the loan}}, \tag{16}$$

Table 3 IV Selected features

Socio-demographic	Mobile-related	Loan-related
State	Total number of SMS	rs ratio
Age	Number of bank SMS	Monthly Income
Months of Employment	Presence of a certain app	da ratio
	Number of default related words	credit 30 days
	Number of competitors SMS	ac30 ratio
	Number of loans app	total credit
	Number of apps installed	
	Percentage of bank SMS	
	Percentage of competitors SMS	
	Number of telecoms SMS	
	average words of incoming SMS	
	average words of outcoming SMS	

$$\text{ac30 ratio} = \frac{\text{amount of the loan}}{\text{credit in the last 30 days}}. \quad (17)$$

Most of the IV selected features were mobile-related, namely app-related features and SMS related features, as in [3,8,9]. Regarding the app-related features, we can highlight the presence of the number of installed apps, the number of installed competitors' apps and the presence of the installation of a loan competitor's app. The majority of selected features, however, were SMS related. Some took into account the average number of incoming/outcoming words in an SMS or the number of default related words. Other were related to the number/percentage of SMS itself, as the total number of SMS sent/received or the percentage of bank's/loan competitors' SMS.

6 Node embedding algorithm choice and relevance

We test different node embedding algorithms and the centrality model, in order to verify if there are significant differences between them and in comparison with the baseline model (adding those features to the ones from the latter). In order to achieve that, we apply Friedman's test to compare the accuracy scores of the different network models using the folds of the tenfold cross-validation. Applying this test to our dataset, using the centrality model and the given node embedding algorithms:

- Node2Vec [17],
- DeepWalk [16],
- NetMF [21],
- NMF-ADMM [22],
- GraRep [23],
- BoostNE [24]

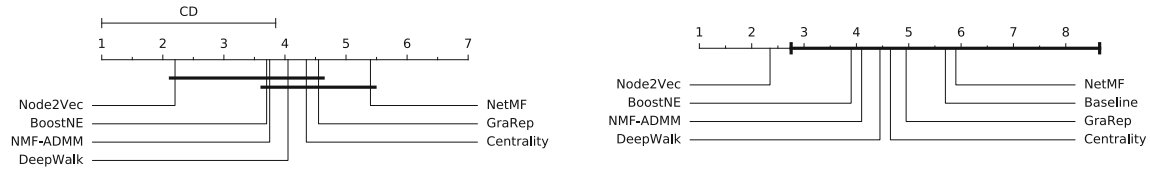
we obtain a statistic of approximately 12.61 and a p-value of approximately 0.0496 meaning we should reject the null hypotheses that there are no significant differences between the algorithms, at a 5% significance level. We use the implementations of the node embedding algorithms from the KarateClub [25] Python's package.

We apply the Nemenyi post hoc test [26] in order to verify which of the algorithms are significantly different and concluded that Node2Vec, BoostNE, NMF-ADMM, DeepWalk, Centrality and GraRep are not, as well as NetMF, GraRep, Centrality, DeepWalk, NMF-ADMM and BoostNE, as observed in Fig. 10a. To know if there is a significant difference between a node embedding algorithm/centrality model and the baseline model, we apply the Bonferroni–Dunn post hoc test [27] and conclude that the performance of the model using the Node2Vec algorithm is the only one significantly different (for better), comparing to the baseline model, as shown in Fig. 10b.

Therefore, in the last model considered in our study, we have used the Node2Vec [17] algorithm, proposed by Grover and Leskovec, to learn graph representations of nodes and add these new features to the ones from the baseline model, for comparison. In this algorithm we have used 128 as the dimension of the embedding space, 30 as the length of the random walks and 200 as the number of walks per node. We also performed hyper-parameter tuning on p and q to optimize the results ($p = 2$ and $q = 1$ were the optima parameters found over the set $\{0.25, 0.5, 1, 2, 4\}$).

Observing the results from Fig. 10 we can infer that, in general, random walk-based methods tend to perform better, comparing with matrix factorization-based approaches, being consisting with several state-of-the art studies (see, for example, Empirical Comparison of Graph Embeddings for Trust-Based Collaborative Filtering).

At a more fine-grained level, comparing random walk-based approaches, Node2Vec addresses the limitations of



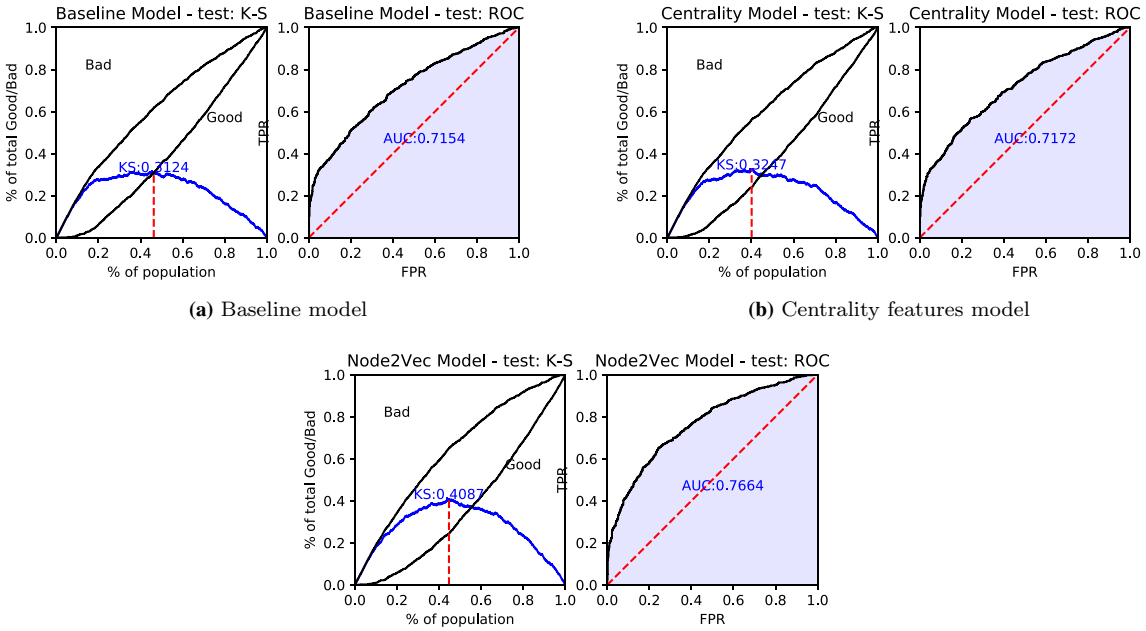
(a) Nemenyi **(b)** Bonferroni-Dunn

Fig. 10 Friedman's post hoc tests for the node embedding algorithms

Table 4 Comparison of different models

Model	K-S	AUROC	Mean accuracy
Baseline Model (1)	0.3124	0.7154	0.6450
Centrality Features Model (2)	0.3247	0.7172	0.6506
Node2Vec Model (3)	**0.4087**	**0.7664**	**0.6820**
Improvement (%) of (2) relative to (1)	3.94%	0.25%	0.87%
Improvement (%) of (3) relative to (1)	**30.83%**	**7.13%**	**5.74%**

Bold values refer to the best results in that column

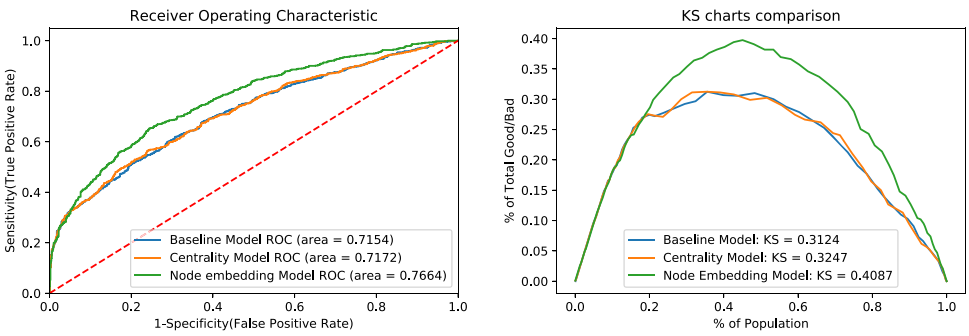


(a) Baseline model **(b)** Centrality features model

(c) Node2Vec model

Fig. 11 K-S chart and AUROC from the considered models

Fig. 12 Models comparison



(a) ROC-AUC curves **(b)** K-S charts

unweighted random walk approaches, such as DeepWalk, controlling the random behavior of the walks, through the interpolation between classic search strategies BFS and DFS, with the introduction of the hyperparameters p and q. With this, we can capture a richer kind of information on the embedding through the second order random walks and increase flexibility, capturing not only local, but also global information, resulting in better results on the performed experiments.

7 Performance measures and results analysis

To evaluate the performance of the models used in this study, we train the same algorithm (Logistic Regression—see Sect. 4.3) with the same hyper-parameters ("L1" penalty term, inverse of regularization strength $C = 1$, a SAGA optimization problem solver and a "one-vs-rest" multi-class term) and a 70%-30% train-test split.

In order to measure the model's performance, we plot the Kolmogorov–Smirnov (K-S) chart, which measures the degree of separation between the positive and negative distributions. In this sense, the K-S is 1 if the model partitions the population into two separate groups and 0 if the model can't differentiate between positive and negative classes. Thus, the higher the value is, the better the model is separating between the positive and negative cases. This value is calculated from the biggest difference between the cumulative counts of positive and negative classes of a certain percentage of the population. AUROC score is also calculated to access the quality of the predictions. Finally, we calculate the mean accuracy score, using tenfold cross-validation.

Comparing the results obtained from different models, in terms of the proposed performance metrics, the AUROC score, the Kolmogorov–Smirnov statistic and the mean accuracy, using tenfold cross-validation we can see, from Table 4, that the best model is the one that uses the Node2Vec algorithm, in all of the considered metrics.

Comparing the baseline model to the model adding the centrality features, we can observe that the results of the latter are marginally better. Indeed, there are slight improvements in the K-S statistic score, but the AUROC score and the mean accuracy score are similar to the ones from the baseline model.

Considering the Node2Vec model, and comparing it to the baseline model, we can observe a marginal improvement in the mean accuracy score, a better relative improvement in the AUROC score and a relative improvement of around 30% in the Kolmogorov–Smirnov score. In Fig. 11, we can see graphical representations of the KS and AUROC scores, for those three models. Regarding the Kolmogorov–Smirnov charts, the "Good" term is applied concerning loans that have been repayed on time and the same reasoning for the "'Bad" term.

In Fig. 12, we can see graphical representations, useful for a proper comparison between the tested models. Figure 12a represents the ROC curves and Fig. 12b the K-S graphs and respective scores. With the obtained results, we can infer that the model adding Node2Vec features to the ones from the baseline model is able to differentiate better between "good" and "bad" customers.

8 Ethics and privacy concerns

A main issue in emerging markets is the lack of a verifiable customers' credit history that can prevent their access to financial services. At this matter, smartphone data have great potential in improving customers' credit scoring, contributing to their financial inclusion which, otherwise, with traditional techniques, would be excluded. However, there are ethical concerns regarding the usage of personal information from smartphones, as it could constitute discrimination and prevent loan's accessibility by customers. Therefore, the usage of this kind of data must agree with ethical and privacy regulations. The usage of fully anonymized data and smartphone data only when it has a positive impact on loan's accessibility by customers could address this issues.

9 Conclusion

In this study, we worked with a dataset from a MFI based on the Sub-Saharan region, generated a network of relationships between phone book contacts of a given user and extracted two types of features from the network: centrality measures and node embedding features. Considering a model using WoE binning and IV feature selection, undersampling the majority class and using a Logistic Regression classifier as baseline, we were able to conclude that the model adding the node embedding features, to the ones from the baseline model, outperforms the other models considered in the study.

Applying the Friedman's Test, we could conclude that the node embedding algorithms tested are significantly different, and using the Nemenyi post hoc test we could verify which were the different algorithms. Moreover, using the Bonferroni–Dunn post hoc test, we were able to conclude that the Node2Vec node embedding algorithm is the only one significantly better (at a 5% significance level).

Although this paper proves, for this particular dataset, the relevance of the usage of node embeddings as features in credit scoring classification problems, other types of approaches, such as Deep Learning approaches, could bring value to this kind of problems. Also, the introduction of tem-

poral networks should be interesting in capturing different kinds of similarities between nodes.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. How digital finance could boost growth in emerging economies, www.mckinsey.com/featured-insights/employment-and-growth/how-digital-finance-could-boost-growth-in-emerging-economies, 17 Feb 2020
2. How M-Shwari Works: The story so far, www.cgap.org/research/publication/how-m-shwari-works-story-so-far, 17 Feb 2020
3. Ruiz, S., Gomes, P., Rodrigues,L. , Gama, J.: Credit Scoring in Microfinance Using Non-traditional Data. In: EPIA conference on artificial intelligence. Springer, Cham (2017)
4. Schreiner, M.: Credit scoring for microfinance: can it work? J Microfinance ESR Rev **2**(2), 6 (2000)
5. Bumacov, V., Ashta, A., Singh, P.: The use of credit scoring in microfinance institutions and their outreach. Strateg Change **23**(7–8), 401–413 (2014)
6. Van Gool, J., Baesens, B., Sercu, P., Verbeke, W.: An analysis of the applicability of credit scoring for microfinance. Belgium. University of Southampton, Southampton, United Kingdom, Katholieke Universiteit Leuven Leuven (2009)
7. Sousa, M.R., Gama, J., Brandão, E.: A new dynamic modeling framework for credit risk assessment. Expert Syst Appl **45**, 341–351 (2016)
8. San Pedro, J., Proserpio, D., Oliver, N.: MobiScore: towards universal credit scoring from mobile phone data. In: International conference on user modeling. Adaptation, and personalization. Springer, Cham, pp. 195–207 (2015)
9. Björkegren, D., Grissen D.: Behavior revealed in mobile phone usage predicts loan repayment., arXiv preprint arXiv:1712.05840 (2017)
10. Wei, Y., Yildirim, P., Van den Bulte, C., Dellarocas, C.: Credit scoring with social network data. Mark Sci **35**(2), 234–258 (2016)
11. Misheva, B. H., Giudici P., Pediroda V.: Network-based models to improve credit scoring accuracy., In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). IEEE, pp. 623–630 (2018)
12. María, Ó., Bravo, C., Sarraute C., Baesens B., and Vanthienen, J., "Credit scoring for good: Enhancing financial inclusion with smartphone-based microlending." arXiv preprint arXiv:2001.10994 (2020)
13. Siddiqi, N.: Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring, 3rd edn. Wiley, New Jersey (2012)
14. Shichen, X.: Scorecard Development in Python, GitHub repository, GitHub, https://github.com/ShichenXie/scorecardpy (2020)
15. Newman, M.: Networks. Oxford University Press, New York (2018)
16. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations., In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710 (2014)
17. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International conference on Knowledge discovery and data mining. pp. 855–864 (2016)
18. Janez, D.: Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res **7**, 1–30 (2006)
19. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc **32**(200), 675–701 (1937)
20. Bioinformatics Laboratory, FRI UL.: Orange Development in Python, GitHub repository, GitHub, https://github.com/biolab/orange3 (2020)
21. Jiezhong, Q., Yuxiao, D., Hao, M., Jian, L., Kuansan, W., Jie, T.: Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In: Proceedings of the eleventh ACM international conference on web search and data mining, pp 459–467 (2018)
22. Dennis, L., Sun, Cedric F.: Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 6201–6205 (2014)
23. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM international on conference on information and knowledge management, pp 891–900 (2015)
24. Li, J., Wu, L., Guo, R., Liu, C., Liu, H.: Multi-level network embedding with boosted low-rank matrix approximation. In: Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining, pp 49–56 (2019)
25. Benedek, R.: KarateClub Development in Python, GitHub repository, GitHub, https://github.com/benedekrozemberczki/KarateClub (2020)
26. Nemenyi, P.: Distribution-free multiple comparisons. Biometrics **18**(2), 263 (1962)
27. Dunn, O.J.: Multiple comparisons among means. J Am Stat Assoc **56**(293), 52–64 (1961)
28. Gama, J., Carvalho, A.D.L., Faceli, K., Lorena, A.C., Oliveira,M.: Extração de Conhecimento de Dados. data mining (3rd Edition) Silabo, (2015)