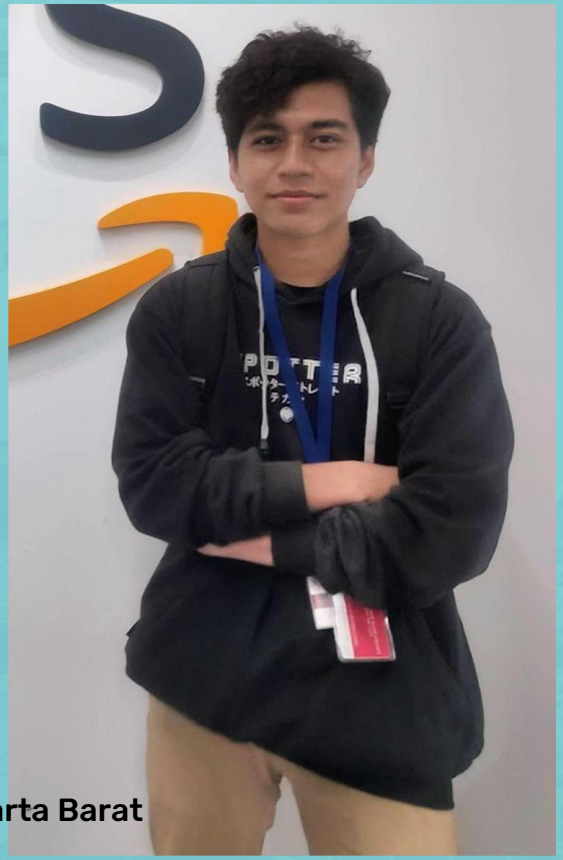
Rakamin Academy X id/x partners

Credit Risk Prediction Model

ID/X Partners - Data Scientist

Presented by
Muhammad Nazaruddin

Rakamin Academy



Muhammad Nazaruddin

Data Scientist

Lulusan Informatika kampus Gunadarma, dengan ketertarikan mendalam pada Machine Learning dan Data Scientist. Saya memiliki sertifikasi resmi TensorFlow Developer dalam bidang Data Scientist. Saya terbiasa menggunakan tools seperti Python, TensorFlow, dan Darknet. Memiliki pengalaman dalam bekerja secara dalam tim maupun mandiri. Saya berpengalaman dalam ikut membuat projek Object Detection, System Recommendation, Sales Forecasting, Price Optimization dan selain itu memiliki pengalaman dalam AWS CLoud.

Jakarta Barat

Muhammad Nazaruddin

muhh.nash@gmail.com

About Company

ID/X Partners didirikan pada tahun 2002 oleh mantan bankir dan konsultan manajemen yang memiliki pengalaman luas dalam siklus dan proses kredit, pengembangan scoring, serta manajemen kinerja. Pengalaman gabungan mereka telah melayani berbagai perusahaan di kawasan Asia dan Australia, serta di berbagai industri seperti jasa keuangan, telekomunikasi, manufaktur, dan ritel.

ID/X Partners menyediakan layanan konsultasi yang mengkhususkan diri dalam pemanfaatan analitik data dan solusi pengambilan keputusan (DAD) yang dikombinasikan dengan disiplin manajemen risiko dan pemasaran terintegrasi untuk membantu klien mengoptimalkan profitabilitas portofolio dan proses bisnis mereka.

Layanan konsultasi yang komprehensif dan solusi teknologi yang ditawarkan menjadikan id/x partners sebagai penyedia layanan yang lengkap dan terpercaya

Project Portfolio

Proyek ini bertujuan untuk memahami dan menganalisis data peminjaman (loan), dalam hal ini dataset yang digunakan adalah data pinjaman yang diberikan kepada anggota. Tujuan utama proyek ini adalah mengidentifikasi faktor-faktor yang berkontribusi terhadap pinjaman bermasalah (bad loans), yaitu pinjaman yang gagal bayar atau tidak dilunasi sesuai ketentuan. Oleh karena itu, tujuan analisis ini adalah:

1. Mengklasifikasikan mana pinjaman yang termasuk bad loan (gagal bayar) berdasarkan status pinjaman (loan_status) menggunakan label target bad_loan.
2. Menganalisis pola dan karakteristik peminjam yang cenderung memiliki bad loan.
3. Memberikan wawasan untuk membantu pengambilan keputusan lebih baik dalam pemberian pinjaman dengan memprediksi kemungkinan risiko kegagalan pembayaran.

Link code here!

Project explanation video here!

1. Data Understanding

Membaca Dataset

Dataset terdiri dari 466.285 baris dan 75 kolom dengan tipe data campuran (numerik dan kategorikal). Beberapa kolom memiliki banyak nilai kosong, bahkan ada yang sepenuhnya kosong. Informasi ini menunjukkan perlunya pembersihan data sebelum analisis lebih lanjut.

	Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	...	total_bal_il
0	0	1077501	1296599	5000	5000	4975.0	36 months	10.65	162.87	B	...	NaN
1	1	1077430	1314167	2500	2500	2500.0	60 months	15.27	59.83	C	...	NaN
2	2	1077175	1313524	2400	2400	2400.0	36 months	15.96	84.33	C	...	NaN
3	3	1076863	1277178	10000	10000	10000.0	36 months	13.49	339.31	C	...	NaN
4	4	1075358	1311748	3000	3000	3000.0	60 months	12.69	67.79	B	...	NaN

5 rows × 75 columns

```
<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
```

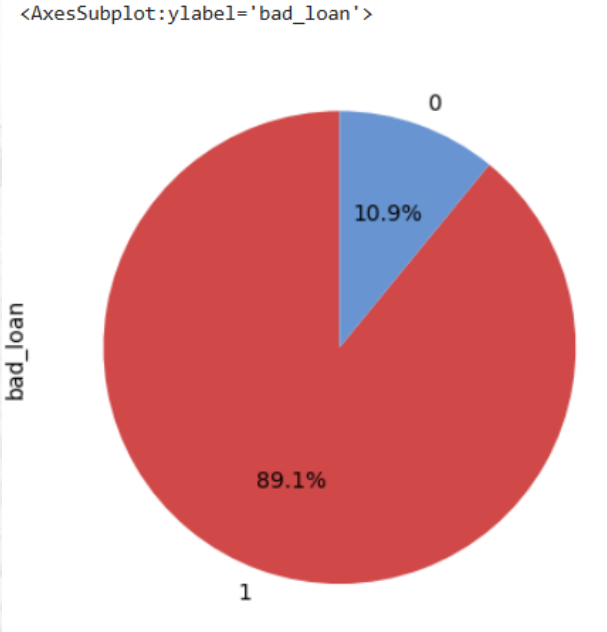
- loan_amnt: Jumlah pinjaman.
- funded_amnt: Jumlah pendanaan yang disetujui.
- term: Durasi pinjaman (misalnya 36 atau 60 bulan).
- int_rate: Suku bunga pinjaman.
- installment: Pembayaran angsuran bulanan.
- grade & sub_grade: Klasifikasi risiko kredit dari peminjam.
- emp_title, emp_length: Informasi pekerjaan peminjam.
- home_ownership: Jenis kepemilikan tempat tinggal.
- annual_inc: Penghasilan tahunan peminjam.
- verification_status: Status verifikasi penghasilan.
- purpose: Tujuan pinjaman.
- addr_state: Negara bagian tempat peminjam tinggal.
- dti: Rasio utang terhadap penghasilan (Debt-to-Income Ratio).
- loan_status: Status akhir dari pinjaman.

Menentukan Target Variabel

Kolom target untuk klasifikasi adalah loan_status. Loan Status sendiri adalah status akhir dari pinjaman (seperti “Sudah membayar", “Belum/gagal membayar")

Perbandingan dari data yang sudah membayar dan belum membayar adalah :
10,9% sudah
89,1% belum

Dari Kolom target data bad_loan ini maka hal yang menjadi tujuan utama adalah mengidentifikasi faktor-faktor yang memengaruhi pinjaman bermasalah (bad loans).



2. Feature Engineering

Melakukan Transformasi Variabel

Fitur	Transformasi
`bad_loan`	Membuat kolom target berdasarkan `loan_status`.
`term`	Konversi dari string ke numerik.
`emp_length`	Konversi ke numerik dan penyimpanan dalam kolom baru `emp_length_int`.
`issue_d`, `earliest_cr_line`, `last_pymnt_d`, `last_credit_pull_d`	Konversi ke datetime dan pembuatan kolom baru dalam satuan bulan (`mnth_since_*`).
Missing Values	Diisi dengan mean (numerik) atau mode (kategorikal), serta penghapusan kolom dengan missing values > 40%.
Kolom tidak relevan	Dihapus dari dataset.

ini bertujuan untuk membersihkan data dan membuatnya lebih mudah digunakan dalam model prediksi atau analisis lanjutan.

Mengatasi Missing Variabel

Jenis Data	Metode Imputasi	Alasan
Numerik	Mean	Menjaga stabilitas distribusi data
Kategorikal	Mode	Nilai yang paling sering muncul

Cara untuk mengatasi missing value pada dataframe bisa dengan melakukan imputasi yaitu mengisi nilai yang hilang dengan nilai tertentu.

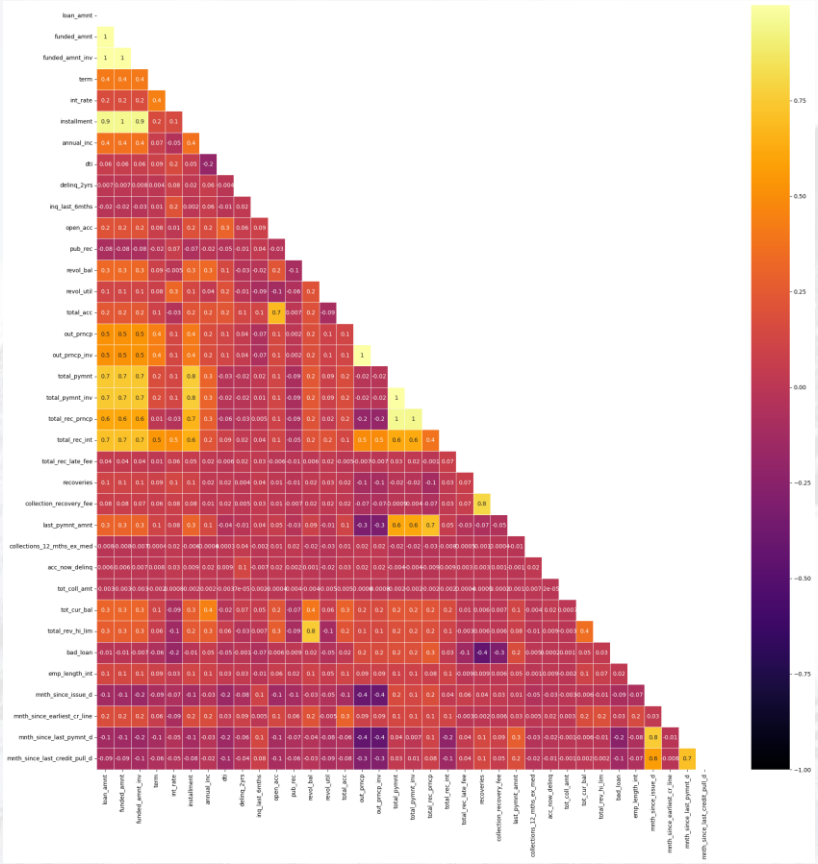
Imputasi Data Hilang pada Kolom Numerik
Metode: Menggunakan rata-rata (*mean*).
Kenapa mean? Untuk menjaga distribusi nilai tetap stabil dan tidak mengganggu analisis statistik.

Imputasi Data Hilang pada Kolom Kategorikal
Metode: Menggunakan modus (*mode*), yaitu nilai yang paling sering muncul.
Kenapa mode? Cocok untuk data diskrit atau kategorikal karena merupakan nilai yang paling umum.

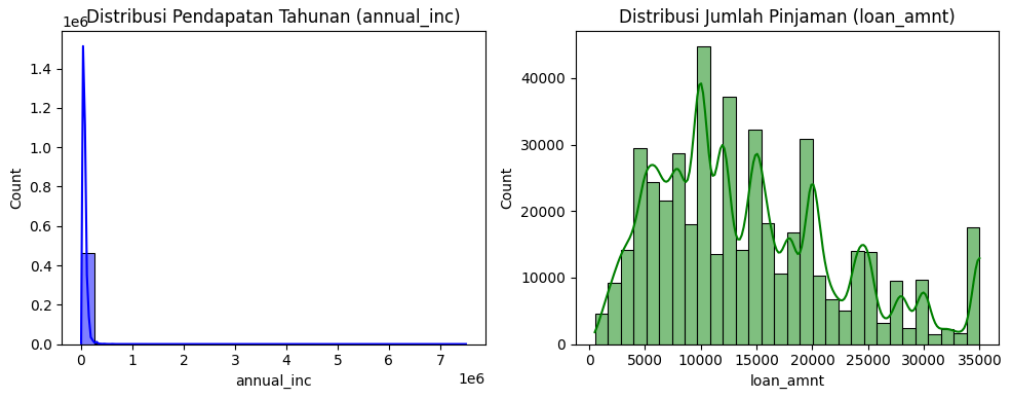
3. Exploratory Data Analysis

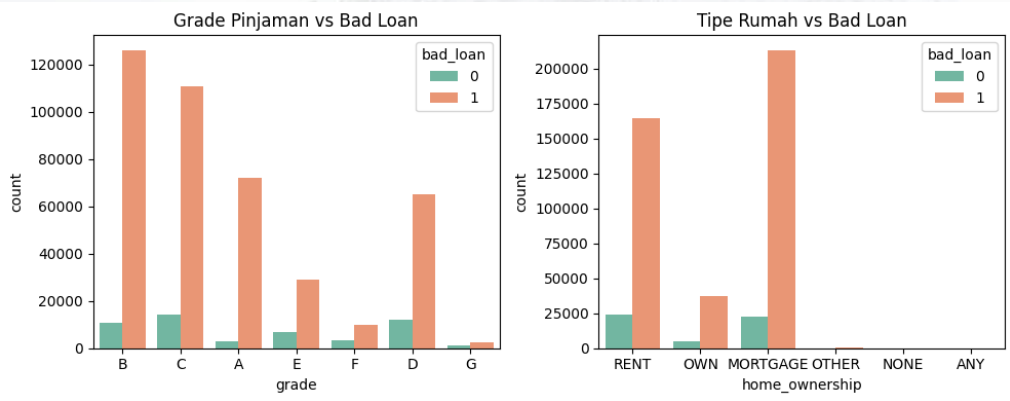
Rakamin Academy

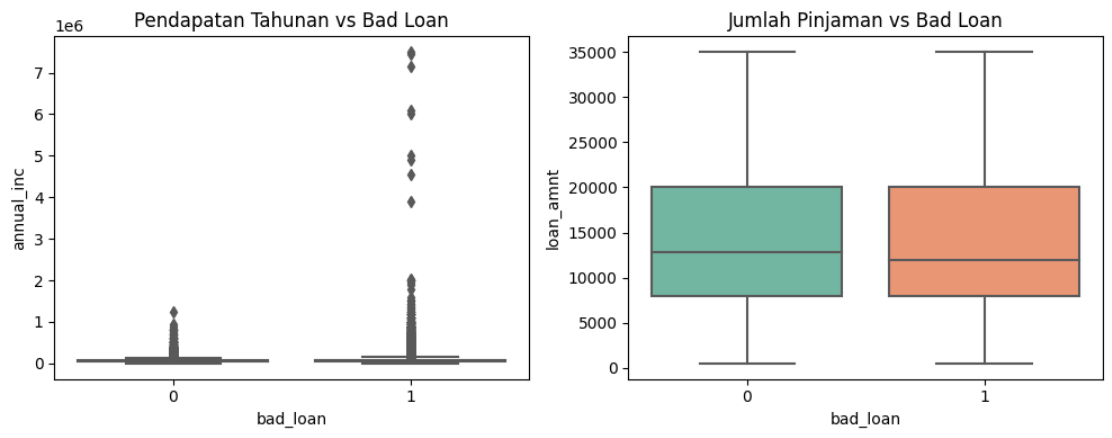
Memeriksa kolerasi antar variable, jika tidak terdapat kolerasi maka akan dihapus

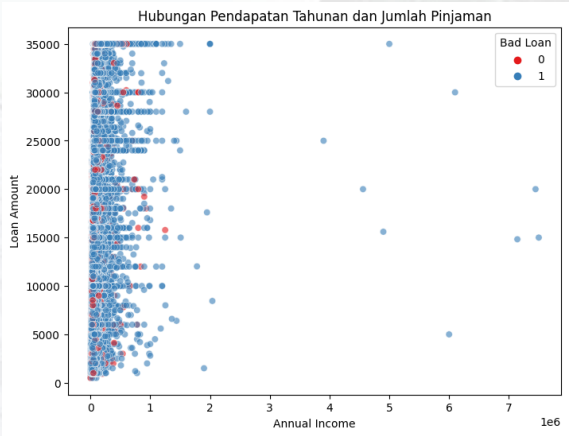


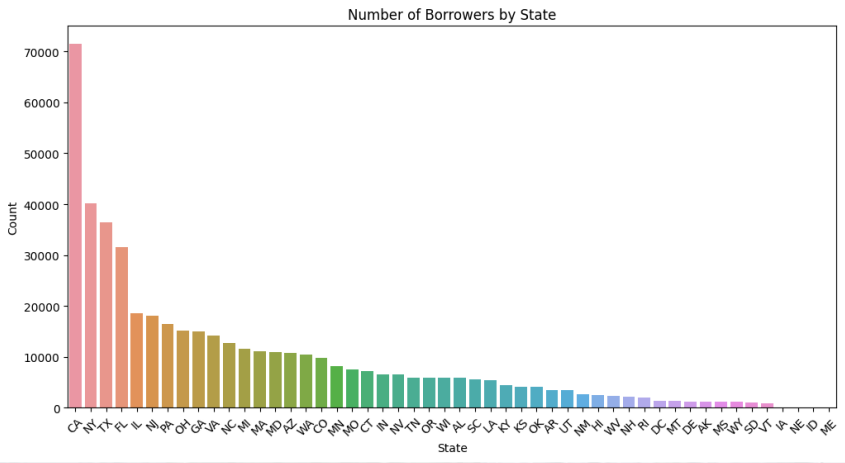
Mendeteksi Skweness dan Outlier

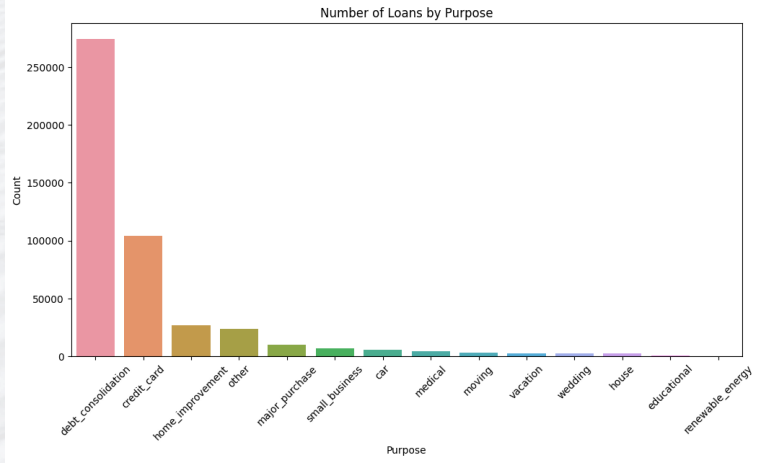





Pendapatan Tahunan vs Bad Loan
1e6
annual_inc
bad_loan
0
1
Jumlah Pinjaman vs Bad Loan
loan_amnt
bad_loan
0
1


Hubungan Pendapatan Tahunan dan Jumlah Pinjaman
Bad Loan
0
1
Loan Amount
Annual Income
1e6


Number of Borrowers by State
Count
State
0
10000
20000
30000
40000
50000
60000
70000
CA NY TX FL IL NJ PA OH GA VA NC MI MA MD AZ WA CO MN MO CT IN NV TN OR WI AL SC LA KY KS OK AR UT NM HI WV NH RI DC MT DE AK MS WY SD VT IA NE ID ME


Number of Loans by Purpose
Count
Purpose
0
50000
100000
150000
200000
250000
debt_consolidation
credit_card
home_improvement
other
major_purchase
small_business
car
medical
moving
vacation
wedding
house
educational
renewable_energy

EDA Menggunakan Metode WoE

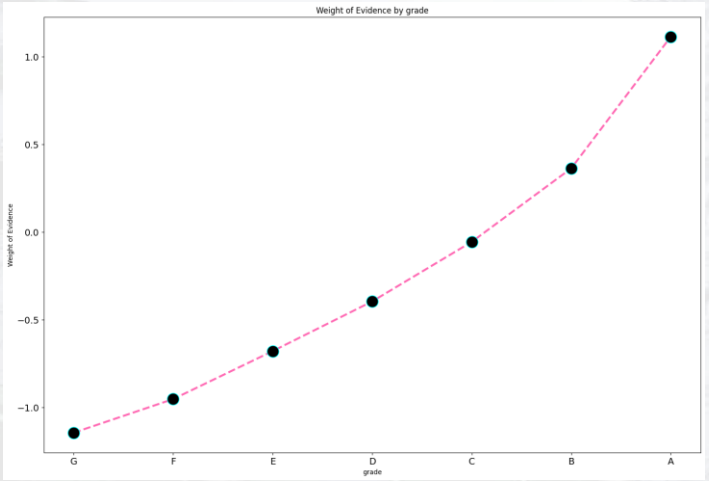
Weight of Evidence (WoE) digunakan untuk mengevaluasi hubungan antara variabel independen dengan target mengukur kekuatan suatu bin (kelompok data) dalam membedakan antara customer yang baik dan buruk. Nilai WoE < 0 menunjukkan bahwa bin variabel tersebut menangkap proporsi akun buruk yang lebih tinggi.

$$WoE = [ln(\frac{\text{Relative frequecy of Goods}}{\text{Relative frequecy of Bads}})] * 100$$

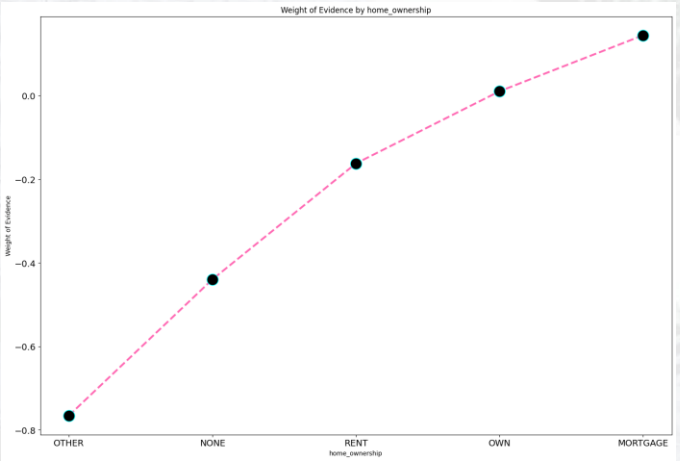
Information Value (IV) digunakan untuk mengevaluasi prediktifitas variabel independen terhadap target. IV (Information Value) adalah ukuran kekuatan prediktif keseluruhan dari suatu variabel dan sangat berguna untuk proses seleksi fitur.

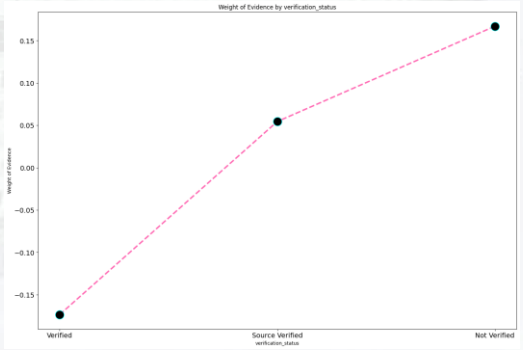
$$IV = \sum(DistributionGood_i - DistributionBad_i) * WoE_i$$

- grade variable

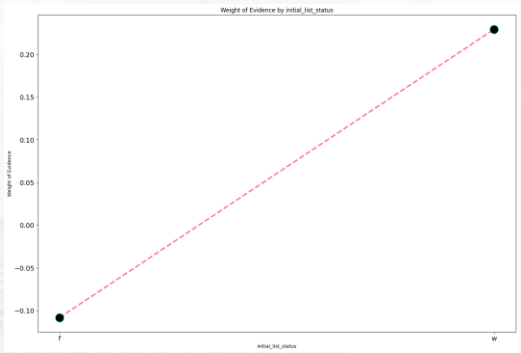


- home ownership variable

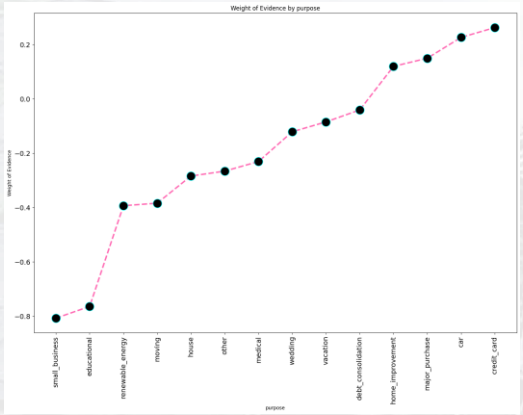




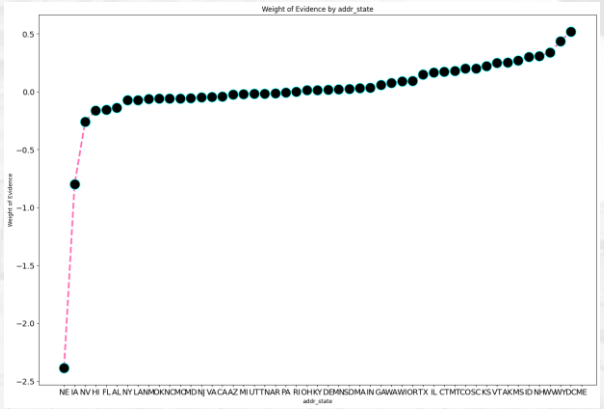
- Verification status



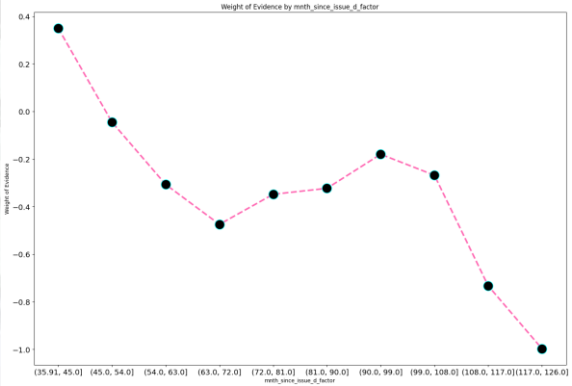
- initial list status



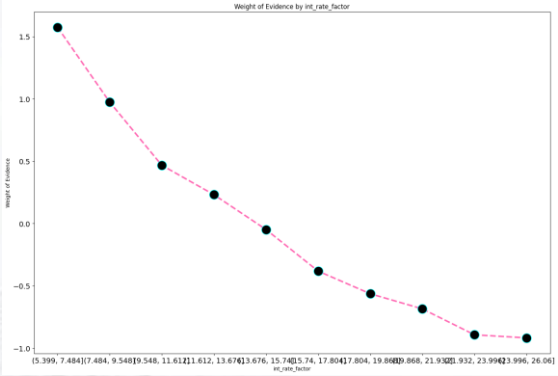
- Purpose



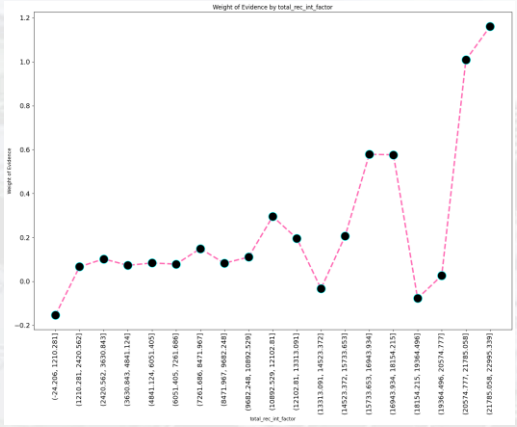
- Address state



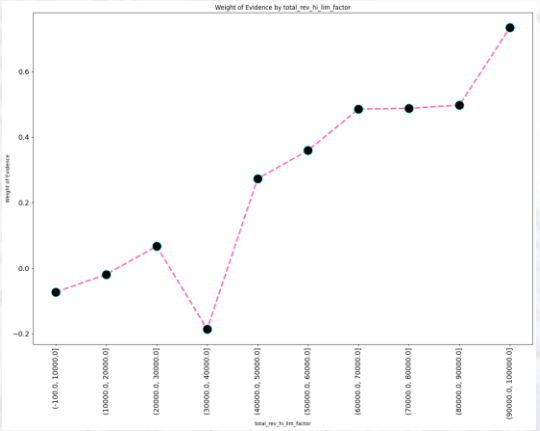
- months since issued



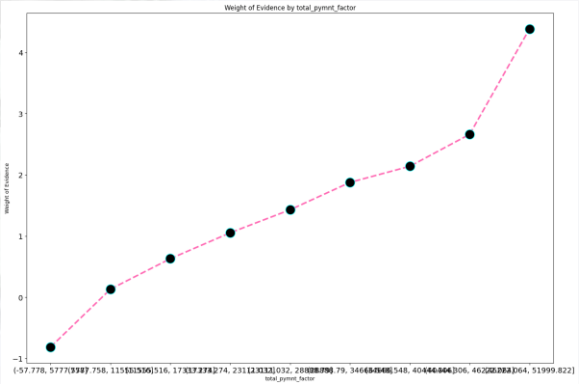
- Interest rate



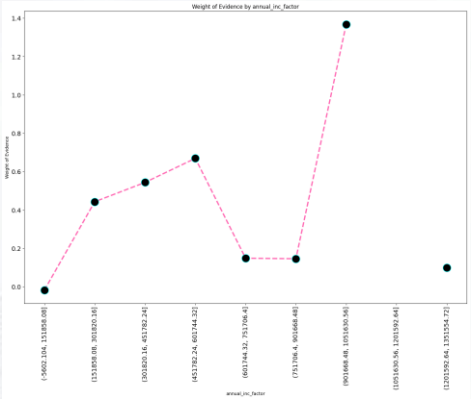
- total_rec_int



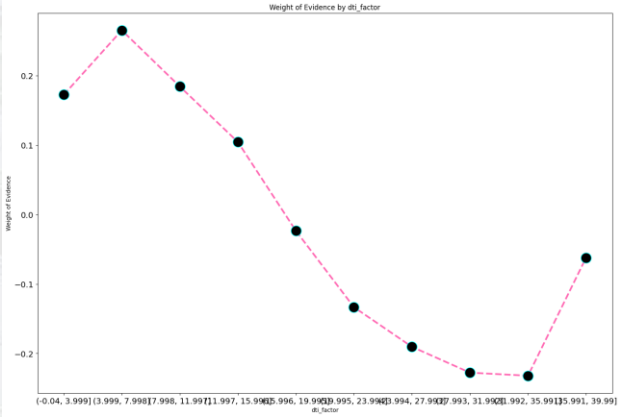
- revolving_high_limit



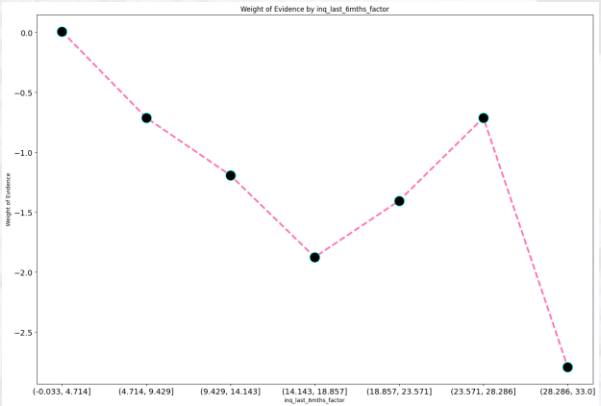
- total_payment



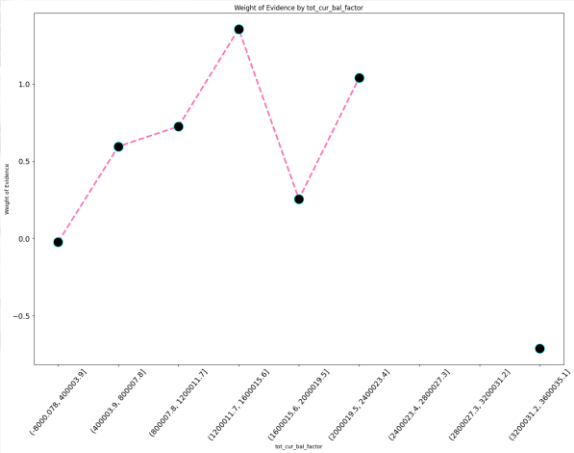
- annual_income



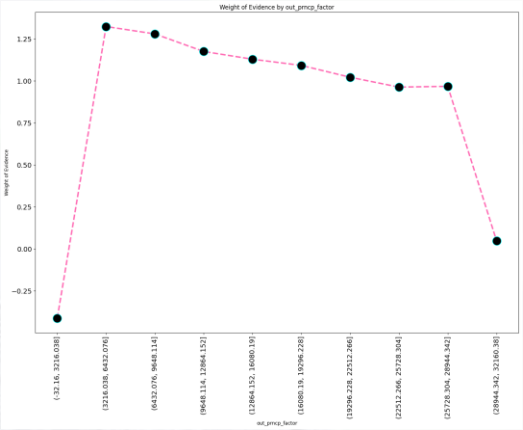
- dti



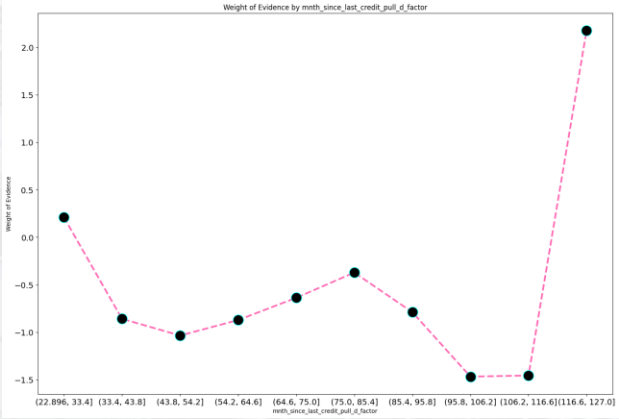
- Income last 6 months



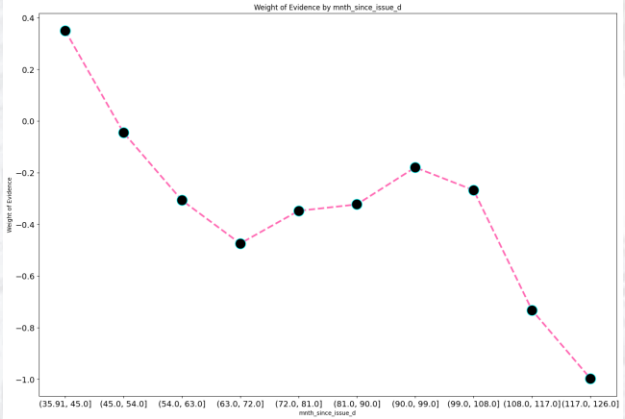
- Total current balance



- Outstanding principal



- Months since last credit pulled



- months since last issued

4. Data Preparation

Drop Data yang memiliki Low IV

Range: 0-1	Predictive powers
IV < 0.02	No power
0.02 < IV < 0.1	Weak power
0.1 < IV < 0.3	Medium power
0.3 < IV < 0.5	Strong power
0.5 < IV	Suspiciously high, too good to be true

Sebagai aturan, semua variabel dengan nilai IV < 0,02 dianggap tidak berguna untuk prediksi, sedangkan variabel dengan IV > 0,5 memiliki kekuatan prediktif yang mencurigakan. Oleh karena itu, variabel-variabel berikut tidak akan disertakan: out_prncp, last_pymnt_amnt, delinq_2yrs, mths_since_last_delinq, open_acc, pub_rec, total_acc, collections_12_mths_ex_med, acc_now_delinq, tot_coll_amt, dan mths_since_last_pymnt_d.

```
Information value of loan_amnt is 0.004021
Information value of funded_amnt is 0.005335
Information value of funded_amnt_inv is 0.008519
Information value of term is 0.03886
Information value of int_rate is 0.330892
Information value of installment is 0.00716
Information value of grade is 0.290782
Information value of home_ownership is 0.021672
Information value of annual_inc is 0.055095
Information value of verification_status is 0.020831
Information value of purpose is 0.03698
Information value of addr_state is 0.012518
Information value of dti is 0.026295
Information value of delinq_2yrs is 4.5e-05
Information value of inq_last_6mths is 0.036969
Information value of open_acc is 0.000743
Information value of pub_rec is 0.00058
Information value of revol_bal is 0.006575
Information value of revol_util is 0.027617
Information value of total_acc is 0.007425
Information value of initial_list_status is 0.024801
Information value of out_prncp is 0.76565
Information value of out_prncp_inv is 0.765591
Information value of total_pymnt is 0.641949
Information value of total_pymnt_inv is 0.647435
Information value of total_rec_prncp is 1.490755
Information value of total_rec_int is 0.025457
Information value of total_rec_late_fee is 0.0
Information value of recoveries is 6.156862
Information value of collection_recovery_fee is 0.0
Information value of last_pymnt_amnt is 1.549867
Information value of collections_12_mths_ex_med is 0.000593
Information value of acc_now_delinq is 5.2e-05
Information value of tot_coll_amt is 0.031008
Information value of tot_cur_bal is 0.060884
Information value of total_rev_hi_lim is 0.055222
Information value of emp_length_int is 0.006028
Information value of mnth_since_issue_d is 0.11029
Information value of mnth_since_earliest_cr_line is 0.014745
Information value of mnth_since_last_pymnt_d is 2.036035
Information value of mnth_since_last_credit_pull_d is 0.227195
```

Drop Not Use Variabel & Missing Value

```
hapus data yang tidak relevan
```

```
col = ['id','member_id','url','sub_grade','zip_code']
data.drop(col, axis=1, inplace= True)
```

Menghapus variable yang tidak membantu dalam pelatihan model nantinya.

```
missing_values = data.isna().mean()*100
col_missingvalues = missing_values[missing_values > 40].index
col_missingvalues
```

```
Index(['desc', 'mths_since_last_delinq', 'mths_since_last_record',
       'next_pymnt_d', 'mths_since_last_major_derog', 'annual_inc_joint',
       'dti_joint', 'verification_status_joint', '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',
       'inq_fi', 'total_cu_tl', 'inq_last_12m'],
      dtype='object')
```

```
hapus data dengan missingvalue mendekati 50%
```

Menghapus variable yang memiliki terlalu banyak missing value agar tidak menghasilkan prediksi yang ambigu

Implementasi Fitur Baru Menggunakan Hasil WoE

Sebelumnya sudah dilakukan Eksplorasi Data Analisis mengunakan metode WoE, kemudian pada tahap ini Data frame dapat dikembangkan lagi dengan cara mengelompokan Variabel dan melakukan Encoding.

- **Grade**; mengubah nilai grade yang sebelumnya object menjadi data binnery

	grade:A	grade:B	grade:C	grade:D	grade:E	grade:F	grade:G
0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	0	1	0	0	0	0

- **Home ownership**; Kategori OTHER, NONE, dan ANY memiliki sangat sedikit observasi dan sebaiknya digabungkan dengan kategori yang memiliki risiko gagal bayar tinggi, yaitu RENT.

home_ownership:OWN	home_ownership:OTHER_NONE_RENT_ANY	home_ownership:MORTGAGE
0	1	0
0	1	0
0	1	0

- **verification_status**; membagi semua unique value menjadi kolom tersendiri NOT_VERIFIED, SOURCE_VERIFIED, dan VERIFIED

- **purpose**; mengelompokan kolom menjadi:

- SMALL_BUSINESS_EDUCATIONAL_RENEWABLE_ENERGY_MOVING
- OTHER_HOUSE_MEDICAL
- WEDDING_VACATION
- HOME_IMPROVEMENT_MAJOR_PURCHASE
- CAR_CREDIT_CARD

- **addr_state**; Dapat digabungkan menjadi satu kategori baru. Proses ini akan digunakan untuk analisis selanjutnya. Adapun kategori-kategori yang akan digabungkan adalah sebagai berikut:

- NE, IA, NV, HI, FL, AL
- NY
- LA, NM, OK, NC, MO, MD, NJ, VA
- CA
- AZ, MI, UT, TN, AR, PA
- RI, OH, KY, DE, MN, SD, MA, IN
- GA, WA
- WI, OR
- TX
- IL, CT,MT
- CO, SC
- KS, VT, AK, MS
- NH, WV, WY, DC

- **initial_list_status**; Memiliki nilai WoE yang sangat berbeda antar kategori, sehingga setiap kategori sebaiknya diperlakukan sebagai variabel terpisah F dan W.

- **term**; memisahkan variabel term yang hanya memiliki nilai 36 dan 60

- **total_rec_int**; Dibuat berdasarkan nilai WoE dan jumlah observasi

  - <1000
  - 1000-2000
  - 2000-9000
  - >9000

- **total_rev_hi_lim**; Dibuat berdasarkan nilai WoE dan jumlah observasi

  - <10000
  - 10000-20000
  - 20000-40000
  - 40000-60000
  - 60000-80000
  - 80000-100000
  - <100000

- **out_prncp**; Dibuat berdasarkan nilai WoE dan jumlah observasi

  - <3000
  - 3000-6000
  - 6000-10000
  - 10000-12000
  - >12000

- **total_pymnt**; Dibuat berdasarkan nilai WoE dan jumlah observasi

  - <5000
  - 5000-11000
  - 11000-16000
  - 16000-22000
  - >22000

- **int_rate**; Dibuat berdasarkan nilai WoE dan jumlah observasi

  - <7.484
  - 7.484 - 9.548
  - 9.548 - 11.612
  - 11.612 - 13.676
  - 13.676 - 15.74
  - 15.74 - 17.804
  - 17.804 - 19.868
  - 7.19.868 - 21.932
  - 21.932 - 26.06

- **dti**; Dibuat berdasarkan nilai WoE dan jumlah observasi

  - (<4)
  - (4-8)
  - (8– 12)
  - (12 – 16)
  - (16 – 20)
  - (20 – 23)
  - (23 – 27)
  - (27 – 40)

- **annual_inc**; Dibuat berdasarkan nilai WoE dan jumlah observasi

  - <32000
  - 32000-50000
  - 50000-60000
  - 60000-75000
  - 75000-90000
  - 90000-120000
  - 120000-135000
  - 135000-150000
  - >150000

- **inq_last_6mths**; Dibuat berdasarkan nilai WoE dan jumlah observasi

- <1
- 1-2
- 2-3
- 4-7

- **tot_cur_bal**; Dibuat berdasarkan nilai WoE dan jumlah observasi

- <40000
- 40000-80000
- 80000-120000
- 120000-160000
- 160000-200000
- 200000-240000
- 240000-320000
- 320000-400000
- >400000

- **mnth_since_last_credit_pull_d**; Dibuat berdasarkan nilai WoE dan jumlah observasi

- <65
- 65 - 76
- >76

- **mnth_since_issue_d**; Dibuat berdasarkan nilai WoE dan jumlah observasi

- (67.97, 70.8)
- (70.8, 73.6)
- (73.6- 76.4)
- (76.4.- 79.2)
- (79.2 - 82)
- (82 - 84)
- (84 - 90.4)
- (90.4 - 96)

5. Data Modeling dan Evaluasi

Memisahkan data training dan data testing

```
X = df_train1.drop(columns='bad_loan', axis=1)
y = df_train1['bad_loan']
```

```
# Membagi data menjadi training dan testing set
x_train, x_test, y_train, y_test = train_test_split(X, y, random_state=42)

# Ubah semua -1.0 menjadi 0 agar kompatibel dengan semua model (terutama XGBoost)
y_train = y_train.replace(-1.0, 0)
y_test = y_test.replace(-1.0, 0)
```

5. Data Modeling dan Evaluasi

=> Random Forest Model

```
# Random Forest Classifier
randof = RandomForestClassifier(random_state = 42)
randof.fit(x_train, y_train)
y_pred = randof.predict(x_test)
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.74	0.34	0.46	12750
1	0.92	0.99	0.95	103822
accuracy			0.91	116572
macro avg	0.83	0.66	0.71	116572
weighted avg	0.90	0.91	0.90	116572

=> Decision Tree Model

```
# Decision Tree Classifier
dtree = DecisionTreeClassifier(random_state = 42)
dtree = dtree.fit(x_train, y_train)
y_pred = dtree.predict(x_test)
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.41	0.44	0.42	12750
1	0.93	0.92	0.93	103822
accuracy			0.87	116572
macro avg	0.67	0.68	0.67	116572
weighted avg	0.87	0.87	0.87	116572

=> K-Nearest Neighbors (KNN)

```
# K-Nearest Neighbors
knn = KNeighborsClassifier()
knn.fit(x_train, y_train)
y_pred = knn.predict(x_test)
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.56	0.24	0.33	12750
1	0.91	0.98	0.94	103822
accuracy			0.90	116572
macro avg	0.73	0.61	0.64	116572
weighted avg	0.87	0.90	0.88	116572

=> XGBoost Model

```
# Now train the XGBoost model
xgb = XGBClassifier(random_state=42)
xgb.fit(x_train, y_train)
y_pred = xgb.predict(x_test)
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.77	0.35	0.49	12750
1	0.93	0.99	0.96	103822
accuracy			0.92	116572
macro avg	0.85	0.67	0.72	116572
weighted avg	0.91	0.92	0.90	116572

=> Logistic Regression Model

```
# Logistic Regression
from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression(max_iter=1000, random_sta
logreg.fit(x_train, y_train)
y_pred = logreg.predict(x_test)
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.74	0.31	0.43	12750
1	0.92	0.99	0.95	103822
accuracy			0.91	116572
macro avg	0.83	0.65	0.69	116572
weighted avg	0.90	0.91	0.90	116572

Mengecek overfit dan underfit dari model Logistic Regression

```
X_train2 = x_train
X_test2 = x_test

logreg = RandomForestClassifier()
logreg.fit(X_train2, y_train)
y_pred_rf = logreg.predict(X_train2)
y_pred_rf_test = logreg.predict(X_test2)

print('Akurasi Train',accuracy_score(y_train, y_pred_rf))
print('Akurasi Test',accuracy_score(y_test, y_pred_rf_test))
```

```
Akurasi Train 0.9981584899617687
Akurasi Test 0.914130322890574
```

model tidak mengalami overfitting maupun underfitting

Model	AUC Score
XGBoost	0.98
Random Forest	0.97
Logistic Regression	0.95
K-Nearest Neighbors	0.93
Decision Tree	0.87

7. Conclusion

Model	Akurasi	F1-score (Kelas 0)	F1-score (Kelas 1)	Catatan
XGBoost	0.92	**0.49**	**0.96**	Paling seimbang dan akurat, **model terbaik**
Random Forest	0.91	0.46	0.95	Akurat untuk kelas 1, lemah di kelas 0
Logistic Regression	0.91	0.43	0.95	Mirip Random Forest, tapi lebih ringan
KNN	0.90	**0.33 (terendah)**	0.94	Buruk untuk kelas 0, tidak cocok untuk data besar
Decision Tree	0.87	0.42	0.93	Performa keseluruhan paling rendah

Thank You

X

Logo Company