# Stage 2 - Data Pre-Preprocessing

1. **Data Cleansing**

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

   A. Handle missing values

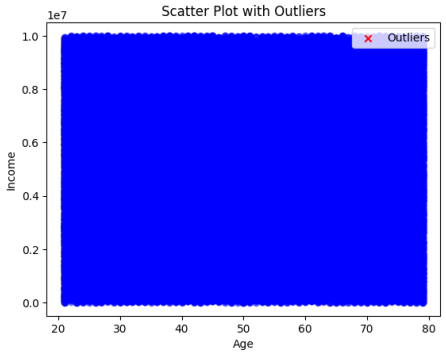
      Pada dataset ini, tidak ditemukan adanya missing values, sehingga tidak dilakukan proses untuk handle missing values

   B. Handle duplicate data

      - Langkah awal handle duplicate adalah menghilangkan kolom identifier dengan menghapus kolom ‘Id’. Jumlah data sebelum dilakukan handle duplicate sebesar 252000 data.
      - Handle duplicated dilakukan dengan cara mendrop data yang terduplikasi sehingga diperoleh data bersih sebesar 43190 data
      - Jadi dapat disimpulkan bahwa data terduplikasi dan telah dilakukan handle duplicate sehingga data telah siap untuk dilakukan tahap selanjutnya.

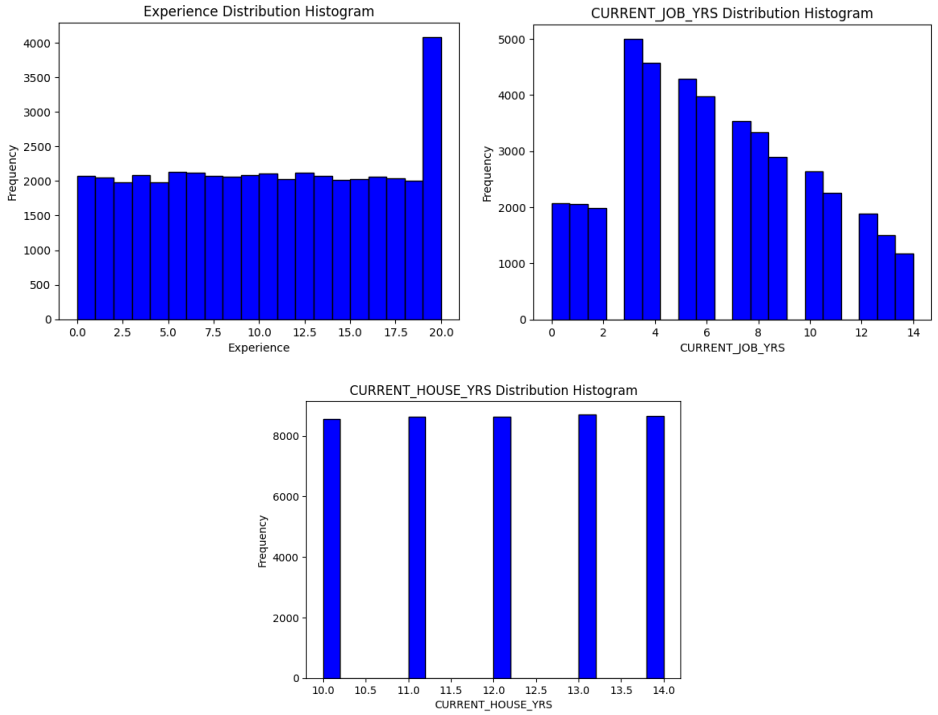
   C. Handle outliers

      - Handle outliers pada kolom ‘Income’ dan ‘Age’




Handle outliers pada kolom Income dan Age diidentifikasi melalui scatter plot serta diperoleh bahwa tidak ada outliers pada kolom Income dan Age. Karena sebaran data tidak memiliki nilai signifikan dan tidak berjauhan dari pola nya. Selain itu, outliers memiliki simbol "X" dimana pada visualisasi tidak ditemukan simbol "X" sehingga dapat disimpulkan tidak ada outliers pada kolom Income dan Age.

      - Handle outliers pada kolom ‘Experience’; ‘Current_house_yrs’; ’Current_job_yrs’




Visualisasi persebaran data melalui histogram pada kolom ‘Experience’; ‘Current_house_yrs’; ’Current_job_yrs’ diperoleh sebaran data yang tidak memiliki nilai signifikan tinggi atau rendah. Kemudian, outliers diidentifikasi menggunakan uji IQR dengan batas outliers seperti pada persamaan berikut,

Lower fence = Q1 -1,5 x IQR

Upper fence = Q3 - 1,5 x IQR

Kemudian di identifikasi outliers dan diperoleh hasil bahwa tidak ada outliers pada kolom ‘Experience’;‘Current_house_yrs’; ’Current_job_yrs’ sehingga outliers tidak perlu di handle.

D. Feature transformation

- Log Transformation:
  Income, Age, dan Experience memiliki distribusi mendekati normal, sehingga tidak perlu Log Transformation.

  CURRENT_JOB_YRS memiliki distribusi right-skewed namun relatif kecil (Skewness: 0.27314433155243134), dan CURRENT_HOUSE_YRS memiliki distribusi multimodal, sehingga Log Transformation tidak diperlukan.

- Normalization/Standardization:
  CURRENT_JOB_YRS dan CURRENT_HOUSE_YRS dilakukan Normalisasi untuk mengatasi perbedaan skala data, karena normalisasi akan membawa data ke rentang yang seragam antara 0 dan 1. Hal ini akan membantu algoritma machine learning yang sensitif terhadap perbedaan skala dalam data, seperti regresi logistik dan k-nearest neighbors, untuk menghasilkan hasil yang lebih

baik. Selain itu, normalisasi juga cocok untuk data dengan distribusi yang tidak normal atau bimodal.

Age, Income, dan Experience dilakukan Standarisasi karena memiliki distribusi mendekati normal. Standarisasi mengubah data menjadi distribusi normal standar dengan rata-rata 0 dan deviasi standar 1, sehingga memenuhi asumsi algoritma machine learning yang mengasumsikan distribusi normal. Dengan standarisasi, variabel-variabel ini akan memiliki dampak yang seimbang pada pemodelan.

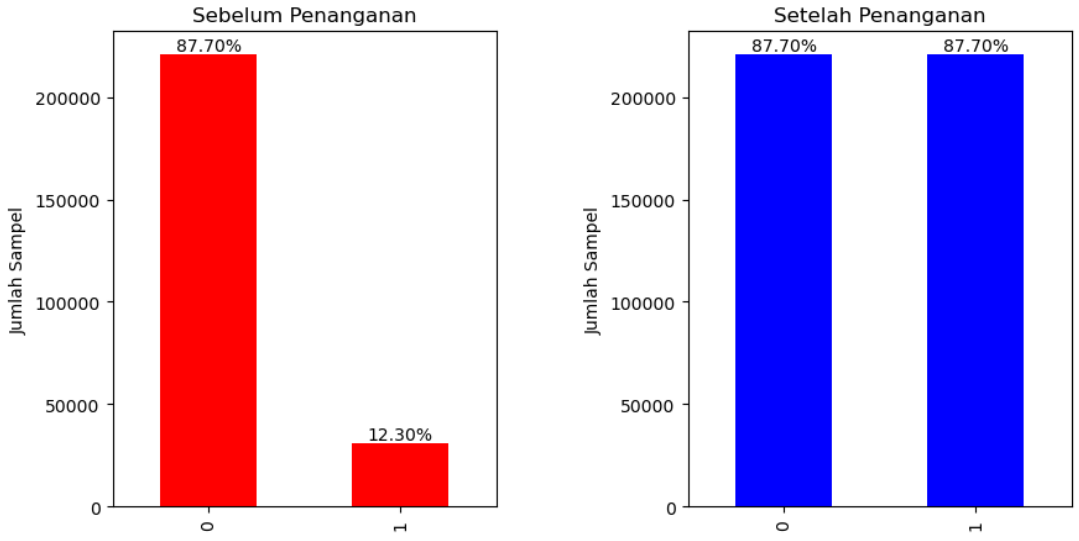
E. Feature encoding - Alyani

Feature encoding dilakukan untuk mengubah data kategorik menjadi angka untuk mempermudah proses learning. Dalam hal ini, ada 2 tipe encoding yang dilakukan, yaitu Ordinal Encoding untuk data yang berurutan/memiliki tingkatan seperti urutan Age, Income, Current Job Years, dan Experience. Lalu, pada data kategorik lainnya seperti State, City, Married/Status, dsb. diterapkan LabelEncode untuk mengubah data kategorik menjadi angka agar angka-angka tersebut berfungsi sebagai representasi untuk learning nantinya.

F. Handle class imbalance

Handle class imbalance dilakukan dimana kondisi suatu kelompok kelas memiliki jumlah data yang berbeda jauh dengan kelas lainnya. Hal ini ditemukan juga pada dataset “Training Data” bahwa data Risk Flag berdistribusi tidak seimbang yang menyebabkan model kalsifikasi menjadi bias dan kurang mampu memprediksi kelas minoritas. Oleh sebab itu, dilakukan handle imbalanced data dengan pendekatan Oversampling. Tujuannya adalah untuk meningkatkan sampel kelas minoritas sampai sama dengan kelas mayoritas lain dengan menduplikasi secara acak sampel kelas minoritas.

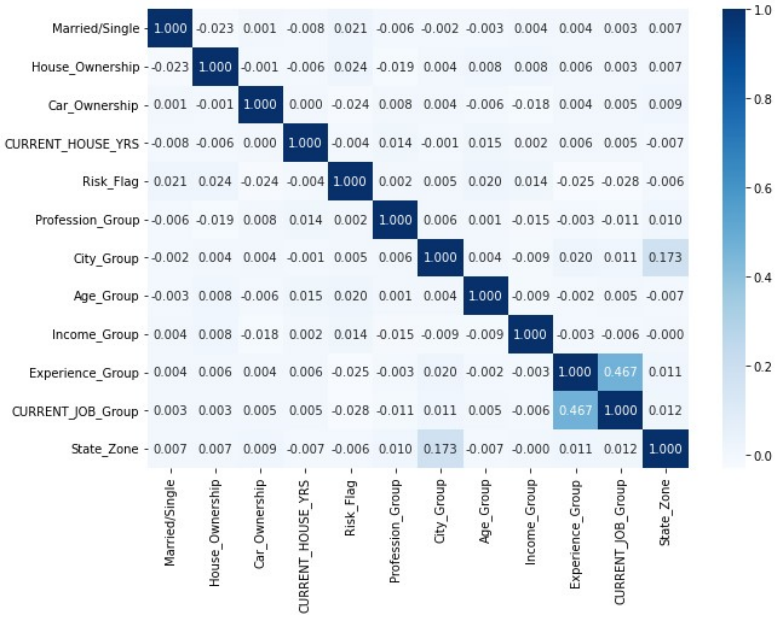
Handle Class Imbalance “Risk_Flag”




## 2. Feature Engineering ()

Cek feature yang ada sekarang, lalu lakukan:

A. Feature selection (membuang feature yang kurang relevan atau redundan)




```
# melihat nilai korelasi feature dengan feature target
data_corr.corr()['Risk_Flag'].sort_values()
```

```
CURRENT_JOB_Group   -0.028410
Experience_Group    -0.025108
Car_Ownership       -0.024036
State_Zone          -0.005805
CURRENT_HOUSE_YRS   -0.004375
Profession_Group     0.002047
City_Group           0.004921
Income_Group         0.013840
Age_Group            0.020129
Married/Single       0.021092
House_Ownership      0.023622
Risk_Flag            1.000000
Name: Risk_Flag, dtype: float64
```

**Kesimpulan:**

1. Feature yang memiliki nilai korelasi tertinggi dengan feature target adalah CURRENT_JOB_Group dengan nilai (-0.284), Experience_Group (-0.251), dan Car_Ownership (-0.24). Namun ketiga feature tersebut memiliki nilai korelasi yang negatif artinya jika Jika nilai feature tsb naik maka feature targetnya turun atau risiko gagal bayar nya turun.
2. Adapun pada feature House_Ownership dengan nilai (0.024) dan feature Married/Single (0.211) memiliki nilai korelasi positif.
3. Sedangkan untuk feature City_Group dengan State_Zone memiliki nilai korelasi yang tinggi (0.173) artinya kedua feature tersebut redundan atau data nya memiliki kesamaan.

**Rekomendasi:**

1. Feature CURRENT_JOB_Group, Experience_Group, Car_Ownership, House_Ownership, feature Married/Single harus dipertahankan ketika melakukan modeling
2. Adapun feature yang nilai korelasinya rendah bisa di drop untuk menhindari kompleksitas feature saat modeling. Namun pertimbangkan juga fetaure lainnya untuk menghindari underfitting.
3. Feature City_Group dengan State_Zone memiliki nilai korelasi yang tinggi atau redundan, maka salah satu dari feature tersebut harus dihapus.

B. Feature extraction (membuat feature baru dari feature yang sudah ada)

Feature extraction dilakukan dengan kombinasi 2 kolom fitur data, hal ini dilakukan dengan tujuan membuat semakin sedikit nya atau compact fitur data yang nantinya digunakan dalam proses modelling prediksi dengan machine learning. Proses ini dilakukan sebelum transformasi dan normalisasi data karena data ini nantinya dapat digunakan dalam modelling.

a. Pembuatan kolom data baru dengan kombinasi data City_Group dan State_Zone

```
data_group['City_State'] = data_group['City_Group'] + '_' + data_group['State_Zone']
```

b. Pembuatan kolom data baru bernama “Asset” dengan data status kepemilikan rumah dan mobil

```
# Membuat mapping dari kombinasi kelas House_Ownership dan Car_Ownership ke nilai baru
mapping = {
    ('owned', 'yes'): 1,
    ('owned', 'no'): 2,
    ('rented', 'yes'): 3,
    ('rented', 'no'): 4,
    ('norent_noown', 'yes'): 5,
    ('norent_noown', 'no'): 6
}

# Menggunakan map untuk menggabungkan kelas
data_extract['Asset'] = data_extract[['House_Ownership', 'Car_Ownership']].apply(lambda x: mapping.get(tuple(x)), axis=1)
data_extract
```

c. Pembuatan kolom data baru bernama “Age_Married” dengan tolak ukur data Married/Singgle dan Age_Group

```
data_group['Age_Married'] = data_group['Age_Group'] + ' ' + data_group['Married/Single']

# Map value
data_group['Age_Married'] = data_group['Age_Married'].replace({
    'Millennials married': 1,
    'Millennials single': 2,
    'Gen X married': 3,
    'Gen X single': 4,
    'Baby Boomers married': 5,
    'Baby Boomers single': 6
})
```

C. Feature Tambahan

Feature tambahan yang dapat membantu dalam membuat performansi model adalah

a. **Pendidikan**: tingkat pendidikan bisa berhubungan dengan jenis pekerjaan dan tingkat pendapatan seseorang.
b. **Kepemilikan Asuransi**: Kepemilikan asuransi dapat menunjukkan tingkat kehati-hatian dan perencanaan keuangan seseorang.
c. **Jumlah Tanggungan**: Jumlah orang yang bergantung pada pendapatan seseorang (misalnya, jumlah anak atau anggota keluarga lainnya) bisa mempengaruhi kemampuan mereka untuk membayar hutang.
d. **Jumlah Pinjaman**: Jumlah total pinjaman yang masih harus dibayar oleh peminjam.

***"Dalam proses Modelling, Tim Distinct akan menggunakan seluruh fitur data dengan perbandingan antara data yang telah di-grouping dan data awal yang didapat dari Kaggle"***

**3. Git**

Link Repository:
https://github.com/AlyaniNS/Loan-Prediction-Based-on-Customer-Behavior