Credit Default Risk Prediction

by Swarupa Vijay

Submission date: 22-Nov-2021 09:23AM (UTC+0530)
Submission ID: 1709725463
File name: DA_literature_review_1.pdf (328.34K)
Word count: 4413
Character count: 21481

Credit Default Risk Prediction

Swarupa Vijaykumar
Computer Science
PES University
Bangalore, India
swarupav01@gmail.com

Suvigya Jain
Computer Science
PES University
Bangalore, India
suvigyajain1@gmail.com

Supreeth G Kurpad
Computer Science
PES University
Bangalore, India
supreethkurpad@gmail.com

Kishan Minna Murthy
Computer Science
PES University
Bangalore, India
kishanminna@gmail.com

***Abstract*—Credit plays a dominant role in our lives, in all industries that involve monetary investment. A bank loaner needs to analyze about their applicants to give a safe loan and determine which individuals are risky to lend money to. Due to this analysis, many people are denied loans on the grounds of insufficient credit information. The aim of our project is to accurately predict if an individual will repay the loan or not. We do this using the Home Credit Default Risk prediction dataset that was part of a $80,000 Kaggle competition. We designed an ensemble model using LightGBM and Artificial Neural Network which gave an ROCAUC score of 75% on the final test data submitted to the competition.**

I. Introduction

Credit risk is the risk that the lender will not be paid back all the principal amount as well as the interest on time as agreed upon by both the parties. Default risk is a type of Credit risk which is the probability that the borrower will fail to repay the loan amount/debt. When a borrower defaults, it leads to a loss for the bank. To prevent such losses, banks do credit risk modelling where they determine the risk involved in extending credit to a borrower. Credit Risk modelling can be done by using previous credit scores or financial statements analysis. When the lender decides that the risk involved in lending money to a certain individual/institution is high, they compensate for it by charging higher interest rates. When an individual has insufficient or non-existent credit scores, they are forced to pay higher interests or sometimes even denied loans. This turns them to untrustworthy lenders who exploit this population and charge very high interest.

To prevent this, we can use an alternative approach to credit risk modeling where we analyze the teleco and transaction information of individuals with insufficient credit score and predict how probable they are to repay the loan using machine learning methods. This is precisely what our project aims to do. We want to predict future payment behaviour of borrowers from application, demographic and historical credit behavior data. We also aim to reduce the training and prediction time of the model without compromising much on its performance. By doing so, the model can be scaled for extremely large data and can be deployed in the real world to get accurate predictions. We also want to make sure that the model can easily accommodate more details of a customers that would help to determine how capable they are of repaying loans.

II. Literature Review

One of the earliest methods proposed was using Support vector machines. However there are many new and sophisticated methods for the classification problem including Logistic regression, Random Forest and Neural Networks.[1] The LightGMB model outperforms the other 2 models and model performance is LightGBM >Logistic Regression >Random Forest. The best ROCAUC score was 78% from LightBGM and can be improved by using Deep forest or model stacking.The models also need to be verified with other datasets for robustness.

The accuracy of random forest was improved by balancing class distribution, and parameter optimization.[2]The parameters include number of trees, number of random variables in each tree, terminal node size and seed value.However, the focus of this paper was on only optimising hyper parameters based on which the hypothesis was given. This could be a biased opinion and may not work for all datasets.

Other models which can be used for the classification task include Naïve Bayes and Artificial Neural Networks [3]. Initally the data was divided into categories and the learning rate, epoch and number of neurons was determined by the best combination of parameters. The ANN approach is to estimate credit risk function and establish the most significant factors.The Naïve Bayes approach finds the probability of credit default when all variables have been measured.

In a comparative study of all types of machine learning algorithms which include single classifiers like logistic regression, decision tree, MLP etc, homogeneous ensembles like XGBoost, GBM, Random Forest and heterogeneous ensemble classifiers like Stacked Ensembles was done [4]. The models were also tuned for various hyper parameters and trained.On comparison of the ROCAUC scores, ensemble classifiers gave the top 10 best scores and stacked ensemble gave the best ROCAUC score of 0.70.The other best performing models include XGBoost, Gradient Boosting Machine (GBM), Deep Learning and Distributed Random Forest (DRF) models.

An imbalanced dataset is has a major impact on the efficiency of a classification model[5] and hence it is essential to balance the data. This was doing using re-sampling techniques. It was found that on the GDBT model, re-sampling improved performance and specifically oversampling methods were better than under sampling with best accuracy of 88.7%.

However the features of the dataset used in this model were less and model needs to be more generalised. In the current world, the amount of data available is huge and traditional machine learning models cannot extract meaning from the high dimensional data.[6] Hence it is essential to use deep learning models to capture these details. But the overall theoretical system of deep learning is not perfect and cannot be used in finance yet. Traditional machine learning models are also easier to deploy.

III. Dataset and Initial Insights

The dataset used for the project is from a Kaggle Competition- "Home Credit Default Risk" organised by the Home Credit Group. The dataset consists of 10 files which include a file for the description of columns and a file sample submission format.

Data that is to be used for model building is present in 8 files and includes over 200 columns.

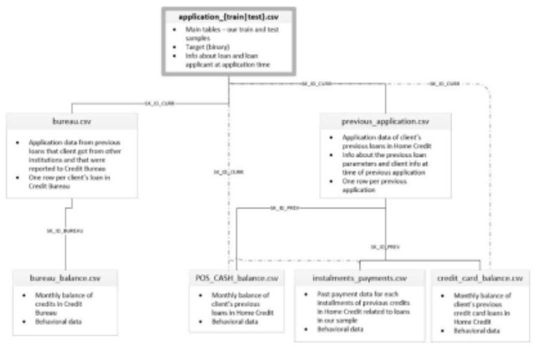
application_{train|test}.csv
Main tables – our train and test samples
Target (binary)
Info about loan and loan applicant at application time
SK_ID_CURR
bureau.csv
Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
One row per client's loan in Credit Bureau
previous_application.csv
Application data of client's previous loans in Home Credit
Info about the previous loan parameters and client info at time of previous application
One row per previous application
SK_ID_BUREAU
SK_ID_PREV
bureau_balance.csv
Monthly balance of credits in Credit Bureau
Behavioral data
POS_CASH_balance.csv
Monthly balance of client's previous loans in Home Credit
Behavioral data
instalments_payments.csv
Past payment data for each installments of previous credits in Home Credit related to loans in our sample
Behavioral data
credit_card_balance.csv
Monthly balance of client's previous credit card loans in Home Credit
Behavioral data

Fig1: Entity relation of files

On finding the number of positive and negative samples in the dataset, it was found that the data is highly imbalanced.

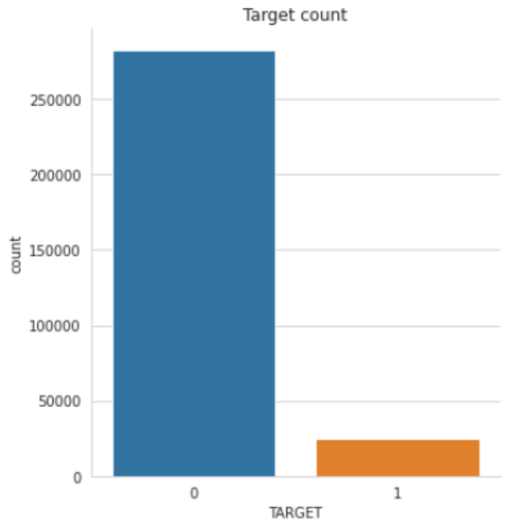

Fig2: Histogram plot of target classes

To find out the non numeric features that would help us most to differentiate between defaulters and non defaulters, cat plots for all such features were made. By looking at the plots, it can be inferred that the distribution of classes in both defaulters and non defaulters was the same and no differentiation can be done based on non numeric variables.

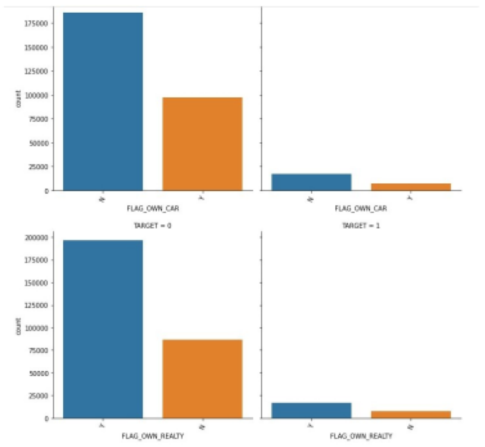

Fig4: Catplot of features for each target class

Similarly a density plot was made for all the quantitative variables and it was found that almost all features had most values overlapping in both classes, however there were few features like age which showed clear distinction of values between the classes. It was inferred that older clients tend to repay the loan on time compared to youngsters.

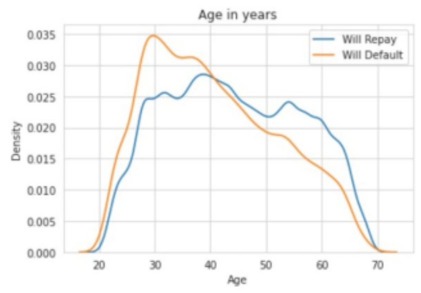

Fig6: Density plot of age for target classes

IV. Proposed solution

The goal of the project is to model credit default risk based on customer details and history of transactions. Since financial data has large volume and high velocity, we want to make sure that our model can be scaled to large data and train and predict in reasonable amount of time. We also want to make sure that adding more details of the customer would not affect the training time of the model. Keeping this as the main goal in mind, we did not want to compromise on the performance as well. So we designed a model using ensemble methods. We used model stacking where a Level 0 model trains on each table of the dataset parallely, and its output is given as input to the Level 1 model. Our choice of Level 0 model was LightGBM and Level 1 model was a simple ANN. The model performance depended heavily upon feature engineering hence we used tailor-made features for each table. The model was evaluated using ROCAUC score and gave an ROCAUC score of 0.90 on the train data, and an ROCAUC score of 0.76 on the hidden test data when submitted to the kaggle competition.

A. *Pre-processing*

The dataset was highly imbalanced with a ratio of classes being 1:12. Hence resampling methods like oversampling and undersampling will either lead to a lot duplicate rows to make up for heavy skew or remove important rows. We decided to allow the class imbalance be taken care by class weights in the models. Many outliers were also present in the main table which could be seen from box plots, but we decided not to remove them as it might be possible that since data is heavily skewed, these outliers are important to classify an observation as belonging to the minority class. Columns with more than 70% missing values were identified and not included in the final features while those with lesser than 70% were filled using median. Categorical variables were encoded with lables. After feature engineering, all the values were normalised to fall in the range of [0,1].

B. *Feature engineering and selection*

Feature engineering and selection was the most crutial and time taking task in the whole process. It was essential in determining the model performance. Initially we started by using correlation plots and XGBoost to determine the top features of the main table. We also tried to use PCA to reduce dimensionality, but almost all features had high covariance. For the other tables, each row on the main table had many corresponding rows in those tables, hence aggregate functions had to be applied. We started by using mean, min and max as the aggregates. But all of this did not yield good results in ROCAUC score.

While going through other solutions, it was noticed that the teams with best scores on the leaderboard had made new features using existing columns and included them while training and testing. There were also different types of aggregates using in the tables. Since we did not have much domain knowledge in finance, we combined features from different existing solutions to the competition. But this increased the number of features of each table and the model training time too. So we used Random Forest's feature importance along with some domain knowledge to pick the best features to be used in model training. For the final feature engineering, the features for the tables other than main table had aggregates applied to them. Mode was used for categorical variables while mean, min, max, sum and variance was used for the numerical columns. For tables which have entries for every month or based on dates, they were grouped based on year or month and then aggregates like min, max, mean and sum were applied.

C. *Model Building*

As new credit or other financial data keeps coming in with time, the model will have to be updated to capture new features added or update its parameters. Using a single computationally expensive strong learner will not be useful here. The winning team of the competition trained their model for 3 days using GPUs, however this will not be possible in the real world. Ensemble methods use many weak learners to train on different features of the dataset or different samples and their results are combined to give the final output. We used model stacking, a type of ensemble learning where heterogeneous weak learners were trained parallely on each table and a meta-model was used to assign weights to predictions of each table and give the final prediction. Figure 1 shows the diagrammatic representation of the general model architecture of our ensemble model

Level 0 models are the models used to train on the features of each table. There are 7 tables but the tables bureau and bureau_balance worked best when combined into 1 table, so we use 6 learning models. In traditionally ensemble learning, 100s of weak learners need to be combined, but since we use only 6 learners, we need to ensure that they do moderately well on the individual tables as well. From previous literature review, we found that Logistic regression, Decision Tress and LightGBM performed well. In order to find which models performed well with our features, we ran each of the models with the same features on all tables and the model that gave the best ROCAUC score for that table was chosen as the Level 0 model for that table. The ROCAUC score was compared for both train data and test data by submitting each table's predictions to the competition.

1) Logistic Regression: Logistic regression is one of the most common models used in binary classification. It is also known to perform well in datasets with large number of features. However, it assumes that the features are all linearly correlated with the log of the target. The features also should not be correlated with eachother else it might adversely effect the model performance. Logictic regression first find the linear hypothesis, then transforms it by applying a sigmoid function. Thus, Logistic regression can be represented as

$$Y = \frac{1}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+......+\beta_N X_N}}$$

To find the values of parameters $\beta_1, \beta_2, ...\beta_N$, the cost function is calculated for each β and is minimised using gradient descent to find the best values.

Since our data was highly imbalanced, we decided to give the classes weights. To find class weights, the formula $n\_samples/(n\_classes*np.bincount(y))$ was used. Once the features were extracted for the table, they were normalised and trained using the model. The model was then used to predict on train data itself.

2) Decision Trees: Decision trees are a supervised tree based learning algorithm that is used for classification. It models data as a tree of hierarchical branches and each node divides the dataset until it reaches the leaf node which gives

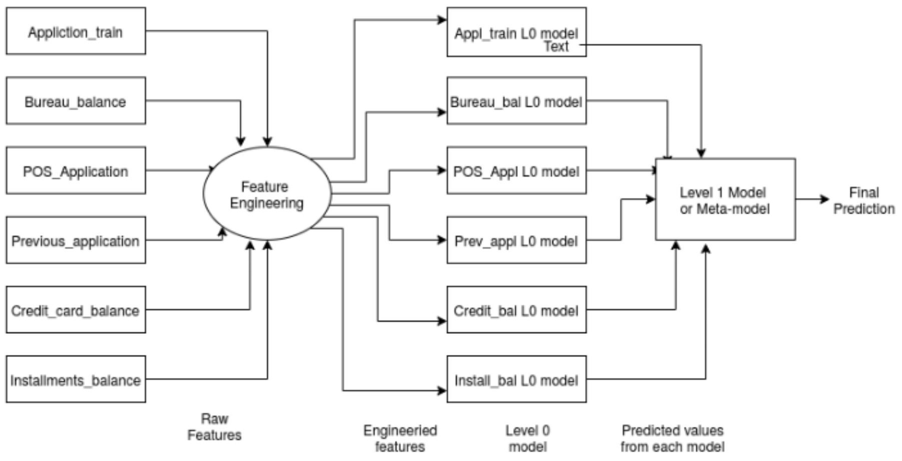

Fig. 1. Model Architecture

the output class. A node splits the dataset based on the best attribute. Best attributes are chosen using attribute selection methods like maximum information gain or minimum entropy. These models can handle non linearity in data and are expected to perform better that logistic regression.

To handle imbalanced data, we continue to use class weights as a parameter to the model. We also limit the depth of tree to 6 as we do not want a fully sophisticated strong learner, but a moderate learner. The model takes more train time that logistic regression but its ROCAUC score is better for most tables.

3) LightGBM: To increase the performance of decision trees, the classification model needs to be able to rank features based on their importance and be able to generalise the model as data has a lot of noise. Gradient Boosting Decision Tree (GBDT) would be a good fit to the dataset based on the above requirements. In each round, GBDT fits n Decision trees if there are n classes and also needs to select the best features. This needs the model to traverse through each data point in every round to find the ideal split making it computationally expensive and time consuming. LightGBM and XGBoost or the two models based on GBDT that are optimised to train faster. Since LightGBM is known to produce better results on larger data with higher dimensionality and train faster than XGBoost, we decided to go with LightGBM. It also splits the data leaf-wise which gives better accuracy compared to splitting level-wise by XGBoost.

We use Gradient based one sided sampling or GOSS as the boosting type. For gradient to converge, GOSS algorithm suggests a sampling technique which does not search through the entire sample space. It selects x% of data based on lowest gradient values and only does random sampling on them which helps the gradient to converge faster. To take care of class imbalance, we set the parameter

$$scale\_pos\_weight = \frac{number\_of\_negative\_samples}{number\_of\_positive\_samples}$$

Since the model trains leaf wise, we set the max depth of the tree as 6 and number of leaves as 30 to prevent overfitting. Subsample value was decided keeping in mind train time and model generalisation. Values of alpha and gamma which are L1 and L2 regularisation values were found by parameter tuning. The dataset was normalised and trained on this model. It gave a better accuracy among the 3 models and was chosen as the Level 0 model for all tables..

4) Level 1 model: The purpose of a Level 1 model is to combine the predictions of the Level 0 models and make the final prediction. The predictions can be combined by either taking a mode of the predictions or by assigning weights to each model's prediction and combining them. Since each table had a different score on the final data it was evident that they all need to have different amounts of influence in the final prediction. Weights can be assigned manually but would be a tedious task and most cases not the best values. So we decided to use a simple Artificial Neural Network for learning the weights for each table.

We used a multi-preceptron model that has 6 input neurons, each for predictions of 1 of the 6 tables, a hidden layer with 6 neurons and an output layer with 1 neuron for the prediction of the output class. Activation function for the hidden layer was relu, a common activation function used in hidden layers and sigmoid for the output layer as it is a binary classification.For the same reason,loss function used was binary cross entropy. Since the test evaluation metrics was ROCAUC, the model trained using the same with stochastic gradient descent as the optimiser. The model was trained for 3 epochs so than there was no overfitting by ensuring that the ROCAUC score of validation set was always better than that of the train set and did not reduce in further epochs. Class imbalance was handled using weights computed for decision trees in the Level 0 model. Batch size was also decided so that validation score increases in each epoch. The model gave an ROCAUC score of 0.91 in the last train epoch.

5) Putting it all together: First feature engineering was done on all the tables separately including the test data. The data from each table was normalised and train on the LightGBM model parallely. Since the tables are independent of eachother, they run parallelly thus saving training time. Once trained, the models are used to predict on train data. These predictions from each table are combined into a new dataset and given as input to the ANN which then trains on these prediction. To predict on test data, The Level 0 models that were trained are used to predict on test data. All this is combined into another new dataframe and given to the ANN. The ANN then uses these test predictions to give final predictions on the test data which was submitted to the competition.

D. *Experimental results*

The dataset used was from Kaggle and the size of all the files was 3GB. Hence, instead of downloading the datasets, we decided to run our models on Kaggle Notebooks which allows code reproduciblity and collaboration. These notebooks use 16gb RAM and 4 cores of CPU to run the models. The whole model pipeline which includes the time to train each table, get its predictions, train the Level 1 model and finally get predictions on test data took approximately 1 hour 30 minutes. This was half the time taken compared to the winning solution that took 3 hours to train the model.

Model evaluation was done using Area Under the Receiver Operating Characteristic Curve or ROCAUC score which was not sensitive to class imbalances. This was also the evaluation metric used on test data in the competition. the ROC curve is got by plotting a curve of $sensitivity$ vs $1 - specificity$. AUC is the area under this ROC curve. The greater the area, the better the model performs in distinguishing between positive and negative classes. This evaluation metric is used when we care equally about the positive and negative classes. Accuracy and other metrics do not work well in imbalanced datasets.

First to choose our Level 1 model, we had trained the tables on 3 different models - Logistic regression, Decision Tress and LightGBM. The ROCAUC score on train data for those models have been given in Table 1. It can be seen that Decision Tress perform better than Logistic regression for many tables except the main table. However LightGBM remained the clear winner among the 3 models for all tables. To make sure than this model wasn't overfitting on the train data, LightGBM was used to give predictions using the test data and submitted to the competition. Our final goal was to ensure that the ROCAUC score of the ensemble model was more than the maximum score of all the tables.

Once LightGBM was choosen as our Level 0 model, the predictions using it on all the table train data was got, and the ANN was trained using it. The Neural network gave an AUC of 0.91 on train data and 0.92 on the validation set after 3 epochs. We decided to stop here so that the model does not overfit. Once this model was trained, the predictions of the Level 0 models on test data was combined and given to

TABLE I
RESULTS OF VARIOUS MODELS ON EACH TABLE

Table name	Logistic regression	Decision Trees	LightGBM
	0.752	0.736	0.876
Credit Card balance	0.564	0.569	0.664
Installments	0.614	0.632	0.803
POS application	0.584	0.602	0.756
Bureau Balance	0.630	0.633	0.809
Previous application	0.666	0.634	0.844

TABLE II
RESULTS OF THE FINAL MODELS

Table name	Train data	Public LB	Private LB
Cred bal	0.664	0.576	0.568
Installments	0.803	0.641	0.629
POS data	0.756	0.599	0.595
Bureau	0.809	0.660	0.664
Prev appl	0.844	0.666	0.676
Appl train	0.800	0.724	0.718
ANN	0.918	0.751	0.748

the ANN to predict the final outputs. On submitting this to the Kaggle competition, the final ROCAUC score was 0.75 on the public LB. This was a better score than the scores of the individual tables. The results of LightGBM model on each table and ANN which was the final prediction of our ensemble model can be seen in Table 2. It has the scores of the models on train data, the public and private leader board scores of the test data who's predictions were submitted to the Kaggle Competition.

E. *Conclusion*

The goal of this project was to reduce training time of the models and predict whether the customer will repay the loan on time or not correctly without a trade off in performance. By using model stacking, the individual models could be trained parallely thus reduces the train time. Even if more features about the customers were added and need to affect the decisions of the model, we can just combine them into another table and train a another Level 1 model on it. This way, the whole single model need not be retrained for another few days making our model scalable. We achieved an ROCAUC score of 75% when the highest was 80%. Hence the performance of the model was also not compromised.

However since the ANN gave an AUC score of 92% on train data and reduced to 75% on the test, it is possible that the model has been overfitted. With enough parameter turning or using simpler models like multiple linear regression, this score could be brought up even higher. The scope of the model is limited to this dataset as feature engineering was the most important step and specific to this data. The model architecture however can perform better in other datasets if good features are extracted.

F. *Contribution of each team member*

The 4 of us started off by using our previous knowledge of preprocessing and model building on the datasets. This did not give us a good score. So each of us looked up all the previously available solutions. Based on what we read, Suvigya came up with the idea of using ensemble models. Supreeth and Kishan did the feature engineering and extraction of all the tables. Then Swarupa used the features to come up with the Level 0 model including choosing the parameters, exploring different models and training them. Finally Suvigya did the Level 1 model including model training and deciding the parameters.

ACKNOWLEDGMENT

We would like to express our gratitude to Dr. Gowri Srinivasa and the teaching assistants for providing us with the opportunity to explore data analytics and get some hands on experenice in it. I would like to thank the teaching assistants for their ”Words of Wisdom” series that helped us in design thinking, form our problem statement and guided us to write the literature survey report. We would also like to thank the Computer Science department from PES University for giving us the chance to explore one of the most trending technologies of the current times.

REFERENCES

[1] Z. Qiu, Y. Li, P. Ni, and G. Li, “Credit risk scoring analysis based on machine learning models,” in *2019 6th International Conference on Information Science and Control Engineering (ICISCE)*, 2019, pp. 220–224.

[2] S. P.M. and V. Paul, “A novel optimized classifier for the loan repayment capability prediction system,” in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, pp. 23–28.

[3] G. Teles, J. Rodrigues, R. Rabelo, and S. Kozlov, “Artificial neural network and bayesian network models for credit risk prediction,” *Journal of Artificial Intelligence and Systems*, vol. 2, pp. 118–132, 01 2020.

[4] Y. Aleksandrova, “Comparing performance of machine learning algorithms for default risk prediction in peer to peer lending,” *TEM Journal*, vol. 10, pp. 133–143, 02 2021.

[5] T. M. Alam, K. Shaukat, I. A. Hameed, S. Luo, M. U. Sarwar, S. Shabbir, J. Li, and M. Khushi, “An investigation of credit card default prediction in the imbalanced datasets,” *IEEE Access*, vol. 8, pp. 201 173–201 198, 2020.

[6] X. Gao, Y. Xiong, Z. Xiong, and H. Xiong, “Credit default risk prediction based on deep learning,” Aug. 2021. [Online]. Available: https://doi.org/10.21203/rs.3.rs-724813/v1

Credit Default Risk Prediction

ORIGINALITY REPORT

4%	1%	3%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	S.J. Shiv, Srinivasa Murthy, Krishnaprasad Challuru. "Credit Risk Analysis Using Machine Learning Techniques", 2018 Fourteenth International Conference on Information Processing (ICINPRO), 2018 Publication	1%
2	Yufei Xia, Junhao Zhao, Lingyun He, Yinguo Li, Xiaoli Yang. "Forecasting loss given default for peer-to-peer loans via heterogeneous stacking ensemble approach", International Journal of Forecasting, 2021 Publication	<1%
3	pdfs.semanticscholar.org Internet Source	<1%
4	Soni P.M., Varghese Paul. "A Novel Optimized Classifier For the Loan Repayment Capability Prediction System", 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019 Publication	<1%

5	Yanka Aleksandrova. "Comparing Performance of Machine Learning Algorithms for Default Risk Prediction in Peer to Peer Lending", TEM Journal, 2021 Publication	<1 %
6	doaj.org Internet Source	<1 %
7	Submitted to California Southern University Student Paper	<1 %
8	Zhongchen Ma, Qun Dai. "Selected an Stacking ELMs for Time Series Prediction", Neural Processing Letters, 2016 Publication	<1 %
9	scholarscompass.vcu.edu Internet Source	<1 %
10	Sumon Biswas, Hridesh Rajan. "Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness", Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020 Publication	<1 %
11	"Advances in Machine Learning and Computational Intelligence", Springer Science and Business Media LLC, 2021	<1 %

Publication

12	Ashwin S. Ravi, Akshay Sarvesh, Koshy George. "Sequential ELM for financial markets", 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), 2016 Publication	<1 %

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		