

SBA LOAN APPROVAL

D-Alchemist

- Omega Delima Munthe (Data Lead)
- Ilham Muhammad Shuhada (Data Analyst)
- Rafael Nicholas Tanaja (Data Engineer)
- Johnny Lim (Data Scientist)
- Ageng Pamungkas (Machine Learning Engineer)



Latar Belakang Masalah

SBA (Small Business Administration) adalah lembaga pemerintah Amerika Serikat yang memberikan dukungan finansial kepada *small business* untuk bisa berkembang, seperti **memberikan jaminan pinjaman**. Terdapat *small business* penerima jaminan pinjaman yang berhasil membayar kembali, namun ada juga *small business* yang gagal membayar pinjaman yang telah dijamin oleh SBA.

Pada dataset diketahui bahwa terdapat **32,3% small business (nasabah)** penerima jaminan pinjaman yang dinyatakan **gagal membayar pinjaman**. Hal ini tentu menyebabkan kerugian finansial pada SBA mencapai **\$5.48 miliar**.

Oleh karena itu, sebagai **Data Scientist di SBA** akan mengatasi permasalahan dengan **membuat model untuk memprediksi calon nasabah** akan membayar lunas atau gagal membayar pinjaman.

Latar Belakang Masalah

Penerapan model tersebut diharapkan dapat mencapai **goals**, yakni **menurunkan persentase gagal bayar** dan **menurunkan total nominal yang dinyatakan gagal bayar**.

Business metric yang dapat mengukur tercapainya **goals**, adalah sebagai berikut:

- **Default Percentage (Persentase gagal bayar)**: mengukur persentase nasabah yang gagal membayar pinjaman.

Dampak terhadap bisnis: semakin rendah *default rate*, maka semakin baik terhadap finansial SBA

- **Charged-off Total (total nominal yang dinyatakan gagal bayar)**: mengukur total uang dari keseluruhan nasabah yang dinyatakan gagal membayar.

Dampak terhadap bisnis: semakin rendah *default rate*, maka semakin baik terhadap finansial SBA.

Exploratory Data Analysis - Descriptive Statistics

Pada tahap ini dilakukan peninjauan terhadap data, seperti tipe data, kolom/fitur, dan persebaran nilai data.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 258966 entries, 0 to 258965
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   LoanNr_ChkDgt    258966 non-null   int64  
 1   Name              258955 non-null   object  
 2   City              258962 non-null   object  
 3   State             258965 non-null   object  
 4   Zip               258966 non-null   int64  
 5   Bank              258791 non-null   object  
 6   BankState          258790 non-null   object  
 7   NAICS             258966 non-null   int64  
 8   ApprovalDate      258966 non-null   int64  
 9   ApprovalFY        258966 non-null   int64  
 10  Term              258966 non-null   int64  
 11  NoEmp             258966 non-null   int64  
 12  NewExist          258966 non-null   int64  
 13  CreateJob         258966 non-null   int64  
 14  RetainedJob       258966 non-null   int64  
 15  FranchiseCode     258966 non-null   int64  
 16  UrbanRural        258966 non-null   int64  
 17  RevLineCr         258930 non-null   object  
 18  LowDoc            258733 non-null   object  
 19  ChgOffDate        85793 non-null    float64 
 20  DisbursementDate  258479 non-null   float64 
 21  DisbursementGross 258966 non-null   object  
 22  BalanceGross       258966 non-null   object  
 23  MIS_Status         258966 non-null   object  
 24  ChgOffPrinGr      258966 non-null   object  
 25  GrAppv            258966 non-null   object  
 26  SBA_Appv          258966 non-null   object  
dtypes: float64(2), int64(12), object(13)
memory usage: 53.3+ MB
```

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

LoanNr_ChkDgt          0
Name                     11
City                      4
State                     1
Zip                       0
Bank                     175
BankState                 176
NAICS                     0
ApprovalDate              0
ApprovalFY                0
Term                      0
NoEmp                     0
NewExist                  0
CreateJob                  0
RetainedJob                0
FranchiseCode              0
UrbanRural                  0
RevLineCr                  36
LowDoc                     233
ChgOffDate                173173
DisbursementDate           487
DisbursementGross            0
BalanceGross                 0
MIS_Status                  0
ChgOffPrinGr                0
GrAppv                     0
SBA_Appv                   0
dtype: int64
```

- Terdapat kolom yang memiliki **tipe data tidak sesuai**, contohnya seperti UrbanRural, NewExist, SBA_Appv, etc.
- Terdapat kolom yang memiliki **missing-value** seperti Name, City, State, Bank, BankState, RevLineCr, LowDoc, ChgOffDate, DisbursementDate.

Exploratory Data Analysis - Descriptive Statistics

df[num].describe()								
	count	mean	std	min	25%	50%	75%	max
ChgOffPrinGr	258966.00	21141.99	76250.87	0.00	0.00	0.00	14606.75	3512596.00
CreateJob	258966.00	23.70	439.76	0.00	0.00	0.00	1.00	8800.00
BalanceGross	258966.00	3.25	1627.03	0.00	0.00	0.00	0.00	827875.00
SBA_Appv	258966.00	101665.68	186904.78	500.00	12500.00	26005.00	96900.00	4000000.00
DisbursementGross	258966.00	152469.37	238123.50	4000.00	30900.75	70893.50	158641.00	6206000.00
Term	258966.00	93.10	69.51	0.00	55.00	84.00	87.00	480.00
GrAppv	258966.00	137714.64	234286.56	1000.00	25000.00	50000.00	135000.00	4000000.00
NoEmp	258966.00	9.09	71.63	0.00	2.00	4.00	8.00	9999.00
RetainedJob	258966.00	26.09	440.22	0.00	0.00	1.00	4.00	9500.00

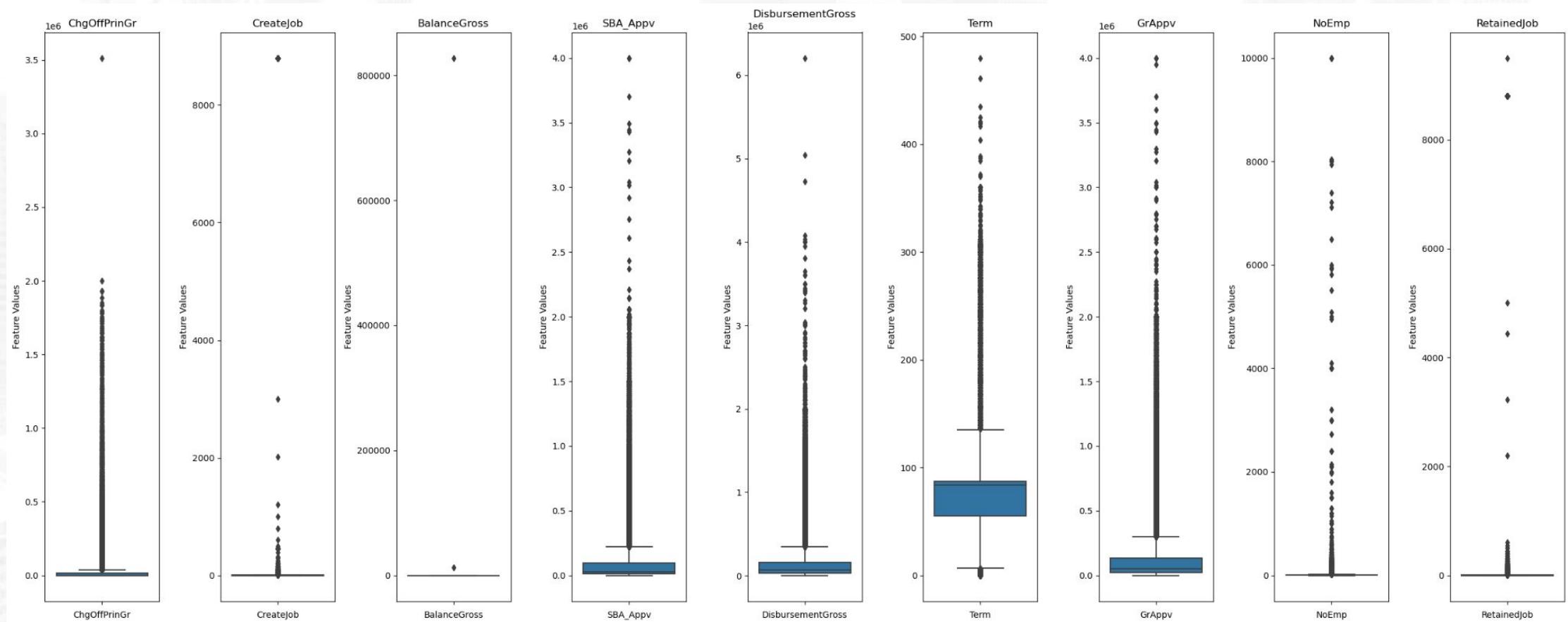
df[cat].describe()				
	count	unique	top	freq
ChgOffDate	85793	5040	13-Mar-10	516
MIS_Status	258966	2	P I F	175428
NAICS	258966	1244	0	48658
LowDoc	258733	7	N	240509
City	258962	19832	LOS ANGELES	3871
BankState	258790	52	NC	38140
DisbursementDate	258479	3120	30-Apr-07	6377
FranchiseCode	258966	1449	1	128766
UrbanRural	258966	3	1	147052
RevLineCr	258930	7	N	112141
ApprovalFY	258966	24	2006	65753
State	258965	51	CA	37290
Name	258955	241588	SUBWAY	145
ApprovalDate	258966	3949	30-Sep-97	465
Zip	258966	22950	90010	384
LoanNr_ChkDgt	258966	258966	1000093009	1
Bank	258791	3667	BANK OF AMERICA NATL ASSOC	38711
NewExist	258966	3	1	176795

Untuk mempermudah saat melakukan EDA, dilakukan pengubahan tipe data pada kolom-kolom yang memiliki tipe data yang tidak sesuai. Dari *descriptive statistics* diperoleh ***anomali summary*** yaitu::

- Selisih nilai yang sangat jauh pada deskripsi data numerik
- **RetainedJob**, nilai **Q3 (75%) sebesar 4** sedangkan nilai **max sebesar 9500**
- **CreateJob**, nilai **Q3 (75%) sebesar 1** sedangkan nilai **max sebesar 8800**
- **NoEmp**, nilai **Q3 (75%) sebesar 8** sedangkan nilai **max sebesar 9999**.
- Kolom yang memiliki *values* yang tidak sesuai
- **Zip**, nilai **min 0** yang mungkin tidak sesuai sebagai kode pos yang valid.

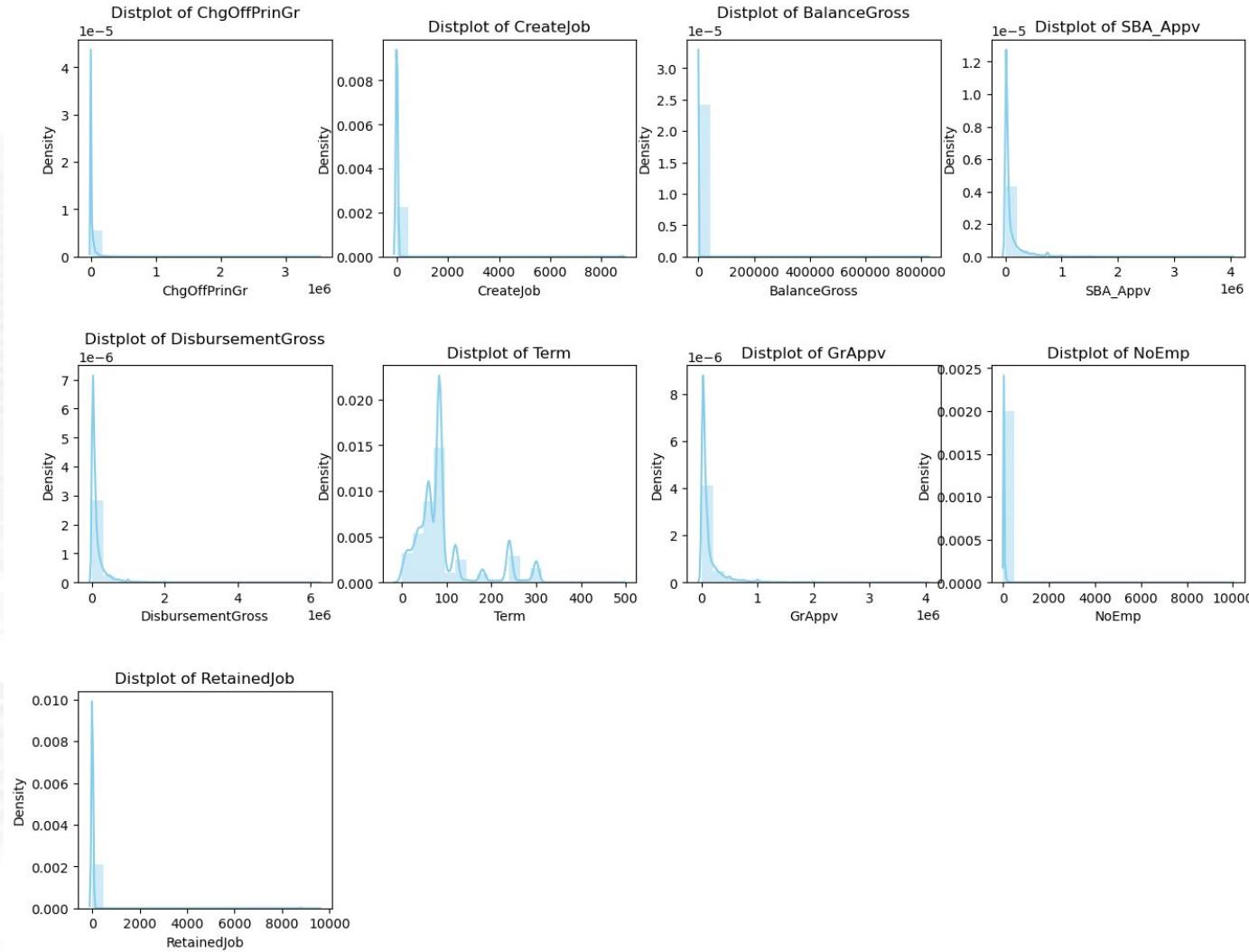
Kolom ChgOffDate, NAICS, City, BankState, DisbursementDate, FranchiseCode, ApprovalFY, State, Name, ApprovalDate, Zip, LoanNr_ChkDgt, dan Bank memiliki **kategori (unique values) terlalu banyak** sehingga **pada data preprocessing dapat melakukan drop kolom atau dilakukan Feature Extraction (membuat derivative feature dari fitur yang ada)**.

Exploratory Data Analysis - Univariate Analysis



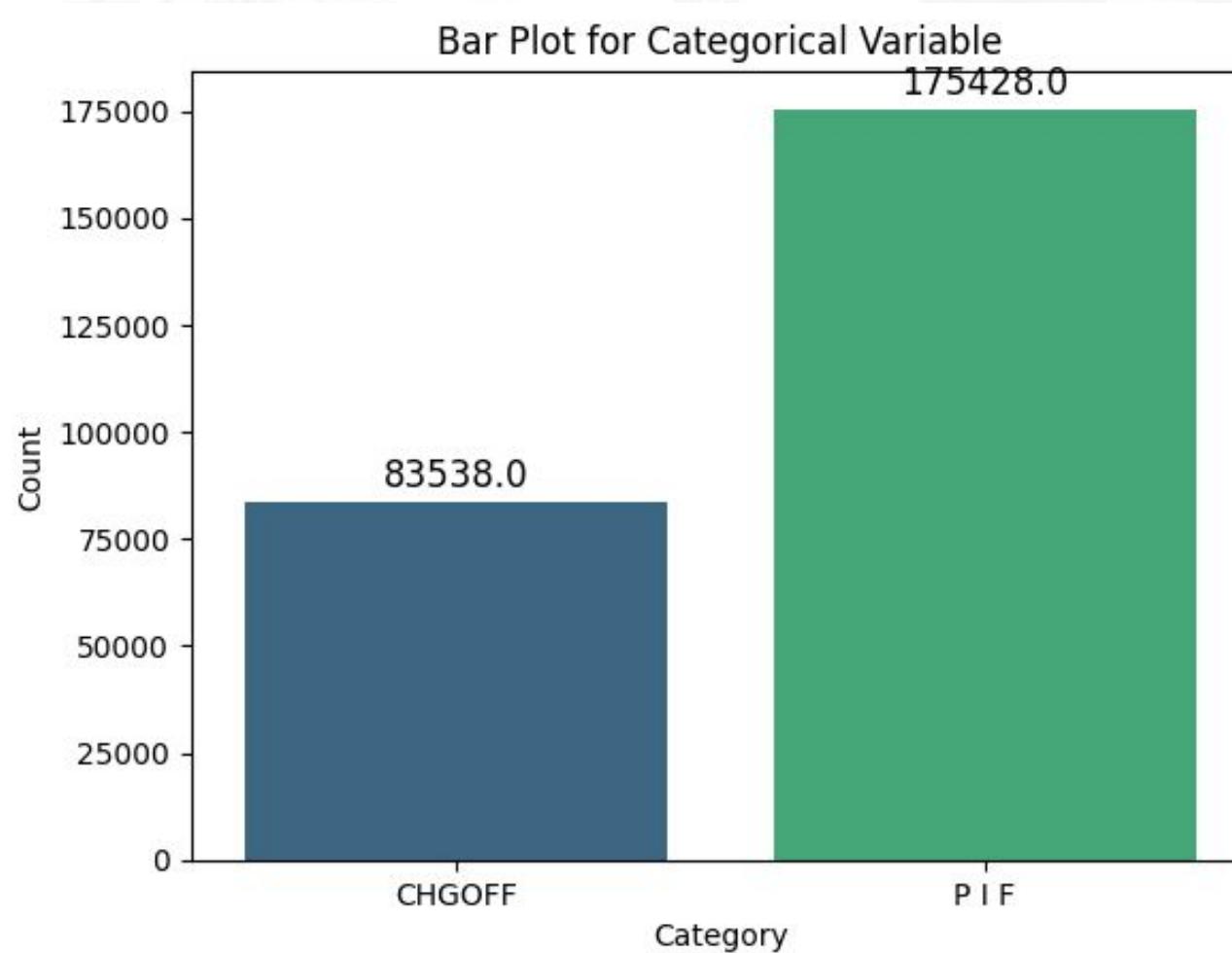
Terlihat ada **banyak outliers** pada kolom-kolom numerik. Oleh karena itu, **pada tahap pre-processing disarankan untuk mengatasi outliers yang terdapat pada data.**

Exploratory Data Analysis - Univariate Analysis



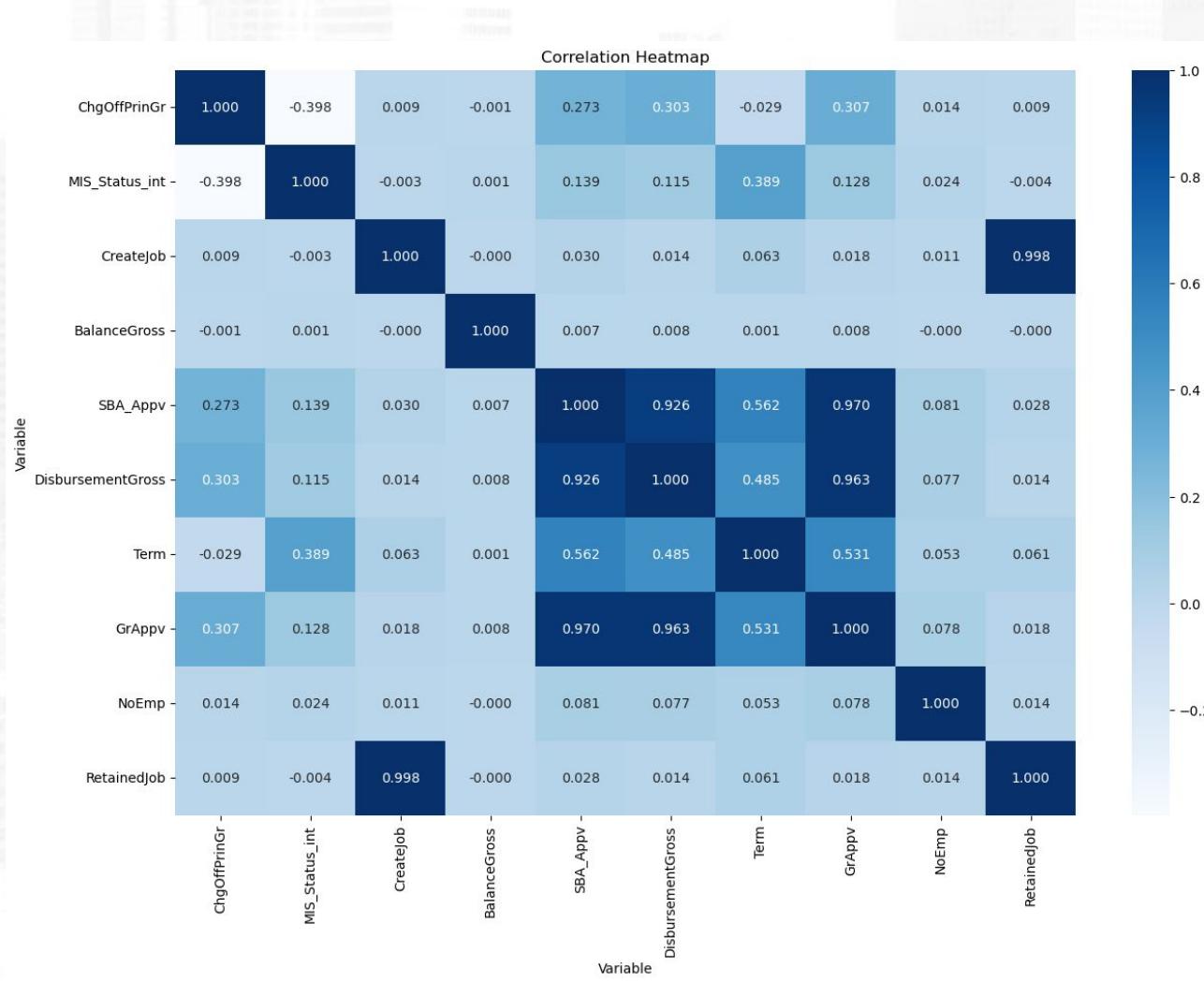
Terlihat bahwa pada **distribusi data pada kolom-kolom numerik cenderung *positively skewed***. Oleh karena itu, pada **preprocessing** disarankan untuk melakukan transformasi data.

Exploratory Data Analysis - Univariate Analysis



Kolom target yaitu **MIS_Status** memiliki **2 kategori** yaitu **CHGOFF** (gagal bayar) dan **PIF** (lunas) dengan proporsi yang tidak seimbang. Oleh karena itu, pada data *preprocessing* disarankan untuk *meng-handle class imbalance*.

Exploratory Data Analysis - Multivariate Analysis

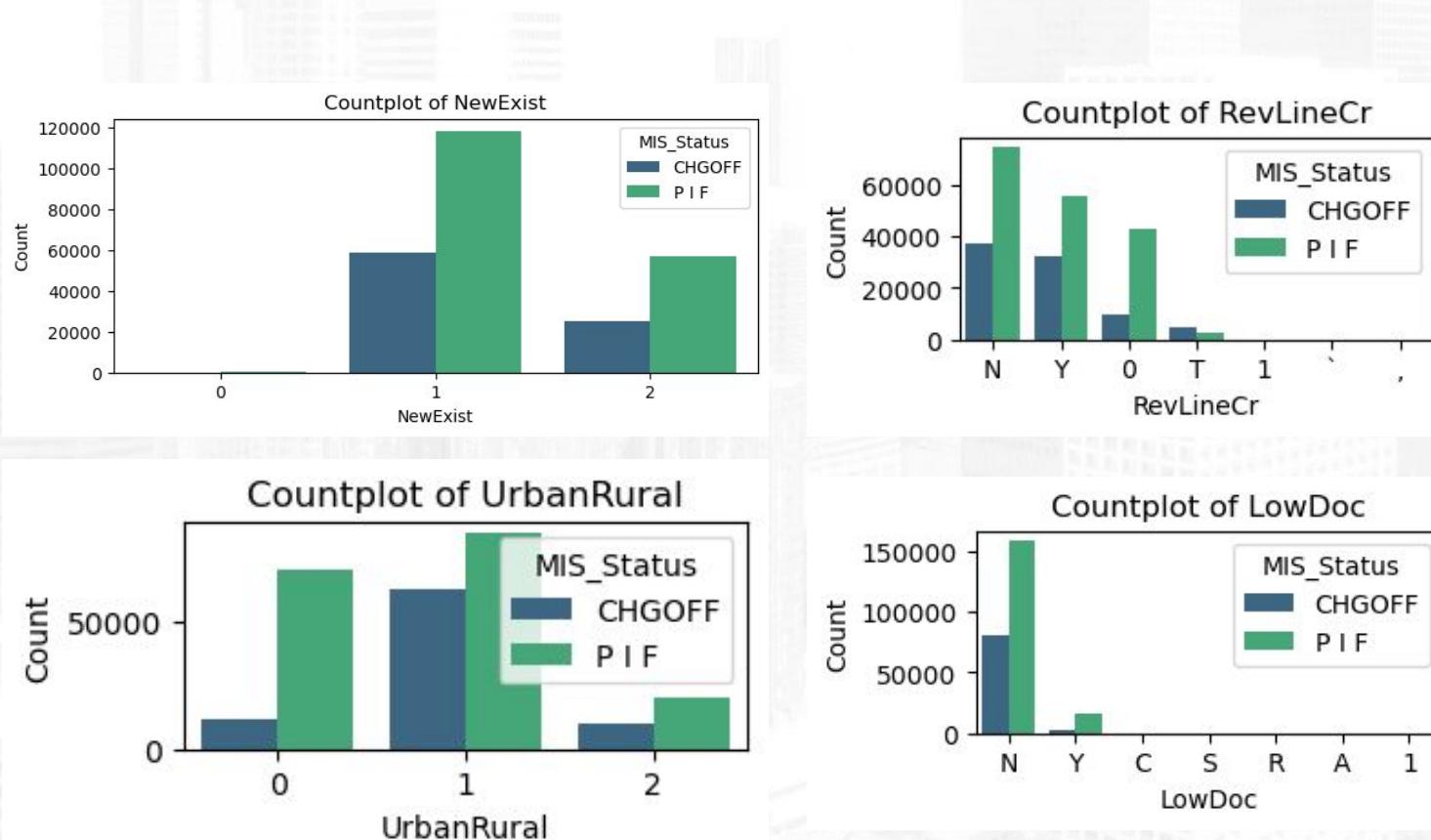


Dari *heatmap correlation*, diperoleh bahwa:

- **Fitur Retained Job** memiliki **korelasi yang tinggi** terhadap **Create Job**.
- **Fitur GrAppv** memiliki **korelasi yang tinggi** terhadap **Disbursement Gross** dan **SBA_Appv**.
- **Fitur Term** terhadap **GRAppv** dan **SBA_Appv** memiliki **korelasi yang positif**. Hal ini menunjukkan bahwa semakin lama jangka pinjaman, jumlah pinjaman kotor yang disetujui bank cenderung semakin besar.

Oleh karena itu, **pada preprocessing disarankan untuk melakukan Feature Extraction (membuat derivative feature)** atau pilih salah satu kolom dari beberapa fitur yang memiliki korelasi kuat (terindikasi redundant).

Exploratory Data Analysis - Multivariate Analysis



Terdapat 4 Feature Categorical yang tidak memiliki *unique values* yang terlalu banyak, yaitu **NewExist**, **RevLineCr**, **UrbanRural**, dan **LowDoc**. Dari plot tersebut diketahui bahwa:

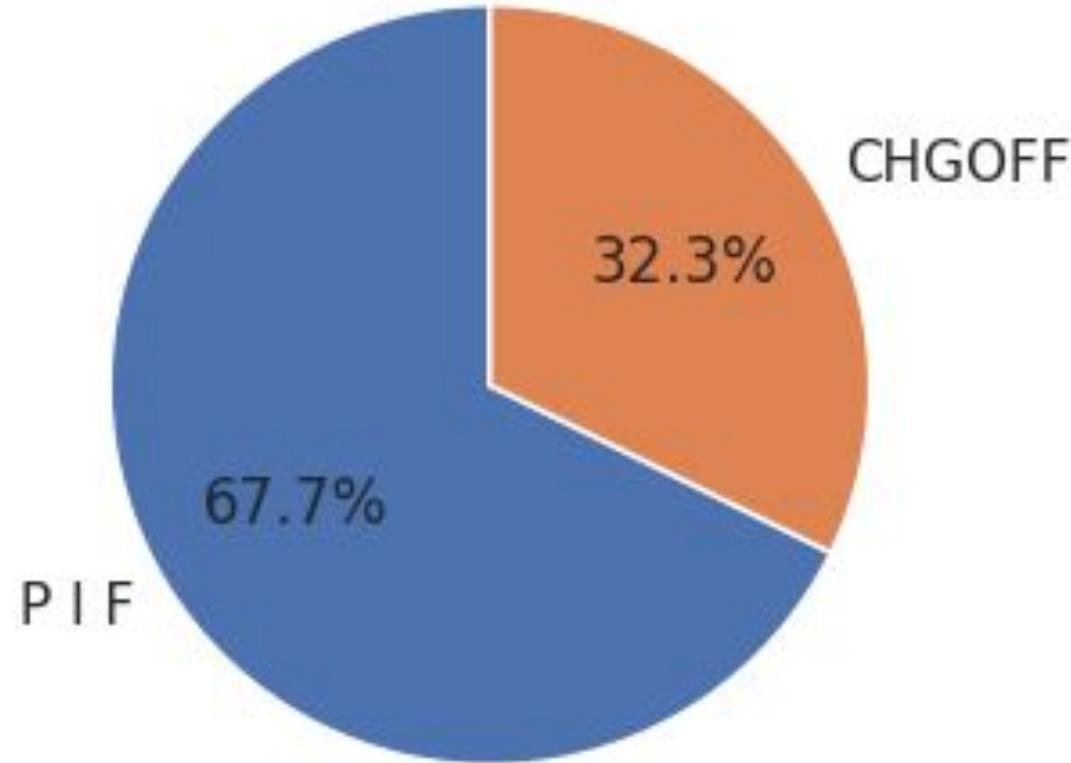
- Fitur **RevLineCr** terdapat **invalid data** seperti 0, T, dan 1.
- Fitur **LowDoc** terdapat **invalid data** seperti C, S, R, A, I meskipun dalam jumlah yang sedikit.
- Fitur **UrbanRural**, terdapat **banyak kategori 0 (Unknown)** sehingga kurang merepresentasikan kolom UrbanRural.

Oleh karena itu, pada **preprocessing** disarankan untuk **merapikan invalid data pada kolom RevLineCr dan LowDoc serta menghapus kolom UrbanRural karena dianggap kurang relevan**.

Selain itu, kolom NewExist, RevLineCr, dan LowDoc terindikasi berasosiasi dengan kolom target (MIS_Status) karena memiliki jangkauan yang cukup luas antara CHGOFF (gagal bayar) dan PIF (lunas). Akan tetapi, **diperlukan analisis statistik yang lebih mendalam untuk memastikan ada atau tidaknya korelasi fitur-fitur tersebut terhadap MIS_Status**

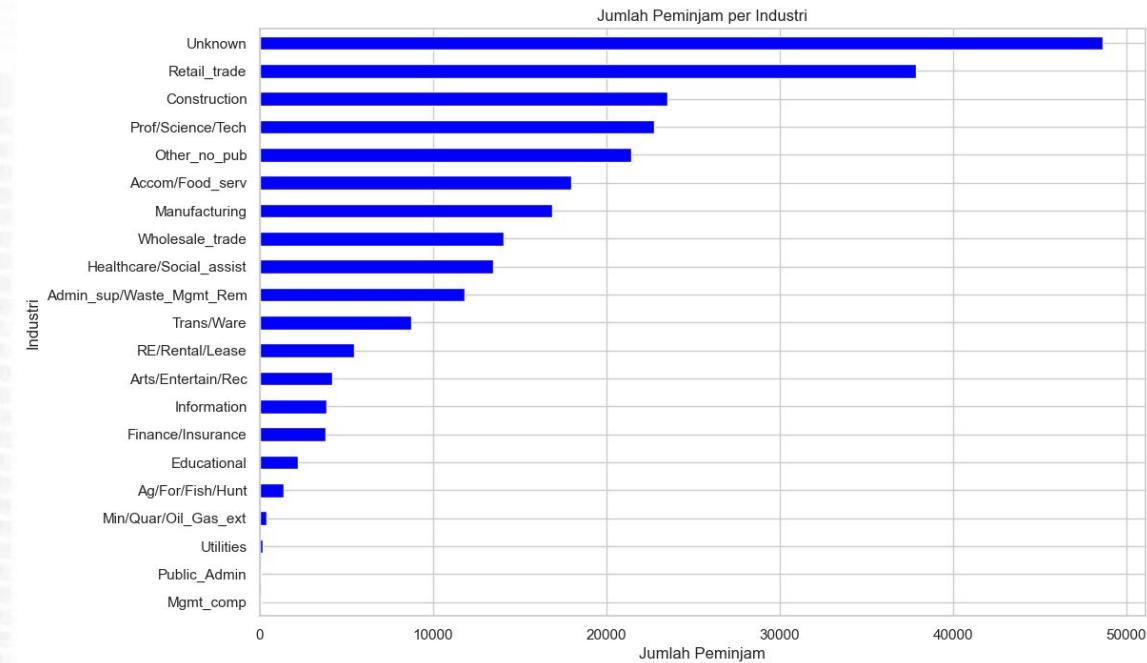
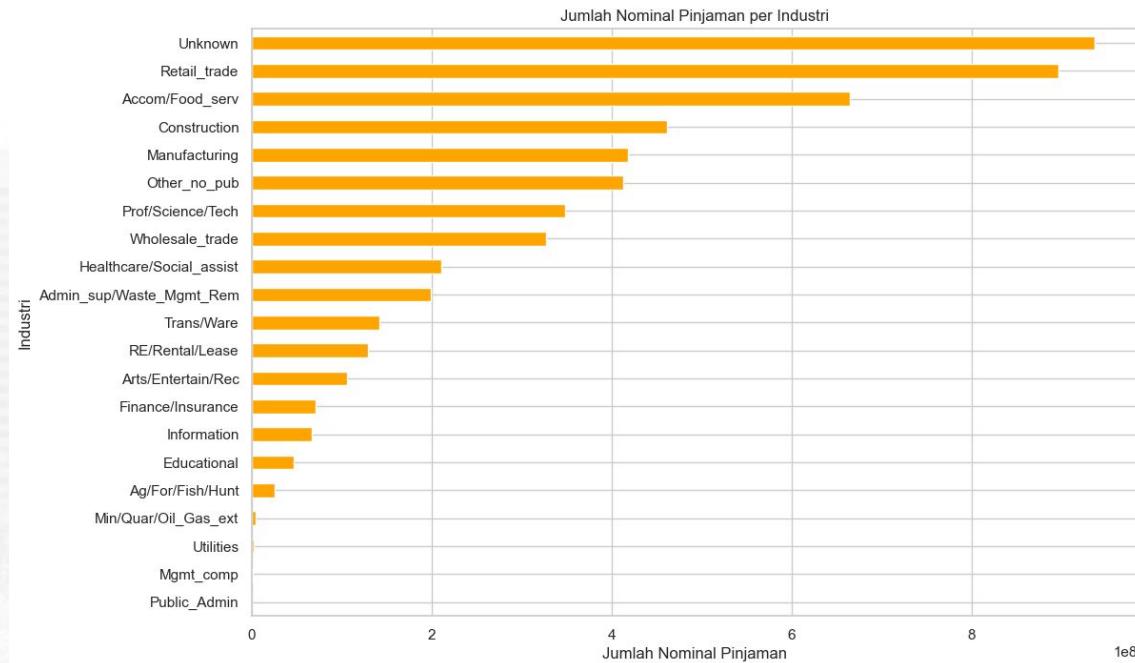
Exploratory Data Analysis - Business Insight

Pie Chart of MIS_Status



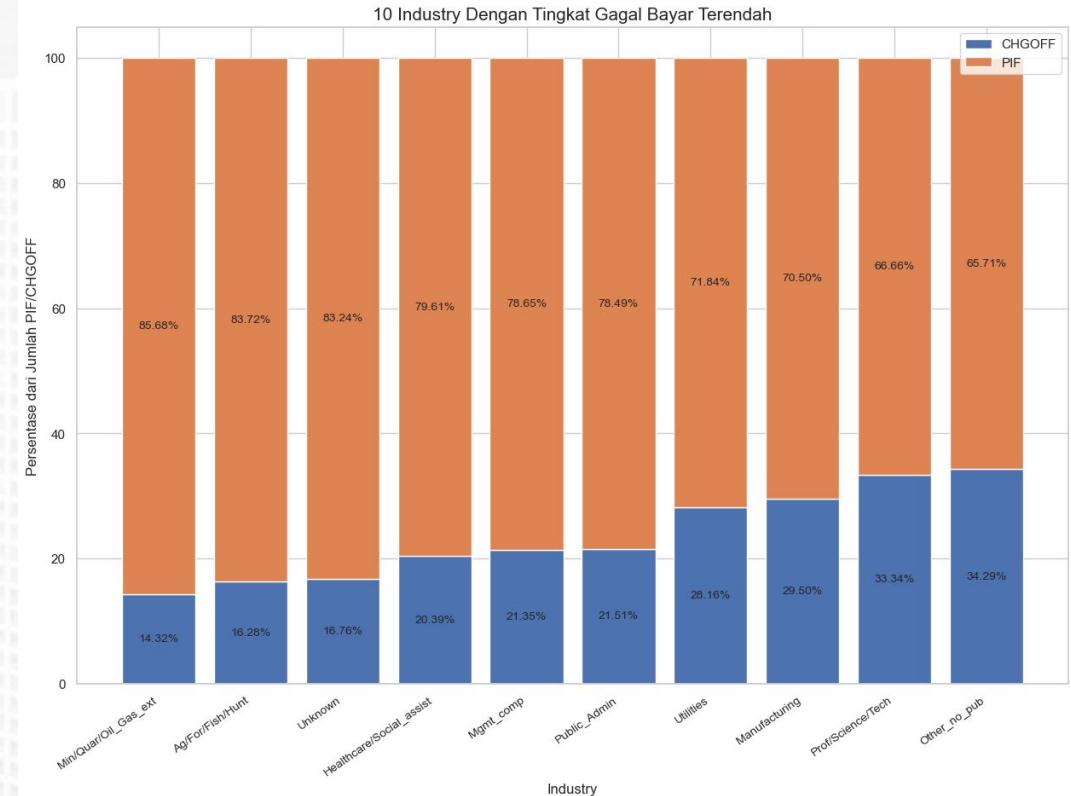
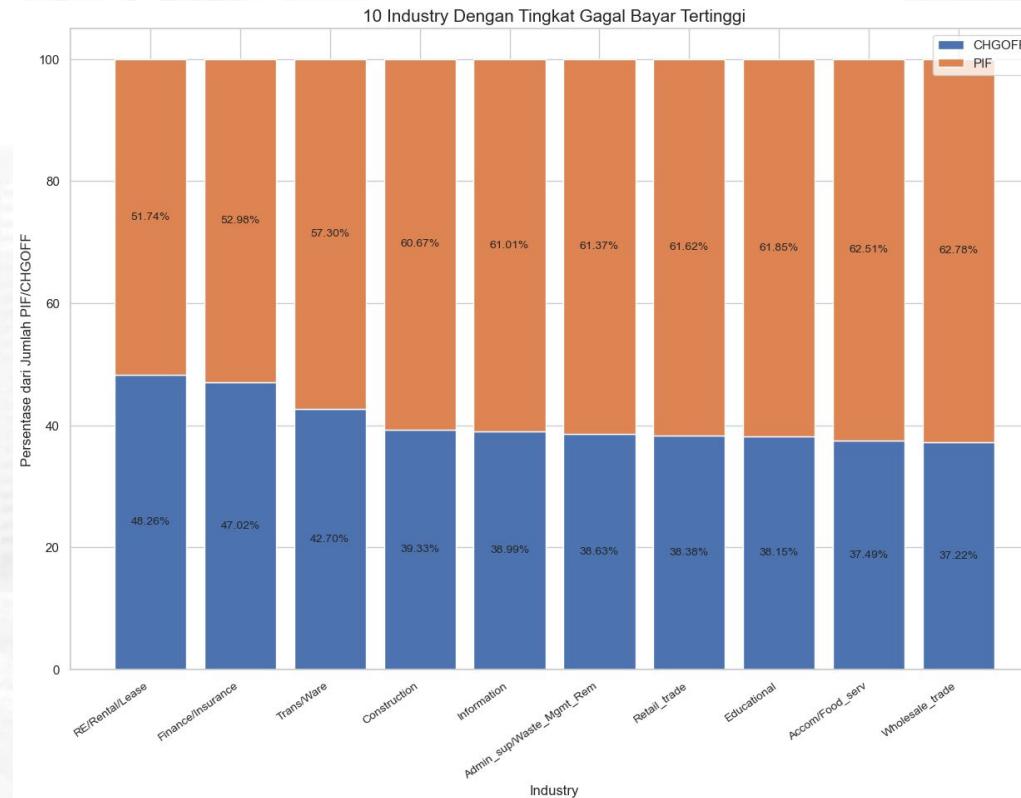
mayoritas *small business* (nasabah) dapat membayar penuh pinjamannya, hanya **32,3%** yang **gagal membayar secara lunas**.

Exploratory Data Analysis - Business Insight



- **Jumlah nominal pinjaman terbesar** dibandingkan industri lainnya yaitu:
 - Unknown
 - Retail trade
 - Accom/Food_serv
- **Public Admin, Mgmt_comp, dan Utilities** memiliki **jumlah nominal pinjaman** dan **jumlah peminjam terkecil** dibandingkan industri lainnya.
- **Jumlah peminjam terbesar** dibandingkan industri lainnya yaitu:
 - Unknown
 - Retail trade
 - Construction

Exploratory Data Analysis - Business Insight



- Tiga Industri dengan **tingkat gagal bayar tertinggi** yaitu:
 - RE/Rental/Lease (48,26%)
 - Finance/Insurance (47,02%)
 - Trans/Ware (42,7%).

- Tiga Industri dengan **tingkat gagal bayar terendah** yaitu:
 - Min/Quar/Oil_Gas_ext (14,32%)
 - Agj/For/Fish/Hunt (16,28%)
 - Unknown (16,76%)

Exploratory Data Analysis - Business Recommendation

- **Target penerima jaminan pinjaman bisa lebih dipromosikan ke industri Public Admin, Mgmt_comp, dan Utilities** karena jumlah peminjam dan total nominal pinjaman yang masih sedikit dengan tingkat gagal bayar yang cukup rendah dibandingkan industri lainnya.
- **Peningkatan Manajemen Risiko di Industri** dengan Tingkat Gagal Bayar Tinggi seperti **RRE/Rental/Lease, Finance/Insurance, dan Trans/Ware.**, SBA harus lebih berhati-hati dalam manajemen risiko pemberian jaminan pinjaman pada industri ini. Ini dapat mencakup peningkatan proses verifikasi, pengembangan strategi pengambilan keputusan yang lebih baik, dan pemantauan aktif pelanggan dalam sektor tersebut.
- **Peningkatan Manajemen Risiko di Industri Retail Trade** karena memiliki nominal pinjaman yang besar dan jumlah pinjaman yang besar tetapi tingkat gagal bayarnya cukup tinggi dibandingkan industri lainnya.

Repository Git

[Link Repository Git D-Alchemist](#)

Preprocessing - Data Cleansing

```
# menghapus feature atau mengisi nilai missing value
df = df.drop(columns=["ChgOffDate"])

df["Name"].fillna("", inplace=True)
df["City"].fillna(df["City"].mode()[0], inplace=True)
df["state"].fillna(df["State"].mode()[0], inplace=True)
df["Bank"].fillna(df["Bank"].mode()[0], inplace=True)
df["BankState"].fillna(df["BankState"].mode()[0], inplace=True)
df["RevLineCr"].fillna(df["RevLineCr"].mode()[0], inplace=True)
df["LowDoc"].fillna(df["LowDoc"].mode()[0], inplace=True)
df["DisbursementDate"].fillna(df["DisbursementDate"].mode()[0], inplace=True)

df.isna().sum()

LoanMr_ChkDgt      0
Name                 0
City                 0
State                0
Zip                  0
Bank                 0
BankState             0
NAICS                0
ApprovalDate         0
ApprovalFY            0
Term                  0
NoEmp                0
NewExist              0
CreateJob              0
RetainedJob            0
FranchiseCode          0
UrbanRural             0
RevLineCr              0
LowDoc                0
DisbursementDate        0
DisbursementGross        0
BalanceGross            0
MIS_Status              0
ChgOffPrinGr            0
GrAppv                0
SBA_Appv                0
MIS_Status_int            0
Industry                0
dtype: int64
```

Handle Missing Value

- Fitur yang memiliki missing values adalah kolom categorical sehingga **handle missing values dilakukan dengan teknik imputasi** yaitu mengisinya dengan modus.
- Namun, untuk Fitur ChgOffDate memiliki **missing values yang sangat banyak** karena fitur tersebut akan berisi tanggal ketika nasabah dinyatakan gagal bayar. Tentu saja hal ini sepenuhnya telah dijelaskan pada kolom target yaitu MIS_Status **sehingga Fitur ChgOffDate harus dihapus.**

Preprocessing - Data Cleansing

Handle Invalid Data

- Pada kolom **RevLineCr**, datanya dirapikan yaitu mengubah **N** menjadi **No** dan **Y** menjadi **Yes**. Kemudian, baris yang berisi selain **N** dan **Y** dirapikan dengan mengubah **Ø** dan **T** menjadi **No**, **1** menjadi **Yes**, dan jika masih ada invalid data lainnya maka baris tersebut dihapus.
- Pada kolom **LowDoc**, datanya dirapikan yaitu mengubah **N** menjadi **No** dan **Y** menjadi **Yes**. Kemudian, baris yang berisi selain **N** dan **Y** dihapus

```
[ ] # Merapikan values
replacement_dict = {
    'N': 'No',
    'Y': 'Yes',
    'Ø': 'No',
    'T': 'No',
    '1': 'Yes'
}

df['RevLineCr'] = df['RevLineCr'].replace(replacement_dict)

# Menghapus nilai selain 'Yes' dan 'No'
df = df[df['RevLineCr'].isin(['Yes', 'No'])]

# Merapikan values
replacement_dict = {
    'N': 'No',
    'Y': 'Yes'
}

df['LowDoc'] = df['LowDoc'].replace(replacement_dict)

# Menghapus nilai selain 'Yes' dan 'No'
df = df[df['LowDoc'].isin(['Yes', 'No'])]
```

```
[ ] # mengecek duplicate
df.duplicated().sum()
```

Handle Duplicated Data

Tidak ada data duplikat.

Preprocessing - Data Cleansing

Handle Outlier

Menggunakan **Z-Score** untuk mendeteksi dan menghapus *outlier*.

Data yang terhapus: 2,76%

```
[ ] # handling outliers
from scipy import stats

print(f'Jumlah baris sebelum memfilter outlier: {len(df)}')

filtered_entries = np.array([True]*len(df))
for col in ['Term', 'NoEmp', 'CreateJob', 'RetainedJob',
            'DisbursementGross', 'ChgOffPrinGr', 'GrAppv',
            'SBA_Appv']:
    z_scores = abs(stats.zscore(df[col]))
    filtered_entries = (z_scores < 3)

df = df[filtered_entries]

print(f'Jumlah baris setelah memfilter outlier: {len(df)}')

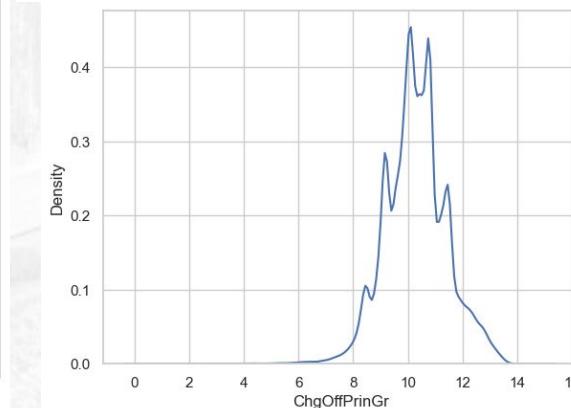
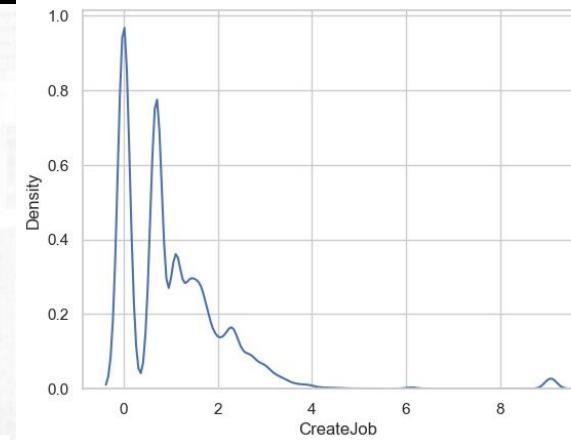
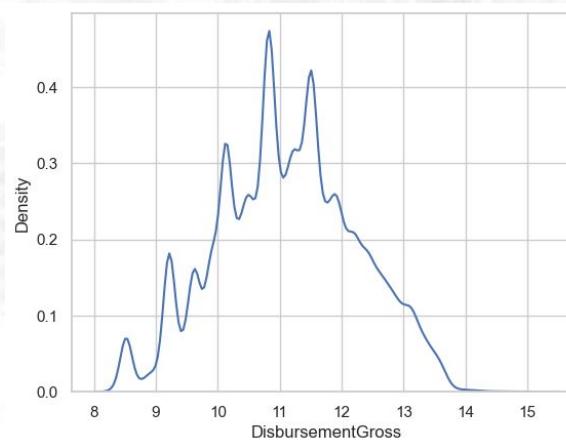
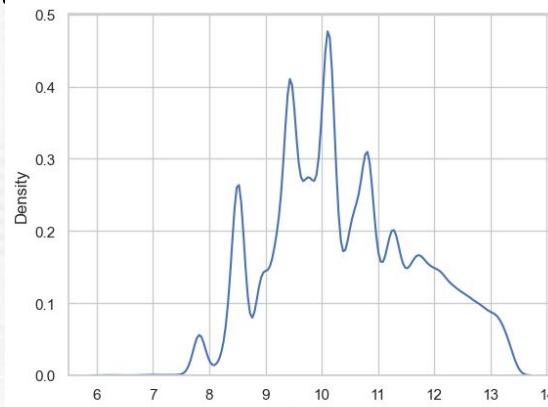
Jumlah baris sebelum memfilter outlier: 258729
Jumlah baris setelah memfilter outlier: 251585
```

Alasan tidak menggunakan IQR karena akan menghapus setengah dari jumlah baris dataset.

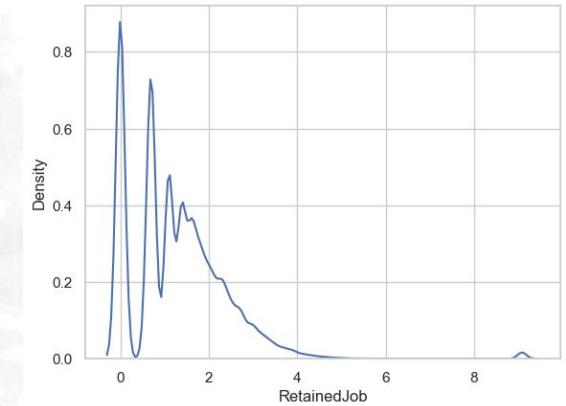
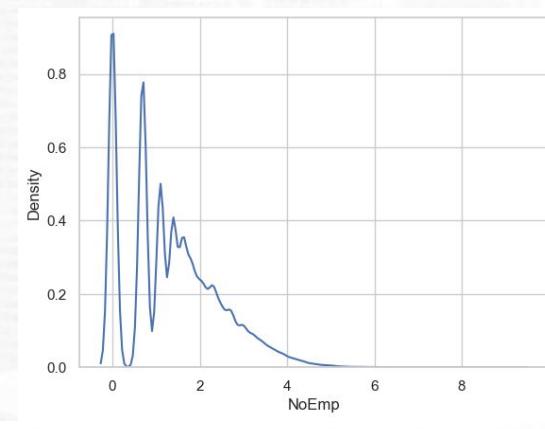
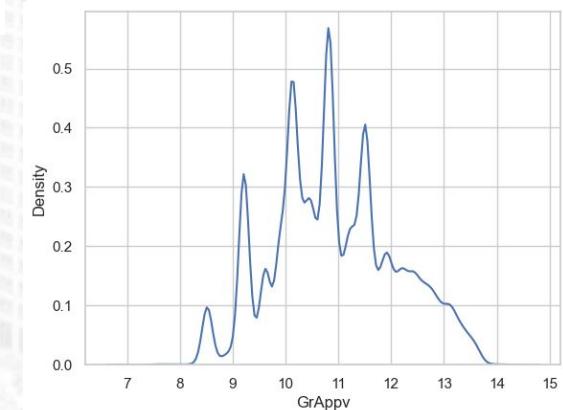
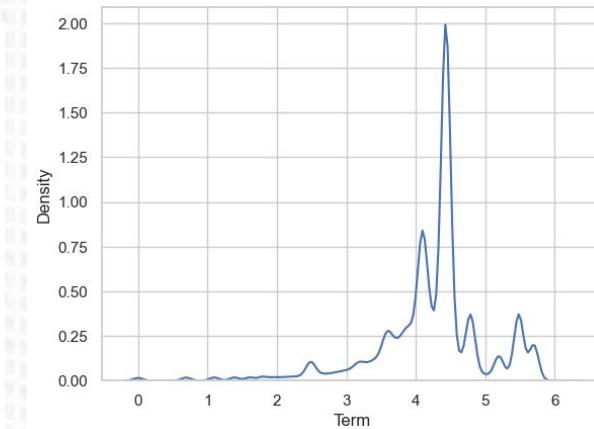
Preprocessing - Data Cleansing

Feature Transformation

Semua fitur yang memiliki tipe data numeric dilakukan Feature Transformation menggunakan *log transformation*.



Beberapa fitur yang memiliki **distribusi normal setelah dilakukan Feature Transformation**, yaitu **SBA_Appv**, **DisbursementGross**, **ChgOffPrinGr**, dan **GrAppv**.



Preprocessing - Feature Engineering

Feature Extraction

Membuat fitur baru, seperti:

- **term_category**, diperoleh dari kolom **Term** (**jangka waktu pinjaman**) yang dikelompokkan menjadi 4 kategori yaitu low (<58 bulan), medium (58-61 bulan), high (62-81 bulan), dan very high (≥ 82 bulan). Artinya, fitur term_category merupakan kategori jangka waktu pinjaman.
- **Job_Stability**, Feature Derivative dari RetainedJob dan CreateJob yaitu akan bernilai 1 jika RetainedJob > CreateJob. Artinya Feature Job_Stability merupakan indikator sejauh mana pekerjaan yang dipertahankan (RetainedJob) lebih besar daripada pekerjaan yang dibuat (CreateJob) pada *small business* yang menjadi nasabah SBA.
- **CompanySize**, Feature Derivative dari NoEmp (**Jumlah karyawan**) yang akan bernilai 0 jika NoEmp < 25, 1 jika NoEmp berkisar 25-100, dan 2 jika NoEmp ≥ 100 . Artinya Feature CompanySize merepresentasikan ukuran perusahaan berdasarkan jumlah karyawan.
- **Recession**, Feature Derivative dari DisbursementDate yaitu akan bernilai 1 jika DisbursementDate berada pada rentang resesi (01 Desember 2007 - 30 Juni 2009) dan jika tidak maka 0. Artinya, Feature Recession merepresentasikan waktu pinjaman berada pada saat resesi atau tidak.
- **Industry**, yaitu feature yang mendefinisikan industry peminjam berdasarkan feature **North American Industry Classification System (NAICS)**.

Preprocessing - Data Cleansing dan Feature Scaling

Feature Encoding

Melakukan Feature Encoding terhadap kolom categorical seperti **FranchiseCode**, **term_category**, **RevLineCr**, **LowDoc**, **Industry**, **NewExist**

Standardisasi

Menerapkan standardisasi terhadap kolom numerik yang telah dilakukan *log transformation*.

Preprocessing - Feature Engineering

Feature Selection

- Berdasarkan EDA, Fitur GrAppv memiliki korelasi yang tinggi dengan DisbursementGross dan SBA_Appv (terindikasi redundant) sehingga kami hanya memilih fitur GrAppv. Fitur tersebut telah kami lakukan *log transformation* dan standardisasi. Kemudian, kami mengecek hubungan Feature log_GrAppv_std terhadap MIS_Status_int (numerik vs categorical) dengan menggunakan t-test.

```
[ ] from scipy.stats import ttest_ind

# Mengambil data untuk dua kelompok
group1 = df[df['MIS_Status_int'] == 0]['log_GrAppv_std']
group2 = df[df['MIS_Status_int'] == 1]['log_GrAppv_std']

# Melakukan t-test
t_statistic, p_value = ttest_ind(group1, group2)

# Menampilkan hasil
print("T-Statistic:", t_statistic)
print("P-Value:", p_value)

# Membuat keputusan berdasarkan nilai p-value
alpha = 0.05
if p_value < alpha:
    print("Terdapat perbedaan signifikan antara kedua kelompok.")
else:
    print("Tidak terdapat perbedaan signifikan antara kedua kelompok.")

T-Statistic: -74.21588051562921
P-Value: 0.0
Terdapat perbedaan signifikan antara kedua kelompok.
```

Diperoleh bahwa **p-value < 0,05** sehingga bisa dikatakan bahwa terdapat perbedaan signifikan antara kedua kelompok yang diujikan yang berarti **Fitur log_GrAppv_std memiliki pengaruh statistik signifikan terhadap variabel target**. Oleh karena itu, kami memilih Fitur log_GrAppv_std sebagai salah satu input pada model *machine learning* kami.

Untuk kolom numerik lainnya (**Term, RetainedJob, CreateJob, dan NoEmp**) tidak dipilih karena telah diekstrak menjadi **derivative feature** pada tahapan Feature Extraction.

Sementara untuk **fitur ChgOffPrinGr juga tidak dipilih karena** berisi 0 jika pinjaman dinyatakan lunas dan bernilai bukan 0 ketika pinjaman dinyatakan gagal bayar. Hal ini berarti fitur tersebut **merepresentasikan kolom target yaitu MIS Status**

Preprocessing - Feature Engineering

Feature Selection

- Pemilihan **Feature Categorical** yang akan digunakan sebagai input model *machine learning* kami lakukan berdasarkan **Chi Square Test** untuk melihat hubungan antara Feature Categorical terhadap MIS_Status_int (categorical vs categorical).

Nilai P-Value:	NewExist_encoded	RevLineCr_encoded	LowDoc_encoded
NewExist_encoded	0.00	0.00	0.00
RevLineCr_encoded	0.00	0.00	0.00
LowDoc_encoded	0.00	0.00	0.00
MIS_Status_int	0.00	0.00	0.00
Industry_encoded	0.00	0.00	0.00
Job_Stability	0.00	0.00	0.00
CompanySize	0.00	0.00	0.00
Franchise	0.00	0.00	0.00
Recession	0.00	0.00	0.00
term_category_encoded	0.00	0.00	0.00

MIS_Status_int	Industry_encoded	Job_Stability	
NewExist_encoded	0.00	0.00	0.00
RevLineCr_encoded	0.00	0.00	0.00
LowDoc_encoded	0.00	0.00	0.00
MIS_Status_int	0.00	0.00	0.00
Industry_encoded	0.00	0.00	0.00
Job_Stability	0.00	0.00	0.00
CompanySize	0.00	0.00	0.00
Franchise	0.00	0.00	0.00
Recession	0.00	0.00	0.00
term_category_encoded	0.00	0.00	0.00

CompanySize	Franchise	Recession	term_category_encoded	
NewExist_encoded	0.00	0.00	0.00	0.00
RevLineCr_encoded	0.00	0.00	0.00	0.00
LowDoc_encoded	0.00	0.00	0.00	0.00
MIS_Status_int	0.00	0.00	0.00	0.00
Industry_encoded	0.00	0.00	0.00	0.00
Job_Stability	0.00	0.00	0.00	0.00
CompanySize	0.00	0.00	0.00	0.00
Franchise	0.00	0.00	0.00	0.00
Recession	0.00	0.00	0.00	0.00
term_category_encoded	0.00	0.00	0.00	0.00

Diperoleh bahwa **p-value < 0,05** sehingga bisa dikatakan bahwa terdapat perbedaan signifikan antara semua kolom *categorical* terhadap kolom MIS_Status_int yang **berarti semua Feature Categorical yang ada memiliki pengaruh statistik signifikan terhadap variabel target**. Oleh karena itu, kami juga memilih semua Feature Categorical untuk pada model *machine learning* kami.

Preprocessing - Feature Engineering

Feature Selection

Jadi, terdapat 11 fitur (termasuk MIS Status sebagai fitur target) yang menjadi input model *machine learning* kami.

	NewExist_encoded	RevLineCr_encoded	LowDoc_encoded	Industry_encoded	Job_Stability	CompanySize	Franchise	Recession	term_category_encoded	MIS_Status_int	log_GrAppv_std
0	2	0	0	18	0	1	0	0	0	0	2.05
1	1	0	1	18	0	0	0	0	3	0	-0.14
2	2	1	0	18	0	0	0	0	3	0	1.11
4	1	0	0	3	0	0	0	0	0	0	-0.68
5	1	0	0	18	0	0	0	0	0	0	-0.88
...
258961	2	0	0	16	0	0	0	1	0	1	-1.47
258962	1	0	0	16	0	0	0	1	3	1	1.73
258963	1	0	0	1	1	0	0	1	3	1	1.90
258964	1	0	0	16	1	0	0	1	3	1	1.38
258965	1	0	0	9	1	1	0	1	3	1	2.26

251585 rows x 11 columns

- New Exist: Bisnis baru/bisnis yang sudah lama didirikan
- Rev Line Cr: Revolving Line of Credit
- Low Doc: Low Documentation
- Industry: kategori sektor bisnis *small business*
- Job Stability: indikator sejauh mana pekerjaan yang dipertahankan lebih besar daripada pekerjaan yang dibuat

- CompanySize: Kategori ukuran small business berdasarkan jumlah karyawan
- Franchise: Bisnis *franchise*/bukan
- Recession: Meminjam saat resesi/tidak
- Term Category: Kategori jangka waktu pinjaman
- GrAppv: Nominal pinjaman ke bank
- MIS Status: Kolom target yang menyatakan lunas atau gagal bayar

Preprocessing - Data Cleansing

Handle Class Imbalance

Seperti yang telah diketahui melalui EDA, Kolom target yaitu **MIS_Status** memiliki 2 kategori yaitu 0 (gagal bayar) dan 1 (lunas) dengan proporsi yang tidak seimbang yang dapat mempengaruhi kinerja model sehingga cenderung membuat model memprediksi kelas mayoritas. Oleh karena itu, dilakukan Handle Class Imbalance dengan teknik *oversampling* SMOTE dengan perbandingan 50:50 pada Data Training.

```
[ ] # melihat jumlah data yang bernilai 1 dan 0 di feature target MIS_Status_int
print(selected_features['MIS_Status_int'].value_counts())
1    169059
0     82526
Name: MIS_Status_int, dtype: int64

[ ] # memisahkan feature target
X = selected_features.drop(columns=['MIS_Status_int'])
y = selected_features['MIS_Status_int']

[ ] # melakukan split data training dan data testing
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split

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

# menggunakan oversampling data
sm = SMOTE(random_state=42)
X_resampled, y_resampled = sm.fit_resample(X_train, y_train)

print(pd.Series(y_resampled).value_counts())
0    135210
1    135210
Name: MIS_Status_int, dtype: int64
```

Repository Git

[Link Github](#)

Modelling

Data Train & Data Test

Data train yang digunakan saat pemodelan adalah `x_resampled` dan `y_resampled` yang merupakan hasil dari proses *handle class imbalance* yang telah dilakukan.

```
[ ] print("Total data dalam X_train:", X_train.shape[0])
    print("Total data dalam X_test:", X_test.shape[0])
    print("Total data dalam y_train:", y_train.shape[0])
    print("Total data dalam y_test:", y_test.shape[0])
    print("Total data dalam X_resampled:", X_resampled.shape[0])
    print("Total data dalam y_resampled:", y_resampled.shape[0])

Total data dalam X_train: 201268
Total data dalam X_test: 50317
Total data dalam y_train: 201268
Total data dalam y_test: 50317
Total data dalam X_resampled: 270420
Total data dalam y_resampled: 270420
```

Modelling

Evaluation Metrics

- **Precision as Primary Metric**

Memilih “Precision” sebagai *primary metric evaluation* sesuai konteks bisnis dari dataset: lebih baik fokus untuk mereduksi **False Positive** (nasabah yang diprediksi akan membayar pinjaman secara lunas, namun kenyataannya gagal bayar).

- **Accuracy as Secondary Metric**

“Accuracy” sebagai metrik sekunder dapat memberikan pemahaman keseluruhan tentang seberapa baik model bekerja pada seluruh dataset. “Accuracy” memberikan gambaran umum tentang sejauh mana model benar-benar memprediksi dengan benar, baik **True Positive** maupun **True Negative**.

Modelling

Logistic Regression

Logistic Regression digunakan karena biasanya cocok untuk masalah klasifikasi dan memiliki interpretasi yang baik. Hyperparameter dipilih untuk memperoleh model yang optimal dengan menghindari *overfitting* dan *underfitting*.

Hyperparameter yang digunakan:

- C
- Penalty

```
Train Precision: 0.79
Train Accuracy: 0.8
Test Precision: 0.88
Test Accuracy: 0.8
```

(best hyperparameter diperoleh melalui *grid search*)

```
Best Hyperparameters: {'C': 10, 'penalty': 'l2'}
Best Model: LogisticRegression(C=10)
```

TUNING HYPERPARAMETER

```
Train Precision: 0.79
Train Accuracy: 0.8
Test Precision: 0.88
Test Accuracy: 0.8
```

Namun, hasil yang diperoleh menunjukkan bahwa **model Logistic Regression bukan merupakan model terbaik karena *Train Precision < Test Precision*** (baik itu sebelum maupun sesudah tuning hyperparameter) yang bisa mengindikasikan bahwa **model Logistic Regression mungkin belum cukup kompleks atau belum menangkap pola yang ada dalam Data Training**. Selain itu, **majoritas fitur yang digunakan adalah Feature Categorical sehingga memungkinkan bahwa hubungan fitur-fitur tersebut dengan kolom target adalah *non-linear***. Oleh karena itu, **Logistic Regression mungkin kurang cocok untuk digunakan**.

Modelling

K-Nearest Neighbor

KNN digunakan karena kemampuannya menangani data *non-linear* dan fleksibilitas dalam menyesuaikan dengan pola yang kompleks. Hyperparameter dipilih untuk memperoleh model yang optimal dengan menghindari *overfitting* dan *underfitting*.

Hyperparameter yang digunakan:

- n_neighbors
- p
- weight

(best hyperparameter diperoleh melalui *grid search*)

```
Train Precision: 0.87
Train Accuracy: 0.88
Test Precision: 0.9
Test Accuracy: 0.85
```

```
Best Hyperparameters: {'n_neighbors': 9, 'p': 2, 'weights': 'uniform'}
Best Model: KNeighborsClassifier(n_neighbors=9)
```

TUNING HYPERPARAMETER

```
Train Precision: 0.87
Train Accuracy: 0.88
Test Precision: 0.91
Test Accuracy: 0.86
```

Namun, hasil yang diperoleh menunjukkan bahwa model KNN bukan merupakan model terbaik karena **Train Precision < Test Precision** (baik itu sebelum maupun sesudah Tuning Hyperparameter) yang bisa mengindikasikan bahwa **model KNN mungkin belum cukup kompleks atau belum menangkap pola yang ada dalam Data Training**.

Modelling

Decision Tree

Decision Tree digunakan karena kemampuannya untuk menangani klasifikasi *non-linear* dan mudah diinterpretasikan. Hyperparameter dipilih untuk mengontrol kompleksitas pohon dan mencegah *overfitting*.

Hyperparameter yang digunakan:

- Criterion
- Max Depth
- Min Samples Leaf
- Min Samples Split

```
Train Precision: 0.94
Train Accuracy: 0.93
Test Precision: 0.91
Test Accuracy: 0.83
```

(best hyperparameter diperoleh melalui *grid search*)

```
Best Hyperparameters: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}
Best Model: DecisionTreeClassifier(min_samples_leaf=2)
```

TUNING HYPERPARAMETER

```
Train Precision: 0.93
Train Accuracy: 0.91
Test Precision: 0.91
Test Accuracy: 0.83
```

Hasil yang diperoleh menunjukkan bahwa **model Decision Tree tidak terindikasi overfitting ataupun underfitting**. **Tuning Hyperparameter** yang dilakukan juga **berhasil memperkecil gap antara data train dan data test**. Oleh karena itu, kami memilih model Decision Tree yang sudah dilakukan tuning hyperparameter.

Modelling

Before Tuning Hyperparameter

	Model	Train Precision	Test Precision	Train Accuracy	Test Accuracy
0	Logistic Regression	0.79	0.88	0.80	0.80
1	K-Nearest Neighbor	0.87	0.90	0.88	0.85
2	Decision Tree	0.94	0.91	0.93	0.83

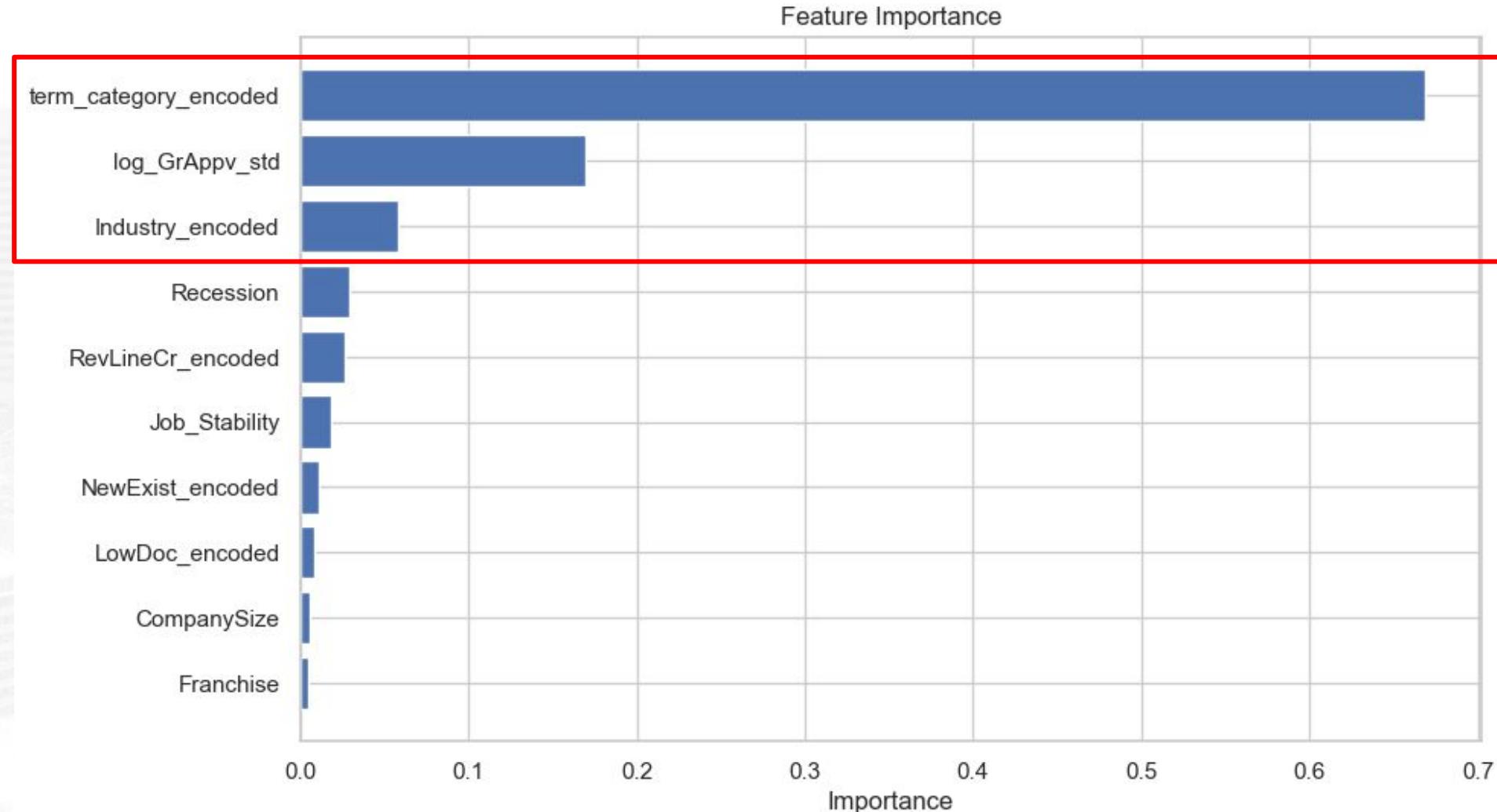
After Tuning Hyperparameter

	Model	Train Precision	Test Precision	Train Accuracy	Test Accuracy
0	Logistic Regression	0.79	0.88	0.80	0.80
1	K-Nearest Neighbor	0.87	0.91	0.88	0.86
2	Decision Tree	0.93	0.91	0.91	0.83

- Decision Tree:
- Memiliki **score Data Train dan Data Test** (baik itu Precision maupun Accuracy) yang **lebih tinggi dibandingkan model lainnya**.
 - Setelah tuning hyperparameter: **Gap antara Data Train dan Data Test rendah**

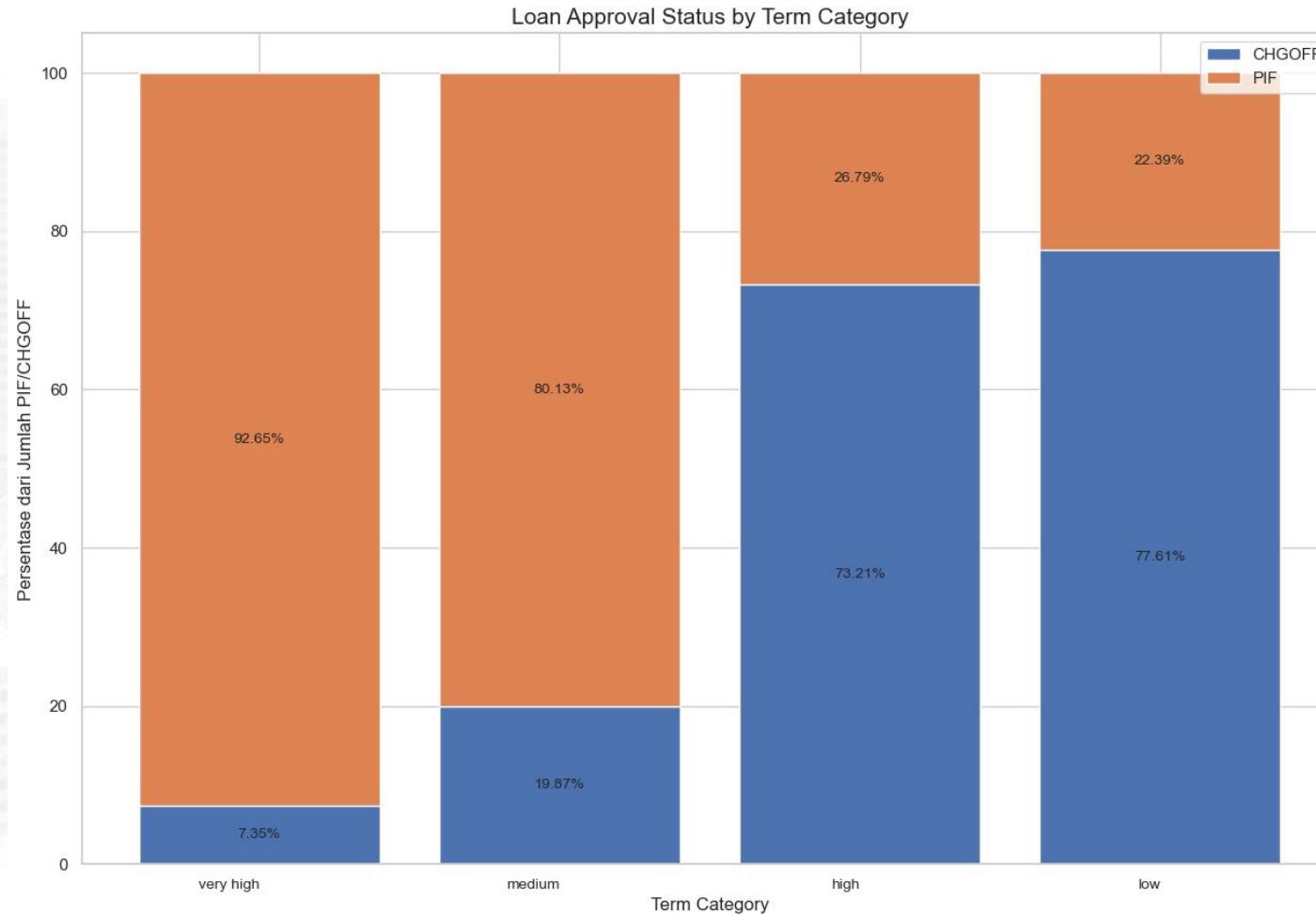
Model yang kami pilih adalah Decision Tree yang sudah dilakukan Tuning Hyperparameter karena model tersebut terindikasi tidak *underfitting* dan tidak *overfitting*, gap antara Data Train dan Data Test rendah.

Modelling - Feature Importance



Pada saat EDA, telah diperoleh rekomendasi bisnis mengenai Industri dan GrAppv (nominal pinjaman).

Modelling - Business Recommendation



Bank dapat fokus memberikan jaminan pinjaman jangka panjang kepada Term peminjaman yang memiliki jangka waktu yang lama atau kategori very high (≥ 82 bulan), karena banyak nasabah yang berhasil melunasi pinjaman pada jangka waktu very high (≥ 82 bulan).

Modelling - Business Recommendation

- **Promosi ke Industri dengan Tingkat Gagal Bayar Rendah:** Fokuskan promosi penerima jaminan pinjaman ke **industri Public Admin, Mgmt_comp, dan Utilities** karena memiliki tingkat gagal bayar yang rendah. Hal ini dapat membantu meningkatkan jumlah peminjam dan total nominal pinjaman dengan risiko yang lebih rendah.
 - **Peningkatan Manajemen Risiko di Industri dengan Tingkat Gagal Bayar Tinggi:** Tingkatkan manajemen risiko khususnya pada **industri RRE/Rental/Lease, Finance/Insurance, dan Trans/Ware** yang memiliki tingkat gagal bayar tinggi. Peningkatan proses verifikasi, strategi pengambilan keputusan yang lebih baik, dan pemantauan aktif terhadap pelanggan dalam sektor ini dapat membantu mengurangi risiko.
 - **Peningkatan Manajemen Risiko di Industri Retail Trade:** Fokuskan pada peningkatan manajemen risiko di industri Retail Trade. Meskipun memiliki jumlah pinjaman yang besar, tingkat gagal bayarnya cukup tinggi. Strategi ini dapat membantu mengurangi risiko kegagalan bayar dan meningkatkan hasil keuangan.
 - **Fokus pada Term Peminjaman yang Panjang:** Berfokus pada peminjam dengan jangka waktu pinjaman yang panjang, terutama pada kategori **very high (≥ 82 bulan)** dan **high (62-81 bulan)**. Data menunjukkan bahwa banyak nasabah yang berhasil melunasi pinjaman pada jangka waktu ini, sehingga dapat menjadi strategi yang baik untuk meningkatkan kesuksesan pembayaran pinjaman.
- :

Business Simulation

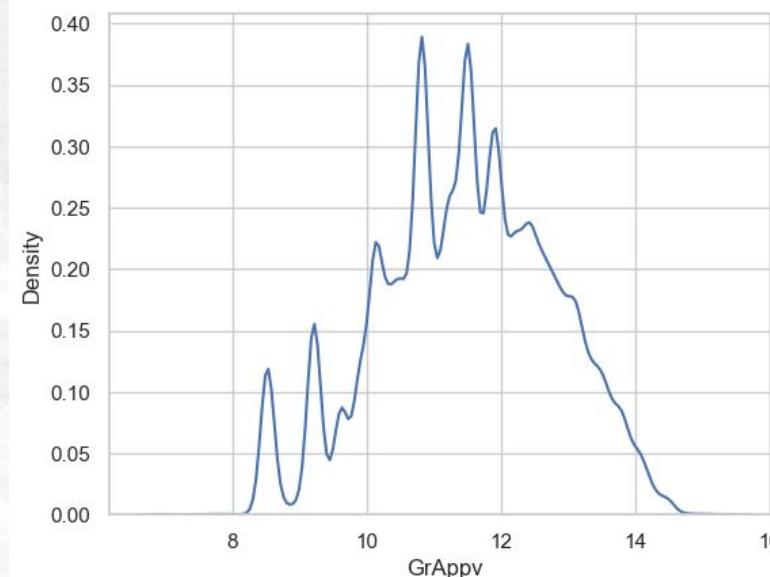
Data Preparation

Simulasi bisnis menggunakan data yang tidak digunakan dalam *training* dan *test*. Sebelum dilakukan simulasi, dilakukan proses *pre-processing* seperti drop kolom yang tidak digunakan, pengecekan data duplikat, *drop missing values*, dan transformasi fitur.

Hasil menunjukkan bahwa tidak ada data duplikat. Transformasi fitur dengan *logaritmik* dilakukan pada fitur GrApp, sementara fitur yang di drop dapat dilihat pada gambar.

```
# mengecek duplicate  
df2.duplicated().sum()
```

```
[ ] # Menghapus satu atau beberapa kolom  
kolom_yang_akan_dihapus = ['LoanNr_ChkDgt', 'Name', 'City', 'State', 'Zip', 'Bank',  
'BankState', 'ApprovalDate', 'ApprovalFY', 'ChgOffDate',  
'DisbursementGross', 'BalanceGross', 'SBA_Appv', 'UrbanRural']  
df2.drop(columns=kolom_yang_akan_dihapus, inplace=True)
```



Business Simulation

Feature Extraction, Scalling & Encoding

- Dilakukan proses Feature Extraction untuk membuat fitur baru yaitu kategorisasi *term*, *job stability*, *recession*, dan *industry*.
- Fitur numerik yang digunakan pada pemodelan (GrAppv) yang sudah dilakukan *log transformasi* kemudian dilakukan standarisasi.
- Dilakukan Feature Encoding pada Feature Categorical.

Pada business simulation, tidak dilakukan *handle outliers* dan *handle class imbalance* agar simulasinya dapat digunakan untuk menguji dan mengevaluasi kinerja model *machine learning* dalam suatu lingkungan yang mendekati kondisi bisnis yang sebenarnya

Penerapan Model

Model yang telah di training pada proses sebelumnya diterapkan pada data sehingga didapatkan nilai prediksi.

Presisi dan akurasi prediksi yang didapatkan dapat dilihat pada gambar.

	NewExist_encoded	RevLineCr_encoded	LowDoc_encoded	Industry_encoded	Job_Stability	CompanySize	Franchise	Recession	term_category_encoded	log_GrAppv_std
0	2	9	3	12	0	0	0	0	1	-1.11
1	1	2	3	1	1	0	0	0	3	-0.25
2	2	2	3	7	0	0	0	0	3	0.84
3	1	2	3	16	0	0	0	0	3	2.25
4	2	9	3	0	0	0	0	0	3	0.34
...
381001	1	2	3	16	0	0	0	0	1	-0.25
381002	1	13	3	16	0	0	0	0	1	-0.10
381003	1	9	3	9	0	1	0	0	3	0.87
381004	1	9	6	18	0	0	0	0	1	-0.20
381005	2	9	3	18	0	0	0	0	0	-0.90

375227 rows × 10 columns

Precision: 0.94

Akurasi: 0.81

Laporan Klasifikasi:

	precision	recall	f1-score	support
0	0.32	0.61	0.42	41921
1	0.94	0.84	0.89	333306
accuracy			0.81	375227
macro avg	0.63	0.72	0.65	375227
weighted avg	0.87	0.81	0.83	375227

Business Simulation



94%
Precision

81%
Accuracy

Confusion Matrix

Hasil prediksi kemudian divisualisasikan menggunakan Confusion Matrix, dan didapatkan hasil.

- **277867 True Positive** (diprediksi berhasil bayar, dan itu benar)
- **25500 True Negative** (diprediksi gagal bayar, dan itu benar)
- **16421 False Positive** (diprediksi berhasil bayar, dan itu salah)
- **55439 False Negative** (diprediksi gagal bayar, dan itu salah)

Business Simulation

Default Percentage (Business Metric 1)

Hasil prediksi model kemudian dibandingkan dengan kondisi sebenarnya menggunakan Confusion Matrix. Tingkat gagal bayar hasil prediksi dihitung dengan rumus **False Positive/(False Positive + True Positive)**.

Sebelum dilakukan pemodelan persentase gagal bayar adalah **11.17 %**. dan setelah dilakukan pemodelan persentase gagal bayar turun 5.59% menjadi **5.58 %**

```
[ ] # Menghitung persentase gagal bayar
persentase_gagal_bayar_sebelum_modelling = (df2[df2['MIS_Status_int'] == 0].shape[0] / len(df2)) * 100
print(f'Persentase Gagal Bayar (Sebelum Modelling): {persentase_gagal_bayar_sebelum_modelling:.2f}%')
Persentase Gagal Bayar (Sebelum Modelling): 11.17%

[ ] # Menghitung persentase false positives
false_positives = len(y_pred[(y_pred == 1) & (y_true == 0)])
# Menghitung total positive predictions
total_positives = len(y_pred[y_pred == 1])

# Menghitung persentase false positives terhadap total positive predictions
persentase_gagal_bayar_setelah_modelling = (false_positives / total_positives) * 100
print(f'Persentase Gagal Bayar (Setelah Modelling): {persentase_gagal_bayar_setelah_modelling:.2f}%')
Persentase Gagal Bayar (Setelah Modelling): 5.59%
```

Business Simulation

Chg-Off Total (Business Metric 2)

Lalu dihitung jumlah uang yang gagal dibayar pada sebelum dan setelah pemodelan. Lalu didapatkan bahwa sebelum pemodelan total uang yang gagal dibayar pada periode data adalah **\$3.38 miliar**, dan setelah pemodelan turun menjadi **\$1.8 miliar**. Hasil menunjukan penurunan hingga **\$1.58 miliar**.

```
[ ] # Menghitung total chgoffprinGr
nominal_gagal_bayar_sebelum_modelling = df2['ChgOffPrinGr'].sum()

print(f'Nominal Gagal Bayar (Sebelum Modelling): {nominal_gagal_bayar_sebelum_modelling}')

Nominal Gagal Bayar (Sebelum Modelling): 3376815398.0

[ ] df2['predicted_status'] = y_pred
# Pilih baris yang diprediksi lunas padahal gagal bayar
gagal_bayar = df2[(df2['predicted_status'] == 1) & (df2['MIS_Status_int'] == 0)]

# Hitung total ChgOffPrinGr pada baris-baris tersebut
nominal_gagal_bayar_setelah_modelling = gagal_bayar['ChgOffPrinGr'].sum()
print(f'Nominal Gagal Bayar (Setelah Modelling): {nominal_gagal_bayar_setelah_modelling}')

Nominal Gagal Bayar (Setelah Modelling): 1852454234.0
```

THANK YOU!

Additional Information:

[Link Dataset \(Modelling\)](#)

[Link Dataset \(Business Simulation\)](#)

[Link Google Colab](#)

[Link Dataset \(Kaggle\)](#)

Notes: Dataset diambil dari Kaggle yang kemudian displit. Sebagian digunakan untuk modelling, sebagian lainnya digunakan untuk business simulation