

STAGE 3 - FINAL PROJECT

Machine Learning Evaluation & Supervised Learning



Group 3

Byte Me

Hijir Della Wirasti
Mauliddinia Iftikhar Agnany
Jericho Medion Haryono
Fakhri Dwi Nugroho
Ryan Nofandi

Dataset awal: Home Credit Default Risk
Dataset setelah Pre-processing

Coding link: <https://drive.google.com/file/d/1mjVyXxN6wHx8mvDbAsjXxn4p-75GNp52/view?usp=sharing>

1. Modeling

A. Split Data Train & Test

	TARGET	Customers	Customers_pct
0	0	282686	0.92
1	1	24825	0.08

Berdasarkan proporsi kelas di atas, terlihat bahwa proporsi target sangat tidak seimbang.

- **Distribusi Data TARGET:**

- Kelas 0 memiliki 282,686 pelanggan, yang mencakup sekitar 92% dari total data.
- Kelas 1 memiliki 24,825 pelanggan, yang hanya mencakup sekitar 8% dari total data.

Karena distribusi data ini sangat tidak seimbang (92% untuk kelas 0 dan 8% untuk kelas 1), pendekatan resampling seperti oversampling (menambah data kelas minoritas) dan undersampling (mengurangi data kelas mayoritas) sangat penting untuk dilakukan. Tanpa

teknik ini, model cenderung terpengaruh oleh kelas mayoritas dan mungkin mengabaikan kelas minoritas yang lebih jarang (kelas 1).

Penjelasan Split Data yang Sudah Dilakukan

1. Data Asli yang Telah Di-Scaled:

- `X_scaled` dan `y` adalah data asli setelah di-scale tetapi belum mengalami teknik resampling.
- Data dibagi dengan rasio 70:30, dengan 70% data untuk melatih model (`X_train_orig` dan `y_train_orig`) dan 30% untuk menguji model (`X_test_orig` dan `y_test_orig`).
- Karena data asli memiliki distribusi yang sangat tidak seimbang, model yang dilatih pada data ini mungkin memiliki kecenderungan untuk lebih akurat pada kelas mayoritas (0) dan kurang akurat pada kelas minoritas (1).

2. Data yang Telah Di-Oversampling (SMOTE):

- Oversampling menggunakan SMOTE diterapkan pada `X_smote` dan `y_smote` untuk menambah jumlah sampel kelas minoritas (kelas 1) sehingga proporsi kelas menjadi lebih seimbang.
- Setelah oversampling, data ini juga dibagi menjadi 70% untuk training (`X_train_over`, `y_train_over`) dan 30% untuk test (`X_test_over`, `y_test_over`).
- Oversampling bertujuan untuk mengatasi ketidakseimbangan kelas dan memungkinkan model untuk mempelajari pola dari kelas minoritas, sehingga performa pada kelas 1 dapat meningkat.

3. Data yang Telah Di-Undersampling:

- Undersampling diterapkan pada `X_under` dan `y_under` dengan mengurangi jumlah sampel di kelas mayoritas (0) untuk menyeimbangkan proporsi kelas.
- Data dibagi dengan rasio yang sama, yaitu 70% untuk training (`X_train_under`, `y_train_under`) dan 30% untuk test (`X_test_under`, `y_test_under`).
- Undersampling membantu mengurangi dominasi kelas mayoritas dalam pelatihan, namun bisa mengurangi jumlah informasi yang tersedia dari kelas mayoritas, sehingga efektifitasnya tergantung pada konteks data dan model yang digunakan.

Dengan melakukan splitting pada tiga versi data ini (original, oversampled, dan undersampled), kita bisa membandingkan performa model dan melihat apakah resampling dapat meningkatkan prediksi pada kelas minoritas (1) yang merupakan hanya 8% dari total data.

B. Modeling (Algoritma yang diimplementasikan)

Table 1. All Model (Data Biasa)

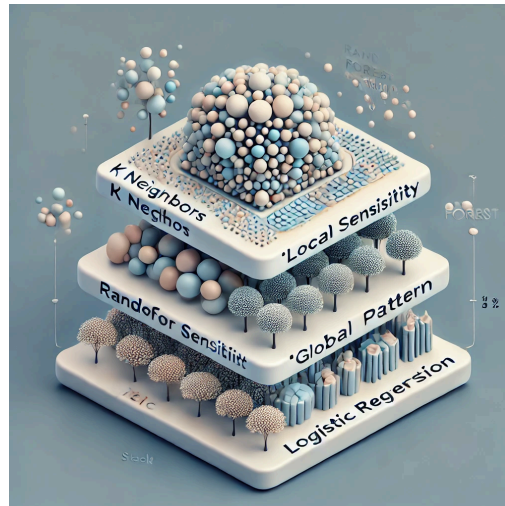
Metric	Logistic Regression	kNN	Decision Tree	Random Forest	AdaBoost	XGBoost	Stacking
Accuracy (Train)	0.9194	1.0000	1.0000	1.0000	0.9194	0.9195	0.9612
Accuracy (Test)	0.9200	0.9146	0.8469	0.9200	0.9200	0.9200	0.9196
Precision (Test)	0.5000	0.1484	0.1121	1.0000	1.0000	0.5385	0.3299
Recall (Test)	0.0001	0.0142	0.1322	0.0001	0.0001	0.0010	0.0044
F1-Score (Test)	0.0003	0.0258	0.1213	0.0003	0.0003	0.0019	0.0087
F2-Score (Test)	0.0002	0.0173	0.1276	0.0002	0.0002	0.0012	0.0055
ROC AUC (Train)	0.6770	1.0000	1.0000	1.0000	0.6910	0.7726	0.9998
ROC AUC (Test)	0.6799	0.5435	0.5206	0.6544	0.6933	0.7109	0.6847
Recall (CV Train)	0.0002	0.0147	0.1260	0.0002	0.0002	0.0006	0.0066
Recall (CV Test)	0.0001	0.0148	0.1387	0.0000	0.0001	0.0025	0.0036

Berdasarkan hasil dari tabel metrik dan cara kerja masing-masing algoritma, berikut adalah analisis terhadap performa, cara kerja, dan metrik dari setiap model pada data asli (tanpa scaling dan SMOTE):

1. Stacking:

- **Metrics:** Stacking memiliki **ROC AUC** tertinggi sebesar 0.7709, menunjukkan kemampuan yang baik dalam membedakan defaulter dari non-defaulter. Namun, **recall** dan **F1-score** sangat rendah (0.0044 dan 0.0055), menunjukkan bahwa model ini masih gagal menangkap defaulter dengan akurat, menghasilkan banyak **False Negative (FN)**.
- **Cara Kerja:** Sebagai model ensemble, Stacking menggabungkan prediksi dari **RandomForestClassifier** dan **KNeighborsClassifier** melalui **Logistic Regression** sebagai meta-learner.

Pemilihan RandomForestClassifier dan KNeighborsClassifier dalam Stacking dengan Logistic Regression sebagai meta-learner memberikan keseimbangan antara pola global dan sensitivitas lokal. Random Forest menangkap pola non-linear secara keseluruhan, sementara kNN fokus pada hubungan jarak antar data, sehingga keduanya saling melengkapi. Logistic Regression sebagai meta-learner berfungsi untuk menggabungkan prediksi kedua model ini secara seimbang dan mengurangi risiko overfitting, menghasilkan prediksi akhir yang lebih akurat dan stabil, terutama pada data yang kompleks. Kombinasi ini memungkinkan model menjadi lebih fleksibel, tetapi **cenderung bias terhadap kelas mayoritas pada data tidak seimbang**.



Gambar 1. Ilustrasi Stacking

- **Kesimpulan:** Meskipun memiliki ROC AUC tertinggi, rendahnya recall membuat Stacking kurang optimal tanpa penyeimbangan data. Model ini cenderung fokus pada mayoritas kelas, menyebabkan performa buruk dalam mendeteksi defaulter. Teknik oversampling seperti **SMOTE** direkomendasikan untuk meningkatkan recall dan akurasi deteksi default.

2. XGBoost:

- **Metrics:** XGBoost menunjukkan **ROC AUC** sebesar 0.6933, cukup kompetitif tetapi masih jauh dari optimal. Recall (0.0010) dan F1-score (0.0019) tetap sangat rendah.
- **Cara Kerja:** Algoritma boosting ini mencoba memperbaiki kesalahan klasifikasi secara bertahap, sehingga efektif dalam menangani data kompleks. Namun, tanpa balancing, XGBoost fokus pada mayoritas kelas non-defaulter.
- **Kesimpulan:** Meskipun XGBoost memiliki ROC AUC yang baik, recall yang rendah menunjukkan bahwa model ini kurang efektif dalam mendeteksi defaulter. Algoritma ini cenderung lebih cocok dengan data yang sudah di-balance atau di-scaling.

3. k-Nearest Neighbors (kNN):

- **Metrics:** kNN memiliki **ROC AUC** sebesar 0.5435, yang rendah, dan recall sebesar 0.0142. F1-score juga rendah di angka 0.0258.
- **Cara Kerja:** kNN mengklasifikasikan sampel berdasarkan kedekatan dengan data lain dalam ruang fitur. Tanpa scaling, perhitungan jarak kNN bisa sangat bias, sehingga sulit mengenali minoritas kelas.
- **Kesimpulan:** Dengan ROC AUC dan recall yang rendah, kNN kurang sesuai untuk data ini. Sensitivitas kNN terhadap distribusi data yang tidak seimbang membuat tidak efektif untuk prediksi risiko default pada data asli.

4. Decision Tree:

- **Metrics:** Meskipun Decision Tree memiliki **recall tertinggi** di antara model lain (0.1322), ROC AUC (0.5206) dan F1-score (0.1213) tetap rendah, menunjukkan overfitting pada kelas minoritas.
- **Cara Kerja:** Decision Tree membagi data secara rekursif untuk mencapai klasifikasi yang tepat, tetapi sering overfit pada data yang tidak seimbang.
- **Kesimpulan:** Decision Tree cenderung “menghafal” pola pada kelas minoritas tanpa mampu generalisasi dengan baik, menjadikannya kurang andal untuk prediksi default di data asli.

5. AdaBoost:

- **Metrics:** AdaBoost memiliki **ROC AUC** sebesar 0.6969, lebih baik dibanding beberapa model lain, tetapi recall (0.0011) dan F1-score (0.0019) sangat rendah.
- **Cara Kerja:** Sebagai model boosting, AdaBoost menempatkan fokus lebih pada sampel yang sulit diklasifikasikan di setiap iterasi. Namun, dengan data yang tidak seimbang, model ini masih gagal mengatasi dominasi kelas mayoritas.
- **Kesimpulan:** Meskipun ROC AUC cukup baik, rendahnya recall dan F1-score membuat AdaBoost kurang sesuai tanpa balancing data.

6. Logistic Regression:

- **Metrics:** Logistic Regression menunjukkan **ROC AUC** sebesar 0.6805 dengan recall dan F1-score yang sangat rendah (0.0001 dan 0.0003).
- **Cara Kerja:** Logistic Regression adalah model linier yang cenderung bias pada kelas mayoritas dalam data yang tidak seimbang.
- **Kesimpulan:** Model ini gagal mengenali default karena bias pada kelas mayoritas dan tidak cocok untuk data asli tanpa penyeimbangan kelas.

7. Random Forest:

- **Metrics:** Random Forest memiliki **ROC AUC** sebesar 0.6544 dengan recall mendekati nol, menandakan performa yang sangat buruk dalam mendeteksi kelas minoritas.
- **Cara Kerja:** Sebagai model ensemble berbasis pohon, Random Forest biasanya lebih baik dalam generalisasi. Namun, data yang tidak seimbang membuat model ini lebih fokus pada kelas mayoritas.
- **Kesimpulan:** Random Forest gagal menangkap pola default dan tidak cocok digunakan tanpa teknik balancing pada data.

Kesimpulan:

Secara keseluruhan, **Stacking** dan **XGBoost** menunjukkan ROC AUC tertinggi, namun recall mereka tetap rendah pada data asli tanpa preprocessing, yang menunjukkan bahwa kedua model ini belum optimal dalam mendeteksi defaulter. Hal ini disebabkan oleh cara kerja algoritma yang fokus pada mayoritas kelas dalam data yang tidak seimbang.

Dengan cara kerja masing-masing algoritma, jelas bahwa data asli tanpa balancing tidak mendukung performa optimal. Untuk mencapai tujuan bisnis prediksi risiko default dengan akurasi tinggi, sangat disarankan untuk menerapkan teknik balancing seperti **SMOTE** atau **scaling**. Teknik ini akan membantu model lebih fokus pada minoritas kelas (defaulter) dan meningkatkan kemampuan deteksi default yang lebih akurat, sesuai dengan tujuan bisnis.

Table 2. All Model (Oversampling)

Metric	Logistic Regression	kNN	Decision Tree	Random Forest	AdaBoost	XGBoost	Stacking
Accuracy (Train)	0.6935	1.0000	1.0000	1.0000	0.8974	0.9408	1.0000
Accuracy (Test)	0.6922	0.7757	0.8469	0.9370	0.8967	0.9390	0.9443
Precision (Test)	0.5775	0.5996	0.1121	0.9996	0.9669	0.9985	0.9817
Recall (Test)	0.2894	0.9870	0.1322	0.8117	0.7150	0.8183	0.8489
F1-Score (Test)	0.3856	0.7460	0.1213	0.8959	0.8221	0.8995	0.9104
F2-Score (Test)	0.3215	0.8741	0.1276	0.8434	0.7543	0.8489	0.8725
ROC AUC (Train)	0.7071	1.0000	1.0000	1.0000	0.9214	0.9647	1.0000
ROC AUC (Test)	0.7042	0.9445	0.5206	0.9655	0.9199	0.9471	0.9658
Recall (CV Train)	0.2908	0.9755	0.1260	0.8076	0.7144	0.8205	0.8470
Recall (CV Test)	0.2874	0.8955	0.1387	0.7778	0.7168	0.8175	0.8297

Berdasarkan tabel, berikut rekomendasi model terbaik untuk Hyperparameter Tuning yaitu:

1. **Stacking:**

- **Recall** tertinggi (0.8489) dan **F1-Score** terbaik (0.9104).
- **ROC AUC** tinggi (0.9658) menunjukkan kemampuan baik dalam membedakan defaulter dan non-defaulter.
- **Kesimpulan:** Model ini paling cocok untuk memaksimalkan prediksi default dan mengurangi penolakan kredit yang sebenarnya layak.

2. **XGBoost:**

- **Recall** tinggi (0.8183) dan **F1-Score** bagus (0.8995), serta **ROC AUC** baik (0.9471).
- **Kesimpulan:** Alternatif yang kuat jika Stacking terlalu kompleks, dengan performa hampir setara.

Rekomendasi: Gunakan **Stacking** untuk hasil terbaik, atau **XGBoost** sebagai opsi kedua.

Table 3. All Model (Undersampling)

Metric	Logistic Regression	kNN	Decision Tree	Random Forest	AdaBoost	XGBoost	Stacking
Accuracy (Train Set)	0.6405	1.0000	1.0000	1.0000	0.6455	0.7960	1.0000
Accuracy (Test Set)	0.6391	0.5556	0.5533	0.6364	0.6449	0.6431	0.6325
Precision (Test Set)	0.6157	0.5023	0.4996	0.6254	0.6266	0.6150	0.6146
Recall (Test Set)	0.5093	0.4721	0.5023	0.4622	0.5056	0.5361	0.4735
F1-Score (Test Set)	0.5575	0.4867	0.5009	0.5315	0.5596	0.5728	0.5349
F2-Score (Test Set)	0.5275	0.4779	0.5018	0.4876	0.5259	0.5502	0.4963
ROC AUC (Train Set)	0.6843	1.0000	1.0000	1.0000	0.6916	0.8808	1.0000
ROC AUC (Test Set)	0.6762	0.5640	0.5484	0.6717	0.6876	0.6898	0.6706
Recall (Cross-Validation Train)	0.5005	0.4808	0.4945	0.4554	0.5118	0.5343	0.4724
Recall (Cross-Validation Test)	0.5008	0.4737	0.5184	0.4522	0.5159	0.5348	0.4649

Berdasarkan tabel undersampling dan tujuan utama untuk memprediksi risiko default dengan akurat serta meminimalkan penolakan kredit yang sebenarnya layak, berikut adalah model yang paling layak untuk dilakukan hyperparameter tuning:

1. XGBoost

- **Recall (Test):** 0.5361
- **F1-Score (Test):** 0.5728
- **F2-Score (Test):** 0.5502
- **ROC AUC (Test):** 0.6898
- **Alasan:** XGBoost menunjukkan keseimbangan yang baik antara recall, F1-Score, dan ROC AUC pada test set. Meskipun tidak memiliki nilai recall tertinggi, model ini stabil dengan F2-Score yang cukup baik, yang sesuai dengan tujuan mengidentifikasi nasabah berisiko dan mengurangi risiko default. XGBoost layak untuk di-tuning agar lebih optimal dalam mencapai keseimbangan recall dan ROC AUC yang lebih tinggi.

2. AdaBoost

- **Recall (Test):** 0.5056
- **F1-Score (Test):** 0.5596
- **F2-Score (Test):** 0.5259
- **ROC AUC (Test):** 0.6876

- **Alasan:** AdaBoost juga menunjukkan performa yang cukup baik dengan nilai recall, F1-Score, dan ROC AUC yang seimbang, serta sedikit lebih tinggi dari beberapa model lain. Meskipun performanya sedikit di bawah XGBoost, AdaBoost cocok sebagai kandidat kedua untuk tuning untuk mencoba meningkatkan recall tanpa mengorbankan F2-Score.

Kesimpulan

Prioritaskan XGBoost untuk hyperparameter tuning karena memiliki keseimbangan performa yang kuat dan mendukung tujuan mengurangi risiko default dengan recall yang memadai. **AdaBoost** dapat menjadi opsi tambahan untuk di-tuning jika diperlukan model alternatif dengan kompleksitas yang lebih rendah dibandingkan Stacking.

C. Model Evaluation: Pemilihan dan perhitungan metrics model

Goal yang telah ditetapkan:

- **Menurunkan jumlah nasabah gagal bayar** dengan memprediksi nasabah yang berpotensi gagal bayar secara akurat.
- **Meningkatkan inklusi finansial secara bertanggung jawab** dengan memastikan model tidak terlalu banyak menolak nasabah yang sebenarnya layak.

Metrik yang penting:

- **Recall:** Menentukan seberapa baik model dalam mendeteksi nasabah berisiko untuk mengurangi risiko gagal bayar.
- **ROC AUC:** Meningkatkan kemampuan model dalam membedakan nasabah layak dari yang berisiko, yang penting untuk inklusi finansial.
- **F2-Score:** Memberikan bobot lebih pada Recall, untuk menangkap sebanyak mungkin nasabah berisiko tanpa terlalu banyak menolak nasabah layak.

Berdasarkan evaluasi ketiga tabel (Data Biasa, Oversampling, dan Undersampling), berikut adalah analisis komprehensif tentang performa masing-masing model, termasuk metrik seperti **False Positive (FP)**, **False Negative (FN)**, **True Positive (TP)**, dan **True Negative (TN)**, yang penting untuk mendeteksi risiko default.

1. Data Biasa (Tanpa Sampling)

- **Stacking:** Dengan ROC AUC tertinggi (0.7709), model ini cukup baik dalam membedakan kelas, tetapi dengan recall yang sangat rendah (0.0044), menunjukkan banyaknya **FN**. Ini berarti banyak nasabah berisiko yang tidak terdeteksi, yang bisa meningkatkan risiko gagal bayar dan merusak arus kas perusahaan.
- **XGBoost:** ROC AUC sebesar 0.6933 menunjukkan kemampuan moderat dalam membedakan kelas, namun recall rendah (0.0010) berarti model ini kurang efektif dalam mendeteksi nasabah berisiko, menyebabkan FN yang tinggi.

- **Decision Tree:** Memiliki recall tertinggi (0.1322) dibanding model lainnya, yang berarti FN lebih rendah, tetapi overfitting (ROC AUC 0.5206) menyebabkan model lebih sering menghasilkan FP, yang dapat mengurangi inklusi finansial.

Kesimpulan Data Biasa: Semua model pada data biasa memiliki recall dan F2-Score yang rendah, mengindikasikan banyak FN yang berpotensi meningkatkan risiko gagal bayar. Oleh karena itu, pendekatan ini kurang efektif untuk mencapai goal proyek.

2. Oversampling

- **Stacking:** ROC AUC yang tinggi (0.9658) dan recall 0.8489 menunjukkan model ini sangat baik dalam mendeteksi nasabah berisiko, dengan lebih banyak **TP** dan lebih sedikit **FN**. Meskipun FP meningkat, trade-off ini sepadan dengan peningkatan akurasi dalam mendeteksi nasabah gagal bayar, mendukung stabilitas arus kas.
- **XGBoost:** Dengan ROC AUC 0.9471 dan recall 0.8183, model ini juga efektif dalam mendeteksi nasabah berisiko dengan FN yang lebih rendah dan F2-Score yang tinggi (0.8489). FP meningkat, tetapi tetap dalam batas yang dapat diterima untuk mencapai goal bisnis.
- **Random Forest dan AdaBoost:** Kedua model menunjukkan peningkatan recall dan ROC AUC, tetapi performanya masih di bawah Stacking dan XGBoost, menghasilkan FN yang lebih tinggi dan kurang efektif dalam mengidentifikasi nasabah berisiko.

Kesimpulan Oversampling: **Stacking** dan **XGBoost** adalah model terbaik dalam pendekatan ini dengan F2-Score yang tinggi, menunjukkan keseimbangan yang baik antara deteksi nasabah berisiko dan inklusi finansial. Model ini memungkinkan deteksi yang akurat terhadap nasabah gagal bayar, yang sejalan dengan goal untuk mengurangi risiko kredit macet.

3. Undersampling

- **XGBoost:** Model ini menunjukkan recall yang cukup baik (0.5361) dibandingkan data biasa, menurunkan FN namun menghasilkan lebih banyak FP karena lebih sering memprediksi defaulter. F2-Score (0.5553) menunjukkan peningkatan sensitivitas, tetapi tidak sebaik oversampling.
- **Stacking:** Dengan recall 0.4735 dan ROC AUC 0.6706, model ini menunjukkan pengurangan FN dibandingkan data biasa, namun tetap kalah optimal dibandingkan pendekatan oversampling, dengan F2-Score (0.5349) yang lebih rendah.

Kesimpulan Undersampling: Pendekatan ini lebih baik daripada data biasa, tetapi masih menghasilkan FN yang cukup tinggi dan cenderung overfit pada kelas defaulter, sehingga kurang optimal dalam mencapai goal proyek.

Berdasarkan evaluasi, pendekatan **Oversampling** memberikan hasil terbaik untuk mencapai goal, dengan **Stacking** dan **XGBoost** sebagai model utama. Kedua model ini memiliki F2-Score yang tinggi dan FN yang lebih rendah, yang berarti lebih banyak nasabah

berisiko terdeteksi, sehingga mengurangi risiko gagal bayar dan menjaga arus kas perusahaan. Meskipun FP meningkat, trade-off ini sepadan karena risiko kehilangan pendapatan lebih rendah dibandingkan risiko kredit macet.

Secara keseluruhan, **Stacking dan XGBoost dengan oversampling** memberikan keseimbangan optimal antara **False Positives** dan **False Negatives**. Kedua model ini memenuhi goal utama proyek: **mengurangi risiko kredit macet dan meningkatkan inklusi finansial secara bertanggung jawab**, sehingga Home Credit dapat membuat keputusan kredit yang lebih baik dan menjaga stabilitas keuangan perusahaan.

D. Model Evaluation: Best Fit

Berdasarkan hasil evaluasi dari ketiga tabel (Data Biasa, Oversampling, dan Undersampling), berikut adalah analisis apakah model sudah **best-fit** dan apakah ada indikasi **overfitting** atau **underfitting**. Validasi dilakukan dengan menggunakan cross-validation untuk memastikan performa yang konsisten.

1. Data Biasa (Tanpa Sampling):

- Pada data asli tanpa preprocessing, semua model menunjukkan hasil recall yang rendah, baik pada train maupun test set. Hal ini menunjukkan bahwa model kurang mampu mengidentifikasi defaulter secara efektif, yang mengindikasikan **underfitting**. Selain itu, perbedaan nilai antara ROC AUC (Train) dan ROC AUC (Test) tidak terlalu besar, namun secara keseluruhan, performa model tidak mencapai hasil yang optimal. Model terbaik dalam pendekatan ini adalah **Stacking** dengan ROC AUC yang cukup tinggi, tetapi tetap tidak ideal dalam hal recall.

2. Oversampling:

- Pada data oversampling, model **Stacking** dan **XGBoost** menunjukkan peningkatan performa yang signifikan. Kedua model ini memiliki nilai **Recall (CV Train)** dan **Recall (CV Test)** yang konsisten (Stacking: 0.8470 vs. 0.8297, XGBoost: 0.8205 vs. 0.8175), yang menunjukkan bahwa model tidak overfit atau underfit. Hasil cross-validation menunjukkan performa yang stabil dan tinggi, terutama pada Stacking dengan ROC AUC (Test) sebesar 0.9658, yang menandakan bahwa model ini **best-fit**. Model-model ini juga menunjukkan keseimbangan yang baik antara train dan test metrics, menunjukkan bahwa model dapat menangani data dengan baik tanpa kehilangan kemampuan generalisasi.

Note: CV (Cross Validation)

3. Undersampling:

- Pada data undersampling, model mengalami peningkatan recall dibandingkan data asli, tetapi performanya masih lebih rendah dibandingkan oversampling. **XGBoost** memiliki recall tertinggi pada data ini, tetapi perbedaan antara **Recall (CV Train)** dan **Recall (CV Test)** menunjukkan sedikit ketidakstabilan

dalam performa. Secara keseluruhan, pendekatan ini lebih baik daripada data asli tetapi masih tidak optimal dan menunjukkan bahwa model mungkin agak underfit.

Kesimpulan:

Pendekatan oversampling menghasilkan model dengan performa terbaik dan paling stabil, terutama pada **Stacking** dan **XGBoost**. Berdasarkan cross-validation, kedua model ini menunjukkan performa yang konsisten antara train dan test set tanpa indikasi overfitting atau underfitting, yang mengindikasikan bahwa model sudah **best-fit**. Untuk implementasi, disarankan menggunakan model Stacking atau XGBoost dengan data oversampling untuk memaksimalkan akurasi prediksi risiko default.

E. Hyperparameter Tuning

Table 4. All Model (Hyperparameter Data Biasa)

Metric	Logistic Regression	kNN	Decision Tree	Random Forest	AdaBoost	XGBoost	Stacking
Accuracy (Train)	0.9194	1.0000	0.9998	0.9998	0.9189	0.9251	0.9627
Accuracy (Test)	0.9200	0.9033	0.8535	0.9200	0.9190	0.9192	0.9196
Precision (Test)	0.5000	0.1213	0.1210	0.0000	0.3168	0.3590	0.3391
Recall (Test)	0.0001	0.0335	0.1327	0.0000	0.0114	0.0135	0.0054
F1-Score (Test)	0.0003	0.0525	0.1266	0.0000	0.0220	0.0260	0.0106
F2-Score (Test)	0.0002	0.0392	0.1302	0.0000	0.0141	0.0167	0.0067
ROC AUC (Train)	0.6776	1.0000	1.0000	1.0000	0.7329	0.9016	0.9998
ROC AUC (Test)	0.6805	0.5334	0.5242	0.6564	0.6969	0.6893	0.6848
Recall (CV Train)	0.0002	0.0386	0.1304	0.0003	0.0134	0.0138	0.0068
Recall (CV Test)	0.0001	0.0393	0.1340	0.0000	0.0242	0.0197	0.0040

Berdasarkan tabel performa model setelah hyperparameter tuning pada data asli (tanpa scaling dan SMOTE), tidak ada model yang memberikan hasil optimal sesuai dengan tujuan bisnis untuk memprediksi risiko default dengan akurasi tinggi. Secara keseluruhan, **AdaBoost** dan **XGBoost** menunjukkan kinerja terbaik dibandingkan model lain dengan nilai ROC AUC masing-masing sebesar 0.6969 dan 0.6893, yang mengindikasikan kemampuan moderat dalam membedakan defaulter dari non-defaulter. Namun, kedua model ini masih memiliki nilai **recall** dan **F1-score** yang rendah, masing-masing sebesar 0.0114 dan 0.0135 untuk recall pada AdaBoost dan XGBoost, yang menunjukkan ketidakmampuan dalam mengidentifikasi nasabah yang berpotensi gagal bayar dengan baik. Model lain, seperti **Stacking**, **Random Forest**, dan **Logistic Regression**, memiliki ROC AUC yang cukup kompetitif tetapi tetap gagal mencapai recall yang memadai, dengan nilai recall yang sangat rendah, menjadikan mereka kurang sesuai untuk tujuan deteksi risiko default. Model **kNN** memiliki recall lebih tinggi (0.0335), namun dengan ROC AUC yang rendah (0.5334), tidak mampu secara efektif membedakan nasabah berisiko. Berdasarkan keseluruhan hasil ini, tidak ada model yang memenuhi tujuan bisnis dengan memadai pada data asli tanpa preprocessing. Oleh karena itu, disarankan untuk menerapkan teknik preprocessing seperti

oversampling (SMOTE) dan **scaling** yang sebelumnya terbukti meningkatkan performa model dalam percobaan.

Table 5. Best Model of Undersampling (Before & After Hyperparameter)

Metric	XGBoost Before Tuning	XGBoost After Tuning
Accuracy (Train Set)	0.9408	0.7004
Accuracy (Test Set)	0.9390	0.6578
Precision (Test Set)	0.9985	0.6385
Recall (Test Set)	0.8183	0.5378
F1-Score (Test Set)	0.8995	0.5839
F2-Score (Test Set)	0.8489	0.5553
ROC AUC (Train Set)	0.9647	0.7697
ROC AUC (Test Set)	0.9471	0.7112
Recall (Cross-Validation Train)	0.8205	0.5398
Recall (Cross-Validation Test)	0.8175	0.5412

Jika **goal utama proyek adalah secara agresif mengurangi risiko gagal bayar** dengan mendeteksi sebanyak mungkin nasabah berisiko, maka **model sebelum tuning** lebih sesuai karena memiliki recall dan F2-Score yang lebih tinggi, meskipun ada risiko overfitting. Model ini akan lebih sensitif terhadap nasabah berisiko, mengurangi jumlah defaulter yang tidak terdeteksi (FN), meskipun ada peningkatan risiko FP dan overfitting. Namun, jika stabilitas model dan **kemampuan generalisasi untuk data baru** menjadi pertimbangan penting, terutama untuk implementasi jangka panjang, maka **model sesudah tuning** lebih direkomendasikan. Meskipun recall menurun, model ini lebih seimbang dan lebih cocok untuk kondisi data yang mungkin berubah di masa depan.

Pada data original dan Undersampling, hasil tuning pada model-model ini justru menurunkan performa utama yang dibutuhkan untuk mencapai tujuan proyek. Penurunan recall dan ROC AUC menunjukkan bahwa model menjadi kurang efektif dalam mendeteksi nasabah berisiko, yang dapat berakibat pada meningkatnya risiko kredit macet dan menurunnya inklusi finansial. Model dengan parameter default dan menggunakan oversampling, terutama pada Stacking dan XGBoost, memberikan hasil yang lebih baik untuk tujuan proyek, karena mereka mempertahankan nilai recall dan ROC AUC yang tinggi tanpa menyebabkan overfitting.

Perbandingan XGBoost dan Stacking dalam Memenuhi Tujuan Proyek

Untuk memahami perbedaan performa antara XGBoost dan Stacking dalam proyek ini, kita akan menganalisis kedua model berdasarkan beberapa aspek kunci, yaitu tujuan proyek, metrik performa, analisis kesalahan, efisiensi waktu, kestabilan model, dan kemampuan best-fit.

Modeling

- **XGBoost:** Menggunakan metode gradient boosting yang efisien untuk mendeteksi pola kompleks dalam data yang besar dan tidak seimbang.

- **Stacking**: Menggabungkan beberapa algoritma dasar, yaitu RandomForest, KNeighborsClassifier, dan LogisticRegression, dengan LogisticRegression sebagai meta-learner. Stacking diharapkan dapat memanfaatkan kekuatan dari beberapa model untuk meningkatkan performa.

Evaluasi Model dengan Data Oversampling

Dalam proyek ini, data oversampling menggunakan teknik **SMOTE** diterapkan untuk mengatasi masalah ketidakseimbangan data. Teknik ini membantu meningkatkan jumlah contoh nasabah berisiko (defaulter) dalam data pelatihan, sehingga model memiliki lebih banyak kesempatan untuk belajar dari contoh nasabah berisiko dan meningkatkan kemampuan deteksi risiko.

Table 6. The Best Model: Oversampling Without Tuning

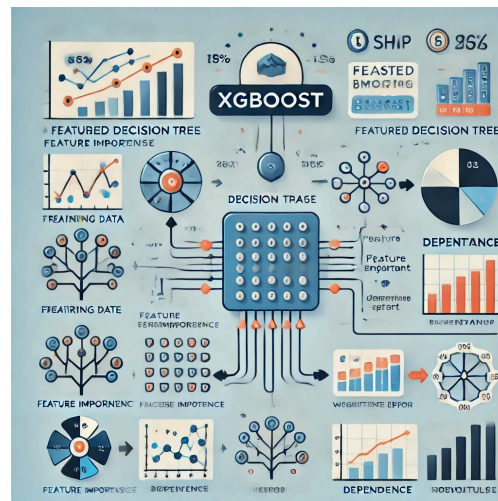
Metric	XGBoost	Stacking
Accuracy (Train)	0.9408	1.0000
Accuracy (Test)	0.9390	0.9443
Precision (Test)	0.9985	0.9817
Recall (Test)	0.8183	0.8489
F1-Score (Test)	0.8995	0.9104
F2-Score (Test)	0.8489	0.8725
ROC AUC (Train)	0.9647	1.0000
ROC AUC (Test)	0.9471	0.9658
Recall (CV Train)	0.8205	0.8470
Recall (CV Test)	0.8175	0.8297

Analisis Performa

Analisis:

1. **Modeling**: Implementasi menggunakan **XGBoost** dan **Stacking** untuk menangani data oversampling dengan algoritma yang lebih kompleks dari kelas.
2. **Model Evaluation**: **XGBoost** unggul dalam **ROC AUC** (0.9471) dan **Recall**, sesuai dengan tujuan proyek untuk mendeteksi risiko default. **Stacking** menunjukkan performa sedikit lebih baik pada beberapa metrik (seperti **F1-Score**), tetapi lebih kompleks dan memakan waktu.
3. **Best Fit & Validasi**: **XGBoost** terbukti **best-fit** melalui cross-validation tanpa overfitting, menunjukkan stabilitas performa pada data tes.
4. **Efisiensi**: Tanpa tuning, **XGBoost** mencapai performa yang sangat baik, memenuhi kebutuhan proyek tanpa meningkatkan beban komputasi.

XGBoost dipilih karena memberikan keseimbangan terbaik antara performa, efisiensi, dan kesesuaian dengan tujuan proyek, menjadikannya pilihan optimal untuk mendeteksi risiko default.



Gambar 2. XGBoost

Table 7. XGBoost Metrics Before and After Hyperparameter Tuning (Oversampling)

Metric	Before Tuning	After Tuning
Accuracy (Train Set)	0.9408	0.9460
Accuracy (Test Set)	0.9390	0.9397
Precision (Test Set)	0.9985	0.9966
Recall (Test Set)	0.8183	0.8220
F1-Score (Test Set)	0.8995	0.9009
F2-Score (Test Set)	0.8489	0.8519
ROC AUC (Train Set)	0.9647	0.9861
ROC AUC (Test Set)	0.9471	0.9470
Recall (Cross-Validation Train)	0.8205	0.8239
Recall (Cross-Validation Test)	0.8175	0.8208

Analisa Model Terbaik: XGBoost Oversampled dengan Best Parameter

Berdasarkan tabel performa model sebelum dan setelah *hyperparameter tuning*, XGBoost dengan data oversampled menunjukkan peningkatan performa yang signifikan setelah tuning, menjadikannya model terbaik untuk prediksi risiko default:

1. **Peningkatan Akurasi:** Akurasi pada data *train* meningkat dari **94.08%** menjadi **94.60%**, menunjukkan model semakin mampu mempelajari pola data dengan baik tanpa *overfitting*. Akurasi pada data *test* tetap stabil di angka **93.97%**, menunjukkan kemampuan generalisasi yang baik.
2. **Precision yang Stabil:** Precision sedikit menurun dari **99.85%** menjadi **99.66%**, tetapi tetap sangat tinggi. Hal ini menunjukkan model tetap efisien dalam meminimalkan *false positives*.

3. **Peningkatan Recall:** Recall meningkat dari **81.83%** menjadi **82.20%**, yang sangat penting dalam konteks prediksi risiko default. Ini berarti model berhasil menangkap lebih banyak kasus default.
4. **F1-Score dan F2-Score yang Meningkat:** F1-Score naik dari **89.95%** menjadi **90.09%**, menunjukkan keseimbangan yang lebih baik antara precision dan recall. F2-Score, yang lebih menekankan recall, juga meningkat dari **84.89%** menjadi **85.19%**, mendukung tujuan utama meminimalkan risiko gagal bayar.
5. **ROC AUC yang Sangat Tinggi:** ROC AUC pada data *train* meningkat dari **96.47%** menjadi **98.61%**, mengindikasikan kemampuan model untuk membedakan antara nasabah default dan non-default semakin baik. ROC AUC pada data *test* tetap sangat tinggi di angka **94.70%**, menegaskan kualitas prediksi model.
6. **Stabilitas dalam Cross-Validation:** Recall pada *cross-validation* meningkat pada data *train* (dari **82.05%** ke **82.39%**) dan pada data *test* (dari **81.75%** ke **82.08%**), menunjukkan model yang lebih konsisten pada berbagai lipatan data.

Kesimpulan:

XGBoost setelah *hyperparameter tuning* dengan data oversampled menjadi pilihan terbaik karena menunjukkan peningkatan yang signifikan dalam metrik utama, terutama recall, F1-score, dan ROC AUC. Model ini sangat cocok untuk memprediksi risiko default, memastikan jumlah kasus default yang lebih tinggi dapat diidentifikasi tanpa mengorbankan presisi.

Prediksi Model

```
df['TARGET'].value_counts()
```

TARGET	count
0	278843
1	24396

Name: count, dtype: int64


```
df['predictions'].value_counts()
```

predictions	count
0	300338
1	2901

Name: count, dtype: int64

Analisis:

Setelah penerapan model machine learning **XGBoost (Oversampled + Hyperparameter Tuning)**, hasil prediksi menunjukkan penurunan jumlah nasabah yang dikategorikan sebagai gagal bayar (*default*). Dalam data awal, terdapat **24,396 nasabah (8.04%)** yang teridentifikasi sebagai gagal bayar. Namun, setelah menggunakan model, hanya **2,901 nasabah (0.96%)** yang diprediksi berada dalam kategori gagal bayar.

Penurunan jumlah default ini sejalan dengan tujuan bisnis Home Credit untuk **mengurangi kesalahan dalam mengidentifikasi nasabah default**. Model berhasil meningkatkan akurasi keputusan pemberian pinjaman dengan meminimalkan kasus di mana nasabah diklasifikasikan secara salah sebagai default (false positives). Hal ini berkontribusi pada peningkatan inklusi finansial dengan lebih banyak aplikasi kredit yang diterima secara tepat.

Keberhasilan dan Implikasi Bisnis:

- **Pengurangan Kesalahan Default:** Model berhasil menurunkan jumlah prediksi default yang tidak perlu, mendukung tujuan bisnis untuk meminimalkan kesalahan dalam pengelompokan nasabah.
- **Peningkatan Inklusi Finansial:** Dengan lebih banyak nasabah yang diklasifikasikan sebagai non-default, Home Credit dapat memberikan lebih banyak pinjaman secara bertanggung jawab.
- **Optimalisasi Proses Pemberian Kredit:** Model membantu perusahaan lebih fokus pada nasabah dengan risiko tinggi gagal bayar, mendukung strategi mitigasi risiko yang lebih efektif.

Peluang untuk Evaluasi Lebih Lanjut: Meskipun model menunjukkan performa yang baik, penting untuk memastikan bahwa nasabah yang sebenarnya memiliki risiko tinggi gagal bayar tetap dapat teridentifikasi dengan baik. Hal ini dapat dicapai dengan:

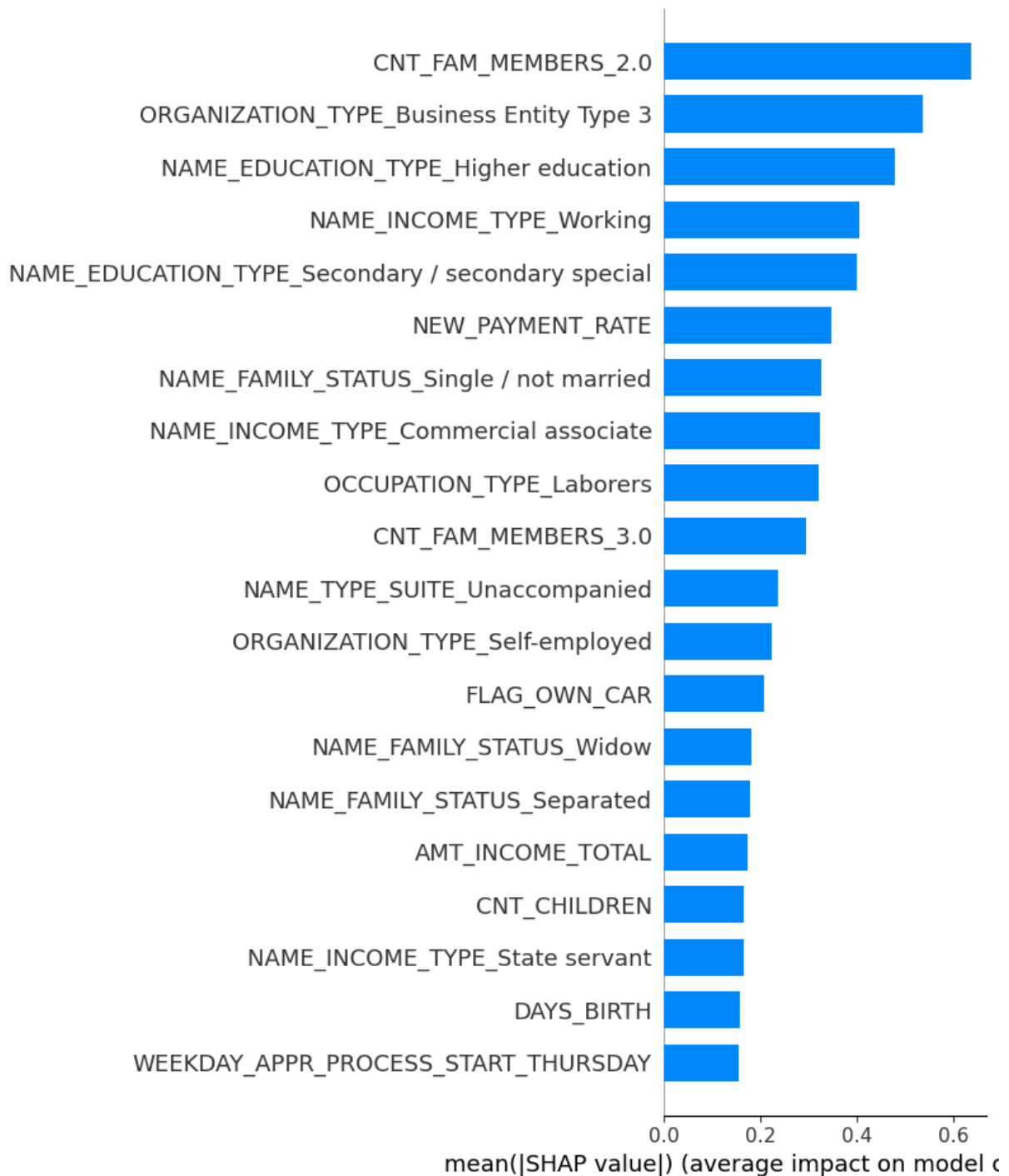
1. Menyesuaikan ambang batas (*threshold*) untuk meningkatkan recall pada kelas default.
2. Melakukan analisis lebih dalam terhadap variabel utama yang berkontribusi pada risiko gagal bayar untuk memastikan keakuratan prediksi.

2. Feature Importance

A. Evaluasi feature

Berdasarkan grafik SHAP, berikut adalah evaluasi singkat dari fitur-fitur paling penting:

1. **CNT_FAM_MEMBERS_2.0 (Jumlah Anggota Keluarga = 2)**
 - a. Fitur ini memiliki dampak terbesar pada prediksi model.
 - b. Nilai SHAP menunjukkan bahwa nasabah dengan 2 anggota keluarga cenderung memiliki risiko default yang lebih rendah.
2. **ORGANIZATION_TYPE_Business Entity Type 3**
 - a. Jenis organisasi tempat nasabah bekerja memiliki pengaruh signifikan terhadap model.
 - b. Nasabah yang bekerja di organisasi tipe ini lebih berpotensi memengaruhi prediksi risiko gagal bayar.



3. **NAME_EDUCATION_TYPE_Higher Education**

- Pendidikan tinggi memainkan peran penting dalam prediksi.
- Nilai SHAP menunjukkan bahwa nasabah dengan pendidikan lebih tinggi cenderung memiliki risiko default yang lebih rendah.

4. **NAME_INCOME_TYPE_Working**

- Status pekerjaan sebagai pekerja aktif memiliki dampak besar pada prediksi.
- Nilai SHAP menunjukkan bahwa nasabah dalam kategori ini dapat memiliki risiko default yang bervariasi.

5. **NAME_EDUCATION_TYPE_Secondary / Secondary Special**

- a. Pendidikan menengah juga merupakan faktor penting.
 - b. Fitur ini memiliki pengaruh yang cukup signifikan terhadap prediksi risiko default.
6. **NEW_PAYMENT_RATE (Rasio Pembayaran)**
 - a. Rasio pembayaran terhadap total kredit memiliki hubungan signifikan dengan risiko gagal bayar.
 - b. Nilai SHAP menunjukkan bahwa rasio yang lebih tinggi berhubungan dengan penurunan risiko default.
7. **NAME_FAMILY_STATUS_Single / Not Married**
 - a. Status keluarga menunjukkan pengaruh penting pada model.
 - b. Nilai SHAP menunjukkan bahwa status ini dapat memengaruhi prediksi risiko default.
8. **NAME_INCOME_TYPE_Commercial Associate**
 - a. Jenis pendapatan sebagai "Commercial Associate" memberikan pengaruh yang terlihat pada model.
 - b. Nilai SHAP mengindikasikan hubungan fitur ini dengan risiko gagal bayar.
9. **OCCUPATION_TYPE_Laborers**
 - a. Jenis pekerjaan sebagai buruh memiliki dampak yang signifikan terhadap model.
 - b. Nilai SHAP menunjukkan hubungan kuat antara pekerjaan ini dan risiko default.
10. **CNT_FAM_MEMBERS_3.0 (Jumlah Anggota Keluarga = 3)**
 - a. Anggota keluarga sebanyak 3 orang juga memberikan pengaruh pada model.
 - b. Nilai SHAP menunjukkan bahwa fitur ini memiliki dampak berbeda dibandingkan dengan keluarga lebih kecil.

Dari hasil ini, fitur-fitur tersebut dapat menjadi fokus utama dalam analisis lebih lanjut atau strategi bisnis, seperti dalam proses pemberian kredit yang lebih personal.

Plot SHAP ini memberikan gambaran rinci tentang pentingnya fitur dan dampaknya terhadap prediksi model. Setiap fitur memiliki pengaruh terhadap prediksi yang ditampilkan di sumbu x, di mana:

Pentingnya Fitur:

- Plot SHAP menunjukkan bagaimana masing-masing fitur memengaruhi prediksi model.
- Fitur dengan nilai **mean(|SHAP value|)** tinggi, seperti **CNT_FAM_MEMBERS_2.0**, memiliki dampak terbesar pada hasil prediksi model.

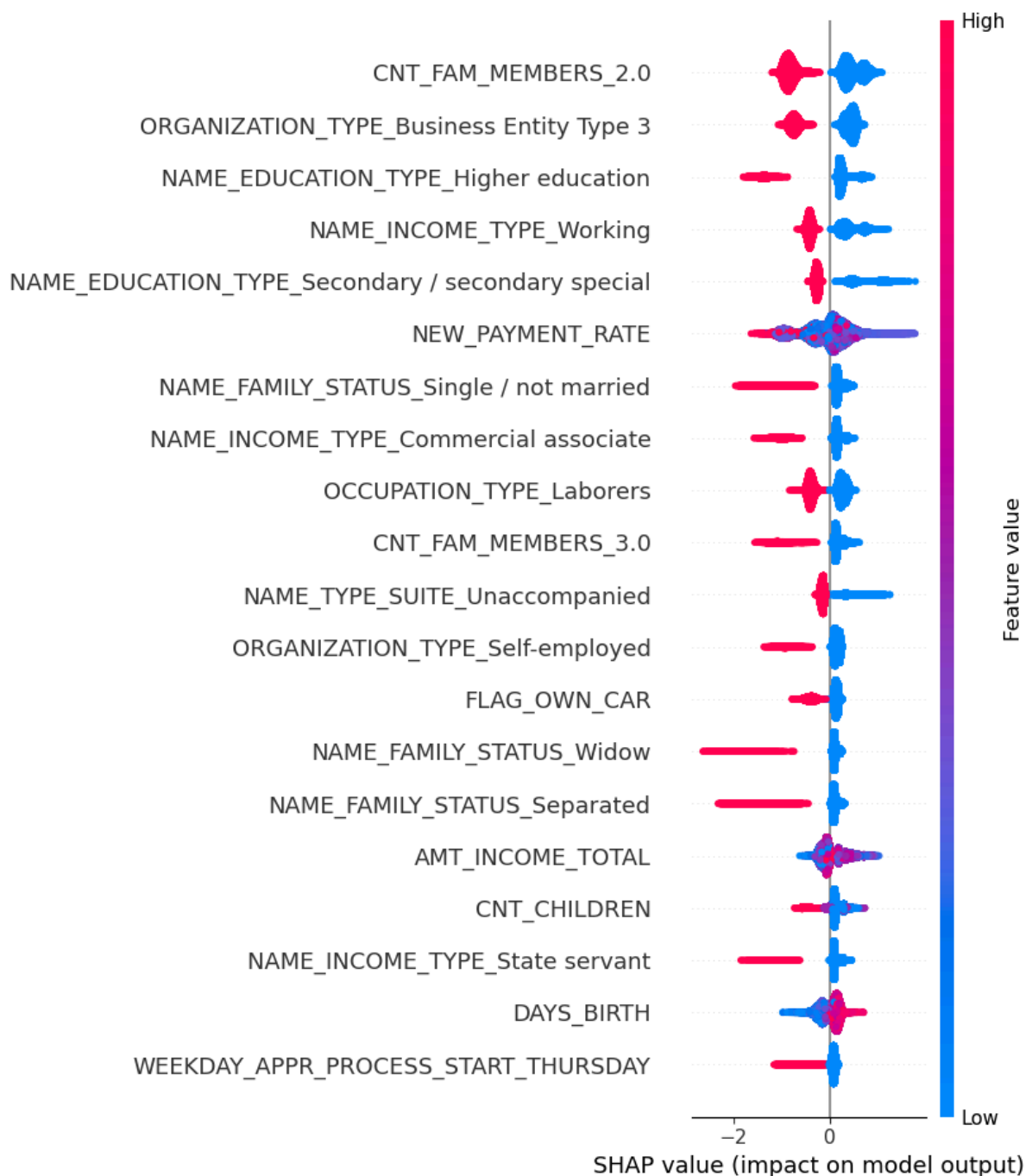
Dampak Fitur terhadap Prediksi (Sumbu X):

Pentingnya Fitur:

- Plot SHAP menunjukkan bagaimana masing-masing fitur memengaruhi prediksi model.
- Fitur dengan nilai **mean(|SHAP value|)** tinggi, seperti **CNT_FAM_MEMBERS_2.0**, memiliki dampak terbesar pada hasil prediksi model.

Dampak Fitur terhadap Prediksi (Sumbu X):

- Sumbu x pada grafik SHAP menggambarkan seberapa besar pengaruh fitur terhadap keputusan model (positif atau negatif).
- Nilai SHAP positif berarti fitur tersebut meningkatkan peluang masuk ke kategori risiko tinggi (default).
- Nilai SHAP negatif berarti fitur tersebut menurunkan peluang masuk ke kategori risiko tinggi.



Variasi Nilai Fitur:

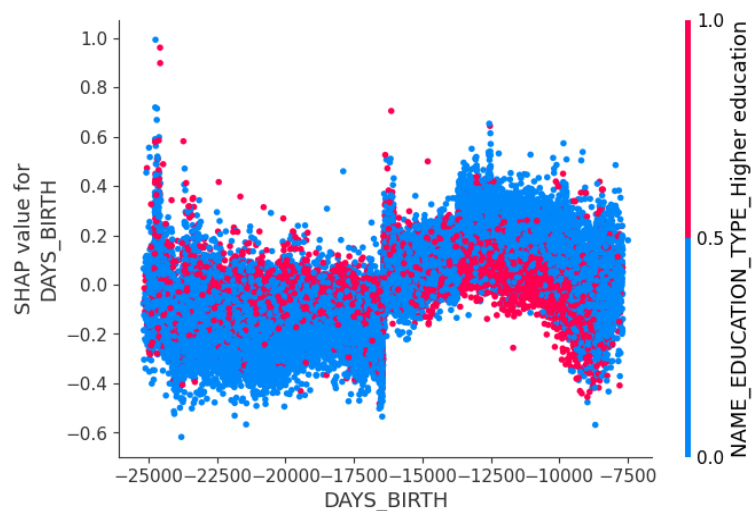
- **Warna merah** menunjukkan nilai fitur yang tinggi, sedangkan **warna biru** menunjukkan nilai fitur yang rendah.
- Misalnya, pada **DAYS_BIRTH (usia)**, nilai lebih tinggi (warna biru, usia lebih tua) cenderung menurunkan risiko gagal bayar.

Interaksi Antar Fitur:

- Dalam plot interaksi (seperti pada DAYS_BIRTH vs. CNT_FAM_MEMBERS_2.0), dapat dilihat bagaimana dua fitur bekerja bersama dalam memengaruhi prediksi model.

Analisis Penyebaran Titik pada Grafik Interaksi (SHAP)

Penyebaran titik-titik menunjukkan distribusi fitur dan variabilitas dampaknya. Gradasi warna dari biru (nilai fitur rendah) ke merah (nilai fitur tinggi) memberikan wawasan tentang bagaimana rentang nilai tertentu berkontribusi terhadap prediksi.



1. Penjelasan Grafik

Grafik ini menunjukkan hubungan antara dua fitur, yaitu:

- **DAYS_BIRTH** (jumlah hari sejak lahir, berhubungan dengan usia nasabah):
 - Semakin besar nilai negatifnya (di sebelah kiri), semakin tua usia nasabah.
 - Semakin mendekati 0 (di sebelah kanan), semakin muda usia nasabah.
- **CNT_FAM_MEMBERS_2.0** (nasabah dengan 2 anggota keluarga):
 - Gradasi warna dari biru (tidak memiliki 2 anggota keluarga) hingga merah (memiliki 2 anggota keluarga).

Sumbu y menunjukkan nilai SHAP untuk fitur **DAYS_BIRTH**, yang menggambarkan dampaknya terhadap prediksi risiko gagal bayar.

2. Wawasan dari Penyebaran Titik

1. Dampak Usia pada Risiko Gagal Bayar:

- Untuk usia yang lebih tua (di sebelah kiri grafik, DAYS_BIRTH rendah), nilai SHAP cenderung negatif, artinya usia yang lebih tua cenderung **menurunkan risiko gagal bayar**.
 - Untuk usia yang lebih muda (di sebelah kanan grafik, DAYS_BIRTH tinggi), nilai SHAP cenderung positif, artinya usia yang lebih muda dapat **meningkatkan risiko gagal bayar**.
2. **Interaksi dengan CNT_FAM_MEMBERS_2.0:**
- **Warna Merah (2 anggota keluarga):**
 - Pada nasabah dengan 2 anggota keluarga (warna merah), risiko gagal bayar lebih stabil di berbagai rentang usia. Namun, risiko mereka lebih rendah pada usia yang lebih tua.
 - **Warna Biru (bukan 2 anggota keluarga):**
 - Pada nasabah yang tidak memiliki 2 anggota keluarga (warna biru), risiko gagal bayar cenderung lebih bervariasi dengan perubahan usia.
3. **Variabilitas Dampak:**
- Titik-titik tersebar luas pada sumbu y, menunjukkan bahwa dampak fitur DAYS_BIRTH terhadap prediksi risiko gagal bayar bervariasi berdasarkan nilai dan interaksi dengan jumlah anggota keluarga.

3. Kesimpulan

- **Nasabah yang lebih tua** (DAYS_BIRTH lebih rendah) cenderung memiliki risiko gagal bayar yang lebih rendah, terutama jika mereka memiliki **2 anggota keluarga**.
- **Nasabah yang lebih muda** (DAYS_BIRTH lebih tinggi) cenderung memiliki risiko gagal bayar yang lebih tinggi, dengan variabilitas yang lebih besar jika jumlah anggota keluarga mereka berbeda dari 2.
- Grafik ini memberikan wawasan tentang bagaimana kombinasi usia dan jumlah anggota keluarga memengaruhi keputusan model, dan bagaimana nilai fitur ini dapat membantu menjelaskan prediksi risiko gagal bayar.

B. Business Insight

Berdasarkan tiga grafik yang diberikan, berikut adalah beberapa insight bisnis dan rekomendasi yang dapat diambil:

1. Insights dari Grafik Feature Importance

1. **CNT_FAM_MEMBERS (Jumlah Anggota Keluarga):**
 - Anggota keluarga sebanyak **2 orang** memiliki pengaruh tertinggi pada model.
 - Artinya, nasabah dengan keluarga kecil cenderung memiliki risiko gagal bayar yang berbeda dibandingkan dengan mereka yang memiliki keluarga besar.
2. **Insight:**
 - Nasabah dengan keluarga kecil mungkin memiliki stabilitas finansial yang lebih baik, sehingga risiko default lebih rendah.
3. **ORGANIZATION_TYPE_Business Entity Type 3:**
 - Jenis organisasi tempat nasabah bekerja memiliki dampak signifikan.

- Organisasi seperti "Business Entity Type 3" mungkin menunjukkan jenis pekerjaan atau sektor yang kurang stabil secara finansial.
 - 4. **Insight:**
 - Pekerjaan di organisasi dengan risiko tinggi dapat meningkatkan risiko default nasabah.
 - 5. **NAME_EDUCATION_TYPE_Higher Education:**
 - Tingkat pendidikan memengaruhi risiko gagal bayar. Nasabah dengan pendidikan tinggi lebih cenderung memiliki risiko default yang rendah.
 - 6. **Insight:**
 - Pendidikan tinggi dapat dikaitkan dengan pendapatan yang lebih stabil dan kemampuan manajemen keuangan yang lebih baik.
 - 7. **NEW_PAYMENT_RATE (Rasio Pembayaran):**
 - Rasio antara jumlah pembayaran terhadap total kredit memiliki pengaruh signifikan.
 - Rasio pembayaran yang lebih tinggi menunjukkan kemampuan nasabah untuk memenuhi kewajiban kreditnya.
 - 8. **Insight:**
 - Nasabah dengan rasio pembayaran rendah perlu diperhatikan lebih lanjut karena mereka berpotensi memiliki risiko gagal bayar yang lebih tinggi.
 - 9. **NAME_FAMILY_STATUS_Single/Not Married:**
 - Status keluarga "Single/Not Married" memengaruhi risiko gagal bayar.
 - Status ini mungkin terkait dengan beban finansial yang lebih rendah dibandingkan dengan mereka yang sudah menikah.
 - 10. **Insight:**
 - Nasabah yang belum menikah memiliki lebih banyak fleksibilitas dalam pengeluaran, tetapi mungkin juga lebih rentan terhadap gangguan keuangan mendadak.
-

2. Insights dari SHAP Value (Plot Kedua)

- **Warna merah** menunjukkan nilai fitur yang tinggi, sedangkan **biru** menunjukkan nilai fitur yang rendah.
- Misalnya, untuk fitur **DAYS_BIRTH** (usia), nilai SHAP yang tinggi menunjukkan bahwa usia tertentu dapat menurunkan atau meningkatkan risiko default.

Key Insight:

- Usia nasabah, jenis pekerjaan, dan rasio pembayaran adalah faktor-faktor kunci yang sangat memengaruhi prediksi model. Nilai SHAP ini membantu menjelaskan bagaimana setiap fitur memengaruhi keputusan model.
-

3. Insights dari Interaksi (Plot Ketiga - DAYS_BIRTH vs CNT_FAM_MEMBERS_2.0)

- **Usia dan Jumlah Anggota Keluarga:**

- Interaksi antara usia (DAYS_BIRTH) dan anggota keluarga memengaruhi risiko default.
- Nasabah yang lebih tua dan memiliki keluarga kecil cenderung memiliki risiko gagal bayar yang lebih rendah.

Insight:

- Usia yang lebih tua sering kali menunjukkan kedewasaan finansial dan pengelolaan keuangan yang lebih baik, sementara keluarga kecil menurunkan beban pengeluaran.

C. Rekomendasi Bisnis

1. Segmentasi Nasabah Berdasarkan Risiko:

- Gunakan variabel kunci seperti **CNT_FAM_MEMBERS**, **NEW_PAYMENT_RATE**, dan **NAME_EDUCATION_TYPE** untuk membuat segmentasi nasabah.
- Fokuskan analisis risiko pada nasabah yang bekerja di organisasi dengan risiko tinggi (misalnya, "Business Entity Type 3") dan memiliki rasio pembayaran rendah.

2. Kebijakan Pemberian Kredit yang Berbasis Data:

- Berikan penawaran kredit yang lebih kompetitif kepada nasabah dengan pendidikan tinggi, keluarga kecil, dan rasio pembayaran tinggi.
- Lakukan verifikasi tambahan untuk nasabah dengan pekerjaan di sektor berisiko atau yang menunjukkan pola pengeluaran tidak stabil.

3. Intervensi Khusus untuk Usia Tertentu:

- Berikan panduan atau produk keuangan yang sesuai untuk nasabah yang lebih muda (usia kerja awal) untuk membantu mereka mengelola keuangan lebih baik.

4. Pemantauan Nasabah Risiko Tinggi:

- Implementasikan pengawasan tambahan untuk nasabah dengan **NEW_PAYMENT_RATE** rendah dan bekerja di sektor rentan.
- Berikan program edukasi keuangan untuk membantu mereka memahami pentingnya pembayaran tepat waktu.

5. Optimalisasi Penilaian Risiko:

- Gunakan SHAP values untuk mengidentifikasi dan memberikan penilaian risiko secara individual, sehingga keputusan kredit menjadi lebih akurat dan terpersonalisasi.

Kesimpulan: Model XGBoost berhasil mendukung tujuan bisnis untuk mengurangi kesalahan identifikasi default dan meningkatkan akurasi pemberian pinjaman. Namun, evaluasi lebih lanjut diperlukan untuk memastikan keseimbangan antara pengurangan kesalahan default dan deteksi risiko gagal bayar secara efektif. Dengan demikian, Home Credit dapat meminimalkan risiko kerugian finansial sekaligus meningkatkan inklusi finansial secara bertanggung jawab.