Prediction Model

Credit Risk Analysis: Modeling and Prediction Using Machine Learning

ID/X Partners - Data Scientist

Presented by
Muhammad Devanda Hendra Kusuma

Rakamin Academy X id/x partners

Introducing
About Me

Jakarta, Indonesia

devandakusuma20@gmail.com

Muhammad Devanda Hendra Kusuma

Hi! I am participating in the Project-Based Virtual Internship Program: Data Scientist ID/X Partners x Rakamin Academy from February to March 2025, and I am excited to share my final project. I will use Google Colab to analyze loan data and create a model to predict credit risk.

My process will include Business Understanding, Data Understanding, Exploratory Data Analysis (EDA), Data Preparation, Data Modeling, and Evaluation. The goal is to predict credit risk from the dataset, identify risk factors, optimize decisions, and build predictive models. I look forward to demonstrating how data science can enhance awareness and improve risk factor identification.

About Company

id/x partners was founded in 2002 by former bankers and management consultants with extensive experience in credit cycle and process management, scoring development, and performance management. Our combined experience has served corporations across Asia and Australia in a range of industries, particularly in financial services, telecommunications, manufacturing, and retail.

id/x partners provides consulting services that specialize in leveraging data analytics and decision-making (DAD) solutions, combined with integrated risk management and marketing disciplines, to help clients optimize portfolio profitability and business processes. Our comprehensive consulting services and technology solutions make id/x partners an integrated service provider.

Project Guidlines.

"Sebagai Data Scientist di ID/X Partners, Anda akan terlibat dalam sebuah proyek dari perusahaan pemberi pinjaman (multifinance), dimana client Anda ingin meningkatkan keakuratan dalam menilai dan mengelola risiko kredit, sehingga dapat mengoptimalkan keputusan bisnis mereka dan mengurangi potensi kerugian. Tugas Anda adalah mengembangkan model machine learning yang dapat memprediksi risiko kredit (credit risk) berdasarkan dataset yang disediakan, yang mencakup data pinjaman yang disetujui dan ditolak. Dalam pengembangan modelnya Anda juga perlu melakukan beberapa tahap dimulai dengan Data Understanding, Exploratory Data Analysis (EDA), Data Preparation, Data Modelling, dan Evaluation."

Rakamin Academy X id/x partners

Business Understanding

Background

High credit demand requires precise and accurate risk analysis.

Unmanaged credit risk will result in financial losses

Borrower behavior and historical data can be used as important indicators in predicting risk

Purpose of Analysis

High credit demand requires precise and accurate risk analysis.

The identification of risk factors is essential.

Optimize Decisions

Data Understanding

Data Understanding (1)

```
[11] data_df.shape
(466285, 75)
```

This dataset contains 466,285 rows and 75 features.

```
data_df.head()
```

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	insta
0	0	1077501	1296599	5000	5000	4975.0	36 months	10.65	
1	1	1077430	1314167	2500	2500	2500.0	60 months	15.27	
2	2	1077175	1313524	2400	2400	2400.0	36 months	15.96	
3	3	1076863	1277178	10000	10000	10000.0	36 months	13.49	
4	4	1075358	1311748	3000	3000	3000.0	60 months	12.69	

5 rows × 75 columns

Here are the 5 sample columns from the dataset above.

It is also known that the dataset consists of 22 categorical variables and 53 numerical variables, including:

Categorical Variables: ['term', 'grade', 'sub_grade', 'emp_title', 'emp_length', 'home_ownership', 'verification_status', 'issue_d', 'loan_status', 'pymnt_plan', 'url', 'desc', 'purpose', 'title', 'zip_code', 'addr_state', 'earliest_cr_line', 'initial_list_status', 'last_pymnt_d', 'next_pymnt_d', 'last_credit_pull_d', 'application_type']

Data Understanding (2)

```
[11] data_df.shape

(466285, 75)
```

This dataset contains 466,285 rows and 75 features.

```
data_df.head()
```

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	insta
0	0	1077501	1296599	5000	5000	4975.0	36 months	10.65	
1	1	1077430	1314167	2500	2500	2500.0	60 months	15.27	
2	2	1077175	1313524	2400	2400	2400.0	36 months	15.96	
3	3	1076863	1277178	10000	10000	10000.0	36 months	13.49	
4	4	1075358	1311748	3000	3000	3000.0	60 months	12.69	

5 rows × 75 columns

Here are the 5 sample columns from the dataset above.

It is also known that the dataset consists of 22 categorical variables and 53 numerical variables, including:

Numerical Variables: ['Unnamed: 0', 'id', 'member_id', 'loan_amnt', 'funded_amnt', 'funded_amnt_inv', 'int_rate', 'installment', 'annual_inc', 'dti', 'delinq_2yrs', 'inq_last_6mths', 'mths_since_last_delinq', 'mths_since_last_record', 'open_acc', 'pub_rec', 'revol_bal', 'revol_util', 'total_acc', 'out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'total_rec_int', 'total_rec_late_fee', 'recoveries', 'collection_recovery_fee', 'last_pymnt_amnt', 'collections_12_mths_ex_med', mths_since_last_major_derog', 'policy_code', 'annual_inc_joint', 'dti_joint', 'verification_status_joint', 'acc_now_delinq', 'tot_coll_amt', 'tot_cur_bal', 'open_acc_6m', 'open_il_6m', 'open_il_12m', 'open_il_24m', 'mths_since_rcnt_il', 'total_bal_il', 'il_util', 'open_rv_12m', 'open_rv_24m', 'max_bal_bc', 'all_util', 'total_rev_hi_lim', 'inq_fi', 'total_cu_tl', 'inq_last_12m']

Data Understanding (3)

```
# print categorical variables containing missing values

cat1 = [var for var in categorical if data_df[var].isnull().sum()!=0]

print(data_df[cat1].isnull().sum())
```

```
emp_title              27588
emp_length             21008
desc                  340304
title                     21
earliest_cr_line          29
last_pymnt_d             376
next_pymnt_d          227214
last_credit_pull_d        42
dtype: int64
```

Missing values in categorical variables.

```
# print numerical variables containing missing values

cat1 = [var for var in numerical if data_df[var].isnull().sum()!=0]

print(data_df[cat1].isnull().sum())
```

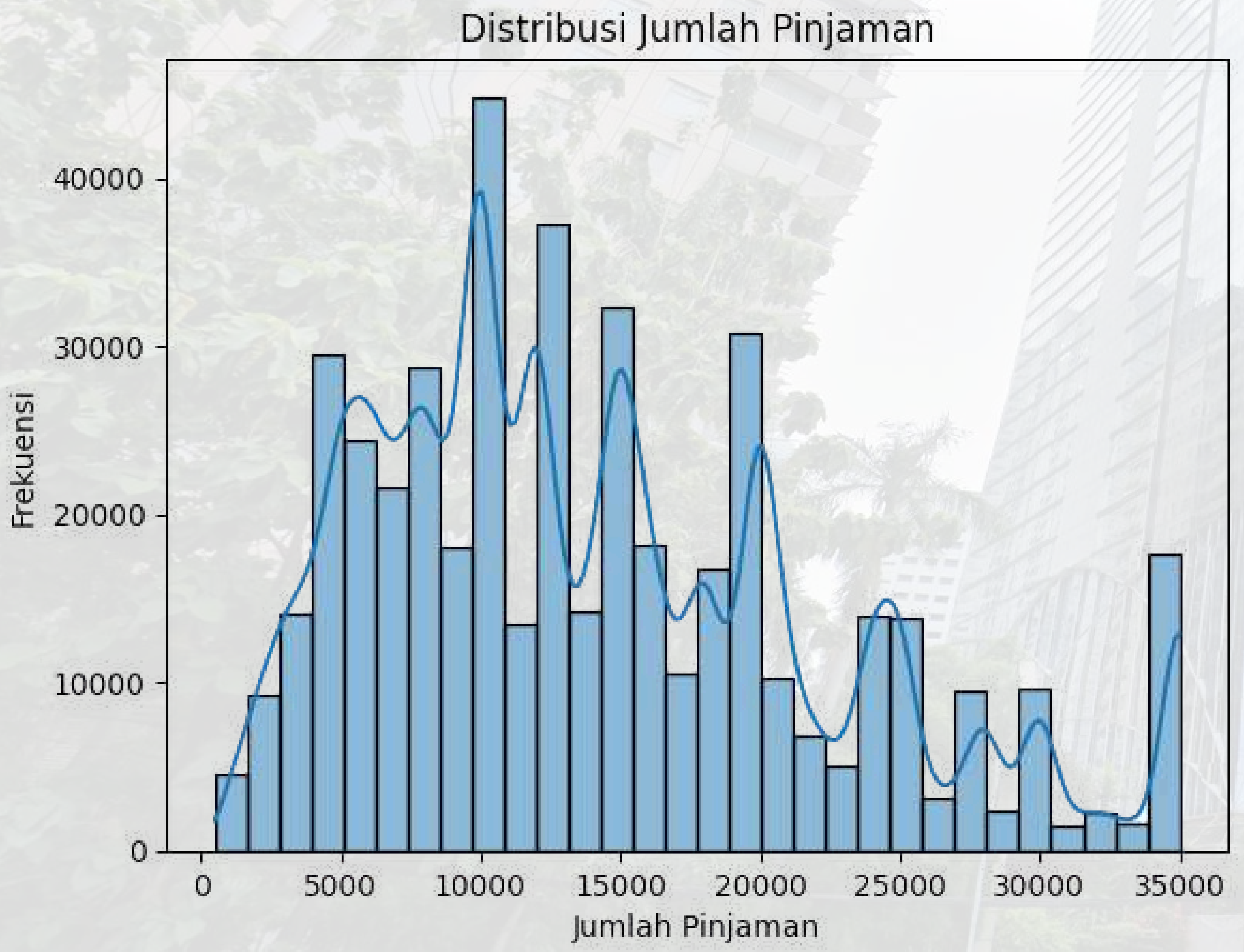
```
annual_inc                          4
delinq_2yrs                        29
inq_last_6mths                     29
mths_since_last_delinq         250351
mths_since_last_record         403647
open_acc                           29
pub_rec                            29
revol_util                        340
total_acc                          29
collections_12_mths_ex_med        145
mths_since_last_major_derog    367311
annual_inc_joint               466285
dti_joint                      466285
verification_status_joint      466285
acc_now_delinq                     29
tot_coll_amt                    70276
tot_cur_bal                     70276
open_acc_6m                    466285
open_il_6m                     466285
open_il_12m                    466285
open_il_24m                    466285
mths_since_rcnt_il             466285
total_bal_il                   466285
il_util                        466285
open_rv_12m                    466285
open_rv_24m                    466285
max_bal_bc                     466285
all_util                       466285
total_rev_hi_lim                70276
inq_fi                         466285
total_cu_tl                    466285
inq_last_12m                   466285
dtype: int64
```

Missing values in numerical variables.

Rakamin Academy X id/x partners

Exploratory Data Analysis

Uncovering Loan Patterns: Distribution Visualization with KDE.

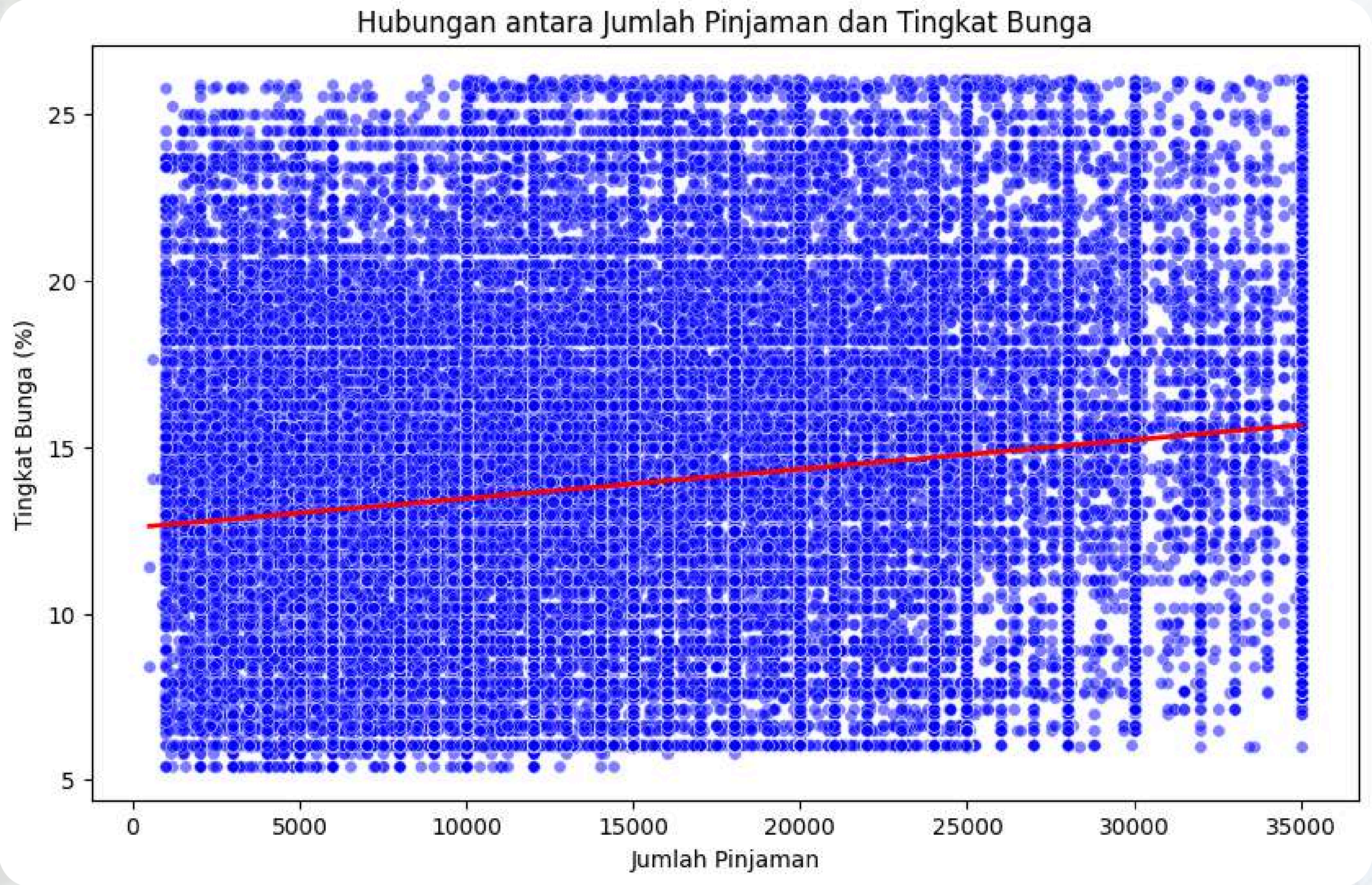


Insights

- The highest loan amount, **around $10,000**, is often associated with **debt consolidation** and **credit card repayment**, as many use loans to manage credit card debt, which typically falls within this range,
- Larger loans, typically **between $20,000 and $35,000**, are commonly used for **home improvements, major purchases, small businesses**, or **down payments,**
- Loan amounts from **$10,000 to $20,000** and **$5,000 to $15,000** are frequently allocated for **cars, medical expenses, weddings, and vacations**, with the latter range also applicable for moving,
- For smaller loans, generally **between $5,000 and $10,000**, funds are often used for **moving, vacations, weddings, and medical expenses.**

Rakamin Academy X id/x partners

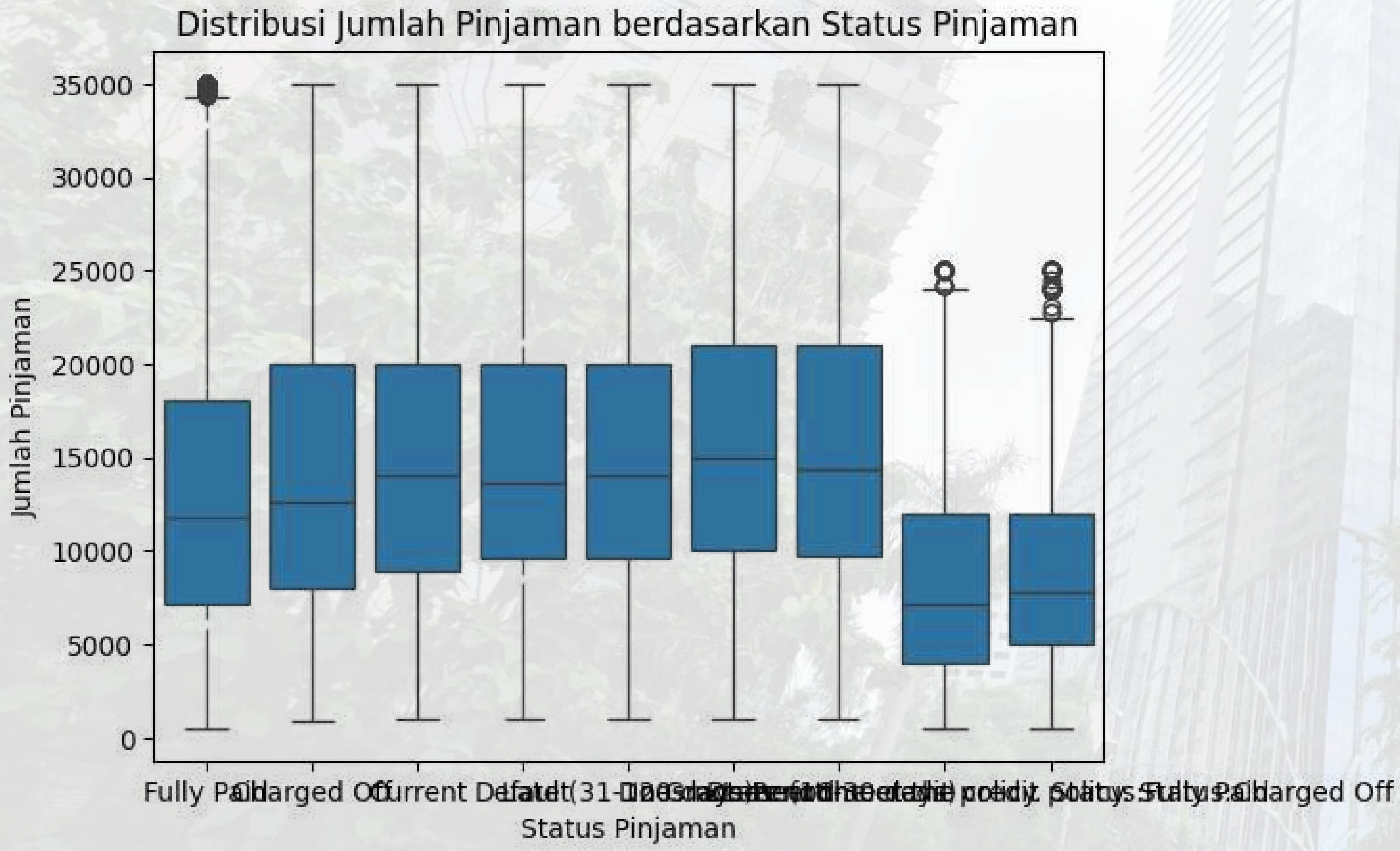
Loans and Interest Rates: The Key to Increasing Your Business Profitability.



Insights

- The data points are dense and clustered, particularly between loan amounts of **$5,000 and $20,000**, indicating that most loans fall within this range,
- The red line shows **a weak positive trend**, suggesting a **slight increase** in interest rates as loan amounts rise; however, this relationship is not strong, implying that loan amounts may **not** be the primary factor influencing interest rates,
- Most points are concentrated in the interest rate **range of 10% to 20%**, indicating this is the most common rate offered,
- At each loan amount level, there is **significant variation in interest rates**; for instance, loans around $10,000 can have rates ranging from **5% to over 25%.** This variation suggests that factors other than loan amounts, such as credit scores, loan purposes, or borrower risk profiles, also influence interest rates.

Boxplot Visualization: Understanding Variation in Loan Amounts in Your Business.

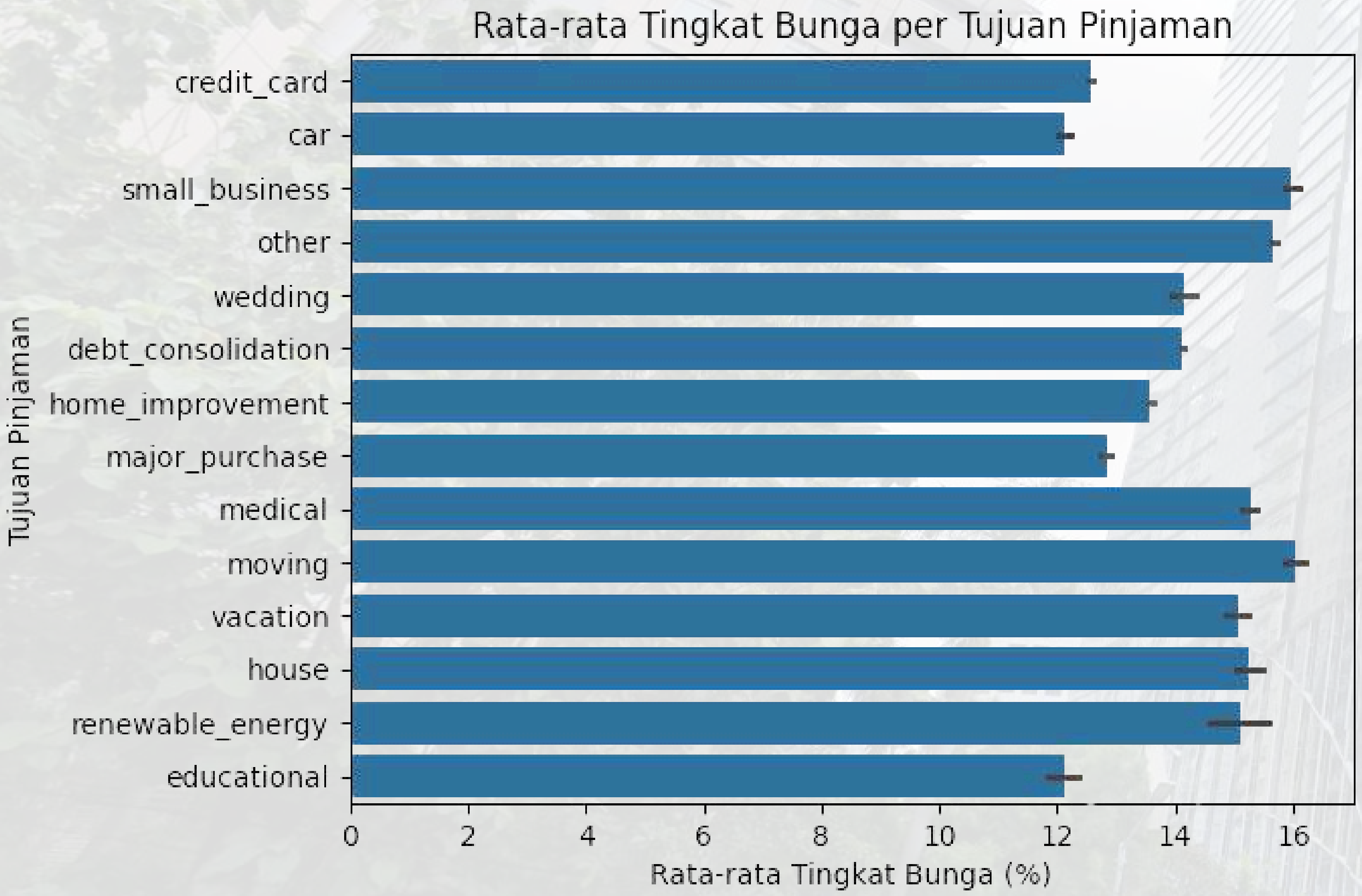


Insights

- The boxplot illustrates the **difference in median loan amounts between loan statuses**. For example, the **Fully Paid** status may have a higher median than the **Default** status,
- Certain loan statuses exhibit **higher variability,** indicated by longer whiskers. This suggests that there is **greater variation** in loan amounts for those statuses,
- The **Default** loan status shows a **lower likelihood of higher** loan amounts. It also **has more outliers**, which could indicate potential risks or repayment issues.

Rakamin Academy X id/x partners

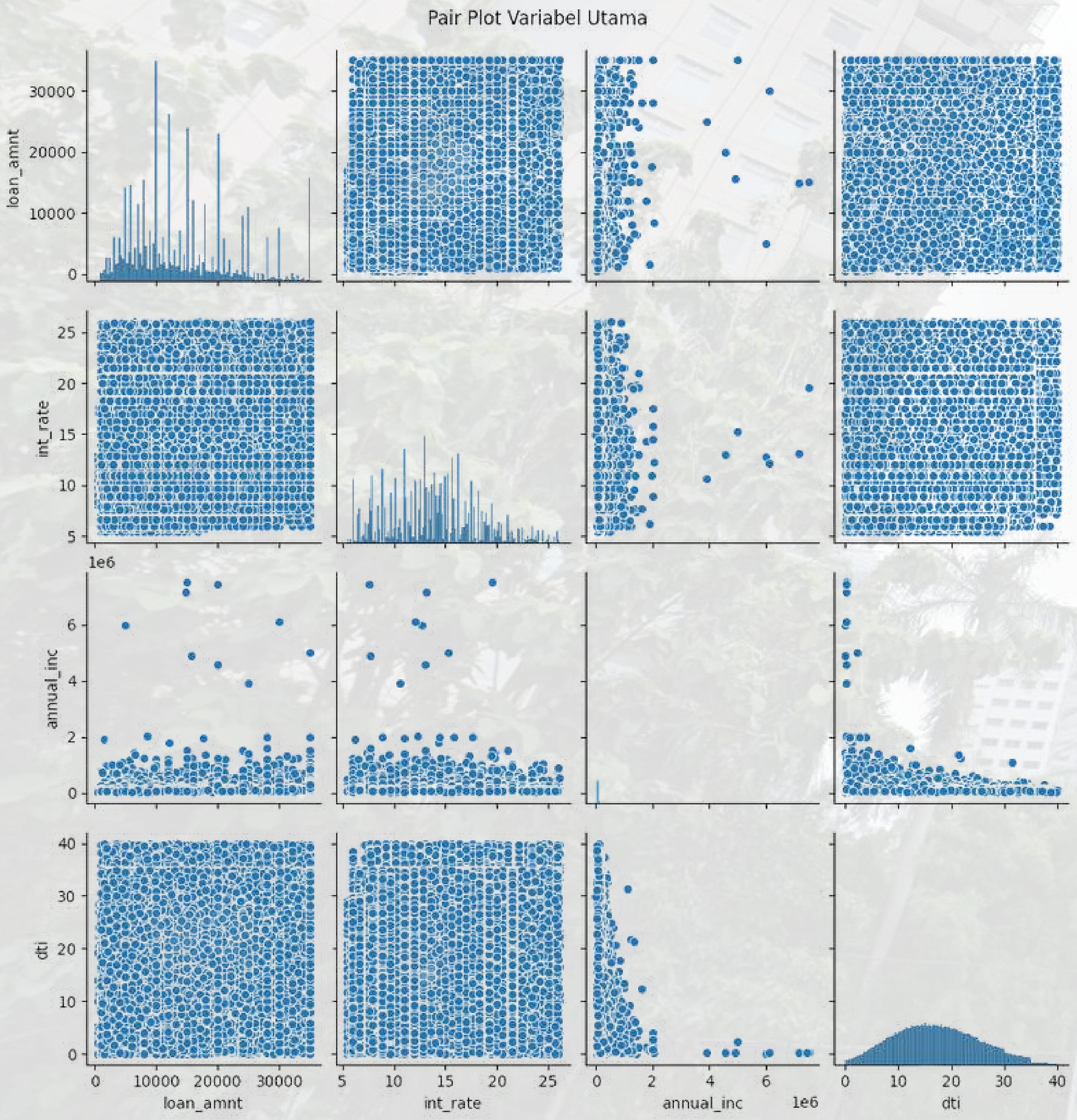
Interest Rates and Loan Purposes: A Visual Guide for Business Owners



Insights

- The graph indicates that loan purposes such as **credit card** and **small business** have **higher average interest** rates, which may **suggest greater risk or higher** costs associated with these types of loans,
- In contrast, loan purposes like **educational** and **renewable energy** tend to have lower interest rates, implying that lenders may **offer more incentives** for these categories,
- The elevated interest rates for certain categories, particularly **small business**, may reflect the increased risk associated with borrowers using funds for business ventures, which **can be more susceptible** to **market fluctuations,**
- High interest rates in specific categories may also signal broader economic conditions. For instance, if many borrowers fall into the **small business** category but face **high interest rates**, it could indicate that lenders perceive investing in this **sector as riskier.**

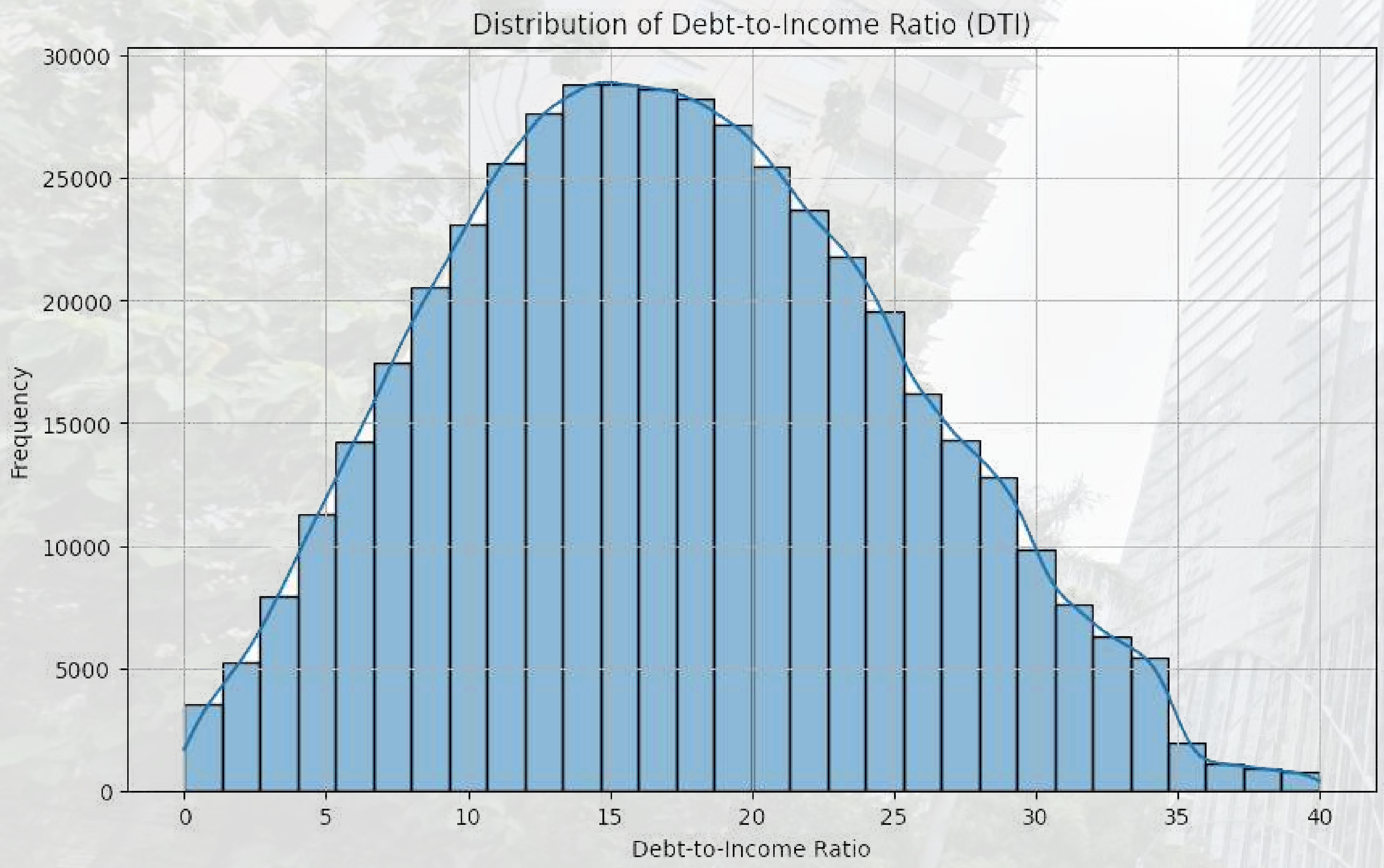
Optimize Your Business Performance: Discover Valuable Insights with Pair Plot



Insights

- The boxplot illustrates the **difference in median loan amounts between loan statuses**. For example, the **Fully Paid** status may have a higher median than the **Default** status,
- Certain loan statuses exhibit **higher variability,** indicated by longer whiskers. This suggests that there is **greater variation** in loan amounts for those statuses,
- The **Default** loan status shows a **lower likelihood of higher** loan amounts. It also **has more outliers**, which could indicate potential risks or repayment issues.

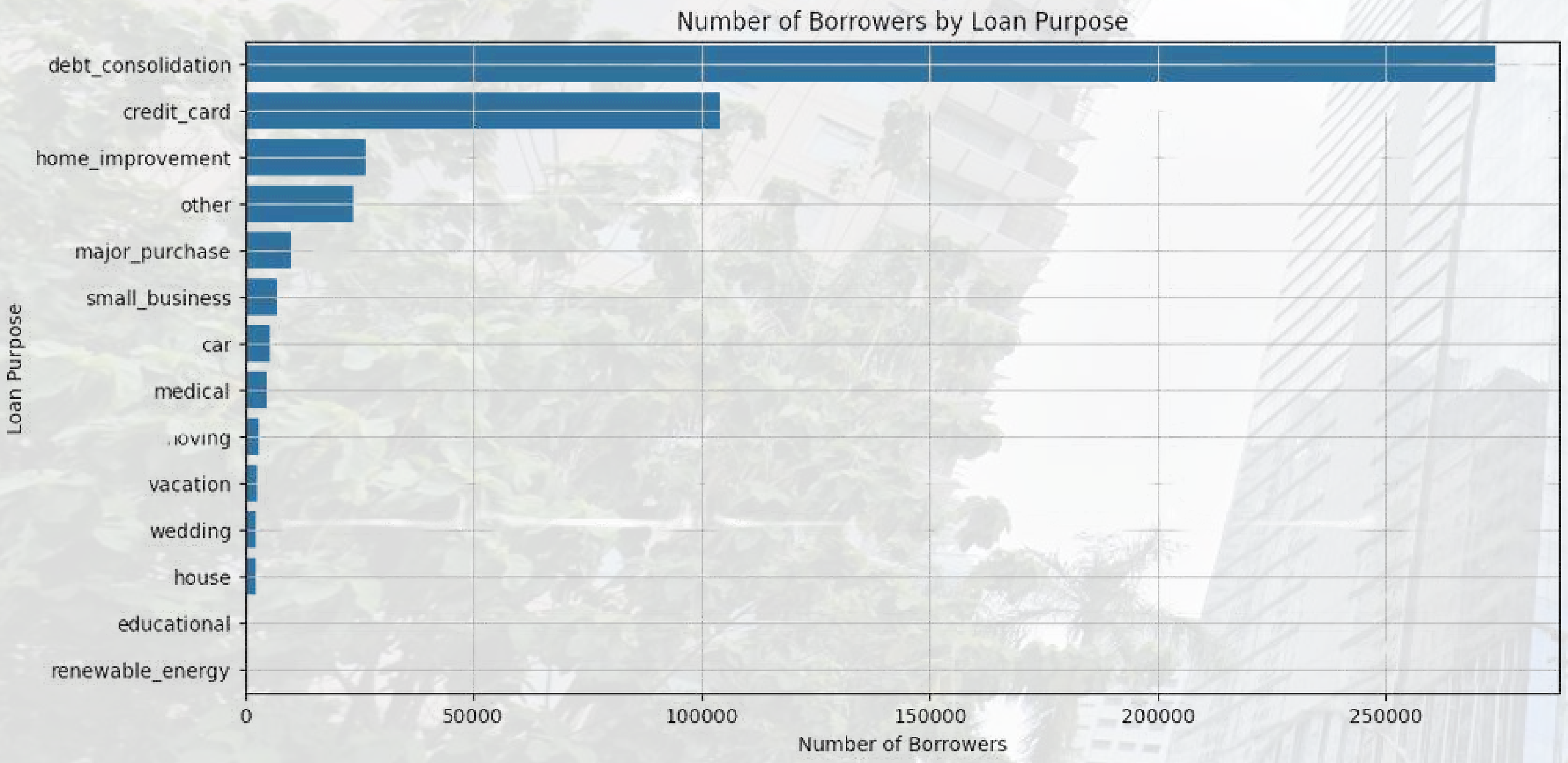
Increasing Profitability: Understanding DTI for Business Owners



Insights

- The histogram reveals that most borrowers have relatively low debt-to-income ratios (DTIs), with a **peak frequency** observed in the 10-15 range. This suggests that many borrowers carry **relatively small** debts compared to their income.
- **A low DTI (e.g., below 20)** signifies that borrowers **have a better capacity** to repay their debts, which can instill confidence in lenders regarding their repayment ability.
- Conversely, **a very high DTI (e.g., above 30)** may indicate **greater risk**, prompting lenders to exercise caution when considering loans to these borrowers.

Uncovering Purpose Loans: Analyze the Number of Borrowers for Your Business

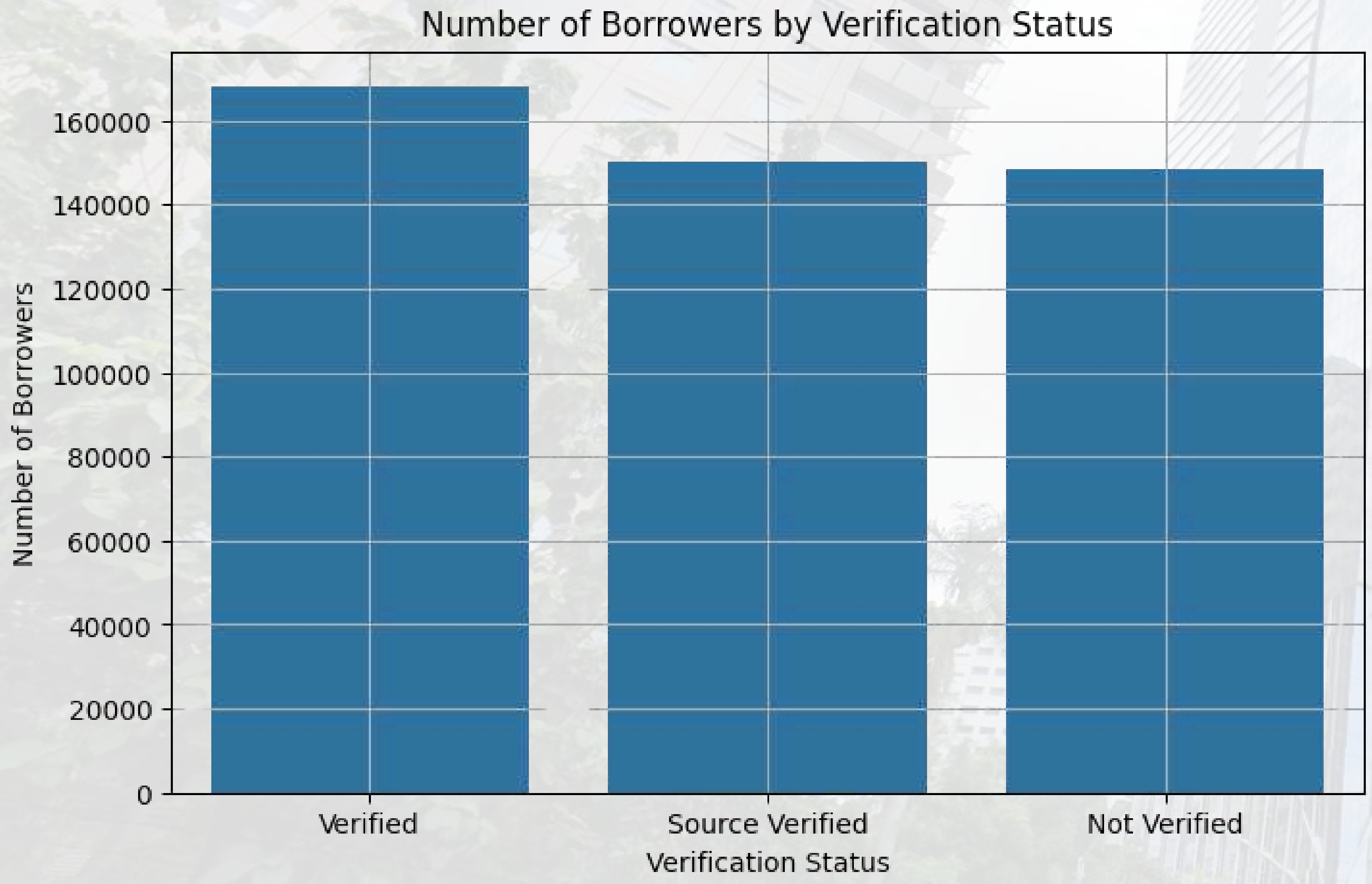


Insights

- The **debt_consolidation** category has a s**ignificantly higher number** of borrowers compared to other categories. This indicates that **many individuals** are utilizing loans to **consolidate their debts**, likely in an effort to reduce their overall debt burden or secure a lower interest rate.
- The **credit card** and **home improvement** categories also show considerable numbers of borrowers, but these figures are much lower than those for **debt consolidation**. This suggests that borrowers are more **inclined to use loans for debt consolidation** rather than for consumer goods or home improvements.
- In contrast, categories such as **educational** and **renewable energy** have **notably** fewer borrowers. This may imply that there is **less interest** in obtaining loans for these purposes, or it could indicate a l**imited availability of loan products tailored** to these categories.

Rakamin Academy X id/x partners

Understanding Trust: Analysis of Borrower Numbers Based on Verification Status



Insights

- The **Verified** category has the **highest number** of borrowers, slightly surpassing the other categories. This suggests that most borrowers have successfully completed the verification process.
- While the **Not Verified** category has fewer borrowers than **Verified**, the number remains significant. This may indicate that many borrowers either do not meet the verification criteria or are still undergoing evaluation.
- If **unverified** borrowers exhibit a higher default rate, this could raise concerns for lenders, prompting them to be more selective in their funding decisions. Conversely, if there are many unverified borrowers who are still trustworthy, there is an opportunity to enhance verification requirements to attract more borrowers to the "Verified" category.

Data Prepocessing

Data Prepocessing

Removing features with lots of 0 data

```
[ ]  data_df = data_df.dropna(axis=1, how='all')
```

Removal of data features that are not very informative for the analysis process

```
data_df.drop(['application_type'], axis=1, inplace=True)
data_df.drop(['zip_code'], axis=1, inplace=True)
data_df.drop(['desc'], axis=1, inplace=True)
data_df.drop(['title'], axis=1, inplace=True)
data_df.drop(['pymnt_plan'], axis=1, inplace=True)
data_df.drop(['member_id'], axis=1, inplace=True)
data_df.drop(['id'], axis=1, inplace=True)
data_df.drop(['Unnamed: 0'], axis=1, inplace=True)
data_df.drop(['url'], axis=1, inplace=True)
```

Converting Datetime

Sometimes, features are converted into numeric form by extracting only the **month**

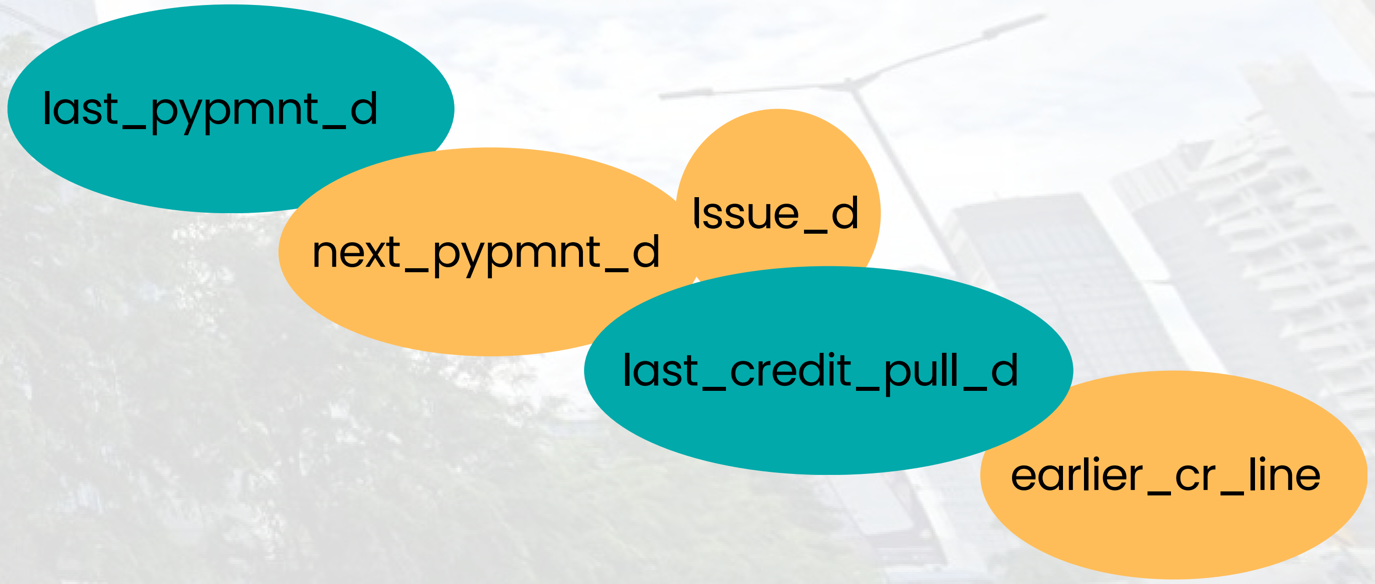
Before

issue_d		last_pymnt_d		next_pymnt_d		last_credit_pull_d		earliest_cr_line	
Oct-14	0.083172	Jan-16	0.385215	Feb-16	0.446922	Jan-16	0.702787	Oct-00	0.007879
Jul-14	0.062850	Dec-15	0.132966	Jan-16	0.059882	Dec-15	0.030007	Aug-00	0.007714
Nov-14	0.053731	Jul-15	0.025098	Mar-11	0.000229	Nov-15	0.017980	Aug-01	0.007410
May-14	0.040960	Oct-15	0.024123	Apr-11	0.000217	Sep-15	0.017266	Oct-99	0.007305
Apr-14	0.040900	Sep-15	0.021884	Feb-11	0.000195	Oct-15	0.017065	Oct-01	0.007139
	...		...		...		...		...
Aug-07	0.000159	Jun-08	0.000043	Oct-14	0.000004	Nov-07	0.000006	Jul-55	0.000002
Jul-07	0.000135	Mar-08	0.000039	Feb-08	0.000004	May-08	0.000002	Feb-57	0.000002
Sep-08	0.000122	Jan-08	0.000024	May-08	0.000002	Jun-08	0.000002	Oct-54	0.000002
Sep-07	0.000114	Feb-08	0.000017	Mar-15	0.000002	Jul-08	0.000002	May-53	0.000002
Jun-07	0.000051	Dec-07	0.000004	Dec-07	0.000002	Jul-07	0.000002	Nov-56	0.000002

After

```
        issue_d_month  last_pymnt_d_month  next_pymnt_d_month  \
0                  12                 1.0                 NaN
1                  12                 4.0                 NaN
2                  12                 6.0                 NaN
3                  12                 1.0                 NaN
4                  12                 1.0                 2.0
...               ...                 ...                 ...
466280              1                 1.0                 2.0
466281              1                12.0                 NaN
466282              1                 1.0                 2.0
466283              1                12.0                 NaN
466284              1                 1.0                 2.0

        last_credit_pull_d_month  earliest_cr_line_month
0                            1.0                     1.0
1                            9.0                     4.0
2                            1.0                    11.0
3                            1.0                     2.0
4                            1.0                     1.0
...                          ...                     ...
466280                       1.0                     4.0
466281                       1.0                     6.0
466282                      12.0                    12.0
466283                       4.0                     2.0
466284                       1.0                     2.0
```



Rakamin Academy X id/x partners

Labelling Process

The **loan_status** feature serves as the prediction target. The classification is done as follows:

Bad → Charged Off | Default | Late (31-120 days) | Late (16-30 days)

👉 **This category is considered high-risk (bad)**, due to late or failed payments.

Good → Fully Paid | Current | In Grace Period

👉 **This category is considered safe (good)** because the borrower demonstrates payment compliance.

Features Engineering

The **loan_status** feature serves as the prediction target. The classification is done as follows:

ordinal Encoding

Converts categories to numbers based on a specific order.

1. term
2. grade
3. sub_grade
4. emp_length
5. verification_status

ordinal Encoding

Converts each category to a unique number (without considering the order).

1. home_ownership
2. purpose
3. addr_state
4. initial_list_status

One Hot Encoding

Converts a categorical variable to a binary number.

.1. loan_status

Rakamin Academy X id/x partners

Train Test Split dan Fitur Scaling

Train Test Split

Divide the dataset into 20% test data and 80% train data.

Fitur Scaling

Feature scaling with Min-Max Scaler is a technique used to normalize the range of features in a dataset. It transforms the data to a specified range, typically [0, 1], by scaling each feature individually based on its minimum and maximum values, ensuring that all features contribute equally to the model.

Rakamin Academy X id/x partners

Data Modelling & Evaluation.

Rakamin Academy X id/x partners

Data Modelling (1) with logistic regression

Training Model → Prediction → Prediction

```
Training set score: 0.9717
Test set score: 0.9716
Logistic Regression Classifier:
              precision    recall  f1-score   support

           0       1.00      0.76      0.86     11051
           1       0.97      1.00      0.98     82206

    accuracy                           0.97     93257
   macro avg       0.98      0.88      0.92     93257
weighted avg       0.97      0.97      0.97     93257

Confusion matrix

 [[ 8427  2624]
 [   23 82183]]

True Positives(TP) =  8427

True Negatives(TN) =  82183

False Positives(FP) =  2624

False Negatives(FN) =  23
```

- Accuracy: 97.17% on training and 97.43% on testing.
- Precision: High, with a recall of 1.00 for the positive class.
- F1-Score: Good, showing a good balance between precision and recall.
- Conclusion: The model is effective in detecting the positive class, with good performance on both datasets. However, false positives are quite significant, which can be an area for improvement.

Rakamin Academy X id/x partners

Data Modelling (1) with random forest classifier

Training Model → Prediction → Prediction

```
Accuracy on Traing set:  0.999989276944358
Accuracy on Testing set:  0.9864353346129513
Random Forest Classifier:
              precision    recall  f1-score   support

           0       1.00      0.89      0.94     11051
           1       0.99      1.00      0.99     82206

    accuracy                           0.99     93257
   macro avg       0.99      0.94      0.97     93257
weighted avg       0.99      0.99      0.99     93257

Confusion matrix

 [[ 8427  2624]
 [   23 82183]]
```

- Accuracy: 99.99% on training and 98.64% on testing.
- Precision: 100% for class 0, 99% for class 1.
- Recall: 89% for class 0, 100% for class 1.
- F1-Score: Good, but recall for class 0 shows room for improvement.
- Conclusion: The model is very effective with high accuracy, but there is some difficulty in detecting negative cases. This shows potential for further performance improvement.

Rakamin Academy X id/x partners

Data Modelling (1) with XGBoost Classifier

Training Model → Prediction → Prediction

```
XGBoost Classifier:
Accuracy on Training set:  0.9893439634558264
Accuracy on Testing set:  0.9870036565619739

Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.90      0.94     11051
           1       0.99      1.00      0.99     82206

    accuracy                           0.99     93257
   macro avg       0.99      0.95      0.97     93257
weighted avg       0.99      0.99      0.99     93257

XGBoost Classifier:
              precision    recall  f1-score   support

           0       0.99      0.90      0.94     11051
           1       0.99      1.00      0.99     82206

    accuracy                           0.99     93257
   macro avg       0.99      0.95      0.97     93257
weighted avg       0.99      0.99      0.99     93257
```

- Accuracy: 98.93% in training and 98.70% in testing.
- Precision: Very high for class 1, 99% for class 0.
- Recall: 1.00 for class 1.
- F1-Score: Very good, showing a balanced performance between precision and recall.
- Conclusion: This model shows very satisfactory results, with excellent ability to detect positive classes and few prediction errors.

Data Modelling (1) with LightGBM

Training Model → Prediction → Prediction

```
LightGBM Regressor:
Accuracy on Training set:  0.893476704156367
Accuracy on Testing set:  0.8828238015070072
[LightGBM] [Info] Number of positive: 328747, number of
[LightGBM] [Info] Auto-choosing row-wise multi-threading
You can set `force_row_wise=true` to remove the overhead
And if memory is not enough, you can set `force_col_wise
[LightGBM] [Info] Total Bins 6349
[LightGBM] [Info] Number of data points in the train set
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.881293
[LightGBM] [Info] Start training from score 2.004733
LightGBM Classifier:
              precision    recall  f1-score   support

           0       0.99      0.89      0.94     11051
           1       0.99      1.00      0.99     82206

    accuracy                           0.99     93257
   macro avg       0.99      0.95      0.97     93257
weighted avg       0.99      0.99      0.99     93257

Confusion matrix

 [[ 8427  2624]
 [   23 82183]]
```

- Accuracy: 89.35% on training and 88.23% on testing.
- Precision: 99% for class 1.
- Recall: 1.00 for class 1.
- F1-Score: Very good, especially for the positive class.
- Conclusion: This model is very effective in detecting the positive class, but has a slightly lower performance on the training data compared to other models. This indicates good generalization ability.

Rakamin Academy X id/x partners

Data Modelling (1) with Decision Tree Classifier

Training Model → Prediction → Prediction

```
Decision Tree Regressor:
Accuracy on Training set:  1.0
Accuracy on Testing set:  0.9742861125706381
Decision Tree Classifier:
              precision    recall  f1-score   support

           0       0.89      0.90      0.89     11051
           1       0.99      0.98      0.99     82206

    accuracy                           0.97     93257
   macro avg       0.94      0.94      0.94     93257
weighted avg       0.97      0.97      0.97     93257

Confusion matrix

 [[ 8427  2624]
 [   23 82183]]
```

- Accuracy: 100% on training and 97.43% on testing.
- Precision: 89% for class 0, 99% for class 1.
- Recall: 90% for class 0, 99% for class 1.
- F1-Score: Good, especially for the positive class.
- Conclusion: The model shows good performance, but there is a risk of overfitting due to the very high training accuracy. Some negative cases are not detected well.

CONCLUSION.

Rakamin Academy X id/x partners

CONCLUSION.

Based on the results explained previously, the XGBoost model and the Random Forest model can be considered as the closest to perfect and very good. Here are the reasons for each model:

1. XGBoost
   - High Accuracy: This model has an accuracy of 98.93% on the training data and 98.70% on the testing data, indicating excellent generalization ability.
   - Precision and Recall: Very high precision (99% for class 0 and 100% for class 1) and perfect recall for class 1 indicate that the model is very effective in detecting positive cases without many errors.
   - F1-Score: A high F1-score indicates a good balance between precision and recall, indicating stable performance across classes.

2. Random Forest
   - Very High Accuracy: This model has an accuracy of 99.99% on the training data and 98.64% on the testing data, also indicating good generalization ability.
   - Perfect Precision: Precision of 100% for class 0 and very high for class 1 (99%) indicates that the model is very accurate in predicting both classes.
   - Recall: Although the recall for class 0 (89%) is slightly lower, the recall for class 1 is 100%, indicating that the model is very good at detecting all positive cases.

Recommendation.

If we have to choose one model that is closest to perfect, XGBoost can be considered the best choice because of its combination of high accuracy, precision, recall, and balanced F1-score.

Thank You

Rakamin Academy X id/x partners