Vol: XX Issue: XX
International Journal of Data Science and Advanced Analytics (ISSN: 2563-4429)

Probability Graphical Model for Predicting Probability of Default For Mortgage Loans

Trupti Wagh[1], Jolnar Assi[2] and Ammar H Mohammed[3]
[1, 2, 3]Liverpool John Moores University, UK, Traders Island Ltd, UK, Iraqi Prime Minister's Office, UK
[1] trupti96@gmail.com, [2]jaa06@hotmail.com, [3]ammarhussen659@gmail.com
Corresponding author email: trupti96@gmail.com

Abstract*—Assessment of Default risk of borrowers is important for lending institutions as it directly affects profits and losses of the firm and guides in compensating the risk by taking appropriate majors for loans having higher probability of default. Predicting probability of default using statistical and machine learning models has been a popular research topic in data science community. While different types of classification models have been proposed historically, there is scope to apply probabilistic inference to the mortgage default analysis to support decision making. Probabilistic Graphical Model (PGM) are a powerful framework for compactly encoding probability distributions over complex multivariate domains using graphical representations. Due to the high interpretability and inherent support for probabilistic inference, the PGM models have widely been used under various domains such as medical diagnosis, text, audio, video processing. However, the causal and evidential reasoning capabilities of the PGM framework are not fully applied to the domain of credit scoring and for inferring probability of default, more so for large-scale real-life datasets. This study successfully built Bayesian Network as a PGM on the real-life mortgage loan dataset from Fannie Mae, USA to demonstrate inference and querying capabilities of the graphical models to render useful insights for decision makers in the lending institutions. As the percent of defaults has fallen in recent past, the dataset is highly skewed. This study used under-sampling and over sampling methods to get more balanced representation of the dataset with respect to Last Status of the loan as a multinomial target variable. Data-driven approach was used for the structure learning of PGM, to get important insights about relationship of variables in the Train dataset. Further, a custom Bayesian Network was built using hybrid approach by manually specifying the structure and using parameter learning algorithms for fitting parameters. The custom-built Bayesian Network was used to get probability of default on the mortgage loan Test dataset to enable comparison of the PGM with Logistic Regression as a benchmark classification algorithm. The custom- built PAG gave better predictive performance on the imbalanced Test dataset as compared with Logistic Regression. Usage of the Last Status variable allowed to capture interdependencies between variables at more granular level leading to the improved performance of the model. The custom-built Bayesian Network was utilized to demonstrate causal and evidential querying capabilities of the PGM framework on the mortgage loan dataset, exploring local independencies in the network. Thus, PGM with its powerful probabilistic inference framework was proved to be a practically useful tool for the mortgage default analysis.
***Keywords*—Default Risk; Machine Learning; Probability; Probabilistic Graphical Model; Classification**

1. Introduction

Predicting mortgage loan is important to the financial sector globally. Specifically in the US, mortgage industry is a major part of the financial sector [1]. In the US, lenders are required to follow accurate and strict controls for approval of mortgages. In this respect, predicting default is crucial where default is the probability of a borrower repaying the loan taken [2].

As such, lending institutions are expected to assess credit risk for loan applications where assessing risks informs about profits or losses of a credit firm [3]. If a probability of default is above a certain threshold, then the lender rejects the loan application and chargers higher interest rate to compensate for potential future losses [4].

Machine learning algorithms (MLAs) have been utilized for predicting probability of default predictions [5-7]. Predicting default has always been considered as a classification problem and clustering or statistical algorithms have been used [8-10]. Common classifier algorithms used in such predictions were logistic regression (LR) and support vector machines where these two algorithms are considered benchmarks for predicting probability of default [8-10].

More recently, deep learning algorithms (DLAs) have been deployed for predicting default yet their use have not been popular due to issues linked to their interpretability [11-13].

MLAs applied to default predictions included linear discriminant analysis (LDA), LR, naïve bayes, K-nearest neighbour, decision trees, random forests (RF), boosting, bagging, support vector machines (SVM), neural networks (NN), restricted Boltzmann machines, deep belief neural networks [14].

LDA is a dimensionality reduction technique that has been historically applied for supervised classification problem, including credit scoring. LDA assumes normal distribution of each feature in the dataset. However, this assumption may not always hold well in practice [15]. In non-linear cases, SVM has been commonly used for credit scoring, and risk assessment of loan datasets [16-18]. SVM showed high accuracy for predicting credit scoring when combined with K-mean clustering [19]. In another study, adaptive learning boost (AdaBoost) surpassed other MLAs (e.g. LR, RF) in predictions [20].

A limitation of MLAs is the requirement of large volumes of data for training. Moreover, MLAs and DLAs require values for all feature variables. Most of the developed models in the literature were made with synthetic data and very few studies were applied to real-life datasets [13].

In case of limited size datasets, statistical models offer an accurate alternative to ML-based classification models. Hence PGM can work on small datasets because it is a generative model [21]. In particular, probabilistic graphical model (PGM) is founded around Bayes' theorem and has outperformed Bayes' theorem in classification [22,23]. PGM has been used in many applications related to medical diagnosis [23-26]; speech recognition [27,28] and genetics [29,30]. The popularity of PGM in such applications is related to its opacity and interpretability. Graph-based PGM allow understand complex

multivariate systems that have complex amount of interpretability. PGM also enables reasoning the probability of a query variable given observations of more than one evidence variables.
Therefore, the present research proposes the application of PGM for predicting default comparing it to LR. The study applies PGM to dataset of single-family mortgage loans from Fannie Mae (USA) that was accessed from Kaggle [30].

2. Materials and Methods

2.1. Dataset

The dataset used was obtained from Kaggle open access platform and was the Fannie Mae loan acquisition and performance dataset on a subset of its Single-Family mortgage loans acquired from 2009 till last quarter [30]. The loan population contains two datasets, a primary dataset, and the HARP dataset. The original primary dataset containing loan acquisition and performance data from 2000 till 2012 was released by Fannie Mae in year 2012. For every quarter, Fannie Mae provides acquisition and performance data as of last quarter for all loans acquired since 2009 till date. The latest available update for the primary dataset is from 2022 Q2. The study will use the loans originated through years 2018 to 2022 for the study. The HARP dataset contains data of loans that were acquired by Fannie Mae between year 2000 to 2015 and refinanced via the HARP program. As the HARP program was not active during the time window under scope of this study, the HARP dataset will be excluded from scope.

2.2. Data pre-processing

Data pre-processing commenced with univariate analysis in order to understand the distribution of variables in the dataset.
The statistical summary dataset showed 74 columns, of which 13columns were of type integer, 35 columns are of type float and 26 columns are of type object. Variables that were not related to the scope of the study were excluded from the dataset.

2.3. Feature engineering

The statistical summary dataset has multiple variables related to credit events and impact\loss caused by the credit events. The field 'LAST_STAT' gives last status of the loan. Loans for which last status falls in either of below values are considered as defaulted. Table 1 gives list of status values that decide if loan is defaulted.

Table 1. Criteria of Loan Default.

Last Status code	Last Status Description
F	Deed-in-Lieu; REO Disposition
S	Short sale
T	Third party sale
N	Notes sale
9	270 Day Delinquency / 270+ Day Delinquency
8	240 Day Delinquency
7	210 Day Delinquency
6	180 Day Delinquency

In addition, a new Boolean variable named 'Default' was added for which value will be set as True when Last Status of the loan matches any of values from Table 1. Both 'Default' and 'Last Status' variables could be considered as Target variables. In case of PGM, both the target variables could be included as nodes of Bayesian Network. For LR, 'Default' will be considered as the target variable.

Date fields such as Origination Date, Maturity Date, Zero Balance Effective Date needed to be discretised in the form of number of quarters passed from an appropriate date taken as a reference date. E.g. for the period covering years 2018 to 2021, if '1st January 2018' was considered as the reference date, then the date variables would be encoded as quarters with range 1 to 16. Fields such as State code, Zip code would be mapped to regions in the USA and used as a categorical variable. Other Feature variables would be created as required during the implementation.

2.4. Train and test variable

Splitting the primary dataset into train and test sets, after initial pre-processing steps were completed. The Train Test split needs to be conducted prior to re-balancing the dataset. Splitting dataset was not a necessity for PGMs, however it was required to evaluate prediction performance with respect to the LR classifier model.

2.5. Addressing class imbalance

Class imbalance was addressed in the dataset where the dataset was highly skewed. Hence, the percentage of default loan was below 1% of the total loans. In this respect, class rebalance was applied using under-sampling of the majority class and synthetic samples generation (SMOTE) for the minority class.
The study proposed including Last Status variable in PGM, to allow querying probability of a loan entering any of the status values. Hence, it was crucial to consider Last Status as the target variable while addressing class re-balance. As Last Status variable contained more granular information that what was contained in the Default variable, it offered best choice as a Target variable for re-sampling. This meant under-sampling and SMOTE would need to be performed for considering Last Status as the target variable.
Another consideration for re-sampling is that the dataset contained both integer and categorical variables. Hence, under-sampling and over sampling tools would be elected such that they support multi-class target variables as well as categorical random variables.

2.6. Exploratory data analysis

Exploratory data analysis was conducted on the resampled dataset to identify distribution with respect to default and non-default loans, visible patterns in the data and correlation between variables.

2.7. Data preparation for PGM and LR

Data preparation for PGM and LR was made by enriching the re-sampled train dataset by deriving calculated variables e.g. default that were needed for modelling, transformation of categorical variables into numeric variables.
PGM supported building Bayesian Networks on both discrete and hybrid (discrete + continuous) data. Hence, for PGM, numeric variables would be retained as-is in train dataset. Also, categorical counterparts for those numeric variables would also be included in the dataset. The study created discrete Bayesian Networks by including all variables as a categorical variable. In parallel, study attempted to get mixed Bayesian Networks by using combination of discrete and continuous variables.
In case of Logistic Regression, one hot encoding will be done for categorical variables. Also, feature transform and scaling were applied for numeric variables in the LR train dataset.

2.8. Network structure and parameter learning

Learning of Bayesian Network was done as a two-step process, that included structure learning followed by parameter learning. A score-based structure learning algorithms such as Hill Climb Search, Tabu was applied on the Train dataset using scoring functions such as BDe to get DAGs representing the Train dataset. In parallel, a constraint-based structure learning algorithms such as PC, Growth Shrink was applied to get DAGs for the dataset. Then calculation of DAGs scores was learned by the structure learning algorithms and evaluate the performance of the algorithms. Then the local probability distribution was calculated for the DAGs by fitting them on the dataset using parameter learning algorithms viz. Maximum Likelihood Estimates, Bayesian Posterior Estimation. Then queries were run on the fitted model for each DAG, to get probabilities of commonly known domain knowledge, to verify if the network is able to give practically relevant results.
This was followed by analysis of the structures (i.e., nodes and edge) learned by both approaches in the context of domain knowledge and understanding common patterns or visible dependencies to decide upon the edges and directed relationships to be selected for analysis.
Then a custom DAG was built using the selected nodes and directed relationships between them. The process was be carried out iteratively, by selecting just a few nodes / edges at the start and adding more nodes / edges based on evaluation of the custom DAG. Then local probability distribution was estimated using Maximum Likelihood Estimator and Bayesian Parameter Estimation.

2.9. Plot DAGs for Bayesian Network and explore CPTs

Plot the relationships learned by the Bayesian Network in the form of the DAGs, to explore the insights rendered by the model. Identify variables having major influence on the probability of default. An example of a DAG for the Bayesian Network structure using few variables is depicted in Figure 1.

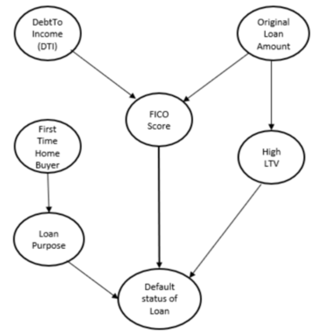

Figure 1. Sample Bayesian Network for the dataset.

The variables represented in circles are called nodes of the network and the arrows connecting nodes are called the edges. FICO Score is the credit score used by Fannie Mae. It will possibly have high influence on the loan default. Hence there is a direct relationship from FICO score to defaulted loan. When Debt to Income ratio of a borrower is on higher side, there is a high possibility that the same will be reflected in the FICO score. Also, the original loan amount approved by the lender has a positive relation with the FICO score. If the higher amount was approved for the loan, there is a possibility that Loan to Value ratio was higher for the loan. This will also have influence on the loan default. On other hand, if the borrower is a first-time buyer, it will direct the purpose of the loan. This can also influence the probability of default.
To understand the Conditional Probability Table, let's consider the distribution between FICO score and Loan Default status. To demonstrate the probability distribution for a discrete variable, consider that the FICO scores will be discretised as presented in Table 2 based on the range of values.

Table 2. FICO score level.

FICO score ranges	FICO score levels
300 – 579	Poor
580 - 669	Fair
670 – 739	Good
740 – 799	Very good
800 – 850	Exceptional

Table 2 shows that the FICO Score variable will take five levels, i.e. Poor, Fair, Good, Very Good, Excellent. The Loan Default Status will take two levels i.e., 0 and 1, wherein 0 means no-default and 1 means default. The conditional probability distribution between these two variables will take a form of a table as presented in Table 3.

Table 3. Sample CPT.

Loan default status	Loan default status	
	0	1
Poor	0.2	0.8
Fair	0.25	0.75

Good	0.45	0.55
Very good	0.65	0.35
Exceptional	0.2	0.2

Table 3 gives probability of default for each possible level of the FICO score. Such CPT tables will be learned by the Bayesian Network for every discrete node in the network, for all possible combination of values for its parents.

2.10. **Query probability default on the Bayesian network**
Infer probability of default for random samples in the PGM Train dataset using the causal reasoning capabilities of the model. The inference computations will be supported by the network based on the Conditional Probability Tables obtained as part of network structure learning and inference algorithms such as variable elimination.

2.11. **Perform evidential inference on the Bayesian network**
Explore evidential reasoning capabilities of the Bayesian Network model on the PGM Train dataset. For a loan default case, queries can be made to get probability of a certain influencing variable following under certain range. Various patterns of queries will be performed on the network to understand interdependencies of variables and understand their significance in the loan default scenarios.

2.12. **Build logistic regression classifier**
Build LR based classifier model on the train dataset while performing feature selection using Recursive Feature Elimination and Variable Inflation Factor (VIF). Predict probability of default on the train dataset. Identify optimal cut off point of probability by plotting sensitivity, specificity and accuracy against different values of probabilities. Evaluate the performance of the Logistic Regression on the Test dataset using metrics such as Accuracy and Confusion Metrics.

2.13. **Compare PGM with LR classifier**
As PGM and LR come from different families of machine learning models, direct comparison of PGM with LR is not possible. Hence, a custom approach needs to be taken to enable comparison of PGM with LR, as detailed out below.
LR falls under category of models called as Discriminative models, which supports predicting class on unseen data. The model is built on train dataset. Probability of default is calculated on the test dataset and the model's performance is evaluated by measuring the difference between the actual and predicted values.
PGM is part of category of models called Generative models, for which the primary use case is to infer probabilities, find hidden patterns from underlying distribution of data and support probabilistic inference. When PGM structure is learnt on the train dataset, it captures the discrete levels for each variable based on data distribution. Parameters of the network are again fitted based on underlying data distribution. Hence given PGM model works only for the dataset on which the structure learning and parameter learning was performed. It cannot be directely extended to predict probabilities on the test dataset. To evaluate performance of PGM with respect to benchmark algorithms, this study will re-build PGM on the Test dataset, using the same DAG that was identified to be best fitting for the train dataset. To support this, it will be assumed that the Train and Test datasets have similar probability distribution as they are originally drawn from the same population.
Given below are steps for this process:

- Build Bayesian Network on Test dataset using the DAG that was identified to be best fitting on train dataset.
- Re-perform parameter learning for this model on the Test dataset. This process captures the parameters of local distribution that are best fitted to test dataset.
- Perform queries to get probability and predicted values of Last Status for all loan applications in the PGM Test dataset. Get predicted value for the Default variable, using value of predicted Last Status. This gives actual and predicted values for the Default variable in binary format.
- As these values are binary, standard evaluation metrics can be applied to measure distance between actual and predicted values for PGM Test dataset. Examples of such metrics are Log Loss, Confusion Matrix including F1 Score
- Calculate the Log Loss and Confusion Matrix for the Logistic Regression on Test dataset.
- Compare values of metrics obtained with the PGM model and LR model on the Test dataset to document the findings.

3. Results

3.1. Evaluation of sampling method

Initially, oversampling was performed on the sample of primary dataset using SMOTE-NC algorithm to confirm that the approach works to get synthetic samples for minority classes of the multinominal Last Status variable, for dataset that includes both numeric and categorical variables. The primary statistical summary dataset had size of 14.4 million rows and 74 columns. Oversampling of this dataset would have increased the size of dataset even further posing challenge with respect to availability of hardware and speed of execution. Hence, the implementation of under-sampling was very important.
After pre-processing the primary dataset, Train – Test split was performed so as to proceed with re-sampling the Train dataset. To start with, under-sampling was performed using 'ROSE package, to balance the majority classes C and P, with respect to other minority classes. Table 4 shows distribution of classes in primary dataset and in under-sampled dataset. Count of the majority class datapoints C and P are largely reduced in the under-sampled dataset, as seen from the rows highlighted in green. On other hand, counts of all other minority classes has remained same.

Table 4. Results of under-sampling.

Target class	Count in Primary dataset	Count in under-sampled dataset
C	7630421	164974
P	3117257	57959

1	42041	42041
9	16094	16094
R	10237	10237
2	8429	8429
3	4477	4477
4	3256	3256
5	3181	3181
6	2567	2567
7	1979	1979
8	1758	1758
L	1580	1580
F	670	670
T	465	465
S	263	263
N	70	70

Thus, under-sampling helped to bring drastic reduction in datapoints of majority classes while not reducing datapoints of minority classes as well as keeping their percentage distribution intact. The under-sampled dataset was further processed and given as input to SMOTE-NC algorithm. An oversampling strategy was specified for the SMOTE-NC algorithm that included expected counts for each class in the re-sampled dataset.

Table 5 shows results of the SMOTE-NC algorithm. The approach helped to increase datapoints corresponding to minority classes, while keeping overall distribution of minority classes intact. Values of Last Status variable were encoded while giving as input to the SMOTE- NC algorithm. Table 5 specifies encoded values for the Last status variables. With this encoding, a loan is considered as defaulted when Last Status falls in 5, 6, 7, 8, 12, 12, 15 or 16.

Table 5. Results of SMOTE oversampling.

Target class	Count in under-sampled dataset	Encoded target class	Count in over-sampled dataset
C	164974	9	164974
P	57959	13	57959
1	42041	0	42041
9	16094	8	52000
R	10237	14	33487
2	8429	1	27572
3	4477	2	14645
4	3256	3	10650
5	3181	4	10405
6	2567	5	8397
7	1979	6	6473
8	1758	7	5750
L	1580	11	5168
F	670	10	2191
T	465	16	1521
S	263	15	860
N	70	12	228

Thus, over-sampling with SMOTE helped increasing datapoints of minority classes, while keeping percentage distribution of minority classes intact. Table 6 gives sizes of various datasets generated in the process of getting re-sampled train dataset, along with distribution of default and non-default loans for those datasets.

Table 6. Sizes of datasets.

	Primary dataset	Train dataset	Under-sampled train dataset	Balanced train dataset	Test dataset
Total datapoints	14459660	10844745	320000	444321	3614915
Non-defaults	14427811	10820879	296134	366901	36006932
Default	31849	23866	23866	77420	7983
Non-default (%)	99.77	899.77	92.54	82.57	99.77
Default (%)	0.22	0.22	7.54	17.42	0.22

Notice that process of under-sampling did not reduce the count of minority datapoints. On other hand, datapoints of majority classes were increased in the balanced Train dataset, with respect to the under-sampled Train dataset. Percentage of minority class was extremely low i.e. 0.22 in the primary dataset, which also reflects in the percentage distribution of Train and Test dataset.

In the under-sampled Train dataset, this percentage was increased to 7.45. With SMOTE over sampling, the percentage of Default class was further increased from 7.45 to 17.42.

Alternate settings were tried to increase percentage of minority classes even further. This necessitated that more datapoints of classes P and 1 be included in the target dataset during under-sampling, so as to keep distribution of all minority classes intact. This resulted in higher size of the Train dataset giving out of memory issues with the available hardware.

3.2. Results of structure learning

To learn DAG from the balanced train dataset, score-based and constraint-based algorithms were applied, initially on hybrid data that included discrete and numeric variables. Some challenges and limitations were encountered in applying the Bnlearn package to hybrid data as detailed out in sections below. Hence, final DAG was prepared on the dataset that had all variables converted into their discrete counterparts.

3.3. Parameter learning on train dataset

Maximum Likelihood estimation and Bayesian Posterior Estimates algorithms were used to fit DAGs on the balanced dataset. Predict method offered by bnlearn package was used to predict value of Last Status on the Train dataset, using the fitted networks. The Predict method ignores values of the target variable from given dataset and returns a new vector for the target variable, based on probability distribution of the fitted network.

The prediction was done for all discrete DAGs achieved with structure learning and manual inputs. As actual and predicted values were made available by the Predict method, it was possible to calculate metrics pertaining to information gain, using the actual and predicted values. Accuracy, Confusion Metrics and Log-Loss metrics were calculated for the networks fitted using both MLE and Bayesian Posterior Estimates algorithms. Table 7 shows the results of the parameter learning process. It is evident from the results that both parameter learning algorithms yielded similar metrics for given algorithm. For example, for a discrete DAG obtained with Hill Climb algorithm, the metrics achieved using MLE and Bayesian Posterior Estimates algorithm were exactly same. Thus, accuracy of both parameter learning algorithms was found to be same. While reviewing performance of the parameter learning algorithms, Maximum Likelihood estimation was found to be very slow compared to Bayesian Posterior Estimates. Behaviour of Maximum Likelihood estimation was unstable, leading to out of memory exceptions in some scenarios.

Bayesian Posterior Estimates was identified to be more efficient and stable algorithm, Hence, Bayesian Posterior Estimates algorithm was used for further evaluation of the Bayesian Networks on Test dataset.

Table 7. Performance of PGM on the train dataset.

	Parameter learning algorithm	Log loss	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-score (%)
HC	MLE	8.44	75.54	33.94	42.71	82.46	37.83
HC	Bayes	8.44	75.54	33.94	42.71	82.46	37.83
Tabu	MLE	6.64	80.75	39.78	20.39	93.48	26.96
Tabu	Bayes	6.64	80.75	39.78	20.39	93.48	26.96
Inter-IAMB	MLE	1.45	95.78	81.4	98.25	95.26	89.03
Inter-IAMB	Bayes	1.45	95.78	81.4	98.25	95.26	89.03
Custom built DAG	MLE	0.93	97.3	88.34	97.36	97.29	92.63
Custom built DAG	Bayes	0.93	97.29	88.3	97.36	97.28	92.62

Table 7 also helps evaluate performance of various fitted DAGs on the Train dataset. The DAGs achieved using Hill Climb, Tabu and Inter-IAMB algorithms yielded higher log-loss and poor values of Accuracy, Precision, Recall, Specificity and F1 score. This means they had poor goodness of fit and there is scope to fine tune the network structure and parameters further. The custom-built DAG was found to be giving best performance on the Train the lowest Log Loss, highest value of Accuracy, Precision, Recall, Specificity and F1 score.

3.4. PGM and LR evaluation on test dataset

Log Loss and Confusion Metrics were evaluated for both the final PGM model and LR model on the Test dataset. LR gave log loss of 1.3 and accuracy of 96.15% on train dataset; yet, it performed poorly on the test dataset. wrongly classified many of

the non-defaulted cases as defaulted. While the test dataset has 7648 default cases, the LR

model reported 34030 loans as defaulted. Hence, the Precision score was very low (22.5%), which reflected in the F1 score (36.4%). On other hand, PGM has delivered better values of Accuracy, Precision, Specificity and overall F1-score. The latter values were 99.9%, 84.4%, 99.9% and 75.3% respectively.

4. Conclusions

The present study, analysed the Fannie Mae Mortgage loan dataset to understand distribution of data and relationships between variables. Modelling for the mortgage loan dataset can be approached as a binary classification problem or multi-class classification problem. This study presented an approach to achieve both binary classification and multi-class classification with same setup, with some extra steps taken during re-sampling of data and dataset preparations.

Train Test split was done prior to proceeding with re-sampling of dataset. Class imbalance was addressed successfully utilizing mix of under-sampling and oversampling methods. Last Status was considered as a multinomial target variable for the re-sampling process. The dataset was under-sampled using ROSE package as it supports both numeric and categorical variables for under-sampling. ROSE is designed to work with binary target variables. Hence a special method was devised to enable under-sampling for multi-class target variable. The method involved under-sampling majority class with respect to one minority class at a time and combining resulting subsets of data. This method was found to be giving satisfactory results on the mortgage loan dataset.

SMOTE-NC algorithm was applied successfully on the under-sampled data to get synthetic samples for all minority classes, keeping percentage distribution of minority classes intact.

Percentage of Default cases was increased from 0.22% in the Train dataset to 17.42 % in the re-sampled Train dataset. Further increase in the default cases could not be achieved as a greater number of datapoints would need to be selected for some of the majority classes for keeping their ratio with minority class intact. This would lead to increasing size of the dataset as well as hardware and processing capabilities.

Further, separate datasets were created for model building each PGM and LR model building. For structure learning of PGM, data driven approach was sought to get idea of underlying data distribution and relationship of variables. Various score-based and constraint-based algorithms were used to get DAG for Bayesian Network. GM was able to deliver probabilities of various events, given different set of evidences.

References

[1] Fligstein, N., 2021. The banks did it: An anatomy of the financial crisis. Harvard University Press.

[2] Aslam, M., Kumar, S. and Sorooshian, S., 2020. Predicting likelihood for loan default among bank borrowers. International Journal of Financial Research, 11(1), pp.318-328.

[3] Bouteille, S. and Coogan-Pushner, D., 2021. The handbook of credit risk management: originating, assessing, and managing credit exposures. John Wiley & Sons.

[4] Oketch, J.R., 2020. Effect of Financial Sector Policies on Commercial Bank Performance in Kenya (Doctoral dissertation, JKUAT-COHRED)

[5] Liu, Y., Yang, M., Wang, Y., Li, Y., Xiong, T. and Li, A., 2022. Applying machine learning algorithms to predict default probability in the online credit market: Evidence from China. International Review of Financial Analysis, 79, p.101971.

[6] Kornfeld, S., 2020. Predicting Default Probability in Credit Risk using Machine Learning Algorithms.

[7] Jacobs Jr, M., 2024. Benchmarking alternative interpretable machine learning models for corporate probability of default. Data Science in Finance and Economics, 4(1), pp.1-52.

[8] Teles, G., Rodrigues, J.J., Rabelo, R.A. and Kozlov, S.A., 2021. Comparative study of support vector machines and random forests machine learning algorithms on credit operation. Software: Practice and Experience, 51(12), pp.2492-2500.

[9] Costa e Silva, E., Lopes, I.C., Correia, A. and Faria, S., 2020. A logistic regression model for consumer default risk. Journal of Applied Statistics, 47(13-15), pp.2879-2894.

[10] Dastile, X., Celik, T. and Potsane, M., 2020. Statistical and machine learning models in credit scoring: A systematic literature survey. Applied Soft Computing, 91, p.106263.

[11] Ozbayoglu, A.M., Gudelek, M.U. and Sezer, O.B., 2020. Deep learning for financial applications: A survey. Applied soft computing, 93, p.106384.

[12] Sezer, O.B., Gudelek, M.U. and Ozbayoglu, A.M., 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied soft computing, 90, p.106181.

[13] Ramezan, C.A., Warner, T.A., Maxwell, A.E. and Price, B.S., 2021. Effects of training set size on supervised machine-learning land-cover classification of large-area high-resolution remotely sensed data. Remote Sensing, 13(3), p.368.

[14] Dastile, X., Celik, T. and Potsane, M., 2020. Statistical and machine learning models in credit scoring: A systematic literature survey. Applied Soft Computing, 91, p.106263. Gasse, M., 2017. Probabilistic graphical model structure learning: application to multi-label classification.

[15] Eisenbeis, R.A., 1978. Problems in applying discriminant analysis in credit scoring models. Journal of Banking & Finance, 2(3), pp.205-219.

[16] Goh, R.Y. and Lee, L.S., 2019. Credit scoring: a review on support vector machines and metaheuristic approaches. Advances in Operations Research, 2019.

[17] Harris, T., 2013. Default definition selection for credit scoring. Artif. Intell. Res., 2(4), pp.49-62.

[18] Fitzpatrick, T. and Mues, C., 2021. How can lenders prosper? Comparing machine learning approaches to identify profitable peer-to-peer loan investments. European Journal of Operational Research, 294(2), pp.711-722.

[19] Yuan, K., Chi, G., Zhou, Y. and Yin, H., 2022. A novel two-stage hybrid default prediction model with k-means clustering and support vector domain description. Research in International Business and Finance, 59, p.101536.

[20] Guo, W. and Zhou, Z.Z., 2022. A comparative study of combining tree-based feature selection methods and classifiers in personal loan default prediction. Journal of Forecasting, 41(6), pp.1248-1313.

[21] Pernkopf, F., Peharz, R. and Tschiatschek, S., 2014. Introduction to probabilistic graphical models. In Academic Press Library in Signal Processing (Vol. 1, pp. 989-1064). Elsevier.

[22] Nguyen, T.M., Poh, K.L., Chong, S.L. and Lee, J.H., 2023. Effective diagnosis of sepsis in critically ill children using probabilistic graphical model. Translational Pediatrics, 12(4), p.538.

[23] Jiang, J., Li, X., Zhao, C., Guan, Y. and Yu, Q., 2017. Learning and inference in knowledge-based probabilistic model for medical diagnosis. Knowledge-Based Systems, 138, pp.58-68.

[24] Khademi, M. and Nedialkov, N.S., 2015, December. Probabilistic graphical models and deep belief networks for prognosis of breast cancer. In 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA) (pp. 727-732). IEEE.

[25] Gupta, A., Slater, J.J., Boyne, D., Mitsakakis, N., Béliveau, A., Druzdzel, M.J., Brenner, D.R., Hussain, S. and Arora, P., 2019. Probabilistic graphical modeling for estimating risk of coronary artery disease: applications of a flexible machine-learning method. Medical Decision Making, 39(8), pp.1032-1044.

[26] Bilmes, J.A. and Bartels, C., 2005. Graphical model architectures for speech recognition. IEEE signal processing magazine, 22(5), pp.89-100.

[27] Bilmes, J.A., 2003. Buried Markov models: A graphical-modeling approach to automatic speech recognition. Computer Speech & Language, 17(2-3), pp.213-231.

[28] Sinoquet, C., 2014. Probabilistic graphical models for genetics, genomics, and postgenomics. OUP Oxford.

[29] Mourad, R., Sinoquet, C. and Leray, P., 2012. Probabilistic graphical models for genetic association studies. Briefings in bioinformatics, 13(1), pp.20-33.

[30] Fannie Mae & Freddie Mac Database 2008-2018 ∑∞ (2024). Available at: https://www.kaggle.com/datasets/jeromeblanchet/fannie-mae-freddie-mac-public-use-database Accessed: 30-03-2024.