Comparison of Different Machine Learning Algorithms for Predicting LOAN RISK Categories

1st Salomo Hendrian Sudjono
Data Science Program, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
salomo.sudjono@binus.ac.id

2nd Favian HN Adrian
Data Science Program, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
favian.adrian@binus.ac.id

3rd Caroline Angelina Sunarya
Data Science Program, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
caroline.sunarya@binus.ac.id

4th Gabrielle Felicia Ariyanto
Data Science Program, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
gabrielle.ariyanto@binus.ac.id

5th Noviyanti T M Sagala
Statistics Department, School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
noviyanti.sagala@binus.edu

Abstract-- There are many cases where borrowed money by debtors is not returned. It is because the company misjudged in determining the risk of lending. Thus, debtors cannot repay their debts and end up in losses on the company's side. Thus, this research aims to create an accurate classification model for determining loan risk categories to minimize business losses. We used the public dataset from kaggle website. Then we perform several data transformation and analysis methods are performed to obtain information from the data. We built 2 classification models to predict the original label. Unfortunately, the desired results did not achieve based on the f1-score. Therefore, the clustering method using K-Means is applied to make a new label for the dataset and determine the behavior of each cluster. Each cluster is labeled then build a model to perform cluster classification. We created several classification models and found that gradient tree boosting is the most accurate predictor of the loan risk category based on the accuracy. We used clustering to create a classification model that can have high accuracy. The classification model we built can have high accuracy because it uses clustering. The empirical results of this study and others pave the way for further in-depth future work to guarantee that the algorithm performs acceptably when confronted with more complicated real data.

Keywords—kmeans clustering, boosting algorithm, loan risk, predictive model

I. INTRODUCTION

Credits and loans are frequently utilized for many different things, including consumer, educational, medical, travel, and business objectives [1]. In lending money to a company, money will not necessarily be given directly when you want to borrow. There are several steps that the company must take to assess whether the debtor's loan is a high risk or low risk for the company. This is called risk management [2]. In risk management, a debtor's credit profile becomes a reference as a benchmark for the company to borrow from debtors. Suppose the debtor has a problematic credit profile, such as having missed payments. A problematic credit profile refers to a high-risk loan because there is a big concern that the debtor will not make a refund at a certain time [3]. Due to inadequate credit risk management procedures, some financial institutions have failed or had financial issues [4]. One of the biggest impacts of not being careful in determining lending risk is the subprime crisis, which started with bad loans in the property sector. The global financial crisis started the debtors with bad credit histories, and low incomes could also purchase real estate. However, when the lending boom came to an end, credit standards further fell as teaser rates and "stated income" loans (minimal documentation loans with mortgages) emerged as well as "ninja" loans (for borrowers with "no income, no employment, and no assets," brokers and borrowers just reported the borrower's income) [5].

Building classification model in machine learning can help in determining the risk category of debtors, it will be impactful to generate enough revenue for the businesses [6]. Using this model, businesses may actively manage their loan portfolios and lower the risk of default, which will ultimately increase their overall financial stability and profitability.

II. RELATED WORK

Many researchers have discussed the implementation of machine learning in classifying loan categories. Aboobyda Jafar Hamid and Tarig Mohammed Ahmed used data mining techniques to build models to classify a good or bad loan. Of the three models they built, the j48 classification algorithm is the best model in classifying loan risk, based on the correct percentage in classifying classes, which reaches 78% [6].

Amir E. Khandani et al. created a machine learning model to predict credit risk from bureau data from January 2005 to April 2009. The results obtained from the CART decision tree algorithm resulted in 99% correct predictions by the model [7].

Nazeeh Ghatasheh conducted his study to investigate the opportunities of the Decision Tree by focusing on Random Forests to research Forest Tree performance, contrasting various methodologies, identifying the optimum configurations for greater prediction accuracy, and identifying the shortcomings of the chosen prediction

approach [8]. The findings of this study demonstrate that the Random Forest Trees algorithm offers Business Analytics a good chance to identify credit risk [8]. The findings of this study demonstrate that the Random Forest Trees algorithm offers Business Analytics a good chance to identify credit risk [8].

Ashwini S. Kadam et al. Concluded with certainty that the Naive Bayes model is highly effective and provides a superior outcome to SVM model [9].

M. Xiaojun et al used LightGBM and XGBoost algorithms to make prediction of P2P network loan. They concluded that the 'multi-observation' data cleaning method is superior to the'multidimensional' method, making the LightGBM algorithm superior to the XGboost algorithm. Multi-observation data sets outperform multidimensional data sets for the same technique. The classification prediction results of LightGBM are superior to those of XGboost for the identical data set. With an error rate of 19.9% and an accuracy of 80.1%, the results demonstrate that the classification prediction made by the LightGBM method using multidimensional data sets is the best [10].

In August 2020, Amruta S. Aphale and his partner Dr. Sandeep build a Nearest Centroid and Gaussian Naïve Bayes for predict a customer's creditworthiness and loan repayment capacity [11]. The results of the experiment show that, with the exception of the Nearest Centroid and Gaussian Naive Bayes, all other algorithms exhibit respectable performance in terms of accuracy and other performance assessment measures. Each of these algorithms has an accuracy rate ranging from 76% to more than 80% [11].

III. METHODOLOGY

Before we dive into the work process, it is better to know what stages are carried out in this project. All stages are shown in Figure 1.

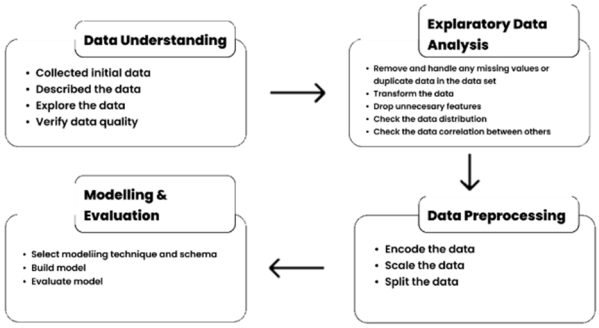


Fig. 1. Research workflow

A. Data Gathering

Data gathering is the next process after we determine the goal of a machine learning project. In this stage, all the data used in the project will be collected to analyze, transform, and build a model.

B. Data Understanding

Understanding the data is an important step before building a model. The main goal at this stage is to learn the existing patterns and observations that include knowing the number of dimensions in the dataset, the type of data contained, the year the data was created, etc.

C. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a method that aims to gain information or pattern from the data. This included changing the form of data in terms of dimensions and content in the data. The purpose of the transformation is to make it easier to analyze.

D. Data scaling and Encoding

Scaling and Encoding are included in data preprocessing, where the data is transformed with the aim that the model can improve its learning ability. Scaling is a method so that the range of values in each column has the same range. Encoding changes, the categorical data type to numerical because the model can only recognize numeric data.

E. Data Splitting

The machine learning model used is a supervised model. This model requires input data as training data and testing data. At this stage, the data that has been processed previously will be divided into two parts, namely, train data and test data.

F. Model Building

At this stage, we build a model determined based on the objectives to be achieved with the training data created at the data splitting stage.

G. Model Evaluation

After building a model with training data, it is necessary to evaluate the prediction results of the model by looking at the appropriate evaluation metrics.

Data will be first gathered for the study from the Kaggle website. For the time being, we only have access to datasets and information that are publicly licensed. Therefore, as there are no prerequisites for accessing public datasets, that is why we chose them for our study. The data contains records of borrowers who have borrowed. The data is processed using the python programming language. In the data understanding stage, we take a quick look at the values in the data and investigate what data types are contained in the dataset. Preview of dataset can be seen in Figure 2 and Figure 3.

Number_Open_...	Total_Accounts i...	Gender object	Interest_Rate int...
9	14	Female	1
12	24	Female	3
12	16	Male	3
16	22	Male	3
19	30	Female	1

Fig. 2. Dataset preview 1

Income_Verified	Purpose_Of_Loan	Debt_To_Income	Inquiries_Last_6...	Months_Since_D...
not verified	car	18.37	0	nan
VERIFIED - income	debt_consolidation	14.93	0	17.0
VERIFIED - income source	debt_consolidation	15.88	0	nan
VERIFIED - income source	debt_consolidation	14.34	3	nan
VERIFIED - income source	debt_consolidation	22.17	1	nan

Fig. 3. Dataset preview 2

Figure 4 represents the data types of each column in dataset.

#	Column	Non-Null Count	Dtype
0	Loan_ID	164309 non-null	int64
1	Loan_Amount_Requested	164309 non-null	object
2	Length_Employed	156938 non-null	object
3	Home_Owner	138960 non-null	object
4	Annual_Income	139207 non-null	float64
5	Income_Verified	164309 non-null	object
6	Purpose_Of_Loan	164309 non-null	object
7	Debt_To_Income	164309 non-null	float64
8	Inquiries_Last_6Mo	164309 non-null	int64
9	Months_Since_Deliquency	75930 non-null	float64
10	Number_Open_Accounts	164309 non-null	int64
11	Total_Accounts	164309 non-null	int64
12	Gender	164309 non-null	object
13	Interest_Rate	164309 non-null	int64

Fig. 4. Data type info

From the information above, it can be seen that there are columns that are not suitable for representing data values. Therefore, before further analysis, we change the column data type first. After that, we do several data understanding and preprocessing stages, including several methods written below.

- Drop missing/null values.
  Removing rows that contain missing values.

- Investigating multicollinearity.
  To make the model easier to learn, we discard independent variables that strongly correlate with other independent variables.

- Scaling.
  Since the data has different units, scaling is necessary to improve the model performance.

- Encoding.
  The data contains columns with ordinal and nominal data types, but they are still strings. We need to convert them to numerals to represent each value so that the machine can understand them.

- Splitting the dataset into train data and test data.
  The proportion in dividing the dataset is 20% for testing data and the rest for training data. The sampling method used is stratified sampling to balance the proportion of labels in the train data and test data.

In the first modeling stage, we used a simple model, such as Logistic Regression with default parameters, to see the recall value of each label. We need to investigate the macro avg recall value because the dataset has unbalanced labels; you can see in the support section of the classification report below that label 1, 2, and 3 have an unbalanced number of observations. In logistic regression, the resulting recall value is very low, less than 50% as shown in Figure 5.

	precision	recall	f1-score	support
1	0.32	0.28	0.30	4645
2	0.45	0.47	0.46	9682
3	0.47	0.48	0.48	8158
accuracy			0.44	22485
macro avg	0.41	0.41	0.41	22485
weighted avg	0.43	0.44	0.43	22485

Fig. 5. Logistic regression classification report

Then, we switched to a fairly complex model, Random Forest, hoping that the macro avg recall value would touch 60%. However, the evaluation results of random forest were also similar to the previous model as shown in Figure 6.

	precision	recall	f1-score	support
1	0.49	0.12	0.19	4645
2	0.48	0.65	0.55	9682
3	0.53	0.53	0.53	8158
accuracy			0.50	22485
macro avg	0.50	0.43	0.42	22485
weighted avg	0.50	0.50	0.47	22485

Fig. 6. Random forest classification report

After reviewing both models, we concluded that the existing labels did not match the borrower's behavior, so we decided to load new labels using the KMeans algorithm. The cluster generated by means will later be used as a new label. To determine the value of K in K-means, we use the elbow method and silhouette plot. The results can be seen in Figure 7.

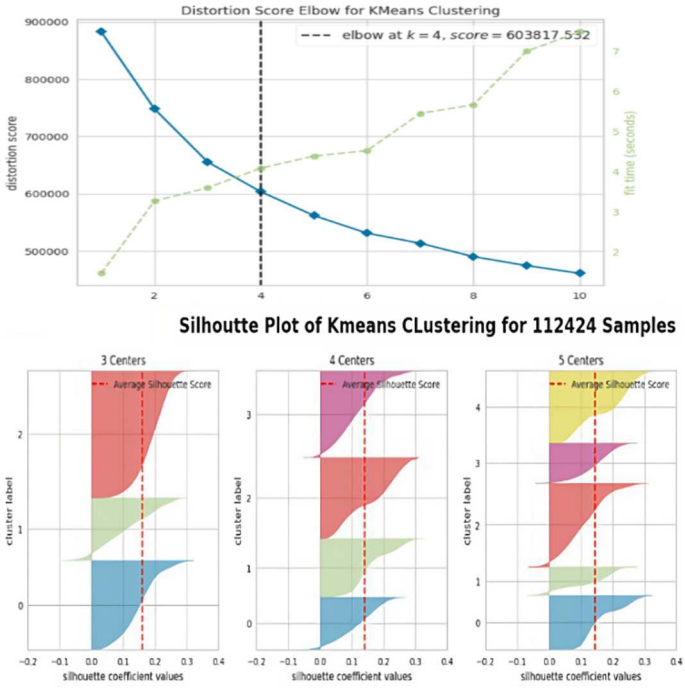


Fig. 7. Elbow method & Silhoutte Plot

Based on these two methods, the most optimal k value is k = 4. After that, we built the KMeans model with k = 4. Then, we return to the preprocessing stage, which includes.

- Feature selection.
  In Feature Selection, we use Decision Tree with 99% accuracy to evaluate the columns contributing to label classification. We take the five most contributing columns for the next stage

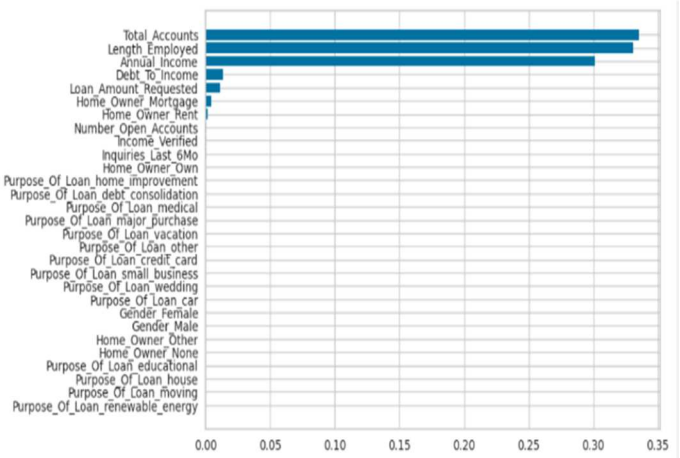


Fig. 8. Most important features

- Scaling.
  Because we are using data that has not been scaled, we need to rescale the data with new labels.

- Splitting.
  The splitting method is the same as before, using stratified sampling and the proportion of test data is 20% and 80% for train data.

In the second modeling stage we used logistic regression again with default parameters. The results can be seen in Figure 9.

```
=======================================
Logistic Regression Classification Report
=======================================
              precision    recall  f1-score   support

           0       0.90      0.92      0.91      4268
           1       0.52      0.30      0.38      4714
           2       0.59      0.72      0.65      6521
           3       0.90      0.97      0.94      6982

    accuracy                           0.75     22485
   macro avg       0.73      0.73      0.72     22485
weighted avg       0.73      0.75      0.73     22485
```

Fig. 9. Logistic Regression classification report

The results generated in this first model have an avg recall value that far exceeds the previous one, 73%. But we still have a problem; 73% is the overall value, so if we evaluate the f1-score for each label, we find that label 1 has a low f1 score. Thus, we built several more models in the hope that the recall value would increase. To ensure the metrics generated are not from an overfit model, we use the cross-validation score, then take the average value. Here are some evaluations of the models we built.

```
=====================================
K-Nearest Neighbors Classifier Report
=====================================
              precision    recall  f1-score   support

           0       0.94      0.83      0.88      4268
           1       0.95      0.98      0.96      4714
           2       0.95      0.98      0.96      6521
           3       0.96      0.98      0.97      6982

    accuracy                           0.95     22485
   macro avg       0.95      0.94      0.95     22485
weighted avg       0.95      0.95      0.95     22485

Average ROC score: 0.991217307441998
Average accuracy: 0.9477757163688884
```

Fig. 10. KNN classification report

After we evaluated several models, we concluded that the best model we built was gradient tree boosting.

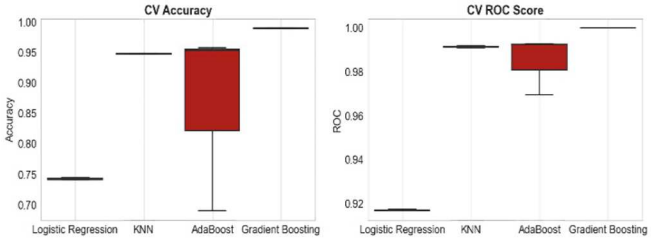
```
==================================
Gradient Boosting Classifier Report
==================================
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      4268
           1       1.00      0.99      0.99      4714
           2       0.99      0.99      0.99      6521
           3       0.99      0.99      0.99      6982

    accuracy                           0.99     22485
   macro avg       0.99      0.99      0.99     22485
weighted avg       0.99      0.99      0.99     22485

Average ROC score: 0.9998679169584203
Average accuracy: 0.9895484677833523
```

Fig. 11. Gradient Boosting Classification report

Now, we can retrieve the metrics results with accuracy because the f1 score and the number of observations of each label are quite balanced. The model using the Gradient Boosting Classifier algorithm produces an accuracy of around 99% and its ROC AUC value is up to 99.9%. Seeing from the difference in precision and recall values, it can be concluded that the model can predict well.

IV. RESULT & DISCUSSION

Using the dataset's target variable provided by the dataset, the modeling results have a poor accuracy value. However, after performing Clustering, it was found that the optimal number of clusters was 4. So, modeling was done again. Each



model performance are shown in the figure below.

Fig. 13. Models cross-validation score

It was found that Ada Boost algorithm performance is not stable in each training iteration (Fig. 13). In other hand, the Gradient Tree Boosting algorithm that has the best performance, which has an accuracy value of up to 99%, and the ROC AUC value is as high as 99.9% (Fig. 11). The algorithm has been able to predict the value of each class well. It can be proven through the precision and recall values where the maximum difference between precision and recall values is only 1% for all classes, unlike the other models that have the minimum difference between precision and recall values as high as 1% or even more. Besides, we get accurate model results on this data, we get some characteristics of each label obtained by using Clustering through KMeans. The following are the characteristics of each label generated from KMeans clustering.

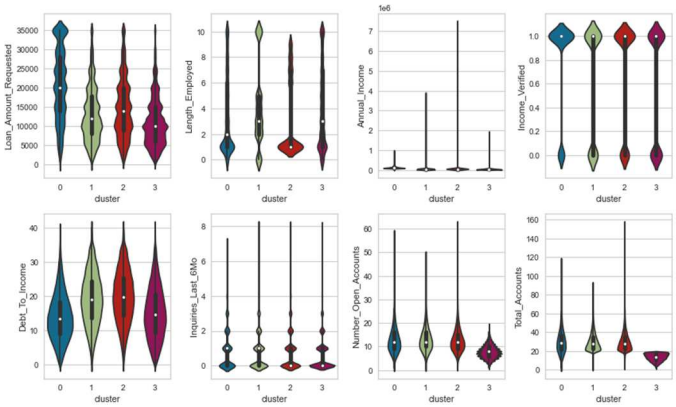


Fig. 14. Cluster based on cumerical features

Based on figure 14, it can be concluded that the numerical features that affect borrower characteristics are determined by *Loan_Amount_Requested, Length_Employed, Annual_Income, Debt_To_Income,* and *Total_Accounts*.

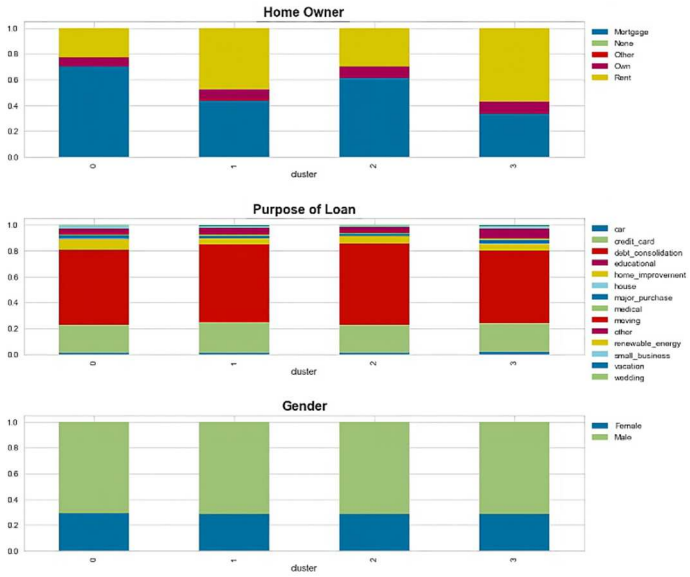


Fig. 15. Cluster based on categorical features

According to Figure 15, *Home_Owner* or the status of residential ownership, is the sole categorical feature that has a significant bearing on differentiating each cluster.

- Cluster 0 (Refers to Low-Risk Loan).
  Debtors that belong to this cluster have the characteristics: of having the highest loan rate amongst all clusters, having been employed for more than five years, having the highest annual income, often making loans, and having the lowest debt-to-income ratio.

- Cluster 1 (Refers to Low-Medium Risk Loan).
  Debtors that belong to this cluster have the characteristics: a pretty low loan rate, has been employed for more than ten years, have a pretty low annual income, often make loans, and have a pretty high debt-to-income ratio.

- Cluster 2 (Refers to High-Medium Risk Loan).
  Debtors that belong to this cluster have the characteristics: have the least loan rate amongst all clusters, have been employed for more than ten years, have the lowest annual income, rarely make loans, and have a pretty low debt-to-income ratio.

- Cluster 3 (Refers to High-Risk Loan).
  Debtors that belong to this cluster have the characteristics: of having a pretty high loan rate, having been employed for more than one year, having a pretty high annual income, often making loans, and having the highest debt-to-income ratio.

CONCLUSION

Using the dataset's target variable provided by the dataset, the modeling results have a poor accuracy value. However, after performing Clustering, it was found that the optimal number of clusters was 4. So, modeling was done again. It was found that the algorithm that has the best performance is the Gradient Tree Boosting algorithm. This is interesting to note that boosting algorithms are not always the best. This refers to the computational aspect, which is more complex and thus has a longer training time than others. The classification model we built can have high accuracy because it uses clustering. The issue is that clustering labels are not always pertinent to the established business objectives. Therefore, it is necessary to be re-analyzed each customer characteristic in order to align with the current goal. On the other hand, we are also not sure how well the model will perform when faced with more real data that is likely to contain outliers. Secondly, although the boosting approach performed well on this dataset, we cannot be certain that our model can handle more complicated actual data due to the boosting algorithm's propensity for overfitting. In reality, the availability and caliber of data always provide a challenge to the model-building process. The data frequently have inconsistencies or missing numbers since the data collection method is never perfect or entirely correct. Because they are characterized by increased noise, heavy-tailed distributions, nonlinear patterns, and temporal relationships, large financial datasets typically provide major statistical hurdles [12]. The empirical results of this study and others pave the way for further in-depth future work to guarantee that the algorithm performs acceptably when confronted with more complicated real data.

REFERENCES

[1] A. Mayank, “Prediction of loan behaviour with machine learning models for secure banking,” February 2022.

[2] W. Kinyua, “Effects of credit risk management practices on loan performance of commercial banks in nyeri country, kenya,” vol. II, 2017.

[3] D. Sandeep et al, “Debtor-in-possession financingand bankruptcy resolution: empirical evidence,” March 2003.

[4] K. Njanike, “The impact of effective credit risk management on bank survival,” vol. IX, Universitas Publishing House Petroşani - Romania, 2009, pp.173.

[5] A. Josef, “The subprime crisis and its consequences,” April 2008.

[6] J. H. Aboobyda and A. Tarig, “Developing prediction model of loan risk in banks using data mining,“ March 2016.

[7] E. K. Amir et al, “Consumer credit-risk models via machine-learning algorithms,” June 2010.

[8] G. Nazeeh, “Business analytics using random forest trees for credit risk prediction: a comparison study,” vol. LXXII, 2014, pp. 2

[9] S. K. Ashwini et al, “Prediction for loan approval using machine learning algorithm,” April 2021.

[10] M. Xiaojun et al, “Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGBoost algortihms according to different high dimensional data cleaning,” August 2018.

[11] E. Zoran. “Predicting default loans using machine learning (optiML),” November 2019.

[12] S. A. Amruta and R. S. Sandeep, “Predict loan approval in banking system machine learning approach for cooperative banks loan approval,” August 2020.