Rakamin Academy

id/x partners

CREDIT RISK PREDICTION

ID/X Partners Data Scientist Project Based Internship

Presented By: Lukas Destria Putra Ginting

SELF INTRODUCTION

Saya Lukas Destria Putra Ginting, seorang fresh graduate dari Universitas Negeri Medan. Memiliki ketertarikan menjadi seorang data scientist. Saya merupakan peserta dalam Project Based Internship Rakamin yang bekerja sama dengan ID/X Partners terkait bidang Data Science.

lukasginting45@gmail.com

@lukas_gint

Lukas Ginting

COURSE & CERTIFICATION

<Fundamental Data Science> <July,2025>

<Intermediate Data Science> <August,2025>

<Asociate Data Scientist Certification> <September,2025>

Rakamin Academy

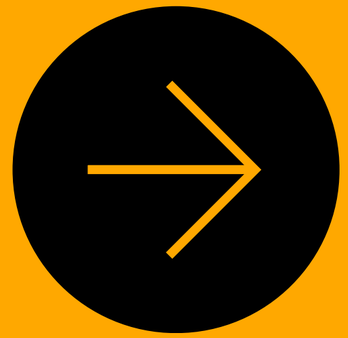
id/x partners

TABLE OF CONTENT

Rakamin Academy

id/x partners

Introduction



id/x partners

ABOUT ID/X PARTNERS

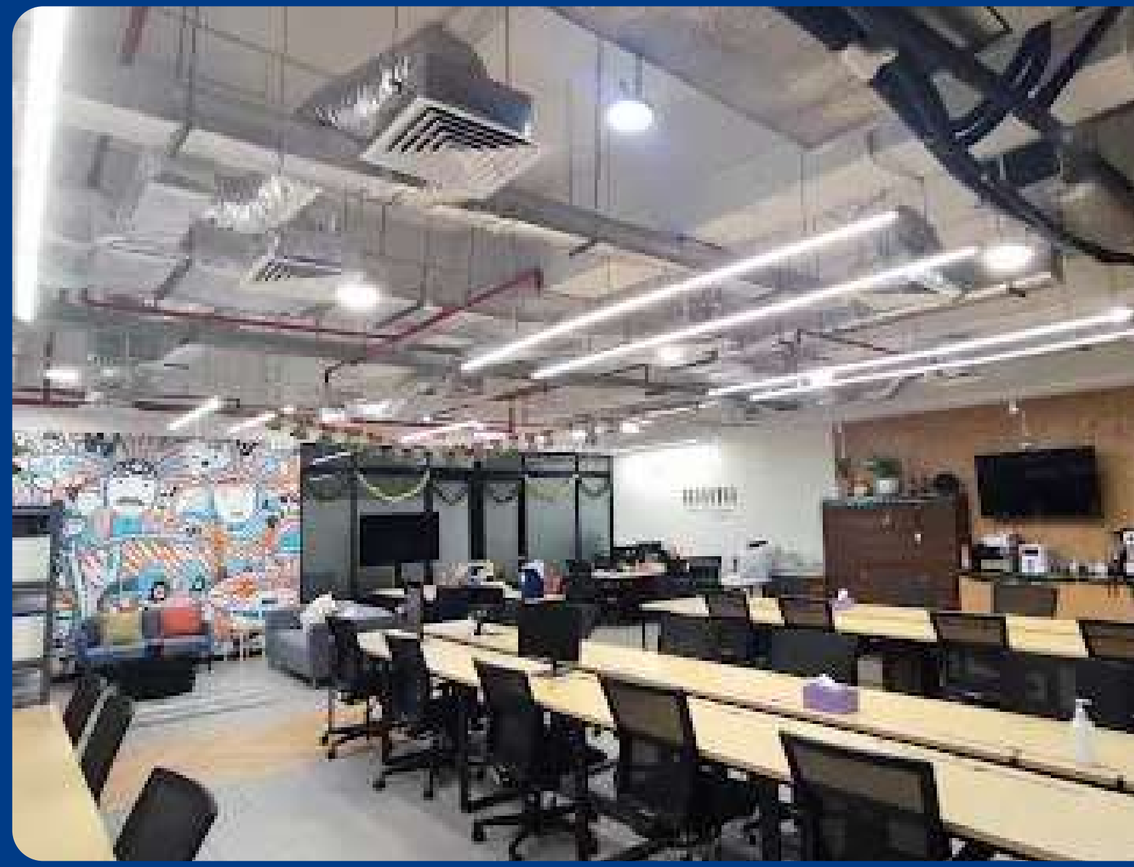
id/x partners didirikan pada tahun 2002 oleh mantan bankir dan konsultan manajemen yang memiliki pengalaman luas dalam manajemen siklus kredit dan proses, pengembangan skor, serta manajemen kinerja. Pengalaman gabungan kami telah melayani korporasi di wilayah Asia dan Australia serta berbagai industri, khususnya layanan keuangan, telekomunikasi, manufaktur, dan ritel.

id/x partners menyediakan layanan konsultasi yang berfokus pada pemanfaatan solusi analitik data dan pengambilan keputusan (DAD) yang diintegrasikan dengan disiplin manajemen risiko dan pemasaran untuk membantu klien mengoptimalkan keuntungan portofolio dan proses bisnis.

Layanan konsultasi komprehensif dan solusi teknologi yang ditawarkan oleh id/x partners menjadikannya sebagai penyedia layanan satu atap.

Rakamin Academy

id/x partners



CREDIT RISK

Risiko kredit (Credit Risk) adalah probabilitas kerugian finansial yang dihadapi oleh pemberi pinjaman akibat kegagalan pihak peminjam dalam melunasi kewajiban utangnya, baik pokok pinjaman maupun bunganya. Ini merupakan aspek krusial dalam industri keuangan karena berdampak langsung pada stabilitas dan profitabilitas perusahaan. Di satu sisi, kredit macet yang tidak terkendali dapat menyebabkan kerugian besar. Di sisi lain, penolakan pinjaman yang terlalu ketat terhadap peminjam yang sebenarnya layak justru akan menghambat pertumbuhan bisnis dan potensi keuntungan. Oleh karena itu, manajemen risiko kredit yang efektif sangat penting untuk mencapai keseimbangan yang tepat: meminimalkan kerugian dari pinjaman bermasalah sambil tetap memfasilitasi pertumbuhan ekonomi.

TOOLS & DOCUMENTATION

Rakamin Academy

id/x partners

TOOLS

DOCUMENTATION

CODE

VIDEO

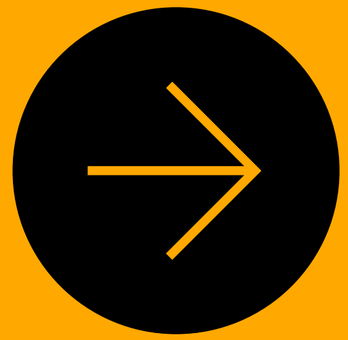
Untuk melihat python dan python notebook klik “HERE!”

www.reallygreatsite.com

Rakamin Academy

id/x partners

Business Understanding



BUSINESS UNDERSTANDING

Pemberian kredit yang tidak terkelola dengan baik dapat menimbulkan kerugian besar, mengancam stabilitas keuangan perusahaan. Oleh karena itu, mengelola risiko kredit menjadi sangat penting. Tantangannya adalah banyaknya faktor yang memengaruhi kelayakan seseorang atau entitas untuk menerima kredit, sehingga evaluasi menjadi lebih kompleks. Dengan demikian, penggunaan pendekatan berbasis data menjadi solusi untuk meningkatkan akurasi dalam mengidentifikasi dan menilai risiko, yang pada akhirnya akan menghasilkan keputusan pemberian kredit yang lebih efektif.

MASALAH

- Kesulitan Mengukur Risiko: Menilai seberapa besar risiko kredit dari seorang peminjam tidaklah mudah. Hal ini dipengaruhi oleh banyak faktor yang kompleks, seperti kondisi keuangan, rekam jejak, dan bahkan kualitas manajemen peminjam.
- Potensi Kerugian Finansial: Jika lembaga pemberi pinjaman salah dalam menilai risiko, mereka bisa memberikan pinjaman kepada pihak yang tidak mampu membayar. Ini dapat berujung pada kredit macet dan kerugian finansial yang signifikan bagi perusahaan.
- Kehilangan Peluang Bisnis: Di sisi lain, jika proses evaluasi terlalu ketat, perusahaan bisa menolak pemohon kredit yang sebenarnya layak. Akibatnya, perusahaan kehilangan potensi keuntungan dan kesempatan untuk mengembangkan bisnis.
- Proses Manual yang Tidak Efisien: Pengambilan keputusan yang masih mengandalkan proses manual dan data yang terbatas sering kali memakan waktu lama, tidak konsisten, dan kurang akurat dalam memprediksi risiko. Hal ini bisa menghambat kecepatan layanan dan menghambat pertumbuhan bisnis.

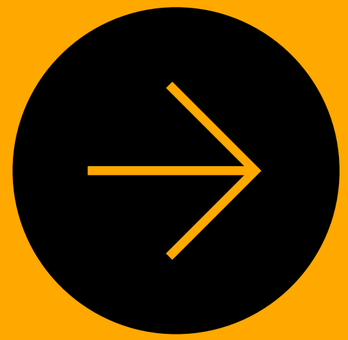
SOLUSI

- Penerapan Machine Learning: Menggunakan model machine learning untuk menganalisis data historis peminjam. Dengan begitu, sistem dapat memprediksi risiko kredit secara lebih akurat dan objektif, tidak hanya bergantung pada penilaian manual.
- Peningkatan Akurasi Evaluasi: Dengan analisis data yang mendalam, lembaga keuangan bisa mengidentifikasi pola-pola dan faktor-faktor risiko yang sebelumnya tidak terlihat. Hal ini membuat keputusan pemberian kredit menjadi lebih tepat dan terukur.
- Optimasi Pengambilan Keputusan: Hasil prediksi dari model machine learning dapat digunakan untuk menyusun strategi yang lebih efektif, seperti menentukan suku bunga yang sesuai, limit pinjaman, atau bahkan menolak permohonan yang berisiko tinggi. Ini membantu mengurangi kredit macet dan meningkatkan keuntungan.
- Otomatisasi dan Efisiensi Proses: Solusi berbasis data dapat mengotomatisasi sebagian besar proses evaluasi kredit. Hal ini tidak hanya mempercepat waktu persetujuan pinjaman, tetapi juga mengurangi bias manusia dan memastikan konsistensi dalam setiap keputusan yang diambil.

Rakamin Academy

id/x partners

Data Understanding



DATASET INFORMATION

DATA

1. Format: CSV
2. Nama File: loan_data_ 2007_2014.csv
3. Industri: Keuangan
4. Baris: 466285
5. Kolom: 75
6. Tipe data: Kategori (object) dan numerik (float & integer)
7. Float: 45 kolom
8. Object: 22 kolom
9. Integer: 7 kolom

VALIDASI

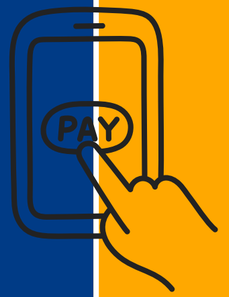
1. Total missing value sebanyak: 9776227
2. Terdapat 17 kolom dengan 100% baris berupa missing value
3. Terdapat 21 kolom dengan lebih dari 50% baris berupa missing value
4. Tidak terdapat duplikat data
5. Seluruh data konsisten, tetapi terdapat beberapa kolom yang hanya memiliki 1 nilai kategorial saja sehingga sebaiknya dihapus
6. Terdapat outlier pada banyak kolom

BEBERAPA HASIL ANALISA

LOAN AMOUNT

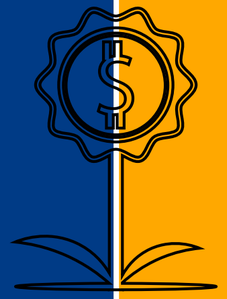
Total pinjaman yang dikeluarkan selama 2007 sampai dengan 2014 sebanyak $ 6.675.931.775 dengan rata-rata pemberian pinjaman sebesar $14.317

INSTALLMENT



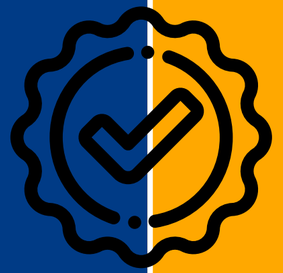
Pembayaran kredit bulanan nasabah rata-rata sebesar $432 dengan tertinggi $1409 dan terendah $15,67

INTEREST RATE



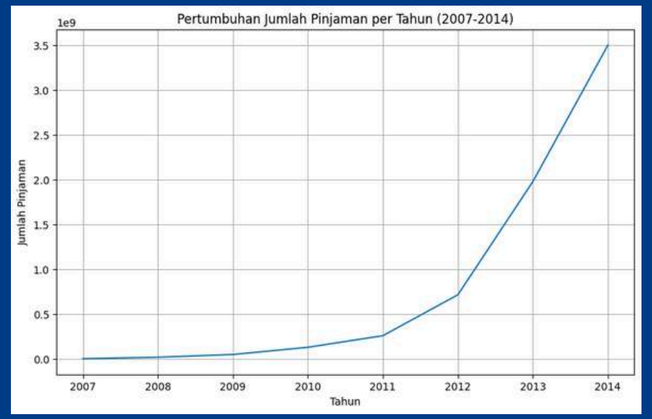
Bunga tertinggi 26%, bunga terendah 5,42%, dan rata-rata 13,82%

VERIFICATION STATUS



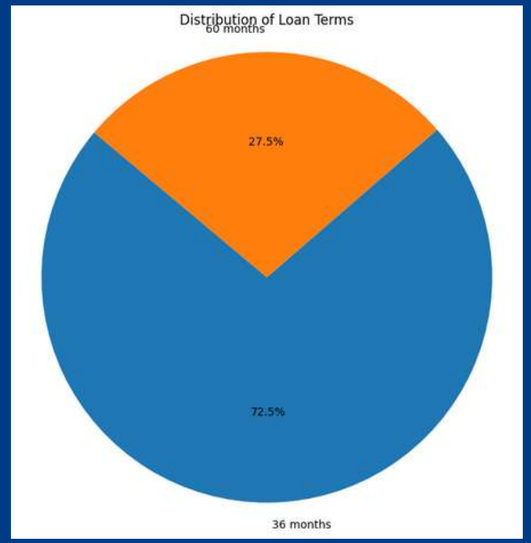
Terdapat 148237 data peminjam yang belum terverifikasi

VISUALISASI

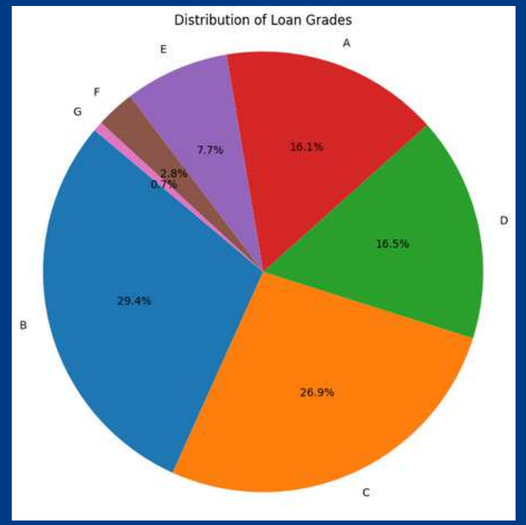


Dari grafik garis di kiri dapat diketahui bahwa mulai terjadi kenaikan yang tinggi mulai dari tahun 2012 ke 2013 dengan kenaikan tertinggi dari tahun 2013 ke 2014. Hal ini dapat mengindentifikasi pertumbuhan bisnis ataupun keberhasilan ekspansi.

Dari pie chart di kanan dapat diketahui bahwa kebanyakan orang yang mengajukan pinjaman mengambil jangka waktu pengembalian paling banyak dengan jangka waktu 36 bulan sebesar 72,5%. Hal ini mengindentifikasikan peminjaman yang dilakukan untuk kebutuhan jangka pendek.

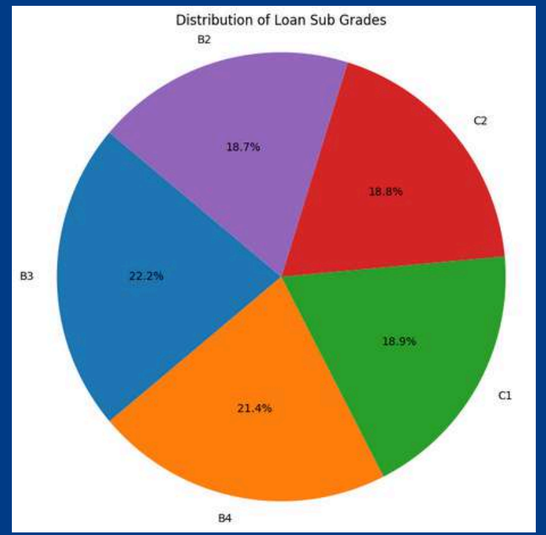


VISUALISASI

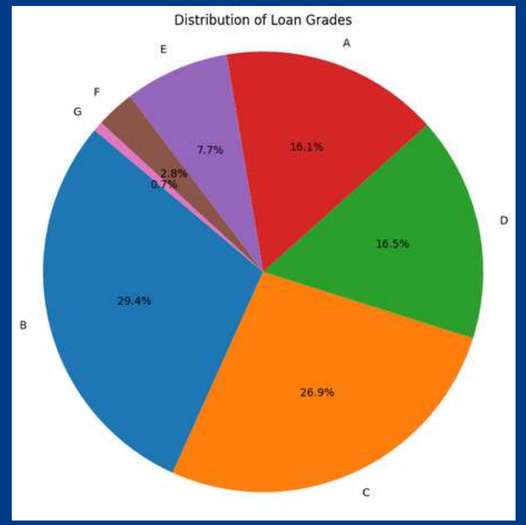


Dari pie chart di kiri dapat diketahui bahwa 3 top pilihan grade yang diambil oleh pangaju pinjaman adalah grade “B”, “C”, dan “D”. Diikuti dengan grade “A” yang hanya berbeda 0.4% dengan grade “D”. Sedangkan grade lainnya minim peminat

Dari pie chart di kanan dapat diketahui bahwa 5 TOP sub kategori yang dipilih seperti yang tampak pada chart. Meskipun grade “D” termasuk ke dalam 3 top grade yang diminati, tetapi secara sub grade belum masuk ke dalam 5 top sub grade yang dipilih peminjam.

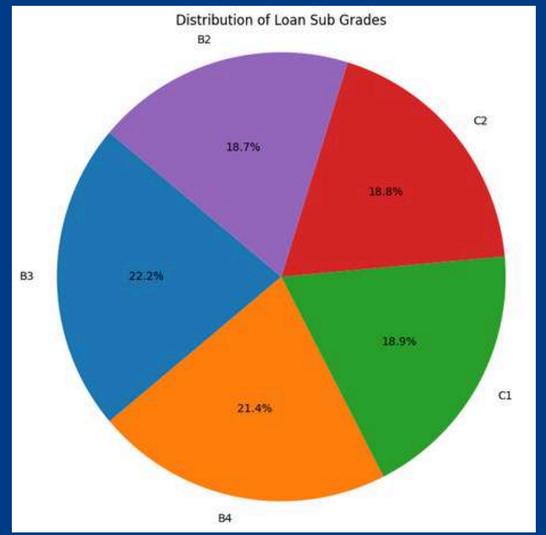


VISUALISASI



Dari pie chart di kiri dapat diketahui bahwa 3 top pilihan grade yang diambil oleh pangaju pinjaman adalah grade “B”, “C”, dan “D”. Diikuti dengan grade “A” yang hanya berbeda 0.4% dengan grade “D”. Sedangkan grade lainnya minim peminat

Dari pie chart di kanan dapat diketahui bahwa 5 TOP sub kategori yang dipilih seperti yang tampak pada chart. Meskipun grade “D” termasuk ke dalam 3 top grade yang diminati, tetapi secara sub grade belum masuk ke dalam 5 top sub grade yang dipilih peminjam.



VISUALISASI

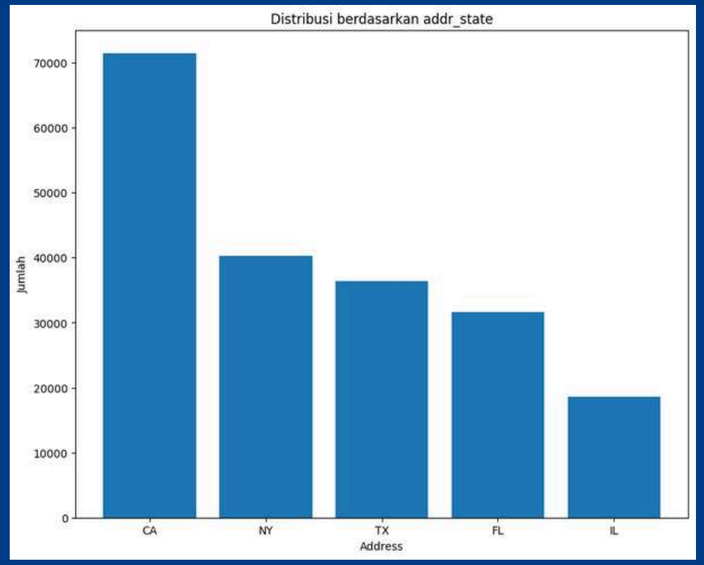
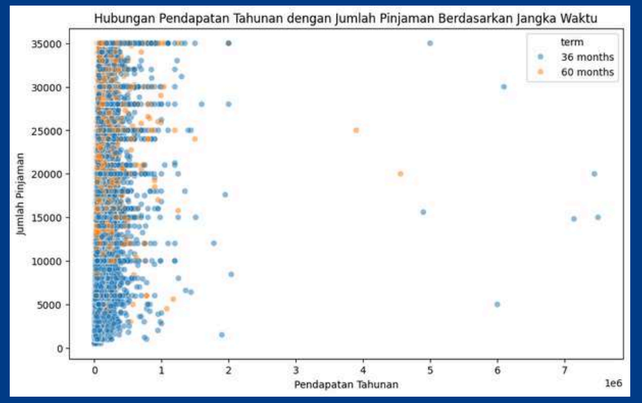


Diagram ini menampilkan distribusi data pinjaman berdasarkan negara bagian. California (CA) memiliki jumlah data terbanyak, di atas 70.000, diikuti oleh New York (NY). Sementara itu, Texas (TX), Florida (FL), dan Illinois (IL) berada di posisi berikutnya. Secara singkat, grafik ini menunjukkan bahwa sebagian besar data pinjaman terkonsentrasi di California dan New York.

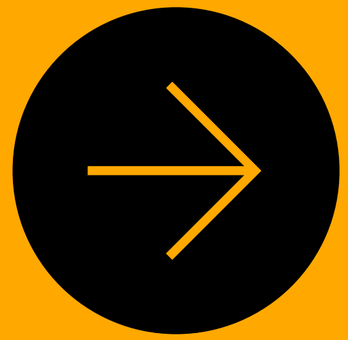
Berdasarkan diagram tersebut, terlihat bahwa mayoritas peminjam memiliki pendapatan rendah dan mengambil pinjaman dengan jumlah yang tidak terlalu besar. Pinjaman dengan jangka waktu 36 bulan lebih umum diberikan daripada yang 60 bulan. Namun, ada juga kasus di mana individu berpendapatan sangat tinggi mengambil pinjaman dengan jumlah dan jangka waktu yang bervariasi.



Rakamin Academy

id/x partners

Data Preparation



Rakamin Academy

id/x partners

MEMBERSIHKAN OUTLIER

```
''' Membersihkan kolom data yang mengandung outlier '''
for col_used in df.columns[:-1]:
  if pd.api.types.is_numeric_dtype(df[col_used]):
    Q1 = df[col_used].quantile(0.25)
    Q3 = df[col_used].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df[col_used] = df[col_used].clip(lower_bound, upper_bound)
```

MEMBUANG DATA DENGAN MEMILIH KOLOM YANG INGIN DIPERTAHANKAN

```python
# Membuang data dengan memilih kolom yang ingin dipertahankan
columns_to_remove = ['desc', 'mths_since_last_delinq', 'mths_since_last_record',
                     'mths_since_last_major_derog', 'annual_inc_joint', 'dti_joint',
                     'verification_status_joint', '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', 'inq_fi',
                     'total_cu_tl', 'inq_last_12m', 'next_pymnt_d','Column1', 'id', 'member_id', 'emp_title',
                     'url', 'title', 'policy_code', 'application_type', 'Unnamed: 0'
                     , 'zip_code']

# Dapatkan semua nama kolom
all_columns = df.columns

# Dapatkan nama kolom yang ingin dipertahankan (semua kolom dikurangi kolom yang akan dihapus)
columns_to_keep = all_columns.difference(columns_to_remove)

# Buat DataFrame baru dengan hanya kolom yang dipertahankan
df = df[columns_to_keep]

print('\nInformasi DataFrame setelah menghapus kolom:')
print(df.info())
print('\nDimensi DataFrame setelah menghapus kolom:')
print(df.shape)
#=====================================================================
```

```
Dimensi DataFrame setelah menghapus kolom:
(466285, 44)
```

MENTRANSFORMASIKAN KOLOM WAKTU MENJADI DATETIME

```
# Mentransformasikan data objek menjadi waktu
datetime_col = ['issue_d', 'earliest_cr_line', 'last_pymnt_d', 'last_credit_pull_d']
# Mengecek apakah date time masih berada di kolom df
datetime_col_present = [col for col in datetime_col if col in df.columns]
if datetime_col_present:
  print(df[datetime_col_present].info())
  print(df[datetime_col_present])
  for col in datetime_col_present:
      df[col] = pd.to_datetime(df[col], format='%b-%y', errors='coerce')
else:
  print("\nDatetime columns were already removed.")
```

MENENTUKAN LOAN STATUS SEBAGAI LABEL ATAU TARGET DENGAN MAPPING NILAI DAN MENGGANTI NAMANYA MENJADI KOLOM GOOD_BAD_STATUS

```python
# Melakukan encoding loan_status untuk menjadi kolom good_bad_status
# Encoding loan_status
loan_status_mapping = {
    'Fully Paid': 1,
    'Charged Off': 0,
    'Current': 1,
    'Default': 0,
    'Late (31-120 days)': 0,
    'In Grace Period': 1,
    'Late (16-30 days)': 0,
    'Does not meet the credit policy. Status:Fully Paid': 1,
    'Does not meet the credit policy. Status:Charged Off': 0
}

# Menambahkan kolom baru 'good_bad_status' berdasarkan mapping
df['good_bad_status'] = df['loan_status'].map(loan_status_mapping)

# Menampilkan hasil
print('\nBerikut value terbaru dari kolom good_bad_status:', df['good_bad_status'])
print('\nBerikut jumlah dari setiap value kolom good_bad_status', df['good_bad_status'].value_counts(), '\n')


# Menghapus kolom loan_status karena sudah diganti menjadi good_bad_status
df = df.drop('loan_status', axis=1)
```

```
Berikut value terbaru dari kolom good_bad_status: 0         1
1         0
2         1
3         1
4         1
         ..
466280    1
466281    0
466282    1
466283    1
466284    1
Name: good_bad_status, Length: 466285, dtype: int64

Berikut jumlah dari setiap value kolom good_bad_status good_bad_status
1    414099
0     52186
Name: count, dtype: int64
```

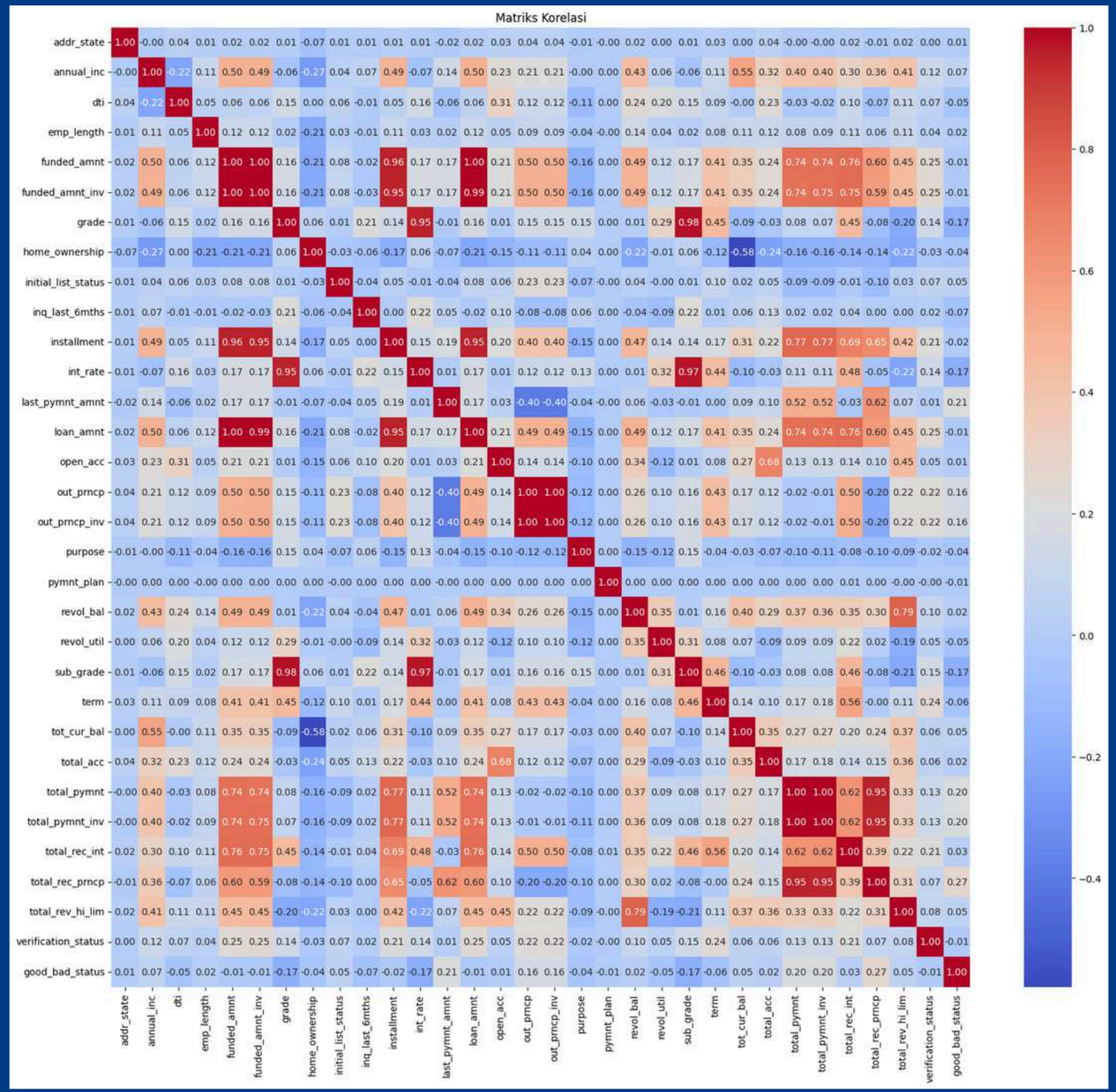
MELAKUKAN LABEL ENCODING

```python
# Melakukan label encoding
# Kolom yang ingin di-label encode

Label_Encoding_kolom = ['home_ownership', 'purpose', 'addr_state', 'initial_list_status']

# Dictionary untuk menyimpan encoder tiap kolom
encoders = {}

for col in Label_Encoding_kolom:
    le = preprocessing.LabelEncoder()
    df[col] = le.fit_transform(df[col])
    encoders[col] = le

# Menampilkan hasil label encoding
print('\nIni adalah hasil label encoding\n', df)
# ============================================================
```

MELAKUKAN ORDINAL ENCODING

```python
# Melakukan ordinal encoding
# Definisikan urutan nilai untuk setiap kolom ordinal
ordinal_mapping = {
    'term': {' 36 months': 0, ' 60 months': 1},
    'grade': {'A':0, 'B':1, 'C':2, 'D':3, 'E':4, 'F':5, 'G':6},
    'sub_grade': {
        'A1':0,'A2':1,'A3':2,'A4':3,'A5':4,
        'B1':5,'B2':6,'B3':7,'B4':8,'B5':9,
        'C1':10,'C2':11,'C3':12,'C4':13,'C5':14,
        'D1':15,'D2':16,'D3':17,'D4':18,'D5':19,
        'E1':20,'E2':21,'E3':22,'E4':23,'E5':24,
        'F1':25,'F2':26,'F3':27,'F4':28,'F5':29,
        'G1':30,'G2':31,'G3':32,'G4':33,'G5':34
    },
    'emp_length': {
        '< 1 year':0,'1 year':1,'2 years':2,'3 years':3,'4 years':4,
        '5 years':5,'6 years':6,'7 years':7,'8 years':8,'9 years':9,'10+ years':10
    },
    'verification_status': {'Not Verified':0, 'Verified':1, 'Source Verified':2},
    'pymnt_plan': {'n':0, 'y':1}
}

# Kolom yang ingin di-ordinal encode
Ordinal_Encoding_kolom = list(ordinal_mapping.keys())

# Terapkan mapping ke DataFrame
for col in Ordinal_Encoding_kolom:
    df[col] = df[col].map(ordinal_mapping[col])

# Menampilkan hasil ordinal encoding
print('\nIni adalah hasil ordinal encoding\n', df)
# ================================================
```

MELIHAT KORELASI



Pada dasarnya, matriks korelasi menunjukkan beberapa hubungan yang sangat kuat dan logis antar variabel. Hubungan ini sering kali bukan disebabkan oleh kausalitas, melainkan karena variabel-variabel tersebut memang memiliki definisi yang erat atau merupakan bagian dari satu kesatuan. Contohnya, loan_amnt (jumlah pinjaman yang diminta) dan funded_amnt (jumlah dana yang disetujui) memiliki korelasi yang nyaris sempurna karena keduanya pada dasarnya merepresentasikan nilai yang sama. Demikian pula, total_pymnt (total pembayaran) memiliki korelasi kuat dengan total_rec_prncp (total pokok yang diterima) dan total_rec_int (total bunga yang diterima), yang mana secara matematis total pembayaran adalah jumlah dari keduanya. Hubungan-hubungan ini menunjukkan konsistensi data, bukan suatu penemuan yang memerlukan analisis mendalam.

Di sisi lain, sebagian besar variabel dalam matriks korelasi menunjukkan hubungan yang lemah atau tidak signifikan, yang berarti tidak ada hubungan linear yang jelas antar variabel-variabel tersebut. Misalnya, variabel demografi seperti addr_state (negara bagian) dan emp_length (lama bekerja) tidak menunjukkan korelasi yang kuat dengan variabel finansial lainnya seperti loan_amnt atau annual_inc (pendapatan tahunan). Hal ini mengindikasikan bahwa lokasi geografis atau durasi kerja seseorang tidak secara langsung menentukan seberapa besar pinjaman yang mereka ambil atau berapa pendapatan mereka, setidaknya dalam hubungan linear. Meskipun korelasi lemah, tidak berarti variabel-variabel ini tidak penting. Mereka mungkin memiliki pengaruh non-linear yang kompleks terhadap risiko kredit dan bisa menjadi prediktor penting dalam model yang lebih canggih.

MELIHAT TIPE DATA TERBARU

```
Berikut Info Tipe Data terbaru dari dataframe:

acc_now_delinq                float64
addr_state                      int64
annual_inc                    float64
collection_recovery_fee       float64
collections_12_mths_ex_med    float64
delinq_2yrs                   float64
dti                           float64
earliest_cr_line       datetime64[ns]
emp_length                    float64
funded_amnt                     int64
funded_amnt_inv               float64
grade                           int64
home_ownership                  int64
initial_list_status             int64
inq_last_6mths                float64
installment                   float64
int_rate                      float64
issue_d                datetime64[ns]
last_credit_pull_d     datetime64[ns]
last_pymnt_amnt               float64
last_pymnt_d           datetime64[ns]
loan_amnt                       int64
open_acc                      float64
out_prncp                     float64
out_prncp_inv                 float64
pub_rec                       float64
purpose                         int64
pymnt_plan                      int64
recoveries                    float64
revol_bal                       int64
revol_util                    float64
sub_grade                       int64
term                            int64
tot_coll_amt                  float64
tot_cur_bal                   float64
total_acc                     float64
total_pymnt                   float64
total_pymnt_inv               float64
total_rec_int                 float64
total_rec_late_fee            float64
total_rec_prncp               float64
total_rev_hi_lim              float64
verification_status             int64
good_bad_status                 int64
dtype: object
```

```
dtype: object
float64            27
int64              13
datetime64[ns]      4
Name: count, dtype: int64
```

HANDLING MISSING VALUE

```
# Membuang seluruh baris yang mengandung missing value
df.dropna(inplace = True) # inplace = True menetapkan secara permanent data frame baru dengan kondisi sudah membuang missing value (karena diguna

# Data Frame setelah handling missing value
print('\nCek ulang data terbaru setelah dropna')
print(df.isna().sum())
print('\nCek ulang jumlah missing value data terbaru sebanyak:', df.isna().sum().sum())
print('Dimensi ulang data:', df.shape)
```

```
Cek ulang jumlah missing value data terbaru sebanyak: 0
Dimensi ulang data: (376571, 44)
```

NORMALISASI DATA

```
   acc_now_delinq  addr_state  annual_inc  collection_recovery_fee  \
0             0.0    0.448980    0.337972                      0.0
1             0.0    0.081633    0.145792                      0.0
2             0.0    0.122449    0.669317                      0.0
3             0.0    0.632653    0.238569                      0.0
4             0.0    0.183673    0.390987                      0.0

   collections_12_mths_ex_med  delinq_2yrs       dti  emp_length  funded_amnt  \
0                         0.0          0.0  0.573039         1.0     0.766176
1                         0.0          0.0  0.629416         0.1     0.257353
2                         0.0          0.0  0.352042         1.0     0.323529
3                         0.0          0.0  0.424455         1.0     0.323529
4                         0.0          0.0  0.413681         0.2     0.411765

   funded_amnt_inv  ...  tot_cur_bal  total_acc  total_pymnt  total_pymnt_inv  \
0         0.766520  ...     0.239509   0.476190     0.709389         0.712075
1         0.258443  ...     0.029456   0.495238     0.266134         0.267142
2         0.324523  ...     0.558229   0.380952     0.294488         0.295603
3         0.324523  ...     0.028376   0.571429     0.445524         0.447211
4         0.412628  ...     0.568336   0.514286     0.381038         0.382480

   total_rec_int  total_rec_late_fee  total_rec_prncp  total_rev_hi_lim  \
0       0.618270                 0.0         0.696935          0.803763
1       0.286546                 0.0         0.247125          0.201613
2       0.162543                 0.0         0.316460          0.819892
3       0.199223                 0.0         0.491055          0.107527
4       0.275510                 0.0         0.390828          0.205645

   verification_status  good_bad_status
0                  0.5              1.0
1                  0.0              1.0
2                  0.0              1.0
3                  1.0              1.0
4                  0.0              1.0

[5 rows x 40 columns]
```

```
Grouped data for column 'collections_12_mths_ex_med':

collections_12_mths_ex_med
0.0    376571
Name: collections_12_mths_ex_med, dtype: int64
------------------------------

Grouped data for column 'delinq_2yrs':

delinq_2yrs
0.0    376571
Name: delinq_2yrs, dtype: int64
------------------------------
```

```
Grouped data for column 'recoveries':

recoveries
0.0    376571
Name: recoveries, dtype: int64
------------------------------
```

```
Grouped data for column 'pub_rec':

pub_rec
0.0    376571
Name: pub_rec, dtype: int64
------------------------------
```

CEK ULANG DATA

Terdapat kolom yang memiliki variansi hanya 0 atau kosong karena proses handling value: ['acc_now_delinq', 'collection_recovery_fee', 'collections_12_mths_ex_med', 'delinq_2yrs', 'pub_rec', 'recoveries', 'tot_coll_amt', 'total_rec_late_fee']

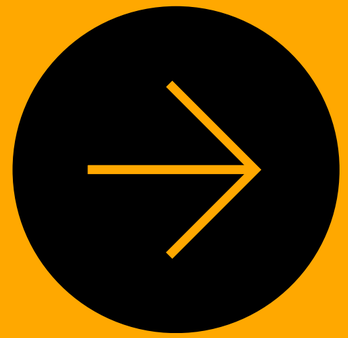
HAPUS KOLOM DENGAN VARIANSI 0 ATAU KOSONG

```
Dimensi DataFrame setelah menghapus kolom dengan varians nol:
(376571, 32)
```

Rakamin Academy

id/x partners

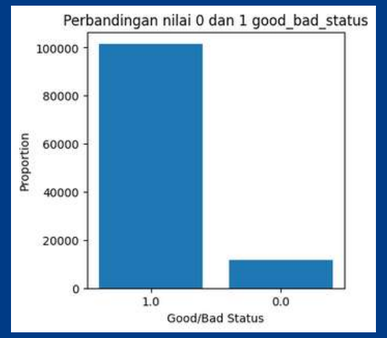
Modelling



PROSES PENENTUAN

Menentukan variabel X dan y

```
# Pisahkan data menjadi nilai X dan y
X = df.drop('good_bad_status', axis=1)
y = df['good_bad_status']
```



Status
0: Bad
1: Good

Metode Sampling

```
# Stratified Sampling
X_sample, _, y_sample, _= train_test_split(
    X, y,
    train_size=0.3,       # Jumlah persen data yang diambil
    stratify=y,           # stratified berdasarkan distribusi target
    random_state=42
)
print("Jumlah data asli :", len(y))
print("Jumlah data sampel:", len(y_sample))
print("Distribusi target sampel:")
print(y_sample.value_counts(normalize=True))
print(y_sample.value_counts())
```

```
Jumlah data asli : 376571
Jumlah data sampel: 112971
Distribusi target sampel:
good_bad_status
1.0    0.897611
0.0    0.102389
Name: proportion, dtype: float64
good_bad_status
1.0    101404
0.0     11567
```

METODE EVALUASI SPLIT PERCENTAGE

```
# Menggunakan sampel data yang sudah distratifikasi untuk pembagian data latih dan uji
X_train, X_test, y_train, y_test = train_test_split(
    X_sample[selected_features], y_sample, test_size=0.2, random_state=42
)

print('\nDimensi X latih:', X_train.shape)
print('Dimensi X uji:', X_test.shape)
print('Dimensi y latih:', y_train.shape)
print('Dimensi y uji:', y_test.shape)
print('\nSebelum oversampling:', pd.Series(y_train).value_counts())
```

```
Dimensi X latih: (90376, 31)
Dimensi X uji: (22595, 31)
Dimensi y latih: (90376,)
Dimensi y uji: (22595,)
```

Rakamin Academy

id/x partners

MENGATASI IMBALANCED DATA

```python
# Oversampling dengan SMOTE
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state=42)
X_train_over, y_train_over = smote.fit_resample(X_train, y_train)
print("Setelah oversampling:", pd.Series(y_train_over).value_counts())
```

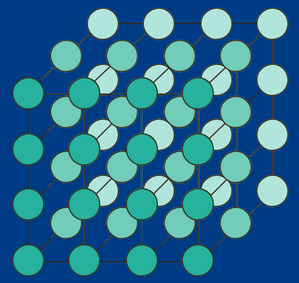
```
Sebelum oversampling: good_bad_status
1.0    81084
0.0     9292
Name: count, dtype: int64
```

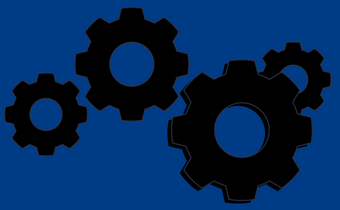
```
Setelah oversampling: good_bad_status
1.0    81084
0.0    81084
Name: count, dtype: int64
```

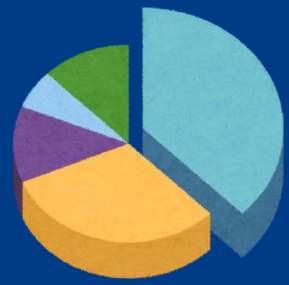
Rakamin Academy

id/x partners

MODEL ALGORITMA YANG DIGUNAKAN







LOGISTIK REGRESI

XGBOOST

LOGISTIK REGRESI

```
Hasil Evaluasi pada Data Latih:
- Akurasi: 0.902508509693651
- ROC-AUC Score: 0.958932678232835

Hasil Eksekusi pada Data Uji:
- Algoritma yang digunakan: Logistic Regression
- Fitur yang digunakan: Index(['addr_state', 'annual_inc', 'dti', 'emp_length', 'funded_amnt',
       'funded_amnt_inv', 'grade', 'home_ownership', 'initial_list_status',
       'inq_last_6mths', 'installment', 'int_rate', 'last_pymnt_amnt',
       'loan_amnt', 'open_acc', 'out_prncp', 'out_prncp_inv', 'purpose',
       'pymnt_plan', 'revol_bal', 'revol_util', 'sub_grade', 'term',
       'tot_cur_bal', 'total_acc', 'total_pymnt', 'total_pymnt_inv',
       'total_rec_int', 'total_rec_prncp', 'total_rev_hi_lim',
       'verification_status'],
      dtype='object')
- Akurasi model terbaik: 0.9452091170612967
- ROC-AUC Score terbaik: 0.9504858527299472
- Laporan Klasifikasi model terbaik:
              precision    recall  f1-score   support

         0.0       0.69      0.82      0.75      2275
         1.0       0.98      0.96      0.97     20320

    accuracy                           0.95     22595
   macro avg       0.84      0.89      0.86     22595
weighted avg       0.95      0.95      0.95     22595
```

Berdasarkan hasil evaluasi model regresi logistik ini, model menunjukkan performa yang sangat baik. Akurasi pada data uji mencapai 94.52%, dan skor AUC-ROC yang tinggi di angka 0.95. Angka-angka ini menunjukkan bahwa model sangat andal dalam membedakan antara pinjaman yang baik dan yang buruk. Laporan klasifikasi lebih lanjut memperkuat hasil ini, menunjukkan presisi dan recall yang tinggi untuk kedua kelas, terutama untuk kelas 1.0 (pinjaman baik) dengan presisi 0.98 dan recall 0.96. Ini berarti model mampu mengidentifikasi pinjaman yang akan berhasil dengan sangat akurat, sementara juga menjaga kinerja yang baik untuk pinjaman yang berisiko.

XGBOOST

```
Hasil Evaluasi pada Data Latih:
- Akurasi: 0.9893937151595876
- ROC-AUC Score: 0.9992536652019615

Hasil Eksekusi pada Data Uji:
- Algoritma yang digunakan: XGBoost
- Fitur yang digunakan: Index(['addr_state', 'annual_inc', 'dti', 'emp_length', 'funded_amnt',
       'funded_amnt_inv', 'grade', 'home_ownership', 'initial_list_status',
       'inq_last_6mths', 'installment', 'int_rate', 'last_pymnt_amnt',
       'loan_amnt', 'open_acc', 'out_prncp', 'out_prncp_inv', 'purpose',
       'pymnt_plan', 'revol_bal', 'revol_util', 'sub_grade', 'term',
       'tot_cur_bal', 'total_acc', 'total_pymnt', 'total_pymnt_inv',
       'total_rec_int', 'total_rec_prncp', 'total_rev_hi_lim',
       'verification_status'],
      dtype='object')
- Akurasi model terbaik: 0.9799955742420889
- ROC-AUC Score terbaik: 0.9620811196677339
- Laporan Klasifikasi model terbaik:
              precision    recall  f1-score   support

         0.0       0.99      0.81      0.89      2275
         1.0       0.98      1.00      0.99     20320

    accuracy                           0.98     22595
   macro avg       0.99      0.90      0.94     22595
weighted avg       0.98      0.98      0.98     22595
```

Berdasarkan hasil evaluasi, model XGBoost menunjukkan performa yang sangat luar biasa, bahkan melebihi model regresi logistik sebelumnya. Model ini mencapai akurasi yang sangat tinggi, yaitu 97.99% pada data uji. Skor ROC-AUC juga sangat impresif, yaitu 0.96, yang menunjukkan kemampuan model yang luar biasa dalam membedakan kelas pinjaman yang baik dan buruk. Dalam laporan klasifikasi, model ini memiliki presisi sebesar 0.98 dan recall 1.00 untuk kelas 1.0 (pinjaman baik), yang berarti model hampir sempurna dalam mengidentifikasi semua kasus pinjaman baik. Meskipun recall untuk kelas 0.0 (pinjaman buruk) sedikit lebih rendah, model tetap memiliki kinerja yang sangat kuat secara keseluruhan, menjadikannya pilihan yang sangat efektif untuk memprediksi risiko kredit.

LR GRID SEARCH

```
Hasil Evaluasi Model Terbaik (Grid Search LR) pada Data Latih:
- Akurasi: 0.9024160130235311
- ROC-AUC Score: 0.9592593463315306

Hasil Grid Search - Logistic Regression:
- Parameter terbaik: {'C': 100, 'penalty': 'l1', 'solver': 'saga'}
- ROC-AUC terbaik pada cross-validation: 0.959174094338913

Hasil Evaluasi Model Terbaik pada Data Testing:
- Akurasi: 0.9435273290551007
- ROC-AUC Score: 0.9503750757116901
- Laporan Klasifikasi:
              precision    recall  f1-score   support

         0.0       0.68      0.82      0.75      2275
         1.0       0.98      0.96      0.97     20320

    accuracy                           0.94     22595
   macro avg       0.83      0.89      0.86     22595
weighted avg       0.95      0.94      0.95     22595
```

Hasil evaluasi model regresi logistik dengan Grid Search menunjukkan performa yang sangat baik dan konsisten. Model terbaik yang ditemukan oleh Grid Search menggunakan parameter 'C': 100, 'penalty': 'l1', dan 'solver': 'saga'. Model ini mencapai akurasi sebesar 94.35% dan skor ROC-AUC sebesar 0.95 pada data pengujian. Performa ini sangat mirip dengan model tanpa hyperparameter tuning, menandakan bahwa model regresi logistik sudah cukup optimal dari awal. Laporan klasifikasi menunjukkan bahwa model memiliki kemampuan prediksi yang luar biasa, terutama dalam mengidentifikasi kelas 1.0 (pinjaman baik) dengan presisi 0.98 dan recall 0.96. Ini menunjukkan bahwa Grid Search berhasil menyempurnakan model tanpa mengubah performa secara drastis

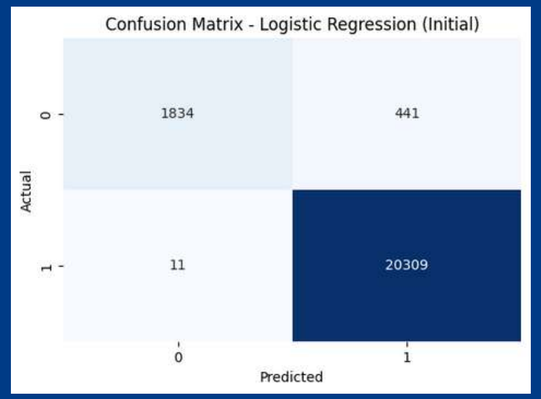
XGB GRID SEARCH

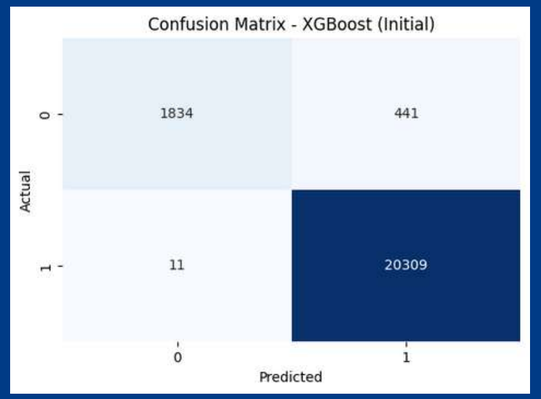
```
Hasil Evaluasi Model Terbaik (Grid Search XGBoost) pada Data Latih:
- Akurasi: 0.9990442010754279
- ROC-AUC Score: 0.9999998492687416

Hasil Grid Search - XGBoost:
- Parameter terbaik: {'colsample_bytree': 1.0, 'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 300, 'subsample': 0.8}
- ROC-AUC terbaik pada cross-validation: 0.9963471068198766

Hasil Evaluasi Model Terbaik pada Data Testing:
- Akurasi: 0.9806151803496349
- ROC-AUC Score: 0.9642993423898935
- Laporan Klasifikasi:
              precision    recall  f1-score   support

         0.0       1.00      0.81      0.89      2275
         1.0       0.98      1.00      0.99     20320

    accuracy                           0.98     22595
   macro avg       0.99      0.90      0.94     22595
weighted avg       0.98      0.98      0.98     22595
```

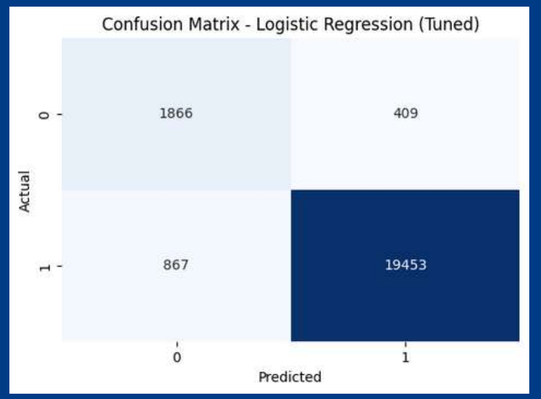
Hasil evaluasi model XGBoost setelah Grid Search menunjukkan performa yang luar biasa dan sedikit lebih baik dari model sebelumnya. Model terbaik ini mencapai akurasi 98.06% pada data pengujian, yang merupakan angka sangat tinggi. Skor ROC-AUC juga sangat impresif, yaitu 0.96, yang menegaskan kemampuan model dalam membedakan antara pinjaman baik dan buruk dengan presisi tinggi. Laporan klasifikasi menunjukkan bahwa model ini hampir sempurna dalam mengidentifikasi pinjaman yang baik (kelas 1.0) dengan presisi 0.98 dan recall 1.00. Meskipun kinerja untuk kelas 0.0 (pinjaman buruk) sedikit lebih rendah, model tetap sangat kuat secara keseluruhan, menunjukkan bahwa hyperparameter tuning berhasil meningkatkan akurasi model dan menjadikannya alat prediksi yang sangat efektif.

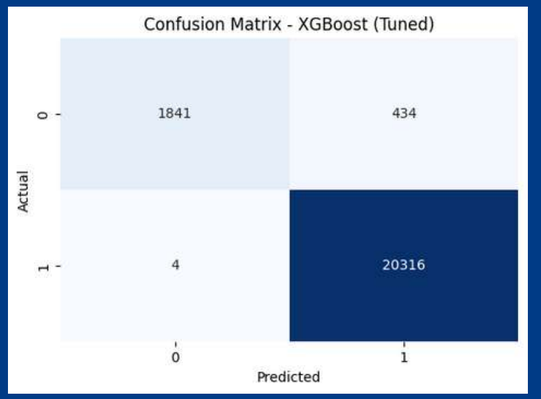
CONFUSION MATRIX

TUNED = GRID SEARCH
INITIAL = TANPA HYPERPARAMETER TUNING









Hasil matriks konfusi menunjukkan perbandingan kinerja antara model Regresi Logistik dan XGBoost, baik sebelum maupun sesudah hyperparameter tuning. Secara umum, keempat model memiliki performa yang sangat baik dalam memprediksi pinjaman "baik" (kelas 1.0), dengan akurasi yang tinggi. Namun, model XGBoost Tuned menunjukkan kinerja terbaik, terutama dalam memprediksi pinjaman "buruk" (kelas 0.0), yang ditunjukkan oleh jumlah False Negatives (prediksi 1 padahal aktualnya 0) yang sangat rendah, yaitu 4. Angka ini jauh lebih baik dibandingkan dengan Regresi Logistik Tuned yang memiliki 867 False Negatives. Hal ini sangat krusial dalam mitigasi risiko kredit, karena kesalahan memprediksi pinjaman buruk sebagai pinjaman baik dapat menyebabkan kerugian finansial yang signifikan bagi perusahaan. Meskipun semua model akurat secara keseluruhan, model XGBoost Tuned adalah yang paling efektif dalam mengidentifikasi risiko kredit secara tepat.

CONCLUTION

Secara keseluruhan, analisis data menunjukkan bahwa manajemen risiko kredit yang efektif dapat dicapai melalui pendekatan berbasis data. Distribusi data pinjaman yang beragam menggarisbawahi pentingnya memilih fitur prediktif yang valid dan menghindari variabel yang datanya baru tersedia setelah pinjaman selesai. Hal ini krusial agar model dapat digunakan secara praktis untuk memprediksi risiko pinjaman baru.

Penerapan data science memberikan solusi prediktif yang kuat, dengan model XGBoost terbukti sangat efektif dalam mengidentifikasi risiko kredit. Berdasarkan temuan ini, perusahaan disarankan untuk menerapkan sistem peringatan dini (Early Warning System) untuk mendeteksi peminjam berisiko, mengadopsi penentuan harga berbasis risiko (risk-based pricing) untuk menyesuaikan suku bunga, dan menerapkan manajemen portofolio dinamis untuk mengoptimalkan pengembalian. Langkah-langkah strategis ini tidak hanya memitigasi risiko, tetapi juga meningkatkan profitabilitas bisnis secara keseluruhan.

Rakamin Academy

id/x partners

THANK YOU