

Homework

Data Pre-Processing

Final Project - Stage 2



Estimasi Waktu Pengerjaan



3 - 5 jam

Jumlah Soal



2 Soal

Total Point



100 poin

Teknis Pengerjaan

1. Pekerjaan dilakukan secara **berkelompok, sesuai kelompok Final Project**
2. Masing-masing anggota kelompok tetap perlu submit ke LMS (jadi bukan perwakilan)
3. File yang perlu dikumpulkan:
 - File **jupyter notebook** (.ipynb) yang berisi source code.
 - File **laporan homework** (.pdf) yang berisi rangkuman dari apa saja yang telah dilakukan.
4. Upload hasil pengerjaanmu melalui LMS.
 - Masukkan semua file ke dalam **1file** dengan format **ZIP**.
 - Nama File:
Preprocessing - <Nama Kelompok>.zip

1 Data Cleansing (50 poin)

Lakukan pembersihan data, sesuai yang diajarkan di kelas, seperti:

- A. Handle missing values
- B. Handle duplicated data
- C. Handle outliers
- D. Feature transformation
- E. Feature encoding
- F. Handle classimbalance

Di laporan homework, tuliskan apa saja yang telah dilakukan dan metode yang digunakan.

*Tetap tuliskan jika memang ada tidak yang perlu di-handle (contoh: "Tidak perlu feature encoding karena semua feature sudah numerical" atau "Outlier tidak di-handle karena akan fokus menggunakan model yang robust terhadap outlier").

A. Handle missing values

Didapatkan hasil bahwa total baris dataframe terdiri dari 21000 baris, dan tidak terdapat missing value pada dataset.

```
print(f'Terdapat missing data sebanyak {dftrain.isnull().sum().sum()}')
```

```
Terdapat missing data sebanyak 0
```

B. Handle duplicated data

Tidak terdapat duplicated data pada dataset.

```
print(f'Jumlah baris data duplikat = {dftrain.duplicated().sum()}')
```

```
Jumlah baris data duplikat = 0
```

C. Handle outliers

Sebelum melakukan handle outliers dan tahapan data preprocessing lainnya, akan dilakukan data splitting terlebih dahulu karena data train dan data test memerlukan treatment yang berbeda.

```
ind_var = dftrain.drop('default_payment_next_month',axis = 1)
ind_var = ind_var.columns.to_list() #kolom independent variable

X = dftrain[ind_var]
y = dftrain[['default_payment_next_month']]

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)
```

Untuk penentuan data target disini adalah kolom default_payment_next_month, dan dataset dibagi ke dalam data train dan data test dengan rasio 70:30.

C. Handle outliers

Handle outlier ini hanya dilakukan pada data train saja untuk memastikan bahwa model tidak hanya belajar dari nilai ekstrim yang tidak biasa, tetapi juga dari nilai-nilai yang lebih representatif dari dataset.

Outlier Handling ini dilakukan menggunakan IQR karena lebih robust terhadap data yang skewed.

```
# Outliers handling

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

for i in nums:
    q1 = merge_train[i].quantile(0.25)
    q3 = merge_train[i].quantile(0.75)
    iqr = q3 - q1
    low_limit = q1 - (1.5 * iqr)
    high_limit = q3 + (1.5 * iqr)
    filtered_entries = ((merge_train[i] >= low_limit) & (merge_train[i] <= high_limit))

merge_train = merge_train[filtered_entries] #filter untuk hanya mengambil value di atas lower bound atau upper bound

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

Jumlah baris sebelum memfilter outlier : 14700

Jumlah baris setelah memfilter outlier : 13245

D. Feature transformation

Langkah selanjutnya melakukan scaler menggunakan standard scaler pada data train maupun data test agar rentang skala data seragam, hal ini bertujuan untuk memastikan bahwa setiap feature berkontribusi pada hasil model dengan cara yang seimbang

```
#menggunakan standardization (scaler)

from sklearn.preprocessing import StandardScaler
ss = StandardScaler()

numerical_features = X[nums].columns.to_list()
for i in numerical_features:
    scaler = ss.fit(X_train[[i]])
    X_train[i] = scaler.transform(X_train[[i]])
    X_test[i] = scaler.transform(X_test[[i]])
```


E. Feature encoding

Dari dataset diketahui ada kolom yang dengan tipe data kategorikal seperti kolom age, marriage, educational, etc. yang pada awal nilai nya dalam bentuk integer. Tapi untuk proses encoding ini tetap dilakukan karena terdapat value yang ambigu pada kolom kategorikal ini, sehingga untuk alur pengerjaannya dari int -> obj -> int untuk memperjelas value dari masing2 kolom ini.

Terdapat beberapa feature encoding process yang dilakukan, yaitu :

- Pada feature MARRIAGE akan dilakukan label encoding dan dikerucutkan kembali menjadi dua kategori yaitu 'in_relationship'(1) yang berisi data yang berkategori 'married', dan 'not_in_relationship'(0) yang berisi data dengan kategori 'single', 'divorce', atau '0'.
- Pada Feature Education akan dikerucutkan kembali menjadi dua ketegori yaitu educated (1) dan others (0), kemudian dilakukan label encoding untuk menghindari model yang terlalu kompleks dan terlalu banyak feature yang tidak relevan.
- Untuk feature SEX, PAY_1 hingga PAY_6, dan payment_default_next_month tetap.

E. Class Imbalance

Dilakukan class imbalance handling karena terdapat ketimpangan unique value pada data target. Terlihat hanya terdapat 3076 baris data pada kategori 1, hal ini sangat timpang dengan banyak data pada kategori 0.

```
0    10169
1     3076
Name: default_payment_next_month, dtype: int64
```

Dengan melihat hasil yang diperoleh dari tiga metode, selanjutnya akan digunakan hasil Class Imbalance handling dengan metode SMOTE karena jika kita menggunakan under sampling maka data train yang digunakan kurang maksimal karena banyak informasi yang dihapus.

```
0    3076
1    3076
Name: default_payment_next_month, dtype: int64
# Under Sampling

0    10169
1    10169
Name: default_payment_next_month, dtype: int64
# Over Sampling

0    10169
1    10169
Name: default_payment_next_month, dtype: int64
# SMOTE
```

2. Feature Engineering (35 poin)

Cek feature yang ada sekarang, lalu lakukan:

- A. Feature selection (membuang feature yang kurang relevan atau redundan)
 - B. Feature extraction (membuat feature baru dari feature yang sudah ada)
 - C. Tuliskan minimal 4 feature tambahan (selain yang sudah tersedia di dataset) yang mungkin akan sangat membantu membuat performansi model semakin bagus (ini hanya ide saja, untuk menguji kreativitas teman-teman, tidak perlu benar-benar dicari datanya dan tidak perlu diimplementasikan)
- *Untuk 2A & 2B, tetap tuliskan jika memang tidak bisa dilakukan (contoh: "Semua feature digunakan untuk modelling (tidak ada yang dihapus), karena semua feature relevan")

2.A Feature selection

Dari data yang kita gunakan untuk mencari data yang kurang relevan, kita menggunakan *feature importance score* dengan feature ini kita dapat mengetahui feature apa saja dengan score terendah. tetapi dari hasil score terendah perlu kita tinjau lagi kegunaan dari beberapa feature terendah yang mungkin relevan dengan tujuan kita

```
print("\nbottom 10 feature importance scores:")
for feature, score in bottom_scores:
    print(f"{feature}: {score}")
```

Top 10 feature importance scores:

```
PAY_1: 0.15054637268431673
PAY_2: 0.11161158150537759
PAY_4: 0.062068270872733505
PAY_3: 0.06051967016210288
PAY_6: 0.05220417703499967
PAY_5: 0.04938460892558256
ID: 0.04319908623059101
BILL_AMT1: 0.03891510738315949
LIMIT_BAL: 0.03861987589111762
AGE: 0.03555923313702414
```

```
bottom 10 feature importance scores:
BILL_AMT4: 0.03272014059148174
BILL_AMT5: 0.031181702901095603
BILL_AMT6: 0.03032102449486271
PAY_AMT3: 0.029108399036011735
PAY_AMT5: 0.028271118660457412
PAY_AMT4: 0.027978222364433283
PAY_AMT6: 0.027702342852294972
SEX: 0.007432218729567466
MARRIAGE: 0.007014625680425036
EDUCATION: 0.0008996626034117408
```

<< dari 10 feature terbawah kita hanya mengambil feature SEX, MARRIAGE, EDUCATION DAN 1 feature teratas bernama ID dengan kesimpulan

kesimpulan

dari metode yang digunakan untuk mencari feature yang kurang relevan dan dapat dihapuskan disimpulkan Feature ID, education, sex, dan marriage merupakan fitur yang dianggap kurang relevan dalam kasus Payment Default Credit karena tidak memiliki korelasi yang kuat dengan target variabel (default payment next month).

- fitur ID dianggap kurang relevan karena tidak terlalu mempengaruhi seseorang dalam membayar hutang.
- fitur sex (jenis kelamin) juga dianggap kurang relevan karena pada umumnya jenis kelamin tidak mempengaruhi kemampuan seseorang dalam membayar hutang.
- Feature marriage (status pernikahan) dihapus karena tidak relevan dengan risiko default pada kartu kredit. Dalam konteks ini, status pernikahan tidak memberikan informasi yang signifikan dalam memprediksi apakah nasabah akan mengalami keterlambatan pembayaran atau wanprestasi. Dengan menghapus fitur marriage, model prediksi risiko kredit dapat menjadi lebih fokus pada faktor-faktor yang lebih relevan dan dapat meningkatkan akurasi dalam memprediksi risiko default pada kartu kredit.
- feature education (tingkat pendidikan) karena tingkat pendidikan seseorang tidak menjamin seseorang memiliki sebuah pekerjaan atau kepribadian dalam mengolah keuangan. maka dari itu feature ini kurang relevan

2A.Feature selection

Dari hasil diatas kita mulai meninjau performa data sebelum feature dibuang dan sesudah

a.Performa sebelum feature dibuang

```
# Memprediksi data validasi
y_pred = model.predict(X_test)

# Mencetak laporan klasifikasi
print(classification_report(y_test, y_pred))

# Mencetak matriks kebingungan (confusion matrix)
print(confusion_matrix(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.66	0.69	0.67	2019
1	0.68	0.65	0.66	2049
accuracy			0.67	4068
macro avg	0.67	0.67	0.67	4068
weighted avg	0.67	0.67	0.67	4068

```
[[1388 631]
 [ 719 1330]]
```

b.hasil performa setelah feature dibuang

```
model.fit(X_train, y_train)

# Memprediksi data validasi
y_pred = model.predict(X_test)

# Mencetak laporan klasifikasi
print(classification_report(y_test, y_pred))

# Mencetak matriks kebingungan (confusion matrix)
print(confusion_matrix(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.67	0.68	0.68	2019
1	0.68	0.67	0.68	2049
accuracy			0.68	4068
macro avg	0.68	0.68	0.68	4068
weighted avg	0.68	0.68	0.68	4068

```
[[1382 637]
 [ 666 1383]]
```

dapat disimpulkan bahwa performa meningkat ketika 4 feature diatas dibuang

2A.feature selection

Selanjutnya kita mulai membuang 4 feature [id, sex, marriage, education] yang kurang relevan bahkan meningkatkan performa.

```

✓ s ▶ train_over = train_over.drop(['ID', 'SEX', 'EDUCATION', 'MARRIAGE'], axis=1)

#Menyimpan file dtaframe yang telah diubah ke file csv yang baru
train_over.to_csv('trainx.csv', index=False)

```

2B.Feature extraction

Ada 3 feature baru yang kita tambahkan dari feature lama
1.Feature total payment rasio (rasio total pembayaran)

• feature total_payment_ratio

```
# Hitung total pembayaran dan total tagihan
train_over['total_payment'] = train_over['PAY_AMT1'] + train_over['PAY_AMT2'] + train_over['PAY_AMT3'] + train_over['PAY_4']
train_over['total_bill'] = train_over['BILL_AMT1'] + train_over['BILL_AMT2'] + train_over['BILL_AMT3'] + train_over['BILL_4']

# Hitung rasio total pembayaran dengan total tagihan
train_over['total_payment_ratio'] = train_over['total_payment'] / train_over['total_bill']

# Cetak hasil
train_over.head(5)
```

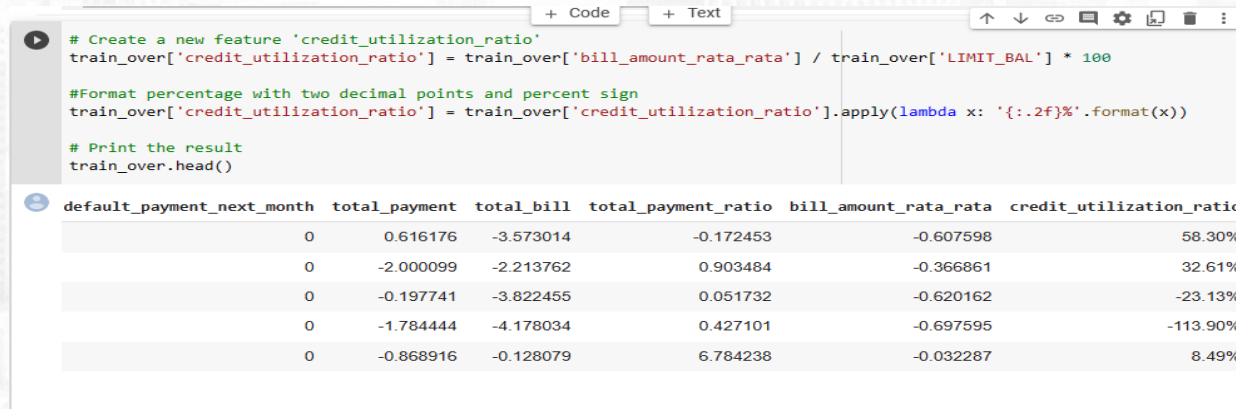
AY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default_payment_next_month	total_payment	total_bill	total_payment_ratio
1.371369	-0.313067	0.342145	-0.184595	0.312780	0	0.616176	-3.573014	-0.172453
1.197996	-0.227015	-0.251604	-0.250182	-0.825059	0	-2.000099	-2.213762	0.903484
1.264379	-0.187070	-0.290846	1.594598	-0.718702	0	-0.197741	-3.822455	0.051732
1.156776	-0.273716	-0.123148	-0.199462	-0.714118	0	-1.784444	-4.178034	0.427101
1.168453	-0.178309	-0.145732	-0.190117	0.028550	0	-0.868916	-0.128079	6.784238

Rasio total pembayaran atau total payment ratio adalah rasio antara jumlah pembayaran yang telah dilakukan oleh nasabah pada tagihan kartu kredit mereka dalam periode waktu tertentu (misalnya, 6 bulan terakhir) dibandingkan dengan total tagihan yang harus dibayarkan dalam periode waktu yang sama. Dengan kata lain, rasio ini mengukur seberapa banyak dari total tagihan yang telah dibayarkan oleh nasabah. Rasio total pembayaran dapat memberikan indikasi tentang kemampuan nasabah dalam membayar tagihan mereka. Semakin tinggi rasio pembayaran, semakin baik kemampuan nasabah untuk membayar tagihan mereka dalam waktu yang ditentukan. Sebaliknya, jika rasio pembayaran rendah, maka ini bisa menjadi tanda bahwa nasabah mungkin mengalami kesulitan dalam membayar tagihan mereka tepat waktu. Dalam industri kartu kredit, rasio total pembayaran ini merupakan salah satu indikator kredit yang penting untuk mengevaluasi risiko kredit nasabah. Semakin tinggi rasio pembayaran, semakin rendah risiko kredit nasabah, dan semakin rendah kemungkinan mereka akan menjadi pembayaran yang macet atau wanprestasi.

2B.feature extraction

Oleh karena itu, rasio total pembayaran juga dapat digunakan sebagai salah satu fitur dalam model prediksi risiko kredit atau default pada industri kartu kredit.

2.credit_utilizitation_ratio (feature rasio penggunaan credit)



```
# Create a new feature 'credit_utilization_ratio'
train_over['credit_utilization_ratio'] = train_over['bill_amount_rata_rata'] / train_over['LIMIT_BAL'] * 100

#Format percentage with two decimal points and percent sign
train_over['credit_utilization_ratio'] = train_over['credit_utilization_ratio'].apply(lambda x: '{:.2f}%'.format(x))

# Print the result
train_over.head()
```

default_payment_next_month	total_payment	total_bill	total_payment_ratio	bill_amount_rata_rata	credit_utilization_ratio
0	0.616176	-3.573014	-0.172453	-0.607598	58.30%
0	-2.000099	-2.213762	0.903484	-0.366861	32.61%
0	-0.197741	-3.822455	0.051732	-0.620162	-23.13%
0	-1.784444	-4.178034	0.427101	-0.697595	-113.90%
0	-0.868916	-0.128079	6.784238	-0.032287	8.49%

Rasio penggunaan kredit (credit utilization ratio) menggambarkan seberapa banyak kredit yang digunakan oleh pengguna dalam persentase dari total kredit yang tersedia. Rasio penggunaan kredit biasanya dihitung dengan membagi saldo tagihan saat ini dengan batas kredit yang tersedia. Contohnya, jika pengguna memiliki saldo tagihan sebesar 5000 dan batas kredit sebesar 10000, maka rasio penggunaan kreditnya adalah 50%.

2B.feature extraction

Rasio penggunaan kredit dapat menjadi indikator penting untuk menentukan resiko pembayaran yang tidak lancar atau mengalami keterlambatan. Semakin tinggi rasio penggunaan kredit, semakin tinggi resiko pembayaran yang tidak lancar, karena pengguna mungkin kesulitan membayar tagihan bulanan yang semakin tinggi. Oleh karena itu, pada dataset "Default of Credit Card Clients", rasio penggunaan kredit sering digunakan sebagai salah satu fitur dalam memprediksi default pembayaran bulanan.

Beberapa studi menunjukkan bahwa rasio penggunaan kredit mempengaruhi skor kredit individu dan tingkat kepercayaan kreditur terhadap pengguna. Selain itu, rasio penggunaan kredit juga dapat memengaruhi kemampuan seseorang untuk memperoleh kredit lebih lanjut di masa depan. Oleh karena itu, rasio penggunaan kredit dapat menjadi indikator penting dalam menentukan kesehatan keuangan seseorang dan kelayakan kredit.

Rasio penggunaan kredit dapat dihitung dengan membagi "LIMIT_BAL" (batas kredit) dengan jumlah total saldo tagihan di setiap bulan. Rasio ini digunakan sebagai fitur dalam memprediksi default pembayaran bulanan. Dalam beberapa kasus, rasio penggunaan kredit juga dapat dihitung dengan menggunakan saldo tagihan pada bulan sebelumnya, atau rata-rata saldo tagihan dalam beberapa bulan terakhir, untuk memberi gambaran yang lebih akurat tentang kebiasaan pengguna dalam menggunakan kredit.

2C. Feature Tambahan

Tuliskan minimal 4 feature tambahan (selain yang sudah tersedia di dataset) yang mungkin akan sangat membantu membuat performansi model semakin bagus (ini hanya ide saja, untuk menguji kreativitas teman-teman, tidak perlu benar-benar dicari datanya dan tidak perlu diimplementasikan)

Berikut 4 feature tersebut yang kami anggap memenuhi kriteria tersebut :

1. Kategori Belanja/Penggunaan Kartu Kredit : Untuk melihat pola konsumsi nasabah
2. Jumlah Pengeluaran : Untuk melihat jumlah tagihan
3. Bunga/Denda : Untuk men-track jumlah tagihan
4. Pendapatan : Untuk mengukur kemampuan nasabah membayar tagihan

3. Git (15 poin)

Link Repository Github :

https://github.com/iqbalmudzakky/final_project.git

Selamat Mengerjakan!