Rakamin Academy X id/x partners

Internship Final Project:

Develop Credit Risk Prediction Model using Machine Learning

at ID/X Partners - Data Scientist

Presented by
Raon Spielberg Berek

Rakamin Academy X id/x partners

Kota Bandung, Jawa Barat

raonbereknew@gmail.com

Raon Spielberg Berek

About me:

Hi, I'm Raon Spielberg Berek, a **final-year Information Systems student** at **Harapan Bangsa Institute of Technology** and a **Junior Data Scientist**. I possess **strong public speaking and analytical thinking skills**, honed through experience **coordinating numerous projects and events**, as well as working on **various data analysis projects** and **developing applications using Java, SQL, and Python**.

Employment Objective:

A career in **Information System**, specifically in the areas of **data analysis**, that will optimally utilize **strong analytical skills** as well as **public speaking**, **SQL, and Python-based programming skills.**

Course and Certification

Machine Learning Distinction Graduate
- Google Bangkit Academy 2024 Batch 1

June 2024

Machine Learning Specialization
- DeepLearning.AI, Stanford University, Coursera

April 2024

Oracle Foundations
- ORACLE UNIVERSITY, ITHB Career Resource Center

April 2024

Learn Data Analysis with Python
- Dicoding Indonesia

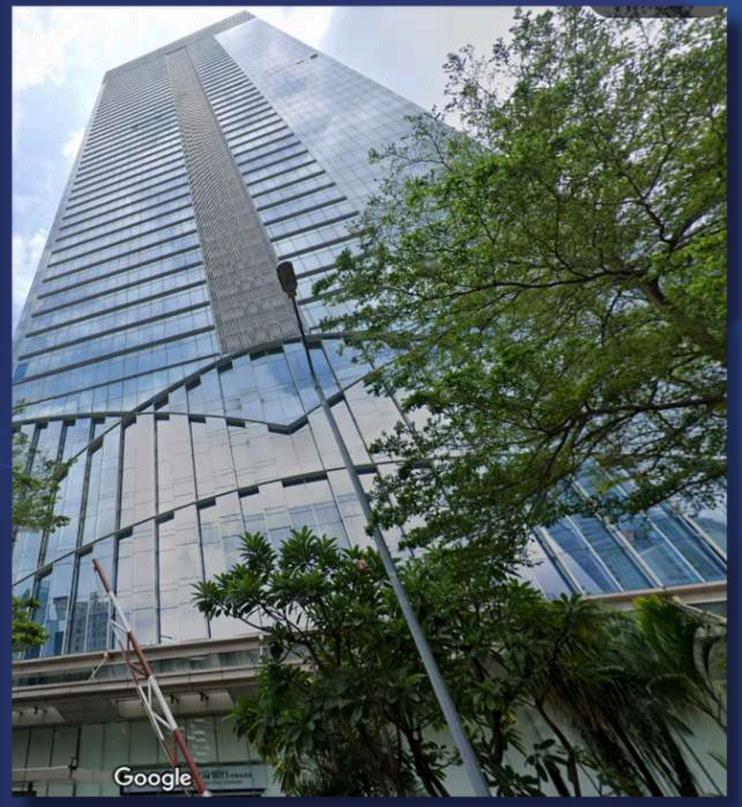
March 2024

Agenda

1. About Company
2. About the Project: Business Understanding
3. Data Understanding
4. Data Preparation 1
5. Exploratory Data Analysis (EDA)
6. Data Preparation 2
7. Data Modelling
8. Model Evaluation
9. Conclusion
10. Business Recommendation

About Company

ID/X Partners (PT IDX Consulting) didirikan pada tahun 2002 dan telah melayani berbagai perusahaan di **Asia dan Australia** di berbagai sektor industri, terutama di bidang **layanan keuangan, telekomunikasi, manufaktur, dan ritel**. Perusahaan ini menawarkan layanan konsultasi yang berfokus pada pemanfaatan **solusi data analytics and decisioning** (DAD), yang dikombinasikan dengan **manajemen risiko serta strategi pemasaran yang terintegrasi, guna membantu klien meningkatkan profitabilitas portofolio serta efisiensi proses bisnis mereka**. Dengan pendekatan konsultasi yang menyeluruh dan solusi teknologi yang lengkap, ID/X Partners menjadi penyedia layanan terpadu (**one-stop service provider**).



About the Project: Business Understanding

Memahami latar belakang proyek, bisnis yang terlibat, dataset yang digunakan, problem statement, dan project objective.

- **Link code here!**
- **GitHub repo here!**
- **Project explanation video here!**

About the Project: Business Understanding

- **Latar Belakang**
  - Kredit merupakan salah satu **layanan keuangan utama** yang diberikan oleh **lembaga keuangan**.
  - Risiko kredit (**credit risk**) adalah kemungkinan peminjam **gagal membayar** pinjamannya, yang dapat berdampak pada **stabilitas finansial** pemberi pinjaman.
  - Penggunaan **Machine Learning** dalam **prediksi risiko kredit** dapat membantu meningkatkan **akurasi penilaian kelayakan pinjaman** dan mengurangi **tingkat kredit bermasalah** (non-performing loans).
- **Dataset yang digunakan:**
  - Dataset yang berisi informasi mengenai **pinjaman**, seperti jumlah pinjaman, lama tenor, riwayat pembayaran, status pekerjaan, dan kepemilikan rumah.
  - Fitur utama meliputi **variabel numerik dan kategorikal** yang mempengaruhi risiko kredit.

About the Project: Business Understanding

- **Problem Statement**
  - **Tingkat keakuratan rendah dalam menilai risiko kredit:**
    - Metode penilaian risiko kredit yang ada **belum optimal**, menyebabkan **kesulitan** dalam **mengidentifikasi** peminjam yang **berisiko tinggi** (good/bad).
  - **Potensi kerugian yang tinggi akibat kredit macet:**
    - Kesalahan dalam menilai risiko kredit dapat mengakibatkan peningkatan **kredit macet** dan **kerugian finansial** bagi perusahaan.
  - **Kesulitan dalam mengoptimalkan keputusan bisnis**
    - Kurangnya informasi yang akurat tentang risiko kredit (good/bad credit risk) **menghambat pengambilan keputusan** bisnis yang optimal.

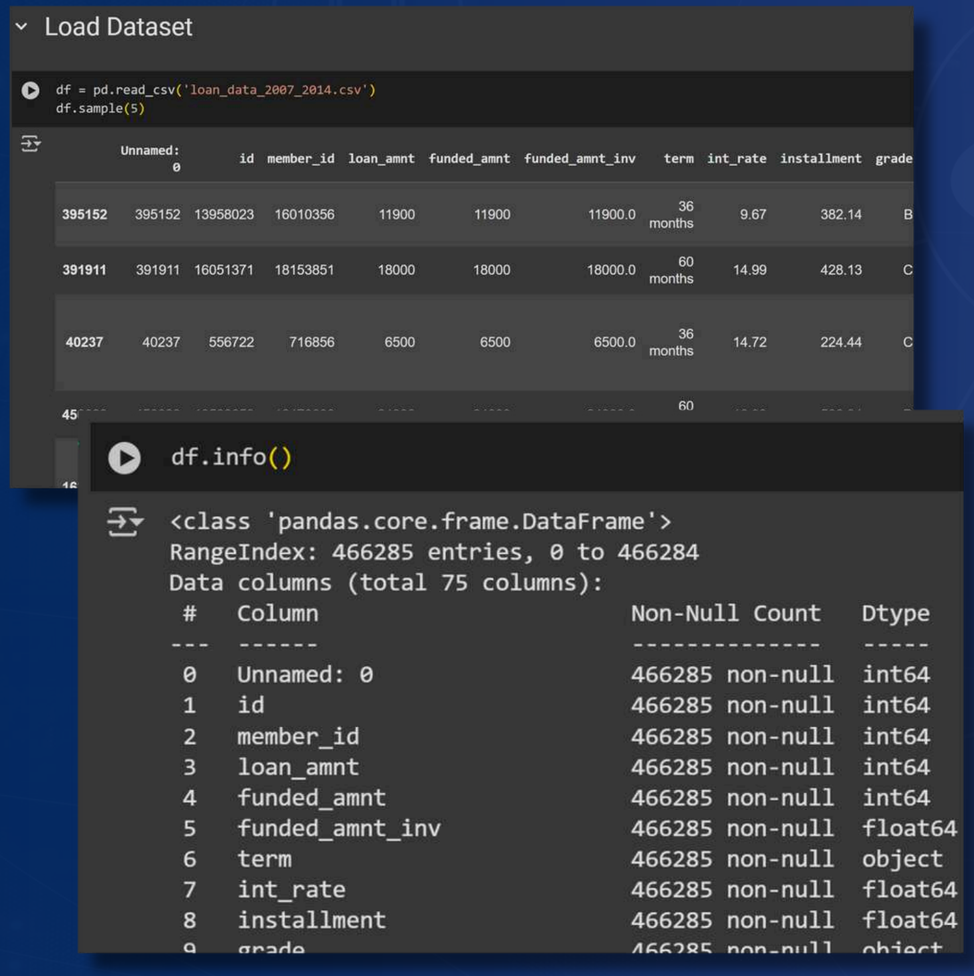
About the Project: Business Understanding

- **Project Objective**
  - **Mengembangkan model machine learning (ML) untuk prediksi risiko kredit yang akurat:**
    - Membangun **model ML klasifikasi** yang dapat memprediksi risiko kredit (**good/bad**) dengan **akurasi** yang **lebih tinggi** daripada metode yang ada.
  - **Mengurangi potensi kerugian akibat kredit macet:**
    - Dengan **mengidentifikasi** peminjam berisiko tinggi secara **lebih akurat**, model akan membantu **mengurangi jumlah kredit macet** dan **meminimalkan kerugian finansial**.
  - **Memberikan informasi yang akurat untuk pengambilan keputusan bisnis:**
    - Model akan menyediakan informasi yang akurat dan handal tentang risiko kredit untuk **mendukung pengambilan keputusan bisnis** yang lebih **efektif** dan **optimal**.

Data Understanding

Mengumpulkan data awal dan menjelajahi karakteristiknya.

Data Understanding: Basic Understanding



Load Dataset

```
df = pd.read_csv('loan_data_2007_2014.csv')
df.sample(5)
```

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade
395152	395152	13958023	16010356	11900	11900	11900.0	36 months	9.67	382.14	B
391911	391911	16051371	18153851	18000	18000	18000.0	60 months	14.99	428.13	C
40237	40237	556722	716856	6500	6500	6500.0	36 months	14.72	224.44	C
45							60			

```
df.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
```

- Dataset yang berisikan data pinjaman dari tahun 2007-2014 (jumlah pinjaman, profil peminjam, dsb).
- Terdapat **75 kolom dan 466.285 baris data**, dengan masing-masing tipe data memiliki jumlah kolom sebagai berikut:
  - **Float: 46 Kolom**
  - **Int: 7 Kolom**
  - **Object: 22 Kolom**

- Terlihat juga **belum terdapat kolom target**/labelnya, maka akan dibuat nanti.
- **Banyak kolom yang tidak relevan untuk diolah model ML**, akan didrop nanti (cth: kolom “Unnamed: 0” hanyalah indeks yang tidak diperlukan).

Data Understanding: Checking Missing Values

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

collection_recovery_fee	0
last_pymnt_d	376
last_pymnt_amnt	0
next_pymnt_d	227214

```
    'Missing Percentage': missing_percentage
}).sort_values(by='Missing Percentage', ascending=False)

print(missing_df)
```

open_acc_6m	open_acc_6m	100.000000
dti_joint	dti_joint	100.000000
annual_inc_joint	annual_inc_joint	100.000000
verification_status_joint	verification_status_joint	100.000000
all_util	all_util	100.000000
inq_last_12m	inq_last_12m	100.000000
total_cu_tl	total_cu_tl	100.000000
mths_since_last_record	mths_since_last_record	86.566585
mths_since_last_major_derog	mths_since_last_major_derog	78.773926
desc	desc	72.981975
mths_since_last_delinq	mths_since_last_delinq	53.690554
next_pymnt_d	next_pymnt_d	48.728567
tot_coll_amt	tot_coll_amt	15.071469
total_rev_hi_lim	total_rev_hi_lim	15.071469
tot_cur_bal	tot_cur_bal	15.071469
emp_title	emp_title	5.916553
emp_length	emp_length	4.505399
last_pymnt_d	last_pymnt_d	0.080637
revol_util	revol_util	0.072917
collections_12_mths_ex_med	collections_12_mths_ex_med	0.031097

- Terdapat **20** kolom memiliki persentase **missing values >= 50%**, dan **17** diantaranya bahkan memiliki **100% missing values.** Kolom-kolom tersebut **akan didrop** nanti guna menjaga kualitas model.

- Terlihat juga **belum terdapat kolom target**/labelnya, maka akan dibuat nanti.

- **Sisa kolom** yang memiliki missing values **<50%** akan dihandle dengan **“replace with median value”** atau **didrop** juga jika **persentase missingnya relatif sedikit**.

Data Understanding: Checking Duplicate Values

```
[9]  df.duplicated().sum()

     np.int64(0)
```

- Terlihat **tidak terdapat duplicate values**, maka tidak perlu menghandle duplicate values.

Data Preparation 1

Membersihkan dan melakukan pra-pemrosesan data awal.

Data Preparation 1: Membuat Kolom Target

```
def classify_credit_risk(status):
    high_risk_status = ['Charged Off', 'Default', 'Late (31-120
                        'Late (16-30 days)', 'Does not meet the

    low_risk_status = ['Fully Paid', 'Current', 'Does not meet

    if st
        r
    elif
        r
```

```
df['credit_risk'].value_counts()
```

credit_risk	count
Good	414099
Bad	52186

dtype: int64

_cu_tl	inq_last_12m	credit_risk
NaN	NaN	Good
NaN	NaN	Bad
NaN	NaN	Good

- Dibuatlah kolom “**credit_risk**” yang merepresentasikan suatu baris data peminjam apakah memiliki karakteristik risiko peminjaman yang:
  - “**good**” → **kemungkinan tinggi lancar melunasi**, atau
  - “**bad**” → kemungkinan rendah lancar melunasi alias **akan telat bayar**

- Dibuat berdasarkan status peminjam dari kolom “**loan_status**”.

- Terlihat **distribusi** jumlah baris data yang memiliki value credit_risk “Good” adalah sangat jauh berbeda dari yang “Bad” (**imbalance**). Imbalance data akan dihandle pada Data Preparation 2.

Data Preparation 1: Removing Irrelevant Features

```
drop_cols = [
    "Unnamed: 0", "id", "member_id", "url", "desc", "zip_code", "policy_code",
    "application_type", "emp_title", "title", "loan_status"
]

df = df.drop(columns=drop_cols)
```

Kolom-kolom yang didrop:

1. Unnamed: 0 → Hanya indeks, tidak diperlukan.
2. id & member_id → ID unik yang tidak berkontribusi pada analisis atau model prediksi.
3. url → Link tidak digunakan dalam analisis.
4. desc → Deskripsi teks panjang, tidak digunakan.
5. zip_code → Data kode pos biasanya tidak berpengaruh signifikan dalam analisis pinjaman.
6. policy_code → Hanya memiliki satu nilai unik, sehingga tidak informatif.
7. application_type → Semua data memiliki nilai yang sama, bisa dihapus.
8. emp_title → Deskripsi teks yang tidak diperlukan
9. title → Deskripsi teks yang tidak diperlukan
10. loan_status → Hanya digunakan untuk menentukan kolom target, setelah digunakan maka akan dihapus guna menghindari korelasi yang sangat tinggi dengan kolom credit_risk.

Data Preparation 1: Handle Missing Values

```
drop_cols = [
    "max_bal_bc", "open_rv_24m", "inq_fi", "open_rv_12m", "il_util", "mths_since_rcnt_il",
    "total_bal_il", "open_il_24m", "open_il_12m", "open_il_6m", "open_acc_6m", "dti_joint",
    "annual_inc_joint", "verification_status_joint", "all_util", "inq_last_12m", "total_cu_tl",
    "mths_since_last_record", "mths_since_last_major_derog", "mths_since_last_delinq"
]

df = df.drop(columns=drop_cols)
```

- Kolom dengan **≥50% Missing Values** dihandle dengan **didrop** langsung.
- Kolom dengan **100% Missing Values**:
  - 17 kolom, termasuk max_bal_bc, open_rv_24m, inq_fi, dti_joint, dan lainnya.
- Kolom dengan **≥50% Missing Values**:
  - 3 kolom, yaitu mths_since_last_record (87.56%), mths_since_last_major_derog (78.77%), dan mths_since_last_delinq (53.69%)

Data Preparation 1: Handle Missing Values

```
# Mengonversi next_pymnt_d ke Format datetime
df['next_pymnt_d'] = pd.to_datetime(df['next_pymnt_d'], format='%b-%y')

## mengisi kolom next_pymnt_d dengan median
df['next_pymnt_d'] = df['next_pymnt_d'].fillna(df['next_pymnt_d'].median())
```

```
# Mengisi kolom tot_coll_amt, tot_cur_bal, total_rev_hi_lim yang memiliki missing value >15%, dengan median
df['tot_coll_amt']= df['tot_coll_amt'].fillna(df['tot_coll_amt'].median())
df['tot_cur_bal']= df['tot_cur_bal'].fillna(df['tot_cur_bal'].median())
df['total_rev_hi_lim']= df['total_rev_hi_lim'].fillna(df['total_rev_hi_lim'].median())
```

```
# menghapus sisa baris data yang memiliki missing value
df.dropna(inplace = True)
df.isnull().sum()
```

sub_grade	0
emp_length	0
home_ownership	0
annual_inc	0
verification_status	0
issue_d	0
pymnt_plan	0
purpose	0
addr_state	0

- **Mengonversi** next_pymnt_d ke format **datetime**, lalu mengisi nilai missing dengan **median**
- **Mengisi** kolom tot_coll_amt, tot_cur_bal, total_rev_hi_lim yang memiliki missing value **>15% dan <49%**, dengan **median**
- **Menghapus** sisa baris data yang memiliki missing values **<5%.**

Data Preparation 1: Changing Data Type

```
[75] # Mengubah kolom ke format datetime
     date_columns = ["issue_d", "earliest_cr_line", "last_pymnt_d", "last_credit_pull_d"]

     for col in date_columns:
         df[col] = pd.to_datetime(df[col], format='%b-%y')  # Mengubah ke datetime
```

```
[76] # mengonversi data kategorikal menjadi numerik untuk kolom term

     df['term'] = df['term'].str.extract('(\d+)').astype(float)  # Ambil angka dari string
```

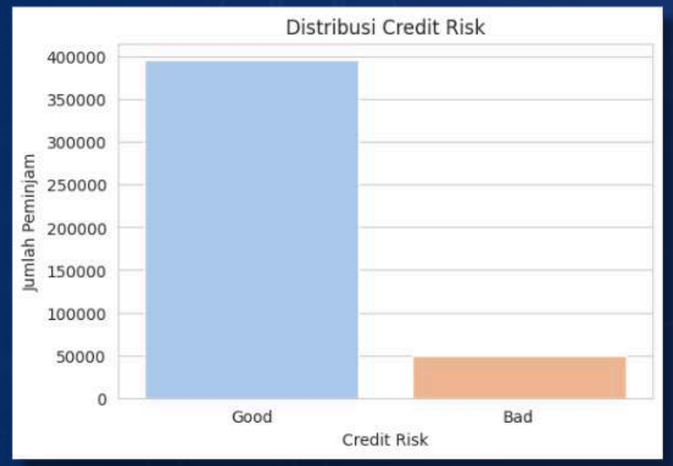
- **Mengubah tipe data kolom object** yang mengandung data tanggal menjadi **bertipe data datetime**, sehingga bisa **dianalisis** lebih lanjut pada tahap **EDA**.

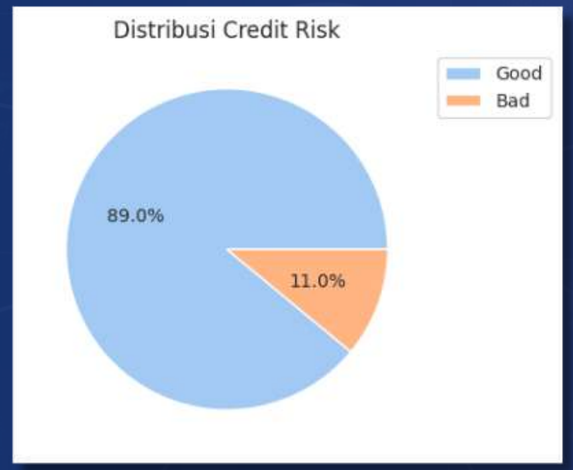
- **Mengonversi data kategorikal** kolom '**term**' menjadi **numerik**. Cth: ' 36 months' menjadi 36 dan ' 60 months' menjadi 60.

Exloratory Data Analysis (EDA)

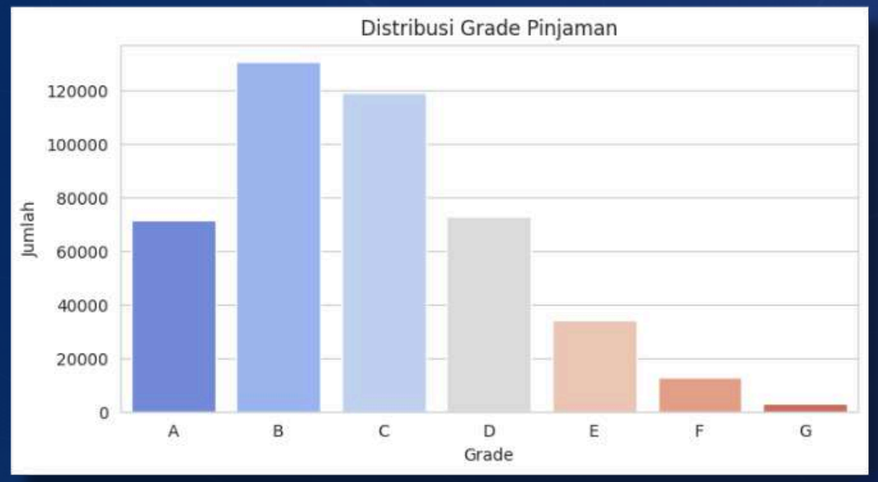
Mengungkap pola dan wawasan dari data.

EDA: Mengecek Distribusi Data Kategorikal





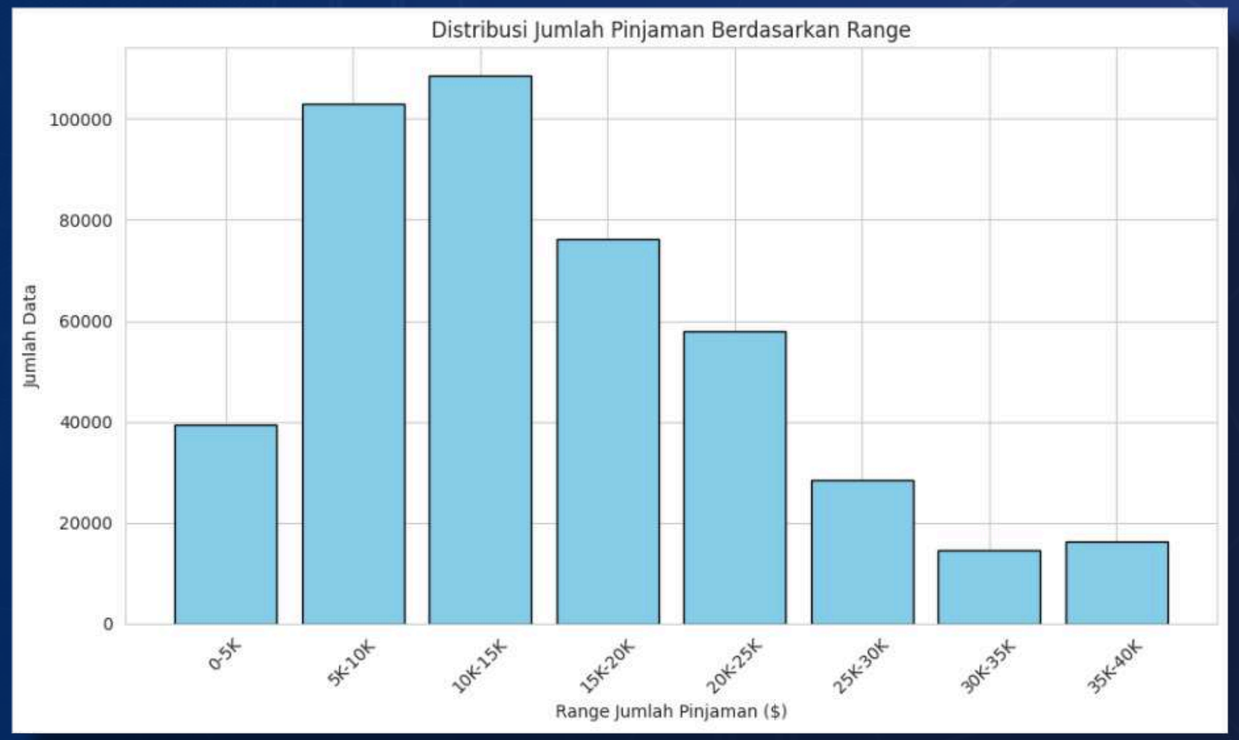
- **Distribusi Credit Risk Good dan Bad sangat jomplang**/jauh berbeda (imbalance), maka perlu **dihandle dengan teknik oversampling** yang memperbanyak **data sistetis Bad** guna **meningkatkan kualitas pelatihan model**.



- **Dominasi risiko menengah**: Grade B, C, dan D **mendominasi**, menunjukkan portofolio dengan kredit **cukup baik** tapi **belum optimal** (seperti Grade A).

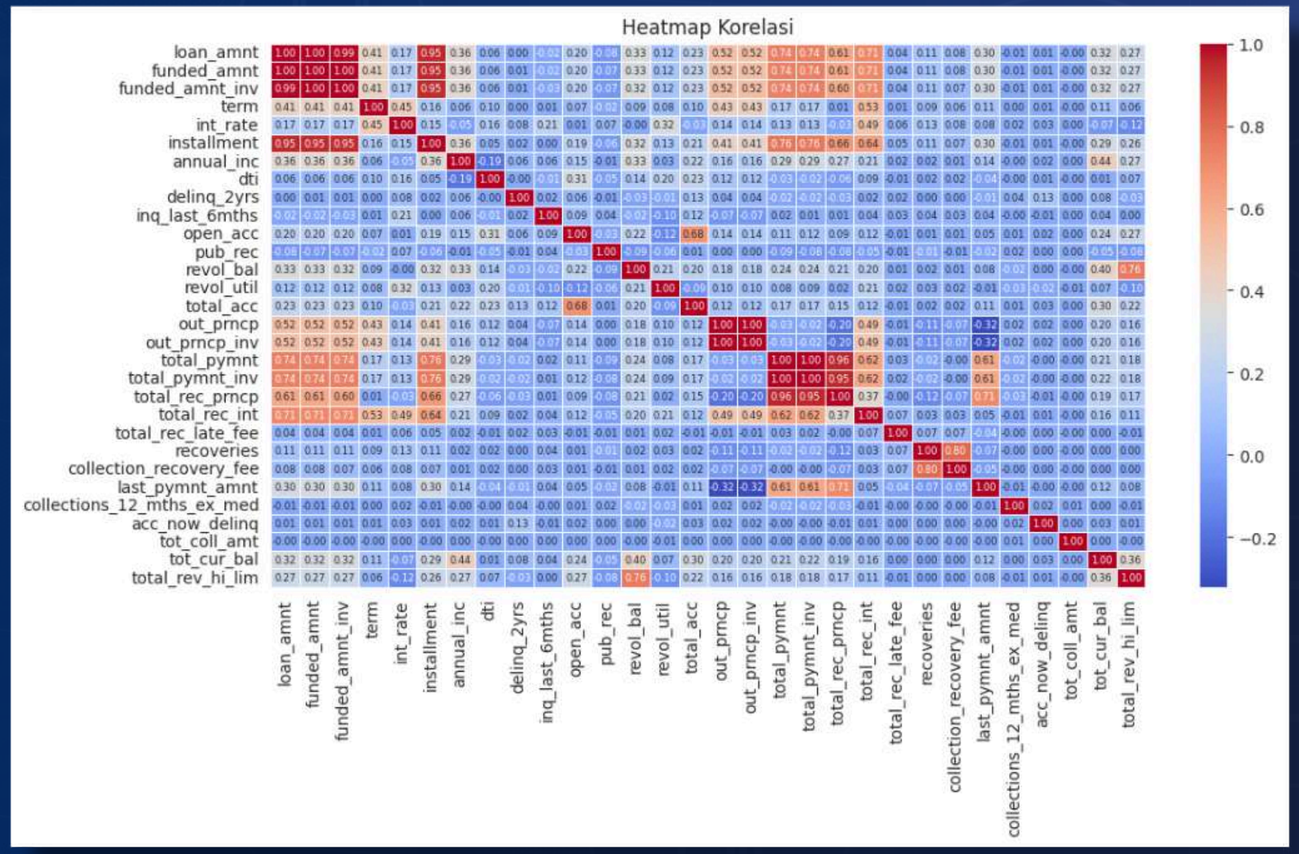
- **Perhatian khusus diperlukan**: Terdapat **50.000** peminjam berisiko tinggi (E, F, dan G) yang **butuh pemantauan intensif**

EDA: Mengecek Distribusi Data Numerik



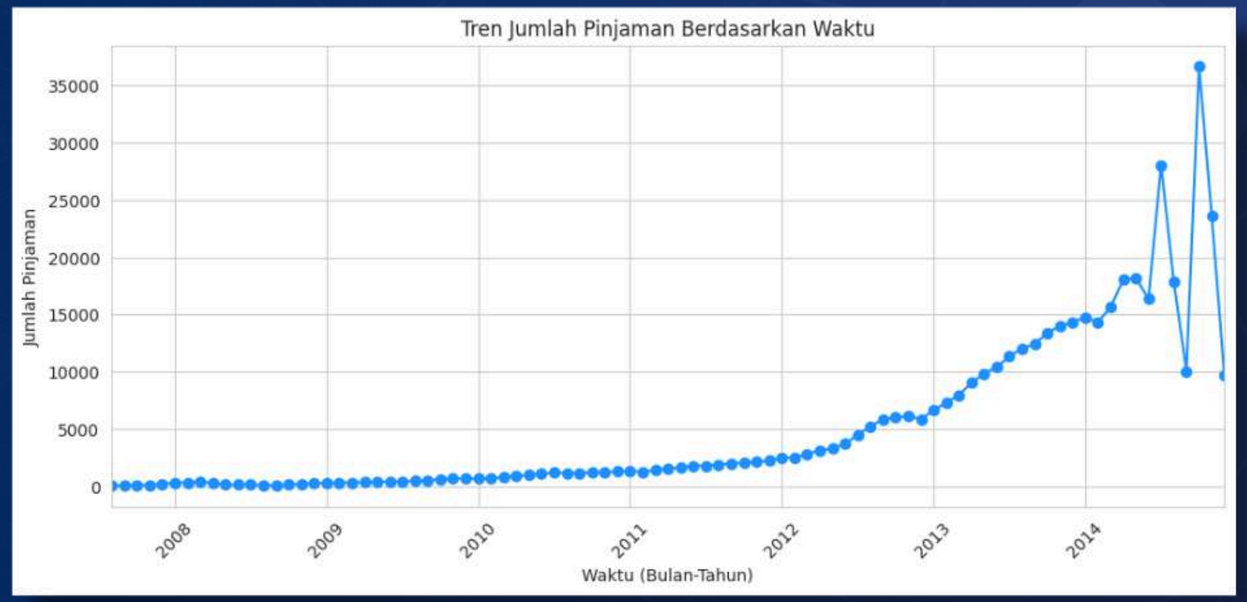
- **Distribusi bersifat right-skewed** (positively skewed), mengindikasikan **konsentrasi data** pada **jumlah pinjaman kecil hingga sedang** ($5.000 – $15.000), dengan sebagian kecil peminjam mengambil pinjaman dalam jumlah besar.
- **Jumlah peminjam menurun signifikan seiring meningkatnya** nilai pinjaman di atas **$20.000**, menandakan j**umlah besar lebih jarang diajukan.**
- **Fokuskan evaluasi risiko lebih ketat pada pinjaman > $20.000**, karena walaupun **jumlahnya lebih sedikit**, nilai **kerugiannya jauh lebih besar** jika terjadi **gagal bayar**.

EDA: Mengecek Korelasi Antar Variabel Numerik



- Terlihat **beberapa kolom memiliki korelasi yang sangat tinggi dengan kolom lainnya** (>=0.8), seperti loan_amnt vs funded_amnt (1.00) dan out_prncp vs out_prncp_inv (1.00).
- **Perlu di drop salah satu kolom yang sangat berkorelasi tinggi** karena informasi yang dikandungnya **sangat mirip** atau hampir identik, sehingga bisa menimbulkan **redundansi** dalam analisis atau model prediktif.
- Drop dilakukan pada Data Prep 2.

EDA: Mengecek Tren Jumlah Pinjaman Berdasarkan Waktu

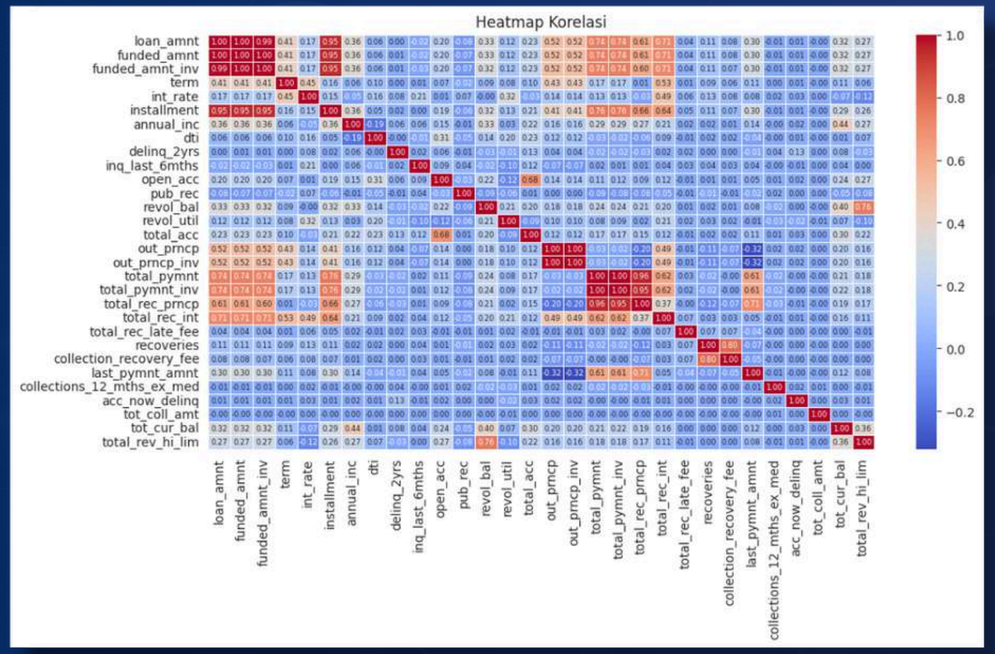


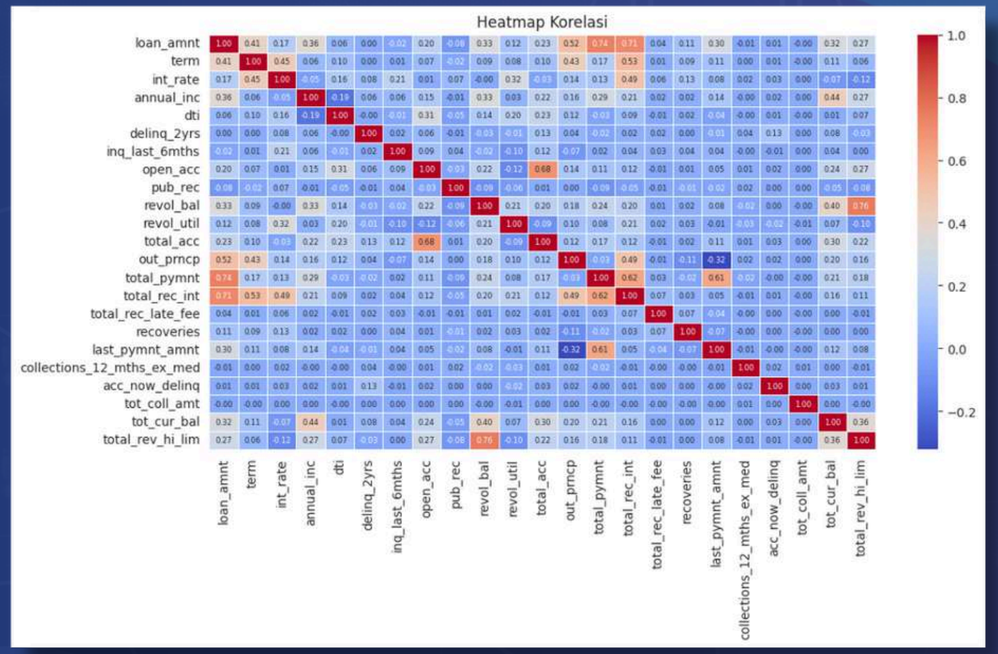
- Jumlah pinjaman **tumbuh perlahan** dari tahun **2007** hingga **pertengahan 2012**, menunjukkan fase **adopsi awal** platform pinjaman.
- Terjadi **lonjakan tajam pada tahun 2013 dan terutama 2014**, mencerminkan **ekspansi besar-besaran** atau meningkatnya **kepercayaan** pengguna.
- **Tahun 2014** menunjukkan pola naik-turun **ekstrem**, menandakan kemungkinan perubahan kebijakan, promosi besar, atau faktor eksternal lainnya.

Data Preparation 2

Perbaikan data lebih lanjut berdasarkan temuan EDA.

Data Preparation 2: Multicollinearity Reduction





- Dilakukan **drop kolom** yang memiliki **korelasi sangat tinggi dengan kolom lainnya** (korelasi >= 0.8).
- **Dipilih salah satu kolom yang representatif dalam pasangan kolom** yang bekorelasi tersebut. Contoh: loan_amnt, funded_amnt, dan funded_amnt_inv → hanya loan_amnt yang disimpan.

Data Preparation 2: Encoding

```python
from sklearn.preprocessing import OrdinalEncoder

# Fitur dengan urutan yang jelas
ordinal_features = ["credit_risk", "grade", "sub_grade", "emp_length", "verification_status"]

# Mapping untuk ordinal encoding
ordinal_mapping = {
    "credit_risk": {"Bad": 0, "Good": 1},
    "grade": {"A": 0, "B": 1, "C": 2, "D": 3, "E": 4, "F": 5, "G": 6},
    "sub_grade": {f"{g}{n}": i for i, (g, n) in enumerate([(g, n) for g in "ABCDEFG" for n in range(1, 6)])},
    "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, "Source Verified": 1, "Verified": 2}
}

# Ubah kategori ke nilai numerik
for col, mapping in ordinal_mapping.items():
    df[col] = df[col].map(mapping)
```

Label Encoding

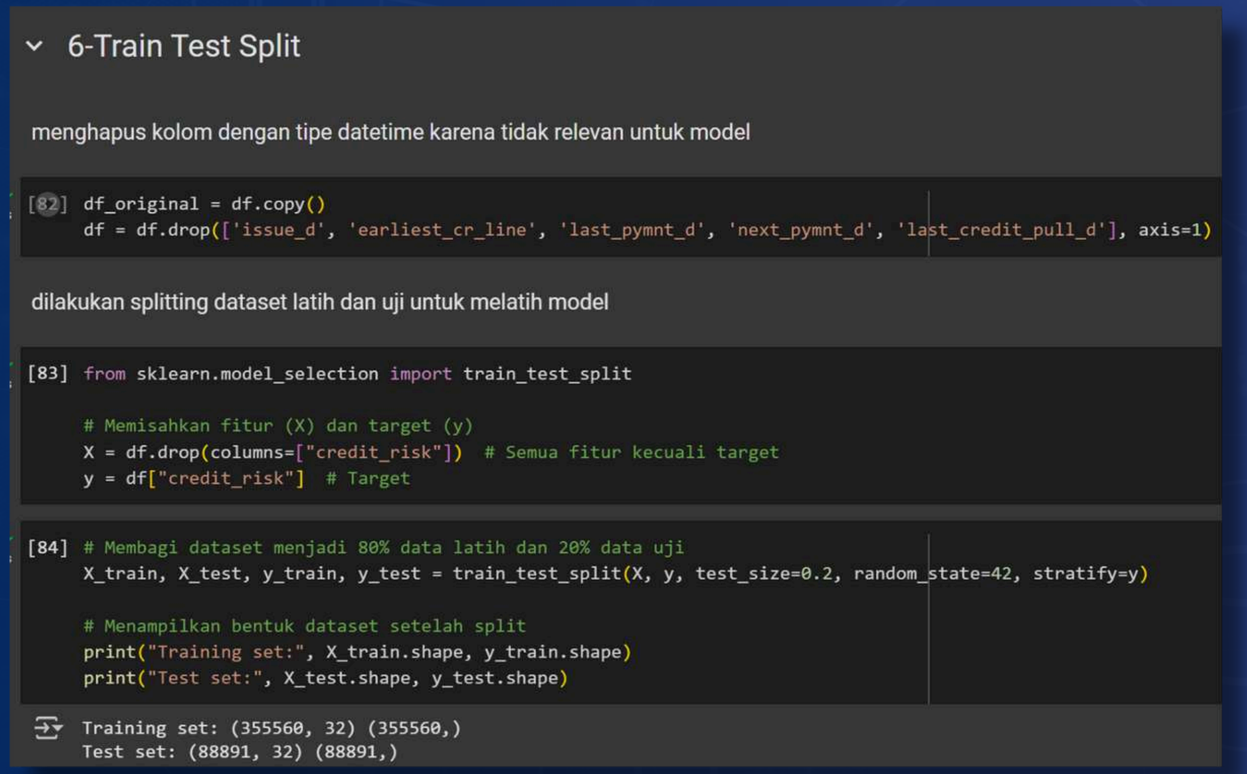
```python
[79] label_features = ["home_ownership", "purpose", "addr_state", "initial_list_status", "pymnt_plan", ]

    for feature in label_features:
      le = LabelEncoder()
      df[feature] = le.fit_transform(df[feature])
```

```python
df.head()
```

	loan_amnt	term	int_rate	grade	sub_grade	emp_length	home_ownership	annual_inc	verification_status
0	5000	36.0	10.65	1	6	10	5	24000.0	2
1	2500	60.0	15.27	2	13	0	5	30000.0	1
2	2400	36.0	15.96	2	14	10	5	12252.0	0
3	10000	36.0	13.49	2	10	10	5	49200.0	1
4	3000	60.0	12.69	1	9	1	5	80000.0	1

- **Dilakukan encoding pada variabel kategorikal** untuk mengubah data non-numerik menjadi format yang dapat diproses oleh model ML.
- **Diterapkan Ordinal Encoding pada kolom dengan hierarki jelas:** "credit_risk" (0 = Bad, 1 = Good), "grade", "sub_grade", "emp_length", dan "verification_status" untuk mempertahankan urutan logika nilai kategorinya.
- **Digunakan Label Encoding pada kolom tanpa hierarki** ("home_ownership", "purpose", "addr_state", "initial_list_status", "pymnt_plan") yang hanya membutuhkan transformasi ke nilai numerik tanpa makna urutan.

Data Preparation 2: Train Test Split



6-Train Test Split

menghapus kolom dengan tipe datetime karena tidak relevan untuk model

```
[82] df_original = df.copy()
     df = df.drop(['issue_d', 'earliest_cr_line', 'last_pymnt_d', 'next_pymnt_d', 'last_credit_pull_d'], axis=1)
```

dilakukan splitting dataset latih dan uji untuk melatih model

```
[83] from sklearn.model_selection import train_test_split

     # Memisahkan fitur (X) dan target (y)
     X = df.drop(columns=["credit_risk"])  # Semua fitur kecuali target
     y = df["credit_risk"]  # Target
```

```
[84] # Membagi dataset menjadi 80% data latih dan 20% data uji
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

     # Menampilkan bentuk dataset setelah split
     print("Training set:", X_train.shape, y_train.shape)
     print("Test set:", X_test.shape, y_test.shape)
```

```
Training set: (355560, 32) (355560,)
Test set: (88891, 32) (88891,)
```

- **Meng-drop kolom** dengan tipe data **datetime** karena tidak relevan untuk model.
- **Memisahkan** fitur (X) dan target (y).
- **Membagi dataset** menjadi 80% data latih dan 20% data uji.

Data Preparation 2: Handling Imbalance Data

```python
from imblearn.over_sampling import SMOTE

# 2. Tangani Imbalance Data di Data Latih dengan SMOTE (contoh oversampling)
smote = SMOTE(random_state=42)
X_train_balanced, y_train_balanced = smote.fit_resample(X_train, y_train)

# Cek distribusi sebelum dan sesudah SMOTE
print("Distribusi y sebelum SMOTE:\n", y_train.value_counts())
print("Distribusi y setelah SMOTE:\n", y_train_balanced.value_counts())
```

```
Distribusi y sebelum SMOTE:
 credit_risk
1    316441
0     39119
Name: count, dtype: int64
Distribusi y setelah SMOTE:
 credit_risk
0    316441
1    316441
Name: count, dtype: int64
```

- **SMOTE** (synthetic minority over-sampling technique) diterapkan **hanya pada data training** untuk **menyeimbangkan** kelas Good dan Bad dengan menambah sampel sintetik **kelas minoritas** (kelas Bad).
- Proses dilakukan **setelah train-test split** guna **mencegah data leakage** dan memastikan evaluasi model tetap **objektif**.
- **Tujuannya meningkatkan kinerja** model pada **prediksi kelas minoritas** tanpa **mengganggu distribusi asli** data test.

Data Preparation 2: Standarization

8-Standarization

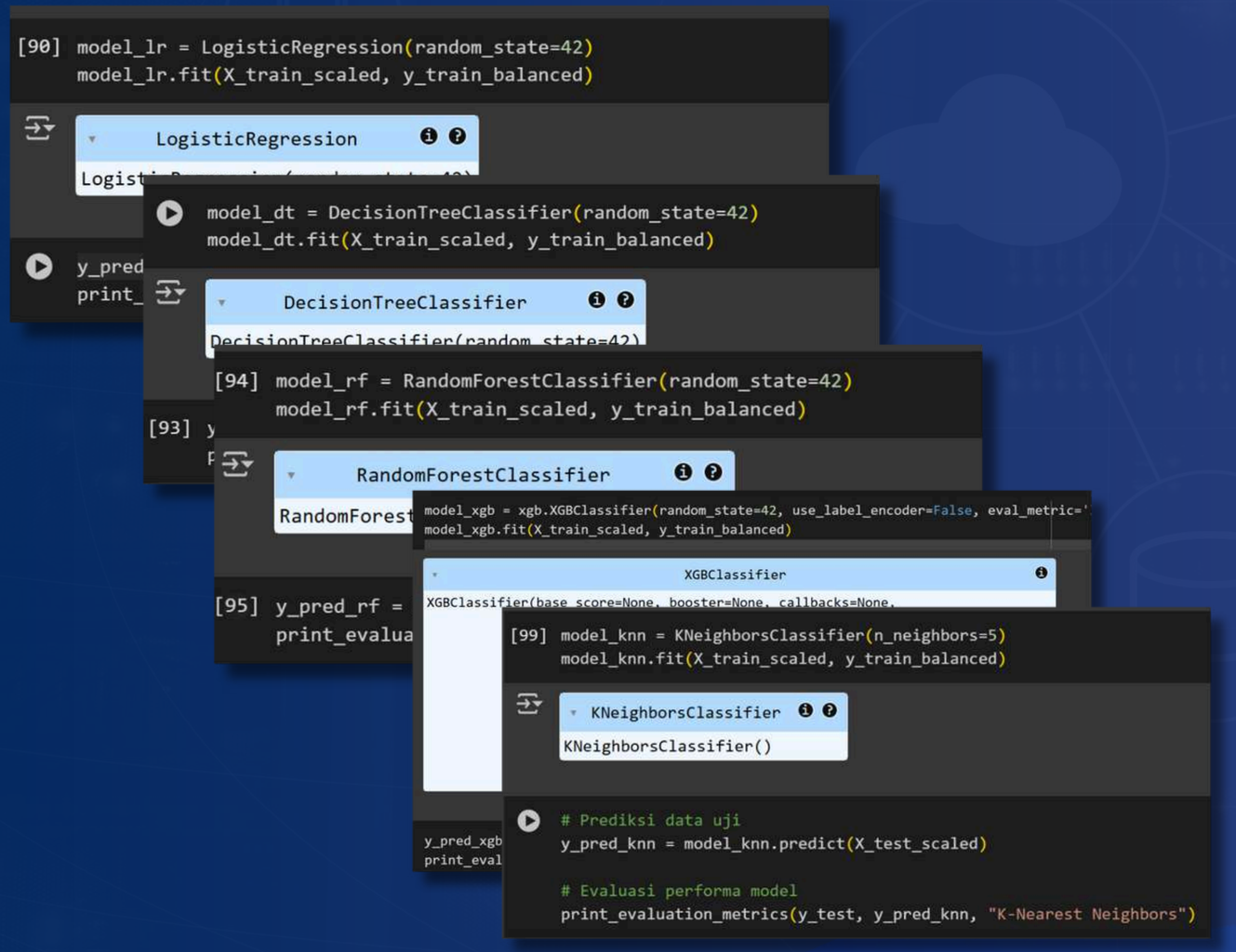
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train_balanced)
X_test_scaled = scaler.transform(X_test)

X_train_scaled = pd.DataFrame(X_train_scaled, columns=X.columns)
X_test_scaled = pd.DataFrame(X_test_scaled, columns=X.columns)
```

- **Data fitur pada train dan test distandarisasi** menggunakan **StandardScaler**, agar memiliki distribusi dengan mean = 0 dan standar deviasi = 1.
- Hasil transformasi **dikonversi** kembali menjadi **DataFrame** dengan nama kolom yang **sama** seperti data awal.

Data Modelling

Membangun dan melatih model prediktif klasifikasi (Good or Bad).

Data Modelling



- Dibangun dan dilatihlah **5 model** ML klasifikasi yang umum digunakan:
  - **Logistic Regression**
  - **Decision Tree**
  - **Random Forest**
  - **XGBoost** (Gradient Boosting)
  - **KNN** (K-Nearest Neighbors).

Model Evaluation (Using Testing Dataset)

Menilai kinerja dan memilih model terbaik.

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

           0       0.85      0.84      0.84      9780
           1       0.98      0.98      0.98     79111

    accuracy                           0.97     88891
   macro avg       0.91      0.91      0.91     88891
weighted avg       0.97      0.97      0.97     88891

Confusion Matrix:
 [[ 8232  1548]
 [ 1481 77630]]
Accuracy: 0.9659245592917168
ROC AUC Score: 0.9114986297500918
Precision: 0.9804491146530602
Recall: 0.9812794680891406
F1 Score: 0.9808641156365888
------------------------------
```

```
--- Random Forest ---
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.82      0.89      9780
           1       0.98      1.00      0.99     79111

    accuracy                           0.98     88891
   macro avg       0.98      0.91      0.94     88891
weighted avg       0.98      0.98      0.98     88891

Confusion Matrix:
 [[ 7997  1783]
 [  191 78920]]
Accuracy: 0.9777930274155989
ROC AUC Score: 0.9076374161602919
Precision: 0.9779066453539521
Recall: 0.9975856707663915
F1 Score: 0.9876481409638705
------------------------------
```

```
--- K-Nearest Neighbors ---
Classification Report:
              precision    recall  f1-score   support

           0       0.41      0.70      0.51      9780
           1       0.96      0.87      0.91     79111

    accuracy                           0.86     88891
   macro avg       0.68      0.79      0.71     88891
weighted avg       0.90      0.86      0.87     88891

Confusion Matrix:
 [[ 6802  2978]
 [ 9911 69200]]
Accuracy: 0.8550021936978996
ROC AUC Score: 0.785110676079136
Precision: 0.9587408905760758
Recall: 0.8747203296633843
F1 Score: 0.9148054385976508
------------------------------
```

```
--- Decision Tree ---
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.85      0.81      9780
           1       0.98      0.97      0.98     79111

    accuracy                           0.96     88891
   macro avg       0.88      0.91      0.89     88891
weighted avg       0.96      0.96      0.96     88891

Confusion Matrix:
 [[ 8339  1441]
 [ 2448 76663]]
Accuracy: 0.9562497890675097
ROC AUC Score: 0.9108573115111823
Precision: 0.9815502407047014
Recall: 0.9690561363147981
F1 Score: 0.9752631746334637
------------------------------
```

```
--- XGBoost ---
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.85      0.91      9780
           1       0.98      1.00      0.99     79111

    accuracy                           0.98     88891
   macro avg       0.99      0.92      0.95     88891
weighted avg       0.98      0.98      0.98     88891

Confusion Matrix:
 [[ 8285  1495]
 [   85 79026]]
Accuracy: 0.982225422146224
ROC AUC Score: 0.9230312873018183
Precision: 0.9814334148855578
Recall: 0.9989255602887083
F1 Score: 0.9901022351408238
------------------------------
```

Model Evaluation (Using Testing Dataset)

Model	Accuracy	F1-Score	ROC AUC	Precision	Recall
XGBoost	**98.22%**	**99.01%**	**92.30%**	98.14%	**99.89%**
Random Forest	97.78%	98.76%	90.76%	97.79%	99.76%
Logistic Reg.	96.59%	98.09%	91.15%	98.04%	98.13%
Decision Tree	95.62%	97.52%	91.08%	**98.15%**	96.90%
K-NN	85.50%	91.48%	78.51%	95.87%	87.47%

- **Logistic Regression** menunjukkan **performa stabil** dengan **F1-score tinggi** (0.9809) dan **keseimbangan precision-recall** yang baik, cocok untuk baseline model.
- **Decision Tree** memiliki **presisi yang baik** namun **recall sedikit lebih rendah**, sehingga cenderung melewatkan lebih banyak kasus positif dibanding model lain.
- **Random Forest** menghasilkan **akurasi** dan **F1-score tinggi** (0.9876) dengan **recall** hampir sempurna (0.9976), menunjukkan kemampuan klasifikasi yang sangat kuat.
- **XGBoost** adalah **model terbaik** dengan **akurasi tertinggi** (98.22%), **F1-score tertinggi** (0.9901), dan **ROC AUC terbaik** (0.923), sangat unggul dalam mendeteksi kelas positif.
- **KNN** memiliki **performa terendah di semua metrik**, dengan recall dan akurasi yang jauh di bawah model lain, sehingga kurang direkomendasikan.

Conclusion

Dari lima model klasifikasi yang dilatih dan dievaluasi, yaitu Logistic Regression, Decision Tree, Random Forest, XGBoost, dan K-Nearest Neighbors, **model XGBoost menunjukkan performa terbaik**. Hal ini ditunjukkan melalui **akurasi yang tinggi serta tingkat kesalahan klasifikasi yang paling rendah** dibandingkan model lainnya. Oleh karena itu, **XGBoost direkomendasikan** sebagai model yang **paling optimal** untuk digunakan dalam **memprediksi risiko kredit**.

Business Recommendation

- **Gunakan Model XGBoost:**
  - **Terapkan model XGBoost** untuk prediksi risiko kredit karena memiliki **akurasi** dan **f1-score tertinggi** dibandingkan model lain.

- **Percepat dan Otomatiskan Proses Evaluasi Kredit:**
  - Model XGBoost dapat **diintegrasikan ke sistem** agar secara **otomatis mengevaluasi pengajuan pinjaman**; misalnya, jika hasil prediksi menunjukkan credit risk = Bad, sistem langsung memberi **notifikasi** ke tim analis untuk **verifikasi** lebih lanjut atau meminta **jaminan tambahan**.

- **Kembangkan Strategi Mitigasi Risiko Berdasarkan Data Historis:**
  - **Data hasil prediksi saat ini** bisa **dianalisis lebih lanjut bersama data baru** untuk mengembangkan **strategi mitigasi**, misalnya **menolak otomatis pengajuan** dari peminjam berprofil serupa dengan **catatan gagal bayar** atau **menyesuaikan bunga pinjaman**.

Terima Kasih

Terima kasih telah menyimak presentasi ini.
Jika ada pertanyaan, diskusi, atau kolaborasi lebih lanjut,
jangan ragu untuk menghubungi saya melalui kontak di bawah ini:


raonbereknew@gmail.com


Raon Spielberg Berek


+6285774886945 (Phone & WhatsApp)


@raonsb