Rakamin Virtual
Internship X11 Program
Data Scientist

Prediksi Klasifikasi Pembayaran Kredit Customer di Home Credit Indonesia

Faradillah Tsalits

Overview

- Problem Research
- Data Pre-Processing
- Data Visualization and Business Insight
- Machine Learning Implementation and Evaluation
- Business Recommendation

Link Github Repository: https://github.com/Faradillahtsalits10/Home-Credit-Indonesia

Problem Research

Home Credit saat ini sedang menggunakan berbagai macam metode statistik dan Machine Learning untuk membuat **prediksi skor kredit**. Diminta untuk membuka potensi maksimal dari data perusahaan. Dengan melakukannya, dapat **memastikan pelanggan yang mampu melakukan pelunasan tidak ditolak ketika melakukan pengajuan pinjaman, dan pinjaman datap diberikan dengan principal, maturity, dan repayment calendar yang akan memotivsi pelanggan untuk sukses**. Menggunakan setidaknya 2 model Machine Learning dimana salah satunya adalah Logistic Regression.

Objektif

Memprediksi skor kredit pelanggan yang mampu melakukan pelunasan tidak ditolak ketika melakukan pengajuan pinjaman dengan membuat machine learning model, serta memberikan insight yang mendorong tingginya kesulitan pembayaran pinjaman dan rekomendasi untuk mempertahankan setiap pelanggan sebagai upaya untuk mengurangi penolakan pengajuan pinjaman.

Goals

Menurunkan penolakan pengajuan pinjaman dan meminimalisir potensi pelanggan yang tidak dapat melakukan pelunasan pembayaran pinjaman.

Data Pre-Processing

```
[ ]  # Menampilkan data yang duplikat pada data train
     data_train.duplicated().sum()

     0

     # Menampilkan data yang duplikat pada data test
     data_train.duplicated().sum()

     0
```

Cek data duplikat

```
[ ]  train=data_train[['TARGET', 'NAME_CONTRACT_TYPE', 'CODE_GENDER',
            'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN',
            'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE',
            'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
            'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE',
            'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH']]

[ ]  test=data_test[['NAME_CONTRACT_TYPE', 'CODE_GENDER',
            'FLAG_OWN_CAR', 'FLAG_OWN_REALTY', 'CNT_CHILDREN',
            'AMT_INCOME_TOTAL', 'AMT_CREDIT', 'AMT_ANNUITY', 'AMT_GOODS_PRICE',
            'NAME_TYPE_SUITE', 'NAME_INCOME_TYPE', 'NAME_EDUCATION_TYPE',
            'NAME_FAMILY_STATUS', 'NAME_HOUSING_TYPE',
            'REGION_POPULATION_RELATIVE', 'DAYS_BIRTH']]
```

Variabel-variabel yang digunakan

```
[25] # Drop data yang tidak tepat
     train.drop(train.index[train['CODE_GENDER']=='XNA'],inplace=True)
     train.drop(train.index[train['NAME_INCOME_TYPE']=='Maternity leave'],inplace=True)
```

Drop isi data pada variabel yang tidak tepat

```
     # Mencari nilai uniq pada setiap variabel di data train dan data test
     print(train.nunique(), '\n',
           test.nunique())
```

Cek nilai unik variabel

```
[ ]  # Drop kolom yang masih memiliki data null
     train=train.dropna()
     test=test.dropna()
```

Drop kolom yang memiliki missing value

```
[27] AGE_train=(train['DAYS_BIRTH']/-365).astype(int)
     AGE_test=(test['DAYS_BIRTH']/-365).astype(int)
```

Ubah nama atau isi variabel sesuai kebutuhan

Dataset

```
train=train.assign(AGE=AGE_train).drop('DAYS_BIRTH',axis=1)
train.head(5)
```

	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	NAME_TYPE_SUITE	NAME_INCOME_TYPE	NAME_EDUCATION_TYPE	NAME_FAMILY_STATUS	NAME_HOUSING_TYPE	REGION_POPULATION_RELATIVE	AGE
0	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700.5	351000.0	Unaccompanied	Working	Secondary / secondary special	Single / not married	House / apartment	0.018801	25
1	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698.5	1129500.0	Family	State servant	Higher education	Married	House / apartment	0.003541	45
2	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750.0	135000.0	Unaccompanied	Working	Secondary / secondary special	Single / not married	House / apartment	0.010032	52
3	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686.5	297000.0	Unaccompanied	Working	Secondary / secondary special	Civil marriage	House / apartment	0.008019	52
4	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865.5	513000.0	Unaccompanied	Working	Secondary / secondary special	Single / not married	House / apartment	0.028663	54

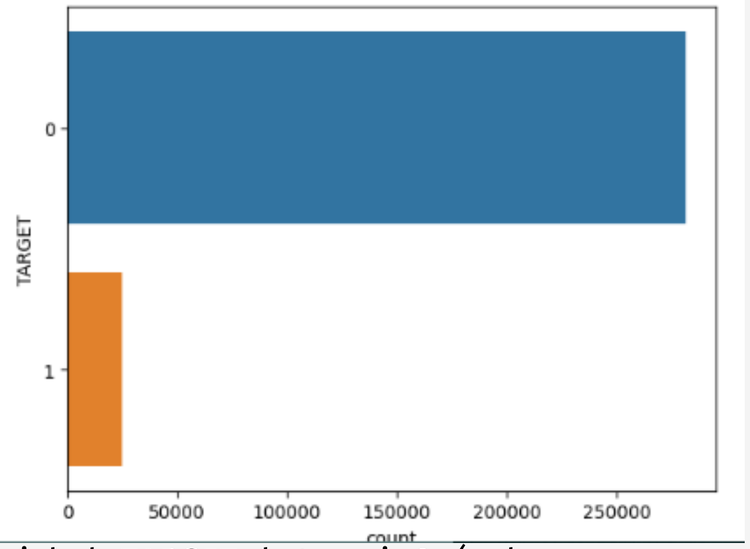
```
test=test.assign(AGE=AGE_test).drop('DAYS_BIRTH',axis=1)
test.head(5)
```

	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_PRICE	NAME_TYPE_SUITE	NAME_INCOME_TYPE	NAME_EDUCATION_TYPE	NAME_FAMILY_STATUS	NAME_HOUSING_TYPE	REGION_POPULATION_RELATIVE	AGE
0	Cash loans	F	N	Y	0	135000.0	568800.0	20560.5	450000.0	Unaccompanied	Working	Higher education	Married	House / apartment	0.018850	52
1	Cash loans	M	N	Y	0	99000.0	222768.0	17370.0	180000.0	Unaccompanied	Working	Secondary / secondary special	Married	House / apartment	0.035792	49
3	Cash loans	F	N	Y	2	315000.0	1575000.0	49018.5	1575000.0	Unaccompanied	Working	Secondary / secondary special	Married	House / apartment	0.026392	38
4	Cash loans	M	Y	N	1	180000.0	625500.0	32067.0	625500.0	Unaccompanied	Working	Secondary / secondary special	Married	House / apartment	0.010032	35
5	Cash loans	F	Y	Y	0	270000.0	959688.0	34600.5	810000.0	Unaccompanied	State servant	Secondary / secondary special	Married	House / apartment	0.025164	50

Data Visualization and Business Insight

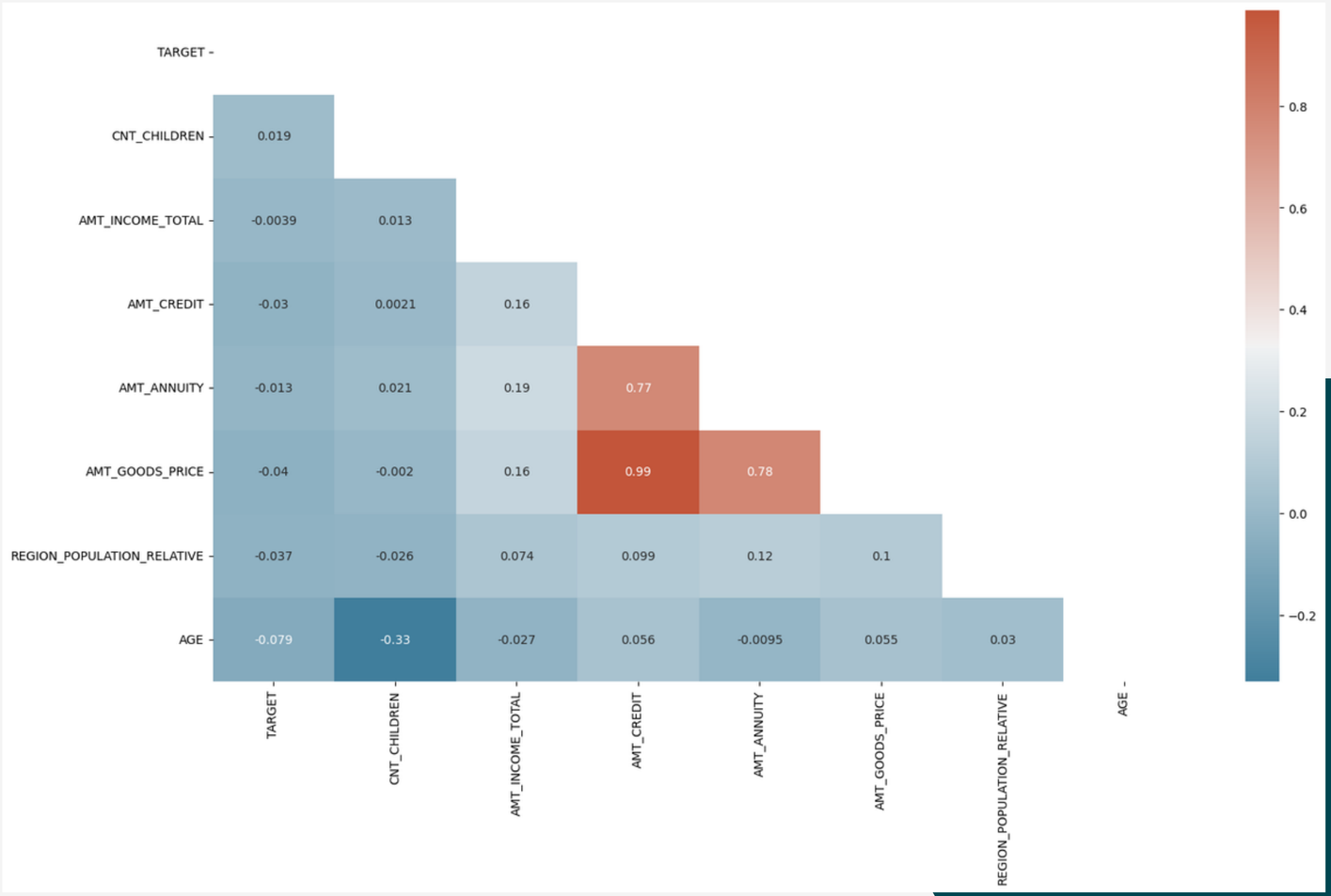
```
# Melihat visualisasi data TARGET
c_palette = ['tab:blue', 'tab:orange','tab:green','tab:grey']
categorical_summarized(train, y = 'TARGET', palette=c_palette)
```

```
count    306198.000000
mean          0.080840
std           0.272589
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max           1.000000
Name: TARGET, dtype: float64
mode:  0    0
Name: TARGET, dtype: int64
============================================================
0    281445
1     24753
Name: TARGET, dtype: int64
```



Variabel TARGET kategori 0 (pelanggan yang tidak memiliki kesulitan pembayaran) sebesar 281445 observasi data lebih banyak dari kategori 1 (pelanggan yang memiliki kesulitan bayaran) sebesar 24753 observasi data

Tabel korelasi menunjukkan variabel CNT_CHILDREN, ANT_INCOME_TOTAL, AMT_CREDIT, AMT_ANNUITY, AMT_GOODS_PRICE, REGION_POPULATION_RELATIVE, dan AGE memiliki nilai korelasi yang rendah terhadap variabel TARGET.



Machine Learning Implementation and Evaluation

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

     X = pd.get_dummies(train.drop(['TARGET'],axis=1))
     Y = train['TARGET']

     X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=60)
     print ('Train set:', X_train.shape, y_train.shape )
     print ('Test set:', X_test.shape,  y_test.shape)

     Train set: (229648, 45) (229648,)
     Test set: (76550, 45) (76550,)
```

Split data 75% data training dan 25% data testing

Tahap modelling, digunakan dua macam metode yaitu:

- **Logistic Regression**
- **Random Forest Classifier**

Hyperparameter tuning yang digunakan adalah Random Search CV.h

```
[104] print(metrics.accuracy_score(y_test, pred_rf)*100)
      print(confusion_matrix(y_test, pred_rf))
      print(classification_report(y_test, pred_rf))

      91.97126061397779
      [[70396    29]
       [ 6117     8]]
                    precision    recall  f1-score   support

                 0       0.92      1.00      0.96     70425
                 1       0.22      0.00      0.00      6125

          accuracy                           0.92     76550
         macro avg       0.57      0.50      0.48     76550
      weighted avg       0.86      0.92      0.88     76550
```

Hasil klasifikasi Random Forest Classifier, nilai akurasi data testing 0.92 atau 92%. Maka, model Random Forest Classifier sangat tepat digunakan untuk prediksi sebab, tingkat kepercayaannya diatas 80%.

```
[101] print(metrics.accuracy_score(y_test, pred_nb)*100)
      print(confusion_matrix(y_test, pred_nb))
      print(classification_report(y_test, pred_nb))

      64.04310907903331
      [[45998 24427]
       [ 3098  3027]]
                    precision    recall  f1-score   support

                 0       0.94      0.65      0.77     70425
                 1       0.11      0.49      0.18      6125

          accuracy                           0.64     76550
         macro avg       0.52      0.57      0.47     76550
      weighted avg       0.87      0.64      0.72     76550
```

Hasil klasifikasi Logistic Regression, nilai akurasi data testing 0.64 atau 64%. Nilai pelanggan yang actual kesulitan pembayaran sebesar 3027 yang lebih kecil dari nilai pelanggan yang diprediksi kesulitan pembayaran padahal tidak sebesar 3098. Maka, model Logistic Regression kurang tepat digunakan untuk prediksi sebab tingkat kepercayaannya dibawah 80%.

Faktor pengaruh kesulitan pembayaran pada pelanggan

```
[106] feat_importances = pd.Series(rf.feature_importances_, index=X_train.columns)
      feat_importances.nlargest(20).plot(kind='barh')
```

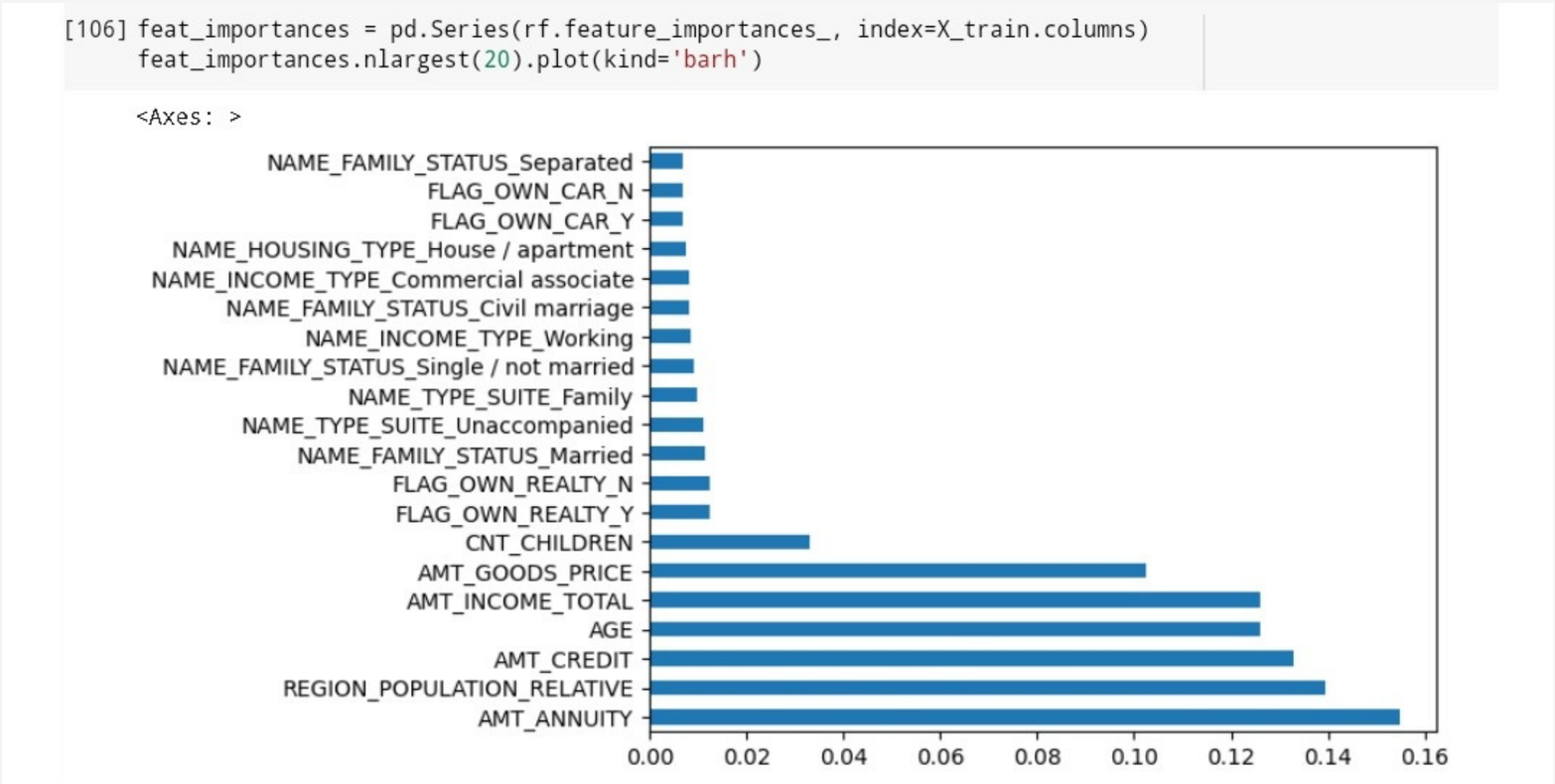


Diagram Faktor kesulitan pembayaran pada Pelanggan dengan menggunakan metode Random Forest Classifier, 3 faktor yang sangat mempengaruhi, yaitu:

- AMT_ANNUITY (Anuitas Pinjaman) 15,5%
- REGION_POPULATION_RELATIVE (Populasi wilayah tempat tinggal pelanggan) 13,9%
- AMT_CREDIT (Jumlah kredit dari pinjeman) 13,2%

Business Recommendation

Rekomendasi yang dapat diberikan kepada perusahaan adalah:

1. Home Credit Indonesia sebaiknya memberikan keringanan batas waktu pembayaran, anuitas pinjaman yang lebih kecil, atau peningkatan batas pinjaman berdasarkan penghasilan pelanggan.
2. Dalam mempertahankan pelanggan yang tidak berpotensi mengalami kesulitan pembayaran dengan memberi perhatian khusus pada pelanggan dengan tipe pinjaman cash loans, sedang bekerja, sudah manikah, dan memiliki rumah atau apartment.
3. Klasifikasi model dengan metode balancing dataset (SMOTE) agar hasil prediksi semakin akurat, sebab jumlah pelanggan yang tidak mengalami kesulitan pembayaran lebih banyak dibandingkn yang mengalami kesulitan pembayaran. Maka dataset dapat dikatakan unstable.