Rakamin Academy

VIRTUAL INTERNSHIP

id/x partners

CREDIT RISK SCORING

By Hanif M Mufid

Abstract

Penelitian ini merupakan sebuah project dari sebuah lending company. Penelitian bertujuan untuk membangun model yang dapat memprediksi credit risk menggunakan dataset yang disediakan company. Hasil dari penelitian ini dapat digunakan untuk mengukur credit risk seseorang yang mengajukan kredit pada lending company

OUTLINE

id/x partners

Rakamin Academy

01 Latar Belakang

02 Data Understanding

03 Data Preparation

04 Model Building

Latar Belakang

Manajemen risiko kredit adalah praktik untuk memitigasi kerugian dengan memahami kecukupan modal bank dan cadangan kerugian pinjaman pada waktu tertentu. Suatu proses yang menjadi tantangan bagi lembaga keuangan.

Salah satu kemajuan teknologi yang saat ini menjadi tren adalah machine learning. Machine learning dianggap menjadi salah satu pendukung kemajuan teknologi untuk segala aspek, khususnya dalam pengolahan data.

Dengan Machine learning ini kita dapat menyediakan solusi teknologi bagi company lending dengan cara membangun model yang dapat memprediksi credit risk.

Dataset

1. Data berisi tentang application loan
2. Data diambil dari tahun 2007 sampai 2014
3. Dataset terdiri dari 466.285 baris dan 75 kolom

Data Understanding

Untuk dapat mengolah data, pemahaman terhadap data sangat diperlukan, agar data dapat diolah dengan tepat. Langkah pertama untuk memahami data adalah dengan melihat data secara keseluruhan, agar mengetahui file yang di-import benar.

```
[1]  import pandas as pd
     import numpy as np

     df = pd.read_csv('/content/drive/MyDrive/Dataset/loan_data_2007_2014.csv')
```

Data Understanding

Tampak data secara keseluruhan

Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	...	total_bal_il	il_util	open_rv_12m	open_rv_24m	max_bal_bc	all_util	total_rev_hi_lim	inq_fi	total_cu_tl	inq_last_12m
0	1077501	1296599	5000	5000	4975.0	36 months	10.65	162.87	B	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1077430	1314167	2500	2500	2500.0	60 months	15.27	59.83	C	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	1077175	1313524	2400	2400	2400.0	36 months	15.96	84.33	C	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	1076863	1277178	10000	10000	10000.0	36 months	13.49	339.31	C	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	1075358	1311748	3000	3000	3000.0	60 months	12.69	67.79	B	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
466280	8598660	1440975	18400	18400	18400.0	60 months	14.47	432.64	C	...	NaN	NaN	NaN	NaN	NaN	NaN	29900.0	NaN	NaN	NaN
466281	9684700	11536848	22000	22000	22000.0	60 months	19.97	582.50	D	...	NaN	NaN	NaN	NaN	NaN	NaN	39400.0	NaN	NaN	NaN
466282	9584776	11436914	20700	20700	20700.0	60 months	16.99	514.34	D	...	NaN	NaN	NaN	NaN	NaN	NaN	13100.0	NaN	NaN	NaN
466283	9604874	11457002	2000	2000	2000.0	36 months	7.90	62.59	A	...	NaN	NaN	NaN	NaN	NaN	NaN	53100.0	NaN	NaN	NaN
466284	9199665	11061576	10000	10000	9975.0	36 months	19.20	367.58	D	...	NaN	NaN	NaN	NaN	NaN	NaN	16000.0	NaN	NaN	NaN

Data Understanding

Kita juga dapat melihat info dari data tersebut.

```
[4]  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 466285 entries, 0 to 466284
Data columns (total 75 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   Unnamed: 0                  466285 non-null  int64
 1   id                          466285 non-null  int64
 2   member_id                   466285 non-null  int64
 3   loan_amnt                   466285 non-null  int64
 4   funded_amnt                 466285 non-null  int64
 5   funded_amnt_inv             466285 non-null  float64
 6   term                        466285 non-null  object
 7   int_rate                    466285 non-null  float64
 8   installment                 466285 non-null  float64
 9   grade                       466285 non-null  object
 10  sub_grade                   466285 non-null  object
 11  emp_title                   438697 non-null  object
 12  emp_length                  445277 non-null  object
 13  home_ownership              466285 non-null  object
 14  annual_inc                  466281 non-null  float64
 15  verification_status         466285 non-null  object
 16  issue_d                     466285 non-null  object
 17  loan_status                 466285 non-null  object
 18  pymnt_plan                  466285 non-null  object
 19  url                         466285 non-null  object
 20  desc                        125983 non-null  object
 21  purpose                     466285 non-null  object
 22  title                       466265 non-null  object
 23  zip_code                    466285 non-null  object
 24  addr_state                  466285 non-null  object
 25  dti                         466285 non-null  float64
 26  delinq_2yrs                 466256 non-null  float64
 27  earliest_cr_line            466256 non-null  object
 28  inq_last_6mths              466256 non-null  float64
 29  mths_since_last_delinq      215934 non-null  float64
 30  mths_since_last_record      62638 non-null   float64
 31  open_acc                    466256 non-null  float64
 32  pub_rec                     466256 non-null  float64
 33  revol_bal                   466285 non-null  int64
 34  revol_util                  465945 non-null  float64
 35  total_acc                   466256 non-null  float64
 36  initial_list_status         466285 non-null  object
 37  out_prncp                   466285 non-null  float64
 38  out_prncp_inv               466285 non-null  float64
 39  total_pymnt                 466285 non-null  float64
 40  total_pymnt_inv             466285 non-null  float64
 41  total_rec_prncp             466285 non-null  float64
 42  total_rec_int               466285 non-null  float64
 43  total_rec_late_fee          466285 non-null  float64
 44  recoveries                  466285 non-null  float64
 45  collection_recovery_fee     466285 non-null  float64
 46  last_pymnt_d                465909 non-null  object
 47  last_pymnt_amnt             466285 non-null  float64
 48  next_pymnt_d                239071 non-null  object
 49  last_credit_pull_d          466243 non-null  object
 50  collections_12_mths_ex_med  466140 non-null  float64
 51  mths_since_last_major_derog 98974 non-null   float64
 52  policy_code                 466285 non-null  int64
 53  application_type            466285 non-null  object
 54  annual_inc_joint            0 non-null       float64
 55  dti_joint                   0 non-null       float64
 56  verification_status_joint   0 non-null       float64
 57  acc_now_delinq              466256 non-null  float64
 58  tot_coll_amt                396009 non-null  float64
 59  tot_cur_bal                 396009 non-null  float64
 60  open_acc_6m                 0 non-null       float64
 61  open_il_6m                  0 non-null       float64
 62  open_il_12m                 0 non-null       float64
 63  open_il_24m                 0 non-null       float64
 64  mths_since_rcnt_il          0 non-null       float64
 65  total_bal_il                0 non-null       float64
 66  il_util                     0 non-null       float64
 67  open_rv_12m                 0 non-null       float64
 68  open_rv_24m                 0 non-null       float64
 69  max_bal_bc                  0 non-null       float64
 70  all_util                    0 non-null       float64
 71  total_rev_hi_lim            396009 non-null  float64
 72  inq_fi                      0 non-null       float64
 73  total_cu_tl                 0 non-null       float64
 74  inq_last_12m                0 non-null       float64
dtypes: float64(46), int64(7), object(22)
memory usage: 266.8+ MB
```

Data Preparation

Berdasarkan info yang sudah dilihat, terdapat kolom yang tidak mempunyai data sama sekali dan perlu dihapus.

```
[8]  df = df.drop(['id', 'member_id', 'Unnamed: 0', 'open_acc_6m', 'open_il_6m', 'open_il_12m', 'open_il_24m',
                   'mths_since_rcnt_il', 'total_bal_il', 'il_util', 'open_rv_12m', 'open_rv_24m', 'max_bal_bc',
                   'all_util', 'annual_inc_joint', 'dti_joint', 'verification_status_joint', 'inq_fi', 'total_cu_tl',
                   'inq_last_12m'], axis = 1)
```

Data Preparation

Cek missing value dan menghapus kolom dengan missing value lebih dari 40%

```
[20] missing_values = pd.DataFrame(df.isnull().sum()/df.shape[0])
     missing_values_filter = missing_values[missing_values[0] > 0.40]
```

```
[21] missing_values_filter.sort_values(0, ascending = False)
```

	0
mths_since_last_record	0.866724
mths_since_last_major_derog	0.786480
desc	0.733440
mths_since_last_delinq	0.537503
next_pymnt_d	0.490176

Data Preparation

Membuat target label untuk klasifikasi

```
df['good_bad'] = np.where(df.loc[:, 'loan_status'].isin(['Charged Off', 'Default', 'Late (31-120 days)',
                                                         'In Grace Period', 'Late (16-30 days)']), 1, 0)

df['good_bad']
```

```
0         0
1         1
2         0
3         0
4         0
         ..
466280    0
466281    1
466282    0
466283    0
466284    0
Name: good_bad, Length: 463536, dtype: int64
```

Model Building

Setelah meng handle missing value dan dilakukan One-Hot encoding pada dataset, bentuk dataset menjadi memiliki 464.536 baris dan 149 kolom. Dataset ini yang akan digunakan untuk membuat model.

```
loan_data_dum = pd.get_dummies(loan_data)
loan_data_dum.shape
```

```
(463536, 149)
```

Model Building

Dilakukan splitting data untuk membagi data menjadi train dan test data

```
from sklearn.model_selection import train_test_split

X = loan_data_dum.drop('good_bad', axis = 1)
y = loan_data_dum['good_bad']

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

Model Building

Pada gambar dilihat tingkat akurasi pada prediksi yang dimiliki oleh model ini adalah 96%

```
from sklearn.linear_model import LogisticRegression

model_1 = LogisticRegression()
```

```
model_1.fit(X_train, y_train)
```

```
y_pred = model_1.predict(X_test)
```

```
result = pd.DataFrame(list(zip(y_pred, y_test)), columns = ['y_pred', 'y_test'])
```

```
result.head()
```

	y_pred	y_test
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

```
0.9693446088794926
```

Thank You