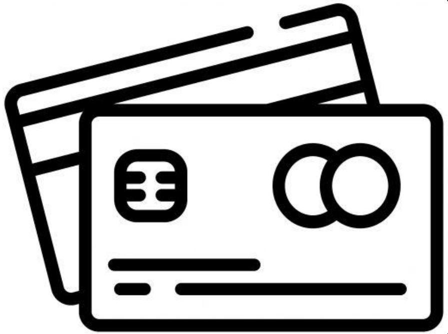
CREDIT RISK
PREDICTION ANALYSIS



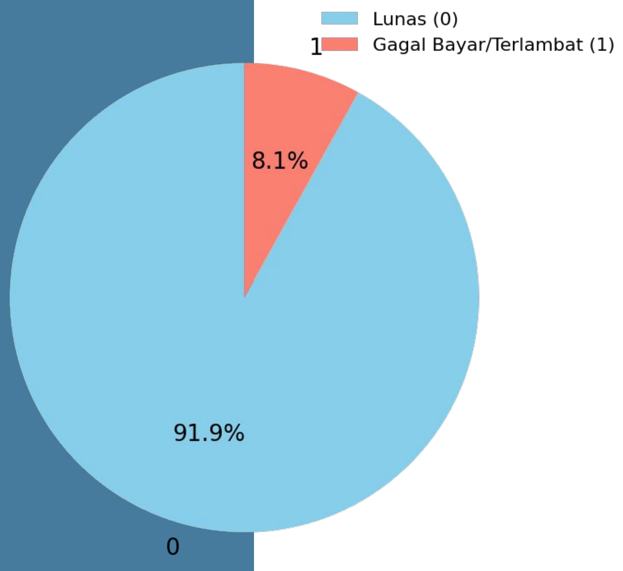
FINAL PROJECT

Project Based Virtual Intern: Data Scientist Rakamin x Home Credit

Francesco Theodore Budiman

Latar Belakang dan Tujuan

Tingkat Keberhasilan Pembayaran Pinjaman Keseluruhan



Total Gagal Bayar: **Rp13,8 Miliar**

Latar Belakang Permasalahan

Home Credit menghadapi **tingginya kredit macet (NPL 8.1%)**, menyebabkan **kerugian signifikan.** Perusahaan ingin **mengurangi risiko gagal bayar** tanpa menghambat peluang pemberian kredit kepada calon debitur yang layak.

Goal

- Mengidentifikasi **karakteristik debitur berisiko** dan **potensial**
- Memprediksi **probabilitas gagal bayar** seakurat mungkin
- Mendukung **keputusan kredit** yang **lebih cepat** dan **tepat**
- Menurunkan **NPL dari 8.1%** → **5%** dalam 1 tahun

Dataset yang digunakan

1

application_{train|test}.csv

Tabel utama, dibagi menjadi dataset train dan test.

Kolom yang digunakan:
CNT_CHILDREN, AMT_CREDIT, AMT_INCOME_TOTAL, DAYS_BIRTH, REGION_POPULATION_RELATIVE, NAME_EDUCATION_TYPE, REGION_RATING_CLIENT_CITY, TARGET

2

bureau.csv

Riwayat pinjaman dari **institusi pinjaman lain** yang dilaporkan kepada Home Credit.

Kolom yang digunakan:
DAYS_CREDIT

3

previous_application.csv

Riwayat pinjaman dari **peminjam sebelumnya** yang pernah **diajukan ke Home Credit**.

Kolom yang digunakan:
NAME_CONTRACT_STATUS: 'Approved`, `Canceled`, `Refused`, `Unused Offer` (hanya digunakan untuk eksplorasi analisis data)

Data Preprocessing

1. Integrasi Data

   train.csv & previous_application → train.csv + `NAME_CONTRACT_STATUS` column

   train.csv & bureau → train.csv + ‘DAYS_CREDIT` → ‘AVG_DAYS_CREDIT` column

2. Menangani missing value (eliminasi kolom dengan missing value lebih dari 40% dan menyeleksi kembali kolom-kolom final yang akan digunakan untuk training model)

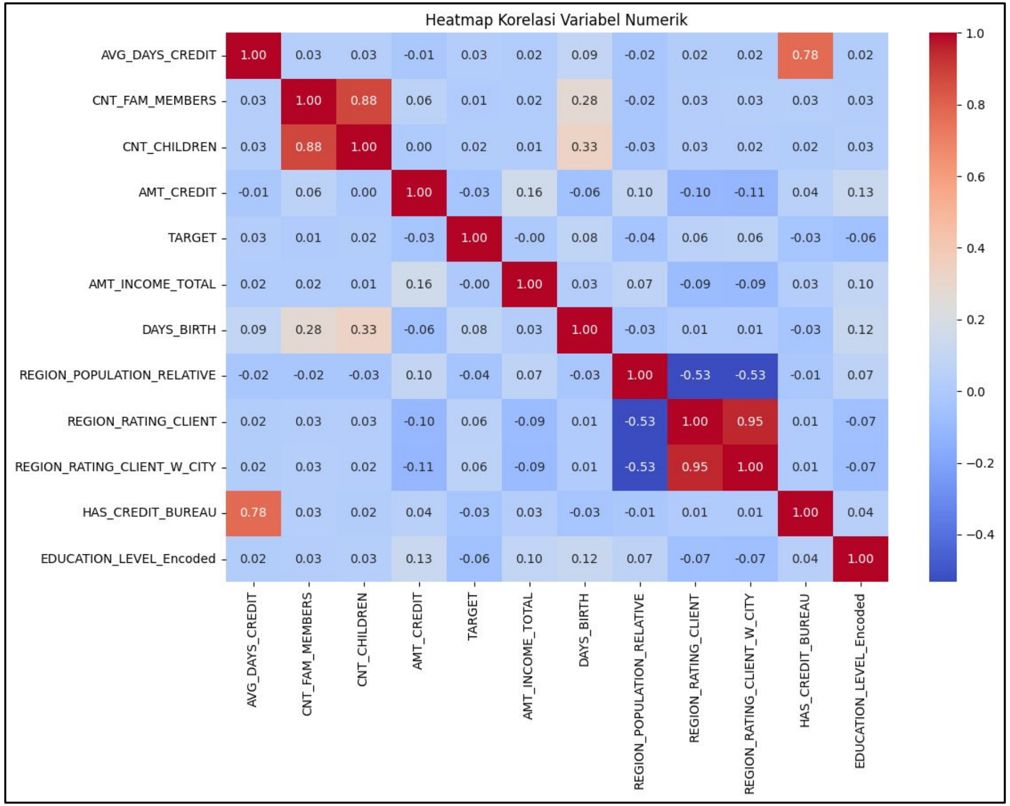
3. Membuat kolom tambahan EDUCATION_LEVEL_Encoded dari kolom EDUCATION_LEVEL, yang merupakan penyederhanaan kategori dari NAME_EDUCATION_TYPE: ‘Higher Education’ (untuk lulusan sarjana hingga doktor) dan ‘Lower Education’ (untuk lulusan sekolah menengah)

Data Preparation

4. Mengecek korelasi antar variabel

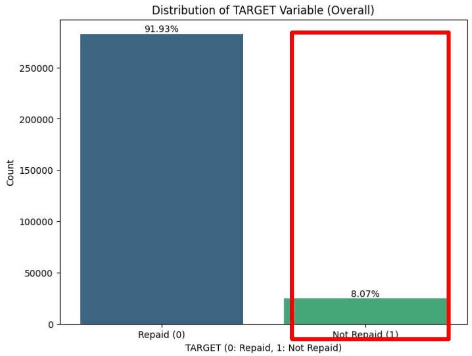
Beberapa kolom yang menyebabkan korelasi tinggi dengan variabel lain akan dieliminasi:

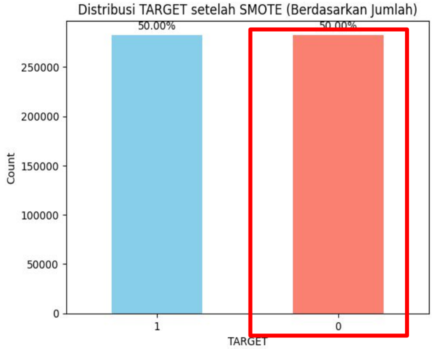
`REGION_RATING_CLIENT`
`HAS_CREDIT_BUREAU`
`CNT_FAM_MEMBERS`



Data Preparation

5. Menangani data target yang tidak seimbang





Reference: [SMOTE]

6. Feature scaling

	AVG_DAYS_CREDIT	CNT_CHILDREN	AMT_CREDIT	AMT_INCOME_TOTAL	DAYS_BIRTH
490359	-2922.000000	1	2.423823e+05	144000.000000	-16615
410241	-1279.761936	0	6.964835e+05	67500.000000	-20538
294752	-2855.500000	0	1.724220e+06	180000.000000	-14789
128893	-1300.666667	0	8.910000e+05	405000.000000	-18132
262129	-572.500000	0	5.450400e+05	180000.000000	-10565

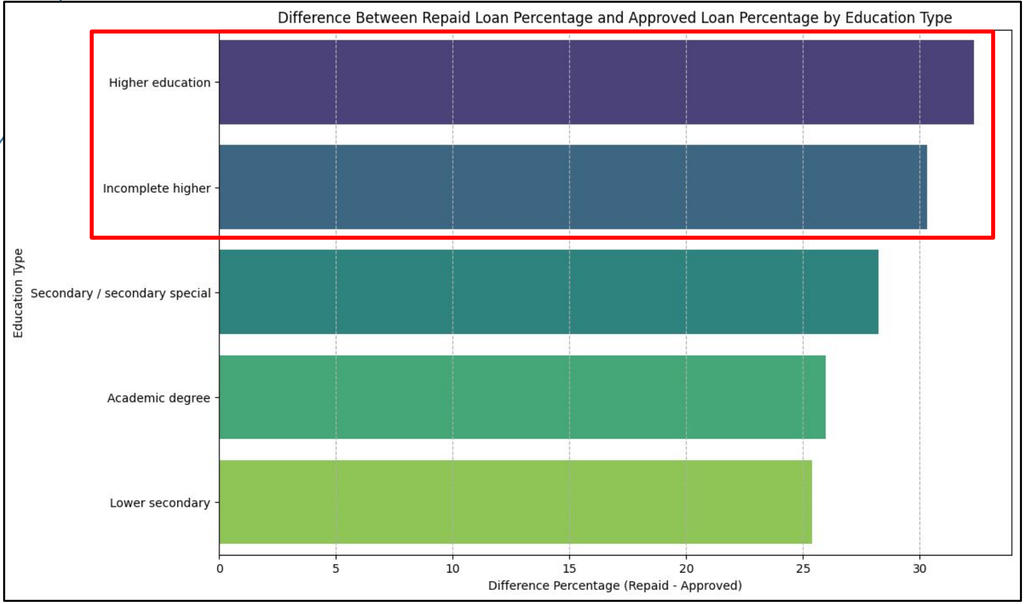
	AVG_DAYS_CREDIT	CNT_CHILDREN	AMT_CREDIT	AMT_INCOME_TOTAL	DAYS_BIRTH
490359	-1.807903	1.0	-0.528054	0.00	-0.229145
410241	-0.166479	0.0	0.373029	-0.85	-0.800262
294752	-1.741436	0.0	2.412389	0.40	0.036687
128893	-0.187374	0.0	0.759012	2.90	-0.449993
262129	0.540432	0.0	0.072516	0.40	0.651623

Reference: [Feature Scaling]

7. Pemisahan dataset tabel utama (application.train) menjadi data train (80%) dan data test (20%)

HOME CREDIT Rakamin

Higher Education (Gelar Sarjana) dan Incomplete Higher (Mahasiswa) merupakan status pendidikan dengan selisih persentase repaid dan approved tertinggi



NAME_EDUCATION_TYPE	Repaid		Approved (prev Tx)		Difference
	Borrower	%	Borrowing	%	
Higher education	70854	94.64	195002	62.33	32.31
Incomplete higher	9405	91.52	27676	61.22	30.3
Secondary / secondary special	198867	91.06	652074	62.83	28.23
Academic degree	161	98.17	418	72.19	25.98
Lower secondary	3399	89.07	10929	63.67	25.4

NB:
*Approved (previous application): pengajuan pinjaman sebelumnya yang disetujui dan dicairkan oleh peminjam

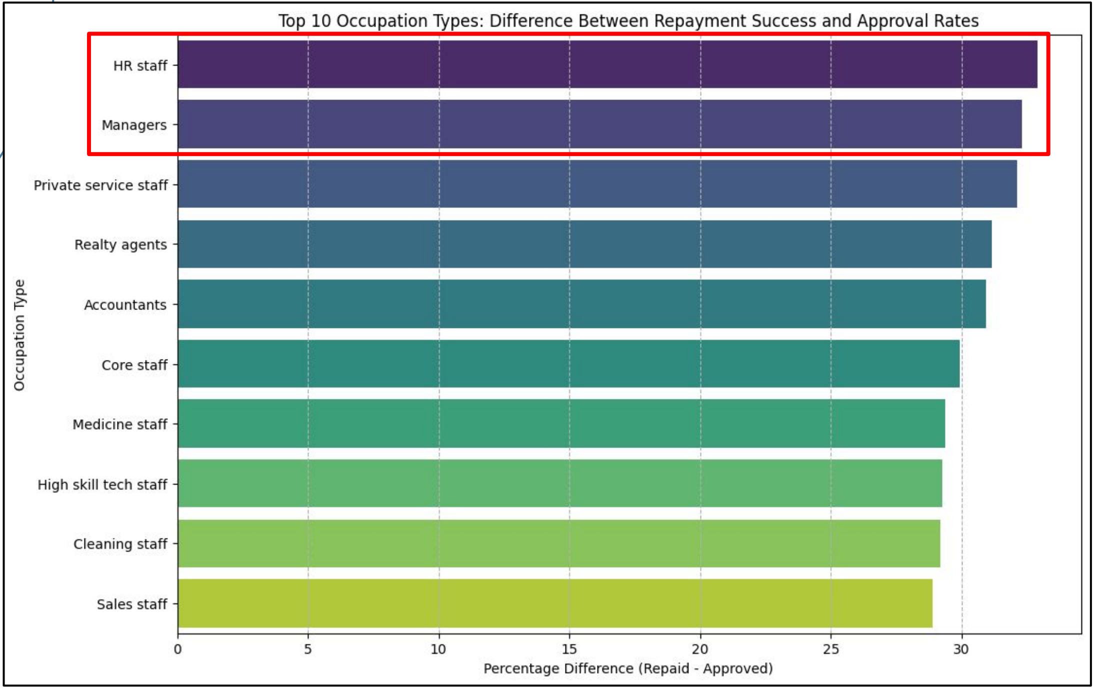
Previous Application:

Academic Degree;
Canceled: 11.05%, Unused: 15.72%

Incomplete Higher;
Canceled: 18.24%, Unused: 17.28%

HOME CREDIT Rakamin

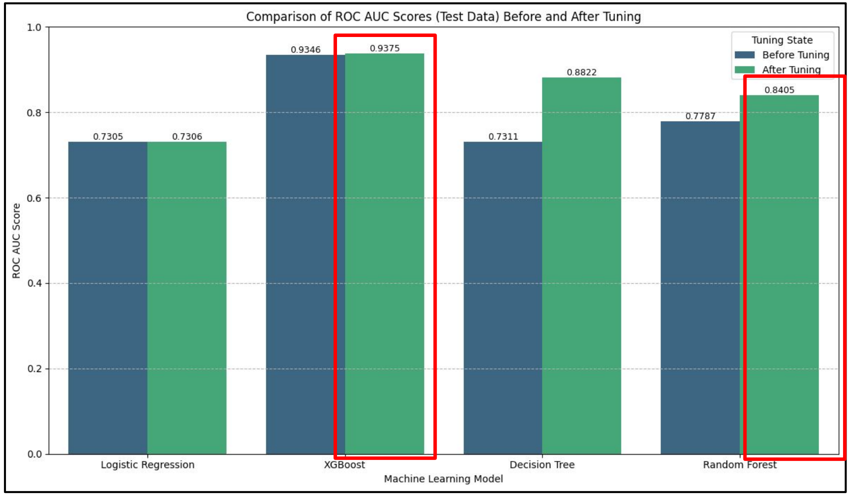
HR Staff dan Manager adalah profesi dengan selisih persentase pengembalian kredit serta persetujuan dan pencairan kredit tertinggi

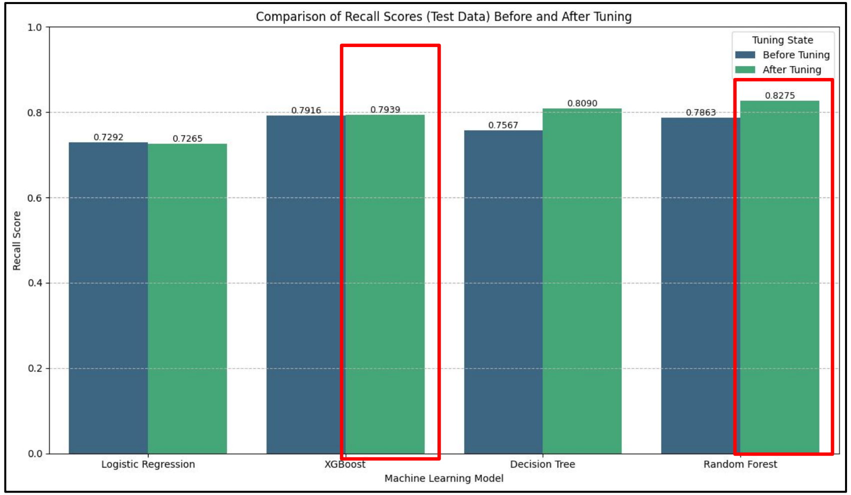


OCCUPATION_TYPE	Repaid		Approved (prev Tx)		Difference
	Borrower	%	Borrowing	%	
HR staff	527	93.61	1404	60.7	32.91
Managers	20043	93.79	58371	61.48	32.31
Private service staff	2477	93.4	7498	61.27	32.13
Realty agents	692	92.14	2161	60.98	31.16
Accountants	9339	95.17	26957	64.25	30.92
Core staff	25832	93.7	75415	63.76	29.94
Medicine staff	7965	93.3	25574	63.94	29.36
High skill tech staff	10679	93.84	31306	64.57	29.27
Cleaning staff	4206	90.39	14888	61.2	29.19
Sales staff	29010	90.37	93060	61.48	28.89

Previous Application:
HR Staff; Canceled: 19.67%, Unused: 1.82%
Manager; Canceled: 17.74%, Unused: 1.67%

Machine Learning Implementation and Evaluation





Meskipun Random Forest memiliki sedikit keunggulan dalam nilai recall (0.82 vs 0.79) untuk prediksi kegagalan tetapi XGBoost memiliki keunggulan yang lebih signifikan dalam AUC (0.93 vs 0.84) (membedakan keberhasilan dan kegagalan).

Performa XGBoost pada data test:
Accuracy : 0.8746 (dari keseluruhan kasus baik kredit macet ataupun tidak macet)
Precision : 0.9466 (dari keseluruhan yang diprediksi macet)
Recall : 0.7939 (dari semua debitur yang mengalami kredit macet)

Reference: [Model Evaluation](), [ROC-AUC]()

REKOMENDASI

1) Permudah proses persetujuan

Target utama:
HR, Manager, Lulusan Sarjana & Mahasiswa

Implementasi:
1) fast auto-approve selama tidak ada masalah di histori kredit & probability return tinggi
2) tawarkan pinjaman kedua secara otomatis apabila peminjam lancar melunaskan dalam periode cicilan 3-6 bulan

Ex Target KPI/Success Metrics:
1) Return rate meningkat +5% dalam 1 tahun
2) Waktu proses persetujuan berkurang 20–30%

2) Atasi masalah cancel/unused credit

Target utama:
HR, Manager, Lulusan Sarjana & Mahasiswa

Implementasi:
1) terapkan bunga lebih rendah (ex: 0.3-0.5%) untuk segmen pengguna tersebut atau peminjam lainnya dengan histori kredit yang baik & probability return tinggi
2) customer insight survey kepada peminjam Home Credit untuk mengetahui faktor penyebab lebih lanjut mengenai keputusan unused atau canceled loan.

Ex Target KPI/Success Metrics:
1) 50% peminjam mengisi survey dalam 6 bulan
2) Cancel/unused rate menurun 5% dalam 1 tahun

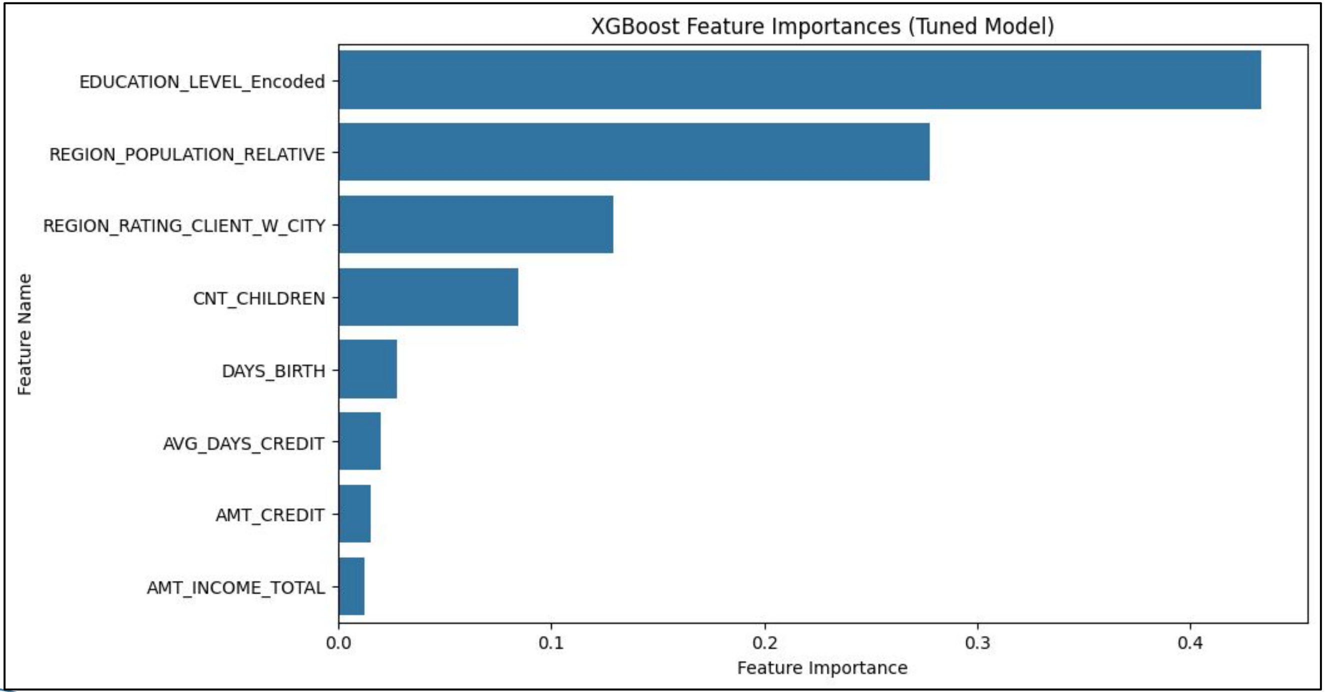
THANK YOU !!

APPENDIX

Presentation Video, Github

Feature Importance

Education Level merupakan variabel yang paling memengaruhi hasil prediksi dengan kontribusi 40% dibandingkan 7 variabel lainnya.



Logistic Regression - Hyperparameter

```
logreg_param_grid = {
    #'C': [0.001],
    'C': [0.001, 0.01, 0.1, 1, 10, 100],
    #'penalty': ['l1'],
    'penalty': ['l1', 'l2'],
    #'solver': ['liblinear'],
    'solver': ['liblinear', 'saga'],
    #'class_weight': [None],
    'class_weight': [None, 'balanced'],
    #'tol' = 0.01
    'tol': [1e-4, 1e-3, 1e-2],
}
print("Logistic Regression hyperparameter grid defined successfully.")
```

```
Logistic Regression hyperparameter grid defined successfully.
```

Logistic Regression - Inisiasi Model

[134]
50m

```
# Inisialisasi model Logistic Regression
logreg_model = LogisticRegression(random_state=42)

# Inisialisasi GridSearchCV
grid_search_logreg = GridSearchCV(
    estimator=logreg_model,
    param_grid=logreg_param_grid,
    scoring='roc_auc',
    cv=StratifiedKFold(n_splits=3, shuffle=True, random_state=42),
    n_jobs=-1,  # Gunakan semua core prosesor yang tersedia
    verbose=2
)

# Fit GridSearchCV
print("Starting GridSearchCV for Logistic Regression...")
grid_search_logreg.fit(X_train, y_train)
print("GridSearchCV for Logistic Regression completed.")
```

```
Starting GridSearchCV for Logistic Regression...
Fitting 3 folds for each of 144 candidates, totalling 432 fits
GridSearchCV for Logistic Regression completed.
```

Logistic Regression - Best Parameter

```
# Dapatkan model terbaik
best_logreg_model = grid_search_logreg.best_estimator_

print("Best Logistic Regression Model:")
print(best_logreg_model)

print("Best Parameters Found:")
print(grid_search_logreg.best_params_)

print("Best ROC AUC Score (from cross-validation):")
print(grid_search_logreg.best_score_)
```

```
Best Logistic Regression Model:
LogisticRegression(C=0.001, penalty='l1', random_state=42, solver='liblinear',
                   tol=0.01)
Best Parameters Found:
{'C': 0.001, 'class_weight': None, 'penalty': 'l1', 'solver': 'liblinear', 'tol': 0.01}
Best ROC AUC Score (from cross-validation):
0.726807430678765
```

XGBoost - Hyperparameter

[141]
✓ 43m

```
# Define the hyperparameter grid for XGBoost
xgb_param_grid = {
    # 'n_estimators': [150],
    'n_estimators': [50, 100, 150],
    #'max_depth': [7],
    'max_depth': [3, 5, 7],
    #'learning_rate': [0.2],
    'learning_rate': [0.01, 0.1, 0.2],
    #'subsample': [0.8], # Fraction of samples used for fitting the trees
    'subsample': [0.7, 0.8],
    #'colsample_bytree': [0.8] # Fraction of features used for fitting the trees
    'colsample_bytree': [0.7, 0.8],
    # Parameter tambahan:
    'min_child_weight': [1, 3, 5],
}
```

XGBoost - Inisiasi Model

```
# Initialize the XGBoost model
xgb_model = XGBClassifier(random_state=42, use_label_encoder=False, eval_metric='logloss')
# Suppress warning for use_label_encoder and set eval_metric for current XGBoost versions

# Initialize GridSearchCV
grid_search_xgb = GridSearchCV(
    estimator=xgb_model,
    param_grid=xgb_param_grid,
    scoring='roc_auc', # Keep ROC AUC for consistency with previous Logistic Regression evaluation
    cv=StratifiedKFold(n_splits=3, shuffle=True, random_state=42), # Using StratifiedKFold for imbalanced data
    n_jobs=-1,  # Use all available processor cores
    verbose=2
)
```

XGBoost - Best Parameter

```
# Fit GridSearchCV to the training data (resampled and scaled)
print("Starting GridSearchCV for XGBoost...")
grid_search_xgb.fit(X_train, y_train)
print("GridSearchCV for XGBoost completed.")

# Summarize results
print("\nBest XGBoost Model Parameters:")
print(grid_search_xgb.best_params_)

print("\nBest ROC AUC Score (from cross-validation):")
print(grid_search_xgb.best_score_)
```

```
Starting GridSearchCV for XGBoost...
Fitting 3 folds for each of 324 candidates, totalling 972 fits
/usr/local/lib/python3.12/dist-packages/xgboost/training.py:199: UserWarning: [01:18:18] WARNING: /workspace/src/learner.cc:790:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
GridSearchCV for XGBoost completed.

Best XGBoost Model Parameters:
{'colsample_bytree': 0.8, 'learning_rate': 0.2, 'max_depth': 7, 'min_child_weight': 1, 'n_estimators': 150, 'subsample': 0.7}

Best ROC AUC Score (from cross-validation):
0.937046614664979
```

Decision Tree - Hyperparameter

```
dt_param_grid = {
    # 'max_depth': [11],
    'max_depth': [7, 9, 11],
    #'min_samples_split': [2],
    'min_samples_split': [2, 5, 10],
    #'min_samples_leaf': [10],
    'min_samples_leaf': [2, 5, 10],
    #'criterion': ['gini'],
    'criterion': ['gini', 'entropy'],
    'max_features': ['sqrt', 'log2', None],
}
print("Decision Tree hyperparameter grid defined successfully.")
```

```
Decision Tree hyperparameter grid defined successfully.
```

Decision Tree - Inisiasi Model

```
# Initialize Decision Tree Classifier
dt_model = DecisionTreeClassifier(random_state=42)

# Initialize GridSearchCV
grid_search_dt = GridSearchCV(
    estimator=dt_model,
    param_grid=dt_param_grid,
    scoring='roc_auc',
    cv=StratifiedKFold(n_splits=3, shuffle=True, random_state=42),
    n_jobs=-1,
    verbose=2
)

# Fit GridSearchCV
print("Starting GridSearchCV for Decision Tree...")
grid_search_dt.fit(X_train, y_train)
print("GridSearchCV for Decision Tree completed.")
```

```
Starting GridSearchCV for Decision Tree...
Fitting 3 folds for each of 162 candidates, totalling 486 fits
GridSearchCV for Decision Tree completed.
```

Decision Tree - Best Parameter

```
best_dt_model = grid_search_dt.best_estimator_

print("Best Decision Tree Model:")
print(best_dt_model)

print("Best Parameters Found:")
print(grid_search_dt.best_params_)

print("Best ROC AUC Score (from cross-validation):")
print(grid_search_dt.best_score_)
```

```
Best Decision Tree Model:
DecisionTreeClassifier(max_depth=11, min_samples_leaf=10, random_state=42)
Best Parameters Found:
{'criterion': 'gini', 'max_depth': 11, 'max_features': None, 'min_samples_leaf': 10, 'min_samples_split': 2}
Best ROC AUC Score (from cross-validation):
0.8781662539933069
```

Random Forest - Hyperparameter

```
rf_param_grid = {
    'max_depth': [7, 9, 11],
    #'max_depth': [11],
    'min_samples_split': [2, 6, 10],
    #'min_samples_split': [2],
    'min_samples_leaf': [2, 6, 10], # 10
    #'min_samples_leaf': [10],
    'criterion': ['gini'],
    'n_estimators': [100, 200, 300],
}
print("Random Forest hyperparameter grid defined successfully.")
```

```
Random Forest hyperparameter grid defined successfully.
```

Random Forest - Inisiasi Model

```python
# Initialize Random Forest Classifier
rf_model = RandomForestClassifier(random_state=42)

# Initialize GridSearchCV
grid_search_rf = GridSearchCV(
    estimator=rf_model,
    param_grid=rf_param_grid,
    scoring='roc_auc',
    cv=StratifiedKFold(n_splits=3, shuffle=True, random_state=42),
    n_jobs=-1,  # Use all available processor cores
    verbose=2
)

# Fit GridSearchCV
print("Starting GridSearchCV for Random Forest...")
grid_search_rf.fit(X_train, y_train)
print("GridSearchCV for Random Forest completed.")
```

```
Starting GridSearchCV for Random Forest...
Fitting 3 folds for each of 81 candidates, totalling 243 fits
GridSearchCV for Random Forest completed.
```

Execution time: 5 hours

Random Forest - Best Hyperparameter

```
best_dt_model = grid_search_dt.best_estimator_

print("Best Decision Tree Model:")
print(best_dt_model)

print("Best Parameters Found:")
print(grid_search_dt.best_params_)

print("Best ROC AUC Score (from cross-validation):")
print(grid_search_dt.best_score_)
```

```
Best Decision Tree Model:
DecisionTreeClassifier(max_depth=11, min_samples_leaf=10, random_state=42)
Best Parameters Found:
{'criterion': 'gini', 'max_depth': 11, 'max_features': None, 'min_samples_leaf': 10, 'min_samples_split': 2}
Best ROC AUC Score (from cross-validation):
0.8781662539933069
```