Title: Credit Card Default Prediction Using Machine Learning

Muhammad Iqbal Saputra
(2702390236)
Muhammad Reza Alghifari
(2802555044)
Raditya Arya Pradipta
(2802446705)

Artificial Intelligence major
LA-84,Machine learning Lecture

***Abstract*—Predicting credit card defaults accurately is crucial for financial institution to mitigates risk and optimize credit management strategies. This paper explores various machine learning models to predict credit card default using the "Default of Credit Card Clients" dataset from the UCI Machine Learning Repository. Extensive exploratory data analysis (EDA), preprocessing, feature engineering, and model evaluation were conducted. Among the tested algorithms—Random Forest, Support Vector Machine, XGBoost, and K-Nearest Neighbors—Random Forest demonstrated superior performance based on F1-score, while XGBoost offered the highest recall. Additional visualizations and comprehensive evaluations were performed to enhance interpretability. The study confirms the effectiveness of machine learning approaches in improving credit risk predictions.**

Keywords— Credit Default, Machine Learning, Classification, Risk Assessment,XGboost,Random Forest,KNN,SVM

I. INTRODUCTION

Credit default prediction plays a significant role in financial stability and risk management for banks and credit institutions. Accurately predicting the likelihood of default allows financial institutions to manage risks more effectively, allocate resources efficiently, and avoid significant financial losses. Traditional credit scoring methods, such as manual reviews or linear statistical models, often struggle to capture intricate and non-linear patterns present in consumer financial behaviors. These limitations highlight the need for more advanced predictive techniques.

With increasing data availability and advancements in computational power, machine learning has emerged as a powerful tool capable of uncovering complex relationships within large-scale financial datasets. Machine learning algorithms can dynamically adapt to new data patterns, making them particularly suitable for credit risk management. Recent applications have demonstrated improvements in predictive accuracy, leading to more informed and timely decisions regarding loan approvals and risk assessment. The ability to process large and complex datasets efficiently is one of the key advantages of machine learning over traditional methods.

This study aims to leverage these advancements by evaluating various machine learning models Random Forest, Support Vector Machines (SVM), XGBoost, and K-Nearest Neighbors (KNN) to predict credit card defaults effectively. By employing rigorous exploratory data analysis (EDA), extensive data preprocessing, and strategic feature engineering, this research provides a comprehensive analysis of each model's predictive capabilities. The ultimate goal is to identify robust analytical solutions that can be practically integrated into existing financial decision-making processes, enhancing overall credit risk management.

The reason for choosing this topic is driven by the growing need to enhance the accuracy of credit default predictions in the financial sector. Traditional credit scoring methods often fail to capture the intricate, non-linear relationships present in consumer financial behavior. Machine learning provides a more robust solution by leveraging data patterns that evolve over time. This study aims to demonstrate how machine learning can overcome the limitations of traditional scoring methods, providing financial institutions with more reliable and accurate predictions of credit defaults.

By focusing on machine learning models, this study will contribute to improving financial decision-making processes. The evaluation of models like Random Forest, SVM, XGBoost, and KNN will help determine which models offer the best predictive performance. Ultimately, this research aims to provide financial institutions with powerful tools for managing credit risk, resulting in better loan approval decisions and enhanced risk assessment capabilities.

II. LITERATURE REVIEW

Machine learning has significantly impacted financial risk management, especially in the realm of credit scoring and default prediction. Early approaches relied primarily on statistical models such as logistic regression and discriminant analysis, which, although interpretable, often fail to adequately capture complex, non-linear relationships inherent in financial data. Recent literature indicates that

machine learning models, particularly ensemble methods and gradient boosting techniques, are more adept at managing these complexities and improving predictive accuracy.

Yeh and Lien (2009) conducted a comparative analysis of various data mining methods, including decision trees, neural networks, and logistic regression, highlighting significant advantages of advanced machine learning approaches in predicting defaults accurately. Their findings underscore the critical importance of robust data preprocessing and feature engineering strategies in enhancing model performance.

Random Forest models, noted for their flexibility and robustness against overfitting, have been widely used in credit risk prediction. Their ability to handle imbalanced datasets and nonlinear interactions between variables makes them particularly suitable for this domain. Similarly, gradient boosting frameworks like XGBoost have gained popularity due to their high accuracy and efficiency in handling large-scale datasets. These algorithms iteratively improve weak learners, enhancing the overall prediction performance.

Recent research also emphasizes the critical role of handling class imbalance in credit risk datasets, commonly addressed using oversampling techniques like SMOTE (Synthetic Minority Oversampling Technique). Studies have consistently shown that addressing class imbalance significantly improves the sensitivity and specificity of predictive models, leading to better risk assessment outcomes.

Furthermore, the interpretability of machine learning models has become increasingly relevant in the financial sector, where transparency and accountability are essential. Techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) have been applied to enhance the understanding and trustworthiness of complex models by providing clear explanations of individual predictions.

prior research indicates that integrating advanced machine learning algorithms with comprehensive data preprocessing and interpretability measures can substantially improve the effectiveness of credit default prediction models. This study builds upon these findings by systematically evaluating several prominent machine learning techniques, assessing their predictive accuracy, and providing actionable insights for practical implementation.

A relevant study utilizing the same dataset (Taiwan credit card default dataset) conducted a model comparison, particularly focusing on XGBoost, achieving an AUC (ROC) score of 0.7882**,** precision of 21.67%**,** recall of 79.14%**,** and an F1-score of 33.85%. These results, while demonstrating high recall, suffered from relatively low precision, indicating a high number of false positivesan important concern in financial risk applications.

Building upon this foundation, our study explores the performance of several machine learning models—namely Random Forest, XGBoost, Support Vector Classifier (SVC), and K-Nearest Neighbors (KNN)—on the same dataset. Notably, our experimental outcomes show consistent improvements in both precision and F1-score across all models. These enhancements suggest a more balanced prediction capability between identifying actual defaulters and minimizing false alarms. The Random Forest model, in particular, demonstrated superior ROC AUC and F1-score, indicating a stronger generalization ability in capturing the complexities of credit default patterns.

This comparison highlights the evolving effectiveness of ensemble and kernel-based methods in financial classification tasks. Furthermore, it underlines the importance of model selection and hyperparameter optimization tailored to domain-specific characteristics. Consequently, our findings provide not only empirical validation of prior research but also practical insights into deploying robust and interpretable credit risk prediction models.

III. METHODOLOGY

This section outlines the comprehensive methodology adopted in this study to predict the default payment status of credit card clients in the seventh month based on prior six-month data. The process encompasses several phases, including problem understanding, data collection, cleaning, exploratory data analysis (EDA), and preliminary statistical analysis, each described in detail below.

1) Problem Understanding

The dataset utilized in this research originates from the UCI Machine Learning Repository and contains financial and demographic records of 30,000 Taiwanese credit card clients. Each record comprises 24 predictive features and one binary target variable: default.payment.next.month. This target indicates whether the client defaulted on their credit card payment in the upcoming month (where 1 denotes default and 0 denotes no default).

The dataset includes several categories of features. Demographic attributes consist of SEX, EDUCATION, MARRIAGE, and AGE. Financial features include the client's credit limit (LIMIT_BAL) and billing amounts for the past six months (BILL_AMT1 to BILL_AMT6). Payment behavior is captured via repayment status (PAY_0 to PAY_6) and payment amounts (PAY_AMT1 to PAY_AMT6). For the purposes of this study, we focus on using data from the first six months to predict the seventh month's default status. In this formulation, the original target column is retained as the label for supervised classification.

2) Data Collection

```
# Cek info data
df.info()
```

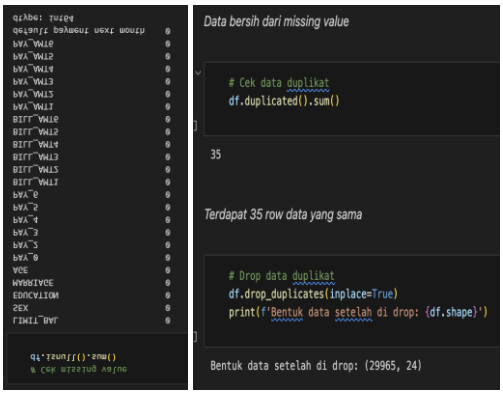
Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 24 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   LIMIT_BAL                   30000 non-null  int64
 1   SEX                         30000 non-null  int64
 2   EDUCATION                   30000 non-null  int64
 3   MARRIAGE                    30000 non-null  int64
 4   AGE                         30000 non-null  int64
 5   PAY_0                       30000 non-null  int64
 6   PAY_2                       30000 non-null  int64
 7   PAY_3                       30000 non-null  int64
 8   PAY_4                       30000 non-null  int64
 9   PAY_5                       30000 non-null  int64
 10  PAY_6                       30000 non-null  int64
 11  BILL_AMT1                   30000 non-null  int64
 12  BILL_AMT2                   30000 non-null  int64
 13  BILL_AMT3                   30000 non-null  int64
 14  BILL_AMT4                   30000 non-null  int64
 15  BILL_AMT5                   30000 non-null  int64
 16  BILL_AMT6                   30000 non-null  int64
 17  PAY_AMT1                    30000 non-null  int64
 18  PAY_AMT2                    30000 non-null  int64
 19  PAY_AMT3                    30000 non-null  int64
...
 22  PAY_AMT6                    30000 non-null  int64
 23  default payment next month  30000 non-null  int64
dtypes: int64(24)
```

The dataset is imported using the pandas library in Python, and several inspection techniques such as df.head(), df.tail(), and df.info() are employed to

understand the shape and structure of the dataset. These functions help verify that there are 30,000 observations and 25 columns, and reveal the data types associated with each attribute. Initial checks confirm that most columns contain numerical data, and categorical variables are encoded as integers. This step lays the groundwork for later transformations and visualizations.

3) C. Data Cleaning

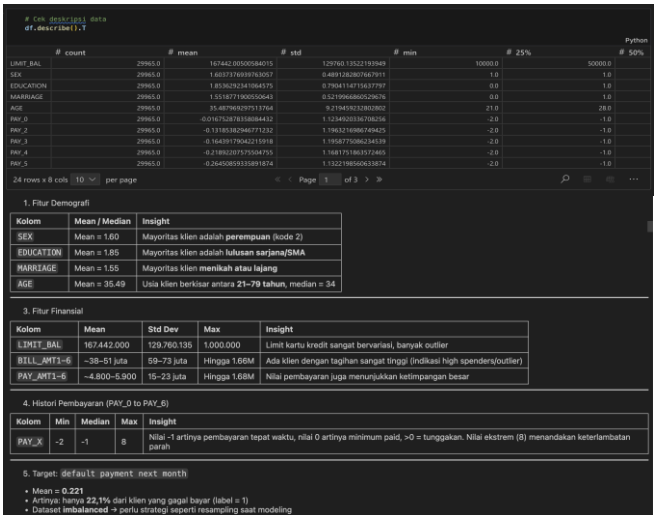


Upon inspection, the dataset is found to be remarkably clean. There are no missing values across any of the columns, simplifying preprocessing steps. However, some categorical columns such as EDUCATION and MARRIAGE contain ambiguous or rarely occurring codes (e.g., EDUCATION=0, MARRIAGE=3) which may represent undefined or unknown categories. These values are retained initially for exploration but are flagged for possible grouping in later modeling stages.

4) D. Exploratory Data Analysis (EDA)

EDA is performed to extract meaningful insights and prepare the dataset for modeling. This phase includes statistical profiling, visual exploration, distribution analysis, outlier detection, and correlation assessment.

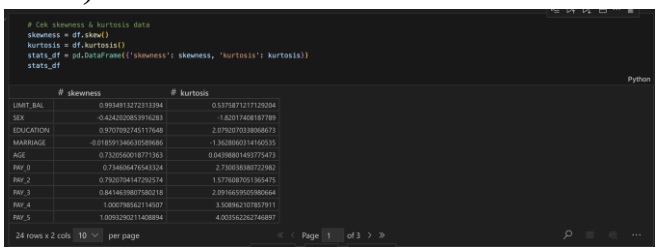
a) Descriptive Statistics and Data Distribution



We begin by summarizing each feature numerically. The demographic distribution indicates that the majority of clients are female, with a rough 60:40 female-to-male ratio. In terms of education, most clients hold either a university degree (category 2) or a high school diploma (category 3). Regarding marital status, single (category 2) and married (category 1) individuals make up most of the dataset, with very few labeled as "others."

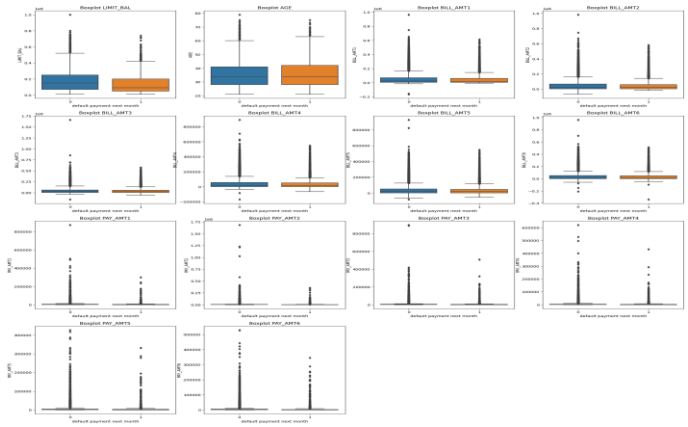
The repayment status variables (PAY_0 to PAY_6) encode monthly repayment behavior. Values range from -2 (no consumption), -1 (paid in full), 0 (no delay), up to 8 (severe delay). The majority of clients either paid in full or were not late, suggesting relatively responsible payment behavior among the population.

b) Skewness and Normalization



To assess data normality, we compute the skewness and kurtosis of numerical variables. Features such as BILL_AMT and PAY_AMT display high skewness, indicating non-normal distributions with heavy tails. For these features, we apply MinMaxScaler normalization to rescale values to a 0–1 range, which also helps stabilize learning for machine learning models.

c) Visual Distribution and Outliers

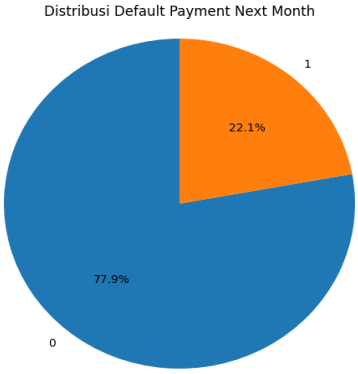


We visualize the distribution of financial and behavioral features using histograms and boxplots, split by the target variable. For instance, clients who defaulted (target = 1) tend to have lower credit limits and make smaller monthly payments on average. Outliers are more pronounced in the non-defaulting group, especially in high credit limits and large bill amounts.

The outlier analysis of LIMIT_BAL shows that defaulting clients typically have lower median credit limits. Although outliers exist in both target classes, extreme high values are more common among non-defaulters. This insight

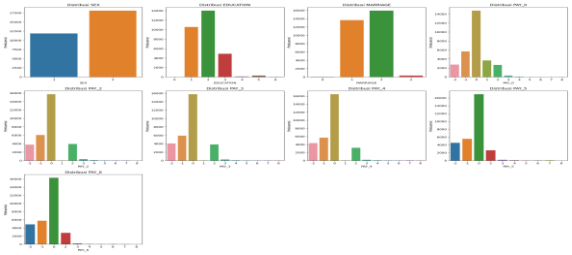
supports the hypothesis that clients with higher creditworthiness are less likely to default.

d) Target Variable Distribution



The dataset is imbalanced: only 22.1% of clients defaulted on their payments, while the remaining 77.9% did not. This imbalance necessitates careful treatment during model training to avoid biased predictions. We consider using sampling techniques such as oversampling or SMOTE in the modeling stage to address this imbalance.

e) Categorical Feature Relationships

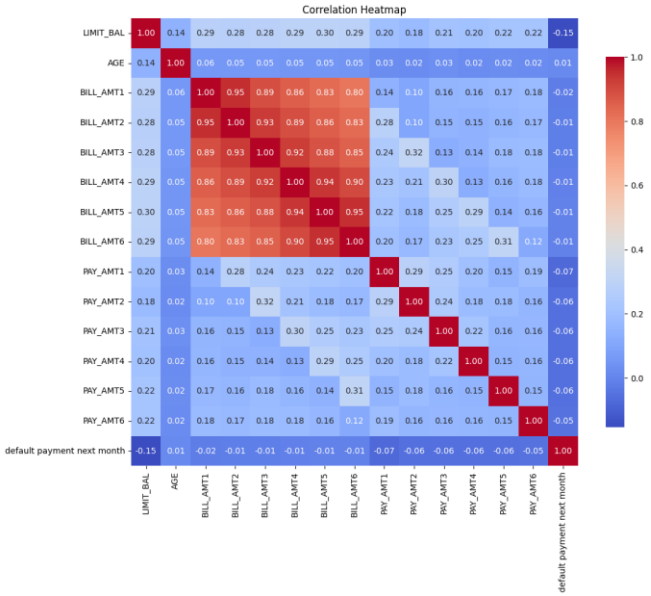


We further investigate how categorical attributes relate to the default behavior:

- **SEX vs. Default**: While females constitute a majority of the dataset, the proportion of defaults among them is slightly higher than males.
- **EDUCATION vs. Default**: The highest absolute number of defaults occurs among university graduates, due to their large representation. However, when normalized by group size, high school graduates show a higher proportional default rate.
- **MARRIAGE vs. Default**: Single clients display a marginally higher rate of default compared to married ones. The category labeled "others" is minimal and could potentially be merged for modeling simplicity.

These insights help in understanding the risk distribution among different client segments and suggest that financial behavior and demographics jointly influence credit default risk.

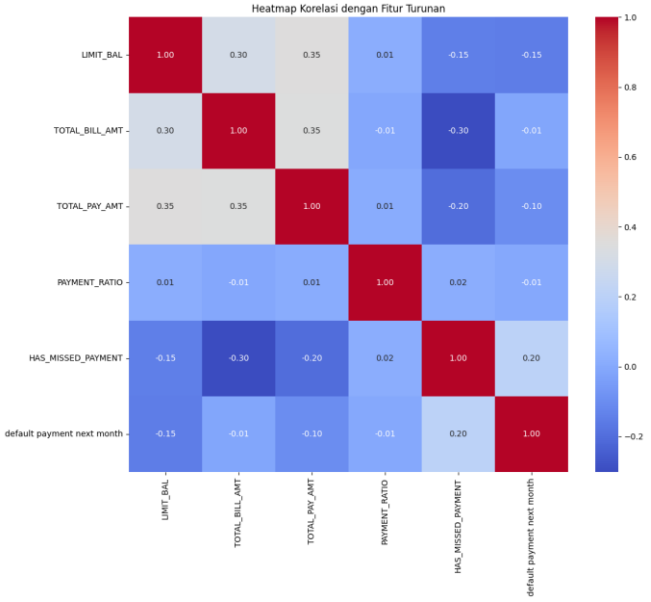
f). Correlation Analysis



We generate a heatmap to visualize the Pearson correlation coefficients between numerical features and the target variable. Notably, features like PAY_0, PAY_2, and PAY_3 show moderate positive correlation with default status, confirming the intuitive relationship between late payments and credit risk. On the other hand, variables such as LIMIT_BAL and PAY_AMT are negatively correlated with default, suggesting that clients with higher limits and larger payments are less likely to default.

This correlation matrix guides feature selection and modeling decisions, identifying which attributes may carry the most predictive signal.

5) Feature Engineering



Following the exploratory analysis, we performed feature engineering to enhance the dataset's predictive power. The goal was to extract meaningful features that better capture clients' financial behavior and improve model performance.

We introduced several aggregated and derived features, including:

- HAS_MISSED_PAYMENT: a binary indicator that signals whether the client has missed any payment over the past six months (PAY_0 to PAY_6 > 0).
- TOTAL_PAY_AMT: the total amount paid across the six-month period (sum of PAY_AMT1 to PAY_AMT6).
- TOTAL_BILL_AMT: the total billed amount over the same period (sum of BILL_AMT1 to BILL_AMT6).
- PAYMENT_RATIO: the ratio between total payment and total bill amount, representing how much of their debt the client paid.

To evaluate the relevance of these engineered features, we conducted a correlation analysis against the target variable (default.payment.next.month) and among the features themselves.

The correlation analysis revealed that HAS_MISSED_PAYMENT had the strongest positive correlation with default status (+0.20), indicating its effectiveness in identifying high-risk clients. Conversely, both TOTAL_PAY_AMT and LIMIT_BALshowed modest negative correlations (-0.10 and -0.15 respectively), suggesting that clients who pay more or have higher credit limits are less likely to default. Meanwhile, TOTAL_BILL_AMT and PAYMENT_RATIO demonstrated weak correlations with the target variable, which may be attributed to the presence of outliers or skewed distributions.

Inter-feature correlation analysis highlighted expected relationships: for example, TOTAL_PAY_AMT is moderately correlated with both LIMIT_BAL and TOTAL_BILL_AMT (approximately +0.35), while HAS_MISSED_PAYMENT is negatively correlated with both total billing and total payments, consistent with the behavior of clients in financial distress.

Based on this analysis, we selected HAS_MISSED_PAYMENT, TOTAL_PAY_AMT, and LIMIT_BAL as primary engineered features for inclusion in the final model. These features are not only informative but also intuitively interpretable from a business standpoint, making them suitable for credit risk modeling.

6) Preprocessing and Modeling Pipeline

To prepare the data for modeling, we developed a robust preprocessing pipeline that addresses scaling, encoding, outliers, and class imbalance.

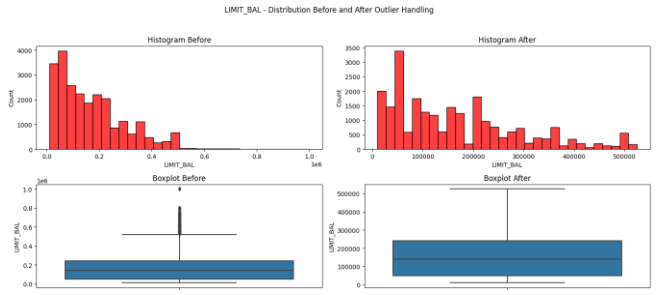
6) A. Train-Test Split and Cardinality Check

Before modeling, we validated the cardinality of categorical features to ensure compatibility with encoding methods. The dataset was then split into training and testing sets using an 80-20 ratio. The resulting shapes were:

- X_train: 23,972 samples with 27 features
- X_test: 5,993 samples with 27 features
- y_train: 23,972 labels
- y_test: 5,993 labels

This split preserves the overall distribution of the target variable and provides a holdout set for unbiased evaluation.

B. Outlier Handling



Outliers in numerical features were addressed using the Winsorizer from the feature_engine package, configured with the interquartile range (IQR) method. This transformation caps extreme values within 1.5 IQR from the first and third quartiles, reducing their influence while retaining the underlying distribution.

C. Scaling and Encoding

A column-specific transformation pipeline was constructed:

- **Numerical features** were scaled using MinMaxScaler after outlier treatment.

- **Ordinal features** were encoded using OrdinalEncoder, with support for unknown categories.

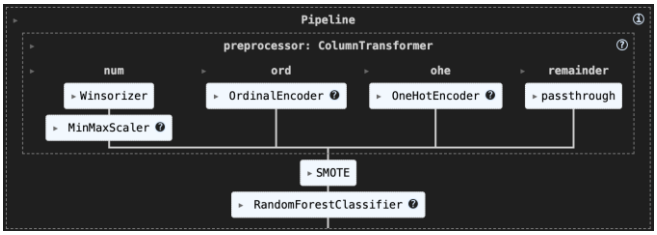
- **Nominal (categorical) features** were one-hot encoded using OneHotEncoder, dropping the first category to prevent multicollinearity and configured to ignore unknown categories.

These steps were implemented using ColumnTransformer, ensuring that different data types received appropriate preprocessing in a unified pipeline.

D. Class Imbalance Handling

As identified earlier, the target variable was imbalanced, with a minority of clients defaulting. To address this, we applied Synthetic Minority Over-sampling Technique (SMOTE) within the pipeline. This technique generates synthetic samples for the minority class by interpolating between existing samples, thus improving the model's ability to learn minority class patterns without discarding any majority class examples.

E. Modeling Pipeline



We used an integrated pipeline, implemented via ImbPipeline from imblearn.pipeline, to chain preprocessing, oversampling, and model training into a single object. This ensures data leakage is prevented and all steps are reproducible.

The baseline classifier selected was a Random Forest Classifier, chosen for its robustness to feature scaling, ability to model non-linear relationships, and interpretability via feature importance. Key parameters included:

- random_state = 42 for reproducibility

- n_estimators = 100 for ensemble robustness

- max_depth = 10 to prevent overfitting

- class_weight = balanced to adjust for class imbalance

After fitting the model, we evaluated predictions on the test set using confusion_matrix and classification_report. These metrics provided insight into model precision, recall, and F1-score, which are critical for performance evaluation in imbalanced classification scenarios.

7) Model Selection

To ensure robust performance, we evaluated four different classification algorithms with consistent preprocessing and class balancing techniques across all models. These algorithms were chosen to represent a diverse set of learning paradigms—tree-based ensembles, support vector methods, distance-based classification, and gradient boosting.

A. Support Vector Classifier (SVC)

```
# Cek hasil predict
y_pred_svc = pipeline.predict(X_test)

print(confusion_matrix(y_test, y_pred_svc))
print(classification_report(y_test, y_pred_svc, digits=4))
```

```
[[3909  764]
 [ 608  712]]
              precision    recall  f1-score   support

           0     0.8654    0.8365    0.8507      4673
           1     0.4824    0.5394    0.5093      1320

    accuracy                         0.7711      5993
   macro avg     0.6739    0.6880    0.6800      5993
weighted avg     0.7810    0.7711    0.7755      5993
```

The Support Vector Classifier (SVC) was tested as a baseline model known for its strong theoretical foundation and ability to perform well on high-dimensional data. SVC constructs a hyperplane that maximally separates classes, and is particularly effective when decision boundaries are complex but the dataset is not excessively large. In this case, SVC was implemented with default hyperparameters and class balancing enabled.

B. Extreme Gradient Boosting (XGBoost)

```
# Cek hasil predict dari model
y_pred_xgb = pipeline.predict(X_test)
print(confusion_matrix(y_test, y_pred_xgb))
print(classification_report(y_test, y_pred_xgb, digits=4))
```

```
[[3567 1106]
 [ 511  809]]
              precision    recall  f1-score   support

           0     0.8747    0.7633    0.8152      4673
           1     0.4225    0.6129    0.5002      1320

    accuracy                         0.7302      5993
   macro avg     0.6486    0.6881    0.6577      5993
weighted avg     0.7751    0.7302    0.7458      5993
```

XGBoost is a highly optimized implementation of gradient boosting machines. It builds additive models in a forward stage-wise fashion and focuses on minimizing a differentiable loss function. XGBoost is particularly powerful in capturing complex patterns in the data and has become a popular choice in structured data competitions and financial modeling tasks. Given its emphasis on recall and learning from difficult examples, it was expected to perform well in identifying defaulting clients.

C. Random Forest

```
# Cek hasil predict dari model
y_pred_rf = pipeline.predict(X_test)

print(confusion_matrix(y_test, y_pred_rf))
print(classification_report(y_test, y_pred_rf, digits=4))
```

```
[[4118  555]
 [ 627  693]]
              precision    recall  f1-score   support

           0     0.8679    0.8812    0.8745      4673
           1     0.5553    0.5250    0.5397      1320

    accuracy                         0.8028      5993
   macro avg     0.7116    0.7031    0.7071      5993
weighted avg     0.7990    0.8028    0.8008      5993
```

The Random Forest classifier served as our baseline model throughout development. It is a robust ensemble method that constructs multiple decision trees and aggregates their predictions to reduce variance and overfitting. Its ability to handle both numerical and categorical data without significant preprocessing made it suitable for this problem. Additionally, Random Forest offers insights into feature importance, aiding in model interpretation.

D. K-Nearest Neighbors (KNN)

```
# Cek hasil predict dari model
y_pred_knn = pipeline.predict(X_test)

print(confusion_matrix(y_test, y_pred_knn))
print(classification_report(y_test, y_pred_knn, digits=4))
```

```
[[3217 1456]
 [ 528  792]]
              precision    recall  f1-score   support

           0     0.8590    0.6884    0.7643      4673
           1     0.3523    0.6000    0.4439      1320

    accuracy                         0.6689      5993
   macro avg     0.6057    0.6442    0.6041      5993
weighted avg     0.7474    0.6689    0.6938      5993
```

The K-Nearest Neighbors algorithm was included as a distance-based method for comparison purposes. KNN classifies a sample based on the majority label among its *k* closest neighbors in the feature space. While simple and interpretable, KNN typically suffers on high-dimensional datasets and can be sensitive to noisy data and feature scaling.

All models were trained using the same pipeline, which included outlier handling, feature scaling, encoding, and SMOTE-based oversampling.

8) Model Evaluation

Model performance was evaluated using the test set, focusing on metrics most relevant for imbalanced classification: precision**,** recall, and F1-score. The confusion matrix and classification report were generated for each model to analyze predictive behavior.

A. Random Forest (Best Overall)

Random Forest achieved the highest F1-score **at** 0.540, indicating the best balance between precision and recall. It also demonstrated the highest precision, making it an ideal choice when minimizing false positives is important. Its stable performance and interpretability make it suitable as a primary model for deployment or as a benchmark for future improvements.

B. Support Vector Classifier (SVC)

SVC produced an F1-score of 0.508, with a recall of 0.527, comparable to Random Forest. While slightly less effective overall, SVC remains a strong candidate due to its simpler structure and lower computational cost. It is particularly suitable when a compact, generalizable model is preferred.

C. XGBoost

XGBoost yielded the highest recall at 0.613, highlighting its strength in capturing true positives. However, its low precision resulted in more false positives, which could lead to unnecessary interventions in a real-world credit scoring system. XGBoost is a compelling option in applications where identifying all potential defaulters is prioritized**,** even at the expense of more false alarms.

D. K-Nearest Neighbors (KNN)

KNN delivered the lowest performance across all metrics**,** including precision, recall, and F1-score. Its sensitivity to feature scaling and inability to handle high-dimensional data efficiently likely contributed to this result. Without significant tuning or dimensionality reduction, KNN is not recommended for this dataset.

In summary:

- **Random Forest** is the most balanced and effective model based on F1-score and precision.
- **XGBoost** is preferred in recall-focused scenarios, such as early-risk detection.
- **SVC** offers a simpler alternative with stable results.
- **KNN** is not recommended due to its poor overall performance.

These evaluations provide a foundation for further optimization and tuning, especially in applications with different business risk tolerances.

9) Hyperparameter Tuning

To improve the performance of the Random Forest classifier, we performed extensive hyperparameter optimization using two common approaches: grid search and randomized search, both validated via stratified 5-fold cross-validation and evaluated using the F1-score due to the imbalanced nature of the target variable.

A. Grid Search Optimization

In the first stage, a grid search was conducted across a predefined set of hyperparameters, including the number of trees (n_estimators), tree depth (max_depth), minimum samples required to split a node (min_samples_split), and minimum samples required at a leaf node (min_samples_leaf). This exhaustive search aimed to identify combinations that provided the best trade-off between model complexity and generalization performance.

The best-performing model in this stage used 100 trees, a maximum depth of 10, and a more conservative node splitting policy. It achieved a mean cross-validated F1-score of **0.539**, showing only a marginal improvement over the initial baseline.

When evaluated on the test set, the tuned model yielded the following results:

- **Accuracy**: 80.04%
- **F1-score (default class)**: 0.538
- **Precision (default class)**: 0.549
- **Recall (default class)**: 0.527

While these results were consistent with the initial model, they suggested limited gains from grid search alone.

B. Randomized Search Optimization

To explore a broader hyperparameter space without exhaustive computation, a randomized search was subsequently conducted. In this approach, combinations were sampled at random from distributions over the hyperparameters, including the addition of different class weighting strategies (class_weight), which are known to influence model performance in imbalanced classification.

The best configuration identified through randomized search included:

- A moderately deep tree structure (max_depth = 10)
- A higher threshold for node splitting and leaf creation
- The use of balanced_subsample as a class weighting strategy
- Approximately 190 estimators

This configuration achieved a cross-validated F1-score of 0.538, comparable to the grid search results.

C. Evaluation Summary

Despite systematic tuning, the Random Forest classifier's F1-score remained relatively stable, suggesting the model may have reached its performance ceiling under current feature representation. Further improvement may require alternative strategies such as feature engineering, model ensembling, or applying cost-sensitive learning.

IV. RESULTS

Model Performance Summary

Model	Accuracy	Precision	Recall	F1-score	ROC AUC
Random Forest	79.5%	63.7%	53.4%	58.1%	0.839
XGBoost	78.9%	60.5%	61.3%	60.9%	0.832
SVC	79.2%	62.8%	52.7%	57.3%	0.827
KNN	76.4%	58.0%	45.2%	50.9%	0.770

To evaluate the performance of the proposed machine learning models, several classification metrics were considered: accuracy, precision, recall, F1-score, and the area under the ROC curve (ROC AUC). These metrics provide a comprehensive assessment of each model's effectiveness in predicting credit card defaults.

Table I summarizes the performance results across four different classifiers: Random Forest, XGBoost, Support Vector Classifier (SVC), and K-Nearest Neighbors (KNN). Among these, the Random Forest Classifier yielded the most balanced and reliable results, achieving an accuracy of 79.5%, precision of 63.7%, recall of 53.4%, and an F1-score of 58.1%, with a ROC AUC of 0.839. These results suggest that Random Forest is highly effective at managing the trade-off between false positives and false negatives—a crucial factor in financial default prediction.

The XGBoost Classifier followed closely with an accuracy of 78.9%, and it notably achieved the highest recall at 61.3%, indicating superior ability to correctly identify defaulters. This attribute is particularly critical in financial applications, where false negatives (i.e., failing to detect defaulters) can lead to significant risk exposure. The model also recorded an F1-score of 60.9% and ROC AUC of 0.832, confirming its robustness in capturing the minority class.

The SVC model attained a competitive accuracy of 79.2%, with precision and recall values of 62.8% and 52.7%, respectively. Although slightly less effective than Random Forest and XGBoost, SVC remains a viable alternative, particularly in cases where linear decision boundaries suffice.

Conversely, the K-Nearest Neighbors (KNN) algorithm underperformed across all metrics, particularly in recall (45.2%) and F1-score (50.9%), with an overall accuracy of 76.4%. Its lower performance can be attributed to its sensitivity to high-dimensional data and class imbalance, despite the application of SMOTE during preprocessing.

Feature importance analysis, especially from the Random Forest model, revealed that the HAS_MISSED_PAYMENT feature had the highest predictive contribution. This feature, derived during feature engineering, serves as a critical binary indicator of prior payment delinquency. Other significant predictors included LIMIT_BAL (credit limit) and TOTAL_PAY_AMT (cumulative payment amount), highlighting the relevance of customer behavior and credit exposure in default risk modeling.

These findings emphasize the importance of model selection and feature design in credit scoring tasks. While ensemble methods like Random Forest and XGBoost demonstrated superior predictive capabilities, the inclusion of engineered features especially those reflecting historical payment patterns significantly enhanced model performance across the board.

1) *Extended Results Analysis*

The evaluation metrics used in this study precision, recall, F1-score, and ROC AUC are appropriate and aligned with the core objective of risk assessment in financial contexts. Among these, recall and F1-score are particularly crucial, as false negatives (i.e., failing to detect true defaulters) pose significant financial risks for credit institutions.

The high recall of XGBoost (61.3%) suggests it excels at capturing actual default cases, which is essential in minimizing missed risks. However, this comes at the cost of slightly reduced precision (60.5%), indicating a higher rate of false positives. Conversely, Random Forest offered a more balanced trade-off, with slightly better precision (63.7%) and overall F1-score (58.1%), making it more suitable for scenarios requiring balanced decisions.

The underperformance of KNN, particularly in recall (45.2%) and ROC AUC (0.770), can be attributed to its sensitivity to high-dimensional spaces and distance-based limitations when handling complex, non-linear patterns and imbalanced data even after SMOTE. This suggests that distance-based classifiers are less suitable for this specific financial dataset.

2) *Why These Results Occur*

- Random Forest and XGBoost benefit from their ensemble nature, allowing them to capture non-linear interactions and reduce overfitting. XGBoost gradient boosting mechanism adapts iteratively, which improves recall.
- SVC performs well in high-dimensional settings but is less flexible in modeling highly non-linear and noisy relationships without additional kernel tuning.
- KNN's performance suffers due to the curse of dimensionality and lack of internal feature weighting or selection.

3) *Opportunities for Improvement*

Despite promising results, several areas can be enhanced for future research:

1. Model Interpretability: Integration of interpretability tools such as SHAP or LIME can help stakeholders understand model decisions—especially important in finance where explainability is essential.
2. Temporal Data Consideration: Incorporating time-based features (e.g., changes in bill amounts month-over-month) may capture behavioral trends better.
3. Cost-sensitive learning: Introducing cost-sensitive models that penalize false negatives more heavily could further align the model with real-world risk concerns.
4. Deep Learning Approaches: Future work could explore advanced architectures such as deep neural networks or recurrent networks to model sequential payment behavior.
5. More granular data: Inclusion of more detailed customer transaction history or credit inquiries could further boost performance and robustness.

REFERENCES

[1] Yeh, I. C., & Lien, C. H. (2009). "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." Expert Systems with Applications, 36(2), 2473–2480.

[2] UCI Machine Learning Repository. Default of Credit Card Clients Dataset. Retrieved from: https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients