Rakamin Academy X id/x partners

Prediction Model

ID/X Partners - Data Scientist

Presented by
Difta Alzena Sakhi

1. Business Understanding

Dalam proyek ini, tujuan utama adalah meningkatkan akurasi penilaian risiko kredit bagi perusahaan multifinance melalui pengembangan model machine learning yang dapat memprediksi potensi kredit macet atau lancar dari calon peminjam. Dengan analisis mendalam terhadap dataset yang berisi berbagai atribut terkait profil peminjam, perilaku pembayaran, dan karakteristik pinjaman, diharapkan model ini mampu membantu manajemen dalam pengambilan keputusan pemberian kredit secara lebih objektif dan efisien. Peningkatan keakuratan prediksi risiko kredit tidak hanya akan meminimalkan potensi kerugian akibat kredit bermasalah tetapi juga mendukung strategi pemberian pinjaman yang lebih selektif dan bertanggung jawab, sehingga memperkuat portofolio kredit perusahaan serta meningkatkan kepercayaan stakeholder dan pertumbuhan bisnis jangka panjang.

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.

3. Data Understanding

Tampilan Awal Dataset

	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_
0	0	1077501	1296599	5000	5000	4975.0	36 months	10.65	162.87	B	...	NaN	NaN	NaN	
1	1	1077430	1314167	2500	2500	2500.0	60 months	15.27	59.83	C	...	NaN	NaN	NaN	
2	2	1077175	1313524	2400	2400	2400.0	36 months	15.96	84.33	C	...	NaN	NaN	NaN	
3	3	1076863	1277178	10000	10000	10000.0	36 months	13.49	339.31	C	...	NaN	NaN	NaN	
4	4	1075358	1311748	3000	3000	3000.0	60 months	12.69	67.79	B	...	NaN	NaN	NaN	

5 rows × 75 columns

Penentuan Variabel Target

Status pinjaman awal:

- *Current* & *In Grace Period* dihapus karena status pinjaman masih berjalan.

Kategori target:

- Good loan (1): ‘Fully Paid’, ‘Does not meet the credit policy. Status:Fully Paid’
- Bad loan (0): Semua status lain (Charged Off, Default, Late, dll.)

Data setelah filter: 238.913 baris, 76 kolom

Distribusi target:

- Good loan: 186.727 (78%)
- Bad loan: 52.186 (22%)

Catatan: Data target tidak seimbang → perlunya teknik penanganan imbalance (upsampling/downsampling).

Tampilan Info Data

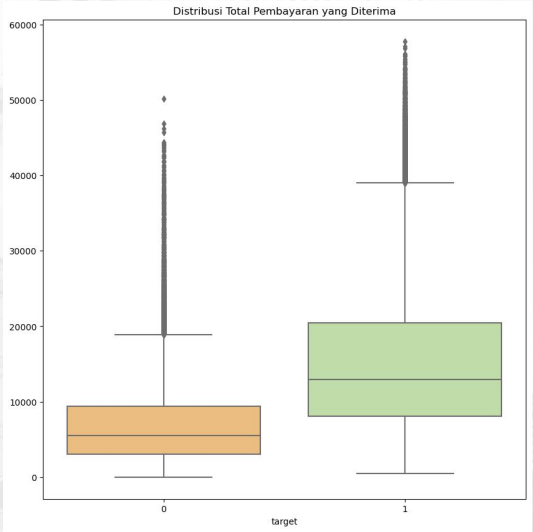
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 466285 entries, 0 to 466284
Data columns (total 75 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   Unnamed: 0                  466285 non-null  int64
 1   id                          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
```

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. Kolom target untuk klasifikasi adalah `loan_status`.

3. Exploratory Data Analysis (EDA)

Rakamin Academy

1. Hubungan Status Pinjaman dengan Total Pembayaran

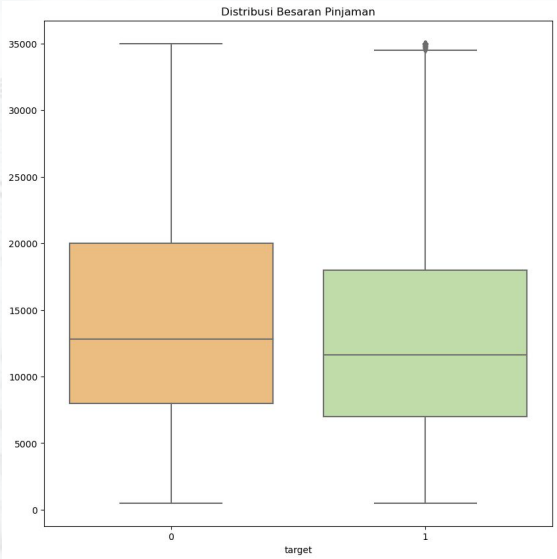


Total pembayaran lebih tinggi pada pinjaman yang lunas (good loan) dibanding yang gagal (bad loan).

Good loan rata-rata total bayar: ±15.160 vs bad loan: ±7.024.

target	count	mean	std	min	25%	50%	75%	max
0	52186.0	7024.191781	5675.182531	0.00	3012.5975	5491.77000	9372.74250	50197.03000
1	186727.0	15160.113585	9452.545563	503.54	8035.1450	12930.93777	20429.38559	57777.57987

2. Status Pinjaman dan Besaran Pinjaman



Rata-rata pinjaman pada bad loan lebih besar (±14.625) daripada good loan (±13.168).

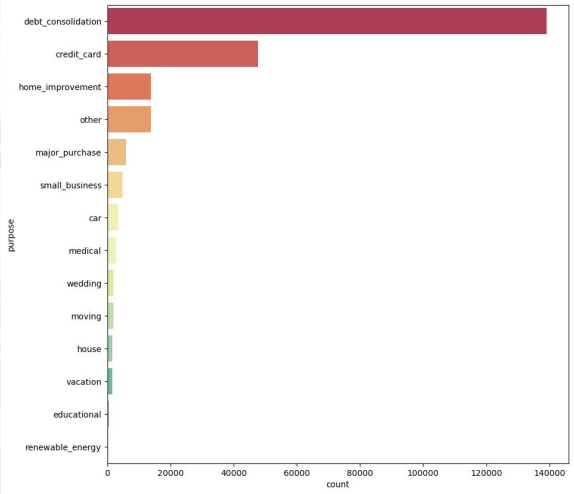
Namun total pembayaran di bad loan cenderung lebih rendah.

target	count	mean	std	min	25%	50%	75%	max
0	52186.0	14624.953053	8390.633508	500.0	8000.0	12800.0	20000.0	35000.0
1	186727.0	13167.962855	7944.742527	500.0	7000.0	11625.0	18000.0	35000.0

3. Exploratory Data Analysis (EDA)

Rakamin Academy

3. Tujuan Pengajuan Pinjaman



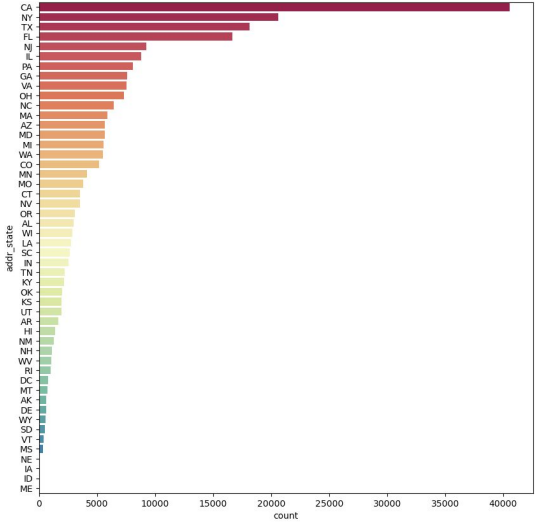
debt_consolidation	58.22%
credit_card	19.94%
home_improvement	5.81%
other	5.74%
major_purchase	2.50%
small_business	2.01%
car	1.47%
medical	1.14%
wedding	0.84%
moving	0.80%
house	0.64%
vacation	0.61%
educational	0.18%
renewable_energy	0.11%

Name: purpose, dtype: object

Mayoritas untuk **debt consolidation (58%)**, diikuti credit card (20%) dan home improvement (6%).

Tujuan pendidikan, kesehatan, dan energi terbarukan sangat kecil porsinya.

4. Distribusi Peminjam Berdasarkan Negara Bagian



CA	16.97%
NY	8.63%
TX	7.60%
FL	6.97%
NJ	3.87%
IL	3.69%
PA	3.38%
GA	3.17%
VA	3.16%
OH	3.07%

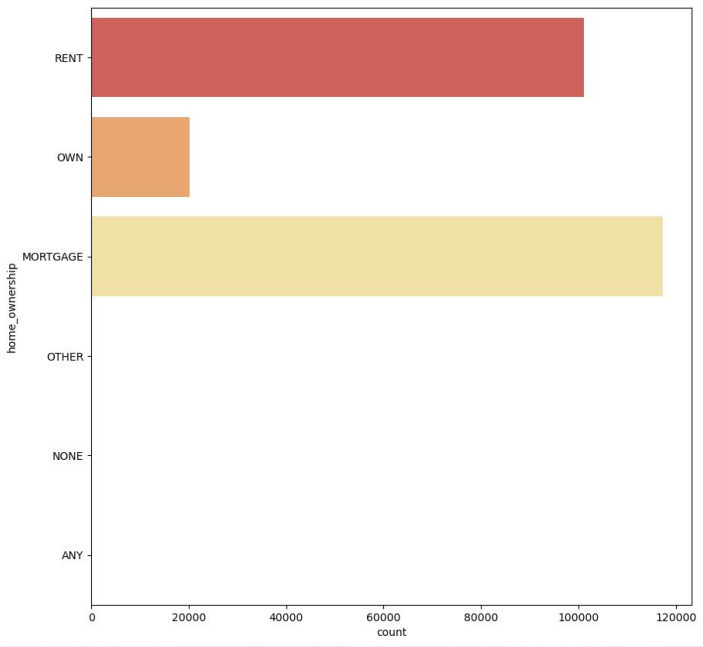
Sebagian besar peminjam berasal dari:

- California (17%)
- New York (9%)
- Texas (8%)

3. Exploratory Data Analysis (EDA)

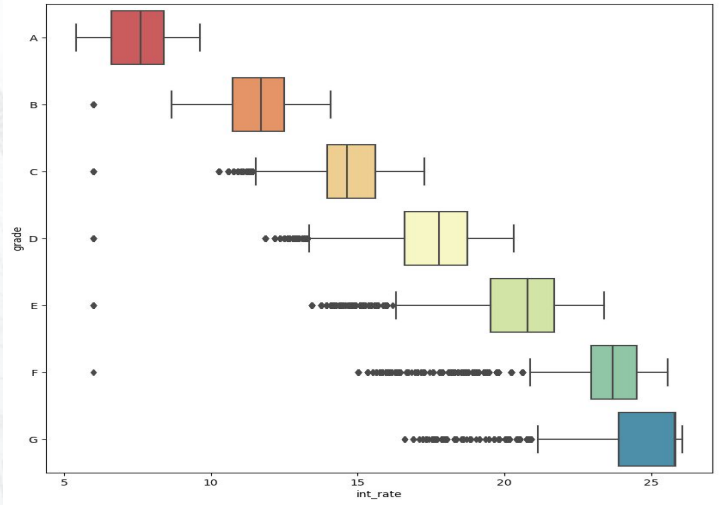
Rakamin Academy

5. Status Kepemilikan Rumah



Terlihat bahwa kebanyakan peminjam menggunakan rumah mereka sebagai jaminan pinjaman, sedangkan yang memiliki rumah secara pribadi lebih sedikit.

6. Tingkat Suku Bunga Berdasarkan Grade Pinjaman



grade	count	mean	std	min	25%	50%	75%	max
A	39500.0	7.548551	1.002864	5.42	6.62	7.62	8.39	9.63
B	72239.0	11.654911	1.280894	6.00	10.74	11.71	12.49	14.09
C	61042.0	14.760351	1.229674	6.00	13.98	14.64	15.61	17.27
D	38715.0	17.623237	1.474671	6.00	16.59	17.77	18.75	20.31
E	18086.0	20.362172	1.887752	6.00	19.52	20.80	21.70	23.40
F	7348.0	23.192361	2.053790	6.00	22.95	23.70	24.50	25.57
G	1983.0	24.290570	2.460741	16.59	23.91	25.80	25.83	26.06

Tingkat suku bunga pinjaman cenderung meningkat dari grade A ke G, yang berarti semakin tinggi grade, semakin besar bunga yang dikenakan.

4. Preprocessing

Rakamin Academy

1. Penanganan Data yang Hilang (Missing Values)

- Identifikasi fitur dengan missing values

	features	missing_value	Percentage
0	open_acc_6m	238913	100.0
1	il_util	238913	100.0
2	verification_status_joint	238913	100.0
3	dti_joint	238913	100.0
4	annual_inc_joint	238913	100.0
...	...	...	...
71	total_pymnt	0	0.0
72	total_pymnt_inv	0	0.0
73	total_rec_prncp	0	0.0
74	total_rec_int	0	0.0
75	target	0	0.0

Output menunjukkan fitur-fitur dengan persentase nilai hilang tertinggi hingga terendah.

- Penghapusan fitur dengan missing values > 25%

- Pengisian missing values pada fitur lain
  - Fitur numerik diisi dengan rata-rata (mean)
  - Fitur non-numerik diisi dengan modus (nilai yang paling sering muncul)

- Pengecekan data duplikat

```
df.duplicated().sum()
```

```
0
```

Tidak ada data yang duplikat, ini menandakan data sudah unik dan bersih dari baris-baris yang sama.

- Informasi ringkas data (info)

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 238913 entries, 0 to 466283
Data columns (total 50 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   id                      238913 non-null  int64
 1   member_id               238913 non-null  int64
 2   loan_amnt               238913 non-null  int64
 3   funded_amnt             238913 non-null  int64
 4   funded_amnt_inv         238913 non-null  float64
 5   term                    238913 non-null  object
 6   int_rate                238913 non-null  float64
 7   installment             238913 non-null  float64
 8   grade                   238913 non-null  object
 9   sub_grade               238913 non-null  object
 10  emp_title               238913 non-null  object
 11  emp_length              238913 non-null  object
 12  home_ownership          238913 non-null  object
 13  annual_inc              238913 non-null  float64
 14  verification_status     238913 non-null  object
 15  issue d                 238913 non-null  object
```

Dataset tersisa 238,913 baris dan 50 kolom dengan tipe data campuran numerik dan objek.

4. Preprocessing

Rakamin Academy

2. Pemilihan Fitur (Feature Selection)

- Identifikasi fitur dengan nilai unik khusus
  - ❖ Fitur dengan nilai unik = 1 (konstan): `['application_type', 'policy_code']`
  - ❖ Fitur dengan nilai unik = jumlah baris (unik semua): `['url', 'id', 'member_id']`
  - ❖ Fitur non-numerik dengan nilai unik sangat banyak (> 500): `['emp_title', 'url', 'title', 'zip_code', 'earliest_cr_line']`
- Gabungkan fitur-fitur di atas untuk dihapus
- Buat dataset baru tanpa fitur-fitur tersebut

```
new_df = df.loc[:, ~df.columns.isin(remove)].copy()
```

- Hapus fitur yang berpotensi menyebabkan data leakage: ['issue_d', 'loan_status', 'out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'total_rec_int', 'total_rec_late_fee', 'recoveries', 'collection_recovery_fee', 'last_pymnt_d', 'last_pymnt_amnt']
- Hapus beberapa fitur tambahan yang tidak diperlukan

```
new_df.drop(columns='sub_grade', axis=1, inplace=True)
new_df.drop(columns='last_credit_pull_d', axis=1, inplace=True)
```

- Hasil Akhir

Dataset dengan fitur terpilih:Ukuran: `(238913, 26)`& Kolom:

```
new_df.columns
```

```
Index(['loan_amnt', 'funded_amnt', 'funded_amnt_inv', 'term', 'int_rate',
       'installment', 'grade', 'emp_length', 'home_ownership', 'annual_inc',
       'verification_status', 'pymnt_plan', 'purpose', 'addr_state', 'dti',
       'delinq_2yrs', 'inq_last_6mths', 'open_acc', 'pub_rec', 'revol_bal',
       'revol_util', 'total_acc', 'initial_list_status',
       'collections_12_mths_ex_med', 'acc_now_delinq', 'target'],
      dtype='object')
```

4. Preprocessing

Rakamin Academy

3. Rekayasa Fitur (Feature Engineering)

- Identifikasi Fitur Kategorikal

  Fitur bertipe *object* (kategorikal) yang ditemukan:

  `'term', 'grade', 'emp_length', 'home_ownership', 'verification_status', 'pymnt_plan', 'purpose', 'addr_state', 'initial_list_status'`

- Label Encoding (Ordinal)

  Digunakan untuk fitur dengan urutan atau tingkatan:

  - `term`: Diubah dari `'36 months'` / `'60 months'` → 36 / 60
  - `grade`: Mapping dari `{'A':1, ..., 'G':7}`
  - `emp_length`: Diubah jadi angka 0–10 berdasarkan lama kerja
  - `initial_list_status`: `'f'` → 1, `'w'` → 0

- Cleaning Fitur Kategorikal 'home_ownership': Nilai 'NONE' dan 'ANY' disamakan ke 'OTHER'

- One-hot Encoding (Nominal): Digunakan untuk fitur tanpa tingkatan:

  - 'verification_status', 'purpose', 'addr_state'
  - Hasil: 51 fitur one-hot baru ditambahkan ke dataset

- Seleksi Fitur Berdasarkan Korelasi: Mengukur korelasi fitur numerik terhadap target.
  - Top 5 korelasi negatif (semakin tinggi nilainya, makin cenderung gagal bayar): int_rate, grade, term, dti, revol_util
  - Top 5 korelasi positif (semakin tinggi, makin cenderung bayar lancar): annual_inc, initial_list_status, total_acc, revol_bal

- Gabungkan Fitur Korelasi & One-hot Encoding

```
concat_df = pd.concat([cor_df, dummy], axis=1)
concat_df.shape
```

```
(238913, 77)
```

4. Preprocessing

Rakamin Academy

4. Penanganan Ketidakseimbangan Kelas (Target)

- Cek Distribusi Awal Target

```
concat_df['target'].value_counts(normalize=True)
```

```
1    0.781569
0    0.218431
Name: target, dtype: float64
```

Data sangat tidak seimbang, berisiko membuat model bias terhadap mayoritas (kelas 1).

- Strategi Penyeimbangan

Saya menerapkan dua teknik resampling secara simultan: (nilai tengah dari 186727 dan 52186)

  - ❖ Oversampling (menambah data minoritas): Kelas minoritas (`target` = 1) disampling ulang sebanyak 119.456 contoh.
  - ❖ Undersampling (mengurangi data mayoritas): Kelas mayoritas (`target` = 0) juga disampling ulang sebanyak 119.456 contoh.

- Verifikasi Distribusi Baru

```
df_upsampled['target'].value_counts(normalize=True)
```

```
1    0.5
0    0.5
Name: target, dtype: float64
```

Data kini sudah seimbang secara proporsional, siap untuk pelatihan model machine learning.

4. Preprocessing

Rakamin Academy

5. Standardisasi Nilai Fitur (Feature Scaling)

Menjadikan semua fitur berada dalam skala yang seragam (μ = 0, σ = 1) untuk meningkatkan kinerja algoritma berbasis jarak atau gradien.

- Standardisasi dilakukan ke seluruh kolom, termasuk fitur hasil one-hot encoding dan setelah penyeimbangan kelas

```
for i in df_upsampled.columns:
  scale = StandardScaler().fit(df_upsampled[[i]])
  df_upsampled[i] = scale.transform(df_upsampled[[i]])
```

- Isi DataFrame

```
df_upsampled.head()
```

	target	int_rate	grade	term	dti	revol_util	annual_inc	initial_list_status	total_acc	revol_bal	...	addr_state_SD	addr_state_TN	addr
217852	1.0	-0.868912	-0.744832	-0.607742	0.350031	-0.951917	-0.899016	0.602514	-1.172083	-0.603726	...	-0.047375	-0.104831	
316157	1.0	-1.821496	-1.466816	-0.607742	0.121130	-0.642365	-0.180985	-1.659713	0.033477	-0.185334	...	-0.047375	-0.104831	
230539	1.0	-0.077708	-0.022849	1.645434	0.208087	1.582284	0.348090	0.602514	-0.741526	0.195450	...	-0.047375	-0.104831	
194264	1.0	-0.064259	-0.022849	-0.607742	-0.423629	0.364712	-0.596687	0.602514	-1.344306	-0.286406	...	-0.047375	-0.104831	
215135	1.0	-1.792358	-1.466816	-0.607742	-1.647418	-1.496729	1.047226	0.602514	0.033477	-0.314919	...	-0.047375	-0.104831	

5 rows × 77 columns

Nilai-nilai dalam dataset sekarang telah berubah ke bentuk standar (rata-rata = 0 dan standar deviasi = 1), baik untuk fitur numerik maupun yang sebelumnya dikonversi dari kategorikal.

5. Pemodelan dan Evaluasi

Rakamin Academy

Decision Tree

Model ini menggunakan struktur pohon keputusan untuk memetakan input ke target. Cocok digunakan untuk interpretasi yang mudah.

```
# 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       0.81      0.90      0.86     29965
         1.0       0.89      0.79      0.84     29763

    accuracy                           0.85     59728
   macro avg       0.85      0.85      0.85     59728
weighted avg       0.85      0.85      0.85     59728
```

Random Forest

Merupakan ensemble method yang menggabungkan beberapa decision tree, meningkatkan akurasi dan mengurangi overfitting.

```
# 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       0.86      0.91      0.89     29965
         1.0       0.90      0.85      0.88     29763

    accuracy                           0.88     59728
   macro avg       0.88      0.88      0.88     59728
weighted avg       0.88      0.88      0.88     59728
```

Hasil Evaluasi:

- Kinerja merata pada kedua kelas
- Akurasi keseluruhan: 88% (tertinggi di antara model lain)

5. Pemodelan dan Evaluasi

Rakamin Academy

XGBoost Classifier

Model boosting yang kuat dan sering digunakan untuk kompetisi machine learning. Meski kuat, ia sensitif terhadap skala dan data tidak seimbang

```
#XGBoost
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       0.68      0.70      0.69     29965
         1.0       0.69      0.67      0.68     29763

    accuracy                           0.69     59728
   macro avg       0.69      0.69      0.69     59728
weighted avg       0.69      0.69      0.69     59728
```

K-Nearest Neighbors (KNN)

Model ini mengklasifikasikan berdasarkan kedekatan jarak dengan tetangga terdekat. Cenderung lambat jika dataset besar.

```
# 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       0.68      0.74      0.71     29965
         1.0       0.72      0.65      0.68     29763

    accuracy                           0.70     59728
   macro avg       0.70      0.70      0.70     59728
weighted avg       0.70      0.70      0.70     59728
```

5. Pemodelan dan Evaluasi

Rakamin Academy

Logistic Regression

Model linier yang cocok untuk prediksi kelas biner. Sangat cepat dan sering digunakan sebagai baseline dalam klasifikasi.

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

logreg = LogisticRegression(max_iter=1000, random_state=42)
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       0.65      0.64      0.64     29965
         1.0       0.64      0.66      0.65     29763

    accuracy                           0.65     59728
   macro avg       0.65      0.65      0.65     59728
weighted avg       0.65      0.65      0.65     59728
```

Hasil & Insight:

- Random Forest menunjukkan performa terbaik dengan akurasi 88%
- Model lain seperti Decision Tree dan KNN cukup kompetitif
- XGBoost & Logistic Regression kurang optimal pada data ini.

6. Kesimpulan dan Saran

Hasil pemodelan menunjukkan bahwa Random Forest Classifier memberikan performa terbaik dengan akurasi 88%, diikuti oleh Decision Tree. Proses penyeimbangan data melalui oversampling dan undersampling serta standardisasi fitur berhasil meningkatkan akurasi model secara signifikan. Sementara itu, model seperti XGBoost dan Logistic Regression belum optimal, kemungkinan karena parameter default yang belum disesuaikan.

Sebagai langkah lanjutan, disarankan melakukan hyperparameter tuning untuk meningkatkan performa, khususnya pada Random Forest dan XGBoost. Teknik validasi seperti k-fold cross-validation dan metode balancing lanjutan seperti SMOTE juga layak dipertimbangkan untuk hasil yang lebih robust dan akurat di masa depan.

7. Lampiran

Link Github Hasil Pengerjaan: Github

Thank You