CREDIT RISK CLASSIFICATION: DEVELOPING A PREDICTIVE MODEL FOR LENDING DECISIONS

ID/X Partners – Data Scientist

Presented by:
Mardio Edana Putra

mardiopq@gmail.com

mardiopq99

Courses and Certification

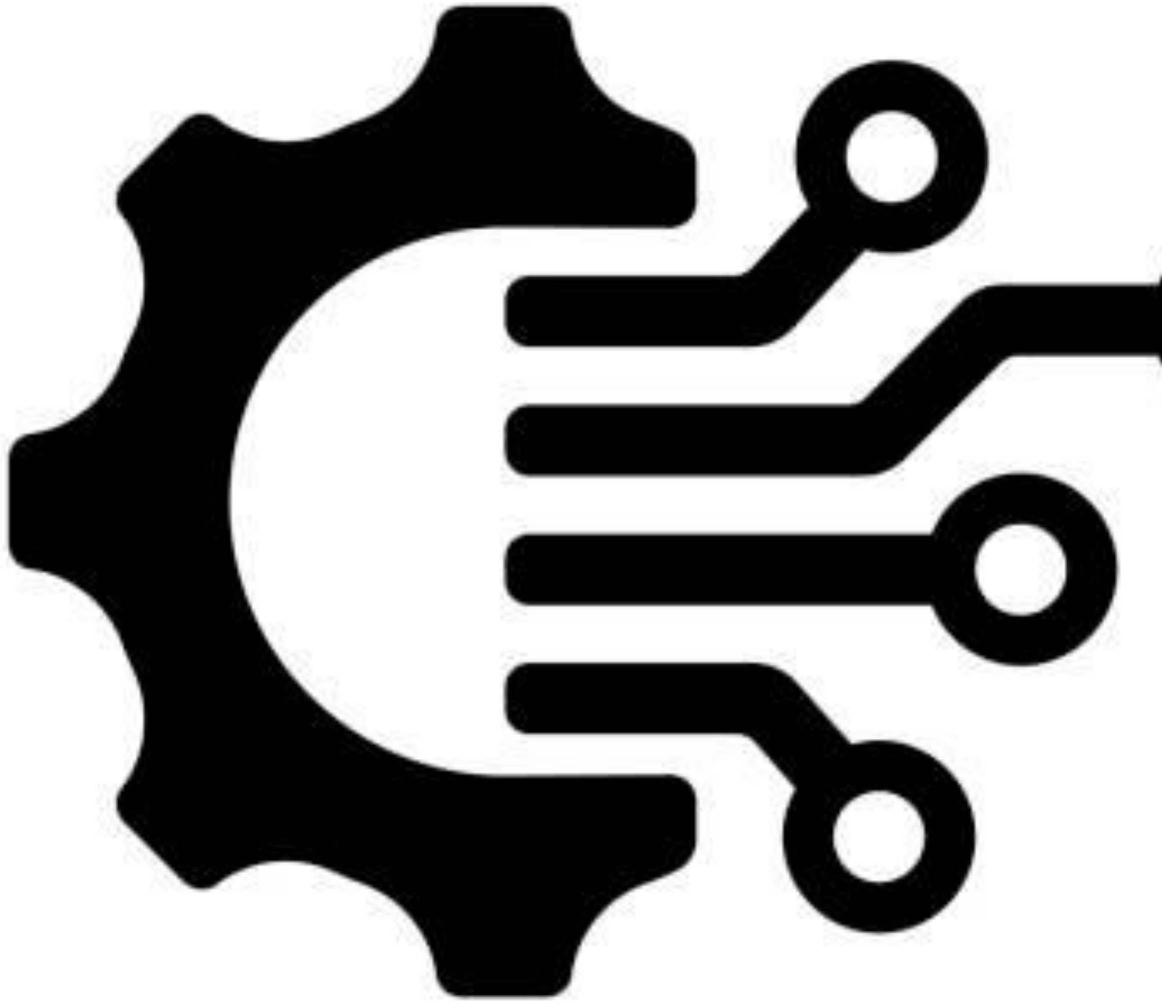
Mardio Edana Putra

Mardio Edana Putra

Data Analyst

I am an aspiring data analyst/scientist with a strong interest in data and its applications in business. I'm passionate about using data to uncover insights that can support better decision-making and drive growth. While I'm currently exploring opportunities, I am eager to apply my skills and knowledge in the world of data

Rakamin Academy



Introduction

Welcome to my portfolio!

Here, you'll find a selection of my projects related to data analysis. These projects showcase my interest in extracting insights from data and applying them to real-world problems, with the goal of driving informed decision-making and delivering impactful solutions.

Education

2017 - 2022
Bandung Institute of Technology
Industrial Engineering

- **GPA 3.38/4.00**
- **Thesis Title: Proposal for Product Quality Improvement of L14 Nails Using Six Sigma Methodology at PT Surabaya Wire**

2014 - 2017
8 Senior High School Jakarta
Math and Science

Experience

•**Leader of Final Project E-Commerce Data Scientist Bootcamp – Rakamin Academy**
August 2024 to January 2025
Led a team in developing machine learning models for an e-commerce case study, achieving the excellence grade of 89.2. Gained experience in SQL, Python, data preprocessing, statistics, machine learning, and data visualization.

•**Quality Control Intern – PT Surabaya Wire**
April 2021 to March 2022
Analyzed nail defects using the DMAIC methodology and recommended improvements to reduce defect rates.

•**Marketing Analyst Intern – PT Enciety Binakarya Cemerlang**
June 2020 to September 2020
Measured participant satisfaction using surveys and statistical methods, providing recommendations to improve training programs.

Skills and Expertise

1. Programming Language
2. Microsoft (Word, Excel, PPT, Visio)
3. Data Analysis
4. Mathematics
5. Data Visualization
6. Machine Learning

About ID/X Partners

ID/X Partners is a **leading data analytics consulting firm** based in Indonesia, specializing in **data-driven solutions** to help businesses optimize their operations and decision-making. With expertise in data science, machine learning, and artificial intelligence, ID/X Partners collaborates with **various industries**, including finance, e-commerce, and telecommunications, to develop innovative and impactful analytical models.

The company is known for its strong capabilities in **predictive modeling, customer analytics, and risk management, providing end-to-end solutions** that transform raw data into actionable insights. By leveraging cutting-edge technologies and industry best practices, ID/X Partners empowers organizations to enhance efficiency, reduce risks, and drive sustainable growth., reduce risks, and drive sustainable growth.

Framework Project

The project follows a structured **Data Science framework** to build a **credit risk classification model**. It begins with **Problem Definition & Business Understanding**, where we identify business objectives and determine key success metrics. In **Data Understanding & Preparation**, we explore borrower, loan, and credit history data while handling missing values, outliers, and inconsistencies. **Exploratory Data Analysis (EDA)** is conducted to uncover patterns and relationships, followed by **Feature Engineering & Selection** to create meaningful variables and retain the most relevant features.

With a refined dataset, we move to **Model Development**, where we train and validate classification models to predict loan risk. **Model Evaluation** ensures performance optimization using appropriate metrics, followed by **Interpretation & Deployment**, where insights are extracted, and a clear presentation is prepared for stakeholders. The final outcome is a **reliable predictive model** that aids in risk assessment, helping the company make data-driven lending decisions.

Objectives & Metrics

Objectives

- **Develop a classification model** to accurately predict loan risk and support better lending decisions.
- **Analyze historical loan data** to identify patterns and key factors influencing loan performance.

Key Success Metrics

•**Accuracy**: Overall model performance.

•**Precision & Recall**: Balance false positives and false negatives.

•**F1-score**: Ensures a reliable trade-off between precision and recall.

Data Description (1)

1		Description	
2	_rec	The total amount	Jumlah total yang dilakukan oleh investor untuk pinjaman itu pada saat itu.
3	acc_now_delinq	The number of	Jumlah akun di mana peminjam sekarang nakal.
4	addr_state	The state provided by	Negara yang disediakan oleh peminjam dalam aplikasi pinjaman
5	all_util	Balance to credit limit	Saldo ke batas kredit untuk semua perdagangan
6	annual_inc	The self-reported	v
7	annual_inc_joint	The combined self-	Penghasilan tahunan yang dilaporkan sendiri gabungan yang disediakan oleh co-peminjam selama pendaftaran
8	application_type	Indicates whether the	
9	collection_recovery_fee	collection fee	Biaya pengumpulan biaya pengumpulan pos
10	collections_12_mths_ex_med	Number of collections	Jumlah koleksi dalam 12 bulan tidak termasuk koleksi medis
11	delinq_2yrs	The number of 30+	Jumlah 30+ hari insiden kenakalan yang lewat dalam file kredit peminjam selama 2 tahun terakhir
12	desc	Loan description	Deskripsi pinjaman yang disediakan oleh peminjam
13	dti_joint	A ratio calculated	Rasio yang dihitung menggunakan total pembayaran bulanan peminjam bersama atas total kewajiban utang, tidak termasuk hipotek dan pinjaman LC yang diminta, dibagi
14	earliest_cr_line	The month the	Bulan jalur kredit yang paling awal yang dilaporkan peminjam dibuka
15	emp_length	Employment length in	Panjang pekerjaan dalam beberapa tahun. Nilai yang mungkin adalah antara 0 dan 10 di mana 0 berarti kurang dari satu tahun dan 10 berarti sepuluh tahun atau lebih.
16	emp_title	The job title supplied	Judul pekerjaan yang disediakan oleh peminjam saat mengajukan pinjaman.*
17	Femp	A ratio calculated	Rasio yang dihitung menggunakan total pembayaran utang bulanan peminjam atas total kewajiban utang, tidak termasuk hipotek dan pinjaman LC yang diminta, dibagi
18	fico_range_high	The upper boundary	Kisaran batas atas fico peminjam dengan pinjaman originasi milik.
19	fico_range_low	The lower boundary	Rentang batas bawah fico peminjam dengan pinjaman originasi milik.
20	funded_amnt	The total amount	Jumlah total yang berkomitmen untuk pinjaman itu pada saat itu.
21	grade	LC assigned loan	**LC menugaskan nilai pinjaman**
22	home_ownership	The home ownership	Status kepemilikan rumah yang disediakan oleh peminjam selama pendaftaran. Nilai -nilai kami adalah: sewa, sendiri, hipotek, lainnya.
23	id	A unique LC assigned	ID yang ditugaskan LC yang unik untuk daftar pinjaman.
24	il_util	Ratio of total current	Rasio total saldo saat ini dengan batas kredit/kredit tinggi pada semua instal acct
25	initial_list_status	The initial listing	Status daftar awal pinjaman. Nilai yang mungkin adalah - utuh, fraksional
26	inq_fi	Number of personal	Jumlah pertanyaan keuangan pribadi
27	inq_last_12m	Number of credit	Jumlah pertanyaan kredit dalam 12 bulan terakhir

Data Description (2)

28	inq_last_6mths	The number of	Jumlah pertanyaan dalam 6 bulan terakhir (tidak termasuk penyelidikan mobil dan hipotek)
29	installment	The monthly payment	Pembayaran bulanan yang terutang oleh peminjam jika pinjaman berasal.
30	int_rate	Indicates if income	Menunjukkan jika pendapatan diverifikasi oleh LC, tidak diverifikasi, atau jika sumber pendapatan diverifikasi
31	is_inc_v		Menunjukkan jika pendapatan diverifikasi oleh LC, tidak diverifikasi, atau jika sumber pendapatan diverifikasi
32	issue_d	The month which the	Bulan yang didanai pinjaman
33	id	The most recent	Bulan terbaru LC menarik kredit untuk pinjaman ini
34	last_fico_range_high	The upper boundary	Rentang batas atas yang ditarik oleh fico terakhir peminjam.
35	last_fico_range_low	The lower boundary	Rentang batas bawah yang dimiliki oleh fico terakhir peminjam.
36	last_pymnt_amnt	Last total payment	Jumlah total pembayaran terakhir yang diterima
37	last_pymnt_d	Last month payment	Bulan lalu pembayaran diterima
38	loan_amnt	Last month payment	Bulan lalu pembayaran diterima
39	loan_status	Current status of the	Status pinjaman saat ini
40	max_bal_bc	Maximum current	Saldo arus maksimum terutang pada semua akun bergulir
41	member_id	Id for the borrower	ID yang ditugaskan LC yang unik untuk anggota peminjam.
42	mths_since_last_delinq	The number of months	Jumlah bulan sejak kenakalan terakhir peminjam.
43	mths_since_last_major_derog	Months since most	Bulan sejak peringkat 90 hari atau lebih buruk terakhir
44	mths_since_last_record	The number of months	Jumlah bulan sejak catatan publik terakhir.
45	mths_since_rcnt_il	Months since most	Bulan sejak akun angsuran terbaru dibuka
46	next_pymnt_d	Next scheduled payment date	Tanggal Pembayaran Terjadwal Berikutnya
47	open_acc	The number of open	Jumlah jalur kredit terbuka dalam file kredit peminjam.
48	open_acc_6m	Number of open	Jumlah perdagangan terbuka dalam 6 bulan terakhir
49	open_il_12m		Jumlah perdagangan terbuka dalam 6 bulan terakhir
50	open_il_24m	Number of	Jumlah akun angsuran yang dibuka dalam 24 bulan terakhir
51	open_il_6m	Number of	Jumlah akun angsuran yang dibuka dalam 12 bulan terakhir
52	open_rv_12m	Number of revolving	Jumlah Perdagangan Revolving Dibuka dalam 12 Bulan Terakhir

Data Description (3)

Rakamin Academy

53	open_rv_24m	Number of revolving	Jumlah perdagangan revolving dibuka dalam 24 bulan terakhir
54	out_prncp	Remaining	Kepala sekolah yang tersisa untuk jumlah total yang didanai
55	out_prncp_inv	Remaining	Kepala sekolah yang tersisa untuk sebagian dari jumlah total yang didanai oleh investor
56	policy_code	publicly available	Policy_code yang tersedia untuk umum = 1
57	pub_rec	Number of derogatory	Jumlah catatan publik yang menghina
58	purpose	A category provided	Kategori yang disediakan oleh peminjam untuk permintaan pinjaman.
59		Indicates if a payment	Menunjukkan jika rencana pembayaran telah diberlakukan untuk pinjaman
60	recoveries	Indicates if a payment	Menunjukkan jika rencana pembayaran telah diberlakukan untuk pinjaman
61	revol_bal	Total credit revolving	Total Saldo Revolving Credit
62	revol_util	Revolving line	Tingkat pemanfaatan jalur bergulir, atau jumlah kredit yang digunakan peminjam relatif terhadap semua kredit revolving yang tersedia.
63	sub_grade	LC assigned loan	LC Ditugaskan Subgrade Pinjaman
64	term	The number of	Jumlah pembayaran atas pinjaman. Nilai dalam beberapa bulan dan dapat berupa 36 atau 60.
65	title	The loan title	Judul pinjaman yang disediakan oleh peminjam
66	tot_coll_amt	Total collection	Total jumlah pengumpulan yang pernah ada
67	tot_cur_bal	Total current balance	Total Saldo Saat Ini dari Semua Akun
68	total_acc	The total number of	Jumlah total jalur kredit saat ini dalam file kredit peminjam
69	total_bal_il	Total current balance	Total saldo saat ini dari semua akun angsuran
70	total_cu_tl	Number of finance	Jumlah Perdagangan Keuangan
71	total_pymnt	Payments received to	Pembayaran diterima hingga saat ini untuk jumlah total yang didanai
72	total_pymnt_inv	Payments received to	Pembayaran diterima hingga saat ini untuk sebagian dari jumlah total yang didanai oleh investor
73	total_rec_int	Interest received to	Bunga diterima hingga saat ini
74	total_rec_late_fee	Late fees received to	Biaya keterlambatan yang diterima hingga saat ini
75	total_rec_prncp	Principal received to	Kepala sekolah diterima hingga saat ini
76	total_rev_hi_lim	Total revolving high	Total Batas Kredit/Kredit Tinggi Revolving
77	url	URL for the LC page	URL untuk halaman LC dengan data daftar.
78	verified_status_joint	Indicates if the co-	Menunjukkan jika pendapatan bersama co-peminjam diverifikasi oleh LC, tidak diverifikasi, atau jika sumber pendapatan diverifikasi
79	zip_code	The first 3 numbers of	3 nomor pertama dari kode pos yang disediakan oleh peminjam dalam aplikasi pinjaman.

1. Exploratory Data Analysis

Dataset Info

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 466285 entries, 0 to 466284
Data columns (total 75 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   Unnamed: 0                   466285 non-null  int64
 1   id                           466285 non-null  int64
 2   member_id                    466285 non-null  int64
 3   loan_amnt                    466285 non-null  int64
 4   funded_amnt                  466285 non-null  int64
 5   funded_amnt_inv              466285 non-null  float64
 6   term                         466285 non-null  object
 7   int_rate                     466285 non-null  float64
 8   installment                  466285 non-null  float64
 9   grade                        466285 non-null  object
 10  sub_grade                    466285 non-null  object
 11  emp_title                    438697 non-null  object
 12  emp_length                   445277 non-null  object
 13  home_ownership               466285 non-null  object
 14  annual_inc                   466281 non-null  float64
 15  verification_status          466285 non-null  object
 16  issue_d                      466285 non-null  object
 17  loan_status                  466285 non-null  object
 18  pymnt_plan                   466285 non-null  object
 19  url                          466285 non-null  object
 20  desc                         125981 non-null  object
 21  purpose                      466285 non-null  object
 22  title                        466264 non-null  object
 23  zip_code                     466285 non-null  object
 24  addr_state                   466285 non-null  object
 25  dti                          466285 non-null  float64
 26  delinq_2yrs                  466256 non-null  float64
 27  earliest_cr_line             466256 non-null  object
 28  inq_last_6mths               466256 non-null  float64
 29  mths_since_last_delinq       215934 non-null  float64
 30  mths_since_last_record       62638 non-null   float64
 31  open_acc                     466256 non-null  float64
 32  pub_rec                      466256 non-null  float64
 33  revol_bal                    466285 non-null  int64
 34  revol_util                   465945 non-null  float64
 35  total_acc                    466256 non-null  float64
 36  initial_list_status          466285 non-null  object
 37  out_prncp                    466285 non-null  float64
 38  out_prncp_inv                466285 non-null  float64
 39  total_pymnt                  466285 non-null  float64
 40  total_pymnt_inv              466285 non-null  float64
 41  total_rec_prncp              466285 non-null  float64
 42  total_rec_int                466285 non-null  float64
 43  total_rec_late_fee           466285 non-null  float64
 44  recoveries                   466285 non-null  float64
 45  collection_recovery_fee      466285 non-null  float64
 46  last_pymnt_d                 465909 non-null  object
 47  last_pymnt_amnt              466285 non-null  float64
 48  next_pymnt_d                 239071 non-null  object
 49  last_credit_pull_d           466243 non-null  object
 50  collections_12_mths_ex_med   466140 non-null  float64
 51  mths_since_last_major_derog  98974 non-null   float64
 52  policy_code                  466285 non-null  int64
 53  application_type             466285 non-null  object
 54  annual_inc_joint             0 non-null       float64
 55  dti_joint                    0 non-null       float64
 56  verification_status_joint    0 non-null       float64
 57  acc_now_delinq               466256 non-null  float64
 58  tot_coll_amt                 396009 non-null  float64
 59  tot_cur_bal                  396009 non-null  float64
 60  open_acc_6m                  0 non-null       float64
 61  open_il_6m                   0 non-null       float64
 62  open_il_12m                  0 non-null       float64
 63  open_il_24m                  0 non-null       float64
 64  mths_since_rcnt_il           0 non-null       float64
 65  total_bal_il                 0 non-null       float64
 66  il_util                      0 non-null       float64
 67  open_rv_12m                  0 non-null       float64
 68  open_rv_24m                  0 non-null       float64
 69  max_bal_bc                   0 non-null       float64
 70  all_util                     0 non-null       float64
 71  total_rev_hi_lim             396009 non-null  float64
 72  inq_fi                       0 non-null       float64
 73  total_cu_tl                  0 non-null       float64
 74  inq_last_12m                 0 non-null       float64
dtypes: float64(46), int64(7), object(22)
memory usage: 266.8+ MB
```

The dataset contains **466,285 rows and 75 columns**. It includes **46 numerical columns (float64)** such as int_rate, installment, dti, total_pymnt, etc., **7 integer columns (int64)** like id, member_id, loan_amnt, etc., and **22 categorical columns (object)** such as term, grade, sub_grade, loan_status, etc. There are **19 columns with missing values**, including emp_title (27,588 missing), emp_length (21,008 missing), annual_inc (4 missing), and desc (340,304 missing). Additionally, **13 columns are entirely empty** with no non-null values. The likely **target column** for analysis is loan_status, which indicates the loan status (e.g., "Charged Off", "Fully Paid").

Target Variable

Rakamin Academy

To create the new target feature status_loan, we classify loan statuses into two categories:

- **Good Loan (1):**
  - Current, Fully Paid, Does not meet the credit policy. Status: Fully Paid (Loans that are either active with on-time payments or fully paid off).
- **Bad Loan (0):**
  - Charged Off, Default, Late (31-120 days), Late (16-30 days), In Grace Period, Does not meet the credit policy. Status: Charged Off (Loans that are in default, severely delayed, or written off as a loss).

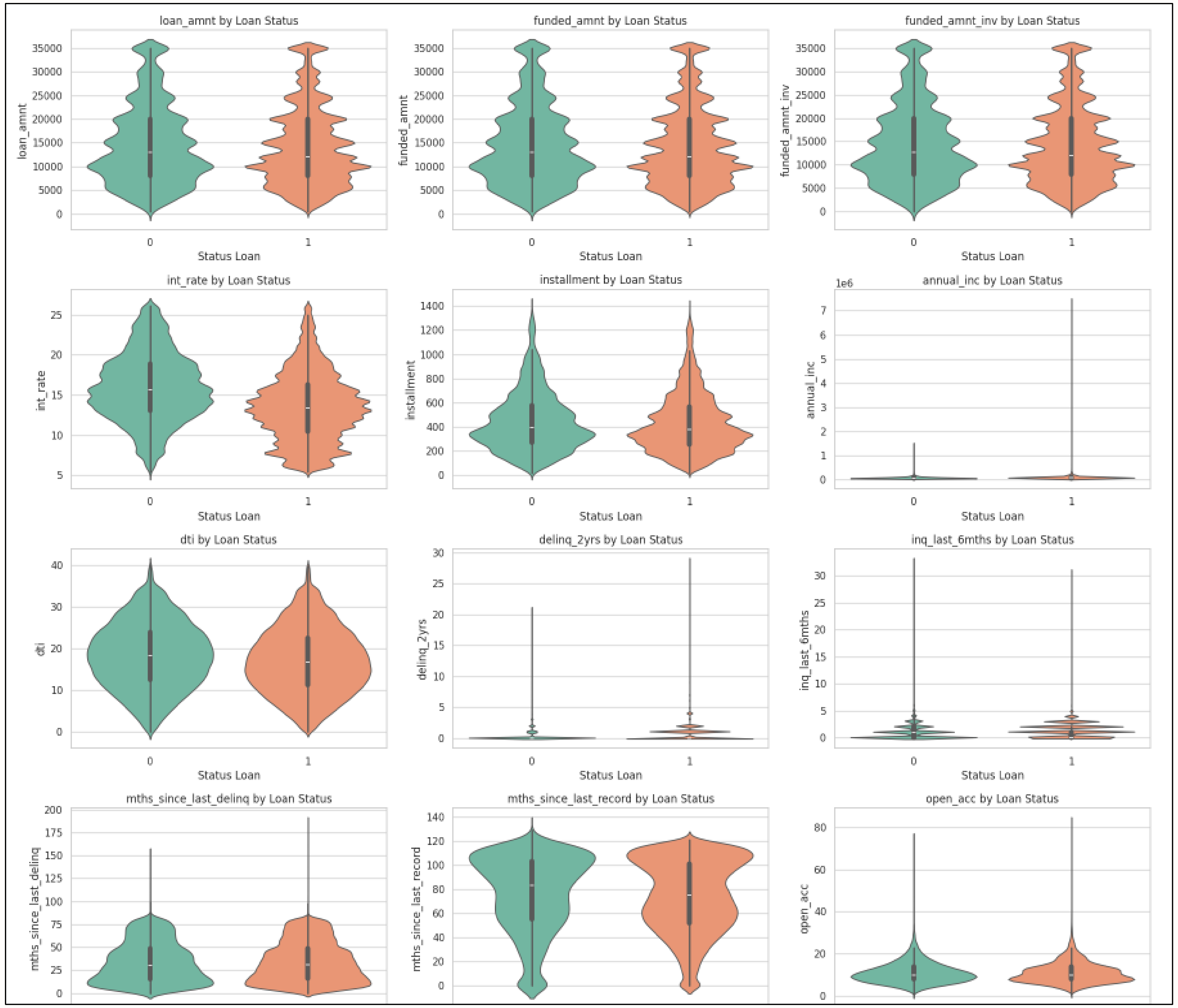
This classification creates a binary feature (1 = Good, 0 = Bad) for further analysis or predictive modeling.

	count
status_loan	
1	410953
0	55332

dtype: int64

Univariate Analysis

Rakamin Academy



◈ General Distribution

Most columns are right-skewed, with smaller values dominating and many large outliers. Examples include annual_inc, revol_bal, and total_pymnt.

◈ Loan & Payment

The distributions of loan_amnt, funded_amnt, and installment are similar and symmetric, while total_pymnt and out_prncp show significant outliers.

◈ Income & Debt

annual_inc varies widely, with many borrowers in the low to middle-income range. dti is generally below 20, with a few very high values.

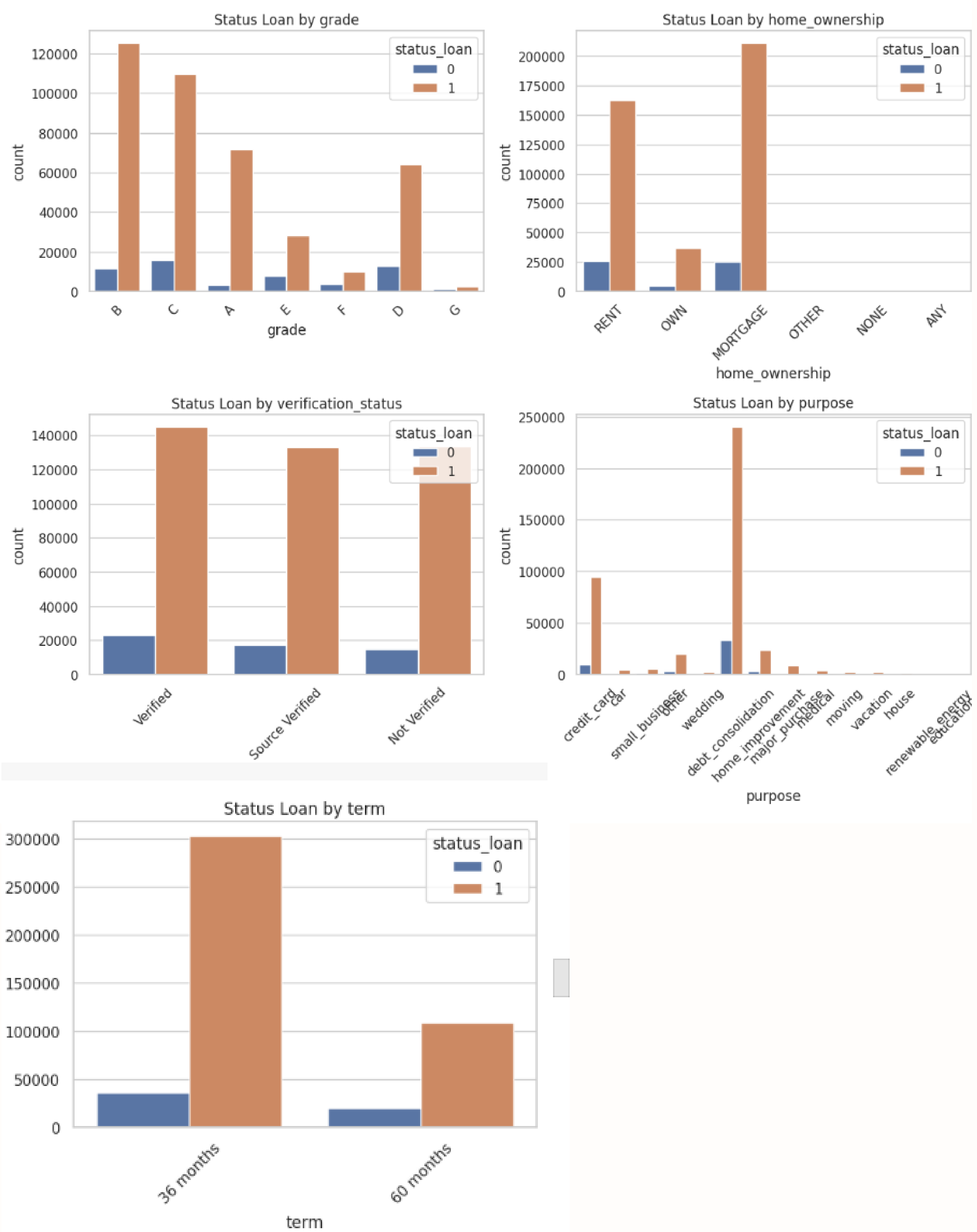
◈ Credit History

Columns like inq_last_6mths, delinq_2yrs, and pub_rec contain many zeros, indicating clean credit histories, while open_acc and total_acc are generally low with a few outliers.

◈ Significant Outliers

Clear outliers are visible in columns like annual_inc, revol_bal, revol_util, and total_pymnt.

EDA Features vs Loan Status



☑ EDA Conclusion on Loan Status (Approved vs Rejected)

1.Grade: Most loans are approved in grades **B, C, and A**, while lower grades like **F and G** have higher rejection rates, indicating that lower credit scores are less likely to be approved.

2.Home Ownership: Borrowers with **MORTGAGE** and **RENT** status dominate loan approvals. However, **OWN** (homeownership) status has a higher rejection rate, possibly due to risk profiles or age-related factors.

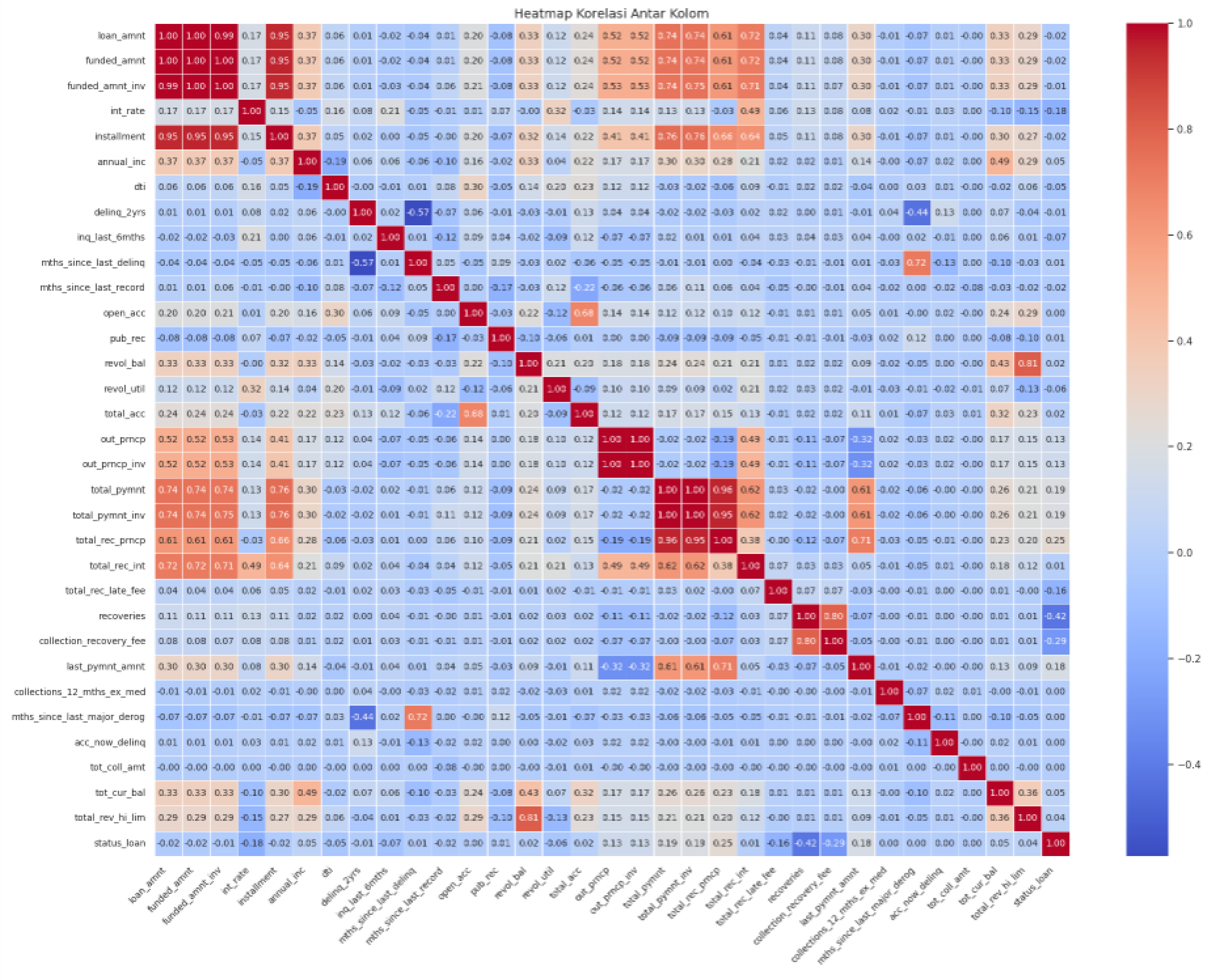
3.Term: Loans with a **36-month term** are more frequently approved compared to **60-month loans**, as longer-term loans tend to be riskier and have higher rejection rates.

4.Verification Status: Most approved loans come from users who are **Verified** or **Source Verified**, while those with **Not Verified** status have a higher rejection rate, highlighting the impact of verification on loan approval.

5.Purpose: Loans for **Debt consolidation** and **credit card** purposes are most frequently approved, whereas loans for **small business**, **education**, and **house** purposes tend to be rejected more often, indicating higher risks associated with these loan purposes.

Multivariate Analysis

Rakamin Academy



1. 🔁 **loan_amnt, funded_amnt, funded_amnt_inv, and installment** are highly correlated with each other → choose one to avoid multicollinearity.
2. 💸 **total_pymnt, total_pymnt_inv, total_rec_prncp, and out_prncp** are strongly correlated → only one or two should be kept.
3. 📈 **int_rate** is relevant and not highly correlated with other features → should be retained.
4. ⚠ **mths_since_last_delinqu vs delinq_2yrs** show a negative correlation → does not directly cause multicollinearity, but caution is needed if used together.
5. □ Features like **pub_rec, recoveries, collection_recovery_fee, and collections_12_mths_ex_med** have low correlation with others → they don't cause multicollinearity but may not be very informative.

2. Data Pre-Processing

Missing Values

```
loan_amnt                      0
funded_amnt                    0
funded_amnt_inv                0
term                           0
int_rate                       0
installment                    0
grade                          0
emp_length                 21008
home_ownership                 0
annual_inc                     4
verification_status            0
issue_d                        0
loan_status                    0
pymnt_plan                     0
purpose                        0
addr_state                     0
dti                            0
delinq_2yrs                   29
earliest_cr_line              29
inq_last_6mths                29
mths_since_last_delinq    250351
mths_since_last_record    403647
open_acc                      29
pub_rec                       29
revol_bal                      0
revol_util                   340
total_acc                     29
initial_list_status            0
out_prncp                      0
out_prncp_inv                  0
total_pymnt                    0
total_pymnt_inv                0
total_rec_prncp                0
total_rec_int                  0
total_rec_late_fee             0
recoveries                     0
collection_recovery_fee        0
last_pymnt_d                 376
last_pymnt_amnt                0
next_pymnt_d              227214
last_credit_pull_d            42
collections_12_mths_ex_med   145
mths_since_last_major_derog 367311
acc_now_delinq                29
tot_coll_amt               70276
tot_cur_bal                70276
total_rev_hi_lim           70276
status_loan                    0
dtype: int64
```

Handling Missing Values

1.Imputation (for columns with small or moderate missing values):

- **emp_length**: Imputed with **"<1 year"** (assumed to be inexperienced).
- **last_pymnt_d, last_credit_pull_d, earliest_cr_line**: Imputed using **mode** (most frequent value).
- **annual_inc** (4 missing): Imputed with **median**.
- **delinq_2yrs, inq_last_6mths, open_acc, pub_rec, total_acc, revol_util, collections_12_mths_ex_med, acc_now_delinq**: All have less than 0.1% missing values → imputed using **median** (numerical columns).

2.Dropped Columns (due to a large amount of missing data):

- **mths_since_last_delinq** (~53.7% missing), **mths_since_last_record** (~86.5% missing), and **mths_since_last_major_derog** (~78.8% missing) were dropped because they have a large percentage of missing values.
- **tot_coll_amt, tot_cur_bal, total_rev_hi_lim, next_pymnt_d**: Dropped due to around **15% missing values** and because these columns may not provide enough information for analysis.

Duplicated Data

Cek Duplikasi Data

```
# Mengecek jumlah baris duplikat
duplicate_rows = df_cleaned2.duplicated().sum()
print(f"Jumlah duplikasi: {duplicate_rows}")
```

```
Jumlah duplikasi: 0
```

Data doesn’t have any duplicated data

Feature Engineering

1. **Loan Age (loan_age)**:
   - Calculates the loan's age in days by subtracting the **issue date** from the **last payment date**.
2. **Credit History (credit_history)**:
   - Calculates the borrower's credit history length in days by subtracting the **earliest credit line date** from the **issue date**.
3. **Days Since Last Credit Pull (days_since_last_credit_pull)**:
   - Calculates how many days have passed since the **last credit pull** by subtracting the **issue date** from the **last credit pull date**.
4. **Dropping Unnecessary Columns**:
   - The original date columns (issue_d, earliest_cr_line, last_pymnt_d, last_credit_pull_d) are dropped since they are no longer needed after creating the new features..

Feature Selection

Features	MI Scores
loan_status_Current	0.155535
recoveries	0.124102
collection_recovery_fee	0.115469
loan_status_Fully Paid	0.113755
total_rec_prncp	0.102211
purpose_debt_consolidation	0.069840
home_ownership_MORTGAGE	0.063449
last_pymnt_amnt	0.045936
loan_age	0.045826
home_ownership_RENT	0.039876
total_pymnt	0.037010
total_pymnt_inv	0.035912
loan_status_Late (31-120 days)	0.033358
initial_list_status	0.031871
verification_status_Verified	0.031430

grade	0.030241
verification_status_Source Verified	0.026068
out_prncp	0.025807
out_prncp_inv	0.025208
int_rate	0.021932
term	0.018999
days_since_last_credit_pull	0.016874
loan_status_In Grace Period	0.014126
purpose_credit_card	0.014017
total_rec_late_fee	0.010696
inq_last_6mths	0.010450
emp_length	0.008665
funded_amnt	0.006451
addr_state_CA	0.006056
open_acc	0.005677
loan_status_Late (16-30 days)	0.005391

This code performs feature selection using **Mutual Information (MI) Scores**, which measure **the dependency between each feature and the target variable**. A new DataFrame (mi_data) is created to store the feature names alongside their corresponding MI scores. The features are then sorted in descending order of importance, helping to identify which variables carry the most useful information for predicting the target. This technique is particularly helpful for filtering out irrelevant or less informative features before model training, potentially improving model performance and interpretability. In the end, **the top 25 features** with the highest MI scores will be selected for modeling.

After that, a **multivariate analysis is conducted** to assess correlation and redundancy between features, and **5 more features are dropped (total_pymnt_inv, out_prncp_inv, out_prncp, purpose_credit_card, and total_rec_late_fee),** resulting in a **final set of 20 features** used for modeling.

Label Encoding

```python
from sklearn.preprocessing import LabelEncoder

# Salin data asli
label_encoded_data = data.copy()

# Kolom-kolom yang akan di-label encode
label_encode_cols = ['term', 'grade', 'emp_length', 'pymnt_plan', 'initial_list_status']

# Simpan encoder kalau butuh inverse transform nanti
label_encoders = {}

for col in label_encode_cols:
    le = LabelEncoder()
    label_encoded_data[col] = le.fit_transform(label_encoded_data[col])
    label_encoders[col] = le
```

This code applies label encoding to five categorical features: **'term', 'grade', 'emp_length', 'pymnt_plan', and 'initial_list_status'.** These features are **either ordinal or contain only two categories**, making them suitable for label encoding, which converts them into numerical values for easier processing by machine learning models while preserving meaningful order or distinction.

One Hot Encoding

```python
# Salin data hasil label encode dulu
onehot_encoded_data = label_encoded_data.copy()

# Kolom-kolom yang akan di-one-hot encode
onehot_encode_cols = ['home_ownership', 'verification_status', 'loan_status', 'purpose', 'addr_state']

# One-hot encode, drop_first=True untuk menghindari multikolinearitas
onehot_encoded_data = pd.get_dummies(onehot_encoded_data, columns=onehot_encode_cols, drop_first=True)

# Konversi True/False ke 1/0
onehot_encoded_data = onehot_encoded_data.astype(int)
```

This code performs one-hot encoding on selected categorical features: **'home_ownership', 'verification_status', 'loan_status', 'purpose', and 'addr_state'.** These features are **nominal (non-ordinal) with multiple categories,** making them suitable for one-hot encoding, which creates binary columns for each category. The drop_first=True parameter is used to avoid multicollinearity by dropping one category per feature. Finally, all True/False values in the dataset are converted to integers (1/0) to ensure compatibility with machine learning algorithms.

Split Data

```python
# Split
from sklearn.model_selection import train_test_split

X = df_modeling.drop('status_loan', axis=1)
y = df_modeling['status_loan']  # ini udah Series

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

The dataset is split into **80% training and 20% testing sets** using train_test_split, with stratification on the target variable status_loan to preserve the class distribution. This ensures that both sets have a similar proportion of each class, which is important for fair model evaluation.

Standardization

```python
from sklearn.preprocessing import StandardScaler

# Daftar kolom yang perlu distandardisasi
scale_cols = ['recoveries', 'collection_recovery_fee', 'total_rec_prncp', 'last_pymnt_amnt',
              'loan_age', 'total_pymnt', 'int_rate', 'days_since_last_credit_pull']

# Simpan kolom sisanya
non_scale_cols = [col for col in X.columns if col not in scale_cols]

# Standardisasi kolom numerik
scaler = StandardScaler()
X_train_scaled = X_train.copy()
X_test_scaled = X_test.copy()

X_train_scaled[scale_cols] = scaler.fit_transform(X_train[scale_cols])
X_test_scaled[scale_cols] = scaler.transform(X_test[scale_cols])
```

This code standardizes only selected numerical columns that have varied scales and continuous values (like **recoveries, int_rate, loan_age, etc.)** to ensure consistency for modeling. **Not all columns are scaled**—only those where differences in scale might bias the model. Categorical and already encoded features are left unchanged, as scaling them is unnecessary and may distort their meaning.

3. Modeling

Model Initiation

```python
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score

# Inisiasi model
models = {
    "Logistic Regression": LogisticRegression(max_iter=1000),
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "Random Forest": RandomForestClassifier(random_state=42),
    "XGBoost": XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42)
}
```

The code initializes four machine learning models to compare their performance in classifying loan status. This model setup is essential to identify the best-performing algorithm for the given problem. The models include:

•**Logistic Regression**, a simple and interpretable linear model,

•**Decision Tree**, which handles non-linear relationships and is easy to visualize,

•**Random Forest**, an ensemble method that improves accuracy and reduces overfitting, and

•**XGBoost**, a powerful gradient boosting model known for its high performance and robustness to outliers.

Modeling Result

Model	Accuracy Train	Accuracy Test	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
Logistic Regression	0.99862	0.99865	0.99862	0.99865	0.99862	0.99865	0.99862	0.99865
Decision Tree	1.00000	0.99925	1.00000	0.99925	1.00000	0.99925	1.00000	0.99925
Random Forest	1.00000	0.99940	1.00000	0.99940	1.00000	0.99940	1.00000	0.99940
XGBoost	0.99999	0.99969	0.99999	0.99969	0.99999	0.99969	0.99999	0.99969

In this case, the **key evaluation metric is precision**, because we want to **minimize false positives**—we want to avoid incorrectly labeling high-risk borrowers as low-risk, which could result in **financial loss**. Precision tells us the proportion of true positive predictions among all positive predictions, making it highly relevant when the cost of false approval is high.

Based on the results, all models perform very well overall, with precision scores above 0.998 on the test set. However, **XGBoost achieves the highest precision (0.99969)**, indicating that its predictions of "good" loans are the most trustworthy. It also performs efficiently with a relatively short training time of 5.02 seconds. Given its superior precision and practical runtime, **XGBoost is the most suitable model for this use case.**

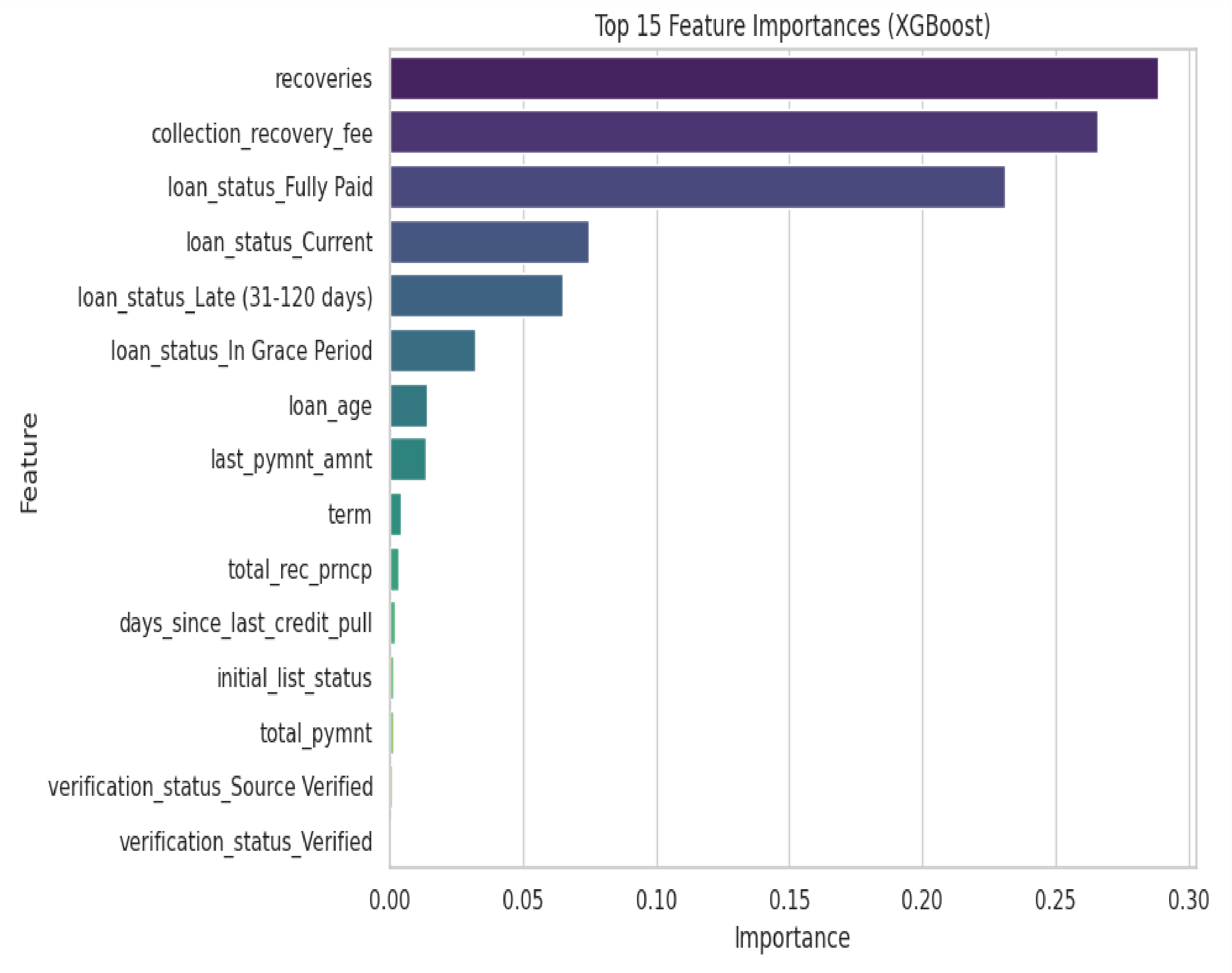
Hyperparameter Tuning

Rakamin Academy

Model	Accuracy Train	Accuracy Test	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
Logistic Regression Tuned	0.99856	0.99856	0.99856	0.99856	0.99856	0.99856	0.99856	0.99856
Decision Tree Tuned	0.99965	0.99924	0.99965	0.99924	0.99965	0.99924	0.99965	0.99924
Random Forest Tuned	0.99998	0.99940	0.99998	0.99940	0.99998	0.99940	0.99998	0.99940
XGBoost Tuned	0.99979	0.99964	0.99979	0.99964	0.99979	0.99964	0.99979	0.99964

After **hyperparameter tuning**, the performance metrics across all models showed **only slight improvements**, indicating consistent model stability. However, **XGBoost** stood out by achieving the **highest test precision (0.99964)** — a crucial metric for this case, as the main goal is to **minimize false positives**. Considering the **previous imbalance in the target variable**, **XGBoost becomes even more favorable** due to its **robustness in handling imbalanced data**, **resilience to outliers**, and ability to model **complex, non-linear relationships**. These strengths make it the **most reliable and effective model** for our classification task.

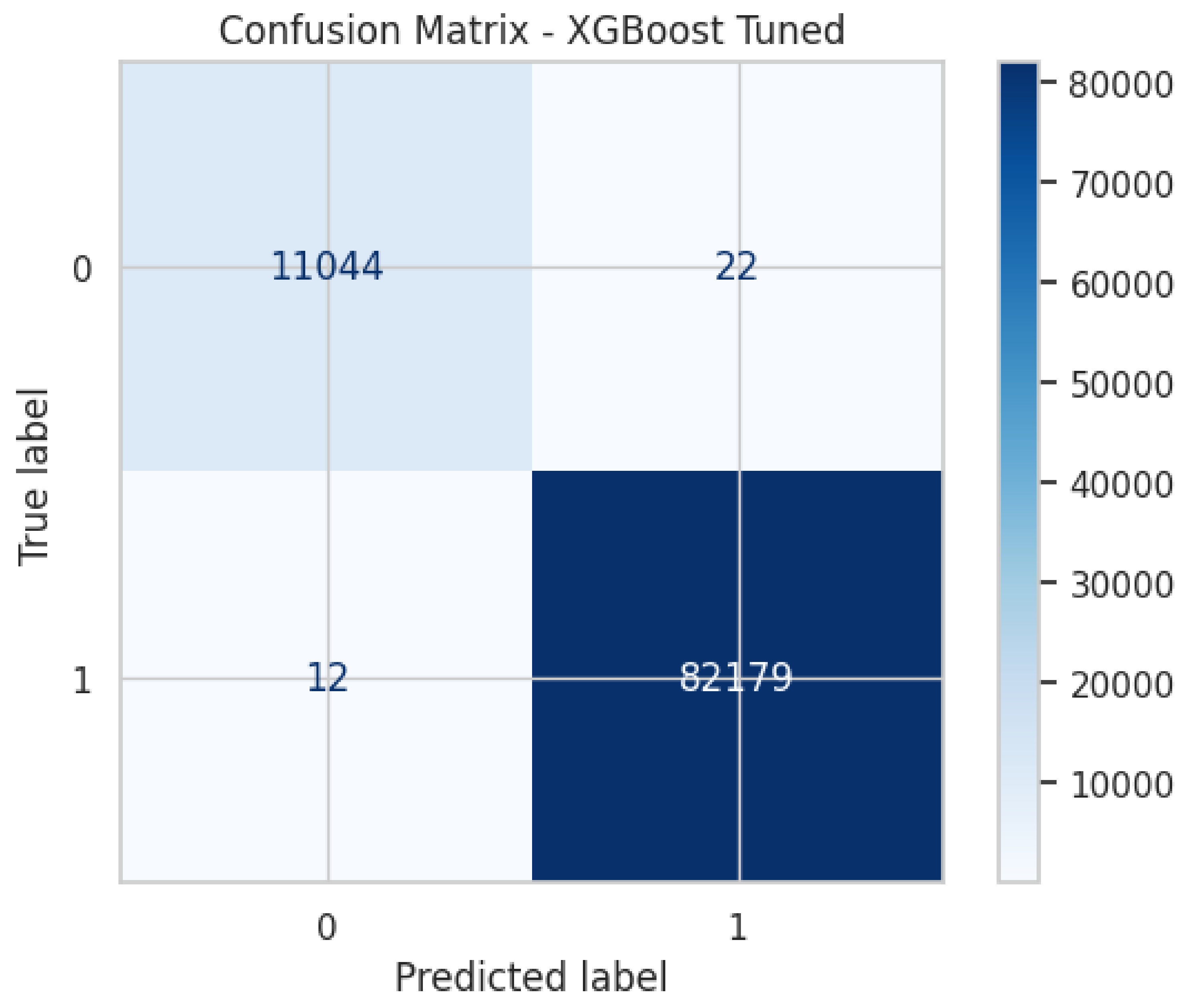
Feature Importance



📊 **Key Insights (Feature Importance Summary):**

1. **Most influential features**:
   - •recoveries and collection_recovery_fee are the **top contributors** to model prediction. This is logical in the context of credit risk:
     - •recoveries reflects how much money has been recovered from defaulted loans.
     - •collection_recovery_fee signals the cost incurred in debt collection, which can indicate higher default risk.
2. **Loan status features are also significant**:
   - •Features like loan_status_Fully Paid, loan_status_Current, and loan_status_Late (31-120 days) provide **strong historical insights** into a borrower's payment behavior, helping assess risk more accurately.
3. **Other notable features**:
   - •loan_age: Older loans may have a higher chance of encountering issues.
   - •last_pymnt_amnt: Indicates if the borrower is still actively making payments.
4. **Least important features**:
   - •Features such as verification_status, total_pymnt, etc., showed **very low importance**, suggesting minimal impact on model decisions.

Confusion Matrix



Confusion Matrix Summary (XGBoost Tuned)

The confusion matrix shows that the model performs **very well**:
✅ It correctly predicts **82,179** positive cases (*True Positives*) and **11,044** negative cases (*True Negatives*), meaning it accurately identifies both eligible and ineligible credit applicants.
❌ Only **22 False Positives** (misclassified ineligible applicants as eligible) and **12 False Negatives** (rejected applicants who were actually eligible) occurred.
This indicates the model is **highly precise and balanced**, minimizing both financial risk and lost opportunities.

4. Business Simulation

id/x partners

Business Simulation (1)

In this business simulation, we will evaluate the financial impact of processing credit applications based on the loan status data from the test set (**y_test**). By considering both **Good Loans** (loans that are successfully repaid) and **Bad Loans** (loans that default), we will assess how the company's revenue and losses are affected under certain assumptions. This simulation will help us understand the financial dynamics and potential risks of issuing loans based on the model's predictions.

- **Good Loans (status_loan = 1)**: 82,191 applicants
- **Bad Loans (status_loan = 0)**: 11,066 applicants
- **Total applicants**: 93,257 applicants

🎯 **Assumptions**:

- The processing cost for one credit application: **Rp10,000**
- If the loan is successfully paid (Good Loan): The company earns **Rp50,000**
- If the loan defaults (Bad Loan): The company loses **Rp100,000**

🚫 **Simulation Without Machine Learning (All Applications Processed)**

Everyone is processed, without filtering who is eligible or not.

- **Total cost**: 93,257 × Rp10,000 = Rp932,570,000
- **Profit from Good Loans**: 82,191 × Rp50,000 = Rp4,109,550,000
- **Loss from Bad Loans**: 11,066 × Rp100,000 = Rp1,106,600,000
- **Total profit**: = 4,109,550,000 − 1,106,600,000 − 932,570,000 = Rp2,070,380,000

Business Simulation (2)

☑ **Simulation with Machine Learning (XGBoost)**
Using the results from the previous confusion matrix:

	Predicted 0	Predicted 1
Actual 0 (Bad Loan)	11,044	22
Actual 1 (Good Loan)	12	82,179

This means the model will only approve loans for those predicted as 1, which are:

• **Total loans approved (Predicted 1)**: 82,179 + 22 = 82,201

• Among them:

o **Good Loans (TP)**: 82,179 → profit

o **Bad Loans (FP)**: 22 → loss

🧮 **Calculations**:

• **Total processing cost**: 82,201 × Rp10,000 = Rp822,010,000

• **Profit from Good Loans**: 82,179 × Rp50,000 = Rp4,108,950,000

• **Loss from Bad Loans**: 22 × Rp100,000 = Rp2,200,000

• **Total profit**: = 4,108,950,000 − 2,200,000 − 822,010,000

= Rp3,284,740,000

Business Simulation (3)

Comparison:

	Without ML	With ML (XGBoost)
Total Operating Cost	Rp932,570,000	Rp822,010,000
Profit from Good Loans	Rp4,109,550,000	Rp4,108,950,000
Loss from Bad Loans	Rp1,106,600,000	Rp2,200,000
Total Profit	Rp2,070,380,000	Rp3,284,740,000

Conclusion:

By using the XGBoost model, the company:

- Reduced the number of applicants with defaults from 11,066 to only 22 (false positives)
- Saved processing costs by not processing all applications
- Increased profit by:

Rp3,284,740,000 − Rp2,070,380,000 = Rp1,214,360,000

Conclusion

In the **credit risk analysis** project based on borrower data from 2007-2014, **Machine Learning (ML)**, specifically the **XGBoost model**, proved to be an effective data-driven solution for classifying creditworthiness 📊. After thorough **data cleaning**, handling **imbalances**, and evaluating multiple models, XGBoost was chosen for its **robustness** against outliers, its ability to process **large datasets**, and its excellent **generalization capabilities** 💪. Despite performing **hyperparameter tuning**, the model showed stable performance, achieving **99.96% accuracy**, with impressive metrics: **precision** (99.97%), **recall** (99.98%), and **F1-score** (99.97%) 🔥. Business simulations showed a **total profit of $460,855,000**, with minimal losses from **false positives** and **false negatives** 💵. Overall, the project highlighted the **tangible business value** of ML, showcasing the crucial role of **Data Scientists** in connecting **technical insights** and **business strategies** 🚀.

LINK FILES

THANK YOU!