STAGE 2

Data Pre-Processing

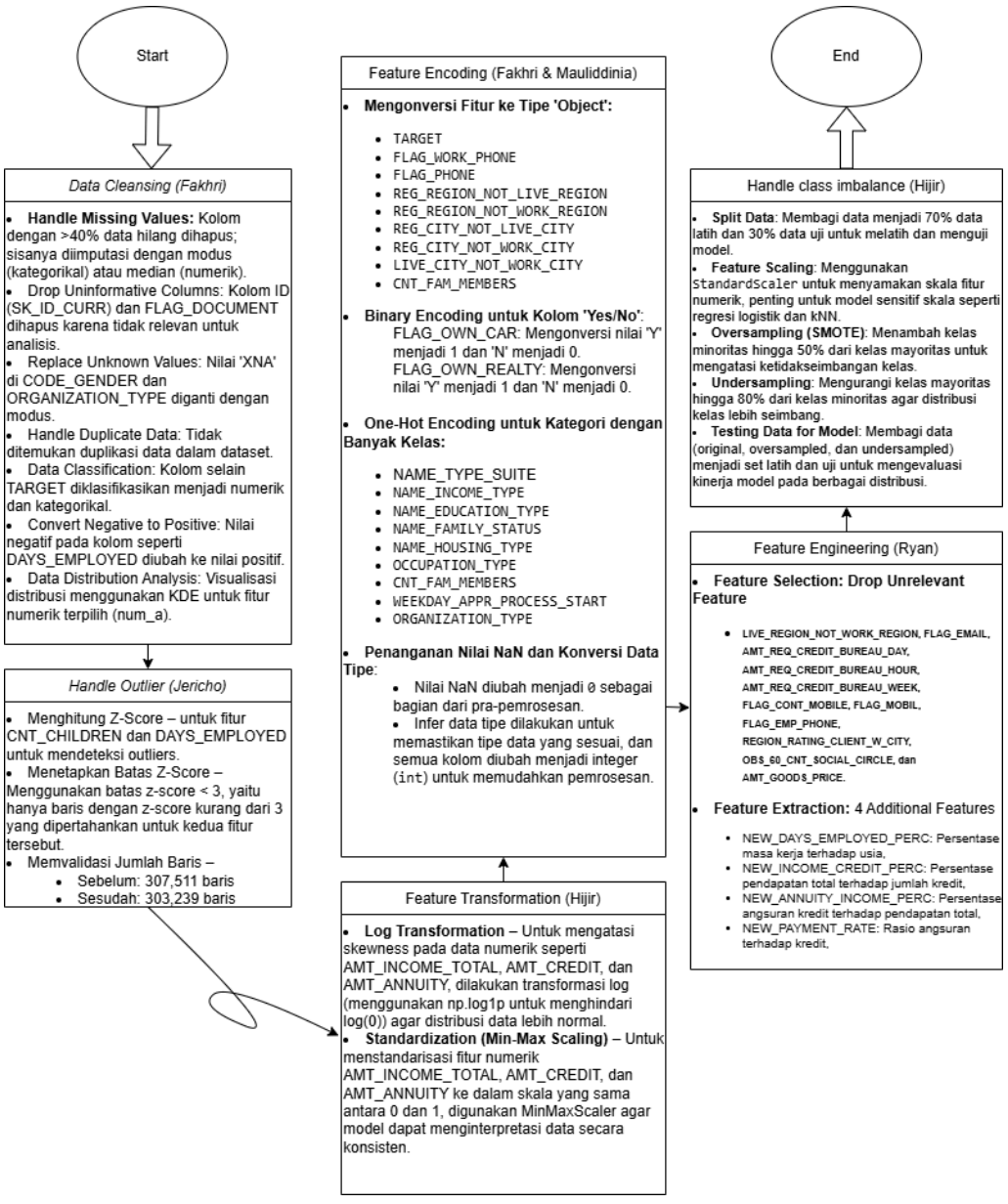
Group 3

Byte Me

Hijir Della Wirasti
Mauliddinia Iftikhar Agnany
Jericho Medion Haryono
Fakhri Dwi Nugroho
Ryan Nofandi

Dataset: Home Credit Default Risk

GITHUB: https://github.com/hijirdella/Preprocessing-Home-Credit



Gambar 1. Flowchart Final Project Stage 2

1. Data Cleansing
   A. Handle missing values

Identifikasi Missing Values:

- Dihitung jumlah nilai hilang untuk setiap kolom dalam dataset, dan persentase missing values di setiap kolom.
- Kolom dengan lebih dari 40% nilai hilang dihapus dari dataset, karena terlalu banyak nilai kosong yang dapat mengganggu analisis dan model.

Imputasi Missing Values Berdasarkan Tipe Data:

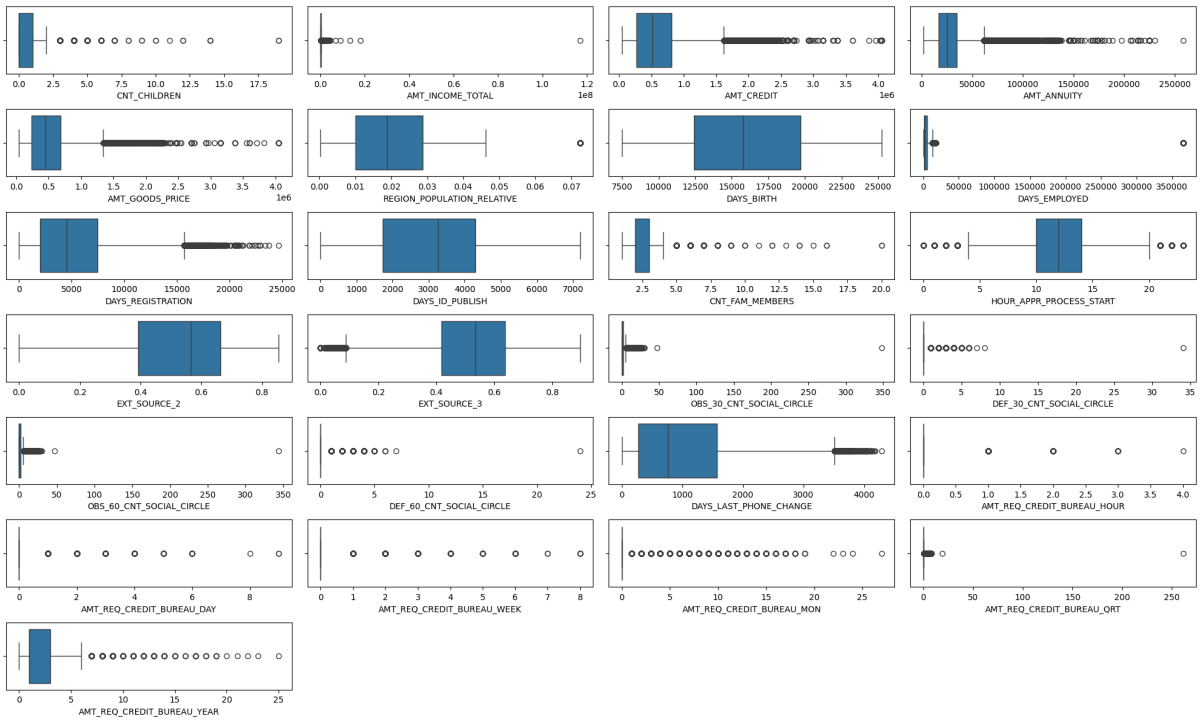
- **Kategorikal**: Untuk kolom yang bersifat kategorikal dan memiliki missing values, diisi dengan modus (nilai yang paling sering muncul). Ini dilakukan agar distribusi data kategorikal tetap terjaga.
- **Numerik**: Untuk kolom numerik, missing values diisi dengan median. Penggunaan median membantu menjaga distribusi data, terutama jika terdapat skewness, karena median kurang sensitif terhadap nilai ekstrem dibandingkan rata-rata.

Verifikasi Missing Values Setelah Imputasi:

- Setelah proses imputasi, dilakukan pengecekan ulang untuk memastikan bahwa semua missing values telah terisi dan dataset siap untuk proses selanjutnya.

B. Handle duplicated data

Hasil pengecekan menunjukkan bahwa tidak ada baris duplikat dalam dataset, sehingga tidak ada tindakan lebih lanjut yang diperlukan untuk menghapus duplikasi.



Kolom yang memiliki nilai abnormal adalah **CNT_CHILDREN** dan **DAYS_EMPLOYED**, sehingga kita perlu menghapus outlier pada kedua kolom ini.

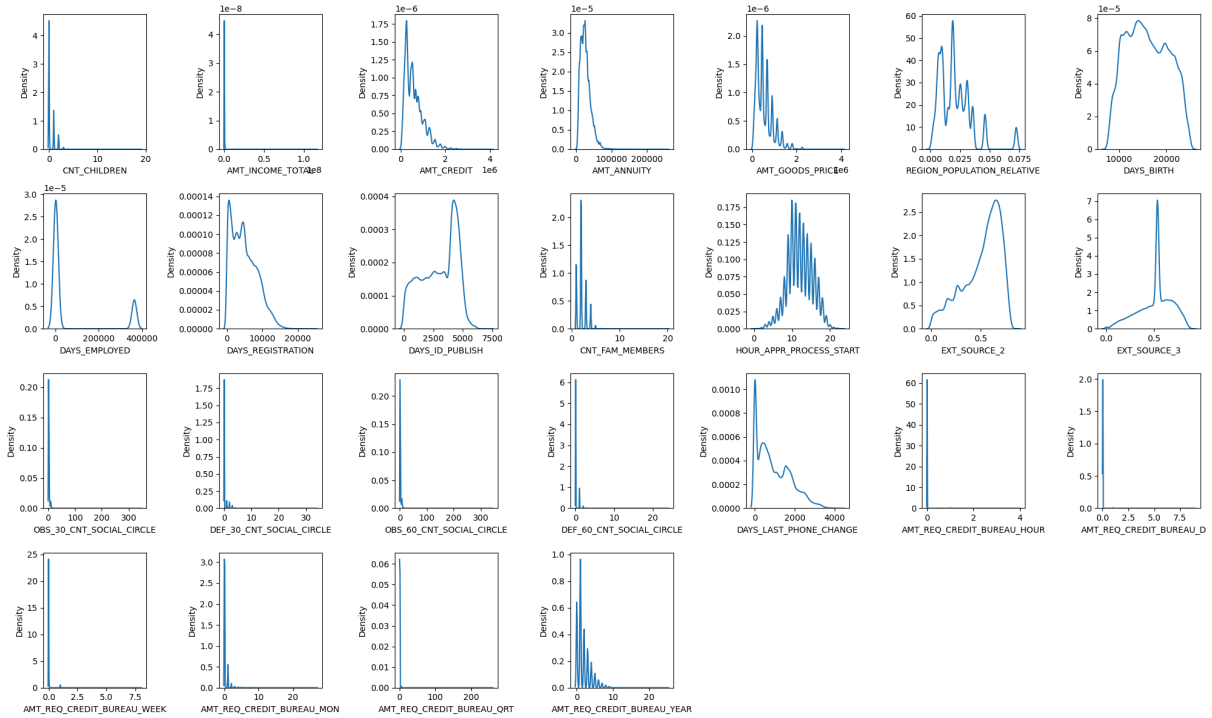
- Untuk **CNT_CHILDREN**, nilai yang sangat tinggi (misalnya, lebih dari 10 anak) jarang terjadi dan dapat mengganggu analisis, sehingga sebaiknya dihapus.
- Untuk **DAYS_EMPLOYED**, nilai yang sangat besar (misalnya, ribuan hari bekerja) kemungkinan merupakan kesalahan atau tidak realistis, sehingga outlier pada kolom ini juga perlu dihapus untuk menjaga integritas data.

C. Handle outliers

Berikut adalah langkah-langkah yang dilakukan untuk menangani **Outliers** dalam dataset:

1. **Identifikasi Outliers dengan Z-Score**:
   - Outliers diidentifikasi pada dua kolom, yaitu `CNT_CHILDREN` dan `DAYS_EMPLOYED`, menggunakan metode Z-score.
   - Z-score dihitung untuk masing-masing kolom, dan nilai yang memiliki Z-score lebih besar dari 3 dianggap sebagai outlier.
2. **Penerapan Filter Outliers**:
   - Hanya baris yang memiliki Z-score kurang dari 3 untuk kedua kolom yang dipertahankan, sehingga baris dengan nilai ekstrem pada `CNT_CHILDREN` dan `DAYS_EMPLOYED` dihapus dari dataset.
3. **Validasi Penghapusan Outliers**:
   - Setelah penghapusan outliers, jumlah baris diperiksa kembali untuk memastikan bahwa data telah dibersihkan dari nilai-nilai ekstrem.

Langkah ini bertujuan untuk menghilangkan pengaruh data ekstrem yang dapat mengganggu performa model, terutama pada algoritma yang sensitif terhadap outliers.



Kesimpulan

1. **Bentuk Distribusi**: Sebagian besar kolom numerik menunjukkan distribusi yang tidak simetris atau skewed, dengan beberapa kolom menunjukkan distribusi normal.
2. **Outliers**: Beberapa kolom menunjukkan lonjakan tajam di ujung distribusi, yang menandakan adanya outliers yang signifikan.
3. **Skewness**: Distribusi data yang skewed menandakan bahwa data tidak terdistribusi merata. Transformasi data mungkin diperlukan untuk mengurangi skewness.
4. **Variabilitas**: Sebaran data cukup lebar pada beberapa kolom, menunjukkan variabilitas yang tinggi.

Kesimpulan keseluruhan: Data distribusi tidak mendekati distribusi normal, sehingga perlu dilakukan **normalisasi** atau transformasi untuk membuat distribusi lebih merata dan mendekati distribusi normal.

D. Feature transformation

Berikut adalah langkah-langkah yang dilakukan untuk **Feature Transformation** dalam dataset:

1. **Log Transformation**:
   - Dilakukan log transformation pada fitur numerik yang memiliki skewness tinggi, seperti `AMT_INCOME_TOTAL`, `AMT_CREDIT`, dan `AMT_ANNUITY`.
   - Transformasi ini menggunakan `np.log1p` untuk menghindari masalah log(0) dan membantu membuat distribusi data lebih mendekati normal, sehingga meningkatkan performa model.
2. **Normalization (Min-Max Scaling)**:

- Setelah log transformation, fitur `AMT_INCOME_TOTAL`, `AMT_CREDIT`, dan `AMT_ANNUITY` dinormalisasi menggunakan `MinMaxScaler`.
- Normalisasi ini menempatkan nilai fitur dalam rentang 0 hingga 1, yang membantu model seperti regresi logistik dan kNN bekerja lebih optimal dengan skala data yang seragam.

Langkah-langkah feature transformation ini dilakukan untuk mengatasi skewness dan memastikan skala data yang seragam, sehingga model dapat menginterpretasi data dengan lebih konsisten.

E. Feature encoding

Berikut adalah langkah-langkah yang dilakukan untuk **Feature Encoding** dalam dataset:

1. **Binary Encoding untuk Kolom Kategorikal 'Yes/No'**:
   - Kolom seperti `FLAG_OWN_CAR` dan `FLAG_OWN_REALTY` dikonversi dari nilai `'Y'` dan `'N'` menjadi `1` dan `0` agar mudah diinterpretasi oleh model.
2. **Mapping Encoding untuk Kolom dengan Dua Kategori**:
   - Kolom `CODE_GENDER` dikonversi dengan mapping `'M'` menjadi `1` dan `'F'` menjadi `0`.
   - Kolom `NAME_CONTRACT_TYPE` dikonversi dari `'Cash loans'` menjadi `1` dan `'Revolving loans'` menjadi `0`.
3. **One-Hot Encoding untuk Kolom dengan Banyak Kategori**:
   - Kolom yang memiliki lebih dari dua kategori, seperti `NAME_TYPE_SUITE`, `NAME_INCOME_TYPE`, `NAME_EDUCATION_TYPE`, `NAME_FAMILY_STATUS`, `NAME_HOUSING_TYPE`, `OCCUPATION_TYPE`, `CNT_FAM_MEMBERS`, `WEEKDAY_APPR_PROCESS_START`, dan `ORGANIZATION_TYPE`, diubah menggunakan one-hot encoding.
   - Dalam proses ini, satu kategori di-drop untuk menghindari multikolinearitas, sehingga setiap kategori direpresentasikan sebagai kolom biner.
4. **Handling Unknown Values**:
   - Pada kolom `CODE_GENDER` dan `ORGANIZATION_TYPE`, nilai `'XNA'` diganti dengan nilai modus agar tidak ada data yang tidak terdefinisi.

Langkah-langkah feature encoding ini bertujuan untuk memastikan semua data kategorikal terkonversi ke dalam format numerik yang sesuai untuk pemodelan, sehingga model dapat membaca dan memproses data dengan optimal.

F. Split Data

Pembagian Data: Dataset dibagi menjadi dua bagian, yaitu 70% data latih dan 30% data uji. Data latih digunakan untuk melatih model, sementara data uji digunakan untuk mengevaluasi kinerja model pada data yang belum pernah dilihat, sehingga memastikan hasil yang lebih objektif.

G. Feature Scaling

StandardScaler: Digunakan untuk menstandardisasi fitur numerik, sehingga setiap fitur memiliki mean 0 dan standar deviasi 1. Standardisasi ini penting untuk model yang sensitif terhadap skala, seperti regresi logistik dan kNN, agar semua fitur memiliki kontribusi yang seimbang dan meningkatkan stabilitas serta kecepatan konvergensi model.

H. Handle class imbalance

	TARGET	Customers	Customers_pct
0	0	282686	0.92
1	1	24825	0.08

Pada dataset ini, terdapat ketidakseimbangan kelas yang signifikan pada target variabel `TARGET`. Distribusi kelas menunjukkan bahwa 92% data berada di kelas `0` (tidak bermasalah), sedangkan hanya 8% berada di kelas `1` (bermasalah). Ketidakseimbangan kelas ini dapat menyebabkan model lebih cenderung memprediksi kelas mayoritas (`0`) dan mengabaikan kelas minoritas (`1`), sehingga menurunkan performa model dalam mendeteksi kelas `1`.

Untuk mengatasi masalah ini, teknik penyeimbangan kelas seperti **SMOTE (Synthetic Minority Over-sampling Technique)** untuk oversampling atau **RandomUnderSampler**.

1. **Oversampling dengan SMOTE**:
    - Untuk mengatasi ketidakseimbangan kelas, metode **SMOTE** (Synthetic Minority Over-sampling Technique) digunakan. Teknik ini menambah sampel sintetis pada kelas minoritas hingga mencapai 50% dari kelas mayoritas, sehingga distribusi kelas menjadi lebih seimbang.
    - SMOTE membantu meningkatkan performa model dalam mengenali pola dari kelas minoritas.
2. **Undersampling dengan RandomUnderSampler**:
    - Selain oversampling, dilakukan juga **undersampling** pada kelas mayoritas menggunakan **RandomUnderSampler** untuk mengurangi jumlah sampel kelas mayoritas hingga mencapai 80% dari jumlah kelas minoritas.
    - Teknik ini memastikan dataset tetap terjaga ukurannya dan model tidak terlalu condong pada kelas mayoritas.
3. **Split Data Berdasarkan Metode Sampling**:
    - Setelah melakukan oversampling dan undersampling, data dibagi menjadi set data latih dan data uji untuk setiap metode (original, oversampled, dan undersampled). Hal ini memungkinkan untuk membandingkan performa model pada berbagai distribusi data dan memilih hasil terbaik.

Langkah-langkah ini membantu mengatasi ketidakseimbangan kelas yang signifikan, sehingga model dapat belajar secara optimal dan mengurangi bias terhadap kelas mayoritas.

2. Feature Engineering
   A. Feature selection

Menghapus Fitur yang Tidak Relevan:

- Kolom-kolom yang tidak memberikan informasi signifikan untuk pemodelan, seperti `SK_ID_CURR` (ID pelanggan) dan beberapa kolom dengan lebih dari 40% missing values, dihapus dari dataset.
- Beberapa kolom dihapus karena tidak memberikan nilai prediktif yang signifikan. Fitur yang di-drop antara lain `LIVE_REGION_NOT_WORK_REGION`, `FLAG_EMAIL`, `AMT_REQ_CREDIT_BUREAU_DAY`, `AMT_REQ_CREDIT_BUREAU_HOUR`, `AMT_REQ_CREDIT_BUREAU_WEEK`, `FLAG_CONT_MOBILE`, `FLAG_MOBIL`, `FLAG_EMP_PHONE`, `REGION_RATING_CLIENT_W_CITY`, `OBS_60_CNT_SOCIAL_CIRCLE`, dan `AMT_GOODS_PRICE`.

Menghapus Kolom FLAG_DOCUMENT:
Kolom `FLAG_DOCUMENT` dihapus karena tidak memiliki nilai prediktif untuk target. Kolom ini hanya menunjukkan dokumen-dokumen yang terkait tetapi tidak berhubungan langsung dengan prediksi target.

B. Kolom `FLAG_DOCUMENT` dihapus karena tidak memiliki nilai prediktif untuk target. Kolom ini hanya menunjukkan dokumen-dokumen yang terkait tetapi tidak berhubungan langsung dengan prediksi target.

C. Feature extraction

Membuat Fitur Baru untuk Menambah Informasi:

a. **`NEW_DAYS_EMPLOYED_PERC`**: Menghitung persentase masa kerja terhadap usia (`DAYS_EMPLOYED / DAYS_BIRTH`) untuk memberikan gambaran kapan seseorang memulai karir.
b. **`NEW_INCOME_CREDIT_PERC`**: Rasio pendapatan terhadap jumlah kredit (`AMT_INCOME_TOTAL / AMT_CREDIT`) yang menunjukkan kemampuan membayar.
c. **`NEW_ANNUITY_INCOME_PERC`**: Rasio angsuran terhadap pendapatan (`AMT_ANNUITY / AMT_INCOME_TOTAL`) untuk melihat beban angsuran nasabah.
d. **`NEW_PAYMENT_RATE`**: Rasio angsuran terhadap jumlah kredit (`AMT_ANNUITY / AMT_CREDIT`) untuk menunjukkan seberapa cepat kredit bisa dilunasi.

Langkah-langkah feature engineering ini bertujuan untuk menghilangkan fitur yang kurang relevan dan menambah fitur baru yang dapat memberikan lebih banyak informasi bagi model, sehingga meningkatkan akurasi dan interpretabilitas prediksi.