p-ISSN: 2301-6949
e-ISSN : 2715-5064

LOAN STATUS PREDICTION USING DECISION TREE CLASSIFIER

Siti Aisyah, S.Tr., M.Sc.
Data Science, Universitas Insan Cita Indonesia
Jl.Rasuna Said Kav. C-18 South Jakarta, Indonesia
email: sitiaisyah@uici.ac.id

Abstrak This paper investigates the effectiveness of the Decision Tree Classifier in predicting loan status, a critical task in the financial sector. The study utilizes a dataset containing various attributes of loan applicants such as income, credit score, employment status, and loan amount. The dataset is preprocessed to handle missing values and categorical variables. Feature importance is analyzed to understand the key factors influencing loan approval decisions. A Decision Tree Classifier model is trained and evaluated using performance metrics such as accuracy, precision, recall, and F1-score. The results demonstrate the feasibility of using Decision Trees for loan status prediction and provide insights into the decision-making process of loan approval.

Keywords: Loan Status Prediction, Decision Tree Classifier, Credit Risk Assessment, Feature Importance, Model Evaluation

I.INTRODUCTION

A loan approval process is a crucial aspect of the financial industry, involving assessing an individual or entity's creditworthiness to determine whether they qualify for a loan. The process typically consists of several key stages, which may vary depending on the loan type and the lending institution's policies such as application submission about the personal details, pre-screening, documentation verification, credit check, and approval or denial. Moreover, the loan approval process is designed to assess the risk of lending money and ensure that loans are granted to individuals or entities who can repay them. Effective risk assessment techniques, such as credit checks and underwriting, are essential for maintaining lending institutions' financial stability and protecting borrowers' interests. Therefore, the objective of this article is to explore the use of machine learning approaches in the loan-taking process, particularly the Decision tree classifiers, to accurately predict whether a loan applicant will be approved or denied based on a set of input features. Decision tree classifier is supervised machine learning algorithms that are particularly well-suited for classification tasks, making them a suitable choice for predicting loan status.

II.LITERATURE REVIEW

This section provides an overview of prior work on creating deep learning and machine learning models using different algorithms to improve loan prediction procedures and help financial institutions and banking regulators choose competent, low-risk candidates. Another study used decision trees to build and evaluate loan prediction models, and on a public test set, they achieved an accuracy rate of 81%[1]. Lastly, using the same dataset, a comparison study between the random forest and decision tree algorithms revealed that the random forest achieved 80% accuracy, while the choice tree only achieved 73%[2]. By adjusting the Random Forest classifier's parameters, the author was able to attain a 78.64% effectiveness rate, which is similar to the decision tree classifier's 85.3% prediction efficiency[3]. The class prediction was generated as the model's output using an ensemble technique known as Random Forest. [4].

III.METHODOLOGY

A. Data Preparation

A.1. Data Understanding

The data contains 12 columns such as Gender, Married, Dependents, Education, Self-Employed, ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term, Credit_History, Property_Area, Loan_Status and 381 rows.

A.2. Handling the Missing Values

There are many columns with null values in the dataset such as Gender, Dependents, Self_Employed, Loan_Amount_Term and Credit History as shown in Figure 1. To solve this problem, the most common method of data imputation was used where it just replaced all missing values with the mode of column. See Figure 1 before imputation and Figure 2 after imputation.

```
df.isnull().sum()

Gender               5
Married              0
Dependents           8
Education            0
Self_Employed       21
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term    11
Credit_History      30
Property_Area        0
Loan_Status          0
dtype: int64
```

Figure 1. Before Imputation

```
df.isnull().sum()

Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
Loan_Status          0
dtype: int64
```

Figure 2. After imputation

A.3. Converting Categorical Data Into Numerical (Hot-Encoding)

Certain variables in our data—such as Gender, Married Status, Education, Self-Employed Status, and Loan Status—are string variables. Categorical variables can theoretically be used directly in some model types (like trees), however Sklearn does not enable this. We must thus change these variables to numeric values for the model to function with this set of data.
Sklearn.preprocessing.OneHotEncoder - For string variables with values that don't naturally follow a sequence (such as location names or work titles), OneHotEncoder is a useful tool. These will generate a new feature with a value of 0 or 1 for every conceivable combination as nominal variables shown in Figure 3.

Gender	Married	Dependents	Education	Self_Employed	Loan_Status
0	0	1.0	0	0	0
0	0	0.0	0	1	1
0	0	0.0	1	0	1
0	1	0.0	0	0	1
0	0	0.0	1	0	1

Figure 3. Hot Encoding for Some Variables

B. Data Visualization

B.1. Histogram

The histogram displays the variables for one or more groups. It shows the distribution of categorical values assigned to columns (Figure 4).

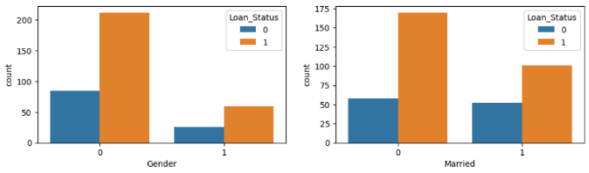

Figure 4. Distribution of Categorical Values

B.2. Boxplot

A boxplot depicts the distribution of a numerical variable among one or more groups. It displays the means, medians, quartiles, and outliers. To create boxplots in Python, use the Seaborn function. It can be seen that the mean of loan_status as the dependent variable toward ApplicatioIncome as an independent variable is about 3500 (Figure 5).

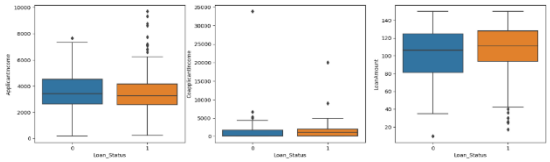

Figure 5. Boxplot

C. Splitting Dataset into Train and Test

We separated the data in an 80:20 ratio to create a training and test dataset (Figure 6). We train the model on the training set and then test it on the testing set to see how well it performed. The primary goal here is to evaluate the training model's performance on previously unknown data. To separate data, the train_test_split sklearn function in Python was used.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 6. Train Test Split

D. Modeling

The Decision Tree Machine Learning model uses the divide and rule technique to forecast player probability. It repeats a training dataset as observations and modifies the parameters that determine whether an observation is accurate or erroneous. A decision tree simplifies decision-making by combining many criteria and categorical variables into decision points, resulting in a single answer value for observations. However, constructing a decision tree may be computationally costly, especially when analysing a big dataset with numerous continuous variables. To make the model, follow these steps:

Step 1: Create a training and test dataset

Step 2: Import the model for usage. A Decision tree model for classification is developed using ***DecisionTreeClassifier*** (Figure 7)**.** The hyperparameters utilized for this model are outlined below :

- max_depth : the tree's maximum depth
- max_features : the amount of features at each split
- min_samples_leaf : the minimal number of samples needed at a leaf node

```
model = DecisionTreeClassifier(max_depth=3,min_samples_leaf = 35)
```

Figure 7. Initiate Model of Decision Tree

Step 3: Fit the model with training data shown in Figure 8.

```
model.fit(X_train,y_train)
```

Figure 8. Fit Model of Decision Tree

IV.RESULT & DISCUSSION

Figure 9 shows that the model trained using a Decision Tree with the best parameter had an accuracy of 81.82%. So, the accuracy meant that the model got 81.82 percent of the forecast accurate out of 100% predictions of decisions.

```
accuracy = accuracy_score(y_test, y_pred)
roc_score = roc_auc_score(y_test, y_pred)

print(f'Accuracy Score: {accuracy*100:0.2f}%')
print(f'Roc Score: {roc_score*100:0.2f}%')
```

```
Accuracy Score: 81.82%
Roc Score: 66.67%
```

Figure 9. Result of Accuracy Score

In this study, we developed a model for loan prediction using the Decision Tree method. The model's results, together with their classification report and confusion matrix, are provided below to help you comprehend the accuracy. The confusion matrix influences the evaluation model's output values. True Positive (TP) and True Negative (TN) indicate the observations that were anticipated correctly. The number of erroneously predicted samples is then denoted by False Positive (FP) and False Negative (FN). The result of the confusion matrix of the Decision Tree is discussed below (Figure 10) :

- True Positive (TP) = 56, meaning the model correctly predicted 56 loans approved.
- True Negative (TN) = 7, meaning the model correctly predicted 7 loans rejected.
- False Positive (FP) = 14, meaning the model incorrectly predicted 14 loans approved as belonging to the rejected status.
- False Negative (FN) = 0, meaning the model incorrectly predicted 0 loan rejected as belonging to the approved status.
-

		Actual Values	
		1	0
Predicted Values	1	56 (TP)	14 (FP)
	0	0 (FN)	7 (TN)

Figure 10. Result of Confusion Matrix

Then, we can know the percentage accuracy of model by calculating predicted values and actual values (Figure 11). So, we get accuracy of Decision Tree's model at 81 %.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Figure 11. Calculation of Accuracy

V. CONCLUSION

This research proposes using the Decision Tree classification method to forecast loan status, which is considered to be appropriate for the real-world environment. The model achieved an accuracy of 82%, indicating that the Decision Tree approach is a suitable model for this type of data. In the future, the project can employ other models to improve its accuracy.

REFERENCES

[1] Sheikh MA, Goel AK, Kumar T. An approach for prediction of loan approval using machine learning algorithm. Paper presented at: 2020
[2] Madaan M, Kumar A, Keshri C, Jain R, Nagrath P. Loan default prediction using decision trees and random forest: a comparative study.
[3] Gautam K, Singh AP, Tyagi K, Kumar MS. Loan prediction using decision tree and Random Forest. 2008.
[4]
Dansana D, Patro SGK, Mishra BK, Prasad V, Razak A, Wodajo AW. Analy zing the impact of loan features on bank loan prediction using Random Forest algorithm. *Engineering Reports*. 2024; 6(2):e12707. doi: 10.1002/eng2.12707